SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Algoritmi za next-generation sequencing

Zvonimir Ilić

Voditelj: Doc. dr. sc. *Mile Šikić*

Zagreb, travanj 2011.

Sadržaj

1. Uvod

Jedna od najvažnijih znanstvenih otkrića 20. stoljeća jest otkriće molekule DNA. Od 1953. kada su ju otkrili James Watson i Francis Crick, razvoj medicine, molekularne biologije, biofizike i ostalih srodnih grana znanosti poprimio je sasvim nov smjer razvoja. Sama činjenica da upravo ova molekula određuje svojstva svake žive jedinke, bila je i više nego dovoljan poticaj za znanstvenike da ju istražuju. Zahvaljujući daljnjim istraživanjima otkrivena je važnost ove molekule i mogućnost primjene njenih svojstava u brojnim ljudskim djelatnostima.

Molekule DNA možemo shvatiti kao nositelje informacije. U njima je zapisana boja očiju, kose, visina itd. Znanstvenici su došli do zaključka da ako bismo mogli manipulirati tim informacijama, mogli bismo dobiti jedinke željenih osobina. Takvo razmišljanje je našlo svoju primjenu u agronomiji. Iz DNA se može pročitati i od kojih bi bolesti neki pojedinac mogao oboljeti u budućnosti. Ako bismo mogli maknuti te 'štetne' informacije i zamijeniti ih nekim povoljnijim, onda taj pojedinac nikad ne bi dobio tu bolest. Kod ljudi ovakva primjena otvara brojne moralne i etičke dileme. No u DNA je također zapisana i evolucijska povijest pojedine jedinke. Upravo zbog mutacija u DNA došlo je do diversifikacije pojedinih jedinki, a preživljavale su samo one najsnažnije i najotpornije što je jedna od temeljnih pretpostavki Darwinove teorije evolucije.

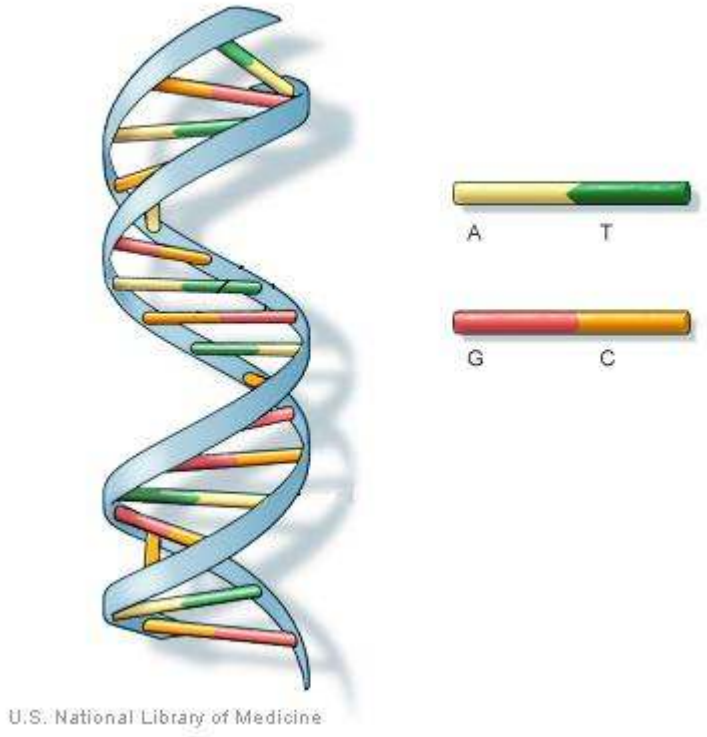
Kako su znanstvenici istraživali DNA, sve više je do izražaja počela dolaziti kompleksnost molekule i sam mehanizam ostvarivanja 'zapisanih' svojstava. Ubrzo je postalo jasno da će računala postati nezaobilazan alat u analizi DNA i obradi dobivenih informacija. Rezultat toga je nova znanstvena disciplina – bioinformatika. Neki od problema kojima se bavi ova disciplina su simuliranje strukture proteina, predviđanje pojedinih mutacija, razvoj boljih i bržih algoritama za analiziranje i usporedbu *stringova,* pronalaženje specifičnih struktura i nizova itd. Bioinformatika je od posebne koristi bila evolucijskim biolozima jer im je pomogla u praćenju mutacija DNA, usporedbi čitavih genoma i rekonstrukciji evolucijskog stabla.

Premda konceptualno zvuči jednostavno, analiza i proučavanje DNA nije nimalo lagan posao. Štoviše radi se o jednom od najvećih znanstvenih izazova današnjice. U ovom seminaru bavit ćemo se metodama sekvenciranja genoma, tj. kako 'pročitati' strukturu DNA. Valja naglasiti da ćemo problematiku sekvenciranja gledati iz perspektive algoritama i računala koja obrađuju dobivene informacije, dok će kemijski i biološki procesi biti u drugom planu.

2. DNA i njena svojstva

2.1. Struktura i kemijska svojstva

Kompleksna i neobična struktura DNA je često predstavljala prepreku za njeno istraživanje i zbunjivala znanstvenike, naročito u samim početcima.



Slika 1: Struktura DNA, izvor: *U. S. National Library of Medicine*

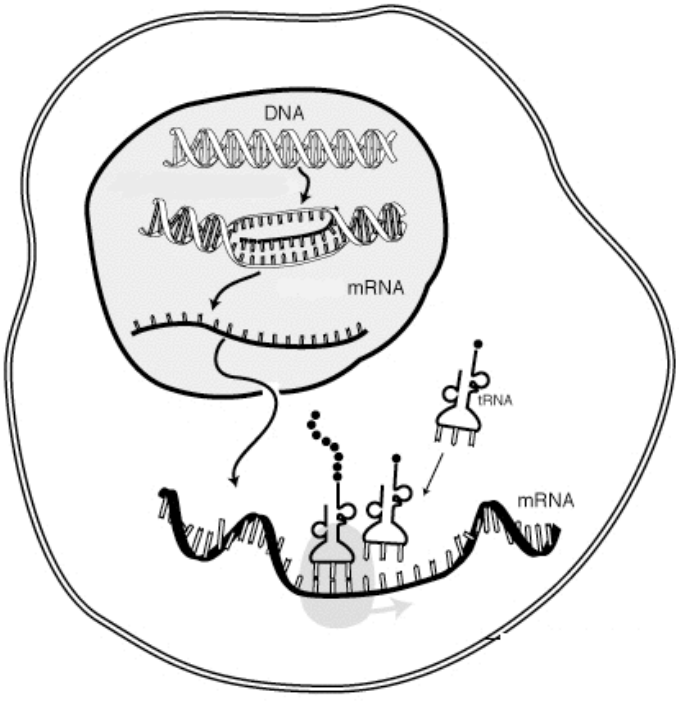
Deoksiribonukleinska kiselina je nasljedni materijal koji se nalazi u svakoj stanici pojedine jedinke. Većina DNA se nalazi u jezgri stanice, gdje je gusto kondenzirana u obliku kromosoma. Dio DNA se također nalazi u mitohondrijima. DNA je građena od osnovnih jedinica koje se zovu nukleotidi. Svaki nukleotid se sastoji od pentoze[1], fosfatne grupe i dušične baze. Postoje četiri vrste dušičnih baza: adenin, timin, gvanin i citozin. Nukleotidi su povezani fosfatnom grupom i šećerom naizmjence u lanac. DNA se sastoji od dva antiparalelna lanca povezana vodikovim vezama i ima oblik dvostruke zavojnice. Vodikovim vezama se povezuju dušične baze. Adenin i timin se povezuju dvostrukim, a citozin i gvanin trostrukim vezama i oni čine parove baza. To znači da se uz adenin u jednom lancu nalazi timin u drugom i obrnuto. Isto vrijedi i za citozin i gvanin. Važno svojstvo DNA je samoreplikacija. Za vrijeme mitoze DNA se raspakira iz kromosoma, lanci se razdvoje, i tada se pomoću enzima i dugog niza kemijskih reakcija nadopunjuju razdvojeni lanci tako da nastaju dvije kopije DNA. Gen koji određuje neko svojstvo, ustvari je niz parova baza, duljine od nekoliko tisuća do nekoliko milijuna. Npr. kod ljudi, koji imaju 23 para kromosoma, kromosm br. 1 ima oko 4200 gena i oko 250 milijuna parova baza.

[1] Šećer s pet atoma ugljika

2.2. Proteini i DNA

Proteini su osnovni građevni materijal svih živih jedinki. Stoga je jasno da će svojstva proteina odrediti i karakteristike pojedine jedinke. Kakvi će ti proteini biti, zapisano je upravo u DNA. DNA je kemijski nositelj informacije. Te informacije su zapisane u obliku redoslijeda dušičnih baza i na temelju njih se razvijaju točno određeni proteini. Na primjer, niz ATTGCCATA određuje jedan protein, dok niz GTCAACTGA neki drugi itd.

Proteini su biokemijski spojevi koji se sastoje od jednog ili više polipeptida. Polipeptidi su građeni od aminokiselina koje su povezane peptidnim vezama. Redoslijed aminokiselina u polipeptidu određuje svojstva proteina. Kojim redoslijedom će se nizati aminokiseline, određuje upravo DNA. Svi proteini u ljudskom tijelu građeni su od 20 različitih aminokiselina. Proces dekodiranja informacije iz DNA u redoslijed aminokiselina je vrlo dug i složen biokemijski proces, što premašuje opseg ovog seminara, pa ćemo samo načelno opisati cijeli proces. Prva faza je faza transkripcije. Kada je potreban određen protein, na tom mjestu se raspetlja dvostruka zavojnica DNA, i enzim RNA-polimeraza gradi mRNA[2]. Time je iz DNA napravljen komplementarna mRNA. Druga faza je faza translacije gdje se iz izgrađene mRNA na ribosomima sintetiziraju proteini. Tu dolazi tRNA[3] koja se s jedne strane veže za tri nukleotida mRNA, a s druge za jednu aminokiselinu. To znači da tri nukleotida u DNA, odnosno mRNA, određuju jednu aminokiselinu. Očito je da 3 nukleotida mogu dekodirati 4^3=64 aminokiseline. Pošto mi imamo na raspolaganju samo 20 aminokiselina, ova redundancija služi kao neka vrsta zaštitnog mehanizma.



Slika 2: Replikacija proteina određena nizom m-RNA, 3 baze na m-RNA određuju jednu aminokiselinu, a niz aminokiselina čini protein, izvor: *Wikipedia*

[2] mRNA-*messenger* RNA; RNA je ribonukleinska kiselina, građena od nukleotida koji mogu imati 4 vrste dušičnih baza: adenin, gvanin, citozin i uracil. RNA ima samo jedan lanac, za razliku od DNA.
[3] tRNA-*transfer* RNA

3. Sekvenciranje DNA

Sekvenciranje DNA je skup postupaka i tehnologija kojima se određuje niz dušičnih baza (A, C, G, T) u molekuli DNA. Pošto taj niz određuje upute za razvoj i funkcioniranje žive jedinke, sekvenciranje je postalo nužan postupak u biotehnologiji, forenzici, medicini itd… Sekvenciranje je tehnički vrlo zahtjevan proces, naročito kod organizama s velikim brojem parova baza i gena. Npr. ljudski genom ima oko 2.9 milijardi parova baza i 30000 gena.

Postoje dva osnovna pristupa sekvenciranju: *Whole-genome Sequencing*(WGS) i hijerarhijski pristup[1],[5]. Kod WGS-a odjednom pokušamo sekvencirati cijelu molekulu DNA, a kod hijerarhijskog molekulu DNA (koja ima nekoliko milijuna parova baza) prvo razlomimo na fragmente duljine nekoliko desetaka tisuća parova baza.

3.1. Sekvenciranje prekidajućim radikalima

Ovu metodu je razvio Frederick Sanger 1975. i njena prednost je što ne koristi previše kemikalija koje kontaminiraju uzorak ili radioaktivne izotope[5]. Princip te metode je da nakon što se umnoži malen uzorak jednolančane DNA (nekoliko 100 parova baza), pomiješamo ga s enzimom koji gradi komplementarni lanac DNA (DNA-polimeraza), zalihom nukleotida i tzv. prekidajućim radikalima koji prekidaju daljnju izgradnju lanca kad naiđu na određeni nukleotid. Nakon toga uzorke razdvojimo po veličini elektroforezom i očitamo slijed nukleotida. Time smo dobili fragment duljine od nekoliko stotina parova baza (čiji nam je redoslijed poznat).

3.2. Next-generation sequencing

NGS se bazira na takozvanom *shotgun* sekvenciranju[4]. Osnovna ideja ovakvog sekvenciranja je:

1. Umnožiti uzorak DNA nekoliko puta (umetanjem u kružne molekule DNA kod bakterija) i razbiti ih na fragmente (pomoću EM zračenja ili visokog pritiska)
2. Sekvencirati razbijene fragmente duljine nekoliko stotina baza
3. Dobivene fragmente spojiti računalnim algoritmom (najčešće na temelju preklapanja)

Uspješnost ove metode je upravo u kratkim fragmentima, čija duljina iznosi oko 600 do 700 parova baza. Naime pokazalo se da je sekvenciranje fragmenata duljih od 1000 parova običnim kemijskim postupcima baza neprecizno i neučinkovito. Također je utvrđeno da preciznost opada što su fragmenti dulji. NGS je ovakvim inovativnim pristupom omogućio sekvenciranje čitavog genoma pojedinih organizama. Ovakav pristup sekvenciranju je puno jeftiniji i djelotvorniji od klasičnog. Samim time je NGS korišten u komercijalne, industrijske i znanstvene svrhe.

4. **Algoritmi za spajanje u NGS-u**

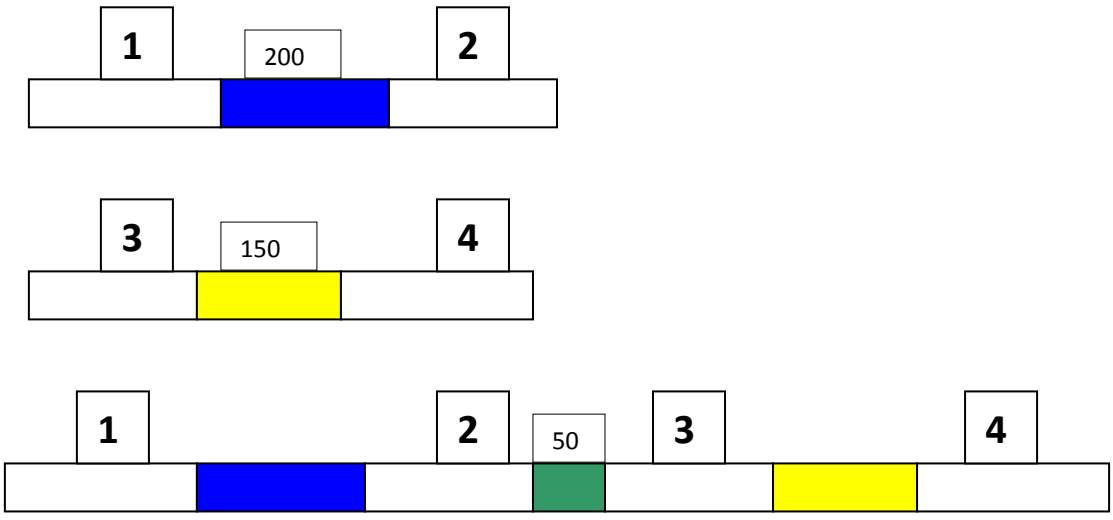
Nakon što smo dug niz nukleotida razbili na puno manjih, radi lakšeg čitanja, sada je potrebno te manje fragmente ponovno sve povezati da bismo otkrili kako je izgledao originalni niz nukleotida. Ovi algoritmi moraju imati i povoljnu vremensku i memorijsku složenost jer valja imati na umu da se radi o tisućama fragmenata. Također, postoje i mnoge poteškoće koje otežavaju rad algoritma i koje mogu dovesti do nepravilnog spajanja, o čemu će također biti riječ.

4.1. Općenit algoritam

Niz nukleotida, odnosno dušičnih baza, zapisan je u obliku znakovnog niza. Pretpostavimo da imamo skup fragmenata $T=(s_1,s_2,\ldots,s_n)$. Pretpostavlja se da ćemo dobiti originalni niz nukleotida ako pronađemo najkraći niz koji u sebi ima sadržane sve elemente iz T. To znači da je svaki element iz T podniz novonastalog niz nukleotida. U samim počecima razvoja algoritama, ovaj je relativno dobro opisivao problem spajanja. No očito je da ovaj algoritam ne mora dati točan rezultat pa su se morala pronaći nova rješenja.

4.1.1. Pohlepni algoritam

Osnovna zamisao svih pohlepnih algoritama je da odaberemo rješenje koje je trenutno najpovoljnije, tj. da pri rješavanju nekog problema ne razvijamo nikakvu dugoročnu strategiju. Pogledajmo sljedeću sliku:



Slika 3: Pohlepni algoritam

Imamo četiri fragmenta. Promotrimo međusobna preklapanja između fragmenata (obojeni pravokutnici). Pod preklapanjem[4] se smatra podudarnost niza nukleotida na završetku jednog fragmenta i početku drugog fragmenta. Na primjer niz ATGC i TGCC

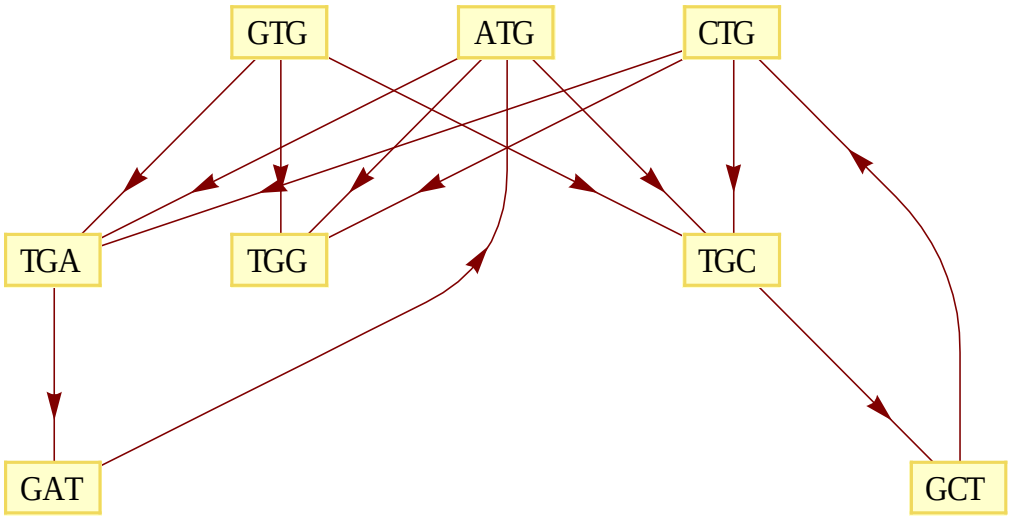
[4] Pošto originalnu DNA razbijemo na puno fragmenata, zbog redundancije određene dijelove DNA 'pokriva' više fragmenata i samim time dolazi do preklapanja.

preklapaju se u tri nukleotida i to u TGC. Prvi i drugi fragment na slici se preklapaju u 200 nukleotida, drugi i treći u 50, a treći i četvrti u 150. Strategija je da uvijek spajamo one fragmente koji se trenutno preklapaju u najviše zajedničkih nukleotida. U prvom koraku fragmenti 1 i 2 se trenutno preklapaju u najviše nukleotida, točnije njih 200, pa ih odmah spojimo. Zatim spojimo treći i četvrti, a potom drugi i treći prema navedenom pravilu. Rezultat algoritma je jedan veliki niz koji sadrži sve fragmente s početka. Kao i kod svih pohlepnih algoritama i kod ovog je najveća slabost nepostojanja dugoročnih strategija i samim time krajnji rezultat može biti pogrešan. Statistika je pokazala da, ako želimo dobiti povoljan rezultat ovom tehnikom, moramo moći svim fragmentima pokriti cijelu molekulu DNA barem osam puta. Ako sekvenciramo uzorak od dva milijuna parova baza, onda ukupna duljina svih fragmenata mora biti 16 milijuna parova baza, odnosno moramo imati oko 27000 fragmenata prosječne duljine oko 600 parova baza[1].

4.2. Algoritmi bazirani na teoriji grafova

4.2.1. De Brujinovi grafovi

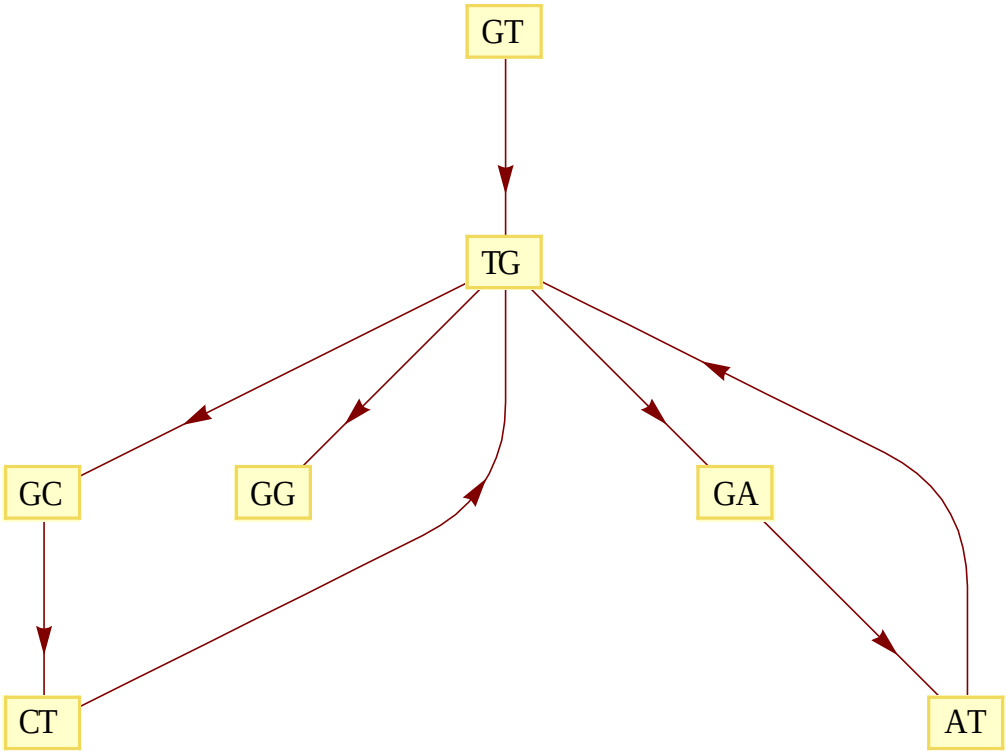
Teorija grafova je izrazito moćan alat u računarstvu, pa samim time i u bioinformatici. U sekvenciranju se koristi kao metoda koja olakšava spajanje fragmenata. Svaki fragment možemo interpretirati kao interval na ciljnom nizu, odnosno redoslijedu nukleotida koji moramo dobiti. Od tih fragmenata možemo sastaviti usmjereni graf na temelju preklapanja. Ovdje će nam koristiti De Brujinov graf[3]. De Brujinov graf se konstruira na temelju preklapanja, tj. završetak jednog fragmenta istovjetan je početku drugog. Ako fragmente interpretiramo kao čvorove, a usmjereni bridovi povezuju čvorove koji zadovoljavaju uvjet preklapanja, tada možemo konstruirati graf. Sljedeći primjer nam pokazuje kako je to moguće. Pretpostavimo da imamo skup fragmenata {GTG,TGA,GAT,ATG,TGC,GCT,CTG,TGG}. Usmjereni graf izgleda ovako za zadani skup.



Slika 4: De Brujinov graf na kojem treba pronaći Hamiltonovski ciklus

Rješenje problema, tj. niz čiji su ovo fragmenti, dobit ćemo tako da pronađemo Hamiltonov[5] ciklus ili skoro-hamiltonovsku stazu. Na ovom grafu to je prilično jednostavno i dobije se da je originalni niz GTGATGCTGG.

Također, moguće je imati i malo drugačiji pristup u uporabi teorije grafova. Uzmimo sve fragmente duljine *n.* Svaku *n*-torku možemo rastaviti na njen prefiks i sufiks. Npr. niz ATGGCA ima prefiks ATGGC i sufiks TGGCA. Ako prefiks stavimo u jedan, a sufiks u drugi čvor, brid koji povezuje ta dva čvora ustvari predstavlja originalnu *n*-torku. Originalni niz ćemo u ovom slučaju rekonstruirati pronalaskom Eulerovog[6] ciklusa ili skoro-eulerovske[7] staze. Na sljedećem primjeru možemo vidjeti kako je moguće rekonstruirati originalni niz.



Slika 5: De Brujinov graf na kojem treba pronaći Eulerov ciklus

Ovaj graf spada u klasu skoro-eulerovskih grafova što znači da, iako nemamo ciklus, ipak možemo pronaći stazu kojom ćemo obići sve bridove. Na ovom grafu možemo pronaći čak dvije skoro-eulerovske staze. Prva staza slijedi redom ove čvorove: GT, TG, GC, CT, TG, GA, AT, TG, GG. Time se dobije slijed nukleotida GTGCTGATGG. Druga staza obilazi čvorove sljedećim redoslijedom: GT, TG, GA, AT, TG, GC, CT, TG, GG. Ovaj obilazak daje slijed GTGATGCTGG.

[5] Ciklus u kojem se iz početnog čvora šetnjom po grafu vrati u taj isti, pri čemu se obiđu svi ostali čvorovi točno jednom.
[6] Ciklus u kojem se iz početnog čvora šetnjom po grafu vrati u taj isti, a da se pritom obiđu svi bridovi točno jednom.
[7] Staza u kojoj obiđemo sve bridove točno jedanput, a ne vratimo se u početni čvor.

4.2.2. Primjena grafova u modernim asemblerima

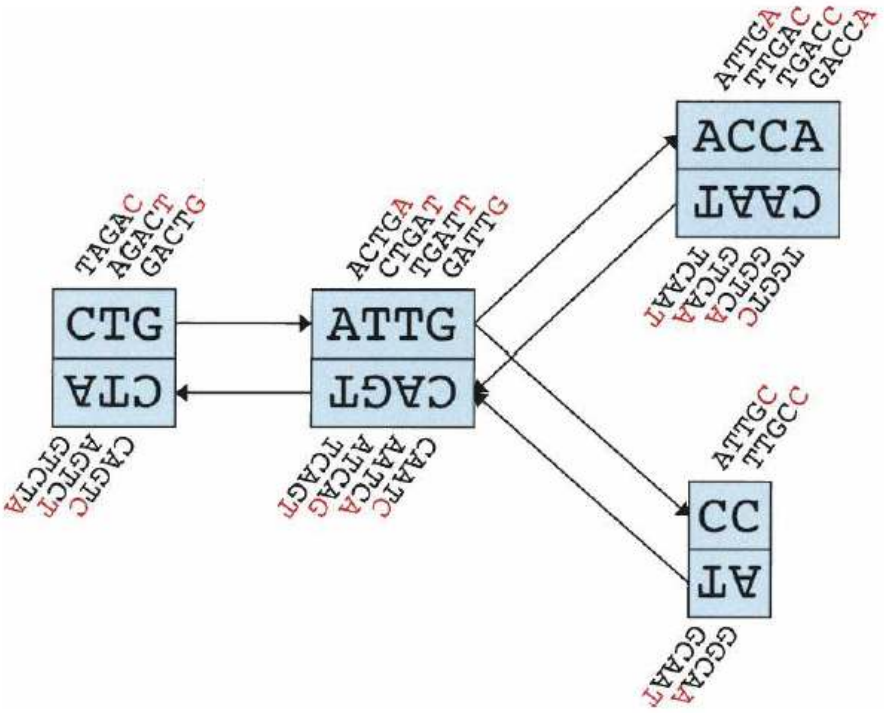
Teorija grafova olakšava povezivanje fragmenata na temelju preklapanja, no ona nije dovoljna da bi se dobio zadovoljavajući rezultat. Većina modernih asemblera koristi grafove u svojem radu, no valja imati na umu da asembler prima ogromnu količinu fragmenata različite duljine koje treba obraditi. Samim tim mora se postići povoljna vremenska i memorijska složenost. Svaki asembler[3] također mora imati i algoritme za otkrivanje određenih grešaka i ponavljanja, o čemu će biti riječ u sljedećem poglavlju. Asembler mora imati i ugrađene algoritme optimizacije grafa da bi spajanje bilo izvedeno što brže i sa što manje grešaka.

5. Asembler *Velvet*

Ovaj asembler[2] razvili su Daniel Zerbino i Ewan Birney na Europskom bioinformatičkom institutu. On uključuje niz algoritama koji spajaju fragmente koristeći De Brujinove grafove i otklanjaju greške. U ovom poglavlju ukratko ćemo opisati njegov rad.

5.1. Struktura i konstrukcija grafa

Velvet niz od *n* nukleotida definira kao n-torku[2] i na njima se bazira konstrukcija grafa. Za svaku pronađenu n-torku u skupu fragmenata, u tablicu se pohranjuju ID fragmenta i pozicija n-troke u fragmentu koji ju sadrži. Uz svaku n-torku pohranjuje se i njezin obrnuti komplement. Svaki obrnuti komplement ne smije biti jednak originalu pa je *n* neparan čime se izbjegava simetrija. Time možemo svaki fragment zapisati kao niz n-torki koje se preklapaju. Takve n-torke stavljamo u isti čvor. Nakon toga se gledaju preklapanja n-torki, ali takva da ne pripadaju istom fragmentu i da je završetak jedne ujedno i početak druge. Takva preklapanja se povezuju bridom. Na sljedećoj slici vidi se konstruiran graf. Sva preklapanja koja pripadaju istom fragmentu su u istom čvoru.



Slika 6: Pohrana fragmenata u *Velvetu*, izvor:[2]

Na slici vidimo da su u čvorovima petorke koje se preklapaju u 4 nukleotida, a oznaka čvora su zadnji nukleotidi. Na svaki čvor su nadovezani obrnuti komplementi n-torki. Valja primijetiti da što je n manji, to je veća vjerojatnost preklapanja pa i graf ima više bridova.

5.2. Pojednostavljenje grafa

Pojednostavljenje u *Velvetu* slično je nadovezivanju dva *stringa*. Algoritam je vrlo jednostavan i očit. Ako čvor A ima jedan izlazni brid prema čvoru B, a čvor B ima samo jedan ulazni brid (i to upravo ovaj), onda možemo spojiti ta dva čvora. Drugim riječima, možemo napraviti uniju ta dva skupa n-torki. Ovakvo pojednostavljenje smanjuje korištenje memorije i skraćuje vrijeme spajanja.

5.3. Otklanjanje grešaka

U procesu sekvenciranja može nastati više grešaka, bilo pri konstrukciji grafa, bilo zbog pogrešnog očitanja nukleotida itd. Neke se ispravljaju odmah nakon konstrukcije grafa, a neke tek kad je dovršen cijeli postupak sklapanja. Ovdje će biti navedene samo neke najvažnije.

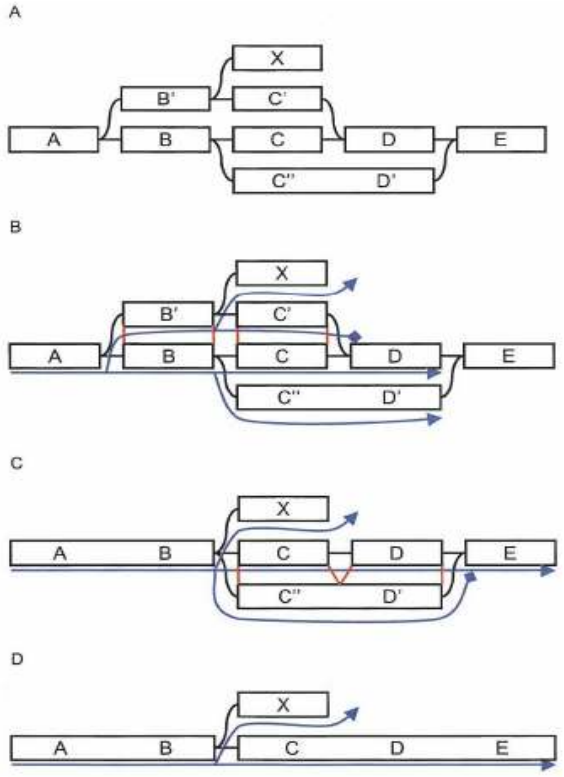
5.3.1. Uklanjanje grana

Grana je niz čvorova koji su spojeni s grafom samo sa svojim jednim krajem, dok drugi kraj 'visi'. Drugim riječima to je podskup grafa koji nije dio nijednog ciklusa u grafu. Pošto pri spajanju moramo obići sve čvorove, očito je da će grane ometati pronalazak ciklusa. *Velvet* koristi dva kriterija kojima odlučuje što će napraviti s nekom granom[2]. Prvi kriterij se odnosi na duljinu niza nukleotida koje opisuje grana. Ako je duljina niza manja od 2*n*, gdje je *n* duljina n-torki, onda tu granu uklonimo iz grafa. Drugi kriterij kaže da ako postoji alternativni put kojim ćemo obići veći broj čvorova, tj. da ne moramo ići u granu, maknemo ju. Oba kriterija primjenjuju se tako da ne dođe do raspada grafa.

5.3.2. Uklanjanje mjehurića *Tour Bus* algoritmom [2]

Mjehurić nastaje kada imamo dva puta, odnosno niza čvorova i bridova koji počinju i završavaju u istim čvorovima. Očito je da ta dva puta daju različite nizove nukleotida, stoga je potrebno i taj problem riješiti. U *Velvetu* je taj problem riješen implementacijom *Tour Bus* algoritma. Za detekciju višestrukih putova koristi se BFS[8]. Asembler označava čvorove koje je posjetio. Kada ponovno dođe do čvora koji je već posjetio, asembler se vraća od trenutnog i prethodno posjećenog čvora da pronađe najbliži zajednički čvor, tj. da nađe gdje višestruke staze počinju. Nakon toga nad nizovima nukleotida, koji su opisani višestrukim stazama, provodi se poravnanje. Ako su nizovi dovoljno slični, spoji ih se. Spajanje je vrlo složeno jer se ne smije narušiti struktura ostatka grafa. Na sljedećoj slici shematski je prikazan rad algoritma. Mjehurić čine staze ABCD i AB'C'D. Pošto smo D već posjetili, nad nizovima BC i B'C' se provodi poravnanje. Zaključeno je da dovoljno sliče pa se spoje. Valja primijetiti da i nakon spajanja veze u ostatku grafa nisu narušene.

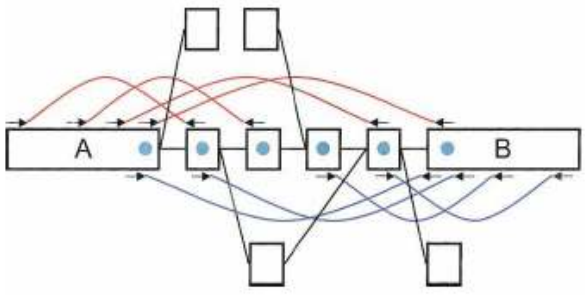
[8] Breadth-first search



Slika 7: *Tour Bus* algoritam, izvor:[2]

5.4. *Breadcrumb* algoritam

Ovaj algoritam služi redukciji De Brujinovog grafa. Ovaj algoritam također pomaže i pri otklanjanju višestrukih ponavljanja sekvenci jer u genomu postoje sekvence koje se višestruko ponavljaju. Ovaj algoritam nastoji povezati duže cjeline (A i B) kraćim fragmentima, tzv. PET-ovima[9]. PET-ovi su parovi kratkih nizova DNA koji se nalaze na krajevima neke duže cjeline i koji se nalaze na jedinstvenom mjestu u molekuli DNA. Ako znamo početak i kraj fragmenta, te međusobnu udaljenost i položaj PET-ova, samim time je lakše detektirati ponavljanja sekvenci[10].



Slika 8: *Breadcrumb* algoritam, izvor:[2]

Na slici vidimo da algoritam nastoji duže cjeline A i B povezati na temelju PET-ova koji pripadaju i jednoj i drugoj cjelini.

[9] Paired-end Tag

[10] Ponavaljanje sekvenci je vrlo ozbiljan problem u modernom sekvenciranju. Ponavljanja mogu prouzročiti krivo spajanje čitave molekule DNA. Npr. u ljudskom genomu postoje nizovi od 100000 nukleotida koji se ponavljaju. Ovo je vrlo opširna tema i samo daljnje izlaganje bi nadišlo opseg ovog seminara.

6. Zaključak

Sekvenciranje DNA je jedan od najvećih izazova moderne znanosti. U ovom seminaru izloženo je nekoliko konceptualnih pristupa sekvenciranju čitavog genoma nekog organizma jer ukoliko poznajemo genom neke jedinke, tada možemo odrediti i njen razvoj i njene biološke karakteristike. No otkrivanje slijeda nukleotida u molekuli DNA nije nimalo jednostavna zadaća. Kao što je rečeno, modernim shotgun sekvenciranjem moguće je odrediti niz nukleotida u fragmentu duljine od nekoliko stotina parova baza. Poznavanje redoslijeda baza nekog fragmenta ne znači ništa ako ne znamo kojem genu pripada taj fragment, s kojeg je kromosoma, koju biološku funkciju regulira itd… Tu nam pomažu računala. Računalni algoritmi za spajanje i optimizaciju pomažu nam da sve razbijene fragmente povežemo i dobijemo cjelovitu sliku molekule DNA. Zbog ogromne duljine molekule DNA bioinformatičari su morali smisliti algoritme optimizacije koji će ogroman broj fragmenata obraditi u zadovoljavajućem vremenu. U proučavanju assemblera Velvet vidjeli smo da puka primjena algoritama nije dovoljna i da računalni algoritmi moraju uzeti u obzir brojne karakteristike DNA koje značajno otežavaju proces spajanja. Na području Next-generation sequencing-a i pripadnih algoritama računalni inženjeri i znanstvenici će morati uložiti još puno napora da cijeli proces sekvenciranja učine bržim i učinkovitijim. Ovaj seminar je pokrio samo jedan malen dio ovog vrlo izazovnog, zahtjevnog, ali i zanimljivog područja.

7. Literatura

[1] Mihai Pop, Steven L. Salzberg, Martin Shumway: *Genome Sequence Assembly: Algorithms and Issues*, The Institute for Genomic Research, 2002.

[2] Daniel R. Zerbino, Ewan Birney: *Velvet: Algorithms for de novo short read assembly using De Brujin graphs,* Genome Res., 2008.

[3] Jason R. Miller, Sergey Koren, Granger Sutton: *Assembly algorithms for next-generation sequencing dana*, J. Craig Venter Institute, 2009.

[4] *Shotgun sequnecing ,* http://en.wikipedia.org/wiki/Shotgun_sequencing, 13. 4. 2011.

[5] *DNA sequencing*, http://en.wikipedia.org/wiki/DNA_sequencing, 12. 4. 2011.

[6] Philipp Bücher, Bernard M. E. Moret: *Algorithms in bioinformatics*, WABI, 2006.

[7] Jan Kieleczawa: DNA sequencing: optimizing the process and analysis, Volume 1, 2005.

8. Sažetak

Pri sekvenciranju genoma nekog organizma postoje dva temeljna pristupa: WGS i hijerarhijski pristup. U modernom sekvenciranju DNA potrebno je koristiti *shotgun* metode kojima se molekula DNA razbija na fragmente. Takve fragmente lakše sekvenciramo i zatim ih obrađujemo računalnim algoritmima. U obradi se koristi teorija grafova i pohlepni algoritmi. Svim algoritmima je zajedničko da traže preklapanja među fragmentima. Od velike korisiti su De Brujinovi grafovi na kojima se traže Hamiltonovi i Eulerovi ciklusi. Na De Brujinovim grafovima bazira se većina modernih assemblera. Moderni programi koriste brojne optimizacije grafova kako bi što djelotvornije spojili brojne fragmente.