SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

BIOINFORMATIKA

Pronalazak mutacija pomoću treće generacije sekvenciranja

Dominik Stanojević, Kristijan Vulinović
Voditelj: *Robert Vaser, mag. ing.*

Zagreb, siječanj 2019.

SADRŽAJ

1. Uvod

Cilj projekta je određivanje mutacija koje su se dogodile na referentnom genomu. U cijelosti je poznat referentni genom, te su također poznate i sekvence očitanja mutiranog genoma. Očitanja je potrebno mapirati na referentni genom, poravnati ih i odrediti mutirani genom. Usporedbom mutiranog genoma s referentnim određuju se mutacije. Moguće mutacije su:

- supstitucija (jedna nukleotidna baza referentnog genoma zamijenjena je drugom)
- umetanje (u referentni genom je umetnuta jedna nukleotidna baza)
- brisanje (iz referentnog genoma je izbrisana jedna nukleotidna baza)

2. Implementacija

2.1. Mapiranje sekvence

Kako je referentni genom dugačak, potrebno je odrediti područje unutar istoga na kojemu će se raditi poravnanje sa očitanim sekvencama. Za tu potrebu koriste se minimizeri, koji su opisani u radu [3]. Minimizer je podniz genoma određene duljine k (k-mer). U općem slučaju postoji velik broj k-mera (jednako duljini genoma), zbog čega se odabiru samo određeni od njih. Odabir se vrši na način da se promatra w uzastopnih k-mera te se od njih sprema samo onaj najmanji (minimizer). U radu [2] se vrijednost k-mera dodatno hashira te se oni međusobno uspoređuju prema veličini njihove hash vrijednosti. Pogledajmo sljedeći primjer:

C G T A G T C G A T G A C G T

Uz k-mere veličine 5 i gledanje 4 uzastopna k-mera, i poredak $A < C < G < T$, prvi minimizer se dobije kao što je prikazano podebljano u nastavku:

C	G	T	A	G	T	C	G	A	T	G	A	C	G	T
C	G	T	A	G										
	G	T	A	G	T									
		T	A	G	T	C								
			A	**G**	**T**	**C**	**G**							

Prema postupku korištenom u [2] se nakon određivanja minimizera radi indeksiranje istih, gdje se minimizeri spremaju u hash tablicu, gjde je hash vrijednost minimizera ključ. Na kraju je potrebno sekvencu upita mapirati na određeni dio referentnog genoma, što se radi na način da se za referentni genom pronađu svi minimizeri, te se oni uspoređuju sa svim minimizerima iz sekvenciranog genoma. Gledaju se oni minimizeri sekveniranog genoma cija hash vrijednost postoji u minimizerima referentnog genoma. Ti se minimizeri sortiraju prema poziciji u genomu te se potom traži najdulji rastući podniz u njima. U tom se podnizu nalaze minimizeri koji se preklapaju u sekvenciranom djelu genoma i referentnom te oni određuju poziciju u referentnom genomu na kojoj se treba vršiti poravnanje. Točna pozicija određena je minimalnom i maksimalnom pozicijom minimizera u najduljem rastućem podnizu.

2.2. Poravnanje

Nakon što je sekvencirani uzorak mapiran na određeno područje referentnog genoma, potrebno ih je poravnati. Za to je korišten Needleman-Wunsch algoritam, opisan u knjizi [1]. Algoritam radi na način da definira vrijednost poravnanja 2 genoma na sljedeći način:

$$V(i,j) = \begin{cases} 0, & i = 0 \wedge j = 0 \\ d * i, & j = 0 \\ d * j, & i = 0 \\ max \begin{cases} V(i-1, j-1) + w(s_i, t_j) \\ V(i-1, j) + d \\ V(i, j-1) + d \end{cases} & \text{inače} \end{cases}$$

Algoritam također definira i 3 parametra koja je potrebno odabrati, a to su cijene slaganja, neslaganja i praznina. Nakon što se na ovaj način popuni matrica sa podatcima o sličnosti, pronalazi se najveća vrijednost u zadnjem retku i stupcu. Od te vrijednosti se rekonstruira put do pozicije $(0, 0)$, čime se određuje poravnanje. Tako se primjerice poravnanjem genoma `CGTAGTCGATGACGT` sa očitanjima `CGTAT`, `TATTCG`, `TCGATG`, `CGATGAC`, `GATGACGT`, `GACGT` dobiju sljedeća poravnanja (uz prethodno mapiranje):

C	G	T	A	G	T	C	G	A	T	G	A	C	G	T
C	G	T	A	T										
		T	A	T	T	C	G							
					T	C	G	A	T	G				
						C	G	A	T	G	A	C		
							G	A	T	G	A	C	G	T
										G	A	C	G	T

Većinskim glasanjem se za svaku poziciju određuje vrijednost te će se za dane sekvence dobiti sljedeći genom `CGTATTCGATGACGT`, koji se od referentnog razlikuje u 5. bazi. Vidi se kako je baza iz `G` prešla u `T`, zbog čega se zaključuje da je ova mutacija bila supstitucija. Prilikom glasanja se također uzima u obzir i broj sekvenci koje su očitane za određenu nuukleotidnu bazu referentnog genoma. Ako je taj broj manji od određenog praga koji se postavlja kao dodatni parametar implementacije, mutacije se ne računaju zbog male pouzdanosti.

3. Rezultati

Prilikom testiranja opisanog postupka korišteni su k-meri veličine 15, a minimizeri su određeni iz prozra veličine $w = 10$. Kod poravnanja korištene su vrijednosti 5 za slaganje, -4 za neslaganje te -8 za praznine. Posljenji parametar je minimalna pokrivenost nakon slaganja sekvenciranih djelova te je on postavljen na 10.

Opisani postupak je implementiran te je mjeren jaccard index. Testiranjem na skupu lambda, postupak je ostvario jaccard index 0.78, te je za to iskorišteno 2.5GB radne memorije, a izračun je trajao manje od 1 minute. Referentna implementacija na istom primjeru ostvaruje jaccard index 0.45.

Testiranjem na skupu ecoli, postupak je ostvario jaccard index 0.74, te je za to iskorišteno 12GB radne memorije, a izračun je trajao 40 minuta. Referentna implementacija na istom primjeru ostvaruje jaccard index 0.82.

4. Literatura

[1] Mile Šikić i Mirjana Domazet-Lošo. Bioinformatika skripta. 2013. URL https://www.fer.unizg.hr/_download/repository/bioinformatika_skripta_v1.2.pdf.

[2] Heng Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14):2103–2110, 2016. doi: 10.1093/bioinformatics/btw152. URL http://dx.doi.org/10.1093/bioinformatics/btw152.

[3] Michael Roberts, Wayne Hayes, Brian R. Hunt, Stephen M. Mount, i James A. Yorke. Reducing storage requirements for biological sequence comparison. *Bioinformatics*, 20 (18):3363–3369, 2004. doi: 10.1093/bioinformatics/bth408. URL http://dx.doi.org/10.1093/bioinformatics/bth408.