#Instellen working directory

#Stel je working directory in.

setwd("C:/Users/Cecil/OneDrive - NHL Stenden/Documents/Van Hall Larenstein/schooljaar 2/jaar 2 periode 4/het project/")

#Je kan met het getwd() commando zien waar je bent. getwd() staat voor get working directory.

getwd()

# Vervang de bestandsnaam hieronder met je eigen zip-bestand

#Hiermee worden de bestanden uitgepakt in een submap

unzip("Data\_RA\_raw.zip", exdir = "Reuma\_data")

# Readmapping

#installeer packages

install.packages('BiocManager')

BiocManager::install('Rsubread')

library(Rsubread)

browseVignettes('Rsubread')

#Met de volgende code kun je het referentiegenoom indexeren.

buildindex(

basename = 'ref\_reuma',

reference = 'Homo\_sapiens.GRCh38.dna.toplevel.fa.gz',

memory = 11000,

indexSplit = TRUE)

# reuma monsters

align.Norm1 <- align(index = "ref\_reuma", readfile1 = "SRR4785819\_1\_subset40k.fastq",readfile2 = "SRR4785819\_2\_subset40k.fastq", output\_file = "Norm1.BAM")

align.Norm2 <- align(index = "ref\_reuma", readfile1 = "SRR4785820\_1\_subset40k.fastq",readfile2 = "SRR4785820\_2\_subset40k.fastq", output\_file = "Norm2.BAM")

align.Norm3 <- align(index = "ref\_reuma", readfile1 = "SRR4785828\_1\_subset40k.fastq",readfile2 = "SRR4785828\_2\_subset40k.fastq", output\_file = "Norm3.BAM")

align.Norm4 <- align(index = "ref\_reuma", readfile1 = "SRR4785831\_1\_subset40k.fastq",readfile2 = "SRR4785831\_2\_subset40k.fastq", output\_file = "Norm4.BAM")

align.RA1 <- align(index = "ref\_reuma", readfile1 = "SRR4785979\_1\_subset40k.fastq",readfile2 = "SRR4785979\_2\_subset40k.fastq", output\_file = "RA1.BAM")

align.RA2 <- align(index = "ref\_reuma", readfile1 = "SRR4785980\_1\_subset40k.fastq",readfile2 = "SRR4785980\_2\_subset40k.fastq", output\_file = "RA2.BAM")

align.RA3 <- align(index = "ref\_reuma", readfile1 = "SRR4785986\_1\_subset40k.fastq",readfile2 = "SRR4785986\_2\_subset40k.fastq", output\_file = "RA3.BAM")

align.RA4 <- align(index = "ref\_reuma", readfile1 = "SRR4785988\_1\_subset40k.fastq",readfile2 = "SRR4785988\_2\_subset40k.fastq", output\_file = "RA4.BAM")

#BAM bestanden genereren

#Wanneer je reads mapt tegen een referentiegenoom met align(), wordt er een BAM-bestand aangemaakt.

#Dit doe je met het Rsamtools package:

# Laad Rsamtools voor sorteren en indexeren

library(Rsamtools)

# Bestandsnamen van de monsters

samples <- c('Norm1', 'Norm2', 'Norm3', 'Norm4', 'RA1', 'RA2', 'RA3', 'RA4')

# Voor elk monster: sorteer en indexeer de BAM-file

# Sorteer BAM-bestanden

lapply(samples, function(s) {sortBam(file = paste0(s, '.BAM'), destination = paste0(s, '.sorted'))

})

file.exists("Reuma\_data.zip")

# dag 2: count matric

#We gebruiken hiervoor het featureCounts() commando uit het Rsubread package.

# benodigde packages

library(readr)

library(dplyr)

library(Rsamtools)

library(Rsubread)

# Je definieert een vector met namen van BAM-bestanden. Elke BAM bevat reads van een RNA-seq-experiment (bijv. behandeld vs. controle).

allsamples <- c("eth1.BAM", "eth2.BAM", "eth3.BAM", "control1.BAM", "control2.BAM", "control3.BAM")

count\_matrix <- featureCounts(

files = allsamples,

annot.ext = "",

isPairedEnd = TRUE,

isGTFAnnotationFile = TRUE,

GTF.attrType = "gene\_id",

useMetaFeatures = TRUE

)

#Resultaten bekijken

head(count\_matrix$annotation)

head(count\_matrix$counts)

#Countmatrix opslaan en inladen

# Bekijk eerst de structuur van het object

# Extract uit featureCounts-object

str(count\_matrix)

# Haal alleen de matrix met tellingen eruit

counts <- count\_matrix$counts

# Stel duidelijke kolomnamen in, zodat ze overeenkomen met je samples. Dit helpt bij herkenning tijdens visualisaties en analyses.

colnames(counts) <- c("GeneID", "eth1", "eth2", "eth3", "control1", "control2", "control3")

# De eerste kolom bevat de gen-ID’s. Je kunt deze beter als rij-namen gebruiken (in plaats van als gewone kolom).

rownames(counts) <- counts[, 1]

#Verwijder nu zelf de eerste kolom (GeneID) uit de matrix. Je hebt deze al gebruikt als rij-naam, dus die dubbele informatie mag eruit. Weet je nog hoe dat moet?

# Voer je eigen code in.

#Nu de matrix klaar is, sla je deze op. Zo kun je hem in Werkcollege 3 direct opnieuw inlezen.

write.csv(counts, "bewerkt\_countmatrix.csv")

#Bekijk de eerste paar rijen.

head(counts)

Dag 3: Statistiek en analyse

#DESeq2-analyse

#Lees het bestand uit werkcollege 2 in.

counts <- read.csv("bewerkt\_countmatrix.csv", row.names = 1)

# Behandelingstabel maken

treatment <- c("ethanol", "ethanol", "ethanol", "control", "control", "control")

treatment\_table <- data.frame(treatment)

rownames(treatment\_table) <- c('eth1', 'eth2', 'eth3', 'control1', 'control2', 'control3')