SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

BIOINFORMATIKA

**Generiranje konsenzusa koristeći ReAligner**

Ivan Bestvina, Karlo Fodor, Zrinka Gavran

Voditelj: izv. prof. dr. sc. Mile Šikić

Zagreb, Siječanj, 2016.

**Sadržaj**

1. Uvod 3

2. Algoritam 4

1.1 Glavni okvir algoritma 4

1.2 Genriranje konsenzusa 4

1.3 Ocjenjivanje konsenzusa 5

1.4 Poravnavanje fragmenta i konsenzusa 5

2. Ulazni podaci 6

2.1 FASTA 6

2.2 MHAP 7

3. Izlazni podaci 8

3.1 Konsenzus 8

3.2 Layout 8

4. Prikaz na primjeru 9

5. Rezultati testiranja 11

5.1 Kvaliteta rezultata 11

5.2 Vrijeme izvođenja 11

5.3 Količina zauzete memorije 11

6. Zaključak 12

7. Literatura 13

8. Sažetak 14

# Uvod

U posljednja dva desetljeća, troškovi sekvenciranja genoma spustili su se za oko pet redova veličine. Na primjer, dok je sekvenciranje ljudskog genoma 2002. godina koštalo oko 100 milijuna dolara, danas je ono iznosi svega nekoliko tisuća dolara. Ovakav pad cijene stvara sve veću potrebu za brzom i preciznom računalnom obradom podataka koji su sekvenciranjem dobiveni. Najčešće je to mnoštvo kratkih fragmenata početnog genoma koji se međusobno preklapaju, te sadrže određenu količinu grešaka.

Centralni zadatak obrade ovakvog skupa fragmenata je njihovo sastavljanje te izvlačenje “konsenzusa” - aproksimacije početnog genoma iz fragmenata. Dva su osnovna pristupa ovom problemu: de-novo pristup koji razmatra isključivo preklapanja među fragmentima, te mapirajući pristup koji u obzir uzima i neka prethodna očitanja genoma te na njih “mapira” dobivene fragmente.

OLC algoritmi (koji spadaju u tzv. algoritme nove generaicja) su de-novo algoritmi koji konsenzus generiraju kroz tri odvojena koraka: O - overlap, L - layout te C - consensus. U prvoj (overlap) fazi generira se graf preklapanja fragmenata, koji daje rudimentalnu informaciju o međusobnim pozicijama pojedinih fragmenata. Druga faza (layout) slaže ove fragmente kako bi se greška u preklapanjima minimizirala. Zadnja faza za svaku poziciju odabire najizgledniju dušičnu bazu te od njih izgrađuje jedinstveni niz - konsenzus.

ReAligner [1] algoritam kojeg smo implementirali unaprijeđuje zadnju fazu, generiranje konsenzusa. On dodatno smanjuje grešku poravnanja te iz takvog, preciznijeg alignmenta generira konsenzus.

U nastavku rada detaljnije je, uz jednostavan primjer, opisan sam algoritam te je dan prikaz rezultata implementacije.

# 2. Algoritam

Cilj algoritma je krenuti iz početnog poravnanja koje smo dobili iz podataka o preklapanju nizova te kroz više iteracija poboljšavati krajnje poravnanje. Prepostavljamo da je početno poravnanje nizova temeljeno na izlazu layout faze OLC postupka globalno točno, ali nije lokalno optimalno.ReAlign algoritam bazira se u svojoj osnovi na tzv. round-robin paradigmi: u svakom koraku ukupno višenizno pravnanje podijeli se na dva skupa koji se međusobno poravnaju i onda spajaju ovisno o rezultatima poravnanja.

Podjela na dva skupa u ovom kontekstu zapravo predstavlja izdvajanje jednog niza iz skupa nizova, te uspoređivanje tog niza s konsenzusom svih ostalih. Prilikom poravnavanja koristi se Needleman–Wunschov dinamički algoritam2, dok se za ocjenjivanje konsenzusa uzima prosijek dviju shema bodovanja. Za generiranje konsenzusa korištena je tzv. majority vote tehnika po promatranom stupcu za određenu poziciju. Na ovaj način postignuto je vrijeme izvođenja linearno s brojem nizova.

## Glavni okvir algoritma

Algoritam započinje svoj rad učitavajući inicijalno poravnanje skupa fragmenata. Neka je s A označen taj skup, te neka je s Ai označen i-ti fragment. U nastavku je dan pseudokod glavnog okvira algoritma. Cilj ovako napisanog algoritma je minimizirati ocjenu konsenzusa.

Neka je k je ukupni broj fragmenata

1. i := 0
2. C := konsenzus(A)
3. 𝛿 := ocjena\_konsenzusa(C, A)
4. Ponavljaj:
5. S := A / Ai
6. Poravnaj Ai prema konsenzusu od S
7. C := konsenzus(A)
8. 𝛿\_new := ocjena\_konsenzusa(C, A)
9. Ako je 𝛿\_new >= 𝛿: prekini petlju.
10. 𝛿 := 𝛿\_new
11. i := (i + 1) % k

## Genriranje konsenzusa

Konsenzus predstavlja jedinstveni niz baza dobivenih iz matrice preklapanja, te se u ovoj implementaciji izvodi majority vote tehnikom. Preciznije, za svaki stupac matrice preklapanja uzimaju se u obzir sve baze koje se u njemu nalaze, te se odabire ona koja se najčešće pojavljuje. Ako postoji više takvih baza (koje se pojavljuju jednak broj puta), uzimaju se zajedno, i ta pozicija u konsenzusu ostaje nedefinirana.

## Ocjenjivanje konsenzusa

Za ocjenjivanje konsenzusa koristimo dvije pomoćne ocjene. U njihovom izračunu koristimo pomoćnu funkciju za ocjenjivanje određenog stupca. U prolasku kroz stupac uspoređujemo sve baze koje se u njemu nalaze s onim bazama koje se u njemu nalaze najviše puta i za svaku bazu koja se nalazi u stupcu, a nije većinska baza povećavamo ocjenu za taj stupac za jedan. To trenutno rješenje dodajemo prvoj pomoćnoj ocjeni, a drugoj dodajemo to trenutno rješenje podijeljeno veličinom stupca.

Nakon što ovaj postupak ponovimo za sve stupce dobili smo ocjenu konsenzusa.

## Poravnavanje fragmenta i konsenzusa

Poravnanje ovih dvaju nizova izvodi se Needleman–Wunschovim dinamičkim algoritmom2. Algoritam izvodi samo lokalna poboljšanja preklapanja, što dovodi do modifikacije originalnog dinamičkog algoritma za poravnanje. Umjesto cjelokupne matrice udaljenosti, u našem je slučaju ona ograničena samo na dijagonalni pojas širine 2𝜺+1. Na ovaj je način osigurano da će svaki znak u nizu, nakon završetka algoritma, biti pomaknut za najviše 𝜺 pozicija u nekom smjeru. Također, ovo ograničenje uvelike ubrzava rad algoritma te smanjuje njegove memorijske zahtjeve.

# Ulazni podaci

## FASTA

FASTA je tekstualni format zapisa datoteka koji se koristi kod zapisa nukleotidnih i peptidnih sekvenci.

Datoteke pisane u FASTA formatu označujemo sa više nastavaka ovisno o sadržaju. Nastavak .fasta ili .fas je generički nastavak, nastavak .fna koristi se kod zapisa nukleinske kiseline, .ffn sadrži kodne regije za genom, .faa sadrži zapis aminokiseline te .frn je nastavak za ne kodirajući RNA.

Format zapisa u FASTA datotetci je podijeljen na dva dijela. Prvi dio označen je sa početnim znakom veće '>' te nakon toga znaka u istome redu nalazi se opis. Format opisa sekvence nije definiran FASTA formatom, ali postoje standardi kojima se može ograničiti oblik opisa. Nakon opisa označenog sa '>', u sljedećem redu nalazi se sekvenca, ona se može nalaziti samo u jednom redu ili više redova odjednom.

Format FASTQ je proširenje formata FASTA i proširen je sa dodatnih dva reda i umjesto znaka '>' koristi se znak '@'. Dodaje se red sa znakom '+' koji može po potrebi imati isti identifikator niza kao i kod znaka '@' i on naznačuje da se u redu ispod nalazi zapis kvalitete očitanja sekvence, taj zapis mora sadržavati jednak broj znakova kao i sekvenca. Svaki znak određuje kvalitetu iščitanja korespodentnog znaka sekvence.

FASTA i FASTQ formati su prošireni u Multi-FASTA i Multi-FASTQ formate, oni za razliku od originalnih formata mogu slijedno sadržavati više sekvenci, svaka počinje znakom '>', odnosno '@'.

## MHAP

MHAP ili MinHash Alignment Process format je format koji definira preklapanja između neke dvije sekvence. U jednoj datotetci, koja ima nastavak .mhap može biti više preklapanja, po jedna u svakom redu.

Svako preklapanje sadrži redom:

* indeks prve sekvence iz redoslijeda u ulaznoj datotetci s fragmentima
* indeks prve sekvence iz redoslijeda u ulaznoj datotetci s fragmentima
* Jaccard parametar koji određuje kvalitetu preklapanja
* broj zajedničkih k-meraorijentacija prvog fragmenta u preklapanju u odnosu na orijentaciju na ulazu, 0 ako su jednako orijentirani, i 1 ako nisu
* indeks početka preklapanja na prvom indexu u odnosu na početak niza s obzirom na orijentaciju
* indeks završetka preklapanja na prvom indexu u odnosu na početak niza s obzirom na orijentaciju
* orijentacija drugog fragmenta u preklapanju u odnosu na orijentaciju na ulazu, 0 ako su jednako orijentirani, i 1 ako nisu
* indeks početka preklapanja na drugom indexu u odnosu na početak niza s obzirom na orijentaciju
* indeks završetka preklapanja na drugom indexu u odnosu na početak niza s obzirom na orijentaciju

# Izlazni podaci

## Konsenzus

Za izlaz konsenzusa koristimo tri različita načina fasta ispisa:

1. kada napišemo n ako imamo više simbola na određenoj poziciji
2. kada napisemo [XY] ako imamo više simbola na određenoj poziciji gdje X i Z predstavljaju te simbole
3. kada odaberemo bilo koji simbol u slučaju kada ih ima više na određenoj poziciji

## Layout

Za ispis završnog layouta koristimo GFA ispis i to u formatu:

a Comment Fixed fields

a Golder path utgName utgStart readName:start-end readOri length

gdje je utgName redni broj fragmenta, utgStart označava gdje u završnom poravnanju počinje naš fragment, readName je ime s kojim je fragment došao u ulaznim podacima, start i end predstavljaju indekse početka i kraja fragmenta, readOri je orijentacija fragmenta (+ za normalnu orijentaciju, - ako je fragment okrenut) i lenght je veličina fragmenta.

# Prikaz na primjeru

Neka je dano preklapanje triju fragmenata i njihov konsenzus:

1. ATCGG

2. AATTC

3. TCCCGGGA

----------

Kons. AATTCCGGGA

U prvom koraku prve iteracije iz skupa fragmenata uklanjamo prvi fragment, konstruiramo konsenzus preostala dva, te ih poravnavamo. Konsenzus drugog i trećeg fragmenta je:

2. AAT T C

3. T C CCGGGA

----------

Kons. AAT[TC]CCGGGA

Kao što je već spomenuto, ako je nekoliko najučestalijih baza u stupcu, sve one se uzimaju u obzir i ulaze u konsenzus, što se označava uglatim zagradama.

Nakon generiranja konsenzusa slijedi poravnavanje prvog fragmentaprema njemu. Sam konsenzus se ne mijenja, jer bi njegova promjena označavala promjenu svih ostalih nizova. Koristeći modificirani Needleman–Wunschov algoritam2, generira se matrica udaljenosti koja na jednoj svojoj osi sadrži dio konsenzusa koji odgovara području promatranog fragmenta, s dodatkom 𝜺 znakova s obje strane, a na drugoj promatrani fragment. U matricu se, na neko polje (i,j), po retcima izračunavaju i upisuju cijene poravnanja nizova do tog polja, tj. fragmenta od 0 do i te konsenzusa od 0 do j. Neka je 𝜺=1. Tada je promatrani dio konsenzusa:

Frag.     A T  CGG

Kons.    AT[TC]CCGG

Matrica udaljenosti u prvom redu i prvom stupcu sadrži znak za početak niza. U prvom redu tablica se puni inicijalnim vrijednostima koje ovise o parametru 𝜺 i određuju pojas u kojem je bazama dopušteno biti u rezultantnom nizu. Ostale vrijednosti određuju se na sljedeći način:

1. Neka je s c označena cijena poravnanja pojedinih baza na poziciji (i,j). Ako skup simbola konsenzusa na poziciji j sadrži simbol fragmenta na poziciji i, c je 0. Inače je –1
2. Ocjena dijagonalnog pomaka je vrijednost na polju (i-1,j-1) + c. Ovo predstavlja poravnanje znaka i iz fragmenta sa znakom j iz konsenzusa.
3. Ocjena vertikalnog pomaka je vrijednost na polju (i,j-1) - 1. Ovo označava poravnanje znaka j iz konsenzusa s novonastalom prazninom (“crticom”) u fragmentu.
4. U polje se upisuje veća od dviju ocjena te identifikator je li se radilo o dijagonalnom ili vertikalnom pomaku.

Na zadanom primjeru ona izgleda ovako (s X su označene pozicije koje zbog 𝜺 pojasa nisu dozvoljene, crvenom bojom su označeni vertikalni, a s plavom dijagonalni pomaci):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ~ | A | T | [TC] | C | C | G | G |
| ~ | -1 | **0** | -1 | X | X | X | X | X |
| A | X | -1 | **-1** | -2 | X | X | X | X |
| T | X | X | -1 | **-1** | **-2** | X | X | X |
| C | X | X | X | -1 | -1 | **-2** | X | X |
| G | X | X | X | X | -2 | -2 | **-2** | X |
| G | X | X | X | X | X | -2 | -2 | **-2** |

Odabiremo najmanju ocjenu u zadnjem redu, krenuvši od kraja te se vraćamo kroz matricu putem kojim smo do tog polja došli (u primjeru označeno **bold**). Za svaki dijagonalni pomak zapisujemo znak, a za svaki vertikalni crticu. Dobiveno poravnanje je:

Frag.     A T  -CGG

Kons.    AT[TC]CCGG

Kao što vidimo, poravnanje nije optimalno, što je posljedica uskog 𝜺-pojasa.

# Rezultati testiranja

Korištene oznake:

* G – duljina genoma
* N – broj fragmenata
* L – prosječna duljina fragmenta
* D – dubina (NxL/G)
* I – broj iteracija
* Sin – zbroj grešaka u inicijalnom konsenzusu
* Sre – zbroj grešaka nakon realigna
* T – ukupno vrijeme svih iteracija u sekundama
* M – ukupno zauzeće memorije u MB
* 𝜺 – epsilon vrijednost slaganja nizova

Svi testovi su izvršeni na računalu s 2.20GHz procesorom u Linux Mint OS-u. Za mjerenje memorijskih zahtjeva korišten je alat cgmemtime3. Za ocjenjivanje točnosti dobivenih konsenzusa korišteni su alati Gepard4 i Dnadiff5.

Zadnji prikazni test odrađen je na genomu bakterije Escherichia coli.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G | N | L | D | 𝜺 | I | Sin | Sre | T | M |
| 500 | 60 | 50 | 6 | 5% | 5 | 3k | 2.8k | 0.10 | 4.8 |
| 1k | 60 | 100 | 6 | 5% | 5 | 2k | 1.6k | 0.07 | 4.8 |
| 2k | 100 | 200 | 10 | 5% | 5 | 4.8k | 2.9k | 0.27 | 4.8 |
| 5k | 200 | 300 | 12 | 5% | 5 | 14.5k | 8.6k | 1.07 | 5.4 |
| 10k | 400 | 200 | 12 | 5% | 5 | 29.6k | 17.9k | 3.54 | 7.3 |
| 20k | 800 | 300 | 12 | 5% | 5 | 59.5k | 34.2k | 17.64 | 10.8 |
| 50k | 2k | 300 | 12 | 5% | 1 | 167.9k | 131k | 21.17 | 15.8 |
| 100k | 3k | 300 | 9 | 5% | 1 | 273.2k | 237.9k | 49.31 | 15.4 |
| 200k | 4k | 400 | 8 | 5% | 1 | 940k | 872k | 264 | 34 |
| 4.6m | 50k | 500 | 5.4 | 5% | 1 | 10.4m | 9.4m | 1672 | 567 |

# Zaključak

Tehnologija sekvenciranja gena se razvija sve brže i brže te je količina podataka koju je potrebno obraditi sve veća i veća. Kako bi svi ti podaci mogli klasificirati, uspoređivati, analizirati i koristiti u različite istraživačke svrhe, potrebno ih je efikasno obraditi. Kod podataka koji nastaju sekvenciranjem genoma efikasna obrada označava slaganje fragmenata u konsenzus koji se onda promatra kao (najvjerojatniji) niz koji odgovara početnom genomu.

Strategija u kojoj konsenzus izvodimo u dva odvojena koraka, prvo izvodeci grubo poravnavanje, koje zatim lokalno sređujemo, pokazala se prilično dobrom. Round-robin strategija za lokalno poravnavanje također je, kao što rezultati pokazuju, efikasna i mogla bi se u praksi koristit.

Neke od naprednih mogućnosti koje bi se u algoritam mogle ugraditi svakako bi uključivale paralelizaciju procesa poravnavanja, zamjenu Needleman–Wunschovog algoritma nekim od naprednijih algoritama za poravnavanje, te ugrađivanje dodatnih funkcija za ocjenjivanje konsenzusa.

Također, mjenjanje funkcije cijene za *indel* kod poravnavanja doprinjela bi generiranju realnijih konsenzusa.

# Literatura

[1] ReAligner: A Program for Refining DNA Sequence Multi-Alignments, E. L. Anson, E. W. Myers

[2] Needleman, Saul B.; and Wunsch, Christian D. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". Journal of Molecular Biology 48 (3): 443–53.

[3] <https://github.com/isovic/cgmemtime>

[4] Bioinformatics. 2007 Apr 15;23(8):1026-8. Epub 2007 Feb 19. Gepard: a rapid and sensitive tool for creating dotplots on genome scale. Krumsiek J1, Arnold R, Rattei T.

[5] Versatile and open software for comparing large genomes. S. Kurtz, A. Phillippy, A.L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S.L. Salzberg. Genome Biology (2004), 5:R12.