MATEMATIČKI FAKULTET
UNIVERZITET U BEOGRADU

Simulacija postupka sekvenciranja genoma

Master rad

Ana Mijalković

Beograd 2012 godine

Univerzitet u Beogradu - Matematički fakultet
Master rad

Autor:	Ana Mijalković 1029/2009

Naslov:	Simulacija postupka sekvenciranja genoma
Mentor:	dr Saša Malkov
Članovi komisije:	dr Nenad Mitić
	dr Miloš Beljanski
Datum:	27.09.2012.

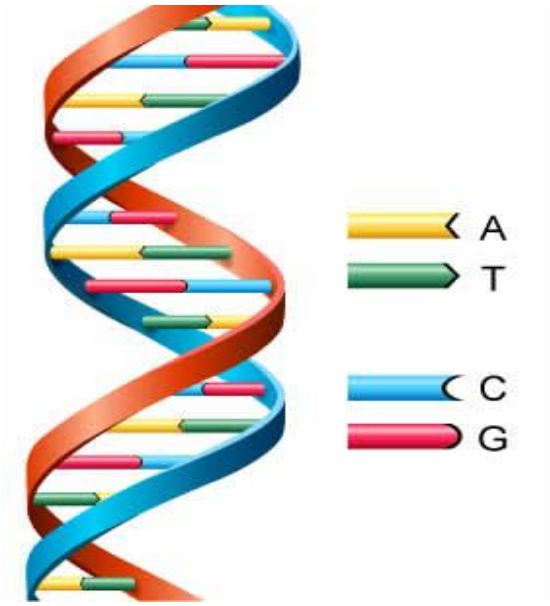
Sadržaj

Spisak slika

1 Uvod

Genom predstavlja celokupni genetski materijal jedinke ili vrste [11]. Zahvaljujući tome što sadrži ceo set naslednih osobina i instrukcija za kreiranje i održavanje organizama, omogućava prenošenje života na naredne generacije. Svaki organizam na Zemlji ima jedinstveni genom koji se sastoji od gena pakovanih u hromozome. Geni obezbeđuju specifične karakteristike organizama. Sastavljeni su od dezoksiribonukleinske kiseline, odnosno DNK. Značaj gena u genetici bi se mogao izjednačiti sa značajem atoma u fizici. Kao što je atom osnovna jedinica čestica, tako je gen osnovna jedinica nasleđa.

DNK je molekul koji predstavlja osnovni genetski materijal svake žive ćelije. Sastavljen je od dva uparena lanca nukleotida koji su spiralno uvijeni jedan oko drugog (**Slika 1**). Svaki nukleotid sadrži tri dela, šećer - dezoksiribozu, fosfatnu grupu i azotnu bazu. U sastavu DNK mogu se naći četiri azotne baze: adenin (A), guanin (G), citozin (C) i timin (T). Lanac DNK je jednoznačno određen redosledom baza.



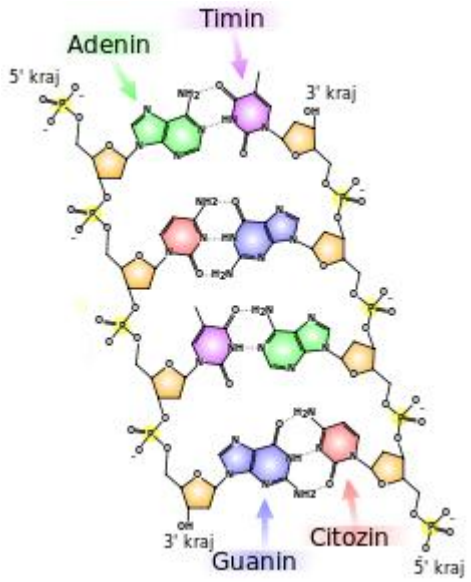
Slika 1. DNK - dva uparena lanca nukleotida[1]

Struktura DNK lanca je takva da su šećeri međusobno povezani fosfatnim grupama. Gradi se fosfodiestarska veza između trećeg ugljenikovog atoma jednog šećera i petog ugljenikovog

[1] Slika preuzeta sa: http://education.technyou.edu.au/view/91/155/what-does-dna-look

atoma sledećeg šećera (**Slika 2**). Pošto su fosfodiestarske veze asimetrične, svaki DNK lanac ima dva kraja. Ti krajevi se označavaju sa 5' i 3'. Na 5' kraju DNK lanca nalazi se fosfatna grupa, dok se 3' kraj završava šećerom. Zahvaljujući asimetričnosti DNK lanca, moguće je odrediti i njegov smer. Svaki DNK lanac se čita od 5' kraja ka 3' kraju.

Dva lanca DNK molekula se međusobno povezuju preko azotnih baza. Svaka baza na jednom lancu povezuje se sa tačno jednom bazom drugog lanca. Purinska baza (adenin i guanin), povezuje se sa pirimidinskom bazom (timin i citozin). Povezivanje se vrši preko vodoničnih veza. Adenin se povezuje sa timinom preko dve vodonične veze, dok se citozin povezuje sa guaninom uz pomoć tri vodonične veze. Vodonične veze se lako raskidaju i ponovo uspostavljaju. Ovakvo povezivanje DNK lanaca naziva se komplementarno povezivanje. Na osnovu baza na jednom DNK lancu mogu se jednoznačno ustanoviti i baze drugog lanca u okviru molekula.



Slika 2. Struktura DNK molekula[2]

DNK je određena kada je određena sekvenca baza koje čine jedan njen lanac. Zbog toga je određivanje sekvence baza DNK lanca jedan od najvažnijih koraka u izučavanju genoma nekog organizma. Postupak određivanja sekvence DNK se naziva sekvenciranje genoma. Azotna baza je nosilac genetske informacije i predstavlja jedinu razliku u okviru nukleotida.

[2] Slika preuzeta sa: http://en.wikipedia.org/wiki/DNA

Na osnovu toga, DNK sekvencu možemo posmatrati kao niz slova A, C, G, T, koja odgovaraju nizu baza u lancu. Jedan DNK lanac preveden na našu azbuku izgleda kao na slici (**Slika 3**):

Slika 3. Primer DNK sekvence[3]

Ukoliko se DNK posmatra na taj način, postupak sekvenciranja genoma svodi se na određivanje redosleda slova A, G, C, T u zapisu sekvence DNK.

Sekvenciranje genoma je početni korak u njegovom razumevanju. Razumevanje genoma se često poredi sa dekodiranjem [2]. Sekvenca genoma dobijena sekvenciranjem je samo jedna duga niska slova. Možemo je uporediti sa rečenicom. Da bismo razumeli značenje pročitane rečenice, pored razumevanja pojedinačnih slova i reči koje sačinjavaju, potrebno je razumeti i značenje koje te reči imaju kada se spoje u celinu. Slična situacija je i kod genoma kod koga takođe postoji unutrašnje značenje koje treba otkriti.

Genom možemo da zamislimo kao knjigu, napisanu bez razmaka, znakova interpunkcije, ali i sa dodatim besmislenim niskama slova između rečenica, kao i unutar njih. Na taj način možemo da zaključimo koliko je, zapravo, teško pročitati ga i razumeti njegovo značenje. Sekvenciranje genoma nam ne otkriva sve njegove tajne. Nakon sekvenciranja i dalje ostaje posao razumevanja značenja dobijenih niski slova. To podrazumeva razumevanje raznovrsnih gena u okviru genoma, načina na koji su oni povezani, kao i kako su povezani različiti delovi genoma. I pored toga, sekvenciranje genoma je veoma važan deo posla, kako bi se uopšte dobila sekvenca koju treba "razumeti".

Na osnovu dobijene sekvence bi trebalo jednostavnije i brže pronaći gene u okviru genoma. Dobijena sekvenca bi takođe, trebalo da omogući i proučavanje drugih delova genoma. Među njima se nalaze regulatorni delovi koji kontrolišu korišćenje određenih gena, i delovi koji predstavljaju nekodirajuće sekvence.

[3] Slika preuzeta sa:
http://www.genomenewsnetwork.org/resources/whats_a_genome/Chp1_1_1.shtml#genome1

1.1 Sekvenciranje genoma

Postupak sekvenciranja genoma obuhvata sekvenciranje fragmenata genoma, rekonstrukciju genoma od dobijenih fragmenata (*assembly*), kao i anotaciju genoma. Rekonstrukcija genoma podrazumeva povezivanje sekvenciranih fragmenata, dok anotacija predstavlja "razumevanje dobijene sekvence" - prepoznavanje delova genoma koji bi mogli da budu geni, prepoznavanje start i stop kodona, nekodirajućih sekvenci i mnogih drugih korisnih informacija o genomu.

Samo sekvenciranje genoma se može obavljati na dva načina. Jedan je **postupno sekvenciranje**, poznato kao klon-po-klon pristup (*BAC-to-BAC Sequencing, Map-Based Sequencing* ili *Hierarchical Shotgun Sequencing*), dok drugi pristup predstavlja **nasumično sekvenciranje** (*Whole Genome Shotgun Sequencing, WGS*). Razlike između ova dva načina sekvenciranja su u kvalitetu dobijenih rezultata, ali i u ceni, kao i vremenu potrebnom da se obavi sekvenciranje. Dok klon-po-klon sekvenciranje daje preciznije i pouzdanije rezultate, nasumično sekvenciranje je sklonije greškama. Sa druge strane, cena i vreme trajanja postupka su značajne prednosti nasumičnog sekvenciranja.

1.1.1 Klon-po-klon strategija

U slučaju klon-po-klon strategije, pre samog sekvenciranja DNK, pravi se gruba mapa celog genoma. Konstrukcija mape se sastoji u deljenju samog hromozoma na velike delove i ustanovljivanju redosleda svih tih delova. Gruba mapa se kreira i za svaki od delova koji se nakon toga deli u još manje sekvence tako da se ostavljaju preklapanja između njih. Preklapajući delovi služe pri ponovnoj rekonstrukciji genoma nakon sekvenciranja. Sekvenciranje i analiziranje manjih delova je sledeći korak.

Klon-po-klon strategija pojednostavljuje proces rekonstrukcije genoma zahvaljujući prethodno kreiranim mapama. Pored toga, prednost ove strategije su i njena preciznost i manja mogućnost greške. Nasuprot tome, kreiranje mape može biti jako skupo i vremenski zahtevno. Ova strategija je zbog toga pogodnija za veće i kompleksnije genome.

1.1.2 Nasumično sekvenciranje

Nasumično sekvenciranje razvijeno od strane Freda Sangera 1982 godine, nasuprot klon-po-klon sekvenciranju, predstavlja princip sekvenciranja odozdo na gore [4]. Prvi korak

postupka je deljenje DNK na delove. Nakon toga se delovi sekvenciraju po slučajnom redosledu. Dobijeni fragmenti se na kraju spajaju u početni genom traženjem preklapanja primenom računarske analize.

Prednost nasumičnog sekvenciranja je u tome što ne zahteva pravljenje mape. To je ujedno i mana na neki način. Zbog ogromnog broja dobijenih delova za koje je potrebno naći preklapanje i spojiti ih u genom, rekonstrukcija genoma je neuporedivo teža i sklonija greškama bez prethodno konstruisane mape. To se pogotovo odnosi na velike genome, pa je ova strategija pogodnija za sekvenciranje manjih genoma sa malo repetitivnih sekvenci, kao što su prokariotski genomi. Prednosti ove strategije su i brzina i cena postupka.

Nasumično sekvenciranje se danas koristi za većinu bakterijskih genoma, kao i dopuna za mnoge druge genomske projekte [4].

1.1.3 Postupak sekvenciranja genoma

Tokom sedamdesetih godina razvijene su dve metode za sekvenciranje genoma [1]. U pitanju su hemijska i enzimska metoda, od kojih se enzimska više koristi u današnjim projektima.

- **Hemijsko sekvenciranje** (*A. Maxim* i *W. Gilbert*)
  Metoda hemijskog sekvenciranja zasnovana je na hemijskoj modifikaciji DNK i uklanjanju baza, nakon čega se lanac prekida i analiziraju se dobijeni delovi. Iako je objavljena dve godine nakon Sangerove metode enzimskog sekvenciranja, ova metoda je vrlo brzo postigla veliki uspeh [5]. Metod zahteva radioaktivno obeležavanje jednog kraja DNK. Nakon toga se odvojeno odvijaju četiri različita skupa reakcija. Svaki od skupova je karakterističan za jedan tip baze i služi kako bi se ta baza uklonila i na tom mestu prekinuo lanac. Reakcije su podešene tako da se u proseku ukloni samo jedna baza po fragmentu. Kao rezultat izvedenih reakcija nastaje niz delova od obeleženog kraja do prvog prekida. Dobijeni fragmenti se razlikuju u dužini za samo jedan nukleotid i dalje se analiziraju postupkom elektroforeze, koja ih razdvaja po dužini. Nakon toga se rendgenskim snimanjem dobija slika čijom analizom se utvrđuje redosled baza u okviru molekula.

- **Enzimsko sekvenciranje** (*chain termination method* ili didezoksi metod, *F. Sanger* i *A. Coulson*)
  Metoda enzimskog sekvenciranja je, kao efikasnija metoda, češći izbor pri sekvenciranju DNK. Ključni princip ove metode je korišćenje didezoksi nukleotida (ddNTPs) koji prekidaju polimerizaciju DNK ugrađujući se u molekul. Za početak procesa potreban je jednolančani uzorak DNK i obeleženi prajmer. Nad uzorkom DNK sprovode se četiri paralelne reakcije polimerizacije. U svaku od reakcija dodaju se sva četiri standardna dezoksi nukleotida (dATP, dGTP, dCTP and dTTP) i DNK polimeraza. Dodavanjem jednog od didezoksi nukleotida svakoj reakciji izaziva se njegova ugradnja u molekul i prekidanje polimerizacije. Kao rezultat postupka, u svakoj reakcionoj smeši, dobijaju se delovi različitih dužina koji se završavaju istom bazom. Razdvajajući svaku smešu posebno, na gelu za sekvenciranje dobijaju se trake koje se nakon toga izlažu rendgenskom snimanju i utvrđivanju redosleda baza.

1.1.4 Rekonstrukcija genoma

Rekonstrukcija genoma (*sequence assembly*) podrazumeva sastavljanje genoma od fragmenata dobijenih sekvenciranjem. Cilj je rekonstruiati polaznu DNK nisku. Softveri koji se bave time uglavnom dele istu ideju, traže preklapanja među dobijenim fragmentima [6]. Zadatak se sastoji u pronalaženju preklapajućih sekvenci, slično postupku slaganja slagalice. Prvi korak je poređenje fragmenata i pronalaženje onih sa odgovarajućom sličnošću. Kao najveći problem u ovom koraku javljaju se neadekvatna preklapanja. Ona mogu da nastanu usled mutacija prilikom replikacije sekvenci. Takođe, mogu da budu i uzrok velikog broja ponavljajućih sekvenci, koje su posledica sličnosti unutar genoma. Nakon pronalaženja preklapajućih fragmenata, oni se spajaju formirajući duže nizove susednih fragmenata (*contigs*). Ovaj proves se suočava sa mnogim problemima. Izdvajaju se delovi kod kojih je moguće nastaviti sekvencu sa više različitih fragmenata. Nasuprot tome, moguće je i da ne postoji odgovarajuća sekvenca za nastavak. Na kraju je potrebno povezati i utvrditi redosled dobijenih delova. Zbog velike količine podataka, problem rekonstrukcije genoma je izuzetno zahtevan bioinformatički problem.

2 Problem simulacije sekvenciranja genoma

Testiranje softvera za rekonstrukciju genoma na osnovu fragmenata dobijenih nasumičnim sekvenciranjem zahteva veći broj test primera, na kojima se mogu proveravati različiti aspekti problema rekonstrukcije. Kako je sekvenciranje fragmenata genoma relativno skup postupak, ukazala se potreba da se simuliranjem postupka sekvenciranja genoma obezbede test primeri po manjoj ceni.

Tema ovog master rada je izrada programa koji simulira postupak sekvenciranja genoma. Kako bi se sekvenciranje predstavilo na što bolji način, potrebno je da što bolje podržava realan postupak sekvenciranja. U narednim odeljcima je opisano kako se odvija savremen postupak nasumičnog sekvenciranja.

Sa informatičke tačke gledišta, genom živih organizama može se posmatrati kao niska znakova, dužine $5*10^5$ do 10^{10} u zavisnosti od veličine genoma u organizmu. Karakteri koji se mogu naći u toj nisci su A - adenin, G - guanin, C- citozin i T - timin. Na taj način posmatrana niska znakova DNK, omogućuje da se transformacije izvršene nad njom predstave programski kao transformacije nad stringovima.

2.1 Delovi simulacije

Simulacija obuhvata dve celine:

- Pravljenje DNK lanca
- Sekvenciranje

Samo sekvenciranje se sastoji od:

- Replikacija DNK lanaca
- Seckanje dobijenih lanaca
- Skeniranje dobijenih fragmenata

2.2 Pravljenje DNK lanca

Početni deo postupka je pravljenje DNK lanca. Pravljenje veštačkog DNK lanca ima dvojaki smisao. Najpre, u cilju simulacije ostalih elemenata procesa sekvenciranja, a zatim i radi omogućavanja testiranja rekonstrukcije genoma u nepozatim uslovima, a ne samo na već rekonstruisanim genomima. Kako bi veštačka DNK sekvenca bila što vernije napravljena, mogu se zadati neke pravilnosti u okviru njene strukture.

Biohemičar *Erwin Chargaff*, dokazao je 1949 godine da iako različiti organizmi imaju različite količine DNK, svaki od njih sadrži jednaku količinu adenina i timina [7]. Isto važi i za citozin i guanin. To pravilo je poznato kao Čargafovo prvo pravilo uparivanja *(Chargaff First Parity Rule)*. Na primer, ljudski genetski materijal sadrži oko 30 procenata adenina i isto toliko timina, dok se citozin i guanin mogu naći u po 20 procenata nukleotida.

Uočena je slična pravilnost vezana za broj azotnih baza u okviru jednog lanca DNK. Čargafovo drugo pravilo uparivanja (*Chargaff Second Parity Rule*) govori o tome da je količina adenina u okviru jednog lanca DNK približno jednaka količini timina. Isto važi i za citozin i guanin. Kao razlog tome navodi se visoka učestalost inverzija (zamena redosleda azotnih baza) [14]. Inverzija se odvija između lanaca DNK tako što se delovi sekvenci premeštaju sa jednog lanca na drugi.

Na osnovu prethodno opisanog odnosa azotnih baza u okviru DNK, jedan od važnih parametara pri pravljenju lanca DNK je i procenat svake od azotnih baza. Zbog toga bi deo programa koji pravi lanac DNK kao parametre trebalo da primi željenu dužinu DNK i procenat CG (citozina i guanina zajedno). Nakon toga je jednostavno izračunati koliki procenat svake od azotnih baza bi trebalo da se nalazi u lancu koji se pravi.

Ukoliko se procenat citozina i guanina označi kao CG%, onda je potrebno procente ostalih azotnih baza izračunati tako da važi:

- procenat citozina ≈ procenat guanina ≈ $CG\%/2$
- procenat adenina ≈ procenat timina ≈ $(1 - CG\%)/2$

Kao izlaz iz ovog dela programa, potrebno je da se dobije tekstualna datoteka sa jednim DNK lancem, zadate dužine i sa procentualno zadatom količinom azotnih baza.

2.3 Sekvenciranje

Sekvenciranje se sastoji iz replikacije, seckanja lanaca i skeniranja. Ulaz za ovaj deo simulacije predstavlja tekstualna datoteka sa sekvencom DNK lanca.

2.3.1 Replikacija

Replikacija je postupak kojim se DNK molekul duplira i od jednog nastaju dva "identična" DNK molekula. Taj postupak se u stvarnosti odvija tako što se DNK molekul ubacuje u veštački bakterijski hromozom (*bacterial artificial chromosome - BAC*[4]). Tako ubačeni DNK se klonira svaki put pri deobi bakterijske ćelije.

Pri prvom kopiranju ubačenog DNK molekula nakon deobe bakterije nastaju dva "identična" DNK molekula. Nakon toga se postupak ponavlja za stari ubačeni DNK molekul, ali i za novonastali molekul, tako da će nakon ovog koraka postojati četiri DNK molekula. Ponavljanjem ovog postupka n puta dobija se 2^n molekula DNK koji su "identični" početnom molekulu.

Naravno, kako je u pitanju biohemijski postupak, ne možemo sa sigurnošću da tvrdimo da će se nakon n ponavljanja dobiti 2^n DNK molekula. Može se dogoditi da se neki deo molekula uopšte ne iskopira. Takođe, klonirani DNK molekul često nije baš identičan kao i molekul koji se klonira. Pri kopiranju se mogu javiti pojedine genske mutacije [8]. Mutacije koje se javljaju su sledeće:

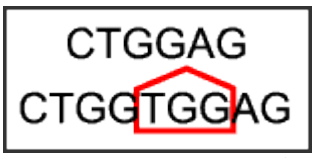
- **DELECIJA** - predstavlja mutaciju pri kojoj se deo DNK lanca izgubi (**Slika 4**).

Slika 4. Primer delecije[5]

- **INSERCIJA** - predstavlja mutaciju pri kojoj se neki suvišni bazni parovi umeću na nekom mestu u lancu (**Slika 5**)

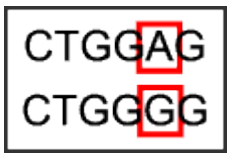
[4] BAC je veštački napravljen deo DNK koji može da se klonira unutar bakterijske ćelije.

[5] Slika preuzeta sa: http://evolution.berkeley.edu/evolibrary/article/mutations_03



Slika 5. Primer insercije[6]

- **SUBSTITUCIJA** - mutacija pri kojoj se jedna azotna baza zamenjuje drugom, na primer adenin - A prelazi u guanin - G (**Slika 6**)



Slika 6. Primer substitucije[6]

Genske mutacije u procesu replikacije nastaju iz različitih razloga [15]. Razlog zamene baze nekom drugom (substitucija) je obično greška prilikom uparivanja baza. Sa druge strane, insercija i delecija nastaju kao posledica malog zavijanja lanca (*strand slippage*). Ukoliko se zavije lanac koji nastaje, prilikom replikacije dolazi do ubacivanja novih nukleotida - insercija. Delecija, odnosno gubljenje nukleotida je posledica zavijanja lanca koji se klonira. Na taj način u novonastalom lancu se izgube nukleotidi koji bi trebalo da budu zakačeni za zavijene nukleotide.

Učestalost grešaka se bitno razlikuje kod različitih vrsta organizama. Razlike su uočljive čak i kod različitih delova genoma u okviru jednog organizma. Greške mogu da variraju od jedne greške u 100 miliona ili čak u milijardu nukleotida kod bakterija, pa sve do jedne greške u sto ili hiljadu nukleotida, kod čoveka.

Da bi se moglo preciznije upravljati ovom fazom simulacije, potrebno je da se kao parametar navede procenat grešaka pri jednom kloniranju DNK lanca.

2.3.2 Seckanje lanaca

U postupku sekvenciranja genoma lanci se dele na manje fragmente kako bi se mogao primeniti odgovarajući proces. Veličina fragmenata zavisi od procesa sekvenciranja. DNK lanac se seče na delove uz pomoć enzima koji raskidaju veze u okviru lanca. Restrikcioni

[6] Slika preuzeta sa: http://evolution.berkeley.edu/evolibrary/article/mutations_03

enzimi presecaju DNK nisku na mestima koja odgovaraju specifičnim kraćim sekvencama nukleotida, poznatim kao restrikciona mesta.

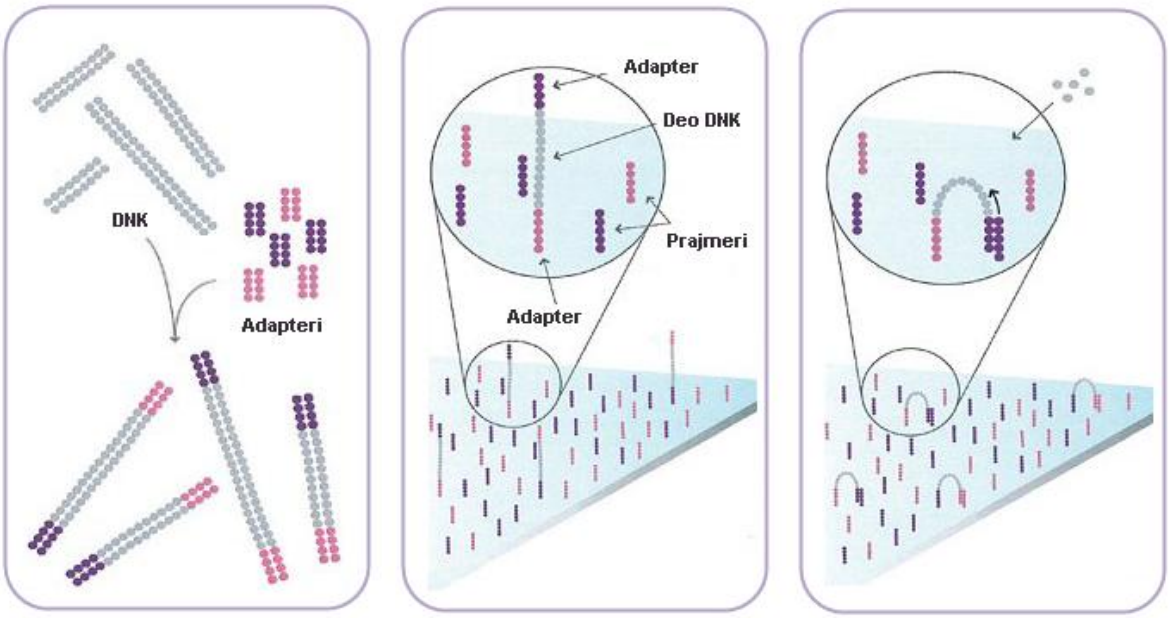
Zadatak simulacije je da na slučajan način podeli DNK lanac na način koji odgovara stvarnom postupku. Problem seckanja bi u programu trebalo da bude rešen tako da je moguće zadati određene kriterijume, odnosno nizove od po nekoliko azotnih baza koji predstavljaju uzorke restrikcionih mesta. Kako bi seckanje bilo što vernije predstavljeno potrebni su i neki dodatni parametri. Jedan od njih je minimalna dužina fragmenata nakon sečenja. Ona bi trebalo da obezbedi da isečene niske ne budu kraće od neke neophodne dužine. Drugi potreban podatak je najveća dopuštena dužina sekvence za koju se ne mora ići u dalje deljenje.

2.3.3 Skeniranje

Skeniranje dobijenih delova sa odvija tako što se dobijeni delovi DNK najpre pripreme dodavanjem adaptera[7] na oba kraja lanca (**Slika 7 - 1**). Nakon toga se adapteri svojim slobodnim krajem kače na površinu na kojoj se već nalaze prajmeri[8] (**Slika 7 - 2**). Povezivanjem slobodnog kraja lanca i zakačenog prajmera kreira se most (**Slika 7 - 3**). Nakon toga se na tako pripremljenu ploču sa mostovima dodaju neophodni enzimi i slobodni nukleotidi koji se, uz pomoć enzima, spajaju sa zakačenim lancem i kreiraju drugi lanac DNK (**Slika 7 - 3** i **Slika 8 - 1**). Nukleotidi se povezuju vodonočnim vezama, tako što se purinske baze u koje spadaju adenin i guanin povezuju sa pirimidinskim bazama - timinom i citozinom.

[7] Adapteri predstavljaju posebne, veštački dodate sekvence. Dodaju se na krajeve isečenih fragmenata kako bi se omogućilo lakše rukovanje fragmentima.

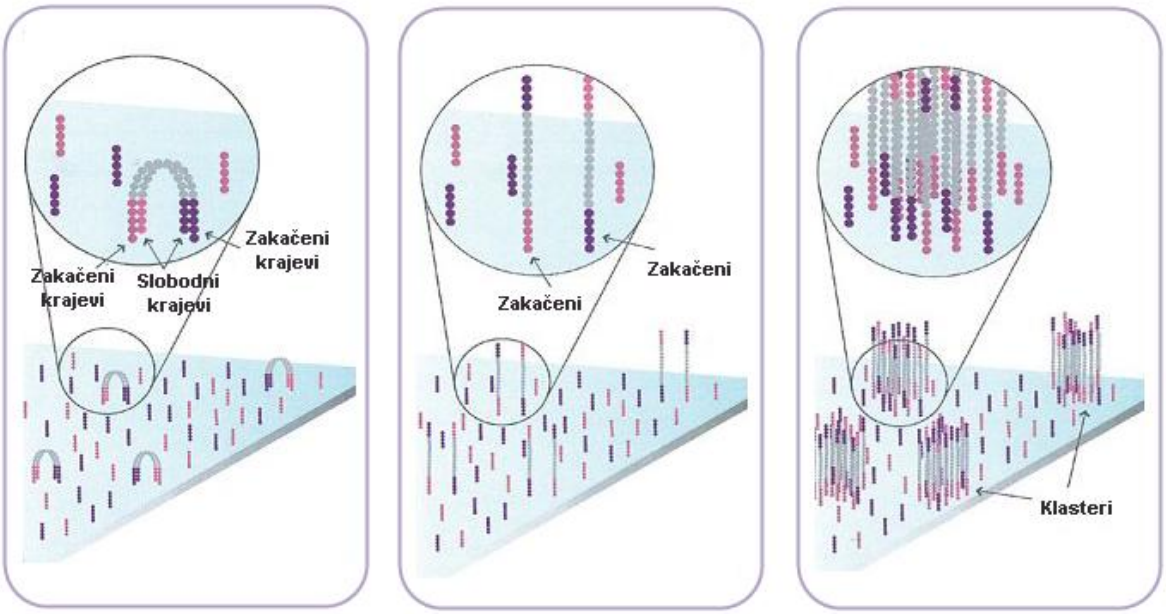
[8] Prajmeri su veštački dodati lanci koji se kače na ploču kako bi se omogućilo da se DNK fragmenati preko njih zakače za površinu.



Slika 7. Skeniranje koraci 1-3[9]

Sintezom drugog lanca DNK, dobija se dvostruki most, pri čemu su suprotni krajevi lanaca zakačeni za površinu (**Slika 8 - 1**). Nakon toga se raskidaju vodonične veze između dva lanca i dobijaju se dva odvojena komplementarna lanca DNK (**Slika 8 - 2**). Kako bi se baze lakše prepoznavale, lanci se umnožavaju i obrazuju grupe "identicnih" lanaca koje se nazivaju klasteri (**Slika 8 - 3**).

[9] Slika preuzeta sa: https://www.uppnex.uu.se/uppnex-book/technologies/solexa-sequencing

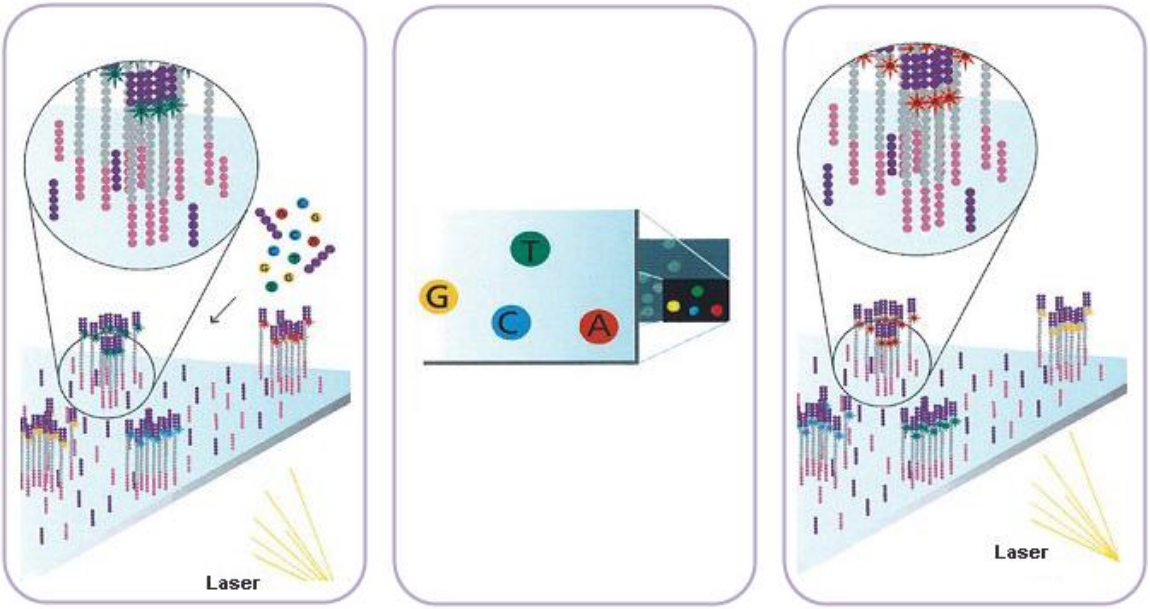


Slika 8. Skeniranje koraci 4-6[10]

Sledeći korak je prepoznavanje nizova baza koje čine lance. Potrebno je redom prepoznavati baze, počevši od slobodnog kraja lanca. Unapred je poznata dužina fragmenta koju je potrebno prepoznati[11]. Pri prepoznavanju svake baze dodaju se fluorescentno označeni nukleotidi, prajmeri i DNK polimeraza (**Slika 9 - 1**). Kako bi se prepoznala jedna baza, odgovarajući fluorescentno označen nukelotid se kači na nju i uz pomoć lasera se dobija slika na osnovu koje se prepoznaje koja baza je u pitanju (**Slika 9 - 2**).

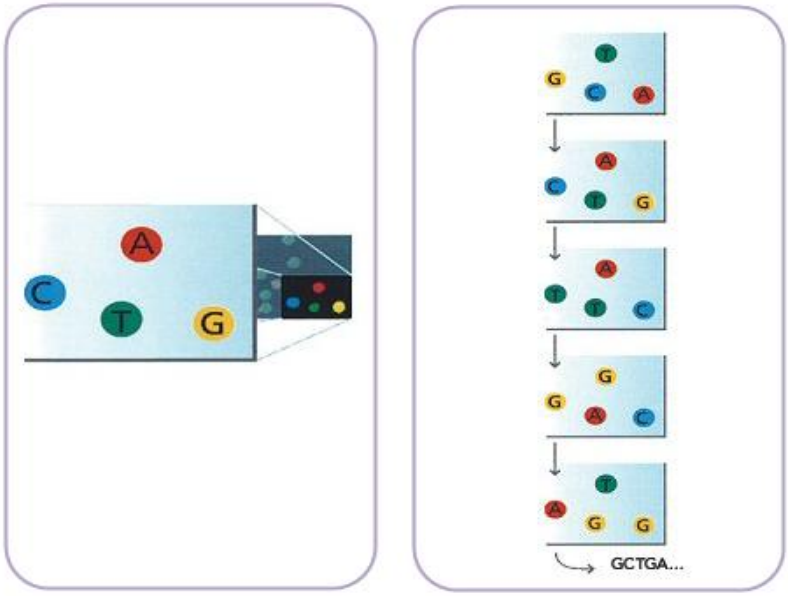
[10] Slika preuzeta sa: https://www.uppnex.uu.se/uppnex-book/technologies/solexa-sequencing

[11] Dužina fragmenata je tehnička karakteristika sistema za sekvenciranje.



Slika 9. Skeniranje koraci 7-9[12]

Ovaj korak se nastavlja sve dok se ne prepozna potreban broj nukleotida sa svakog lanca. Nakon toga se analizom slika dobija skenirana sekvenca DNK (**Slika 10**).



Slika 10. Skeniranje koraci 10-11[12]

[12] Slika preuzeta sa: https://www.uppnex.uu.se/uppnex-book/technologies/solexa-sequencing

Skeniranje fragmenata se može obavljati samo sa jedne (*single-end*) ili sa obe strane (*paired-end*) fragmenta. Pri simulaciji bi trebalo omogućiti izbor načina skeniranja. Ukoliko se podaci skeniraju sa jedne strane, kao izlaz bi trebalo da se napravi samo jedna tekstualna datoteka sa skeniranim fragmentima. Ukoliko se podaci skeniraju sa obe strane fragmenata, potrebna je još jedna tekstualna datoteka, pored već pomenute. U njoj bi se nalazili skenirani fragmenti počevši od pozadi. Kako se skenira fiksan broj azotnih baza, a dužina fragmenata koje treba skenirati je uglavnom veća, skenirani delovi sa jedne strane ne moraju da budu isti kao delovi skenirani sa suprotne strane. Pored toga što se baze čitaju u suprotnom smeru, u drugu tekstualnu datoteku bi trebalo da se upisuje njihov komplement[13]. Razlog za to se može videti na slici (**Slika 8 - 2**): nakon kreiranja dvostrukog mosta i otpuštanja slobodnih krajeva, skeniraju se oba otpuštena lanca, počevši odozgo na dole. Jedan od lanaca je početni fragment, dok drugi predstavlja njegov komplementarni lanac. Skeniranjem prvog od njih, dobija se fragment skeniran od početka. Skeniranjem drugog, dobija se komplementarni lanac i to skeniran u suprotnom smeru.

[13] Komplement azotne baze je baza sa kojom se ona povezuje vodoničnim vezama. Komplement adenina je timin i obrnuto, dok komplement citozina predstavlja guanin i obrnuto.

3 Implementacija simulacije

Zadatak programa je simuliranje prethodno opisanog hemijskog postupka sekvenciranja genoma. Svaki od opisanih koraka sekvenciranja genoma je na odgovarajući način obuhvaćen ovim programom.

Programski jezik korišćen pri razvoju ovog programa je C++. Kao razlog navela bih brzinu izvršavanja, kao i veću kontrolu nad memorijom. Program je napisan tako da koristi komandni interfejs. Kao razvojno okruženje korišćen je Microsoft Visual Studio 2010. Program je rađen po standardu, preveden je i proveren i prevodiocem g++.

3.1 Zahtevi simulacije

Analizom ciljeva koje bi simulacija trebalo da ispuni dolazi se do sledećih funkcionalnih zahteva:

- Napraviti DNK lanac na osnovu zadatog procenta guanina i citozina u lancu
- Umnožiti napravljeni lanac DNK tako da se dobije odgovarajući broj kopija
- Iseckati dobijene lance na pojedinim delovima
- Simulirati skeniranje iseckanih fragmenata

3.2 Algoritam

Program se sastoji iz dva dela od kojih prvi služi za pravljenje DNK lanca, dok je drugi deo sekvenciranje tog lanca.

Ulaz u algoritam:

- dužina DNK lanca
- ukupan procenat citozina i guanina
- očekivana količina uzoraka nakon replikacije (broj kopija)
- procenat greške pri replikaciji DNK lanaca
- uzorci na osnovu kojih se vrši sečenje lanaca
- najmanja dužina isečenih fragmenata

- najveća dopuštena dužina fragmenata za koju se ne mora ići u dalje deljenje
- dužina fragmenta koja se skenira i snima u datoteku
- da li se radi skeniranje sa jedne ili sa obe strane fragmenata

Algoritam

pravljenje DNK lanca sa zadatim procentima azotnih baza;
smeštanje početnog DNK lanca u tekstualnu datoteku;

umnožavanje;

seckanje i snimanje u datoteke;

Izlaz iz algoritma:

- tekstualna datoteka sa lancima zadate dužine skeniranih sa jedne strane DNK lanca

Ukoliko je naznačeno da se skenira sa obe strane, izlaz je i

- tekstualna datoteka sa lancima zadate dužine skeniranih sa druge strane DNK lanca

Korak **umnožavanja** se odvija na sledeći način:

```
procitaj i upisi pocetni lanac u binarnu datoteku;
i = 0;
while ( napravljeneNiske.size() < brojNizova )
begin
        if (napravljeneNiske[i].potrebanBrojKopija > 0 )
        begin
              niskaI = pročitaj nisku i;
              for ( j= napravljeneNiske[i].potrebanBrojKopija - 1; j >= 0; j-- )
              begin
                    novaNiska = kopija (niskaI);
                    dodaj podatke za "novaNiska" u "napravljeneNiske";
                    upisi nisku "novaNiska" u binarnu datoteku;
              end
        end
        i++;
end
```

Korak **seckanja i snimanja u datoteke** se predstavlja na sledeći način:

```
for ( i = 0; i < napravljeneNiske.size(); i++ )
begin
    niskaI = procitaj nisku i;
    dugiFragmenti.isprazni();
    dugiFragmenti.dodaj(niskaI);
    kratkiFragmenti.isprazni();

    while (dugiFragmenti not empty)
    begin
        trenutnaNiska = dugi.prviElement();
        dugiFragmenti.obrisiPrviElement();
        izaberi slučajan broj u opsegu od 0 do duzine niske " trenutnaNiska ";
        nađi najbliži mogući presek izabranom broju;
        preseci nisku trenutnaNiska na delove s1 i s2;
        if (s1 duga)
            dugiFragmenti.dodaj(s1);
        else
            kratkiFragmenti.dodaj (s1);

        if (s2 duga)
            dugiFragmenti.dodaj(s2);
        else
            kratkiFragmenti.dodaj(s2);
    end

    foreach (niska in kratkiFragmenti)
        skeniraj i snimi u datoteke trenutnu nisku;
end
```

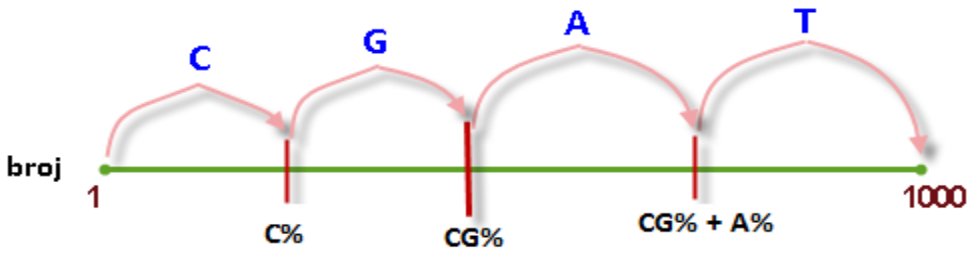
3.3 Implementacija algoritma

3.3.1 Pravljenje DNK lanca

DNK lanac je niz azotnih baza. Azotne baze koje ulaze u sastav DNK su adenin (A), citozin (C), guanin (G) i timin (T). Zbog toga se DNK lanac predstavlja kao niska karaktera koja je sačinjena od slova A, C, G i T. Kako bi se napravio veštački DNK lanac, neophodni parametri su željena dužina lanca i procenat jedne od azotnih baza. Kao procenat azotnih baza se zadaje ukupan procenat citozina i guanina (CG%). Na osnovu tog procenta se računaju procenti pojedinačnih azotnih baza koji bi trebalo da se nalaze u novom lancu. Procenat adenina treba da bude približno jednak procentu timina, pored toga i procenat citozina bi trebalo da bude približan procentu guanina. Ukoliko je maksimalno odstupanje pojedinačne baze od polovine vrednosti određene procentom CG $(x = 2\%)$ i $rand(x)$ realan broj u opsegu $[0, x]$, onda se procenti baza izračunavaju:

- procenat citozina = $CG\%/2 \ - x + rand(2 * x)$
- procenat guanina = $CG\% - "procenat\ citozina"$
- procenat adenina = $(100 - CG\%)/2 - x + rand(2 * x)$
- procenat timina = $(100 - CG\%) - "procenat\ adenina"$

Broj azotnih baza u okviru DNK lanca treba da bude jednak željenoj dužini. Svaka azotna baza se dobija tako što se na slučajan način bira jedan broj od 1 do 1000[14] (uključujući 1 i 1000). Nakon toga se, u zavisnosti od opsega kome dobijeni broj pripada, određuje baza. Kako bi zadati procenti bili ispunjeni, baza se bira na sledeći način (**Slika 11**):



Slika 11. Određivanje azotne baze pri pravljenju DNK lanca

[14] Radi se sa 1000 kako bi tačnost procenata bila veća.

Priložen je deo koda koji se bavi pravljenjem veštačkog DNK lanca:

```
int deviation = 2; //maksimalno odstupanje pojedinacne baze od jednakosti sa
                     komplementom = 2%

char* sequence = new char[length];
percentCG = percentCG * 10;
deviation = deviation * 10;

int percentC = percentCG/2 - deviation + rand() % (deviation*2+1);
int percentA = (1000-percentCG)/2 - deviation + rand() % (deviation*2+1);
int percentCGA = percentCG + percentA;

for(int i=0; i<length; i++)
{
       int randNumber = rand() % 1000 + 1; //slucajan broj od 1 do 1000

       if(randNumber <= percentC)
              sequence[i] = 'C';
       else if (randNumber <= percentCG)
              sequence[i] = 'G';
       else if(randNumber <= percentCGA)
              sequence[i] = 'A';
       else
              sequence[i] = 'T';
}
```

Rezultat rada ovog dela programa je tekstualna datoteka koja sadrži napravljen lanac DNK željene dužine i sa zadatim procentima azotnih baza.

3.3.2 Sekvenciranje

Sekvenciranje se sastoji iz tri koraka:

- replikacija,
- seckanje,
- skeniranje.

Ulazni podatak za ovaj deo programa je tekstualna datoteka sa početnim DNK lancem, koja može biti napravljena ili poticati iz drugog izvora.

3.3.2.1 Replikacija

Replikacija predstavlja proces umnožavanja DNK lanaca. Parametar potreban u ovom delu programa je broj kopija koje je potrebno imati na kraju procesa. Pored broja kopija, potreban podatak je i procenat greške pri svakom kopiranju.

Dužine DNK niski mogu biti izuzetno velike, čak i do nekoliko GB. Takođe, nakon procesa replikacije očekuje se i veliki broj kopija ovih niski. Kako bi se izbeglo skladištenje svih niski u memoriji, za ove potrebe koristi se privremena binarna datoteka.

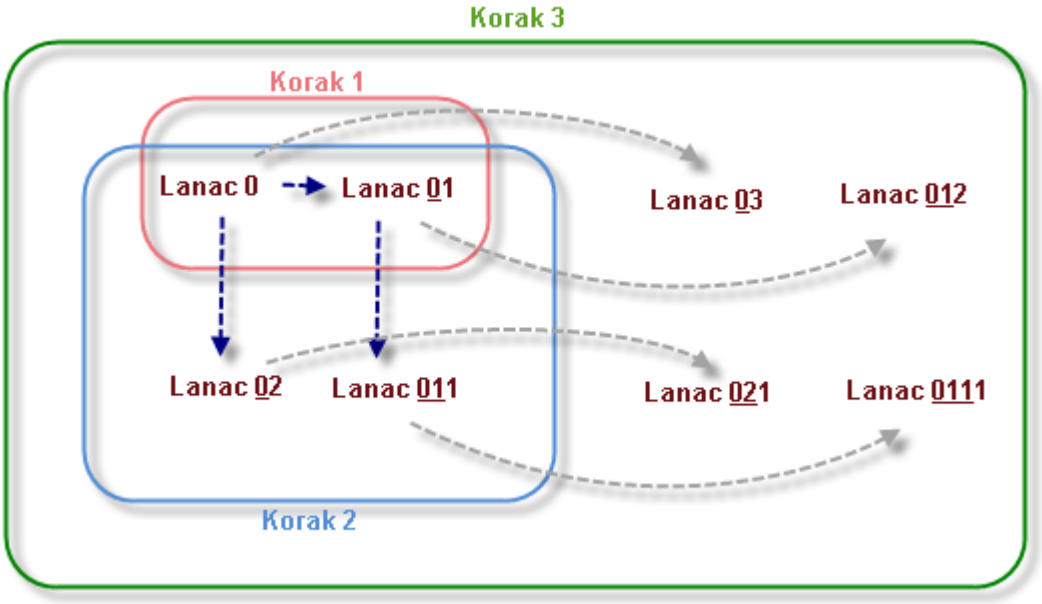
Kako bi se razlikovale različite DNK niske u okviru dugačke binarne datoteke, korišćen je niz pomoćnih struktura u kojima se čuvaju potrebni podaci o svakom DNK lancu. Podaci o umnoženim lancima DNK se čuvaju samo u memoriji.

Korišćena struktura SequenceInfo, sadrži informacije o jednoj niski koja se nalazi u binarnoj datoteci i ima oblik:

```
struct SequenceInfo
{
	__int64 offset; //rastojanje od pocetka fajla gde ta niska pocinje
	__int64 length; //duzina niske
	int copyCount; //koliko puta bi tu nisku trebalo kopirati
};
```

Kao što se može videti u prikazanoj strukturi, neophodni parametri su dužina trenutne niske, rastojanje od početka datoteke gde ta niska počinje kao i broj potrebnih kopiranja. Dužina trenutne niske je neophodna informacija iz razloga što se nakon replikacije dužina niske može razlikovati od dužine početne niske. Razlike u dužini se javljaju zahvaljujući greškama u vidu gubljenja nukleotida (delecija) koje smanjuje polaznu dužinu, ili dodavanja nukleotida (insercija) zbog koje se dužina lanca povećava. Pored dužine, potreban je i podatak o mestu u binarnoj datoteci gde taj lanac počinje. U prikazanoj strukturi, taj podatak je *offset* i označava rastojanje od početka datoteke na kom se nalazi trenutni DNK lanac. Na osnovu podataka o dužini i položaju u odnosu na početak datoteke, moguće je izdvojiti željeni DNK lanac.

Razlog u postojanju trećeg parametra, odnosno potrebnog broja kopiranja trenutnog lanca je u algoritmu replikacije. Kako bi se ubrzao proces replikacije i smanjio broj čitanja iz datoteke, izmenjen je redosled kojim bi trebalo da se odvija replikacija. Umesto da se u svakom sledećem krugu svaka postojeća niska čita iz datoteke i kopira, uočena je pravilnost u broju kopija koje je potrebno napraviti od svake niske. Ideja je da se tačno jedanput pročita svaka niska koju treba kopirati i zatim napravi potreban broj kopija.



Slika 12. Prikaz pravilnosti u broju potrebnih replikacija svake niske

Početni DNK lanac treba kopirati onoliko puta koliko puta treba ponoviti replikaciju. Na slici (**Slika 12**) je ilustrovan proces replikacije koji je ponovljen tri puta. Svaki korak replikacije zaokružen je radi preglednosti. "Lanac 0" je početni lanac. Na slici se, na osnovu prefiksa oznake lanca (podvučenog dela) može zaključiti od kog lanca je nastao posmatrani lanac. Kao što se može primetiti "Lanac 0", odnosno početna niska, kopiran je onoliko puta koliko je bio koraka ("Lanac 01", "Lanac 02" i "Lanac 03"). Svaki od ovih lanaca redom je potrebno kopirati za jedan manje puta: "Lanac 01" kopiran je dva puta ("Lanac 011" i "Lanac 012"), "Lanac 02" jednom ("Lanac 021"), dok "Lanac 03" nije kopiran nijednom. Rezultat ovakvog kopiranja će biti isti kao da je prvi lanac kopiran jednom tako da se dobiju dva lanca. Zatim oba postojeća lanca kopirana tako da se dobiju četiri lanca i na kraju svaki od ova četiri lanca kopiran tako da se dobije osam lanaca. Pri svakom kopiranju lanca potrebno je popuniti strukturu *SequenceInfo* odgovarajućim podacima i za potreban broj kopiranja (*copyCount*) staviti odgovarajući broj na osnovu prethodnog pravila čiji se formalni opis može videti u algoritmu.

Implementacija replikacije

```
vector<SequenceInfo> sequenceInfo; //vektor koji sadrzi informacije o niskama
                                        smestenim u binarni fajl

SequenceInfo si;
si.length = arguments.dnaLen;
si.offset = 0;
si.copyCount = (int)(log((float)(arguments.copyCount-1))/log((float)2) + 1); //broj
                                                    prolaza je log2(brojNizova-1) + 1
sequenceInfo.push_back(si);

unsigned int seqNumber =  (unsigned int)arguments.copyCount;
__int64 offset = sequenceInfo[0].offset + sequenceInfo[0].length;

fstream f("tmp.bin", ios::in|ios::out|ios::trunc);
writeSequence(f, 0, initialSequence);

int i = 0;
while (sequenceInfo.size() < seqNumber)
{
	if(sequenceInfo[i].copyCount > 0)
	{
		char* current = new char[sequenceInfo[i].length];

		current = readCurrentSequence(f, i);

		for(int j=sequenceInfo[i].copyCount-1; j>=0; j--)
		{
			//Transformacija trenutne niske
			string copy(current, sequenceInfo[i].length);

			__int64 newLength = 0;
			newLength = sequenceInfo[i].length;
			replication(copy, newLength);  //kopira se

			//popunjavaju se informacije o novoj nisci
			SequenceInfo ni;
			ni.length = newLength;
			ni.offset = offset;
			ni.copyCount = j;

			writeSequence(f, offset, copy);

			sequenceInfo.push_back(ni);
			offset += ni.length;
			//delete kopija;

			if(sequenceInfo.size() == seqNumber)
				break;
		}

		delete[] current;
	}
	i++;
}
```

Proces replikacije jedne DNK niske predstavlja njeno kopiranje uz primenjivanje određenog broja mutacija. Neophodan podatak je očekivani procenat greške prilikom kopiranja jednog DNK lanca. Očekivani broj mutacija koje je potrebno primeniti nad niskom se računa na osnovu njene dužine i procenta greške, kao $duzina * \frac{procenatGresaka}{100}$. Kako broj grešaka ne bi bio jednak pri svakom kopiranju, broj mutacija se računa kao približna vrednost očekivanog broja mutacija. Dopušteno odstupanje je deseti deo očekivanog procenta grešaka. Mutacije koje je moguće izvršiti su delecija (brisanje neke od azotnih baza), substitucija (zamena jedne azotne baze drugom) i insercija (umetanje azotne baze na novo mesto u okviru lanca). Kada je poznat broj grešaka koje je potrebno napraviti u jednom lancu, svaka od grešaka se određuje na slučajan način. Potrebno je odrediti mesto u lancu na kome će biti sprovedena mutacija. Mesto se određuje slučajnim izborom pozicije u okviru lanca ($rand()\ \%\ duzina$). Pored mesta, potrebno je na slučajan način izabrati mutaciju koja se primenjuje. Kako postoje tri vrste mutacije, bira se slučajan broj od nula do dva i na osnovu dobijenog broja određuje se potrebna mutacija ($0 \rightarrow delecija, 1 \rightarrow insercija, 2 \rightarrow substitucija$). Ukoliko je izabrana mutacija insercija ili substitucija, potrebno je slučajno izabrati i bazu koja se ubacuje ili kojom se zamenjuje postojeća. Baza se bira slučajnim izborom broja u opsegu od nula do tri ($0 \rightarrow Adenin, 1 \rightarrow Citozin, 2 \rightarrow Guanin, 3 \rightarrow Timin$). Prilikom izbora insercije i delecije, moguće je umetanje, odnosno brisanje više od jedne baze na određenoj poziciji. To se rešava izborom slučajnog broja na osnovu koga se proverava da li je potrebno ponoviti odgovarajući proces. Određeno je da se proces ponavlja u dvadeset procenata slučajeva. Substitucija se, u programu, uvek izvršava nad jednom azotnom bazom.

Implementacija replikacije jedne niske

```
void Functions::replication(string &copy, __int64 &length)
{
	float numErrors = length * arguments.errorPercent/100.0f;
	int addition = floor( numErrors * 0,1f + 0.5f );
	numErrors =  (float)(( rand() % ( addition*2 + 1 )) + numErrors - addition );
	int errorNumber = (int)floor( numErrors + 0.5f );

	for(int i = 0; i < errorNumber; i++)
	{
		int errorType = rand() % 3;
		int index = rand() % length;

		int tmp;
		switch( errorType )
		{
			case 0: //delecija
				do
				{
					copy.erase( index, 1 );
					length--;
				}
				while( rand() % 10 >= 8 );
				break;
			case 1: //insercija
				do
				{
					copy.insert( index, 1, randomBase() );
					index++;
					length++;
				}
				while( rand() % 10 >= 8 );
				break;
			case 2: //substitucija
				copy.replace( index, 1, randomBase() );
				break;
		}
	}
}


//Funkcija koja na slučajan način određuje azotnu bazu
char bases[] = "ACGT";
char Functions::randomBase()
{
	return bases[ rand() % 4 ];
}
```

3.3.2.2 Seckanje i skeniranje

Replikacijom je dobijen određeni broj kopija koje je potrebno iseckati i zatim skenirati. Kako se ne bi ista niska dva puta čitala iz binarne datoteke, seckanje i skeniranje su spojeni u jednu celinu. Svaka od kopija se redom čita i pušta u proces seckanja. Odmah po završetku seckanja jedne kopije, dobijeni fragmenti se redom skeniraju i rezultati se snimaju u odgovarajuće datoteke.

Seckanje se odvija uz pomoć određenih uzoraka koji služe za prepoznavanje mesta gde treba preseći lanac. Svaki uzorak se sastoji od nekoliko azotnih baza. Uzorci se čitaju iz tekstualne datoteke koja se zadaje kao parametar. Ukoliko postoji neki od ponuđenih uzoraka, lanac se preseca na samom početku uzorka, pre prve baze. Kako nijedan isečeni fragment ne bi bio manji od neke željene vrednosti, uveden je parametar *minCutLen*, koji predstavlja minimalnu dozvoljenu dužinu isečenih delova. Ukoliko je *minCutLen* manji od dužine skeniranja, nakon seckanja mogu postojati fragmenti koji nemaju dovoljnu dužinu da bi bili skenirani. U praksi se zajedno sa njima skenira i deo dodatih adaptera. U ovoj simulaciji je odlučeno da se ti fragmenti ne skeniraju. Pored minimalne dozvoljene dužine isečenih delova, uveden je i parametar *maxCutLen* koji predstavlja maksimalnu dužinu fragmenata za koju nije potrebno ići u dalje deljenje. Ipak, određen procenat fragmenata koji su između dužina *minCutLen* i *maxCutLen* je potrebno preseći. Taj procenat je moguće podesiti parametrom *shortCutPercent.*

Čim se završi seckanje jedne niske, dobijeni fragmenti se šalju u proces skeniranja. Skeniranje je moguće obavljati na dva načina. Prvi je skeniranje samo sa jedne strane (*single-eng*), dok je drugi skeniranje sa obe strane (*paired-end*) fragmenta.

U programu se skeniranje simulira tako što se svaki fragment dobijen sečenjem čita sa oba kraja. Sa jednog kraja se jednostavno čita određeni broj azotnih baza i dobijena sekvenca se zapisuje u izlaznu datoteku. Sa drugog kraja je proces čitanja malo komplikovaniji. Kako bi se dobio efekat čitanja sa suprotne strane lanca, određeni broj baza, do kraja fragmenta, se pročita, zatim se dobijena sekvenca okrene. Pored toga, svaka pročitana baza se zamenjuje svojim komplementom. Tek tako dobijena sekvenca se snima u drugu izlaznu datoteku.

Izlaz iz programa predstavlja tekstualna datoteka sa skeniranim fragmentima sa jedne strane. Ukoliko je potrebno skenirati fragmente i sa druge strane, pored nje, izlaz je još jedna tekstualna datoteka sa skeniranim komplementarnim fragmentima sa suprotne strane.

3.4 Format zapisa izlaznih podataka

Izlazni podaci se upisuju u datoteku u formatu FASTA [9]. Format FASTA je tekstualni format za predstavjanje nukleotidnih sekvenci ili sekvenci proteina. Nukleotidi i aminokiseline su predstavljeni jedinstvenim jednoslovnim kodovima. Naziv FASTA potiče od naziva programa i skraćenica je od FAST - All. Taj program je unapređenje FAST - P (protein) i FAST - N (nukleotid) jer podržava oba tipa podataka. Format je vremenom postao standardno korišćen format u bioinformatici.

Format FASTA dozvoljava da nazivi sekvenci i komentari prethode samoj sekvenci. Kao velika prednost ističe se i lako pristupanje podacima parsiranjem uz pomoć raznih skript jezika.

Opis sekvence u formatu FASTA se sastoji od [9]:

- jednog reda zaglavlja koji počinje simbolom ">", nakon koga sledi naziv sekvence i opciono drugi potrebni podaci o sekvenci, kao i komentari, međusobno razdvojeni znakom "|" i
- jednog ili više redova koji sadrže zapise baza (ili aminokiselina) koje čine sekvencu.

Primer zapisa sekvence u formatu FASTA [10]:

```
>gi|532319|pir|TVFV2E|TVFV2E envelope protein
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
```

3.5 Parametri i njihov uticaj

Svi parametri potrebni za rad programa se prosleđuju kao argumenti komandne linije. Program se poziva na sledeći način: GenomeSequencing [<options>] <outputFilenameBase>.

<outputFilenameBase> predstavlja naziv datoteke u koju će po izvršenju programa biti smešteni rezultati skeniranja. Ukoliko je potrebno vršiti skeniranje sa obe strane fragmenata, navedenom nazivu će biti dodati nastavci '_s1' za jednu i '_s2' za drugu izlaznu datoteku. Ovaj parametar je obavezno navesti. Svi ostali parametri su opcioni.

Za sve parametre sem 'pairedEnd' je neophodno navesti njihovu vrednost. Sintaksa koja se koristi je **-parametar <argument>**, gde je 'parametar' naziv parametra, a 'argument' njegova vrednost.

3.5.1 Parametri koji utiču na pravljenje DNK lanca

-dnaIn <dnaFile> - Datoteka koja sadrži DNK sekvencu u FASTA formatu. Zadavanjem postojeće DNK sekvence mogu se uporediti stvarni rezultati sekvenciranja sa rezultatima dobijenim simulacijom. Pored toga, tako se omogućuje i višestruko sekvenciranje iste sekvence. Po podrazumevanoj vrednosti, ukoliko se ovaj parametar izostavi, program će napraviti novu DNK sekvencu i nad njom vršiti sekvenciranje.

-dnaOut <dnaFile> - Datoteka u koju se zapisuje napravljena DNK sekvenca. Ukoliko se ovaj parametar izostavi, DNK sekvenca korišćena u simulaciji neće biti zapisana.

-dnaLen <length> - Dužina napravljenog DNK lanca. Podrazumevana vrednost ovog parametra je 10^6 azotnih baza. Ovo je jedan od glavnih parametara koji utiče na dužinu izvršavanja simulacije, kao i na potrebnu memoriju. U tabeli (**Tabela 1**) se mogu videti veličine genoma za neke organizme [11].

Organizam	Veličina genoma (parova baza)
Fag F-X17 (Virus)	**5386**
fag λ (Virus)	**5×10^4**
Escherichia coli (Bakterija)	**4×10^6**
Amoeba dubia (Ameba)	**67×10^{10}**
Fritillary assyrica (Biljka)	**13×10^{10}**
Saccharomyces cerevisiae (Kvasac)	**2×10^7**
Caenorhabditis elegans (Valjkasti crv)	**8×10^7**
Drosophila melanogaster (Insekt)	**2×10^8**
Homo sapiens (Sisar)	**3×10^9**

Tabela 1. Veličine genoma nekih organizama

-gcPercent <percent> - Ukupan procenat citozina i guanina u napravljenom lancu. Promena ovog parametra utiče na strukturu početnog DNK lanca. Povećanjem ili smanjivanjem parametra povećava se, odnosno smanjuje, broj citozina i guanina u okviru lanca. Podrazumevana vrednost ovog parametra je 45 procenata. On određuje procenat zastupljenosti azotnih baza (adenina - A, guanina - G, citozina - C i timina - T) u DNK lancu koji se pravi. Pojedinačni procenti se računaju na osnovu ovog parametra na ranije opisan način (strana 20). Procenti citozina i guanina su približno jednaki polovini ove vrednosti, dok su procenti adenina i timina približno jednaki polovini preostalih procenata.

3.5.2 Parametri koji utiču na replikaciju

-copyCount <copyCount> - Očekivani broj kopija DNK lanca po završetku procesa replikacije. Podrazumevana vrednost ovog parametra je 500. Na osnovu ovog parametra može se izračunati približan broj prolaza kroz proces replikacije i on iznosi $[\log_2(\text{brKopija} - 1)] + 1$. Povećanjem vrednosti ovog parametra znatno se povećava obim posla i produžava izvršavanje programa.

-errorPercent <percent> - Procenat grešaka pri kopiranju jednog DNK lanca. U praksi se greške obično kreću u rasponu od jedne greške u deset miliona nukleotida (10^{-7}) do jedne greške u deset milijardi nukleotida (10^{-10}). Pored grešaka pri replikaciji, u praksi se javljaju i greške u procesu skeniranja fragmenata. Pošto ove greške nisu simulirane, simulirana greška pri replikaciji je nešto veća od uobičajene greške u praksi. Podrazumevana vrednost ovog parametra je 0,0001 procenat, tj. kao podrazumevan broj grešaka se pravi jedna greška u milion nukleotida (10^{-6}). Podrazumevana vrednost parametra označava da za podrazumevanu dužinu lanca koja iznosi 10^6 azotnih baza, broj grešaka pri kopiranju lanca iznosi približno 1. Smanjivanjem ovog parametra povećava se tačnost replikacije DNK lanca.

3.5.3 Parametri koji utiču na sečenje lanaca

-cutPatterns <patterns> - Datoteka u kojoj se nalaze uzorci na osnovu kojih se vrši sečenje lanaca. Podrazumevana vrednost ovog parametra je "Patterns.txt". Primer sadržaja ove datoteke je:

```
AACG
AATG
CAT
```

Seckanje lanca se obavlja tako što se u okviru njega traže ovi uzorci. Ukoliko se naiđe na neki od njih, lanac se tu preseca.

-minCutLen <minLen> - Minimalna dužina fragmenata nakon seckanja. Ukoliko se pokuša sečenje tako da je jedan od delova manji od *minCutLen*, takvo sečenje se ne dozvoljava. Podrazumevana vrednost parametra je 40. Na osnovu toga se može zaključiti da nijedan fragment dobijen sečenjem neće biti kraći od 40 baza. Ukoliko se ne navede željena vrednost ovog parametra, on će biti postavljen na najmanju od sledećih vrednosti: podrazumevana vrednost (40), *maxCutLen* i *scanLen*.

-maxCutLen <maxLen> - Najveća dopuštena dužina sekvence za koju se ne mora ići u dalje deljenje. Podrazumevana vrednost ovog parametra je 150 baza ukoliko se skeniranje vrši sa jedne i 300 baza ukoliko se skeniranje vrši sa obe strane. Sve sekvence duže od *maxCutLen* se moraju dalje deliti, ukoliko je to moguće. Za jedan deo sekvenci kraćih od *maxCutLen* a dužih od *minCutLen* se traži presek, dok ostale ostaju nepresečene. Koliki procenat takvih sekvenci treba preseći određuje parametar *shortCutPercent.* Ukoliko se ne navede vrednost parametra *maxCutLen*, ona će biti postavljena na najveću od vrednosti: podrazumevana vrednost (150/300), *minCutLen* i *scanLen*.

Da bi simulacija bila moguća, potrebno je voditi računa o tome da vrednost ovog parametra ne sme da bude manja od vrednosti parametara *minCutLen* i *scanLen*.

-shortCutPercent <percent> - Procenat sekvenci dužine između *minCutLen* i *maxCutLen* koje je potrebno dalje deliti. Podrazumevana vrednost ovog parametra je 30. Povećanjem ove vrednosti dobijaju se kraći fragmenti nakon seckanja.

3.5.4 Parametri koji utiču na skeniranje

-scanLen <length> - Dužina sekvenci koje se skeniraju i snimaju. Podrazumevana vrednost ovog parametra je 100.

Ukoliko je vrednost parametra *minCutLen* manja od vrednosti parametra *scanLen*, seckanjem je moguće dobiti fragmente kraće od dužine neophodne za skeniranje. Ti fragmenti neće biti skenirani.

-pairedEnd - Skeniranje fragmenata sa obe strane. Skeniranje može da bude jednostrano (*single-end* - skeniranje fragmenata samo sa jedne strane) i dvostrano (*paired-end* - skeniranje fragmenata sa obe strane). Ukoliko se radi jednostrano skeniranje, skenira se određena dužina fragmenta (*scanLen*), od početka. Dobijeni podaci se smeštaju u

jednu tekstualnu datoteku. Kod dvostranog skeniranja kao rezultat se dobijaju dve tekstualne datoteke. Prva je popunjena podacima skeniranim od početka fragmenata (kao i kod jednostranog skeniranja). Druga datoteka sadrži skeniran komplement kraja fragmenta.

4 Diskusija

4.1 Problemi pri izradi programa

Najveći problem pri izradi programa je predstavljala veličina podataka kojima treba rukovati. Jedan DNK lanac se može sastojati i od preko 10^9 azotnih baza, tako da je neophodno raditi sa niskama karaktera tih dužina.

4.2 Brzina izvršavanja

Brzina izvršavanja programa bila je najveći problem na koji se naišlo pri izradi rada. Razlog tome su jako dugačka početna DNK niska, njeno brzo umnožavanje kao i transformacije koje je potrebno izvršiti nad svakom od njih.

Radi poređenja vremena izvršavanja, simulacija je izvršavana na laptop računaru (*CPU: Intel i7, 1,73 GHz; RAM: 8.00 GB, DDR3 1333 MHz*) nekoliko puta sa različitim parametrima.

U poslednjoj verziji, korišćenjem opisanih ideja, program se uz podrazumevane početne parametre izvršava za manje od sat vremena. Replikacija je izvršena za nekoliko sekundi, dok preostali deo, seckanje i snimanje u datoteke traje oko 50 minuta.

Neki parametri veoma utiču na brzinu izvršavanja programa, na primer dužina početnog lanca i broj kopija:

- Za *dnaLen* = 10^5 i *copyCount* = 500, vreme trajanja replikacije je 0.45 sekundi, dok je vreme trajanja seckanja i skeniranja približno minut.
- Za *dnaLen* = $5*10^5$ i *copyCount* = 500, vreme trajanja replikacije je 2 sekunde, dok je vreme trajanja seckanja i skeniranja približno 10 minuta.
- Za *dnaLen* = $5*10^5$ i *copyCount* = 1000, vreme trajanja replikacije je 6,6 sekundi, dok je vreme trajanja seckanja i skeniranja približno 20 minuta.

4.3 Ilustracija simulacije

Radi ilustracije međukoraka i rezultata, u ovom poglavlju prikazan je manji primer sa sledećim parametrima:

- dnaLen = 100
- copyCount = 8
- minCutLen = 5
- maxCutLen = 15
- errorPercent = 20
- scanLen = 10
- pairedEnd

Kao početni lanac dobijena je sledeća niska, dužine 100:

- CCATAAGTATGGTTAGAGATTCACCTTCTCATCACTCGCTCCCATTCGATGGGCATAGAATTAGTATG CTGTGCTAAGACTGTCTACCTTCATTGCGGGG

Da bi se dobilo 8 kopija nakon završetka replikacije, broj potrebnih prolaza kroz proces je približno $[\log_2(\text{brKopija} - 1)] + 1 = [\log_2(8 - 1)] + 1 = 3$.

Prolazom kroz 3 koraka replikacije dobijeno je sedam kopija početne niske, što uključujući i početnu predstavlja željeni broj kopija:

0. "početni lanac", dužina 100, potreban broj kopiranja 3

1. prva kopija pocetnog DNK lanca (**i = 0**): nova dužina 103, potreban broj kopiranja 2
TCCGTAAGTATGGGTTAGGATTCACCTTCACATCACTCGCTCCCATTCGATGGGTACTGAACTATGAT GATGGTGACTAAGACTATGTACCTCACTTGCGGGG

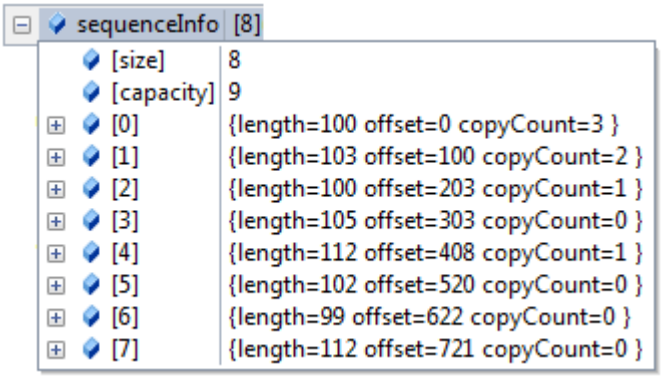
2. druga kopija pocetnog DNK lanca (**i = 0**): nova dužina 100, potreban broj kopiranja 1
CCAAAGGATGGTAATAGAGATTCACTTTCTCACTCACTCCTCCTTTGATGGCATGAAGTTAGTTGATG TCCTAAGACTGTCCTACCCCTTCATTGCGGGG

3. treca kopija pocetnog DNK lanca (**i = 0**): nova dužina 105, potreban broj kopiranja 0
CCATAAGTATGGGTACGTTCACCTCGTCTCAGTCGGTATCCCATTGGCACAATGGGTCATAAATCAGT ATGCTGCGCTCAGACATGTCTACCTTCATTGCGTGGG

4. prva kopija druge niske (**i = 1**): nova dužina 112, potreban broj kopiranja 1
   TCTGTAAGTATGGTTAGGATTACACCCCTTGCACATCACTCGCTCCCATCAGATGGGTACCTGAACTA
   TTGATGCTTCGGTACAATAGACTATTACCTGCGACGTGCGAGGG
5. druga kopija druge niske (**i = 1**): nova dužina 102, potreban broj kopiranja 0
   TCTCGTTTTATTAGTATGGTTGGATTCCTCACATCACTCGCTCCCTTATAGGGTACTGAATATTATGAT
   GGTGATAAGAATATTATACGCTCACTTGCGGGG
6. prva kopija trece niske (**i = 2**): nova dužina 99, potreban broj kopiranja 0
   CCAAAGGATGGTAATAGAGATTCATTATACTCACTCACTCCTCCGTTTTGATGGCATGAGTAGTGATT
   CCTAAACTGTCCTTACTCCCTTCATCGCGGG
7. prva kopija pete niske (**i = 4**): nova dužina 112, potreban broj kopija 0
   TGCTGTAGTACTGTTAGGATAGCACCCCTTGCACATCACCTCGGCTCCGTCCGATGGGTACCTGAACG
   TTTGATGCTTTGGTACAATAGACAATTACTGCGACGTGGAAGAG

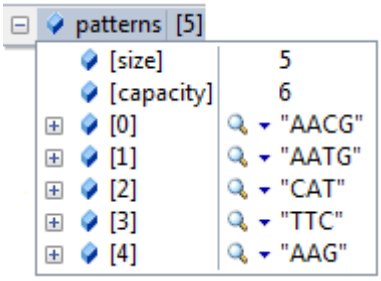
Svaka od novonastalih kopija se za nijansu razlikuje od svog izvornog lanca. Procenat greške je namerno neprirodno visok (20%) kako bi pri svakom kopiranju bilo dovoljno grešaka da se mogu primetiti i u ovako jednostavnoj simulaciji. Kako je procenat greške pri svakom kopiranju postavljen na 20, imajući u vidu dužinu početne niske koja iznosi 100, u svakoj kopiji bi trebalo da bude približno oko $100 * \frac{20}{100} = 20$ grešaka.

Nakon izvršene replikacije podaci o primercima niske u binarnoj datoteci u kojoj se čuvaju kopije bi izgledali kao na slici (**Slika 13**).



Slika 13. Prikaz podataka o binarnoj datoteci

Nakon replikacije dolazi na red sečenje i snimanje skeniranih delova svakog od dobijenih DNK lanaca. Tražeći neki od kriterijuma za sečenje koji se može videti na slici (**Slika 14**) i skeniranjem i snimanjem isečenih delova dobijaju se izlazne datoteke.



Slika 14. Prikaz uzoraka za sečenje lanaca

U tabeli se može videti deo rezultata u izlaznim datotekama (**Tabela 2**):

• **DATOTEKA 1**	• **DATOTEKA 2**
>scan_0 CCATAAGTAT >scan_1 AAGACTGTCT >scan_2 CATCACTCGC >scan_3 AAGACTATGT >scan_4 TTCGATGGGT >scan_5 TCCGTAAGTA >scan_6 TTCACCTTCA >scan_7 TTCATTGCGG >scan_8 CCAAAGGATG >scan_9 TTCTCACTCA >scan_10 CATGAAGTTA ...	>scan_0 TCTCTAACCA >scan_1 AAGGTAGACA >scan_2 AGCACAGCAT >scan_3 CCCCGCAAGT >scan_4 AGTCACCATC >scan_5 TCCTAACCCA >scan_6 TGGGAGCGAG >scan_7 CCCCGCAATG >scan_8 TCTCTATTAC >scan_9 CCATCAAAGG >scan_10 AGGACATCAA ...

Tabela 2. Primer izlaznih datoteka

4.4 Početne ideje pri implementaciji

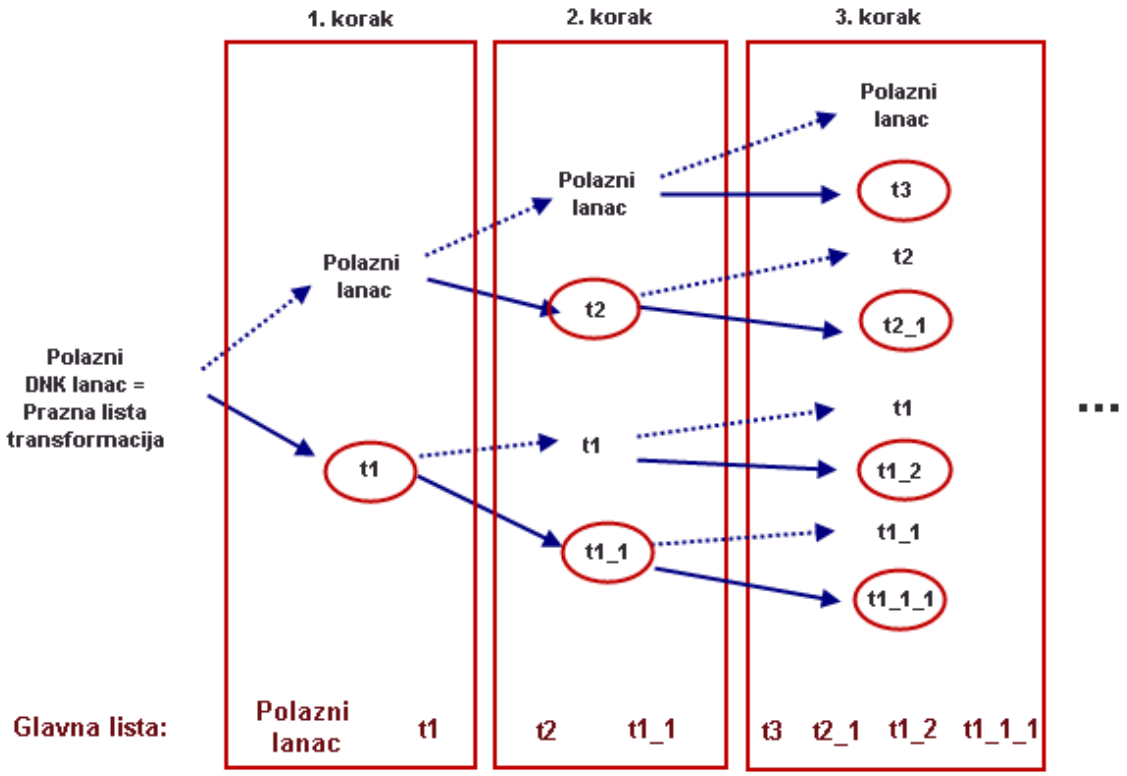
Kao najzahtevniji deo programa pokazao se korak replikacije u kome je potrebno kopirati niske sa određenim procentom greške. Kako je taj procenat greške obično mali, dobijena niska se ne razlikuje mnogo od prethodne. Kako bi se izbeglo čuvanje svih umnoženih niski, početna ideja bila je da se ne menja početna niska, već da se pamte transformacije koje su izvršene pri svakoj replikaciji. Transformacije koje je moguće izvršiti nad niskom su insercija, delecija i substitucija. Ukoliko se transformacija modelira opisnim podacima, opis bi morao da obuhvati tip transformacije (insercija, delecija ili substitucija), indeks karaktera nad kojim je transformacija izvršena, kao i bazu koja se ubacuje u slučaju insercije ili kojom se zamenjuje postojeća, u slučaju substitucije.

Ukoliko se problem posmatra na prethodno opisan način, pri svakoj replikaciji da bi se dobio novi niz dovoljno je nasumično birati transformacije i pamtiti ih u jednoj listi transformacija. Kako u sledećem koraku treba kopirati i svaki od novonastalih nizova, dovoljno je samo na već postojeće transformacije nadovezati nove transformacije. Kao rezultat replikacije dobija se lista listi transformacija. Primena jedne liste transformacija na početnu nisku predstavlja jednu njenu kopiju. Da bi se napravile replike lanaca, dovoljno je proći kroz listu i primeniti svaku listu transformacija na početnu nisku. Na taj način bi bio preskočen problem skladištenja velikog broja velikih niski tokom kopiranja. Potrebno je čuvati samo početni lanac i nad njim svaki put vršiti transformacije.

Eksperimenti su pokazali da ovakav pristup dovodi do niske efikasnosti. Kada su u pitanju manji DNK nizovi i mali procenat greške, ovakav pristup daje dobre rezultate, međutim, problem nastaje kod velikih lanaca DNK. Kako raste dužina početnog lanca DNK, čak i sa jako malim procentom greške pri replikaciji, dobija se relativno veliki broj transformacija, koje je potrebno skladištiti i nakon toga primeniti. Sa druge strane, možemo primetiti da postoji dosta neophodnog ponavljanja koje znatno usporava izvršavanje.

Označimo glavnu listu transformacija sa L. Na početku nije bitan sadržaj početnog lanca DNK, samo njegova dužina. Pre prvog kruga replikacije lista L sadrži samo jednu praznu listu transformacija koja predstavlja početnu nisku. Nakon prve replikacije, u listi L će, pored prazne liste da se nađe još jedna lista transformacija dobijena prvom replikacijom, označimo je sa t1. Na slici (**Slika 15**) može se videti način pravljenja ostalih niski i smeštanje transformacija u glavnu listu. U svakom koraku isprekidanom strelicom je predstavljen samo prelaz te liste transformacija u sledeći korak, dok je punom strelicom predstavljena replikacija, tačnije nova lista transformacija koja čini jednu kopiju. Zaokružene

transformacije su nove i potrebno je ubaciti ih u glavnu listu. Transformacije t1, t2 i t3 nastaju od prazne liste transformacija i one predstavljaju nove transformacije. Liste transformacija t1_1 i t1_2 nastaju od t1 tako što se na transformacije u listi t1 nadovežu nove transformacije. Time se postiže da je kopija koja nastaje primenom transformacija t1_1 nastala od DNK lanca koji je dobijen primenom transformacija iz liste t1.



Slika 15. Prikaz ideje sa pamćenjem transformacija

Nakon pravljenja ovakve liste transformacija, kako bi se dobile prave kopije, potrebno je izvršiti svaku od lista transformacija sadržanih u glavnoj listi. Kao što se može primetiti teško je pronaći najbolji redosled izvršavanja transformacija kako bi bilo što manje ponavljanja. Ukoliko bismo krenuli da izvršavamo transformacije redom kojim su dobijene izvršili bismo transformacije t1 nad početnim DNK lancem, nakon toga bi bile izvršene transformacije t2 nad početnim lancem, zatim transformacije t1_1 koje u sebi već sadrže transformacije t1. Time dobijamo nepotrebno ponavljanje izvršavanja transformacija t1. Svakako bi bolje bilo da kad izvršimo transformacije t1 i dobijemo željenu DNK nisku, odmah nakon toga primenimo transformacije koje slede iz transformacija t1, ali opet, pitanje je kojim redosledom. Kao što se može videti na slici već u trećem koraku postoji transformacija

t1_1_1 nastala od t1_1. Sa povećanjem broja koraka broj mogućih načina izvršavanja transformacija se povećava.

Kao najbolje rešenje neprestanog ponavljanja istih transformacija i ubrzavanje predstavljene ideje nudi se ideja dinamičkog programiranja u kojoj će se pamtiti korisni međukoraci na osnovu kojih je moguće dobiti neke naredne DNK niske. Međutim, ukoliko bi se pamtile niske u međuvremenu, cela početna ideja o smanjenju potrošnje memorije se napušta i zaključujemo da ovakav pristup nema nikakvih prednosti, čak može da bude i izuzetno spor. Umesto toga izabrano je, već opisano, korišćenje binarnih datoteka za skladištenje DNK niski.

4.5 Ideje za dalji rad

Jedna od ideja za unapređenje simulacije je unapređenje procesa replikacije. Prilikom simuliranja grešaka tokom procesa replikacije, svaka od grešaka (insercija, delecija i substitucija) je podjednako moguća. Unapređenje bi bila mogućnost zadavanja proporcija različitih vrsti grešaka.

Poboljšanje simulacije se može postići i unapređenjem procesa skeniranja. Ukoliko je *minCutLen* manje od *scanLen*, nakon seckanja je moguće dobiti fragmente koji su kraći od neophodne dužine za skeniranje. U realnoj situaciji se takvi fragmenti skeniraju zajedno sa delom adaptera. U ovoj verziji simulacije, fragmenti koji nemaju dovoljnu dužinu za skeniranje su preskakani. Omogućavanje skeniranja i kratkih fragmenata uz dodavanje odgovarajućih adaptera bi poboljšalo kvalitet dobijenih rezultata.

5 Zaključak

Razumevanje delova DNK sekvence je važan deo bioloških istraživanja. Sekvenciranje genoma prevodi DNK određenog organizma u format razumljiv naučnicima. Određivanjem sekvence nukleotida, sekvenciranje pruža mogućnost boljeg razumevanja strukture genoma, prepoznavanje gena i njihove uloge u organizmu.

Postupak sekvenciranja genoma se, pored samog sekvenciranja, bavi i rekonstrukcijom genoma kao i "razumevanjem" dobijenih sekvenci. Rekonstrukcija genoma (assembly) se sastoji u spajanju velike sekvence od dobijenih fragmenata. Problem rekonstrukcije, kao takav, predstavlja veliki bioinformatički problem. Ideja u spajanju sekvenci je u potrazi za preklapanjem među malim sekvencama. Kako su u pitanju ogromne količine podataka, brzi algoritmi i jaki računari su svakako ključna stvar koja je potrebna da bi se ovaj korak izvršio.

Postupak sekvenciranja genoma je relativno skup i vremenski zahtevan. Kako bi se obezbedio dovoljan broj primera za testiranje programa za rekonstrukciju genoma, javlja se potreba za kompjuterskom simulacijom postupka sekvenciranja genoma. Zadatak ovog rada bila je izrada i optimizacija programa za simulaciju sekvenciranja genoma. Cilj je bio što vernije predstaviti savremeni postupak sekvenciranja. Svaki od koraka sekvenciranja pažljivo je simuliran i testiran. Kako bi se pružila mogućnost prilagođavanja rezultata, dodati su parametri koje je moguće podešavati kroz prost komandni interfejs. Sa dobro podešenim parametrima uspeli smo da dobijemo rezultate veoma slične realnim rezultatima sekvenciranja u relativno kratkom vremenskom intervalu što je bio dovoljan dokaz da je simulacija sekvenciranja uspešno implementirana.

6 Literatura

1. Dr Ana Simonović, Biotehnologija i genetsko inženjerstvo biljaka, NNK internacional, Beograd 2011

2. Internet izvor: *J. Craig Venter Institute, Genome sequencing*, http://www.genomenewsnetwork.org/resources/whats_a_genome/Chp2_1.shtml#chp2#2, poslednja izmena 15. januar 2003. godine

3. Richard C. Deonier, Simon Tavaré, Michael S. Waterman, Computational Genome Analysis, An Introduction, Springer, 1st ed. 2005.

4. Internet izvor: *Wellcome Trust Sanger Institute, Human Genome Project Approaches, Clone-by-clone sequencing, Whole genome shotgun sequencing* http://www.yourgenome.org/hgp/hgp2/hgp_5.shtml

5. Internet izvor: *DNA Sequencing - Maxam Gilbert Methods in DNA Sequencing* http://www.dnasequencing.org/maxam-gilbert

6. *Daniel Robert Zerbino, Genome assembly and comparison using de Bruijn graphs, PhD thesis, University of Cambridge,* Septembar 2009

7. Internet izvor: *The Discovery of the Molecular Structure of DNA - The Double Helix* http://www.nobelprize.org/educational/medicine/dna_double_helix/readmore.html poslednja izmena 20. maj 2012. godine

8. Internet izvor: *University of California Museum of Paleontology, Types of mutations* http://evolution.berkeley.edu/evolibrary/article/mutations_03, poslednja izmena 22. avgust 2008. godine

9. Internet izvor: *Help - About Nucleotide And Protein Sequence Formats* http://www.ebi.ac.uk/help/formats.html#fasta

10. Internet izvor: *FASTA format description* http://www.genebee.msu.su/blast/fasta.html

11. Internet izvor: Genom http://en.wikipedia.org/wiki/Genome, poslednja izmena 08. maj 2012. godine

12. Internet izvor*: Wellcome Trust, DNA sequencing - the Illumina method* http://www.wellcome.ac.uk/Education-resources/Teaching-and-education/Animations/DNA/WTX056051.htm

13. Internet izvor*: Illumina, Inc, Illumina Sequencing Technology*, http://www.illumina.com/documents/products/techspotlights/techspotlight_sequencing.pdf, poslednja izmena 11. oktobar 2010. godine

14. Internet izvor: *Kohji Okamura, John Wei* i *Stephen W Scherer*, *Evolutionary implications of inversions that have caused intra-strand parity in DNA* http://www.biomedcentral.com/1471-2164/8/160, poslednja izmena 11. jun 2007. godine

15. Internet izvor: *Leslie A. Pray, DNA Replication and Causes of Mutation* http://www.nature.com/scitable/topicpage/dna-replication-and-causes-of-mutation-409, poslednja izmena 2008. godine