SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3254

Evaluacija metode za ispravljanje pogrešaka kod dugačkih očitanja

Dario Pavlović

Zagreb, lipanj 2013.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

Veliko hvala mentoru Mili Šikiću na strpljenju, pomoći, podršci i pljeskavicama. Hvala, također, i kolegi Matiji Korparu koji je uvijek bio u blizini za mnogobrojna pitanja i nedoumice.

SADRŽAJ

1. Uvod

Bioinformatika jedna je od brže rastućih suvremenih znanstvenih disciplina. Ona ujedinjuje nekoliko važnih područja – biologiju, računarstvo, informacijsku tehnologiju te vjerojatnost i statistiku. Unutar bioinformatike izdvaja se nekoliko važnih, a još uvijek neriješenih, problema. Neki od njih jesu poravnanje slijedova na razini kromosoma, uspješno mapiranje podataka dobivenih RNA sekvenciranjem, razumijevanje i analiza proteinskog savijanja te *de novo* sastavljanje genoma. Potonje je ujedno i tema ovoga rada. *de novo* asembliranje transkriptoma jest postupak kreiranja, to jest rekonstrukcije genoma bez pomoći referentnoga, originalnog genoma. Naime, današnje metode sekvenciranja genoma ne mogu pročitati cijeli genom nekog organizma odjednom, već ga čitaju u komadićima različitih duljina uz mogućnost pojave raznih pogrešaka. Uobičajeno je da su duljina očitanja i postotak pogrešaka u korelaciji – što dulje očitanje, to veći broj pogrešaka prilikom čitanja. Iz takvih se očitanja zatim pokušava računalnim algoritmima rekonstruirati originalni genom. Jedan od načina *de novo* sastavljanja genoma naziva se Preklapanje-Izravanavanje-Konsenzus (engl. *Overlap-Layout-Consensus, OLC*). Postupak je to, kako mu i ime kaže, u tri koraka koji će detaljnije biti opisani u nastavku. U ovom radu, pokušat ću analizirati kako duljina pojedinog očitanja uparena s određenim postotcima grešaka utječe na točnost i mogućnosti izvođenja prve faze, to jest detekcije preklapanja. Cilj rada jest analizirati ovaj odnos te ispitati postoji li optimalan par duljina-pogreška koji omogućava zadovoljavajuće performanse uz dobre rezultate. Rad je podijeljen na pet poglavlja. Poglavlje Sastavljanje genoma opisuje općenito kako se danas pristupa sastavljanju genoma te neke karakteristične probleme na koje nailazimo prilikom toga. Drugo poglavlje govori nešto detaljnije o podacima koje sam koristio. Poglavlje Metode opisuje koje sam postupke koristio prilikom izvođenja eksperimenata na očitanjima te računalne alate pomoću kojih sam to provodio. U poglavlju rezultati bit će izložene statistike izvučene iz eksperimenata i pretočene u grafove i tablice. U poglavlju Diskusija bit će učinjen osvrt na rezultate te diskusija o budućnosti sastavljanja genoma i planiranim novim mjerenjima.

2. Sastavljanje genoma

Nukleinske kiseline – RNA i DNA – najvažnije su biološke molekule i osnovica živog svijeta. U njima je pohranjeno naslijeđe svakog organizma. Kako bi uopće bilo moguće shvatiti način i principe funkcioniranja živih bića, nužno je razumjeti strukturu i ulogu ovih molekula i njihovih sastavnih dijelova – nukleotida. Cilj je sastavljanja genoma doći do što više informacija koje genom sadrži te, naravno, potpuna reprezentacija genoma u obliku pogodnom za daljnja istraživanja. Da bi to bilo ostvareno, razvijeno je mnoštvo različitih postupaka očitavanja strukture genoma te sastavljanja iste računalnim metodama. Općenito, suvremeni postupak sastavljanja genoma nekog organizma može se opisati u nekoliko koraka [1].

Prvi korak jest genom rastaviti na velik broj fragmenata čije su veličine slučajno odabrane. Zatim se s početka i kraja (prefiksa i sufiksa) tako dobijenih fragmenata čitaju nukleotidi. Očitanje, to jest poredak nukleotida, sprema se u računalno pogodan oblik. Duljina fragmenata prilagođena je ovom postupku te je uvijek veća od najvećeg broja nukleotida koji je današnjom tehnologijom moguće pročitati iz njih. Ovaj korak provodi se onoliko puta koliko je potrebno da se cijeli genom pročita više puta ili matematički rečeno, kako bi se sa što većom vjerojatnošću osiguralo da je svaki nukleotid genoma pročitan barem jednom, to jest, da nije izostavljen. Faktor koji izražava koliko je puta cijeli genom uzorkovan na ovaj način naziva se pokrivenost (engl. *coverage*), a računa se kao omjer umnoška broja očitanja i njihove prosječne duljine te ukupne duljine genoma.

$$C = \frac{N \cdot L}{G} \tag{2.1}$$

Jednadžba 2.1 prikazuje izrečeno. U jednadžbi je s N označen broj očitanja, L predstavlja prosječnu duljinu očitanja, a G duljinu genoma koji se čita. Naravno, pokrivenost ne garantira očitavanje svakog nukleotida, već izražava vjerojatnost čitanja svakog nukleotida. Ovaj pristup čitanja strukture DNA, uz neke izmjene i poboljšanja, koristi se već tridesetak godina, a predložio ga je Frederick Sanger.

Nakon što prvi korak završi, dobije se velik broj očitanja, također različitih duljina.

Količina očitanja može se ilustrirati jednostavnim primjerom iz stvarnog života. Genom bakterije E.coli sadrži otprilike pet milijuna baza. Ako je tražena pokrivenost od tri puta uz prosječnu duljinu očitanja tisuću petsto baza, dolazimo do brojke od deset tisuća očitanja. Očitanja je zatim potrebno nekako spojiti, to jest iz njih sastaviti genom. Za to su zaduženi računalni programi zvani asembleri. Asembleri rade na različite načine, ovisno o tehnici sastavljanja u kojoj se primjenjuju. U današnje vrijeme, izdvajaju se dva glavna pristupa ovom problemu, detaljnije objašnjena u nastavku.

2.1. Komparativno sastavljanje

Prvi pristup sastavljanju genoma naziva se komparativni. Pristup je to mapiranja dobijenih očitanja na referentni, već postojeći i rekonstruirani genom. Ovaj se način većinom koristi kod sastavljanja genoma koji je sličan nekom već postojećem, sastavljenom genomu. Na primjer, kod sastavljanja genoma različitih podtipova iste bakterije ili genoma srodnih, bliskih vrsta. Ovaj postupak, grubo opisano, ima dva značajna koraka nakon dobijanja očitanja. Prvo se očitanja poravnaju s referentnim (engl. *target*) genomom i tako postave na pozicije na kojima najbolje odgovaraju. Zatim se takva poravnanja grupiraju i spajaju u veće cjeline s ciljem potpune rekonstrukcije željenog genoma. Ovaj pristup ima svoje probleme koji proizlaze iz nekoliko čimbenika. Na primjer, često nije unaprijed poznata duljina genoma koji se želi sastaviti, nije poznato u kojoj mjeri bi mogao biti sličan referentnom genomu i je li mu uopće sličan. Dodatno, postojanje raznih pogrešaka u samim očitanjima, ali i referentnom genomu [2] može znatno otežati ovaj proces. K tome, referentni genom i genom koji se rekonstruira mogu biti značajno različiti u nekim regijama, a to znači da takve regije nemaju dovoljno dobra poravnanja te se mora pribjeći drugim načinima rekonstrukcije. Ipak, komparativno sastavljanje ima niz mogućnosti i primjena u praktičnim istraživanjima. Kao primjer navodim sekvenciranje i sastavljanje genoma raznih vrsta bakterija koje izazivaju bolesti kako bi se razumjele promjene i mutacije u njihovim podtipovima. Za istraživanje otpornosti na lijekove i pronalazak novih lijekova, ovo je od velike važnosti [2]. Ipak, samo po sebi, komparativno sastavljanje nije dovoljno za rekonstrukciju svakoga genoma jer je živi svijet prilično raznolik što implicira velike razlike i na mikrorazini.

2.2. De novo sastavljanje

Drugi način sastavljanja genoma jest takozvani *de novo* pristup. *De novo* je pristup kod kojega se genom sastavlja bez pomoći bilo kakvog referentnog, već sastavljenog genoma. Odmah je uočljivo da je ovakva metoda znatno složenija od prethodne. *De novo* pristup temelji se na ideji traženja najkraćeg zajedničkog višekratnika svih očitanja, u smislu niza znakova (engl. *superstring*) [3]. Štoviše, dokazano je da *de novo* rekonstrukcija pripada u razred NP-teških (engl. *NP-hard*) problema [4][5]. NP-teški problemi jesu vrlo složeni i računalno zahtjevni. Ipak, s obzirom na raznolikost živoga svijeta, ovaj pristup jedini omogućuje potencijalno sekvenciranje i sastavljanje svih vrsta genoma iz prirode jer su oni uglavnom međusobno prilično različiti (izuzev bliskih vrsta i podtipova istog soja) i nije moguće rekonstruirati jedan koristeći drugi kao referentan. Treba napomenuti da *de novo* pristup i komparativno sastavljanje nisu međusobno isključivi već se nadopunjuju. Za primjer je lako uzeti slučaj kad kod komparativnog pristupa postoje pojedine regije dvaju genoma koje su jako različite, što je već spomenuto kao problem komparativnog pristupa. Tipično se onda za rekonstrukciju takvih područja koristi *de novo* postupak [2].

De novo sastavljanje genoma ima pred sobom nekoliko značajnih izazova koji se svladavaju na razne načine. Prvi problem jesu duže ponavljajuće regije unutar DNA niza koji se očitava. Duže ponavljajuće regije područja su u nizu nukleotida koja su identična, a postoje na većem broju mjesta u lancu. Često, pojava takvih regija dovodi do "zbunjivanja" asemblera jer on u takvim slučajevima može previdjeti višestrukost položaja takvih regija. Zbog toga, asembler stvori više slijednih sekvenci (engl. *contigs*) nego bi trebao što kasnije uzrokuje probleme. Zbog toga su za *de novo* pristup pogodnija što je moguće duža očitanja jer ona omogućavaju lakšu detekciju i točnije raspoređivanje ponavljajućih regija. Drugi izazov predstavlja pitanje kako odrediti koji je faktor pokrivenosti optimalan prilikom čitanja genoma da slijednih sekvenci bude što je moguće manje. Eric Lander i Michael Waterman pokazali su da koeficijent pokrivenosti između osam i deset garantira da broj slijednih sekvenci teži k jako malom broju (pet slijednih sekvenci za genom duljine $1Mbp$) [6].

Nadalje, dobro je pogledati različite načine samog izvođenja *de novo* sastavljanja genoma unutar asemblera i u okviru toga izložiti neke dodatne probleme, što ću i učiniti u nastavku. Generalno gledano, izdvajaju se dvije glavne kategorije asemblera – oni zasnovani na pohlepnom pristupu (engl. *greedy*) i asembleri bazirani na grafovima [7]. Razmotrit ću prvo pohlepni pristup.

Pohlepni je pristup zasnovan na jednostavnoj ideji. Algoritam prolazi kroz skup

svih očitanja i u svakom prolazu traži dva očitanja koja se najbolje preklapaju. Takva dva preklapanja algoritam združi i nastavlja dalje te ovo ponavlja dok god postoje nepovezana očitanja. Nakon združivanja dva očitanja provode se i dodatne heurističke ispravke [7]. Tako se iterativno gradi cijeli genom. Odmah je vidljivo kako ova metoda može lako dovesti do pogrešaka i krivog redoslijeda prilikom sastavljanja zato što se u svakom koraku gledaju isključivo lokalni optimumi najbolje sličnosti, a to ne može garantirati globalnu optimalnost i točnost. Zbog toga se pohlepni pristup i rjeđe koristi, iako je računalno puno jednostavniji nego ostali.

Algoritmi temeljeni na grafovima predstavljaju preklapanja kao usmjerene netežinske grafove. Postoje dvije varijante ovih algoritama – OLC i sekvenciranje po hibridima ili SBH (engl. *sequencing-by-hybridization*) [7]. SBH stvara graf na način da čvorovima grafa predstavlja preklapanja, a bridovima ostatak preklapajućih očitanja. Ovo na kraju vodi na traženje Eulerovog puta kroz graf, što u mnogim varijantama predstavlja NP-težak problem zbog pogrešaka i drugih problema u očitanjima [7]. Graf kod OLC-a čine očitanja kao čvorovi i preklapanja kao veze. Preklapanje-Izravnavanje-Konsenzus ujedno je i najpopularniji algoritam sastavljanja genoma. Algoritam je to koji se provodi, kao što mu i ime kaže, u tri glavna koraka. Prvi korak ove metode praktički je identičan načinu na koji funkcionira pohlepni pristup. Pretražuju se očitanja i međusobno uspoređuju. Ovaj dio može biti izveden na više načina. Moguće je uspoređivati svako očitanje sa svakim drugim, no tu automatski nastupa problem kvadratne složenosti koja je nezanemariva. Na primjer, prilikom sekvenciranja i sastavljanja genoma vinske mušice stvoreno je oko 3×10^6 očitanja prosječne duljine 500 baza, što je iziskivalo oko 10^{13} usporedbi [8]. Problem kvadratne složenosti doveo je do rađanja drugih načina pristupa istome problemu. Jedan od poznatijih jest pristup u kojem se prije usporedbe dvaju očitanja zahtjeva da ta očitanja dijele isti, kraći, niz baza fiksne duljine unutar svoje strukture (engl. *seed*). Ako dva očitanja nemaju barem jedan isti takav niz, ona se ne uspoređuju. Ipak, ni ova metoda ne može riješiti sve probleme složenosti prvog koraka te je to i dalje računalno najskuplji i najzahtjevniji dio cijelog *OLC* postupka.

Sljedeći je korak izravnavanje tako dobijenoga grafa ili *layout*. U ovom koraku cilj je izgraditi put kroz graf koji kroz svaki čvor, to jest preko svakog očitanja prolazi samo jednom. Matematički gledano, traži se Hamiltonov ciklus u grafu što predstavlja dodatni izazov u *de novo* pristupu jer je problem nalaženja Hamiltonovog ciklusa dokazano NP-težak, a to znači da pripada u isti razred problema kao i sam *de novo* postupak. Kako bi se doskočilo ovom problemu, koristi se postupni pristup izravnavanju grafa[1]. Prvo se regije grafa koje su sigurno nastale od istog kontinuiranog

dijela genoma, slijedna sekvenca ili *contig* spoje u jedan čvor. Preciznije rečeno, ako čvor ima samo jednog susjeda u jednom smjeru, on s tim susjedom postaje dio slijedne sekvence. Ovo se rekurzivno ponavlja za sve čvorove, a sekvenca završava nakon što se naiđe na čvor koji ima dva ili više susjeda. Praktično gledano, čvor s dva ili više susjeda predstavlja granicu ponavljajuće regije i različitih nastavaka na takvu regiju kroz cijeli genom. Rezultat je ovog koraka veliki broj slijednih sekvenci spremnih za daljnje korake slaganja i spajanja. Veće cjeline dobijene spajanjem slijednih sekvenci nazivaju se okviri (engl. *scaffolds*). S obzirom da u ovom trenutku izvođenja postupka još uvijek nije poznata struktura genoma između pojedinih slijednih sekvenci, različiti asembleri oslanjaju se na različite druge informacije koje mogu dobiti iz prethodnih koraka kako bi ispravno poredali slijedove. Popularan način dobijanja više informacija jest grupirati pojedina očitanja u srodne parove (engl. *mate pairs*, i to na način da jedan takav srodni par čine očitanja dobijena u prvom koraku od istog isječka genoma, sa svakog kraja isječka. Time postaje moguće odrediti relativne položaje slijednih sekvenci jednih prema drugima što olakšava poredak u ispravan redoslijed. Ovdje se, zbog težine problema, također koriste različita ubrzanja i heurističke metode.

Završna faza OLC postupka jest faza konsenzusa. Ovaj korak služi određivanju koja će baza biti smještena na koje mjesto u genomu između više potencijalnih kandidata. Naime, s obzirom na faktor pokrivenosti koji nam govori koliko je puta genom "presekveniciran", često dolazi do pojave da se na isto mjesto u genomu poreda više očitanja. U tome slučaju, potrebno je napraviti dogovor ili konsenzus o tome koje će od tih očitanja odrediti nukleotid na tom mjestu. Nakon ovoga koraka, može se reći da je genom sastavljen. Ipak, s obzirom na količinu problema koje asembler susreće prilikom sastavljanja genoma i mogućnosti pogrešaka u raznim koracima rada asemblera ili pogrešaka u ulaznim podacima, provode se različiti postupci ocjene kakvoće dobijenog genoma. Postoje razni pristupi ovom problemu [1], počevši od statičkih preko empirijskih. Vrlo dobra metoda jest preko već spomenutih parova očitanja. Ako njihov poredak nije očuvan kroz genom na način na koji oni to zahtijevaju kao parovi dobijeni od istih fragmenata, sigurno je postojanje pogrešaka u sastavljanju.

U nastavku rada bit će stavljena posebna pozornost na mogućnosti prve faze, to jest preklapanja. Cilj će biti, kako je u uvodu napomenuto, ispitati kako dužina očitanja uparena s određenim postotkom pogreške na očitanjima utječe na preciznost i mogućnosti overlapa. Bolje rečeno, cilj rada jest vidjeti kako se mijenja postotak ispravno detektiranih preklapanja u ovisnosti o parametrima pogreške te medijana duljine očitanja.

3. Podaci

U živim bićima pohranjena je doista ogromna količina informacija. Baze podataka s kojima bioinformatika radi impozantnih su veličina i svaki dan sve veće kako se otkriva sve više novih i novih znanja. Porast količine znanja je praktički eksponencijalan s tendencijom daljnjeg porasta. No, nije problem samo količina znanja, već i njegova raznolikost. Primjerice, kod genoma, važno je nekako zapisati redoslijed nukleotida, kvalitetu očitanja, početne i konačne pozicije očitanja, taksonomiju živog bića čija su to očitanja. Kod proteina i aminokiselina naglasak je na zapisivanju strukturne informacije. U nekim drugim primjenama važno je međusobno uspoređivati pojedine nizove.

U samim počecima bioinformatike, pokušalo se raditi s jednostavnijim podatkovnim formatima, običnim tekstualnim datotekama s deskriptivnim nazivima. To je dovelo do problema čim je kombinacija količine i raznolikosti znanja počela postavljati nepremostive izazove za takve načine spremanja podataka. Zbog toga, razvijeno je mnoštvo različitih načina spremanja podataka u raznim formatima. Neki od formata jesu MAF, PHYLIP, REF, FASTQ, FASTA. FASTA je ujedno najpopularniji format spremanja informacije o sekvencama danas. U svom radu primarno koristim FASTA i FASTQ formate koje nešto detaljnije opisujem u nastavku.

3.1. FASTA format

Datoteke u FASTA formatu sastoje se od dva dijela. Prvi dio je jedan redak koji se naziva linija definicije. Linija definicije obavezno počinje znakom “>“ nakon kojega, također obavezno, slijedi identifikator sekvence. Dodatno, u njoj mogu biti zapisani podaci kao što su taksonomska klasifikacija organizma kojem pripada slijed, položaj niza u genomu ili nekom od proteinskih lanaca te eventualno naznaku radi li se o kompletnom genu ili potpunom lancu aminokiselina. Treba napomenuti da oblik definicijske linije također slijedi neka važna pravila ovisno o standardu. Na primjer, NCBI (engl. *National Center for Biotechnology Information*) koristi vlastiti standard koji od

```
>gi|10957398|ref|NC_000958.1| Deinococcus radiodurans R1 plasmid MP1, complete sequence
ATTTTGACCCCAAATCCCGCAAAGGTGTCGCTATTTTGACCCCAAATCCCGCAAAGGTGTCGCTAAGAAC
GGCTAGAGTCACCGAAATATCACTTCTTGGCCTTTTGCACGTTGCCATCGGCCTTTTTCCGTTTCCCGCA
AAGGTGTCGCTATTATGCAGGTGGCTGAATAACGCAAAGGTGTCGCTATTTTGACCCCAAATCCCGCAAA
GGTGTCGCTATTCCTGAGAATGGAAAATTTGCCGTCTACAAAGAGAATTTTCTGATGGCTCATACTTTTC
CCGCAAAGGTGTCGCTAGCTCGCCTTTTGATTTTCACTTTGTCAGTAAAGTTATACCGCAAAGGTATCGC
TACTTGTAGGGATGTTTTGGGGAAGTAAGCAACCCCTATGGTCTGCAACTCGTGCTTGTAAAATTCACCA
ACCTTTTGATTGGTTGAGCGCCTGCTGCGGAGTAGAGCCCCAGTTGGGGGCTGATGAGTTCCCATGTTCG
TACATATACGAACGCTTACACTGAGACATATGACGACCATCTTGACGGTTTTTACCCACGCGGGTGGAGC
TGGGAAAACATCGATTGCAGGGAACATTGCACACGAATTCGCCCAGCGTGGACAACACGTGTTACTGATA
GATGGTGATCCACAAAGCAACCTCACCACCAACATGGGCGTTCAAGACGCGGAGTTGCATGAAACTCTGT
TCGATGTGCTGAGCGGTGACGCGCCTCTTCCTGCTCCCCGGCATGTCCACGGCTTTGACCTCATTCCCGC
TGTGATTGATCTGGCAGAAGTCGAGCCGAGCATTCCTGGTCGGGTTGGTGGCATCCTTGCGCTGCGTGAC
GCCCTTCAAAAAGAGAGTGGCCGCTGGGACACTGTAATTATTGACAGTCCCCCTTCTCTCGGGCAGTTGG
```

Slika 3.1: Primjer FASTA formata

definicijske linije zahtijeva da bude sastavljena od barem tri neizostavna dijela.

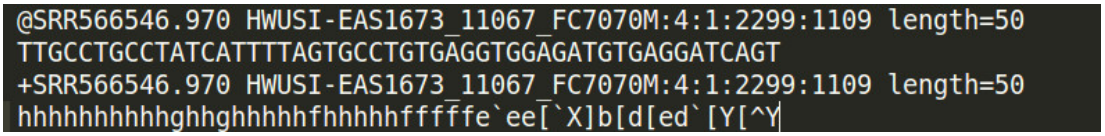
Prvi dio je referentni broj, drugi je pristupni broj,[1] a treći lokus ili položaj slijeda u nizu iz kojeg je dobijen. Nakon same definicijske linije, dolazi slijed nukleotida ili aminokiselina pohranjen po recima. Reci su uobičajeno širine između 60 i 80 znakova. Najčešće, radi se o brojkama 60, 70, 72 i 80. Jedna datoteka može sadržavati više opisanih dvodijelnih unosa, što se tad naziva multiFASTA format. FASTA format pogodan je, također, i za prikaz poravnanja između dva slijeda. Slika 3.1 ilustrira kako izgleda tipični zapis u FASTA formatu.

3.2. FASTQ format

FASTQ format najpopularniji je oblik pohrane očitanja sekvenci danas. Problem kod, na primjer, FASTA formata je što nema predviđeno mjesto za pohranu ocjene kvalitete očitanja pojedine baze što je često važan parametar. Kako bi se to omogućilo, kreiran je FASTQ format. Danas postoje tri varijante FASTQ formata [9]. Originalni FASTQ naziva se Sanger Fastq, a na osnovu njega je kasnije kreiran Solexa Fastq te iza toga i Illumina Fastq. Najvažnija, i praktički jedina razlika ova tri formata jest u funkcijama kojima se opisuju kvalitete očitanja te u zapisu tih kvaliteta u samoj datoteci, to jest kodiranju. To ih, logično, čini međusobno nekompatibilnima.

Svaki zapis u FASTQ datoteci podijeljen je na četiri dijela. Prvi redak zapisa obvezatno počinje znakom "@" nakon kojeg slijedi proizvoljan niz znakova koji najčešće uključuje identifikator niza koji je u retcima nakon prvog te eventualno dodatne podatke. Ovaj redak može biti proizvoljne duljine. U drugom retku (ili u više redaka počevši od drugog) nalazi se niz znakova koji predstavlja nukleotide ili aminokiseline.

[1] Pristupni broj (engl. *accession number* je jedinstveni broj svake sekvence koji omogućava lakše praćenje većeg broja varijanti iste sekvence u pojedinim bazama.)



Slika 3.2: Primjer FASTQ formata

Ovdje također nema ograničenja na pojavljivanje znakova. Nakon tog niza slijedi redak koji započinje znakom "+". On označava početak kodova ocjene kvalitete. Ovaj redak je uobičajeno sadržavao praktički isto što i prvi redak, ali je to naknadno odbačeno kako bi se datoteka smanjila [9]. Na kraju, u četvrtom retku nalaze se kodovi pridijeljeni pojedinim ocjenama kvalitete. Primjer zapisa u FASTQ formatu prikazan je na slici 3.2.

Originalni Sanger Fastq, kvalitetu pojedinog očitanja kodira ASCII znakovima između 33 i 126. Solexa, pak, to čini sa znakovima između 59 i 126, a Illumina uzima znakove iz intervala od 64 do 126. Ovo je prvi razlog nekompatibilnosti. Drugi razlog nekompatibilnost jest način na koji se računaju same kvalitete kod različitih formata. Primjerice, u originalni Sanger Fastq format, zapisuje se kvaliteta sekvence izračunata po Phred metodi. Phred metoda nastala je u Projektu ljudskog genoma (engl. *Human Genome Project*) kao način ocjenjivanja kvalitete DNA sekvenci. S druge strane, Solexa koristi način ocjenjivanja razvijen u istoimenoj kompaniji. U ovom radu, ocjenu kvalitete nisam uzeo u obzir te stoga ona nije presudna.

4. Metode

U ovome poglavlju opisat ću metode i računalne programe korištene pri ostvarivanju ciljeva ovoga rada.

4.1. SwSharp

Program sw# (izgovara se swsharp) je program za poravnavanje proteinskih i nukleinskih slijedova. Koristio sam sw# kako bih simulirao ponašanje programa koji rade detekciju preklapanja između očitanja. Naime, ta detekcija i nije ništa drugo nego traženje najboljih poravnanja između pojedinih krajeva očitanja. Najpoznatiji program iz domene poravnanja je BLAST (engl. *Basic Local Alignment Search Tool*).

Kako je problem poravnanja slijedova također velike složenosti, BLAST koristi različite heurističke optimizacije da bi taj proces ubrzao bez prevelikog utjecaja na krajnju preciznost i točnost poravnanja. Za razliku od BLAST-a, sw# radi poravnanja isključivo egzaktnim metodama. sw# iskorištava prednosti novih tehnologija paralelnog izvođenja programa, prije svega tehnologije CUDA (engl. *Compute Unified Device Architecture*), kako bi amortizirao složenost egzaktnih metoda i rezultate davao što brže. Treba napomenuti da CUDA koristi moć grafičkih kartica te time za nekoliko redova veličine smanjuje vrijeme izvođenja programa koji ju koriste na pravi način. Dodatno, SwSharp je prilagođen radu na više kartica odjednom te čak i više računala od kojih svako ima više kartica paralelno. Ne treba posebno isticati kako je njegova egzaktnost ukomponirana s dobrim performansama upravo ključna prednost nad BLAST-om ili sličnim programima te sam ga zbog toga i odabrao za simulaciju detekcije preklapanja kako bi rezultati doprinijeli tome da cjelokupno simuliranje bude što bliže stvarnosti.

4.2. PBSim

Kao što sam spomenuo u prethodnom poglavlju, prije nego se genom uopće može sastavljati, potrebno ga je sekvencirati. Sekvenciranje provode različiti uređaji raznim metodama. Neke od tih metoda kao produkt daju nešto kraća, a neke pak duža očitanja. Postoji nekoliko poznatih instrumenata za sekvenciranje genoma, od kojih su najpopularniji Illumina, GS FLX+ i PacBio RS. S obzirom da su ti instrumenti vrlo složeni i nisu dostupni u širokom obliku, da bi bilo moguće provoditi istraživanja na sastavljanju genoma, razvijeni su simulatori ovih uređaja. Računalni su to programi koji statističkim i slučajnim postupcima na osnovu empirijskih podataka dobijenih iz instrumenata generiraju očitanja neke ulazne sekvence.

Postoji velik broj simulatora i specijalizirani su, uglavnom, za pojedini realni instrument. Tako postoje Metasim kao simulator očitanja Illumine te PBSim za PacBio RS. U ovom radu koristim PBSim. PBSim je slobodno dostupan simulator očitanja instrumenta PacBio RS, a razvili su ga Yukiteru Ono, Kiyoshi Asai i Michiaki Hamada [10]. PBSim pomoću podataka o duljinama očitanja i lognormalne razdiobe simulira očitanja PacBio RS-a. PacBio RS proizvodi dvije vrste očitanja, kratka i vrlo točna (engl. *CCS - circular consensus reads*) te duga i dosta neprecizna (engl. *CLR - continuous long reads*). Tako je i PBSimu moguće zadati koju vrstu očitanja želimo. Dodatne opcije koje PBSim omogućava jesu zadavanje srednje vrijednosti duljine očitanja, srednje vrijednosti točnosti te standardnih devijacija točnosti i duljine. Moguće mu je zadati i željeni omjer vrsta pogrešaka, umetanja, izbacivanja ili nadomještanja, no to sam redovito ostavljao na podrazumijevanim vrijednostima od 60, 30 i 10, tim redom. On zatim pomoću eksponencijalne i lognormalne razdiobe generira očitanja u formatu FASTQ. Kako je prije napomenuto, kvalitetu očitanja nisam uzeo u obzir u ovom radu, ali ističem da PBSim koristi isključivo Sanger Fastq format i pripadajuće kodiranje kvalitete očitanja.

4.3. Simulacija

Alati za dobijanje očitanja iz genoma griješe, u manjem ili većem postotku, što ovisi uglavnom o duljinama očitanja. Postoji nekoliko vrsta pogrešaka. Može se dogoditi umetanje (engl. *insertion*), to jest da alat koji očitava ubaci nepostojeći nukleotid na neku poziciju u očitanju. Nasuprot toga, može se dogoditi izostavljanje postojećeg nukleotida ili delicija (engl. *deletion*), a dodatno postoji mogućnost krivog očitavanja nukleotida na nekoj poziciji što se naziva mutacija (engl. *mutation*). Simuli-

ranje procesa očitavanja DNA i detekcije preklapanja izveo sam na sljedeći način. Za dobijanje očitanja koristio sam PBSim. PBSim producira očitanja u FASTQ formatu. Radio sam mjerenja na očitanjima s medianom duljine 1000, 4000 i 8000. Za generiranje očitanja koristio sam genom bakterije deinococcus radiodurans – poznate bakterije velike radiootpornosti. Za svaki spomenuti median median, parametri pogreške koje sam zadao iznosili su 1%, 5%, 10%, 15% i 20%. Sveukupno, to predstavlja 15 mogućih testiranih kombinacija. U obzir sam uzeo i reverzne komplemente. Za svaku sam kombinaciju očitanja i pogreške generirao reverzni komplement te tako stvorio bazu očitanja i komplemenata. Nakon generiranja baze očitanja i reverznih komplemenata, krenuo sam na detekciju očitanja pomoću sw#-a. Kao upite na bazu, postavio sam prvotno generiranja očitanja, bez reverznih komplemenata. sw# je nakon toga za svako očitanje izgenerirao nekoliko desetaka najboljih poravnanja koja je detektirao kao alat za detekciju preklapanja.

Daljnju analizu podataka temeljio sam na dva važna pojma iz raspoznavanja uzoraka. To su odziv (engl. *recall*) i preciznost (engl. *precision*). Ove dvije veličine pokazatelj su koliko je detekcija uspješna. Pojednostavljeno rečeno, visoki odziv znači da je uspješno detektirana većina ispravnih pozitiva (engl. *true positives*) dok visoka preciznost kaže da je detektirano puno više ispravnih od neispravnih pozitiva (engl. *false positive*). Ispravni pozitiv je svako preklapanje detektirano sw#-om koje je točno. Lažni pozitiv, s druge strane, svako je preklapanje koje je sw# pogrešno detektirao, to jest ono poravnanje koje je dao na izlaz, a koje ne bi trebalo biti prisutno. Ocjenu pravih i lažnih pozitiva učinio sam jednostavno. PBSim uz svako očitanje daje i podatak o indeksu u referentnom genomu na kojem se originalno nalazi to očitanje. Tako sam običnom Python skriptom utvrdio točan broj preklapanja odmah nakon generiranja očitanja. U najjednostavnijoj varijanti, preklapanje je svaki odnos dva očitanja za koji vrijede sljedeće jednadžbe:

$$S_2 >= S_1 \tag{4.1}$$

te

$$S_2 <= E_1 \tag{4.2}$$

U jednadžbama su s S_1 i S_2 označeni indeksi početka prvog i drugog očitanja, a s E_1 kraj prvog očitanja. Jednadžba 4.1 kaže da početak drugog očitanja mora biti nakon početka prvog očitanja ili početi na istom mjestu kao i prvo. Jednadžba 4.2 ograničava početak drugog očitanja na interval početka i kraja prvog. Preciznost je tad definirana

kao:

$$P = \frac{N_p}{N} \tag{4.3}$$

gdje je s N označen ukupan broj detekcija (svih poravnanja na izlazu sw#-a), a s N_p broj ispravnih pozitiva koje je sw# otkrio. Odziv je definiran s:

$$R = \frac{N_p}{N_t} \tag{4.4}$$

gdje je s N_t označen ukupni broj pravih pozitiva. Za svaku kombinaciju očitanja i pogreške procijenio sam ove dvije veličine. Također, sw# može primiti različite kombinacije parametara koji određuju na koji način traži poravnanja. Ovi parametri bitno utječu na preciznost i odziv. Kao jedini kriteriji broja mogućih poravnanja, koristio sam e-vrijednost u tri različite varijante kako bih ilustrirao utjecaj broja poravnanja na preciznost i odziv. Veličina e-vrijednost inače predstavlja očekivanje broja poravnanja s istom ocjenom poravnanja kao i trenutni par upit–rezultat. Što je to očekivanje niže, rezultat je bolje. Kriterij može biti slabiji što onda znači bitno manju preciznost uz veći odziv jer su se otkrila i preklapanja koja inače ne bi bila otkrivena, ali su se lažno detektirala i brojna druga preklapanja. Naravno, druga krajnost jest dati vrlo stroga ograničenja, što onda znači vrlo veliku preciznost, ali i vrlo mali odziv. Koristio sam, primarno, iznos e-vrijednosti u iznosu 2.5 koji je osiguravao preciznost od otprilike 75%. Zatim sam testiranje proveo na iznosu e-vrijednosti 5.0. Ovaj postupak ponavljao sam za svaku kombinaciju očitanja i pogrešaka. Na kraju, radi bolje mogućnosti usporedbe, iskoristio sam blastn, koji je heuristička metoda traženja poravnanja između nukleotidnih sekvenci, kako bih dobio bolje usporedbe s mojim rezultatima na istom skupu podataka.

5. Rezultati

U ovom poglavlju, predstavljam rezultate simulacija provedenih u okviru ovoga rada. Rezultate prikazujem u obliku tablica i grafova. Svaka tablica predstavlja rezultate za određeni parametar e-vrijednosti korišten prilikom traženja preklapanja.

Tablica 5.1: Odziv i preciznost za e-vrijednost 2.5

Medijan duljine	Postotak pogreške	Odziv	Preciznost
1000	1%	0.405	0.803
1000	5%	0.404	0.803
1000	10%	0.411	0.806
1000	15%	0.410	0.793
1000	20%	0.400	0.801
4000	1%	0.388	0.777
4000	5%	0.384	0.780
4000	10%	0.376	0.734
4000	15%	0.367	0.708
4000	20%	0.370	0.706
8000	1%	0.388	0.791
8000	5%	0.407	0.815
8000	10%	0.411	0.803
8000	15%	0.379	0.750
8000	20%	0.340	0.648

U tablici 5.1 prikazani su rezultati dobijeni kad sam kao parametar koristio e-vrijednost u iznosu 2.5. Za svaki medijan duljine i postotak pogreške prikazani su izmjereni odziv i preciznost.

U tablici 5.2 prikazani su rezultati za e-vrijednost 5.0. Rezultati za e-vrijednost 10.0 praktički su identični onima prikazanima u tablici 5.2 te ih stoga nisam posebno naveo. Nadalje, u tablici 5.3 navodim rezultate dobijene programom blastn.

Tablica 5.2: Odziv i preciznost za e-vrijednost 5.0

Medijan duljine	Postotak pogreške	Odziv	Preciznost
1000	1%	0.729	0.764
1000	5%	0.723	0.765
1000	10%	0.732	0.762
1000	15%	0.728	0.753
1000	20%	0.711	0.735
4000	1%	0.644	0.671
4000	5%	0.633	0.671
4000	10%	0.635	0.652
4000	15%	0.640	0.652
4000	20%	0.645	0.651
8000	1%	0.657	0.689
8000	5%	0.669	0.691
8000	10%	0.651	0.659
8000	15%	0.601	0.622
8000	20%	0.602	0.603

Tablica 5.3: Odziv i preciznost za blastn

Medijan duljine	Postotak pogreške	Odziv	Preciznost
1000	1%	0.972	0.136
1000	5%	0.0707	0.0106
1000	10%	0.066	0.012
1000	15%	0.044	0.012
1000	20%	0.0266	0.013
4000	1%	0.098	0.007
4000	5%	0.090	0.007
4000	10%	0.096	0.009
4000	15%	0.833	0.009
4000	20%	0.037	0.008
8000	1%	0.112	0.005
8000	5%	0.167	0.008
8000	10%	0.161	0.009
8000	15%	0.111	0.008
8000	20%	0.062	0.007

6. Diskusija

Rezultati predstavljeni u prethodnom poglavlju jesu vrlo zanimljivi i prilično neočekivani. Moja početna pretpostavka bila je da će postotak pogreške biti najvažniji, presudni čimbenik utjecaja na preciznost i odziv. Očekivao sam da će se uz veći postotak pogrešaka, obje veličine značajno smanjivati. Pretpostavljao sam da će razlike u rezultatima između, primjerice, očitanja s prosječnim udjelom pogrešaka od 5% i 20% biti velike i da će pogreške u značajnijoj mjeri utjecati na alat za detekciju preklapanja. Ipak, kao što tablice prikazuju, rezultati se poprilično razlikuju od očekivanja. Prije svega, vidljivo je kako su prosječna duljina očitanja i e-vrijednost najvažniji parametri koji mogu i odziv i preciznost povećati ili smanjiti za dosta velik faktor.

Za utjecaj e-vrijednosti, objašnjenje koje mi se čini logično jest sljedeće. E-vrijednost je parametar koji alatu za detekciju kaže koliko može ići u širinu prilikom traženja najboljih preklapanja. U prijevodu, svaka dva para poravnanja imaju neku e-vrijednost koja se računa na osnovu statističkih parametara. Ona predstavlja vjerojatnost da se potpuno slučajno dođe do iste vrijednosti poravnanja u trenutnoj bazi. Poželjno je da e-vrijednost bude što je moguće manja. Ako se alatu za detekciju da manja e-vrijednost, dobit će se manje rezultata zato što će takav stroži kriteriji odrezati većinu lošijih poravnanja. To će jasno, utjecati i na broj točno detektiranih preklapanja, što će automatski smanjiti odziv. Preciznost će pri tome, ili ostati slična ili se povećati, zato što će se ujedno, ali s puno većim faktorom, smanjiti i broj lažnih pozitiva. Povećanje e-vrijednosti dovodi do suprotnog efekta. Povećanjem e-vrijednosti, dobija se više poravnanja, kako dobrih tako i lošijih. Više dobrih poravnanja znači veći odziv, ali ujedno za velik faktor smanjuje preciznost.

Što se tiče duljine, iz tablica je vidljivo da ona nema toliko velik utjecaj na odziv i preciznost, ali je on svakako značajniji od utjecaja pogrešaka. Vrlo je zanimljiva ovisnost preciznosti i odziva o duljini koja se vidi iz rezultata. Za duljinu 1000, odziv i preciznost u prosjeku su nešto veći nego za duljine 4000 i 8000. Uspoređujući samo duljine 4000 i 8000, nema značajne prosječne razlike u ovim veličinama. To bi moglo značiti asimptotsko približavanje određenoj prosječnoj vrijednosti odziva i preciznosti

uz poneku anomaliju – anomaliju kao što je preciznost za parametar duljine 8000 i postotak pogreške od 20%. Ipak, za ovaj zaključak smatram potrebnim mjerenja na većem rasponu medijana duljine. Nadalje, poželjno je da preciznost bude što veća, ali da odziv pri tome ne padne previše nisko. Ono što rezultati ovog rada prikazuju je trenutačna nemogućnost nalaženja prihvatljive sredine, to jest balansa između ove dvije veličine za pojedine parametre.

Iako bi se očekivao veći utjecaj raspona pogrešaka na točnu detekciju, to je očekivanje ispalo pogrešno u mojim mjerenjima. Pripisujem to jako izraženom utjecaju druge dvije veličine, to jest e-vrijednosti i duljine, što je, pretpostavljam, bitno amortiziralo utjecaj pogrešaka. Zanimljivo je da postotak pogrešaka najčešće nije u korelaciji s odzivom. Na primjer, kod očitanja duljine 1000 vidljivo je da pogreška od 10% ima veći odziv od pogreške od 1% što je jako zanimljivo i prilično neočekivano.

Rezultati dobiveni programom blastn najbliži su, barem po korelaciji pogrešaka i odziva, onima koje sam očekivao. U tablici se vidi da, generalno, što je postotak pogreške veći, to je odziv manji. Preciznost blastn programa je generalno niska što objašnjavam heurističkim pristupom i pretpostavljenim parametrima koji generiraju velik broj rezultata. Postoji nekoliko anomalija u tim rezultatima, ali generalno, oni su takvi kako bi se i očekivalo – veći postotak pogreške, manji odziv.

Sumirajući dobijeno, mišljenja sam da je potrebno svakako provesti dodatna istraživanja i mjerenja. Rezultati koje sam dobio izašli su iz očekivanih okvira i u slaboj su korelaciji s početnom pretpostavkom o tome da je utjecaj pogreške najvažniji u detekciji preklapanja. Ispalo je da su veličine koje su trebale biti manje zastupljenje ustvari one koje su značajnijeg utjecaja. To može voditi na dva zaključka. Jedan je da su mjerenja provedena s alatom koji moguće nije dovoljno prikladan za ovakav zadatak. S obzirom da je sw#-u moguće zadati i više parametara osim same e-vrijednosti za koje sam smatrao da ne mogu utjecati na krajnje rezultate, ipak smatram da bi valjalo pogledati kako promjena tih parametara utječe na odziv i preciznost. Tu prije svega mislim na zadavanje fiksnog broja poravnanja za svaki upit (na primjer 5, 10, ili 20) neovisno o e-vrijednosti te mijenjanja parametara unutarnjeg ocjenjivača sličnosti (engl. *scorer*). Drugi je da zaista pogreške nemaju toliki utjecaj na fazu overlapa (već na eventualno kasnije faze) što je teško prihvatiti jer je jasno da detekcija poravnanja mora ovisiti o sličnosti pojedinih očitanja i što rezultati dobijeni BLAST-om direktno opovrgavaju.

7. Zaključak

U ovome radu predstavio sam ideju iza sastavljanja genoma te opisao dvije najvažnije metode iz tog područja, komparativni i *de novo* pristup. Iako jednostavniji, komparativni pristup nije dovoljan za sve moguće potrebe te je zbog toga razvijen *de novo*. *De novo* je klasificiran kao NP-težak problem. Postoje različiti načini izvođenja *de novo* sastavljanja od kojih sam predstavio dvije glavne grupe – one zasnovane na pohlepnom pristupu i nasuprot njima one temeljene na grafovima. Metode temeljene na grafovima daju općenito bolje rezultate i više se koriste, ali pod cijenu puno veće složenosti i računalnih zahtjeva. Opisao sam FASTQ i FASTA formate kao najpopularnije formate datoteka za pohranu genetske informacije u računalnom obliku. Dao sam pregled alata koje sam koristio, konkretno programa za simulaciju očitavanja PBSim i programa za poravnavanje sw#-a. Istaknuto je da je sw# egzaktni alat za razliku od popularnih, ali heuristikama vođenih alata poput BLAST-a.

Nadalje, opisao sam korištene metode i načine simulacije koji su se temeljili na postupku generiranja očitanja preko PBSima uz različite parametre i analiziranja očitanja i traženja preklapanja pomoću sw#-a. Ipak, rezultati koje sam dobio iznenadili su i bitno se razlikuju od početne hipoteze. Pretpostavka da će pogreške na očitanjima biti presudni faktor kod detekcije preklapanja pokazale su se netočnima, pa je tako došlo do toga da je parametar e-vrijednosti ustvari najviše utjecao na dvije veličine kojima sam mjerio rezultat, odziv i preciznost. Zaključak koji se može izvući iz toga je postojanje potencijalnih drugih parametara koji su utjecali na mjerenja, a koje nisam uzeo u obzir i nisam pretpostavio da mogu utjecati ili jednostavna neprikladnost alata koje sam koristio za ovakav postupak simulacije. Druga mogućnost je da jednostavno pogreška zaista ne utječe na očitanja, što je skoro pa sigurno nemoguće. U budućnosti će biti potrebno provesti dodatna mjerenja i simulacije da se utvrde razlozi ovakvog ponašanja rezultata i da se generalno proces detekcije preklapanja pokuša dovesti na najoptimalniju moguću razinu s ciljem ubrzavanja cjelokupnog sastavljanja genoma živih bića.

LITERATURA

[1] M. Pop, "Genome assembly reborn: recent computational challenges", *Briefings in Bioinformatics*, 2009.

[2] M. Pop, A. Phillippy, A. L. Delcher, i S. L. Salzberg, "Comparative genome assembly", *Briefings in Bioinformatics*, 2004.

[3] J. Tarhio i E. Ukkonen, "A greedy approximation algorithm for constructing shortest common superstrings", *Theoretical Computer Science*, 1988.

[4] J.D. Kececioglu i E.W. Myers, "Combinatorial algorithms for dna sequence assembly", *Algorithmica*, 1993.

[5] N. Nagarajan i M. Pop, "Parametric complexity of sequence assembly: theory and applications to next generation sequencing", *Journal of Computational Biology*, 2009.

[6] E. Lander i M.S. Waterman, "Genomic mapping by fingerprinting random clones: a mathematical analysis", *Genomics*, 1988.

[7] G. Narzisi i B. Mishra, "Comparing de novo genome assembly: The long and short of it", *PLoS ONE*, 2011.

[8] R.C. Deonier, S.Tavare, i M.S.Waterman, *Computational Genome Analysis*, Springer, 2005.

[9] P.J. Cock, C.J. Fields, N. Goto, M.L. Heuer, i P.M. Rice, "The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants", *Nucleic Acids Research*, 2010.

[10] Y. Ono, K. Asai, i M. Hamada, "Pbsim: Pacbio reads simulator - toward accurate genome assembly", *Bioinformatics*, 2012.

Evaluacija metode za ispravljanje pogrešaka kod dugačkih očitanja

Sažetak

Sastavljanje genoma jedan je od temeljnih problema bioinformatike. Njegova važnost za razumijevanje brojnih pojava u živom svijetu je neporeciva. Proces sastavljanja genoma je vrlo težak i računalno zahtjevan proces koji uključuje više koraka. Postoji nekoliko metoda izvođenja rekonstrukcije genoma od kojih svaka nudi neke prednosti, ali i nailazi na probleme i izazove. *De novo* pristup izvodi se na više načina od kojih je najznačajni Preklapanje-Izravnavanje-Konsenzus. Prva faza ili preklapanje za cilj ima pronaći najbolja preklapanja između očitanja genoma i izgraditi graf povezanosti tih očitanja. U grafu, čvorovi predstavljaju očitanja, a bridovi preklapanja. Na taj način gradi se usmjeren netežinski graf. Faza preklapanja osjetljiva je na pogreške u očitanjima, a cilj je ovoga rada ispitati kako različiti postotci pogrešaka utječu na sposobnost detekcije točnih preklapanja.

Ključne riječi: *De novo*, sastavljanje genoma, preklapanje, sw#

Evaluation of a method for long read error correction

Abstract

Genome assembly is one of fundamental problems in bioinformatics. Its importance for understanding numerous phenomena in living organisms is undeniable. Process of genome assembly is a very hard and computationaly challenging process including several steps. There exist numerous ways of implementing genome assembly all of which offer some advantages, but also face some problems and challenges. *De novo* assembly can be performed in many ways. Overlap-Layout-Consensus method is the most popular of them all. In its first phase, the overlap, it tries to find all the best alignments between generated reads and build a graph from them. Vertices of such a graph represent the reads, while edges represent overlaps. This produces a directed unweighted graph. Overlap phase is sensitive to errors in the reads and goal of this paper is to test how various percentages of read errors influence its ability to detect correct overlaps.

Keywords: *De novo*, genome assembly, overlap, sw#