SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Poboljšanje djelomično sastavljenog genoma dugim očitanjima

Domagoj Pluščec, Dunja Vesinger

Zagreb, prosinac 2018.

SADRŽAJ

1. Uvod

DNK sekvenciranje je proces određivanja redoslijeda nukleotida u DNK. Sekvenciranje cijelog genoma je ono sekvenciranje u kojem se nastoji odrediti redoslijed nukleotida čitavog genoma nekog organizma. Nažalost, današnje metode za očitavanje DNK ne omogućavaju da se odjednom očita cijeli genom. Umjesto toga, uređaju očitavaju samo kratka očitanja koja je potom potrebno sastaviti. Pored toga, nije poznat redoslijed dobivenih očitanja što dodatno otežava njihovo sastavljanje.

Sastavljanje očitanja vrši se na temelju međusobnih preklapanja između očitanja. Najveća prepreka ovakvom načinu sastavljanja je prisutnost dugačkih identičnih ili gotovo identičnih slijedova nukleotida u genomu koje nazivamo ponavljajuće regije. Druga prepreka sastavljanju genoma su pogreške koje mogu nastati pri očitavanju slijeda nukleotida. One mogu uzrokovati lažna preklapanja u očitanjima.

Sastavljanje očitanja još je uvijek otvoren problem na kojem rade mnogi znanstvenici. U ovom radu predstavljamo implementaciju algoritma koji omogućava efikasno sastavljanje genoma u kojima su prisutne ponavljajuće regije. Algiritam je predstavljen u radu (Du i Liang, 2018.). Algoritam kostiti sirova očitanja i kontinge nastale korištenjem postojećih alata za sastavljanje genoma te nastoji povezati nepovezane kontinge korištenjem sirovih očitanja. Autori u svom radu navode kako je ova metoda drastično poboljšala rezultate sastavljanja u odnosu na dosad korištene metode.

2. Opis algoritma

Algoritam *Highly efficient repeat assembly* (skraćeno HERA) opisan je u radu (Du i Liang, 2018.). Njegov je cilj poboljšati kvalitetu sastavljenih očitanja. U tu svrhu algoritam koristi sirova očitanja te kontinge nastale korištenjem postojećih alata za sastavljanje očitanja. Algoritam nastoji povezati dobivene kotinge koristeći sirova očitanja. Za svaki par kontinga pronalaze se svi mogući putevi sastavljeni od sirovih očitanja koji povezuju ta dva kontinga. U konačnici se između svih dobivenih puteva između kontinga odabiru oni koji su najizgledniji i njima se kontinzi međusobno povezuju. Detaljan pregled algoritma dan je u nastavku poglavlja.

2.1. Ulazni podaci

Algoritam kao ulazne podatke dobiva sirova očitanja i već sastavljene kontinge U FASTA formatu. Uz to, kao ulaz je potrebno proslijediti i preklapanja između kontinga i sirovih očitanja te međusobna preklapanja između očitanja u PAF formatu dobivena primjenom Minimap algiritma. Minimap algoritam i rezultirajući PAF format podataka opisani su u radu (Li, 2017.).

Algoritam Minimap vratit će nam sva moguća preklapanja između očitanja, no HERA koristi samo ona preklapanja kod koji se kraj jedne sekvence preklapa s početkom druge sekvence. (Pri tome *početak* i *kraj* sekvence ne znače da se stogo počinje od prvog nukleotida u sekvenci i završava sa zadnjim nukleotidom, već označavaju približan položaj u očitanju.) Ovo se postiže filtriranjem sekvenci po mjeri koja se naziva identitet sekvence (engl. *sequence identity, SI*) i koja je definirana u PAF formatu zapisa.

Nakon učitavanja i filtriranja podataka, najprije se konstuira graf preklapanja.

2.2. Konstrukcija grafa preklapanja

Graf preklapanja konstruira se na temelju preklapanja između sekvenci koja su dobivena Minimap algoritmom. U grafu preklapanja svaka sekvenca nukleotida (konting ili sirovo očitanje) predstavlja jedan čvor grafa. Svako preklapanje između dvije sekvence, tj. dva čvora,

predstavlja jedan brid u grafu. Kako smo prilikom učitavanja filtrirali preklapanja, preostala su samo ona u kojima se kraj jedne sekvence preklapa s početkom druge sekvence. Ovakva preklapanja jednoznačno definiraju poredak sekvenci pa je jasno da naš graf preklapanja u tom slučaju treba biti usmjereni graf.

Graf konstruiramo na sljedeći način. Najprije za svaku sekvencu nukleotida definiranu u ulaznim podacima stvaramo čvor. Potom prolazimo kroz sva preklapanja i za svako preklapanje dodajemo brid između čvorova koji predstavljaju početnu i završnu sekvencu preklapanja. Nakon što smo generirali graf svih preklapanja, nastojim pronaći puteve u tom grafu koji mogu spojiti dva kontinga.

2.3. Generiranje potencijalnih puteva

Prilikom generiranja potencijalnih puteva nastoje se pronaći putevi kojima bi se mogla povezati neka dva kontinga. Svaki put sastoji se od niza čvorova u grafu preklapanja koji su međusobno povezani bridovima. Svi bridovi na nekom putu moraju biti istog usmjerenja, a svi čvorovi unutar puta moraju biti jedinstveni, tj. mogu biti sadržani u putu najviše jednom.

Potencijalni putevi generiraju se zasebno za svaki konting. Za konting najprije uzmemo sve izlazne bridove prema drugim čvorovima. Potom uzmemo susjedne čvorove koji su na kraju tih bridova i za svaki susjedni čvor nastojimo generirati put od njega do nekog drugog kontinga. Taj put generiramo tako da za susjedni čvor odaberemo jedan brid koji vodi do nekog sljedećeg čvora. Potom iz tog čvora ponovo odabiremo jedan brid i tako sve dok ne dosegnemo jedan od uvjeta zaustavljanja. Bitno je naglasiti da se prilikom generiranja puta iz nekog čvora ne smijemo više puta obići isti čvor, stoga moramo pažljivo odabirati sljedeći brid na putu imajući to na umu. Postoje tri različite vrste selekcije na temelju kojih odabiremo sljedeći brid na putu. Za svaki izlazni brid koji je povezan s početnim čvorom koristimo sve tri metode selekcije te za svaki brid dobivamo maksimalno tri različita puta koja vode do nekog durog kontinga.

Selekcije kojima se odabire sljedeći brid za nastavak puta međusobno se razlikuju po kriteriju odabira brida. Prva selekcija odabire brid na temelju mjere koja se naziva *overlap score, OS*. Metoda uzračuna ove mjere objašnjena je u nastavku. Neka su zadane dvije sekvence, S_1 i S_2, koje imaju preklapanje. Preklapanje je definirano sljedećim veličinama: identitetom sekvence SI, duljinama *overhang*-ova OH_1 i OH_2, duljinama preklapanja OL_1 i OL_2 te duljinama *extension*-a EL_1 i EL_2. Sve navedene veličine dobivaju se kao rezultat Minimap algoritma i zapisane su u PAF zapisu preklapanja. *Overlap score* preklapanja između sekvenci S_1 i S_2 računa se na sljedeći način:

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

.

Selekcija odabire onaj brid za koji je *overlap score* najveći.

Druga selekcija odabire sljedeći brid na temelju veličine koja se naziva *extension score, ES*. Ako postoji brid koji je usmjeren od sekvence S_1 prema sekvenci S_2, onda se *extension score* takvog preklapanja računa na sljedeći način:

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

.

Selekcija odabire onaj brid za koji je *extension score* najveći.

//TODO Monte Carlo selekcija

Postoje tri načina zaustavljanja prilikom generiranja puteva. Prvi je da ne postoji niti jedan brid iz trenutnog čvora ili svi postojeći bridovi vode u čvorove koji su već obiđeni na trenutnom putu. U tom slučaju smo naišli na "slijepu ulicu" prilikom generiranja puta i odbacujemo putanju koju smo dosada generirali. Drugi način je da premašimo maksimalan dozvoljeni broj čvorova unutar puta. Generiranje potencijalnih puteva može biti vrlo zahtjevno za računalne resurse te stoga ograničavamo maksimalnu duljinu generiranih puteva. U slučaju da put premaši maksimalnu dozvoljenu duljinu, odbacuje se. Konačno, put je uspješno generiran kad odaberemo brid na čijem se kraju nalazi drugi konting. U tom slučaju spremamo generirani put.

Jednom kad smo generirali puteve, potrebno je odabrati onaj najvjerojatniji, tj. odrediti konsenzus sekvence.

2.4. Određivanje konsenzus sekvenci

U konačnici je dva kontinga moguće povezati samo pomoću jednog puta. Stoga između svih puteva koje smo generirali za svaki par kontinga treba odabrati jedan za koji je najvjerojatnije da predstavlja slijed nukelotida u originalnom genomu. Algoritam HERA u tu svrhu najprije generira grupu puteva za koje je najvjerojatnije da odgovaraju stvarnom slijedu nukleotida, a potom se iz navedene grupe bira jedan reprezentativni put. U nastavku je opisan način generiranja takvih grupa koje sadržavaju najvjerojatnije puteve, a naziaju se konsenzus sekvence.

Pretpostavimo da želimo odrediti konsenzus sekvencu puteva od kontinga predstavljenog čvorom N_1 do kontinga predstavljenog čvorom N_2. Najprije pronađemo sve puteve koji povezuju čvor N_1 s čvorom N_2 i grupiramo ih po duljini. Ako je razlika duljina između najkraćeg najduljeg puta manja od 10000, tada formiramo samo jednu grupu u kojoj se nalaze svi putevi. U suprotom formiramo grupe tako da najprije uzmemo najkraći put i s njim grupiramo sve puteve čija je duljina od zadanog puta veća za najviše 1000. Potom odabiremo

najkraći od preostalih puteva (onih koje još nismo grupirali) i formiramo novu grupu na analogan način. Postupak ponavljamo sve dok ne grupiramo sve puteve.

Nakon toga za svaku grupu odredimo frekvencije duljina puteva koji se nalaze u toj grupi. Primjerice, ako grupa sadrži 10 puteva duljine 1000, tada je frekvencija puteva duljine 1000 u tvoj grupi jednaka 10. Jednom kad smo odredili sve frekvencije duljina, filtriramo svaku grupu s obzirom na najveću frekvenciju unutar nje. Iz grupe izbacujemo sve puteve čija je duljina takva da je frekvencija pojavljivanja te duljine unutar grupe manja od 50% najveće frekvencije.

Na primjer, recimo da imamo grupu koja se sastoji od 40 puteva duljine 1000, 30 puteva duljine 1100 i 10 puteva duljine 1200. Tada maksimalna frekvencija te grupe iznosi 40. Puteve duljine 1100 zadržavamo u grupi je jer njihova frekvencija veća od 50% najveće frekvencije: $30 > 0.5 \cdot 40$. Sve puteve duljine 1200 izbacujemo iz grupe jer im je frekvencija premala u odnosu na maksimalnu frekvenciju.

Ako imamo samo jednu grupu puteva od čvora N_1 do čvora N_2, onda tu grupu proglašavamo konsenzus sekvencom. U slučaju da imamo dvije grupe, odabiremo onu koja ima više puteva i nju proglašavamo konsenzus sekvencom. Kada imamo više od dvije grupe, provodimo iterativni algoritam za odabir konsenzus sekvence. Najprije odabiremo grupu čija je maksimalna frekvencija veća od maksimalnih frekvencija svih ostalih grupa. Zatim promatramo prvu sljedeću grupu čiji su putevi dulji od odabrane grupe. Ako je njezina maksimalna frekvencija veća od 50% maksimalne frekvencije odabrane grupe, ona postaje nova odabrana grupa. U suprotnom odabrana grupa postaje konsenzus sekvenca.

Kad smo odredili konsenzus sekvencu od čvnora N_1 do čvora N_2, potrebno je također odrediti konsenzus sekvencu od čvora N_2 do čvora N_1. Ovaj postupak ponavljamo za svaki par kontinga u grafu. Ako između neka dva kontinga ne postoji niti jedan put, onda ne postoji niti konsenzus sekvenca između njih.

2.5. Konstrukcija grafa povezivanja

Nakon što smo odredili konsenzus sekvence između svih kontinga, možemo konstruirati graf povezivanja. Čvorovi grafa povezivanja su svi kontinzi, a bridovi su predstavljeni konsenzus sekvencama. Važno je napomenuti kako su bridovi ovog grafa također usmjereni te između dva čvora, N_1 i N_2, mogu postojati dva brida: brid od N_1 do N_2 te brid od N_2 do N_1. U ovom grafu jedan konting može imati bridove prema proizvoljnom broju drugih kontinga. No, u konačnom rješenju algoritma svaki konting može imati najviše jedan ulazni i najviše jedan izlazni brid. Uz to, svaki se čvor u konačnom rješenju može pojaviti najviše jedanput. Stoga je potrebno eliminirati sve cikluse iz grafa.

2.6. Generiranje izlaza

Kako bismo odredili konačno rješenje, moramo odabrati bridove koje je potrebno eliminirati iz grafa poveivanja. U tu svrhu definiramo konflikt indeks, CI, za svaki čvor u grafu povezivanja. Za neki čvor N_1, konflikt index se računa na sljedeći način. Svaka konsenzus sekvenca, tj. svaki brid u grafu povezivanja, ima definiran broj puteva (engl. *path number*, NP) koji su sadržani u njoj. Neka konsenzus sekvenca od čvora N_1 do čvora N_m ima najveći broj puteva od svih konsenzus sekvenci koje izlaze iz čvora N_1, a konsenzus sekvenca od N_1 do N_n drugi najveći broj puteva. Označimo ove brojeve puteva s NP_{1m} i NP_{1n}. Konflikt indeks za čvor N_1 tada iznosi:

$$CI_1 = \frac{NP_{1n}}{NP_{1m}}$$

.

Ako je konflikt indeks nekog čvora velik, onda nije jasno definirano s kojim ga čvorom treba spojiti te se čvor ne spaja. Unaprijed definiramo konstantu C_{max} koja predstavlja najveći dozvoljeni konflikt indeks za koji će se izvršiti spajanje.

Kako bismo spojili čvorove, najprije ih poredamo po njihovim konfliktnim indeksima, od čvora s najmanjim konfliktnim indeksom do čvora s najvećim konfliktnim indeksom. Krećemo od čvora s najmanjim konflikt indeksom N_1 i odabiremo konsenzus sekvencu čvora N_1 koja ima najveći broj puteva te ga pomoću nje spajamo na čvor N_i na koji pokazuje konsenzus sekvenca. Pošto je početak čvora N_i sada spojen na čvor N_1, mičemo čvor N_i iz liste čvorova na koju se moguće spojiti i ponovo računamo kolflikt indekse za preostale čvorove u grafu tako da najprije izbrišemo sve postojeće konsenzus sekvence koje vode prema čvoru N_i, a potom za preostale konsenzus sekvence provodimo izračun. Ovaj postupak se ponavlja sve dok konflit indeksi svih preostalih čvorova ne budu veći od C_{max}.

3. Rezultati

4. Zaključak

Zaključak.

LITERATURA

H. Du i C. Liang. Assembly of chromosome-scale contigs by efficiently resolving repetitive sequences with long reads, 2018.

H. Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences, 2017.