Bij deze opdracht maak je zelf een pipeline, die bestaat uit een combinatie van zelf geschreven scripts en bestaande applicaties. Bij het maken van de pipeline gebruik je de methodologie ‘majops’ (organized scripting) zoals deze beschreven is op de website <http://majops.weebly.com> van Martijn Jansen.

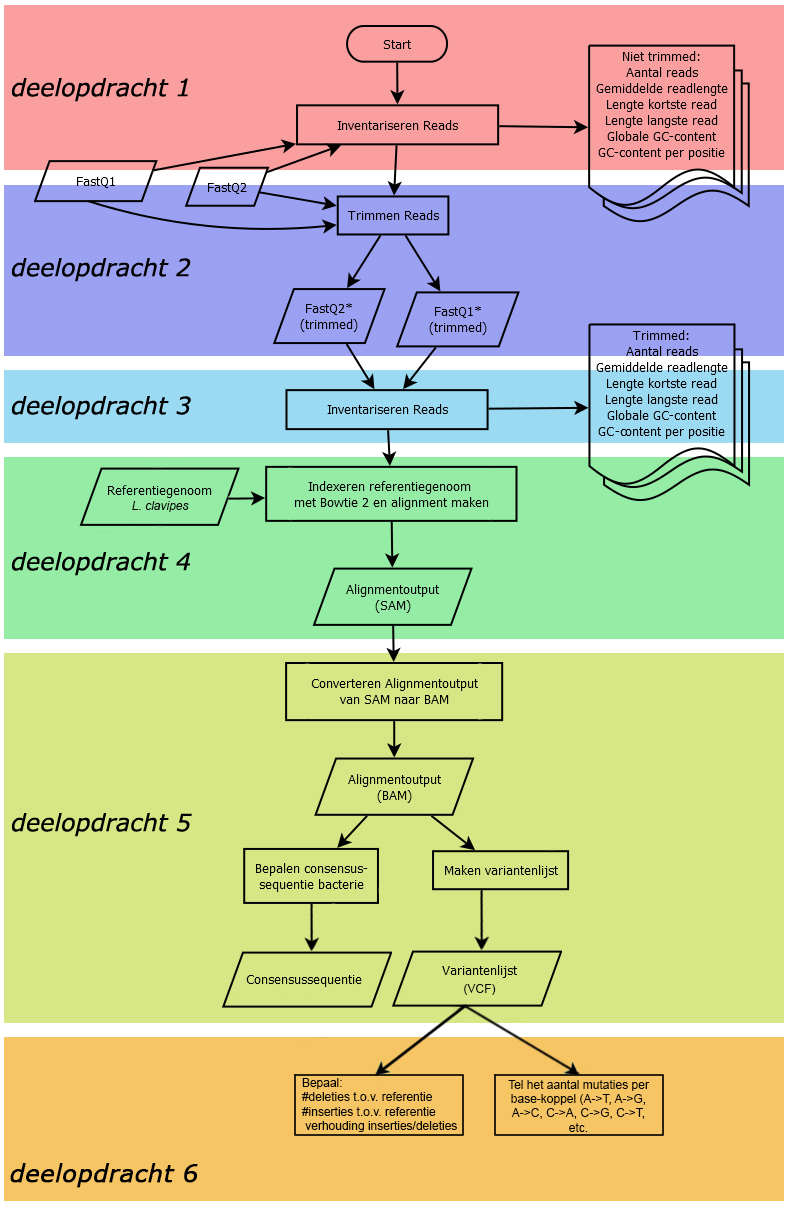
De pipeline die je gaat bouwen volgt de workflow die te zien is op de volgende pagina (fig. 1).

De opdracht is opgesplitst in deelopdrachten. De opdracht omschrijvingen zijn te vinden vanaf pagina 3. De gehele pipeline wordt aan het einde van de module opgeleverd (zie deadline die op ELO zal verschijnen). Bij de deelopdrachten staan steeds tussendeadlines aangegeven. Dit zijn geen harde deadlines, maar de ervaring vanuit eerdere jaren bij deze module is dat het lastig is de eindopdracht op tijd in te leveren als de tussendeadlines niet gehaald worden.

De eisen waaraan de pipeline moet voldoen en het nakijkmodel dat gebruikt zal worden voor de beoordeling van de pipeline zijn te vinden vanaf pagina 9.

**Achtergrond van de data:**

Je gaat werken met paired end Illumina reads, afkomstig van een sluipwesp (*Leptopilina Clavipes*), die geïnfecteerd is met een *Wolbachia­* bacterie. Ken Kraaijeveld zal tijdens een gastcollege meer vertellen over deze organismen en de onderzoeksvragen die hij probeert te beantwoorden. Tijdens deze opdracht ga je de reads alignen tegen een referentiegenoom van een niet-geïnfecteerde *L. Clavipes*. Dat resulteert uiteindelijk in een variantanalyse, waarbij je onder andere gaat kijken naar ratio’s van mutaties.



Figuur : Workflow van de pipeline

**BGSE Deelopdracht 1: Inventarisatie van de reads**

Deze opdracht vormt de eerste ‘pipe’ in je pipeline. Bij deze opdracht voer je een inventarisatie uit van twee Illumina FastQ bestanden met paired end reads, die te vinden zijn op de Bio-server. Maak in eerste instantie gebruik van twee testbestandjes. Dit zijn bestanden die slechts een fractie bevatten van de reads op de bioserver. Je kunt deze testbestanden vinden op ELO. Vergeet niet je script ook af en toe op het grote bestand van de bioserver te testen. Dan kan je zien of het ook werkt op een grote hoeveelheid data, met een acceptabele snelheid!

Schrijf een script dat deze bestanden inventariseert en de volgende output geeft:

1. Het aantal reads
2. De gemiddelde lengte van de reads
3. De minimumlengte (oftewel: de kortste) van de reads
4. Maximumlengte (oftewel: de langste) van de reads
5. De GC content (oftewel: het percentage G+C)
6. De GC content per positie in de read (oftewel: wat is van de eerste nucleotide van alle reads het gemiddelde GC%, wat van de tweede, wat van de derde, enzovoort)

Als je script af is, vergelijk je eigen resultaten dan met bestaande programma’s, zoals de gratis applicatie FastQC (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>).

**BELANGRIJK:**

1. Dit is een deelopdracht en hoeft niet afzonderlijk ingeleverd te worden, maar zal deel uitmaken van de totale eindopdracht die aan het eind van de module opgeleverd dient te worden. Zorg dat je deze deelopdracht **uiterlijk vrijdag 8 september** helemaal af (en getest!) hebt, want de volgende les wordt gestart met de volgende opdracht.

**BGSE Deelopdracht 2: Het trimmen van de reads**

Deze opdracht vormt de tweede ‘pipe’ in je pipeline. Maak voor deze opdracht gebruik van de twee Illumina FastQ bestanden met paired end reads die je ook bij opdracht 1 hebt gebruikt.

Schrijf een script dat de reads trimt:

1. Bepaal de voorwaarden waaronder je de reads wilt trimmen en beargumenteer deze keuze. Zet deze argumentatie als commentaar in de code van je script. (Bijvoorbeeld: ‘Zodra de kwaliteitscore van een read gedurende X opeenvolgende basen lager wordt dan Y, dan moet de read ‘getrimd’ worden, omdat .........’).
2. Het is een vereiste dat het script iedere read individueel evalueert en trimt op basis van de vastgestelde voorwaarden. Met andere woorden: een dynamisch script, waarbij niet alle reads uiteindelijk even lang zullen zijn. Ten overvloede: ‘Trim alle reads na 40 basen’ is dus niet voldoende.
3. Denk ook na over wat je met reads doet waarvan de kwaliteit zo laag is dat er weinig ‘goede’ nucleotiden overblijven. Dit soort reads kunnen worden verwijderd.
4. Denk eraan dat je te maken hebt met paired end reads. Wat gebeurt er als je een read uit één van beide bestanden verwijderd? Zorg ervoor dat de reads ten allen tijde juist met elkaar ‘gepaard’ blijven. Bediscussieer zo nodig met medestudenten of docenten de implicaties hiervan en welke strategie je zou kunnen gebruiken bij het trimmen.

**BELANGRIJK:**

1. Dit is een deelopdracht en hoeft niet afzonderlijk ingeleverd te worden, maar zal deel uitmaken van de totale eindopdracht die aan het eind van de module opgeleverd dient te worden. Zorg dat je deze deelopdracht **uiterlijk de laatste week van september** helemaal af (en getest!) hebt, want daarna wordt gestart met de volgende opdracht.

**BGSE Deelopdracht 3: Inventarisatie van de getrimde reads**

Deze opdracht vormt de derde ‘pipe’ in je pipeline. Voor deze opdracht hoef je niets nieuws te doen. Je moet gewoon het inventarisatiescript (dat je bij deelopdracht 1 hebt geschreven) opnieuw uitvoeren, zodat je de reads, die inmiddels getrimd zijn, kunt inventariseren. Zie je duidelijke verschillen? Zorg dat ook deze stap onderdeel wordt van je pipeline.

**Deelopdracht 4: Alignment met Bowtie 2**

Het is nu tijd om de getrimde reads te gaan alignen tegen het referentiegenoom. Dit doen we met de applicatie Bowtie 2.

* Bowtie 2 staat al geïnstalleerd op de bioserver. Zoek uit hoe je het aan kunt roepen.   
  ***HINT:*** *om de alignment sneller te laten verlopen moet je eerst je referentiegenoom indexeren. dit kan ook met behulp van Bowtie 2.*
* Indexeer het referentiegenoom (te vinden in /home/bnextgen/refgenome/)

Align je getrimde reads tegen je geïndexeerde referentiegenoom en sla het resultaat op in een SAM-bestand.

***LET OP:*** *deze stap is de bottleneck van je pipeline (duurt relatief het langst). Zorg dus dat je zeker weet dat je Bowtie 2 op de juiste manier aanroept. Misschien kan je het eerst wel testen met een fractie van de reads?*

**BELANGRIJK:**

1. Dit is een deelopdracht en hoeft niet afzonderlijk ingeleverd te worden, maar zal deel uitmaken van de totale eindopdracht die aan het eind van de module opgeleverd dient te worden. Zorg dat je deze deelopdracht **uiterlijk vrijdag 6 oktober** helemaal af (en getest!) hebt, want de volgende les wordt gestart met de volgende opdracht.

**Deelopdracht 5: bepalen variantenlijst en consensussequentie**

Nu je het resultaat van je alignment hebt (het SAM-bestand), kun je een lijst gaan opstellen van alle varianten in je eigen data ten opzichte van het referentiegenoom. Deze variantenlijst wordt opgeslagen in een VCF-bestand. De applicatie die je nodig hebt om dit te doen heet SAMtools en staat al geïnstalleerd op de BioServer.

* Zoek uit waar SAMtools allemaal voor gebruikt kan worden en hoe je het aan kan roepen
* Gebruik SAMtools om je SAM-bestand om te zetten naar een BAM-bestand. Dit is een binaire versie van je alignment data. Het is voor mensen niet leesbaar, maar is wel een stuk kleiner dan een SAM-bestand.
* Gebruik SAMtools om je BAM-bestand te sorteren  
  ***TIP:*** *als je je gesorteerde BAM-bestand hebt, kijk dan eens of je deze kunt bekijken in een BAM-viewer (bijv:* [*https://ics.hutton.ac.uk/tablet/*](https://ics.hutton.ac.uk/tablet/)*).*
* Gebruik SAMtools om een PILEUP-bestand te maken.  
  ***LET OP:*** *deze stap kan relatief lang duren. Zorg dus dat je goed onderzoek doet naar hoe je dit moet doen, voordat je het uitvoert.****TIP:*** *zoek op wat een PILEUP-bestand is. Als het gelukt is om het PILEUP-bestand te maken, bekijk dan eens de eerste honderd regels en kijk of je het kunt lezen en begrijpen.*
* gebruik BCFtools (een onderdeel van SAMtools) om je PILEUP-bestand om te zetten naar een BCF-bestand
* gebruik BCFtools om je BCF-bestand om te zetten naar een VCF-bestand

Je hebt nu een VCF-bestand met daarin alle varianten. Dit bestand kan je gebruiken om een consensus-sequentie te maken, oftewel: het genoom van de geinfecteerde sluipwesp!

* Zoek uit hoe je op basis van een VCF-bestand een consensus sequentie kunt maken.  
  ***HINT****: je kunt hier een ander onderdeel van SAMtools voor gebruiken. Dit is een PERL-script genaamd vcfutils.pl.*
* Genereer de consensus-sequentie.

**BELANGRIJK:**

1. Dit is een deelopdracht en hoeft niet afzonderlijk ingeleverd te worden, maar zal deel uitmaken van de totale eindopdracht die aan het eind van de module opgeleverd dient te worden. Zorg dat je deze deelopdracht **uiterlijk eind week 7** helemaal af (en getest!) hebt, want de volgende week wordt gestart met de volgende opdracht.

**Deelopdracht 6: beantwoorden van de onderzoeksvragen**

Je hebt nu alle data tot je beschikking en kunt de resultaten nu gaan interpreteren en er conclusies uit trekken. Beantwoord de volgende vragen m.b.v. een zelfgeschreven script:

1. Hoeveel deleties heeft jouw data ten opzichte van het referentiegenoom?
2. Hoeveel inserties heeft jouw data ten opzichte van het referentiegenoom?
3. Wat is de verhouding van deze deleties/inserties?
4. In welke aantallen komen alle mogelijke combinaties van mutaties voor? (Dus de hoeveelheden A->T, A->C, A->G, T->A, T->C, T->G, C->A, C->G, C->T, G->A, G->C, G->T)

**Let op:** je moet er dus voor zorgen dat de antwoorden op bovenstaande vragen automatisch gegenereerd worden en (als onderdeel van je pipeline) worden weggeschreven naar een tekstbestand.

**Deadline inleveren opdracht:**

Vrijdag 3 november 2017, 23.59 uur

**Voorwaarden voor inlevering:**

Het inleveren moet op drie manieren gebeuren (dit is een voorwaardelijke eis):

* Al je scripts moeten als één ZIP-bestand gemaild worden naar Saskia Stahlecker (de naamgeving van het ZIP-bestand is [NAAM STUDENT]\_[STUDENTNUMMER].ZIP)
* Al je code moet onder elkaar in een tekstdocument (.TXT) worden geplakt en via ephorus worden ingeleverd.
* Je pipeline moet op de bioserver worden geplaatst, zodat de docenten hem kunnen gebruiken:
  1. Maak in je eigen studentfolder (/home/sXXXXXXX/) een subfolder aan genaamd ‘inlevermap’.
  2. Zet al je pipeline-bestanden in deze map /home/sXXXXXXX/inlevermap/
  3. De twee inputbestanden (s\_2\_1\_sequence.txt en s\_2\_2\_sequence.txt) zet je NIET in de inlevermap, maar laat je gewoon staan in /home/bnextgen/reads. Zorg dat jouw pipeline deze inputbestanden gewoon kan gebruiken vanaf de inlevermap. Alle andere bestanden die je nodig hebt plaats je in je inlevermap.
  4. Je outputbestanden moeten terechtkomen in de inlevermap, of een subfolder daarvan.
  5. Zorg dat er op het moment van inleveren geen ouputbestanden in de inlevermap staan. Verwijder dus al je output van je laatste test run.
  6. Omdat de docenten een garantie nodig hebben dat er na de deadline niets meer aan jullie pieline is gewijzigd, doen wij direct na het verstrijken van de deadline een checksum (md5sum) van jullie inlevermap. Op het moment dat wij gaan nakijken doen wij opnieuw een checksum. Als deze checksums gelijk zijn aan elkaar dan is er intussen niets veranderd en kijken we het na. Zo niet, dan kijken we het niet na. Wij doen dan een klein onderzoekje en besluiten dan of we de EC moeten benaderen vanwege een verdenking van fraude.
  7. Kortom: zorg dat je na de deadline niets meer in je inlevermap doet. Ook niet perongeluk en ook niet met goede bedoelingen. Dit voorkomt veel ellende achteraf.

**LET OP: zie volgende pagina voor uitgebreide regels voor de opbouw van je pipeline**

**Regels voor opbouw van de pipeline:**

Je pipeline moet bestaan uit:

1. **een StartBashScript**
   * Dit script heeft de naamgeving ‘Start\_[Voornaam\_student]s\_Pipeline.sh’
   * Bijvoorbeeld: ‘Start\_Jans\_Pipeline.sh’
   * Dit script kan aangeroepen worden met de parameter ‘-h’. Wanneer de gebruiker dit doet wordt de Show Usage Information van de hele pipeline getoond. De pipeline zelf wordt dan niet gestart. De Usage information toont minimaal:
     1. *Functie van de pipeline*
     2. *Manier van gebruik (aanroepen) van de pipeline*
     3. *Voorbeeld voor het starten van de pipeline*
     4. *Een overzicht van de naamgeving van de outputbestanden van de pipeline*
   * Wanneer het StartBashScript wordt aangeroepen zonder ‘-h’, moet de volgende vraag aan de gebruiker worden gesteld: *‘Please enter a prefix that will be used for all output files:’*De gebruiker kan dan een prefix typen en deze wordt vóór alle bestandsnamen van de output gezet.
   * Vervolgens roept het StartBashScript het HighwayBashScript aan.
2. **een HighwayBashScript**
   * Dit script heeft de naamgeving ‘[Voornaam\_student]s\_Pipeline.sh’
   * Bijvoorbeeld: ‘Jans\_Pipeline.sh’
   * Dit script kan aangeroepen worden met de parameter ‘-h’. Wanneer de gebruiker dit doet wordt de Show Usage Information van het HighwayBashScript getoond. Het script zelf wordt dan niet gestart. De Usage information toont minimaal:
     1. *Functie van de pipeline (dit is dezelfde info als bij het StartBashScript)*
     2. *Manier van gebruik (aanroepen) van het HighwayBashScript*
     3. *Voorbeeld van het aanroepen van het HighwayBashScript*
   * *LET OP:* in feite zal het HighwayBashScript niet door de gebruiker aangeroepen hoeven te worden, aangezien de gebruiker het StartBashScript aanroept
   * Dit script bevat verder alle code om de verschillende scripts of bestaande applicaties aan te roepen
3. **verschillende scripts (evt. in een verschillende programmeertalen)**
   * Deze scripts hebben een naamgeving die je zelf mag bedenken, maar het is wel handig om bijvoorbeeld het nummer van de deelopdracht erin te verwerken of op een andere manier duidelijk te maken wat het script doet.
   * Bijv.: *‘deelopd2\_trimming\_script.py’* of iets dergelijks.
   * Alle scripts kunnen aangeroepen worden met de parameter ‘-h’. Wanneer de gebruiker dit doet wordt de Show Usage Information van het script getoond. Het script zelf wordt dan niet gestart. De Usage information toont minimaal:
     1. *Functie van het script*
     2. *Manier van gebruik (aanroepen) van het script*
     3. *Voorbeeld van het aanroepen van het script*

**Nakijkmodel en voorwaardelijke eisen**

|  |  |  |  |
| --- | --- | --- | --- |
| **Voorwaardelijke eisen (niet aan voldaan betekent automatisch een 1 - de opdracht wordt zo mogelijk wel nagekeken ivm feedback):** | | | pnt |
| De 'snelweg' van de pipeline is een bash script, vanwaar uit diverse andere onderdelen worden aangeroepen | |  | 1 |
| De pipeline is in zijn geheel uit te voeren en stopt nergens wegens foutmeldingen e.d. Als aan deze voorwaarde niet wordt voldaan krijgt de student tijdens de inzage de gelegenheid te laten zien dat het wel werkt als hij/zij het uitvoert op de server. In dat geval zal het alsnog nagekeken worden. | |  | 1 |
| Bij het trimmen wordt iedere read individueel geëvalueerd en wordt getrimd op basis van de vastgestelde voorwaarden. Met andere woorden: een dynamisch script, waarbij niet alle reads uiteindelijk even lang zijn | |  | 1 |
| Bij het trimmen wordt alleen aan de uiteinden getrimd. Anders gezegd: de student mag geen deleties in de sequentie introduceren door stukken met slechte kwaliteit er tussenuit te halen. | |  | 1 |
| De pipeline is binnen 24 uur klaar met runnen | |  | 1 |
| Voor deelopdracht 1, 2, 4, 5 en 6 zijn minimaal 5 punten per deelopdracht behaald | |  | **1** |
|  | |  | **6** |
|  | |  |  |
|  | |  |  |
|  | |  |  |
| **Algemeen:** | |  |  |
| De scripts kunnen aangeroepen worden met -h en tonen dan duidelijke usage information | |  | 10 |
| Het script vraagt om een prefix voor de outputbestanden en gebruikt deze ook bij het wegschrijven. | | | 5 |
|  |  | | **15** |
|  |  | |  |
| **Deelopdracht 1:** |  | |  |
| Het script geeft het correcte aantal reads | | | 3 |
| Het script geeft de correcte gemiddelde lengte | | | 4 |
| Het script geeft de correcte minimumlengte | | | 2 |
| Het script geeft de correcte maximumlengte | | | 2 |
| Het script geeft de correcte GC content | | | 5 |
| Het script geeft de correcte GC content per positie |  | | 6 |
|  |  | | **22** |
|  |  | |  |
| **Deelopdracht 2:** |  | |  |
| De voorwaarden waaronder er getrimd wordt zijn netjes als commentaar vermeld in de code van het script | | | 5 |
| De reads in beide bestanden blijven na het trimmen nog steeds 'paired', zodat een aligner deze informatie kan gebruiken | | | 10 |
| De voorwaarden waaronder wordt getrimd zijn uitgebreid, creatief en goed onderbouwd. | | | 5 |
| De reads worden duidelijk getrimd | | | 15 |
|  |  | | **35** |
|  |  | |  |
| **Deelopdracht 3:** |  | |  |
| Het inventarisatiescript is correct uitgevoerd op de getrimde reads | | | 3 |
|  |  | |  |
| **Deelopdracht 4:** |  | |  |
| Het correcte referentiegenoom is gebruikt | | | 4 |
| Het referentiegeoom is geindexeerd |  | | 4 |
| De alignment is uitgevoerd. Er is een SAM bestand met daarin alignment data. | | | 8 |
|  |  | | **16** |
|  |  | |  |
| **Deelopdracht 5:** |  | |  |
| Er is een BAM-file gemaakt | | | 4 |
| Er is een mpileup bestand gemaakt | | | 4 |
| Er is een VCF-bestand gemaakt | | | 4 |
| Er is een consensus sequentie gemaakt | | | 4 |
|  |  | | **16** |
|  |  | |  |
| **Deelopdracht 6:** |  | |  |
| Onderstaande vragen worden mbv een geautomatiseerd script beantwoord en weggeschreven naar een bestand: | | |  |
| Het wordt weggeschreven naar een apart bestand | | | 3 |
| 1. Hoeveel deleties heeft jouw data ten opzichte van het referentiegenoom? | | | 3 |
| 2. Hoeveel inserties heeft jouw data ten opzichte van het referentiegenoom? | | | 3 |
| 3. Wat is de verhouding van deze deleties/inserties? | | | 3 |
| 4. In welke aantallen komen alle mogelijke combinaties van mutaties voor? | | | 3 |
|  |  | | **15** |
|  |  | |  |
|  |  | |  |
|  | Aantal punten: | | **122** |
|  | Cijfer: | | **10,0** |