Genomics en Transcriptomics analyse muis B-cellen

Floris Menninga

2024-11-14

Contents


```
# Libraries
library(tidyverse)
library(ggplot2)
library(VariantAnnotation)
library(GenomicRanges)
library(dplyr)
library("ggVennDiagram")
```

Inleiding:

Het gekozen artikel: (Ma F (2024)) (https://pubmed.ncbi.nlm.nih.gov/38866970/) heeft betrekking tot de verandering in 3d conformatie van chromatine die leiden tot verminderde expressie van het gen Ebf1. Uit het onderzoek bleek de reden voor deze vermindering migratie van genen naar het B compartiment te zijn, hier liggen stukken chromatine die niet op dat moment tot transcriptie hoeven te komen. Het onderzoek is uitgevoerd op voorloper- B-cellen van muizen. Deze cellen hebben het gen Ebf1 nodig om te differentiëren van hematopoetische stamcellen naar uitgerijpte B cellen. Als dit gen minder tot expressie komt kunnen deze B cellen meer eigenschappen van de voorloper cel hebben.

Het onderzoek en achtergrond informatie:

Het doel van het originele onderzoek (Ma F (2024)) was om te achterhalen of verschil is in gen expressie van onder andere ebf1 tussen oude en jonge muizen. (verminderde interactie tussen ebf1 promoter en zijn enhancers)

Met deze genomics analyze trachten we de volgende onderzoeksvraag te beantwoorden: Zijn er varianten aanwezig van PAX5 en Ebf1 in het genoom van de rag2(-/-) muizen die gebruikt zijn?

Deze vraag is relevant omdat voor het transcriptomics gedeelte van het onderzoek gekeken wordt naar verschillen in expressie van deze genen.

Als het blijkt dat er varianten van deze genen aanwezig zijn kan het verschil in expressie tussen de muizen die gebruikt zijn in het onderzoek niet alleen toegewezen worden aan de factoren waar naar getest wordt, de invloed van veroudering op de expressie. Dan zou het ook kunnen komen door mutaties van deze genen.

Er is DNA sequentie data beschikbaar van rag2(-/-) muizen. Daarom kan er niet gekeken worden naar mutaties in het rag2 gen omdat deze niet meer aanwezig is.

Rag2 knockout muizen zijn niet in staat om uitgerijpte T en B lymfocyten te maken. (Shinkai Y (1992))

Het gen Ebf1 (Early B-cell factor 1) is een gen dat bijdraagt aan de differentiatie van voorloper b cellen. (Nechanitzky (2013))

De reden dat gekozen is voor dit gen is dat het resultaat van een verminderde expressie zichtbaar gemaakt kan worden door functieverlies van B cellen. Samen met PAX5 werken deze genen aan de uitrijping van hematopoetische stam cellen. Functieverlies van PAX5 leidt tot accumulatie van snel proliferende lymfoblasten die niet meer normale differentiatie kunnen ondergaan. Ook is PAX5 een tumorsuppressor gen maar dat is niet van invloed op dit onderzoek. (Chivukula and Dabbs (2011))

Verschillen tussen het originele onderzoek en dat van ons: In tegenstelling tot de analyses die betrokken zijn bij het transcriptomics gedeelte van de analyses hebben de onderzoekers van het artikel niet gebruik gemaakt van deze data daarom is er niets om mee te vergelijken. Al hebben ze wel een verouderd muis referentiegenoom gebruikt (mm10) deze vervangen we door mm39.

Workflow (Genomics):

Ondanks dat DNA data onderdeel van de dataset was, staat er in de methode/artikel niet wat ze hiermee gedaan hebben. Daarom hebben we besloten een variant analyse hierop uit te voeren, gericht op de volgende genen: PAX5, Ebf1 en FOXO1.

Het referentiegenoom (muis) zal vergeleken worden met DNA seq data. In het onderzoek was een muis genoom versie uit 2012 gebruikt (mm10), deze gaan we vervangen door de nieuwste versie (GRCm39). De eerste stap is het downloaden van de .SRA archieven die de fastq bestanden bevatten, hier wordt prefetch voor gebruikt.

Daarna worden deze .SRA's uitgepakt met behulp van fasterq-dump. De nieuw verkregen fastq bestanden kunnen nu op kwaliteit gecontroleerd worden met fastqc, hierna worden ze getrimmed met trimmomatic en wordt de kwaliteit nogmaals bekeken. Duplicaten kunnen verwijderd worden.

De software versies/liberaries die gebruitk zijn: (Deze tabel is gemaakt door Jarno)

Tool	Referentie	Versie	Waarom
Featurecounts	https://academic.oup.com/bioinformatics/article/30/7/923/232889?searchresult=1		Featurecounts is een zeer efficiënt algemeen "read" samenvattingsprogramma dat mapped reads telt voor genomische kenmerken zoals genen, exonen, promotor, genlichamen, genomische bins en chromosomale locaties. het kan worden gebruikt om zowel RNA-seq als genomische DNA-seq leesbewerkingen te tellen
FastQC	https://www.bioinformatics.babraham.ac.uk/projects/fastqc/	0.11.9	FastQC wordt gebruikt om de kwaliteit te checken van de rauwe data, hier uit is te zien of de data gelijk te gebruiken is of dat deze moet worden getrimmed. De trimmer kan ook afgesteld worden op basis van de fastqc.
freebayes	https://github.com/freebayes/freebayes	1.3.6 - linux versie	freebayes is een haplotype gebaseerde gen variant detector, ontworpen om kleine polymorfismes te detecteren, SNP's, inserties en deleties in het bijzonder. Dit programma gebruikt .BAM bestanden met een Phred+33 encoding.
seqtk	https://github.com/lh3/seqtk		seqtk wordt gebruikt voor het proceseren van sequences in het FASTA of FASTQ formaat. het "seamlessly parses" beide FASTA en FASTQ welke dan ook optimaal compressed wordt door gzip

Tool	Referentie	Versie	Waarom
Trimmomatic	https://github.com/usadellab/Trimmomatic	0.39	Trimmomatic wordt gebruikt om de data op te schonen nadat dez uit FastQC komt. Deze haalt de slechte kwaliteit paren af van de streng waardoor een hoge kwaliteit RNA- of DNA-streng overblijft die gebruikt kan worden.
bwa mem2	https://github.com/bwa-mem2/bwa-mem2?tab=readme-ov-file	2.2.1	Bwa mem2 wordt gebruikt om DNA en RNA reads te alignen tegen een gekozen referentie genoom.
Samtools	https://www.htslib.org	1.16.1	samtools is een set van "utilities" dat alignments in de SAM, Bam en CRAM formatten kan manipuleren. het kan veranderen tussen de formats, sorteren, samenvoegen en indexen, ook kan het "reads" snel vinden in elke regio
R	https://www.r-project.org	4.4.1	R is de code taal die gebruikt wordt om alle statistieken testen te doen en tevens de visualisatie van de data die komt uit het onderzoek
R-studio	https://posit.co	2023.12.1+402	R studio is het programma wat wordt gebruikt als IDE voor R
NCBI-GEO	https://www.ncbi.nlm.nih.gov/geo/		NCBI-GEO is gebruikt om het originele onderzoek te vinden waar dit onderzoek inspiratie vanaf neemt

Log:

02-09-2024:
Het gekozen artikel: Three-dimensional chromatin reorganization regulates B cell development during ageing. (Ma F (2024)) Gene expression omnibus: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE211975

SRA's downloaden:

De eerste stap om een variant calling analyse uit te voeren is het downloaden van de genetische data die verkregen was door het sequencen van de samples. Met behulp van de SRA run selector van NCBI is een selectie van SRA's gemaakt die gedownload moeten worden.

Een SRA (Sequence Read Archive) is een gecomprimeerd archief dat de sequencing reads bevat. Om deze bestanden te downloaden wordt gebruikt gemaakt van "prefetch", deze commandline tool is onderdeel van de SRA toolkit.

In de volgende stap worden deze bestanden uitgepakt. Het onderstaande stuk code is op mijn computer uitgevoerd en de data is op een externe HDD opgeslagen. Dit zelfde is gedaan op de assemblix computer waar de andere groepsleden ook bij kunnen. export om de "prefetch" binairy (tijdelijk) toe te voegen aan het PATH.

```
export PATH="/home/floris/Documenten/Applicaties/sratoolkit.3.1.1-ubuntu64/bin/:$PATH"

cd /run/media/floris/FLORIS_3/DATA_SETS/3D_Chromatine_Conformatie/DNA/SRA/

prefetch $(</run/media/floris/FLORIS_3/DATA_SETS/3D_Chromatine_Conformatie/DNA/SRA/SraAccList.csv) \
--output-directory "/run/media/floris/FLORIS_3/DATA_SETS/3D_Chromatine_Conformatie/DNA/SRA/" --max-size
```

En het zelfde op assemblix:

```
prefetch $(</students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv) \
--output-directory "/students/2024-2025/Thema05/3dconformatieChromatine/SRA/" --max-size 200G
```

.SRA bestanden naar .FASTQ omzetten:
De .SRA's zijn nu gedownload maar moeten nog omgezet worden naar een bestandstype waar de volgende tool wat mee kan, een FASTQ bestand. Hier wordt fasterq-dump voor gebruikt: Met behulp van fasterq-dump worden de .SRA bestanden uitgepakt.

fasterq-dump maakt ook onderdeel uit van de SRA toolkit en haalt de data uit het .SRA archief naar het FASTQ format.

Een FASTQ bestand bevat tekst met de sequence data, de reads en ook een bijbehorende kwaliteitsscore terwijl een fasta bestand enkel de sequentie data met een header (of meerdere headers + bijbehorende sequenties) bevat. Deze score wordt aangegeven met een ASCII karakter.

Een FASTQ bestand heeft de volgende indeling:

Een header die met "@" begint gevolgd door een sequentie ID en optioneel een beschrijving wat het bestand bevat. Daar onder zitten de sequentie letters.

En daar onder een regel met een "+" karakter. Hier onder staat de kwaliteitsscore, deze gaat van ASCII 33 ("!" teken), laagste kwaliteit tot ASCII 126 ("~" teken).

De kwaliteitsscore wordt ook wel Phred score genoemd. Deze score is logaritmisch gerelateerd aan de waarschijnlijkheid dat de "base call" verkeerd is. Q = -log(E) waarbij Q de phred score is en E de waarschijnlijkheid van verkeerde base call.

Phred-33 is het meest gebruikt maar Phred-64 bestaat ook. Het verschil is dat Phred-33 ge-encodeerd is met met ASCII 33 (!) tot ASCII 126 (~) terwijl Phred-64 van ASCII 64 (@) tot ASCII 126 gaat.

Bron: https://gatk.broadinstitute.org/hc/en-us/articles/360035531872-Phred-scaled-quality-scores

```
# Locale test: (niet op de assemblix server)
cd /run/media/floris/FLORIS_3/DATA_SETS/3D_Chromatine_Conformatie/DNA/SRA


find /run/media/floris/FLORIS_3/DATA_SETS/3D_Chromatine_Conformatie/DNA/SRA -name "*.sra" | \
  parallel fasterq-dump -O /run/media/floris/FLORIS_3/DATA_SETS/3D_Chromatine_Conformatie/DNA/SRA/FASTQ


# parallel gebruiken: cat accessionlist.txt | parallel ls -lah SRA/{}/{}.sra
```

Hier worden de bestanden uitgepakt op assemblix in onze gedeelde directory.

```
find /students/2024-2025/Thema05/3dconformatieChromatine/DNA/SRA -name "*.sra" | \
  parallel fasterq-dump -O /students/2024-2025/Thema05/3dconformatieChromatine/DNA/SRA/FASTQ {}
```

Test data genereren 16-09-2024

Om te voorkomen dat een parallel commando na het uitvoeren van een lange bewerking een fout of onverwacht resultaat geeft moet er eerst een subset van de te gebruiken data gemaakt worden. Hiermee kunnen de volgende commando's eerst uitgevoerd worden om te verifiëren dat alles goed werkt.

Om test data te genereren op basis van twee van de fastq bestanden is de volgende code gebruikt: Per sample bestand zitten er 1000000 reads in. De reden dat er voor een miljoen is gekozen en niet minder is dat de subset wel representatief moet zijn voor voor de hele dataset. Er zijn mogelijk geen varianten te vinden als de subset enkel 1000 reads bevat.

Dit commando is ook uitgevoerd op de assemblix server maar dan met een ander path naar de data. Voor het maken van deze test data is seqkit (versie 2.3.0) gebruikt.

Seqkit kan gebruikt worden voor meerdere bewerkingen zoals het filteren van sequenties op lengte / kwaliteit, DNA/RNA translateren naar een aminozuur sequentie. bron: https://bioinf.shenwei.me/seqkit

```
cd /run/media/floris/FLORIS_3/DATA_SETS/3D_Chromatine_Conformatie/DNA/SRA

seqkit head -n 1000000 SRR26980527_1.fastq > subset_SRR26980527_1.fastq
seqkit head -n 1000000 SRR26980527_2.fastq > subset_SRR26980527_2.fastq
```

Dit script is eerst lokaal uitgevoerd om dit te testen, vervolgens waren de zelfde commando's op assemblix uitgevoerd.

Sample selectie

De fastq bestanden zijn met fastqc gecontroleerd en vervolgens getrimmed met Trimmomatic met het onderstaande stuk code. Voor Trimmomatic zijn de volgende instellingen gebruikt: MINLEN:40 en SLIDINGWINDOW:4:20.

Ook zijn de volgende samples zijn niet gebruikt:
B220+CD43+IgM- sorted primary pro-B cells.
Deze samples zijn verkregen met een Illumina MiSeq.

SRR26980527
SRR26980528

SRR26980529
SRR26980530

De reden hiervoor is dat het samples van WT muizen zijn, hier gaan we het referentiegenoom voor gebruiken.

Ook was de kwaliteit van deze sequenties volgens fastqc veel slechter dan die van de onderstaande samples. Mogelijk komt dit omdat de reads te lang waren en de kwaliteit daardoor te snel naar beneden ging.

De volgende zijn wel gebruikt:
Primary pro-B cells by CD19+ selection (Rag2-/-) Deze samples zijn verkregen met een Illumina NovaSeq 6000. SRR26980549
SRR26980550
SRR26980551
SRR26980552

De onnodige samples zijn ook verwijderd uit de lijst (SraAccList.csv) omdat deze met de pipe operator aan de parallel opdracht gegeven wordt. Als de namen in de accession lijst.

Trimmen van reads

De verkregen sequence reads, korte sequenties van, in ons geval 150 bp die corresponderen met een deel van een DNA sequentie. Het genoom kan niet in zijn geheel gelezen worden door technische beperkingen van de gebruikte machines (Illumina). Namate de read langer wordt is de kans dat er fouten gemaakt worden groter, daarom, mits er genoeg reads zijn die het laaste stuk bevatten, kan er een stuk afgeknipt worden. Ook moeten de adapters verwijderd worden.

Adapters zijn korte stukken DNA (ongeveer 80bp) die aan DNA linkers die op het oppervlak van de flow cells vast zitten. bron: https://www.lubio.ch/blog/ngs-adapters

Voor het trimmen maken we gebruik van Trimmomatic. In het onderstaande code blok staan de commando's die uitgevoerd worden. TrimmomaticPE is de paired-end versie van Trimmomatic. Paired end betekend dat er twee reads zijn die in tegengestelde richting gelezen zijn.

Met -threads 16 worden het aantal treads gespecificeerd, hierdoor kunnen er meerdere bewerkingen parallel uitegevoerd worden.

De argumenten die we gebruiken voor Trimmomatic zijn:

MINLIN:40 Dit betekend dat de minimale lengte van de reads 40 baseparen betreft. **SLINDINGWINDOW:4:20** Het eerste nummer 4 specificeert de grootte van de sliding window en het tweede, 20 is de vereiste gemiddelde read kwaliteit binnen het window van 4 basen.

Een sliding window is dat er steeds, in dit geval 4 basen bekeken worden en dat dan een opgeschoven wordt enzovoort.

ILLUMINACLIP is het path naar een bestand dat adapter sequenties bevat voor Illumina adapters. **De gebruikte adapter:** TruSeq3-SE.fa:2:30:10 De 2:30:10 betekenen het volgende: 2 is de "seed mismatch", het aantal mismatches dat toegestaan is in een sequentie die een adapter kan zijn. 30 is de "palindrome clip threshold en 10 is de"simple clip threshold", specificeerd hoe accuraat de match tussen de adapter sequentie en de mogenlijke adapter in de read.

Bron: http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

De volgende code is door Ivar geschreven, op basis van het voorbeeld dat tijens de les gegeven was:

```
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | \
  parallel 'TrimmomaticPE -threads 80 ' \
                  '/students/2024-2025/Thema05/3dconformatieChromatine/fastq/{}_1.fastq ' \
```

```
                    '/students/2024-2025/Thema05/3dconformatieChromatine/fastq/{}_2.fastq ' \
                    '/students/2024-2025/Thema05/3dconformatieChromatine/Trimmomatic_output/paired/{}_forw
                    '/students/2024-2025/Thema05/3dconformatieChromatine/Trimmomatic_output/unpaired/{}_fo
                    '/students/2024-2025/Thema05/3dconformatieChromatine/Trimmomatic_output/paired/{}_rev_
                    '/students/2024-2025/Thema05/3dconformatieChromatine/Trimmomatic_output/unpaired/{}_re
                    'ILLUMINACLIP:/students/2024-2025/Thema05/3dconformatieChromatine/Trimmomatic/Trimmoma
                    'MINLEN:40 ' \
                    'SLIDINGWINDOW:4:20'
```

FastQC kwaliteits controle

Vervolgens hebben Ivar en Storm met FastQC de kwaliteit vastgesteld van de getrimmde reads, dit gebreurde ook voor het trimmen en dit proces wordt beschreven in hun logboeken. De resultaten: De reads van de Illumina NovaSeq 6000 van voldoende kwaliteit, dit zijn de samples waar we de volgende analyses op uitgevoerd hebben.Hierboven in "Sample selectie" is toegelicht waarom deze keuze gemaakt is.

Een volledig overzicht van de FastQC resultaten staat in het multiQC rapport dat ook in deze repository zit.

Na het controle van de kwaliteit van de fastq bestanden bleek het dat de samples met read lengtes van 250 bp getrimd moeten worden. Deze waren allemaal met een Illumina Miseq gesequenced.

Indexeren en alignen:

Met enkel een fastq bestand met reads is nog niet duidelijk waar in het genoom van het organisme deze reads kwamen. Met read mapping worden de reads vergeleken met een bekend genoom, in dit geval het mm39 muis referentiegenoom en worden de reads er tegen aan gelegd zodat hun locatie in het genoom bekend wordt.

Aangezien er geannoteerde versies van het referentiegenoom zijn, met de namen van de genen kan dan ook achterhaald worden van welke genen de reads deel uitmaken. BWA-mem2 is de software die hiervoor gebruikt wordt.

https://www.ebi.ac.uk/training/online/courses/functional-genomics-ii-common-technologies-and-data-analysis-methods/rna-sequencing/performing-a-rna-seq-experiment/data-analysis/read-mapping-or-alignment/

bron: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3375638/

Omdat het programma dat gebruikt gaat worden voor het alignen, BWA-mem2, een geïndexeerde versie van het referentiegenoom nodig heeft moet deze eerst gemaakt worden. Dit kan gedaan worden met bwa-mem2 index. Indexeren maakt het align proces veel sneller.

De samples SRR26980528_1 en _2 zijn gebruikt na het referentiegenoom te indexeren met BWA_MEM2.

Het gebruikte referentie genoom is mm39 (GCF_000001635.27). In het bestand met het referentiegenoom dat gedowload is (https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000001635.27/) Zijn twee referentiegenomen aanwezig: GCF_000001635.27 en GCA_000001635.9/. Het verschil is dat GCF van Refseq is en GCA van GeneBank.

We hebben gekozen voor GCF omdat de chromosoom namen die hier gebruikt worden overeen komen met de namen die verwacht worden in de database van SnpEff. Hier zijn we achter gekomen door het eerst te indexeren met het Genebank genoom, tijdens de SnpEff annotatie was de foutmelding "ERROR_CHROMOSOME_NOT_FOUND" gegeven.

In de onderstaande code chunk wordt het geïndexeerde muisgenoom "GCF_ref" genoemd. Daarna worden twee samples (forward read en reverse read) ge-aligned met het zojuist verkregen geïndexeerde referentiegenoom. Deze twee bewerkingen worden beiden met bwa-mem2 uitgevoerd. Bwa-mem2 is een snellere versie van bwa-mem.

```
# Het indexeren van het referentiegenoom met bwa-mem2.
/students/2024-2025/Thema05/3dconformatieChromatine/bwa_mem2/bwa-mem2-2.2.1_x64-linux/bwa-mem2 index -p

# Voorbeeld aligning:
bwa-mem2 mem ref.fa read1.fq read2.fq > aln-pe.sam

# Align sample SRR26980549_1.fastq (deel 1 en 2) met het referentiegenoom mm39 muis.
./bwa_mem2/bwa-mem2-2.2.1_x64-linux/bwa-mem2 mem -t 50 GCF_ref fastq/SRR26980549_1.fastq fastq/SRR26980

# Test met parallel:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'bwa_mem2/bwa-mem
```

Het mappen met behulp van BWA-mem2 is uitgevoerd met de onderstaande code. Het referentiegenoom dat in de vorige stap geïndexeerd was wordt nu gebruikt door bwa-mem2 mem. De gebruikte argumenten voor bwa-mem2: "mem": Dit is het specifieke algoritme dat gebruikt wordt voor het mappen. Het staat voor: maximum exact match. GCF_ref: de naam van de index van het refrentiegenoom dat gemaakt was in een van de bovenstaande stappen. "-t 50": 50 threads worden gebruikt.

```
# Parallel met volledig pad naar alle bestanden:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel '/students/2024-2
```

14-09-2024
In tegenstelling tot het artikel gaan we gebruik maken van Muis (*Mus musculus*) referentiegenoom: GRCm39 (GCF_000001635.27) gebruik maken. Deze is nieuwer dan versie mm10 die gebruikt was.

Aangezien het onderzoek dit jaar gepubliceerd is en voltooid was in 2022 vraag ik mij af waarom ze voor een versie uit 2012 gekozen hebben terwijl er een nieuwere beschikbaar was. In een later stadium van de analyse bleek SnpEff niet te werken met de mm39 database omdat hij niet gedownload kon worden. Maar het is onwaarschijnlijk dat dit de onderzoekers tegen gehouden heeft aangezien het ons ook gelukt is om zelf een database te maken voor mm39.

Samtools

De volgende serie bewerkingen maakt gebruik van Samtools over verschillende handelingen uit te voeren zoals: omzetten naar .bam, sorteren, duplicaten verwijderen en indexeren en zal hieronder uitgelegd worden:

.sam bestanden sorteren: 17-09-2024 Na de vorige read mapping stap zijn er .SAM (Sequence alignment map) bestanden verkregen, dit is een tekst gebaseerde manier om sequenties die aligned zijn tegen een referentie sequentie op te slaan. Een .sam bestand bestaat uit een header en een alignment deel.

Omdat de .SAM bestanden niet georderd zijn op positie in het genoom moeten ze eerst met samtools sort omgezet worden naar een .BAM bestand, dit is nu wel gesorteerd door samtools.

Daarnaast kan het opvragen van data sneller gemaakt worden als het geïndexeerd is.

Na de index stap uit gevoerd te hebben is er ook een .bai bestand, dit is een index voor het .bam bestand waardoor andere tools sneller met het .bam bestand kunnen werken.

bron: https://gatk.broadinstitute.org/hc/en-us/articles/360035890791-SAM-or-BAM-or-CRAM-Mapped-sequence-data-formats

Om al deze bewerkingen uit te voeren wordt gebruik gemaakt van samtools (versie 1.16.1). Dit is een collectie tools om met high-throughput sequence data te werken. Zo kan het bijvoorbeeld fastq omzetten naar .bam / .cram bestanden, WGS/WES mapping naar variant calls en het filteren van .vcf bestanden.

Met samtools sort -n worden de .sam bestanden gesorteerd op read naam (De QNAME kolom in het bestand). Ook wordt het bestand geconverteerd van .SAM naar .BAM.

Het format is aangegeven met "-O BAM" en multithreading wordt aangegeven met "-(**?**)" (16 threads). Het .sam bestand wordt dus direct gesorteerd en omgezet naar .bam.

Bron: http://www.htslib.org/doc/samtools-sort.html

```
# Voor een enkel sample:
samtools sort -@40 -n -O BAM -o aligned_sorted_SRR26980549.bam aligned_SRR26980549.sam

# Met parallel:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'samtools sort -@4
```

"fixmate -m" voegt mate score tags toe die gebruikt worden door markdup om de beste reads te selecteren om te houden. De reden dat fixmate nodig is, is dat reads verkregen met paraired-end sequencing niet altijd een "mate" ofterwijl een read in de tegengestelde richting hebben. In het geval dat een een read geen mate of een mate die niet voldoet aan de eisen, moet deze verwijderd worden. En verwijderen is hier gemarkeerd worden als secondaire alignement (niet gemapte reads).

Met "—Threads" wordt het aantal threads gespecificeerd dat de computer moet gebruiken voor dit commando.

Bron: http://www.htslib.org/doc/samtools-markdup.html

```
# Voor een enkel sample:
samtools fixmate -m --threads 40 aligned_sorted_SRR26980549.bam \ fixed_mates_aligned_sorted_SRR2698054

# Met parallel:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'samtools fixmate
```

Daarna met "samtools sort" op coördinaten gesorteerd. Dit betekend dat de reads gesorteerd worden op basis van waar ze voorkomen op het referentiegenoom. Bij samtools sort moet het aantal treads met -(**?**) aangegeven worden.

```
# Voor een enkel sample:
samtools sort -@80 -o sorted_coordinates_SRR26980549.bam \ fixed_mates_aligned_sorted_SRR26980549.bam

# Met parallel:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'samtools sort -@8
```

Met samtools markdup worden duplicaten gemarkeerd. Met het "-r" argument worden deze verwijderd en met "-s" worden statistieken over de data en uitgevoerde handelingen. Na het -s argument moet nog wel een bestandspad opgegeven worden waar het rapport opgeslagen moet worden.

Bron: http://www.htslib.org/doc/samtools-markdup.html

```
# Voor een enkel sample:
samtools markdup -r -s sorted_coordinates_SRR26980549.bam dedup_SRR26980549.bam

# Met parallel:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel  'samtools markdu
```

Met samtools index wordt er een index bestand gemaakt, dit is in de vorm van een .bai index.

```
# Voor een enkel sample:
samtools index -@80 dedup_SRR26980549.bam dedup_SRR26980549.bai  &

# Met parallel:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'samtools index -
```

Variant calling en filteren van varianten:

Variant calling met Freebayes:

Nu er gemapte reads zijn kan er variant calling op uitgevoerd worden. Hiermee kunnen mutaties / varianten gedetecteerd en mits ze statistisch significant zijn gerapporteerd worden.

Dit zou ook grafisch met IGV (Genome browser) kunnen maar zou veel meer tijd kosten om door alle varianten te kammen en te oordelen of ze in exonen liggen van relevante genen.

Voor variant calling is freebayes uitgekozen omdat het snel en minder complex is om mee te werken dan GATK. freebayes versie 1.3.8 is gebruitk.

Freebayes is een genetische variant detector gemaakt om kleine polymorfismen zoals: SNPs, inserties, deleties en MNPs (multi-nucleotide polymorfismes) te herkennen.

Bron: https://github.com/freebayes/freebayes

Na het uitvoeren van het variant calling met freebayes worden de output bestanden, de .vcf's gefilterd door ze in vcffilter te pipen.

In dit onderstaande stuk code worden de resultaten van de freebayes variant calling direct gefilterd op kwaliteit met vcffilter. Met "QUAL = 30" worden enkel de variants geselecteerd die 99.999% kans hebben dat er een variant zit op die plaats. De reden dat op kwaliteit gefilterd moet worden om "varianten" er uit te filteren die waarschijnlijk geen echte variant zijn. Met Vcffilter wordt er op kwaliteit gefilterd, hoe dit in zijn werking gaat is na dit codeblok uitgelegd.

```
# Voor een bestand:
/students/2024-2025/Thema05/3dconformatieChromatine/freebayes/freebayes-1.3.6-linux-amd64-static \
 -f /students/2024-2025/Thema05/3dconformatieChromatine/Mapping_ref/ncbi_dataset/ncbi_dataset/data/GCA_


# Met parallel:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel '/students/2024-2
 -f /students/2024-2025/Thema05/3dconformatieChromatine/Mapping_ref/ncbi_dataset/ncbi_dataset/data/GCA_

 # Variant calling zonder direct te filteren:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel '/students/2024-2
 -f /students/2024-2025/Thema05/3dconformatieChromatine/Mapping_ref/ncbi_dataset/ncbi_dataset/data/GCA_
```

Freebayes neemt een bam bestand als input en vergelijkt deze met een referentiegenoom.

Voordat freebayes uitgevoerd kan worden moet het referentiegenoom weer geïndexeerd worden, net als met BWA-mem2 index maar dan met samtools faidx zoals in de onderstaande code chunk beschreven is.

```
samtools faidx GCA_000001635.9_GRCm39_genomic.fna
```

Met het geïndexeerde referentiegenoom (GCA_000001635.9_GRCm39) was vervolgens de freebayes variant calling uitgevoerd.

```
# Test voor een enkel .bam bestand:
/students/2024-2025/Thema05/3dconformatieChromatine/freebayes/freebayes-1.3.6-linux-amd64-static -f GCA
```

Er ging iets verkeerd tijden het pipen (|) van het resultaat van freebayes in vcffilter ondanks dat dit exacte commando wel op de website van freebayes staat. Nu zijn deze twee bewerkingen in twee delen opgesplitst, eerst varianten zoeken met freebayes en daarna filteren met Vcffilter.

Filteren:

Met –minQualScore 30 wordt gefilterd op de waarschijnlijkheid dat de variant echt aanwezig is.

Met de formule P = 10^(-Q/10) kan de kans uitgerekend worden dat de variant niet echt aanwezig is. (P = probability, waarschijnlijkheid)

Met een score van 30 is dit: P = 10^(-Q/10) = 0.001. In andere woorden: De kans is 100-0.001 = 99.999% dat het wel klopt.

```
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'java -jar /stude
```

De .vcf's zijn nu gefilterd en filtered_variant_SRRnummer genoemd zoals: filterd_variant_SRR26980549.vcf.

Converteren van .bam naar .bed

Eerst is er een .bed bestand van het niet door mensen leesbare .bam bestand gemaakt, de leesbare tegenhanger hiervan is .sam maar voor sommige analyses in R is een .bed bestand handiger omdat het minder ruimte in beslag neemt.

Met behulp van bamToBed wordt het .bam bestand omgezet naar een .bed (Browser extensible data).

De eerste kolom van een .bed bevat de naam van het chromosoom, de tweede kolom het start coördinaat van de feature, de derde kolom het eind coördinaat van de feature, 4e kolom de naam, daarna optioneel een score en dan de streng (-/+).

bron: https://genome.ucsc.edu/FAQ/FAQformat.html

```
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'bamToBed -i dedu
```

Annoteren van de .vcf bestanden:

We hebben nu, na variant calling, een serie grote bestanden die het verschil beschrijven tussen onze samples en het referentiegenoom van de muis (mm39).

Om vragen te kunnen beantwoorden zoals: zit de mutatie in een gen, in een exon, veranderd het eiwit er door of maakt de variant een stopcodon. (nonsense mutatie) moeten de .vcf bestanden geannoteerd worden, hier wordt SnpEff (Cingolani et al. (2012)) voor gebruikt. Aan de hand van een referentiedatabase die gedownload kan worden met SnpEff kan een .vcf geannoteerd worden.

Deze databases zijn voor 38000 genomen gemaakt maar het is ook mogelijk om deze zelf te maken. Met een bekend model organisme zoals de muis (*Mus musculus*) zou dit niet hoeven maar tijdens het uitvoeren van SnpEff bleek dat de nieuwste mm39 database niet gedownload kon worden.

Tenzij we een oude versie zoals mm10 willen gebruiken moeten we zelf een database hiervoor maken.

De genen worden ook geannoteerd zodat ze geen namen hebben zoals SRR26980549.46263152/2 maar bijvoorbeeld: EBF1.

Over zicht van enkele mutaties die met SnpEff gedetecteerd kunnen worden: (afkomstig van de SnpEff website)

Soort mutatie:	Betekenis:	Voorbeeld:
SNP	Single-Nucleotide Polymorphism	Reference = 'A', Sample = 'C'
Ins	Insertion	Reference = 'A', Sample = 'AGT'
Del	Deletion	Reference = 'AC', Sample = 'C'
MNP	Multiple-nucleotide polymorphism	Reference = 'ATA', Sample = 'GTC'
MIXED	Multiple-nucleotide and an InDel	Reference = 'ATA', Sample = 'GTCAGT'

Het onderstaande script was eerst uitgevoerd met mm39 als referentiegenoom voor de annotatie maar dit gaf een foutmelding (Op de github pagina van het programma snpEff waren er meer meldingen van dit gedrag. https://github.com/pcingola/SnpEff/issues/536) dus is het opnieuw uitgevoerd met het oude mm10 referentiegenoom. Dit werkte wel maar zal niet door ons gebruikt worden.

Dit is het zelfde referentiegenoom dat ook het artikel gebruikt was. Maar later hebben we zelf een database gemaakt (SnpEff hem laten maken) voor mm39.

```
# Test:
/students/2024-2025/Thema05/3dconformatieChromatine/snpEff/snpEff.jar mm39 filtered_variants.vcf > anno

# Voor alle .vcf bestanden:
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'java -jar \ /stu

# Alleen het isec .vcf bestand:
java -jar /students/2024-2025/Thema05/3dconformatieChromatine/snpEff/snpEff/snpEff.jar mm39 /students/2
```

Database maken voor SnpEff:

Omdat de database niet gedownload kon worden hebben we hem zelf gemaakt. De instruckties op de snpEff website zijn opgevolgd: https://pcingola.github.io/SnpEff/snpeff/build_db/

```
# Navigeer naar de directory waar de databases opgeslagen worden.
cd snpEff/snpEff/data/

# Download het .gff bestand:
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/635/GCF_000001635.27_GRCm39/GCF_000001635.27_
```

Debuggen van database maken:

Het maken van de mm39 database ging niet zonder problemen, eerst vereiste het programma bestanden die niet aangegeven waren op de website van SnpEff. Dit waren protein.fna en cds.fna. Deze bestanden waren gedownload van de NCBI website: https://ftp.ensembl.org/pub/release-112/fasta/mus_musculus/dna/

Eerst geprobeerd een .GFF annotatie bestand te gebruiken, dit leverde een foutmelding op over het bestand. Daarna een .GTF annotatiebestand gebruikt waarmee we naar de volgende foutmelding gingen.

SnpEff een foutmelding over een missend CDS bestand. Deze melding kan genegeerd worden met de flag "–noCheckCds"

De bestandsstructuur van de /data directory moet er als volgt uitzien: De .bin bestanden zijn gegenereerd na het maken van de database maar zonder de genomes/ directory werkt het niet. Ook moeten er symlinks

naar de bestanden in de mm39 map gemaakt worden. (Of de bestanden kopieren maar dat is minder net) Output van tree in de data/ map:

genomes: Kopie of symlink naar de zelfde bestanden die in de mm39 map staan, zonder deze stap werkte het niet. genes.gtf.gz mm39.fna.gz

mm10: Dit is het oude referentiegenoom, mm10. snpEffectPredictor.bin

mm39: De volgende bestanden zijn nodig, het proces om ze te verkrijgen is hierboven beschreven. cds.fa genes.gtf mm39.fa protein.fa sequences.fa

De volgende bestanden zijn gegenereerd na het uitvoeren van snpEff build: snpEff_genes.txt snpEff_summary.html snpEffectPredictor.bin

Hier worden de vereiste bestanden gedownload:

```
# Downloaden van benodigde bestanden:

# Referentiegenoom (Refseq)
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/635/GCF_000001635.27_GRCm39/GCF_000001635.27_

# Annotatiebestand: (.GTF)
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/635/GCF_000001635.27_GRCm39/GCF_000001635.27_

# protein.faa:
wget https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/635/GCF_000001635.27_GRCm39/GCF_000001635.27_

# Plaats bestanden in SnpEff/snpEff/data/mm39/

# Hernoem bestand GCF_000001635.27_GRCm39_genomic.fna.gz --> sequences.fa

# Hernoem bestand GCF_000001635.27_GRCm39_protein.faa.gz --> protein.fa
```

En vervolgens wordt de database gemaakt met snpEff build. de “-gtf22” flag geeft het versienummer van het .gtf bestand aan. “-nodownload” betekend dat snpEff niet eerst probeert de database te downloaden van het internet omdat hij niet gedownload kan worden. “-v” betekend verbose, hij print alles dat gebeurt naar het scherm, dit is niet nodig maar heeft wel geholpen bij het oplossen van de problemen. En tot slot de naam van het configuratiebestand, mm39. Dit configuratiebestand was al wel aanwezig in de snpEff/ directory, snpEff.config. In dit bestand staat beschreven wat voor organisme het betreft.

```
java -jar /students/2024-2025/Thema05/3dconformatieChromatine/snpEff/snpEff/snpEff.jar build -gtf22 -no
```

Chromosoom namen wijzigen:

De chromosoomnamen waren in de .vcf bestanden met de GeneBank namen aangeduid terwijl SnpEff RefSeq namen verwacht. Om de namen te veranderen aan de hand van een tekst bestand met oude namen aan de linker kant en de nieuwe namen aan de rechterkant volgens de tabel op de volgende website: https://www.ncbi.nlm.nih.gov/grc/mouse/data Dit gebeurt met behulp van “bcftools annotate –rename-chrs” om deze tool te gebruiken moet het .vcf bestand eerst gecomprimeerd worden met bgzip, daarna moet een index gemaakt worden met tabix en daarna kan bcftools annotate zelf uitgevoerd worden.

```
bgzip -c /students/2024-2025/Thema05/3dconformatieChromatine/variant_calling/isec.vcf > /students/2024-
```

```
tabix -p vcf /students/2024-2025/Thema05/3dconformatieChromatine/variant_calling/isec.vcf.gz
```

```
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel \
'bcftools annotate --rename-chrs /students/2024-2025/Thema05/3dconformatieChromatine/variant_calling/nam
```

Annotatie test:

```
java -jar /students/2024-2025/Thema05/3dconformatieChromatine/snpEff/snpEff/snpEff.jar -v -stats /studen
```

Nu met parallel voor alle bestanden: Ook worden er html bestanden met statistieken over de annotatie gegenereerd.

Met "-Xmx80G" wordt de maximale geheugen capaciteit aangegeven voor de java virtuele machine, 80GB in dit geval, dit bleek later niet nodig te zijn omdat freebayes maar een kleine fractie er van gebruikte.

```
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel \
'java -Xmx80G -jar /students/2024-2025/Thema05/3dconformatieChromatine/snpEff/snpEff/snpEff.jar -v -sta
```

Isec

Er zijn vier biologische replicaten, met behulp van "bcftools isec -c all" worden enkel de varianten die in alle vier de muizen aanwezig zijn geselecteerd. Hierdoor worden veel "willekeurige" mutaties er uit gefilterd die niet in alle vier de samples aanwezig waren. Bron: https://samtools.github.io/bcftools/bcftools.html In het onderstaande code chunk wordt bcftools isec uitgevoerd met argumenten: "-c all" dit betekend dat alle varianten die overeen komen tussen de vier samples in het output bestand opgeslagen moeten worden.

Voordat bcftools isec uitgevoerd kan worden moeten de vcf bestanden gecomprimeerd worden met bgzip.

```
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'bgzip annotated_
```

Daarna, om de vervolg processen sneller te maken (en omdat isec anders een foutmelding geeft) moet een index bestand gemaakt worden. Hier wordt "tabix" voor gebruikt. Het onderstaande commando gebruikt de net gemaakte bgzip gecomprimeerde bestanden van de .vcf als input.

```
cat /students/2024-2025/Thema05/3dconformatieChromatine/SRA/SraAccList.csv | parallel 'tabix -p vcf anno
```

Vervolgens is isec zelf uitgevoerd. Het verkregen isec.vcf bestand bevat alle varianten die de vier samples gemeen hebben. Het is ook mogelijk om met isec te specificeren dat er andere soorten overlap moet zijn, wanneer niet alle samples de variaten gemeen hebben maar alleen sample 1 met sample 2. Op de zelfde manier zijn alle andere combinaties ook mogelijk maar om de variant selectie kleiner te maken is er gekozen voor "isec -c all" wat er dus voor zorgt dat de variant in alle vier de biologische replicaten aanwezig moet zijn voordat deze genoteerd wordt in het isec.vcf bestand.

```
bcftools isec -c all annotated_SRR26980549.vcf annotated_SRR26980550.vcf annotated_SRR26980551.vcf annot
```

Annoteren Isec.vcf

Omdat het verkregen isec bestand geen annotatie metadata heeft moet deze stap opnieuw uitgevoerd worden. (of isec uitvoeren op niet geannoteerde .vcf's en daarna een enkele annotatie stap uitvoeren)

```
java -jar /students/2024-2025/Thema05/3dconformatieChromatine/snpEff/snpEff/snpEff.jar -v -stats /stude
```

Kwaliteitscontrole geannoteerde vcf bestanden:

Met de flag -stat maakt SnpEff ook een html bestand met informatie over de geannoteerde bestanden. Hieronder staat een overzicht van de soorten mutaties (variaten) die voorkwamen in deze data.

Number variants by type

Type	Total
SNP	43,232
MNP	11,068
INS	81,250
DEL	18,985
MIXED	877
INV	0
DUP	0
BND	0
INTERVAL	0
Total	**155,412**

Number of effects by impact

Type (alphabetical order)	Count	Percent
HIGH	7,879	1.001%
LOW	2,790	0.355%
MODERATE	4,498	0.572%
MODIFIER	771,771	98.073%

Number of effects by functional class

Type (alphabetical order)	Count	Percent
MISSENSE	1,251	52.829%
NONSENSE	111	4.688%
SILENT	1,006	42.483%

Missense / Silent ratio: 1.2435

SRR26980549:

Base changes (SNPs)

	A	C	G	T
A	0	2,186	3,450	1,863
C	2,812	0	1,984	4,098
G	14,367	1,982	0	2,848
T	1,864	3,448	2,330	0

Number variants by type

Type	Total
SNP	44,543
MNP	13,093
INS	100,782
DEL	18,720
MIXED	744
INV	0
DUP	0
BND	0
INTERVAL	0
Total	177,882

Number of effects by impact

Type (alphabetical order)	Count	Percent
HIGH	10,059	1.079%
LOW	3,215	0.345%
MODERATE	5,591	0.599%
MODIFIER	913,786	97.977%

Number of effects by functional class

Type (alphabetical order)	Count	Percent
MISSENSE	1,334	53.232%
NONSENSE	209	8.34%
SILENT	963	38.428%

Missense / Silent ratio: 1.3853

changes (SNPs)

Rapportage van de verkregen data:

Nu er een bestand is dat alle varianten die de vier replicaten gemeen hebben bevat moet dit gerapporteerd worden. De volgende analyses worden in R uitgevoerd.

Visualisatie van de verkregen data:

Test:

```
#vcf <- readVcf(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR26980550.vcf")
#nonsense_variants <- vcf[info(vcf)$TYPE == "nonsense"]


#test <- vcf[info(vcf)]

#head(test)
```

```
# Test venn diagram:

A <- c("A","B")
B <- c("C","B")
C <- c("A","B")
D <- c("A","B")
x <- list(A, B, C, D)

# 2D Venn diagram
ggVennDiagram(x) +
    scale_fill_gradient(low = "green", high = "blue")
```

Number variants by type

Type	Total
SNP	53,001
MNP	14,126
INS	91,255
DEL	24,244
MIXED	1,099
INV	0
DUP	0
BND	0
INTERVAL	0
Total	**183,725**

Number of effects by impact

Type (alphabetical order)	Count	Percent
HIGH	8,103	0.911%
LOW	3,171	0.357%
MODERATE	4,583	0.515%
MODIFIER	873,191	98.216%

Number of effects by functional class

Type (alphabetical order)	Count	Percent
MISSENSE	1,740	56.021%
NONSENSE	161	5.184%
SILENT	1,205	38.796%

Missense / Silent ratio: 1.444

Figure 1: SRR26980551

Number variants by type

Type	Total
SNP	56,799
MNP	14,792
INS	126,406
DEL	28,046
MIXED	1,091
INV	0
DUP	0
BND	0
INTERVAL	0
Total	**227,134**

Number of effects by impact

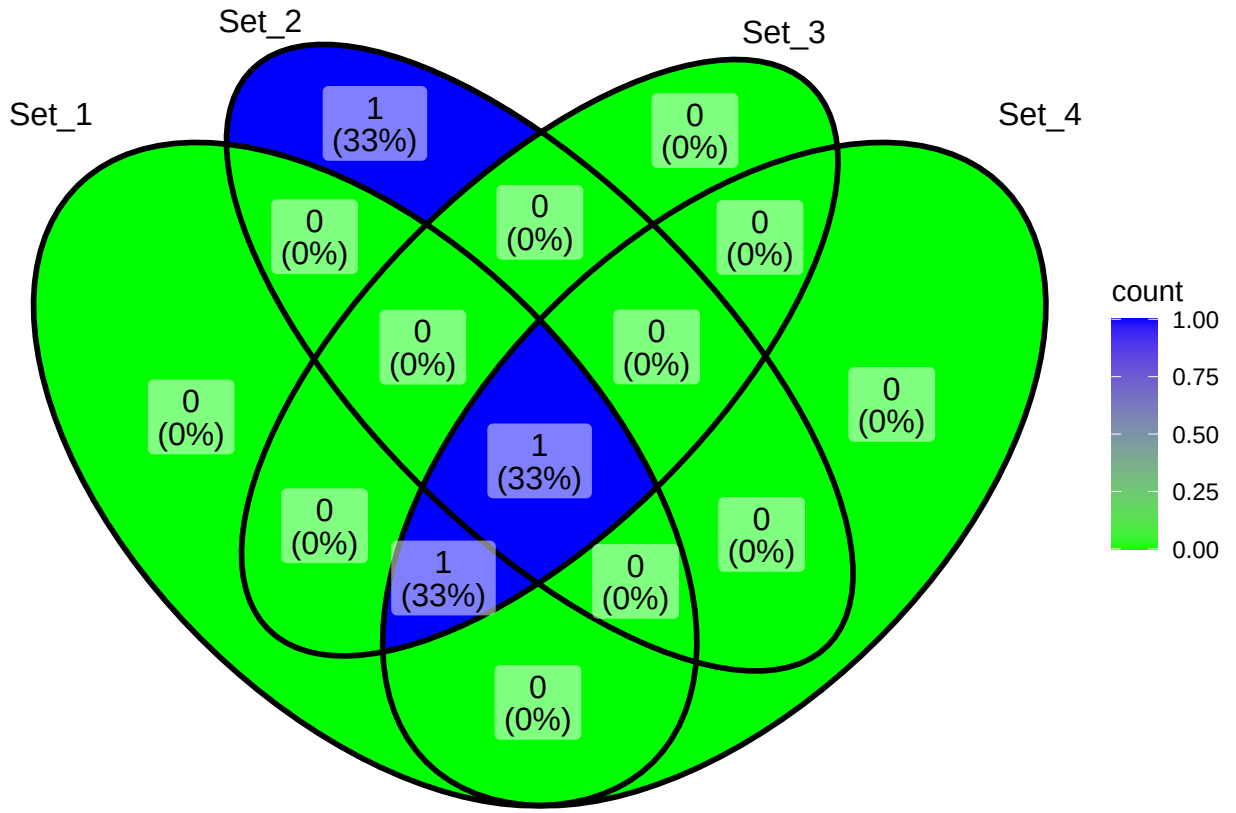
Type (alphabetical order)	Count	Percent
HIGH	11,135	0.989%
LOW	3,770	0.335%
MODERATE	6,495	0.577%
MODIFIER	1,104,032	98.099%

Number of effects by functional class

Type (alphabetical order)	Count	Percent
MISSENSE	1,778	55.684%
NONSENSE	182	5.7%
SILENT	1,233	38.616%

Missense / Silent ratio: 1.442

Figure 2: SRR26980552



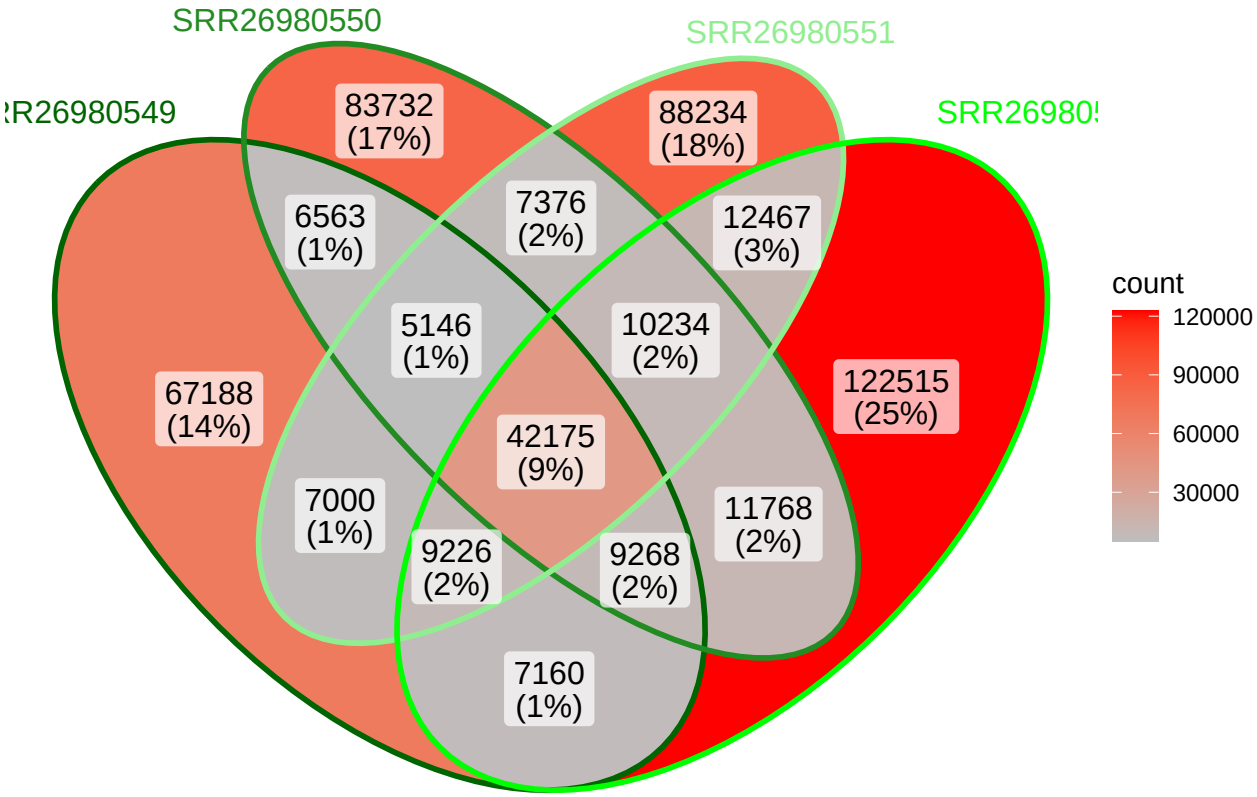
Lees de verkregen .vcf bestanden in om te gebruiken voor het Venn diagram. Kolom V2 is de locatie op het genoom, wanneer deze het zelfde of anders is tussen samples zal dit weergeven worden in het diagram. Dit hoeft echter niet te betekenen dat hier de zelfde mutatie zit, als er bijvoorbeeld een A -> C in het ene sample zit en op de zelfde locatie in het genoom in het andere sample een G -> T is dit niet de zelfde mutatie maar zal het wel "binnen" in het Venn diagram geplaatst worden omdat ze hier beiden een variant hebben.

```
#SRR26980549_vcf <- readVcf(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR26980549.
#SRR26980550_vcf <- readVcf(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR26980550.
#SRR26980551_vcf <- readVcf(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR26980551.
#SRR26980552_vcf <- readVcf(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR26980552.

SRR26980549_vcf <- read.table(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR2698054
SRR26980550_vcf <- read.table(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR2698055
SRR26980551_vcf <- read.table(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR2698055
SRR26980552_vcf <- read.table(file = "/home/floris/Documenten/Data_set/PRJNA885415/annotated_SRR2698055
```

```
overlap_list <- list(SRR26980549_vcf$V2, SRR26980550_vcf$V2, SRR26980551_vcf$V2, SRR26980552_vcf$V2)

ggVennDiagram(overlap_list, set_color = c("darkgreen","forestgreen","lightgreen","green"),
              category.names = c("SRR26980549","SRR26980550","SRR26980551", "SRR26980552"),
              label_alpha=0.7) +
              scale_fill_gradient(low = "gray", high = "red")
```



Het bovenstaande Venn diagram weergeeft de overlap tussen varianten in de vier biologische replicaten. Maar 9% van de varianten komen overeen tussen de vier samples. Daarnaast heeft SRR26980552 de meeste mutaties die de overige samples niet hebben. In het logboek van Storm staat een Grange object met variant informatie.

Discussie:

Omdat de DNA-seq data niet gebruikt was in het originele onderzoek kunnen we de verkregen resultaten niet vergelijken met die van het onderzoek. Wel waren er meer varianten en meer verschil tussen de biologische replicaten dan ik had verwacht. Daarnaast was het isec vcf bestand nog niet benut, al is het Venn diagram wel een visuele intersect tussen de samples.

referenties

Chivukula, Mamatha, and David J. Dabbs. 2011. "Chapter 21 - Immunocytology." In *Diagnostic Immunohistochemistry (Third Edition)*, edited by David J. Dabbs, Third Edition, 890–918. Philadelphia: W.B. Saunders. https://doi.org/https://doi.org/10.1016/B978-1-4160-5766-6.00025-X.

Cingolani, P., A. Platts, M. Coon, T. Nguyen, L. Wang, S. J. Land, X. Lu, and D. M. Ruden. 2012. "A Program for Annotating and Predicting the Effects of Single Nucleotide Polymorphisms, SnpEff: SNPs in the Genome of Drosophila Melanogaster Strain W1118; Iso-2; Iso-3." *Fly* 6 (2): 80–92.

Ma F, Du H, Cao Y. 2024. "Three-Dimensional Chromatin Reorganization Regulates b Cell Development During Ageing." *Nat Cell Biol.* Jun;26(6) (26): 991–1002. https://doi.org/10.1038/s41556-024-01424-9.

Nechanitzky, Akbas, R. 2013. "Transcription Factor EBF1 Is Essential for the Maintenance of b Cell Identity and Prevention of Alternative Fates in Committed Cells." *Nat Immunol* 14 (June): 867–75. https://doi.org/https://doi.org/10.1038/ni.2641.

Shinkai Y, Lam KP, Rathbun G. 1992. "RAG-2-Deficient Mice Lack Mature Lymphocytes Owing to Inability to Initiate v(d)j Rearrangement." *Cell* 6 (March): 855–67. https://doi.org/10.1016/0092-8674(92)90029-c.