Code : BFVH3TBIO	**Tentamen:** Proeftentamen Theorie van Bioinformatica 2 **NIET VERTROUWELIJK**	
Datum: 29-10-2018	**Tijd:** 10:30-12:30	**School:** ILST
Lokaal:	**Klas:** BFV3	**Duur:** 1 1/2 uur
Docent : Martijn Herber **Het secretariaat is tijdens het tentamen te bereiken onder nummer: 050 – 595 45 69.**		**Aantal pagina's:** 2
Hulpmiddelen: Kladpapier	**Overig hulpmiddelen:** Geen	
Opgave inleveren: Ja **Kladpapier inleveren:** Ja		
Bijzonderheden: Bijzonderheden _____		
Naam student: **Klas:**		**Studentnummer:**

Bij het beantwoorden van de vragen, schrijf duidelijk, houd het kort, **maar wees altijd volledig**.

1)Leg uit wat de “molecular clock hypothesis is”, hoe zich dit relateert aan afstanden uit een alignment en waarom we hierop nog een correctie moeten uitvoeren om een “eerlijk” beeld te krijgen van de verstreken tijd tussen twee sequenties die van een gemeenschappelijke voorouder afstammen. (10pt)
De “molecular clock hypothesis” zegt dat elke functionele familie van eiwitten een eigen, constante snelheid heeft van mutaties per tijdseenheid (oorspronkelijk onderzocht voor Aminozuren; maar gaat ook op voor de nucleotiden van de betrokken genen). De mutaties kun je aflezen uit een MSA en dit moet geijkt worden met een externe bron van tijd. Deze zijn echter niet helemaal correct omdat er ook mutaties geweest zijn die je niet meer terugziet; “hidden” door mutaties die weer terug naar een eerdere status zijn gegaan. Daarom worden allerlei correctiemethoden toegepast uit de afstanden uit het MSA om de “werkelijke” aantal mutaties per tijdseenheid te berekenen. [Pevsner p.250, 272-277]

2)Beschrijf het UPGMA algoritme om fylogenetische bomen te construeren. Gebruik tenminste de concepten clade, OTU, branch length, en rooted/unrooted. (12pt)
Het UPGMA algoritme werkt op basis van een afstandstabel die uit een Multiple Sequence Alignment komt. Alle OTU’s (Operational Taxonomic Units; de input sequenties die je gebruikt) worden eerst in zo’n MSA met elkaar aligned. Vervolgens begint het UPGMA algoritme met het zoeken van de twee OTU’s die volgens de afstandstabel het dichtst bij elkaar liggen. Deze worden samen in een “clade” geplaatst, wat leidt tot het maken van een eerste interne node, hun gemeenschappelijke voorouder. De branch length tussen die OTU’s en hun voorouder is het gemiddelde van de afstand tussen de OTU’s, en met die informatie kun je ook de afstand van de nieuwe interne node tot alle andere OTU’s in de afstandtable updaten. Nu wordt opnieuw bepaald wat de korste afstand in de tabel is (tussen de OTU’s en de nieuwe clade) en wordt het proces herhaald. Dit gaat door totdat alle OTU’s/Clades opgebruikt zijn in de tabel en je een laatste voorouder hebt toegevoegd; de root. [Pevsner p.284]

3)Leg uit wanneer je een unrooted phylogenetic tree zou gebruiken, en wanneer een rooted. Welke aanname maak je bij het gebruik van een rooted tree? (10pt)
Je gebruikt een unrooted tree als je een stel eiwitten of genen hebt waarvan je de onderlinge afstanden en organisatie wilt weten. Unrooted trees geven namelijk de meest accurate afstanden weer. Als je echter geinteresseerd bent in fylogenie, dwz. De afstamming van soorten op basis van een aantal genen/eiwitten, dan moet je een rooted tree gebruiken. Deze geeft je dan ook een Last Common Ancestor; de gemeenschappelijke voorouder. Echter, je neemt dan wel aan dat die er is; dit moet je om onafhankelijke redenen al zeker weten, want het algoritme tekent vrolijk altijd een voorouder, of die er nou echt was of niet.

4)a) Leg uit waarom je bij de Bruijn graph assembly altijd meerder kmer-sizes moet uitproberen, om de “beste” assembly te verkrijgen. (5pt)

Je moet meerdere k-mer sizes gebruiken omdat je van te voren niet weet wat de optimale grootte is om de "beste" contigs te genereren. De k-mer size bepaalt hoeveel nodes en paths tussen die nodes worden gemaakt in het DBG; te grote kmer-sizes leveren weinig paden op en dan mis je genoeg paden tussen de nodes om grote contigs te maken. Te kleine k-mer sizes levert te veel mogelijke paden tussen nodes op en dan wordt het ook lastig eenduidige contigs te maken (te veel alternatieve paden levert misassembly, niet bestaand contigs, op). Dus meestal probeer je een heel aantal k-mer sizes uit. [Presentatie; Pevsner p.398 fig.9.11]

b) Wat is de meest gebruikte maat om de kwaliteit van assembly te meten, en wat betekent deze? (5pt)

De meest gebruikte maat om de kwaliteit van assembly te meten is de "N50"; de gewogen mediaan van de contig-grootte waarin 50% van de nucleotiden in de assembly te vinden is. Dus een N50 van 10,000 betekent dat 50% van de nucleotiden uit je dataset in contigs te vinden is van 10Kb of meer. [Pevsner p.395]

5) Wat wordt opgeslagen in het Variant Call Format? Hoe heet de stap in de pipeline die deze files produceert? Waarom is het belangrijk voor de stap in je analysepijplijn die deze files produceert om genoeg *coverage* van het genoom te hebben? (6pt)

Het Variant Call Format bevat voor elke gemapte read uit een NGS fastq file de locatie van een variatie tussen de read en het referentie genoom (index) waartegen gemapt is. Meerdere reads met dezelfde mutatie worden gebundeld. De "Pileup" stap uit de samtools pipeline genereert deze files. Er is genoeg coverage nodig zodat de software kan bepalen of in de pileup een verschil tussen de reads en referentiegenoom komt door een technische (sequencing) fout of een echte mutatie tov. Het referentiegenoom is. [Pevsner p.411, p.405]

5)Beschrijf de algemene onderdelen van een *graaf* of wiskundig *netwerk*. Hoe worden deze toegepast in assembly bij een Overlap Layout Consensus methode ?(10pt)

Een wiskundig netwerk of graaf bestaat uit knopen en paden tussen die knopen (nodes en edges). Deze datastructuur wordt in het geval van OLC assembly gevuld met reads, en de overlap tussen die reads (bepaald dmv alignment). De read zelf wordt in een node geplaatst en de edges naar andere nodes worden gemaakt als er overlap is tussen de reads in die nodes. In het geval van OLC wordt er per edge ook een gewicht bijgehouden die evenredig is aan de lengte van de overlap tussen die twee reads. Bij het aflezen van de contigs (zie vr. 9) worden dan de edges met het meeste gewicht gevolgd. [presentatie assembly en Pevsner p.398]

6)Beschrijf hoe paired-end sequencing je kan helpen om je assembly van losse contigs verder af te maken. Beschrijf twee lab-methoden waarmee je uiteindelijk kunt proberen om het genoom helemaal af te maken. (12pt)

Paired-end sequencing betekent dat er fragmenten DNA van een bekende, precieze grootte worden gemaakt bij de library prep. Bij het sequencen wordt er nog steeds maar 100-150bp van weerszijden van het

fragment gelezen, maar er wordt bijgehouden welke twee reads bijelkaar horen ; ze zijn met een bepaalde afstand van elkaar bekend (de insert of bridge size). Tijdens het assembleren krijg je in eerste instantie een best-effort lijst met contigs die van elkaar gescheiden worden door repeat-regio's die niet eenduidig geassembleerd kunnen worden. Je weet dus ook niet de orientatie of zelfs de afstand tussen de contigs die geassembleerd zijn. Om dit op te lossen kun je een of meerdere libraries met paired-end reads of mate-pair reads (voor langere afstanden) gebruiken. De assembler probeert dan of die reads die op een contig geplaatst kunnen worden een mate pair hebben die op een ander contig past. Als dat zo is, dan weet je de orientatie en de (maximale) afstand tussen die contigs. In de praktijk gebeurt dit automatisch tijdens het assembleren en krijg jij alleen de lijst met super-contigs (scaffolds) die zo gemaakt zijn te zien. Om de dan nog losliggende scaffolds met elkaar te verbinden om het genoom "af te maken" (closing the genome) kun je twee labtechnieken gebruiken. De eerste, ouderwetse methode is om PCR primers te ontwerpen op de uiteinden van de scaffolds, die "naar buiten wijzen" (dwz van het scaffold af PCRen). Dan wordt er een grote combinatie PCR gedaan met steeds paarsgewijs alle combinaties van de ontworpen primers van alle scaffolds.Als er een PCR product tevoorschijnkomt bij een combi van primers, betekent dat dat de betrokken scaffolds binnen bepaalde afstand en orientatie van elkaar liggen. De tweede, nieuwere methode is een optimal mapping strategie. Hierbij wordt het DNA van je sample zowel in het lab als in-silico geknipt met een zeldzaam restrictie enzym. Dit levert een karakteristiek bandenpatroon op zowel in het lab als in de computer, en als je deze dan vergelijkt kun je ook de afstanden en de orientatie van de scaffolds tot elkaar afleiden. (Als de restrictie-sites vaak en uniek genoeg voorkomen)[presentatie Assembly]

7) Stel, je hebt RNAseq data van een organisme waar nog geen volledig genoom van bekend is. Hoe ga je te werk om toch iets over de genexpressie te zeggen? (10pt)
Je kunt dit oplossen door een zgn. "partial assembly" te doen. Op basis van je RNAseq reads assembleer je contigs die op z'n best overeenkomen met de volledige mRNA's uit je sequencingdata. Hiervan maak je dan een index, en vervolgens map je dezelfde reads weer terug op dat nieuw gemaakte index. Dit kun je doen omdat je toch alleen maar geinteresseerd bent in hoeveel reads er per transcript vandaan komen (de read count). Zwakte van deze aanpak is wel dat je de contigs alsnog moet annoteren (anders weet je niet wat ze doen; toch essentieel voor verdere expressieanalyse).

8) Wat wil de term "Eulerian path door een netwerk" zeggen en hoe wordt zo'n netwerk gebruikt om bij assembly "contigs uit te lezen" ? (8pt)
Een Eulerian path door een netwerk is een zo groot mogelijke wandeling van node naar node die alle tussenliggende edges precies eenmaal langsloopt. Omdat bij assemblysoftware de overlap tussen de nodes (met reads voor OLC en kmers voor DBG) in de edges staat, kun je door al die overlaps bijelkaar op te tellen de contigs uit lezen. Als je niet eenduidig je pad door het netwerk kunt vervolgen (vanwege repeats) of je loopt vast op een punt in het netwerk, dan is dat het einde van de contig die je aan het lezen bent. [presentatie Assembly]

9)Wat is het verschil tussen “positieve” en “negatieve” selectie en hoe kun je dit bepalen aan de hand van de “synonymous” en “non-synonymous” substitution rates? (12pt)

Deze begrippen staan eigenlijk los van elkaar. Positieve selectie en negatieve selectie van genen zeggen iets over de overlevingskansen van een organisme als er iets in die genen verandert (muteert). Bij een positieve selectie betekent verandering van gensequentie en de bijbehorende eiwitten dat het organisme beter overleeft (én nakomelingen produceert) en bij negatieve selectie is dit andersom. Dit is echter vaak heel moeilijk zomaar te bepalen op basis van biologische experimenten, dus als je de sequenties hebt van de genen en die van hun voorouders/relaties dan kun je met bioinformatica iets zeggen over deze selectieve druk. Synomymous mutaties in een gen willen zeggen dat ze de aminozuurvolgorde van het bijbehorende eiwit niet veranderen; bij non-synonymous variaties is dit wel zo. Als we dus een gen vinden waarbij de synonieme mutaties lager in aantal (rate) zijn dan non-synonieme, dan vermoeden we dat het gen onder negatieve selectie staat; verandering van het eiwit leidt tot slechtere overleving. Als het aantal non-synonymous mutaties groter is dan de synonymous, dan is er waarschijnlijk sprake van positieve selectie. [Pevsner p.257]