

Logbook Jasper Jonker

Jasper Jonker

2024-16-10

Inleiding

In dit logboek kan je lezen hoe de visualisatie is uit gevoerd op de transcriptomics van het vak 2.1.2 Genomics & Transcriptomics. In dit vak gingen we bezig met een onderzoek van [] en opnieuw doen en kijken welke conclusie wij eruit kunnen halen. Ik ga zelf bezig met het laten zien van de data. Eerst ga ik kijken wat we allemaal kunnen doen en daarna ga ik een vergelijking laten zien.

Hier zie je de library's die ik heb gebruikt.

```
library(DESeq2)
library(dplyr)
library(ggplot2)
library(EnhancedVolcano)
```

Deseq

Hieronder staat code voor de verwerking van deseq dit is meer ook voorbeeld over hoe je het moet gebruiken is dus van alle groepen van ons onderzoek. Dus tussen NSG (slecht immuumsysteem) muisen en C57BL/6 (goed immuumsysteem) muisen. Elke van deze drie groepen zijn weer onder verdeeld in 3 groepen. Verchele, basline en Entinostat. Ik ga het laten zien van C57BL/6 basline tegen C57BL/6 Entinostat.

Dit stukje code is van Ramon dus als je er meer uit leg over wilt hebben moet je in zijn logbook kijken.

```
file.names <- list.files('/students/2024-2025/Thema05/BlaasKanker/Transcriptomics/output/mapping/',
                         pattern = '*_star_ReadsPerGene.out.tab')

## Function for reading in files
read_sample <- function(file.name) {
  ## Extract the sample name for naming the column (retaining the 'SRR....' part)
  sample.name <- strsplit(file.name, ".", fixed = TRUE)[[1]][1]
  sample <- read.table(file.name, header = FALSE, sep="\t",
                       row.names = NULL, skip = 4)
  ## Rename the count column
  names(sample)[2] <- sample.name
  ## Return a subset containing the transcript ID and sample name columns
  return(sample[c(1, 2)])
}

setwd('/students/2024-2025/Thema05/BlaasKanker/Transcriptomics/output/mapping/')
```

```

## Read the FIRST sample
counts <- read_sample(file.names[1])

## Read the remaining files and merge the contents
for (file.name in file.names[2:length(file.names)]) {
  sample <- read_sample(file.name)
  counts <- merge(counts, sample, by = 1)
}

# Set the row names to the transcript IDs
rownames(counts) <- counts$V1
counts <- counts[-1]

col_data <- read.csv("/students/2024-2025/Thema05/BlaasKanker/etc/DSEQ_verwerking(Sheet1).csv", sep = "
rownames(col_data) <- col_data[,1]
col_data$source_name

## [1] "C57BL/6_bladder tumor_Vehicle"      "C57BL/6_bladder tumor_Entinostat"
## [3] "C57BL/6_bladder tumor_Vehicle"      "C57BL/6_bladder tumor_Vehicle"
## [5] "C57BL/6_bladder tumor_Vehicle"      "C57BL/6_bladder tumor_Entinostat"
## [7] "C57BL/6_bladder tumor_Entinostat"   "C57BL/6_bladder tumor_Vehicle"
## [9] "C57BL/6_bladder tumor_Baseline"     "C57BL/6_bladder tumor_Baseline"
## [11] "C57BL/6_bladder tumor_Entinostat"   "C57BL/6_bladder tumor_Baseline"
## [13] "C57BL/6_bladder tumor_Entinostat"   "C57BL/6_bladder tumor_Entinostat"
## [15] "C57BL/6_bladder tumor_Vehicle"      "C57BL/6_bladder tumor_Baseline"
## [17] "NSG_bladder tumor_Baseline"         "NSG_bladder tumor_Baseline"
## [19] "NSG_bladder tumor_Baseline"         "NSG_bladder tumor_Baseline"
## [21] "NSG_bladder tumor_Baseline"         "NSG_bladder tumor_Entinostat"
## [23] "NSG_bladder tumor_Entinostat"       "NSG_bladder tumor_Entinostat"
## [25] "NSG_bladder tumor_Entinostat"       "NSG_bladder tumor_Entinostat"
## [27] "NSG_bladder tumor_Vehicle"          "NSG_bladder tumor_Vehicle"
## [29] "NSG_bladder tumor_Vehicle"          "NSG_bladder tumor_Vehicle"
## [31] "NSG_bladder tumor_Vehicle"          "NSG_bladder tumor_Vehicle"

```

Annot count_data

Het annoteren van onze data zorgt ervoor dat de genen de NCBI naamgeving krijgen. Zo is het straks overzichtelijker om informatie te vinden over de genen.

```

library(dplyr)
library(tidyr)
col_data <- col_data %>%
  dplyr::group_by(strain, treatment) %>%
  dplyr::mutate(r_num = row_number()) %>%
  dplyr::ungroup() %>%
  mutate(condition = paste0(strain, "_", treatment, "_r", r_num))
head(col_data)

## # A tibble: 6 x 6
##   Run                      source_name      strain treatment r_num condition
##   <chr>                    <chr>           <chr>    <chr>     <int> <chr>
## 1 <chr>                    <chr>           <chr>    <chr>     <int> <chr>

```

```

## 1 SRR12129014_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Vehicle      1 C57BL/6_~
## 2 SRR12129015_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Entinost~     1 C57BL/6_~
## 3 SRR12129016_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Vehicle      2 C57BL/6_~
## 4 SRR12129017_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Vehicle      3 C57BL/6_~
## 5 SRR12129018_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Vehicle      4 C57BL/6_~
## 6 SRR12129019_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Entinost~     2 C57BL/6_~

```

In de bestanden kon niks gevonden worden over de genen, maar zo is het wel duidelijker waar elke kolom voor staat. Dat moet nu nog toegepast worden in de counts df

```

for (i in 1:nrow(col_data)) {
  idx <- grep(col_data$Run[i], names(counts))
  names(counts)[idx] <- col_data$condition[i]
}
head(counts)

##          C57BL/6_Vehicle_r1 C57BL/6_Entinostat_r1 C57BL/6_Vehicle_r2
## MissingGeneID           5859            4387           7820
## gene-0610005C13Rik        3              0              0
## gene-0610006L08Rik        0              1              0
## gene-0610009E02Rik        0              1              1
## gene-0610009L18Rik       53              29             25
## gene-0610010K14Rik      1024             790            881
##          C57BL/6_Vehicle_r3 C57BL/6_Vehicle_r4 C57BL/6_Entinostat_r2
## MissingGeneID           8707            3405           3821
## gene-0610005C13Rik        0              0              0
## gene-0610006L08Rik        0              0              2
## gene-0610009E02Rik        1              0              1
## gene-0610009L18Rik       93              43             34
## gene-0610010K14Rik      1408             897            523
##          C57BL/6_Entinostat_r3 C57BL/6_Vehicle_r5 C57BL/6_Baseline_r1
## MissingGeneID           3777            6823           4673
## gene-0610005C13Rik        1              0              0
## gene-0610006L08Rik        0              0              0
## gene-0610009E02Rik        1              4              0
## gene-0610009L18Rik       44              93             25
## gene-0610010K14Rik      492             1304            975
##          C57BL/6_Baseline_r2 C57BL/6_Entinostat_r4
## MissingGeneID           7162            2160
## gene-0610005C13Rik        0              1
## gene-0610006L08Rik        0              1
## gene-0610009E02Rik        0              1
## gene-0610009L18Rik       55              15
## gene-0610010K14Rik      1390             419
##          C57BL/6_Baseline_r3 C57BL/6_Entinostat_r5
## MissingGeneID           5096            2718
## gene-0610005C13Rik        0              0
## gene-0610006L08Rik        0              0
## gene-0610009E02Rik        0              4
## gene-0610009L18Rik       24              49
## gene-0610010K14Rik      972              676
##          C57BL/6_Entinostat_r6 C57BL/6_Vehicle_r6 C57BL/6_Baseline_r4
## MissingGeneID           3870            9562           5163
## gene-0610005C13Rik        0              0              0
## gene-0610006L08Rik        0              0              0

```

```

## gene-0610009E02Rik          6          0          2
## gene-0610009L18Rik         34         43         45
## gene-0610010K14Rik        431        964        876
## NSG_Baseline_r1  NSG_Baseline_r2  NSG_Baseline_r3
## MissingGeneID      8453       8045       5803
## gene-0610005C13Rik       4          1          2
## gene-0610006L08Rik       0          0          0
## gene-0610009E02Rik       7          4          7
## gene-0610009L18Rik       38         47         31
## gene-0610010K14Rik      967        1298       664
## NSG_Baseline_r4  NSG_Baseline_r5  NSG_Entinostat_r1
## MissingGeneID      6777       8639       6767
## gene-0610005C13Rik       1          3          1
## gene-0610006L08Rik       1          0          0
## gene-0610009E02Rik       8          10         6
## gene-0610009L18Rik       45         48         95
## gene-0610010K14Rik      1001       1336       1470
## NSG_Entinostat_r2  NSG_Entinostat_r3  NSG_Entinostat_r4
## MissingGeneID      9366       9204       3295
## gene-0610005C13Rik       6          2          0
## gene-0610006L08Rik       0          0          0
## gene-0610009E02Rik       5          6          0
## gene-0610009L18Rik      139        124         64
## gene-0610010K14Rik      1528       1339       387
## NSG_Entinostat_r5  NSG_Vehicle_r1  NSG_Vehicle_r2
## MissingGeneID      12470      5393       7122
## gene-0610005C13Rik       2          2          1
## gene-0610006L08Rik       1          0          0
## gene-0610009E02Rik       13         6          11
## gene-0610009L18Rik      113        41          50
## gene-0610010K14Rik      1393       735       1145
## NSG_Vehicle_r3  NSG_Vehicle_r4  NSG_Vehicle_r5
## MissingGeneID      8495       8854       7730
## gene-0610005C13Rik       2          4          2
## gene-0610006L08Rik       0          0          1
## gene-0610009E02Rik       5          8          4
## gene-0610009L18Rik      122        150        160
## gene-0610010K14Rik      1618       1689       1511

```

Nu zijn de SRR* namen vervangen met waar ze voor staan en is de df duidelijker te lezen, om straks makkelijker de kolommen op te halen die horen bij elk mogelijke variant groepeer ik deze.

```
print(colnames(counts))
```

```

## [1] "C57BL/6_Vehicle_r1"    "C57BL/6_Entinostat_r1"  "C57BL/6_Vehicle_r2"
## [4] "C57BL/6_Vehicle_r3"    "C57BL/6_Vehicle_r4"    "C57BL/6_Entinostat_r2"
## [7] "C57BL/6_Entinostat_r3"  "C57BL/6_Vehicle_r5"    "C57BL/6_Baseline_r1"
## [10] "C57BL/6_Baseline_r2"   "C57BL/6_Entinostat_r4"  "C57BL/6_Baseline_r3"
## [13] "C57BL/6_Entinostat_r5"  "C57BL/6_Entinostat_r6"  "C57BL/6_Vehicle_r6"
## [16] "C57BL/6_Baseline_r4"   "NSG_Baseline_r1"      "NSG_Baseline_r2"
## [19] "NSG_Baseline_r3"       "NSG_Baseline_r4"      "NSG_Baseline_r5"
## [22] "NSG_Entinostat_r1"     "NSG_Entinostat_r2"     "NSG_Entinostat_r3"
## [25] "NSG_Entinostat_r4"     "NSG_Entinostat_r5"     "NSG_Vehicle_r1"
## [28] "NSG_Vehicle_r2"        "NSG_Vehicle_r3"      "NSG_Vehicle_r4"
## [31] "NSG_Vehicle_r5"

```

```

C57BL_Vehicle <- grep("C57BL/6_Vehicle", names(counts))
C57BL_Entinostat <- grep("C57BL/6_Entinostat", names(counts))
C57BL_Baseline <- grep("C57BL/6_Baseline", names(counts))

nsg_Vehicle <- grep("NSG_Vehicle", names(counts))
nsg_Entinostat <- grep("NSG_Entinostat", names(counts))
nsg_Baseline <- grep("NSG_Baseline", names(counts))

```

Nu kunnen deze variabelen gebruikt worden om de juiste kolommen te selecteren.

Dan kan dit nu samengevoegd worden met DESEQ, wat Janine heeft uitgezocht. Bekijk haar logboek voor meer informatie over de code.

```

dds <- DESeqDataSetFromMatrix(countData = counts,
                               colData = col_data,
                               design = ~ 0 + source_name)
head(dds$source_name)

## [1] C57BL/6_bladder tumor_Vehicle      C57BL/6_bladder tumor_Entinostat
## [3] C57BL/6_bladder tumor_Vehicle      C57BL/6_bladder tumor_Vehicle
## [5] C57BL/6_bladder tumor_Vehicle      C57BL/6_bladder tumor_Entinostat
## 6 Levels: C57BL/6_bladder tumor_Baseline ... NSG_bladder tumor_Vehicle

# Prefiltering on low gene counts
keep <- rowSums(counts(dds)) >= 10
ddst <- dds[keep,]

# Setting factor level
#dds$treatment <- relevel(dds$treatment, ref = "Baseline")
ddst$treatment <- factor(dds$treatment, levels = c("Baseline", "Entinostat", "Vehicle"), )
ddst$treatment

## [1] Vehicle      Entinostat   Vehicle      Vehicle      Vehicle      Entinostat
## [7] Entinostat   Vehicle      Baseline    Baseline    Entinostat   Baseline
## [13] Entinostat   Entinostat   Vehicle      Baseline    Baseline    Baseline
## [19] Baseline     Baseline    Baseline    Entinostat   Entinostat   Entinostat
## [25] Entinostat   Entinostat   Vehicle      Vehicle     Vehicle     Vehicle
## [31] Vehicle

## Levels: Baseline Entinostat Vehicle

# Running deseq
ddst <- DESeq(ddst)
res <- results(ddst)

head(res)

## log2 fold change (MLE): source_name NSG_bladder tumor_Vehicle vs C57BL/6_bladder tumor_Baseline
## Wald test p-value: source_name NSG_bladder tumor_Vehicle vs C57BL/6_bladder tumor_Baseline
## DataFrame with 6 rows and 6 columns
##           baseMean log2FoldChange      lfcSE       stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## MissingGeneID      6213.959413      0.0613458  0.281620  0.2178319  0.82756011
## gene-0610005C13Rik      0.928517      3.1919781  1.353181  2.3588702  0.01833067
## gene-0610006L08Rik      0.325219      0.1850353  3.878567  0.0477071  0.96194965
## gene-0610009E02Rik      3.241600      3.3513189  1.108739  3.0226393  0.00250581

```

```

## gene-0610009L18Rik 60.496770      1.1274910  0.398268  2.8309856 0.00464048
## gene-0610010K14Rik 998.035854     -0.0444090  0.294859 -0.1506110 0.88028262
##                                     padj
##                                     <numeric>
## MissingGeneID      0.8980987
## gene-0610005C13Rik 0.0596486
## gene-0610006L08Rik NA
## gene-0610009E02Rik 0.0128493
## gene-0610009L18Rik 0.0208305
## gene-0610010K14Rik 0.9335965

```

Dus dit is niet wat ik wou hebben of kan gebruiken voor mijn gedeelte. Nu is er vergelijking gedaan van NSG_bladder tumor_Vehicle tegen C57BL/6_bladder tumor_Baseline. Ik ga een subset maken van alle C57BL/6 baseline en C57BL/6 Entinostat maken. Deze twee situaties ga ik dus ui werken.

```

C57BL_subset <- dds[, c(C57BL_Entinostat,C57BL_Baseline)]
```

```

C57BL_subset <- DESeqDataSet(C57BL_subset, design = ~ treatment)
C57BL_subset$treatment <- relevel(C57BL_subset$treatment, ref = "Baseline")
C57BL_subset <- DESeq(C57BL_subset)
```

```

resultaat_C57BL <- results(C57BL_subset)
```

```

head(resultaat_C57BL)
```

```

## log2 fold change (MLE): treatment Entinostat vs Baseline
## Wald test p-value: treatment Entinostat vs Baseline
## DataFrame with 6 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## MissingGeneID      6168.262965    0.17444819  0.261976  0.66589386 0.5054789
## gene-0610005C13Rik 0.405535      1.77823220  3.225919  0.55123283 0.5814741
## gene-0610006L08Rik 0.796704      2.73551203  2.365204  1.15656470 0.2474503
## gene-0610009E02Rik 2.715126      2.96435360  1.375938  2.15442441 0.0312069
## gene-0610009L18Rik 52.611245     0.70508328  0.357148  1.97420351 0.0483586
## gene-0610010K14Rik 1092.089170    0.00181008  0.364426  0.00496695 0.9960370
##           padj
##           <numeric>
## MissingGeneID      0.6245088
## gene-0610005C13Rik NA
## gene-0610006L08Rik NA
## gene-0610009E02Rik 0.0737484
## gene-0610009L18Rik 0.1045137
## gene-0610010K14Rik 0.9977485

```

Visualization oefen

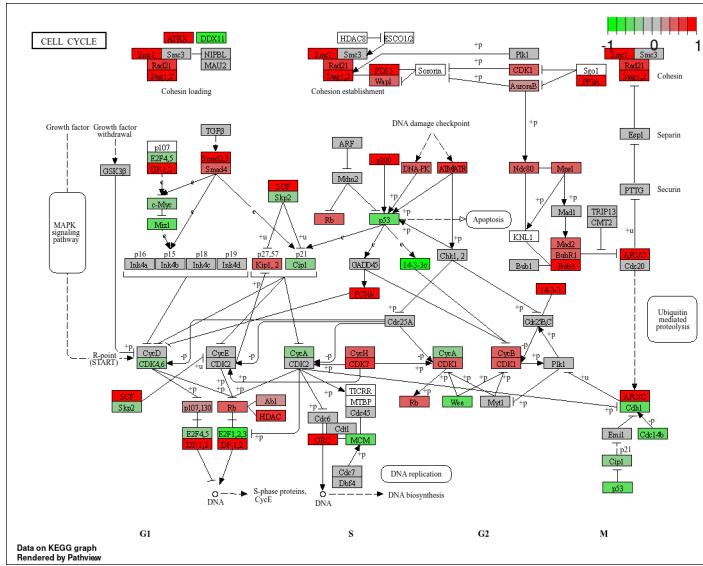
Dus ik ga als eerst bezig met alles laten zien en mogelijke manieren om het te laten zien. Dit doe ik omdat het niet nodig is om met zijn vieren bezig te gaan met trimmen en met deseq bezig te zijn. We hebben niet alle tijd dus moeten we slim bezig gaan met onze tijd.

Pathway Analysis

Hier is een pathway analysis. Hier kan je dus zien als een gen word aangetast wat voor invloed het heeft op bepaalde biologische processen.

<https://pathview.r-forge.r-project.org/pathview.pdf> is een een pdf met alle informatie

```
library(pathview)
data(gse16873.d)
pv.out <- pathview(gene.data = gse16873.d[, 1], pathway.id = "04110",
species = "hsa", out.suffix = "gse16873")
```



Volcano plot oefen

Een volcano plot is handig om te gebruiken want daar in kan je zien hoe groot een verandering is op de x-as (de log2FoldChange) en hoe significant een verandering is (p-value)

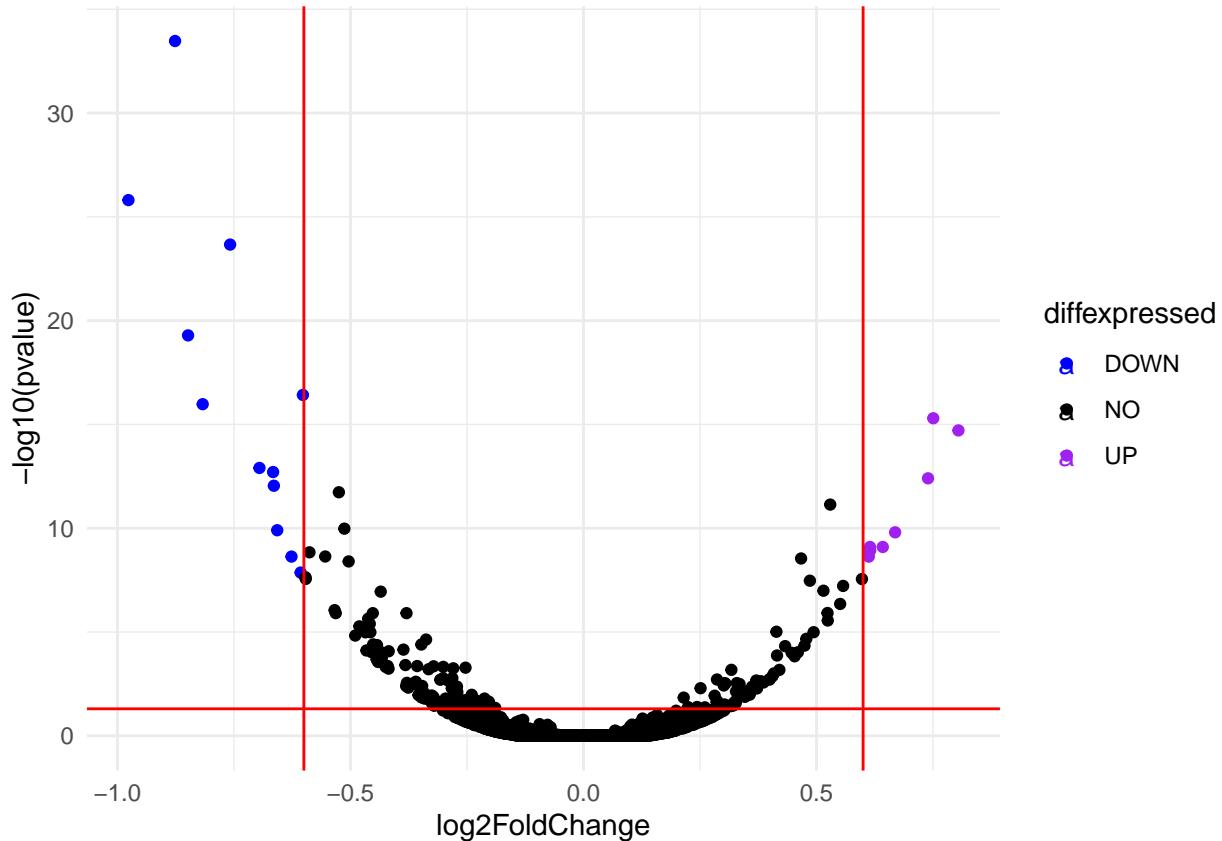
Manier 1

Dit is 1 manier om een volcano plot te maken. het op deze manier word het hem denk ik niet het word hier nogal moeilijk gemaakt terwijl het makkelijker kan.

```
tmp <- readRDS("/students/2024-2025/Thema05/BlaasKanker/Transcriptomics/testData/de_df_for_volcano.rds")
de <- tmp[complete.cases(tmp), ]
de$delabel <- NA
de$diffexpressed <- "NO"
de$delabel[de$diffexpressed != "NO"] <- de$gene_symbol[de$diffexpressed != "NO"]

de$diffexpressed[de$log2FoldChange > 0.6 & de$pvalue < 0.05] <- "UP"
# if log2Foldchange < -0.6 and pvalue < 0.05, set as "DOWN"
de$diffexpressed[de$log2FoldChange < -0.6 & de$pvalue < 0.05] <- "DOWN"

ggplot(data=de, aes(x=log2FoldChange, y=-log10(pvalue), col=diffexpressed, label=delabel)) +
  geom_point() +
  theme_minimal() +
  geom_text() +
  geom_vline(xintercept=c(-0.6, 0.6), col="red") +
  geom_hline(yintercept=-log10(0.05), col="red") +
  scale_color_manual(values=c("blue", "black", "purple"))
```



Manier 2

Hier zie je ook dat het met Deseq word gerund. Hier is het makkelijker om te laten zien en zie ook gelijk hoe de deseq werkt en hoe makkelijk je er data uit kan halen.

```

library(airway)
library(magrittr)
data('airway')
airway$dex %>% relevel('untrt')

ens <- rownames(airway)

library(org.Hs.eg.db)
symbols <- mapIds(org.Hs.eg.db, keys = ens,
  column = c('SYMBOL'), keytype = 'ENSEMBL')
symbols <- symbols[!is.na(symbols)]
symbols <- symbols[match(rownames(airway), names(symbols))]
rownames(airway) <- symbols
keep <- !is.na(rownames(airway))
airway <- airway[keep,]

library('DESeq2')

dds <- DESeqDataSet(airway, design = ~ cell + dex)

dds <- DESeq(dds, betaPrior=FALSE)

res <- results(dds,

```

```

contrast = c('dex','trt','untrt')

res_1 <- lfcShrink(dds,
contrast = c('dex','trt','untrt'), res=res, type = 'normal')

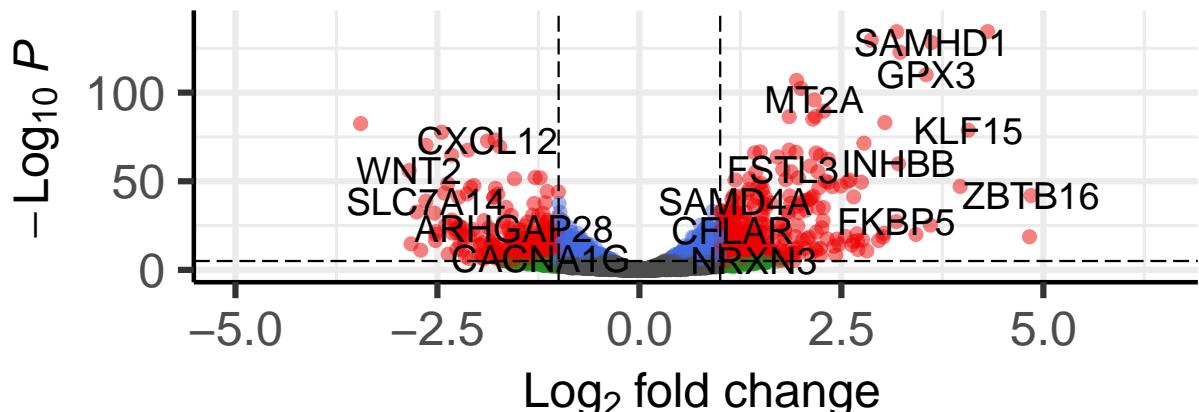
EnhancedVolcano(res_1,
  lab = rownames(res_1),
  x = 'log2FoldChange',
  y = 'pvalue')

```

Volcano plot

EnhancedVolcano

● NS ● Log₂ FC ● p-value ● p – value and log₂ FC



total = 35329 variables

Clusterprofiling gsa oefen

<http://yulab-smu.top/biomedical-knowledge-mining-book/enrichment-overview.html#gsea-algorithm>

```

library("clusterProfiler")

data(geneList, package="DOSE")
head(geneList)

##      4312     8318    10874    55143    55388      991
## 4.572613 4.514594 4.418218 4.144075 3.876258 3.677857

gene <- names(geneList)[abs(geneList) > 2]
head(gene)

## [1] "4312"  "8318"  "10874" "55143" "55388" "991"
GO classification

```

```

ggo <- groupGO(gene      = gene,
                 OrgDb     = org.Hs.eg.db,
                 ont       = "CC",
                 level     = 3,
                 readable  = TRUE)

head(ggo)

##          ID             Description Count GeneRatio geneID
## GO:0000133 GO:0000133          polarisome    0   0/207
## GO:0000408 GO:0000408        EKC/KEOPS complex    0   0/207
## GO:0000417 GO:0000417           HIR complex    0   0/207
## GO:0000444 GO:0000444      MIS12/MIND type complex    0   0/207
## GO:0000808 GO:0000808 origin recognition complex    0   0/207
## GO:0000930 GO:0000930 gamma-tubulin complex    0   0/207

```

GO over-representation analysis

Hier kan je zien dat bepaalde genen een over presentatie hebben omdat ze een p.dajust hebben kleiner dan 0.05 en quaule kleiner dan 0,05 dat betekent dat de deze genen meer tot uiting komen dan verwacht wordt.
<https://www.youtube.com/watch?v=egO7Lt92gDY>

```

ego <- enrichGO(gene      = gene,
                  universe   = names(geneList),
                  OrgDb     = org.Hs.eg.db,
                  ont       = "CC",
                  pAdjustMethod = "BH",
                  pvalueCutoff  = 0.01,
                  qvalueCutoff  = 0.05,
                  readable   = TRUE)

head(ego)

##          ID             Description GeneRatio
## GO:0005819 GO:0005819          spindle    26/201
## GO:0000775 GO:0000775 chromosome, centromeric region 19/201
## GO:0072686 GO:0072686           mitotic spindle 17/201
## GO:0000779 GO:0000779 condensed chromosome, centromeric region 16/201
## GO:0098687 GO:0098687      chromosomal region 23/201
## GO:0000793 GO:0000793      condensed chromosome 19/201
##          BgRatio      pvalue      p.adjust      qvalue
## GO:0005819 332/11884 6.308871e-11 1.886352e-08 1.713356e-08
## GO:0000775 188/11884 3.940180e-10 5.890568e-08 5.350349e-08
## GO:0072686 151/11884 6.423389e-10 6.401978e-08 5.814858e-08
## GO:0000779 138/11884 1.350430e-09 1.009446e-07 9.168708e-08
## GO:0098687 305/11884 1.819474e-09 1.088045e-07 9.882616e-08
## GO:0000793 210/11884 2.580005e-09 1.285703e-07 1.167792e-07
##
## GO:0005819 CDCA8/CDC20/KIF23/CENPE/ASPM/DLGAP5/SKA1/NUSAP1/TPX2/TACC3/NEK2/CDK1/MAD2L1/KIF18A/BIRC5/1
## GO:0000775                                         CDCA8/CENPE/NDC80/TOP2A/HJURP/SKA1/NEK2/CENPM/
## GO:0072686                                         KIF23/CENPE/ASPM/SKA1/NUSAP1/TPX2/TAC
## GO:0000779                                         CENPE/NDC80/HJURP/SKA1/NEK2/CI
## GO:0098687 CDCA8/CENPE/NDC80/TOP2A/HJURP/SKA1/NEK2/CENPM/RAD51AP1/CENPN/CDK1/ERCC61
## GO:0000793                                         CENPE/NDC80/TOP2A/NCAPH/HJURP/SKA1/NEK2/CENPM/CI
##          Count
## GO:0005819    26

```

```

## GO:0000775    19
## GO:0072686    17
## GO:0000779    16
## GO:0098687    23
## GO:0000793    19

```

GO Gene Set Enrichment Analysis

<https://www.youtube.com/watch?v=egO7Lt92gDY> <https://www.youtube.com/watch?v=Yi4d7JIlAsM>

Dus je ziet hier GO:0000775 het meest interessant is door nes met een score van 0.6230073

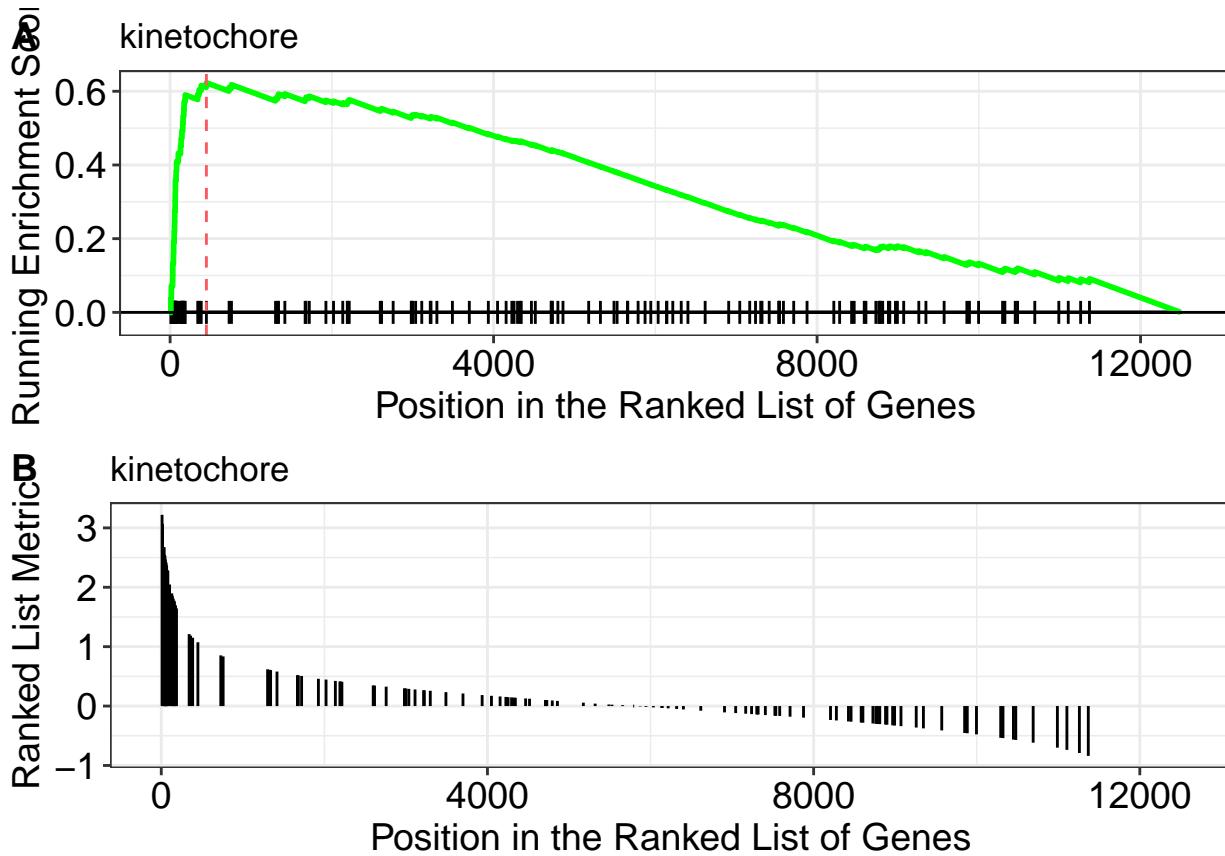
```

ego3 <- gseGO(geneList      = geneList,
                OrgDb        = org.Hs.eg.db,
                ont          = "CC",
                minGSSize    = 100,
                maxGSSize    = 500,
                pvalueCutoff = 0.05,
                verbose      = FALSE)

head(ego3)

```

ID	Description	setSize			
## GO:0000776 GO:0000776	kinetochore	130			
## GO:0000779 GO:0000779	condensed chromosome, centromeric region	138			
## GO:0000775 GO:0000775	chromosome, centromeric region	188			
## GO:0000228 GO:0000228	nuclear chromosome	175			
## GO:0098687 GO:0098687	chromosomal region	305			
## GO:0000793 GO:0000793	condensed chromosome	210			
enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
## GO:0000776	0.6230073	2.653186	1e-10	1.62e-09	1.031579e-09
## GO:0000779	0.6216009	2.652795	1e-10	1.62e-09	1.031579e-09
## GO:0000775	0.5970689	2.646887	1e-10	1.62e-09	1.031579e-09
## GO:0000228	0.5875327	2.578399	1e-10	1.62e-09	1.031579e-09
## GO:0098687	0.5419949	2.526146	1e-10	1.62e-09	1.031579e-09
## GO:0000793	0.5507360	2.486529	1e-10	1.62e-09	1.031579e-09
leading_edge					
## GO:0000776	tags=21%, list=4%, signal=20%				
## GO:0000779	tags=24%, list=6%, signal=23%				
## GO:0000775	tags=20%, list=4%, signal=19%				
## GO:0000228	tags=34%, list=15%, signal=29%				
## GO:0098687	tags=27%, list=14%, signal=24%				
## GO:0000793	tags=28%, list=14%, signal=24%				
## GO:0000776					
## GO:0000779					
## GO:0000775					
## GO:0000228					
## GO:0098687	55143/1062/10403/7153/55355/220134/4751/79019/10635/55839/983/54821/4085/81930/81620/332/				
## GO:0000793					
p1 <- gseaplot(ego3, geneSetID = 1, by = "runningScore", title = ego3\$Description[1])					
p2 <- gseaplot(ego3, geneSetID = 1, by = "preranked", title = ego3\$Description[1])					
cowplot::plot_grid(p1, p2, ncol=1, labels=LETTERS[1:3])					



<https://yulab-smu.top/biomedical-knowledge-mining-book/enrichplot.html>

Visualization C57BL ent tegen Basline

Clusterprofiling gsa

Hier ga ik de resultaten laten zien van de enrichment result. Je kan het op veel mogelijk laten zien dus laten we maar eerst even de resultaten maken. Hier onder zie je een aantal librarys die ik ga gebruiken bij de visualisatie van de plotjes.

```
library(clusterProfiler)
library(org.Mm.eg.db)
library(ggplot2)
library(enrichplot)
```

Voor de enrichment result heb je de deseq nodig van je data. Hieronder zie nog wel even snel de deseq voor treatment Entinostat vs Baseline in de C57BL/6 muizen soort.

```
head(resultaat_C57BL)

## log2 fold change (MLE): treatment Entinostat vs Baseline
## Wald test p-value: treatment Entinostat vs Baseline
## DataFrame with 6 rows and 6 columns
##           baseMean log2FoldChange      lfcSE       stat     pvalue
## <numeric>    <numeric>    <numeric>    <numeric>    <numeric>
## MissingGeneID 6168.262965  0.17444819  0.261976  0.66589386 0.5054789
## gene-0610005C13Rik 0.405535   1.77823220  3.225919  0.55123283 0.5814741
## gene-0610006L08Rik 0.796704   2.73551203  2.365204  1.15656470 0.2474503
```

```

## gene-0610009E02Rik      2.715126      2.96435360  1.375938 2.15442441 0.0312069
## gene-0610009L18Rik     52.611245      0.70508328  0.357148 1.97420351 0.0483586
## gene-0610010K14Rik    1092.089170     0.00181008  0.364426 0.00496695 0.9960370
##                                     padj
##                               <numeric>
## MissingGeneID          0.6245088
## gene-0610005C13Rik      NA
## gene-0610006L08Rik      NA
## gene-0610009E02Rik     0.0737484
## gene-0610009L18Rik     0.1045137
## gene-0610010K14Rik     0.9977485

```

Maar eerst moeten wij een genlijst maken want die heeft gseGO nodig om de enrichment results te doen. De genlijst maken we met behulp van de deseq. Tijdens het maken van de genlijst halen we de NA resultaten van de log2FoldChange er uit. Hier onder zie je de code die er voor gebruikt word. We sorteren het gelijk op volgorde van hoog naar laag.

```

resultaat_C57BL_2 <- resultaat_C57BL[!is.na(resultaat_C57BL$log2FoldChange),]
gen_lijst <- resultaat_C57BL_2$log2FoldChange
names(gen_lijst) <- gsub("gene-", "", row.names(resultaat_C57BL_2))

gen_lijst <- sort(gen_lijst, decreasing = TRUE)
head(gen_lijst)

```

```

##           Myl3        Myh7        Fgb        Fgg        Csn3
## 24.57719  24.02822  23.08704  22.98055  22.10822
## 2310065F04Rik
##           20.36657

```

Hier zie je een genlijst met genen met een log2FoldChange van hoog naar laag gesorteerd.

Met de genlijst doen we nu een gseGo met een cutoffvalue van 0.05 dat is normaal de p cut off waarde en niet te vergeten dat we de muis gebruiken Dus Mm en niet Hs gebruiken.

```

gsa_C57BL <- gseGO(geneList = gen_lijst,
                      keyType = "SYMBOL",
                      OrgDb = org.Mm.eg.db,
                      ont ="BP",
                      pvalueCutoff = 0.05,
                      verbose = T)
head(as.data.frame(gsa_C57BL))

```

	ID	Description	setSize	
## GO:0031424	GO:0031424	keratinization	59	
## GO:0030216	GO:0030216	keratinocyte differentiation	152	
## GO:0033561	GO:0033561	regulation of water loss via skin	40	
## GO:0061436	GO:0061436	establishment of skin barrier	35	
## GO:0006364	GO:0006364	rRNA processing	212	
## GO:0045109	GO:0045109	intermediate filament organization	71	
##	enrichmentScore	NES pvalue p.adjust	qvalue rank	
## GO:0031424	-0.8460711	-3.514255 1e-10	1.752778e-08 1.330702e-08	1145
## GO:0030216	-0.7074820	-3.356669 1e-10	1.752778e-08 1.330702e-08	2451
## GO:0033561	-0.7897880	-3.022982 1e-10	1.752778e-08 1.330702e-08	1719
## GO:0061436	-0.7973078	-2.931304 1e-10	1.752778e-08 1.330702e-08	1485
## GO:0006364	-0.5902418	-2.915287 1e-10	1.752778e-08 1.330702e-08	8482
## GO:0045109	-0.6598070	-2.858474 1e-10	1.752778e-08 1.330702e-08	2426
##	leading_edge			

```

## GO:0031424 tags=64%, list=4%, signal=62%
## GO:0030216 tags=47%, list=9%, signal=43%
## GO:0033561 tags=52%, list=6%, signal=49%
## GO:0061436 tags=51%, list=5%, signal=49%
## GO:0006364 tags=74%, list=31%, signal=51%
## GO:0045109 tags=55%, list=9%, signal=50%
##
## GO:0031424
## GO:0030216
## GO:0033561
## GO:0061436
## GO:0006364 Mphosph10/Rpl27/Nsun3/Mettl5/Mterf4/Exosc9/Utp25/Exosc5/Fcf1/Rpusd1/Lsm6/Rps15/Rpl1711/Wdr
## GO:0045109

```

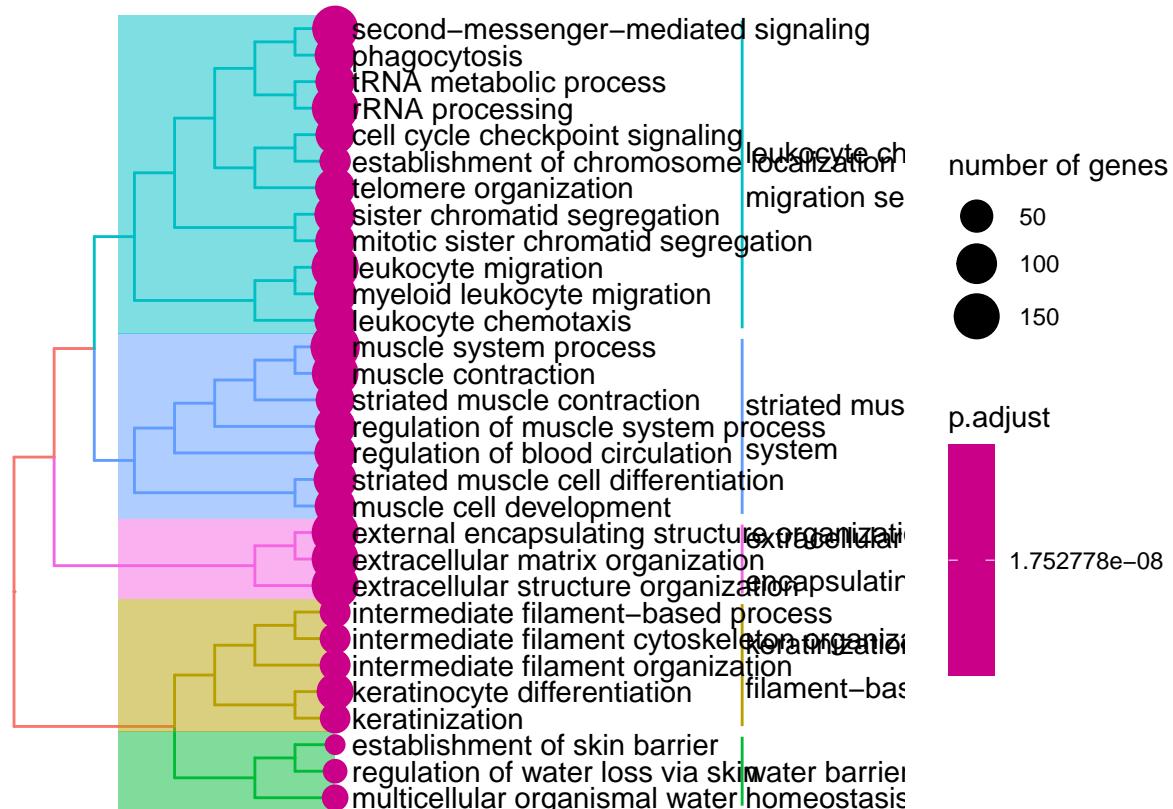
Hier zie je dus een Description met daar aan gekoppelde values die we gaan gebruiken.

Maar eerst even laten zien waar alles een beetje bij hoort

```

boompie <- pairwise_termsim(gsa_C57BL)
treeplot(boompie)

```

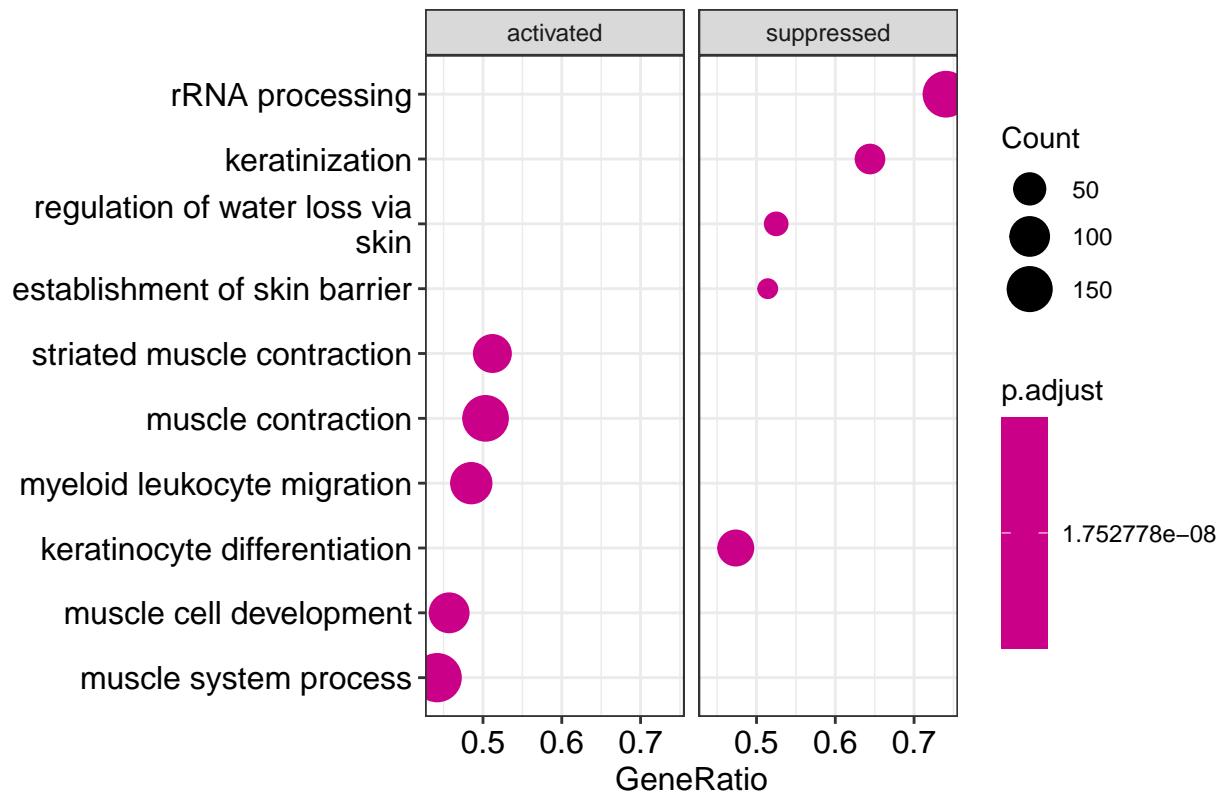


Hier maken we een dotplot waar je kan zien van 5 Description of ze worden onderdrukt of juist meer tot uiting komen met de setSize en de generatio.

```

dotplot(gsa_C57BL, split=".sign", showCategory = 5) + facet_grid(.~.sign)

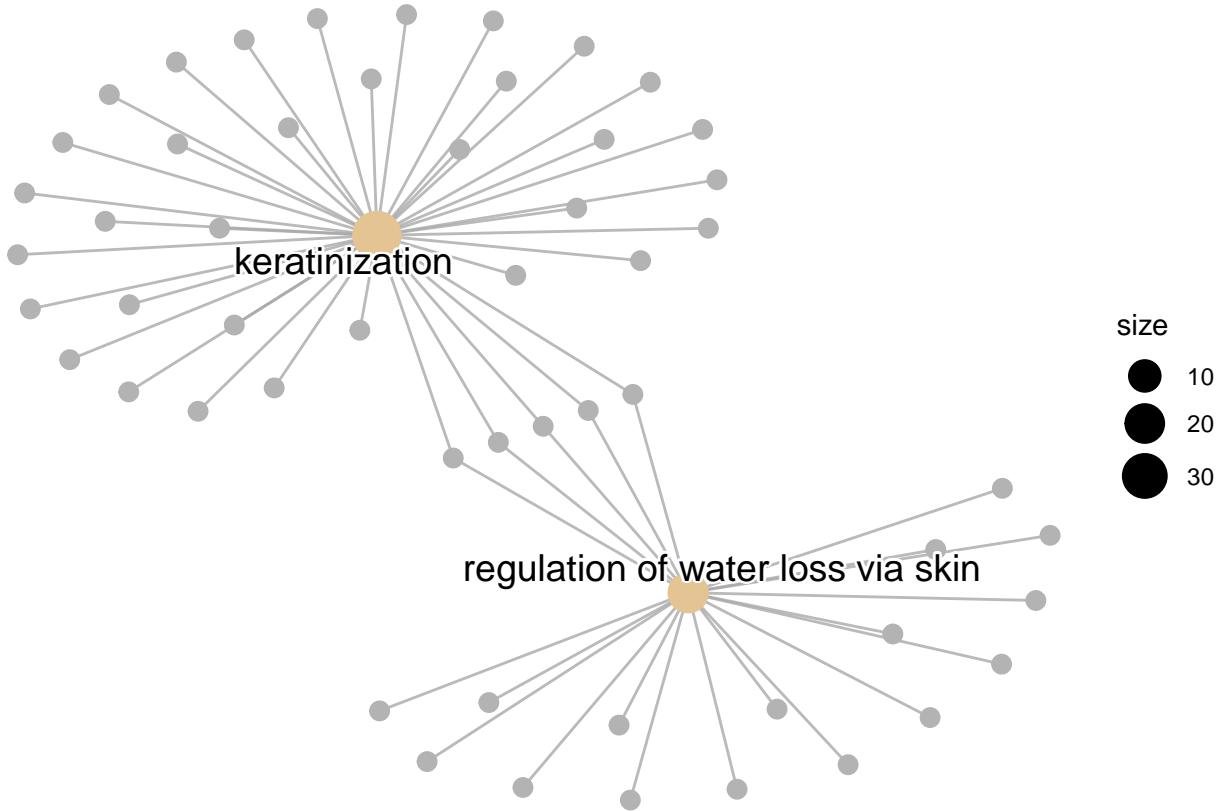
```



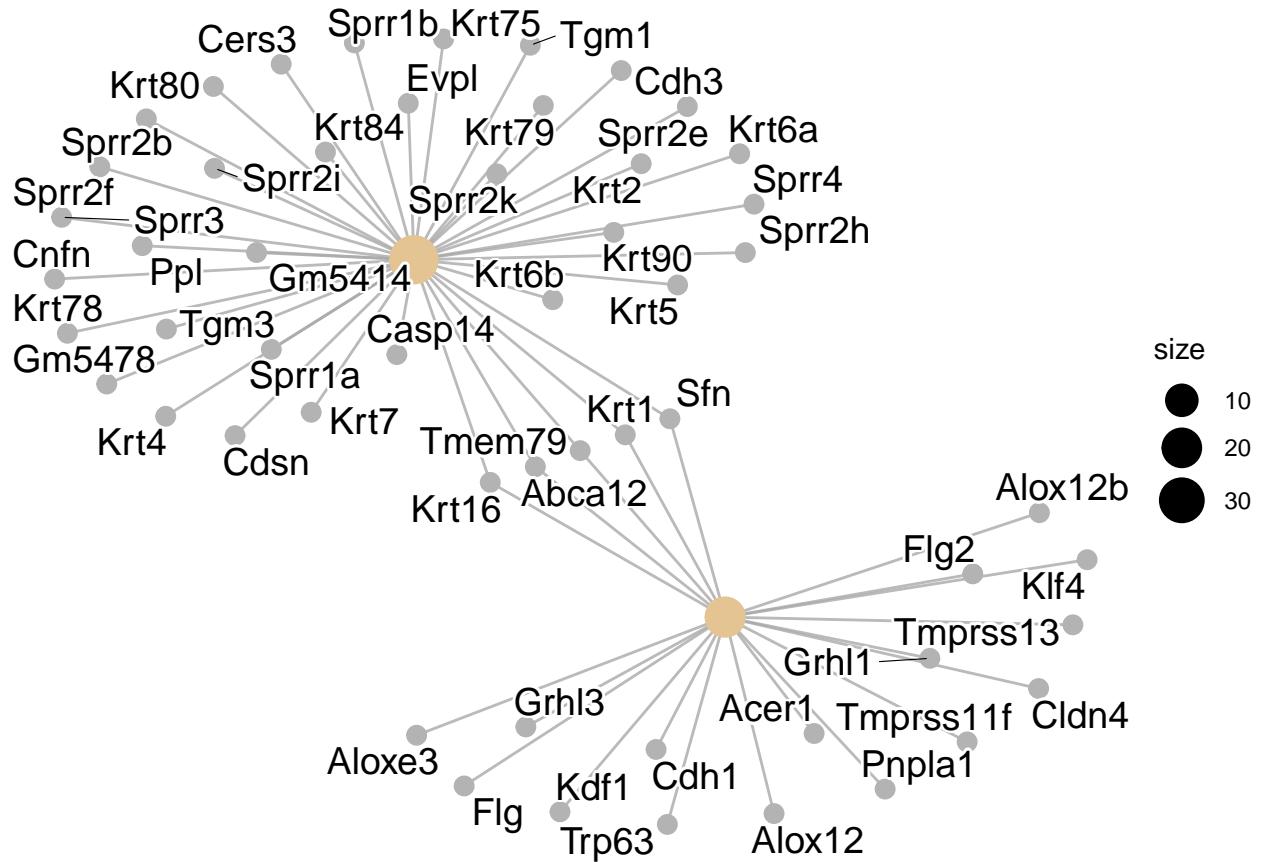
Hier zie je dus dat keratinization onderukt word en de genen er voor veel voorkomen en zie je ook dat striated muscle contraction meer tot uiting komt.

Hier onder laat ik zien welke gene er overeen komen tussen keratinization en regulation of water loss via skin

```
cnetplot(gsa_C57BL,
         node_label="category",
         showCategory = c("keratinization", "regulation of water loss via skin"))
```



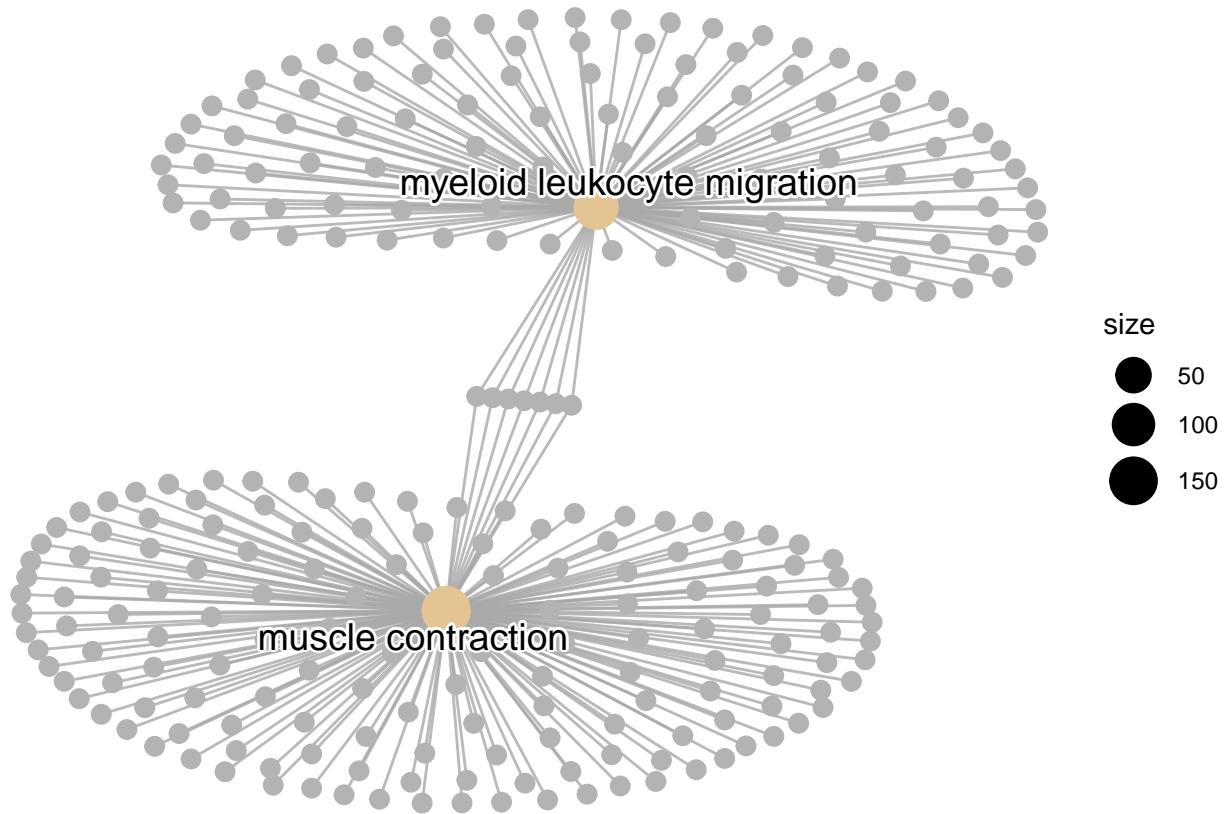
```
cnetplot(gsa_C57BL,  
         node_label="gene",  
         showCategory = c("keratinization", "regulation of water loss via skin"))
```



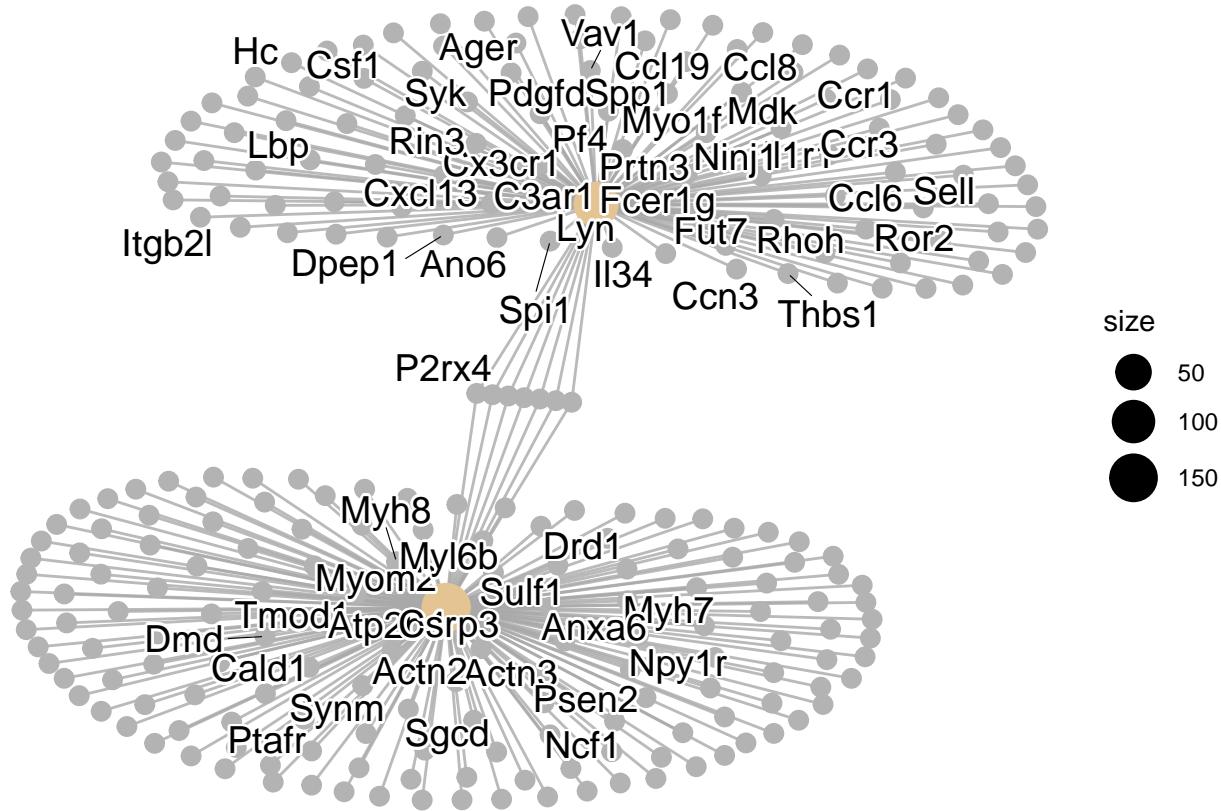
Hier uit kan je concluderen dat die genen misschien minder tot uiting komt.

hier onder laat ik dan zien de overeen komsten van myeloid leukocyte migration muscle contraction

```
cnetplot(gsa_C57BL,
          node_label="category",
          showCategory = c("myeloid leukocyte migration", "muscle contraction"))
```



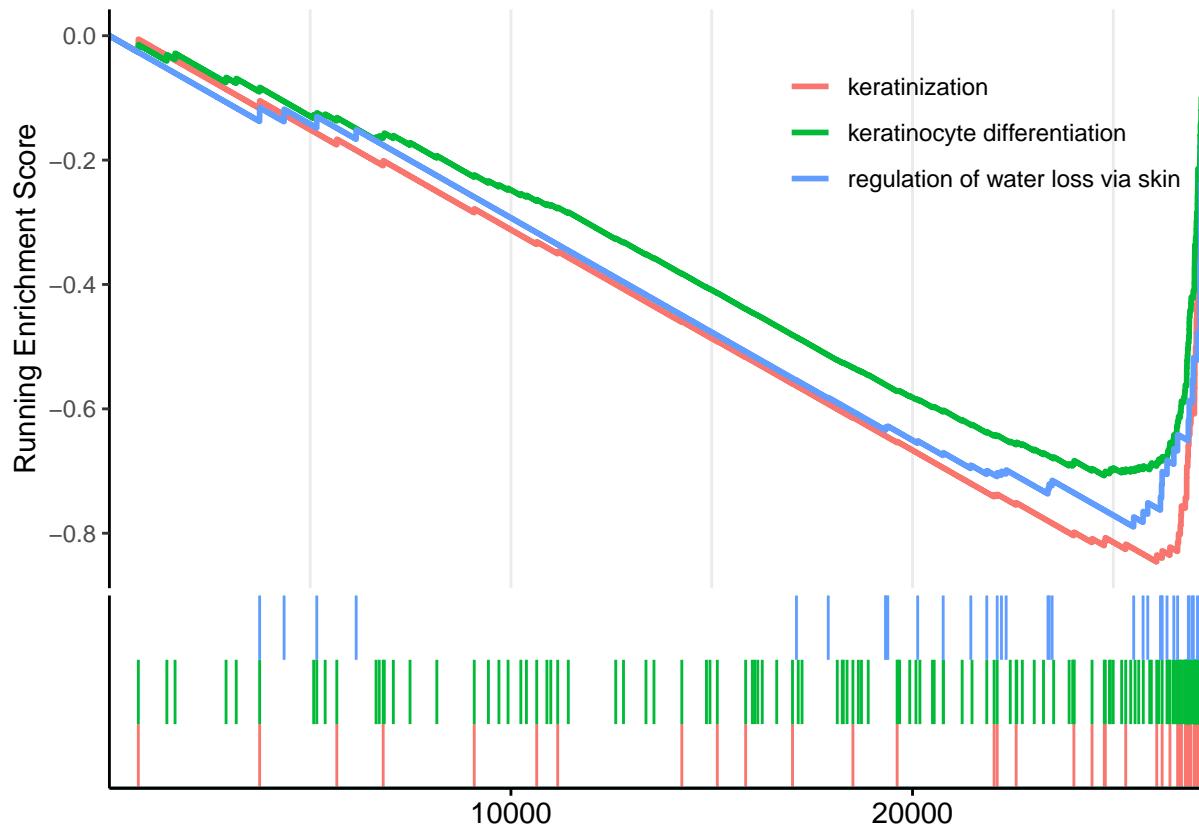
```
cnetplot(gsa_C57BL,
          node_label="gene",
          showCategory = c("myeloid leukocyte migration", "muscle contraction"))
```



Hier uit kan je concluderen dat die genen misschien meer tot uiting komt.

Hier maak ik een plot gsa van de bovenste 3 discriptions.

```
gseaplot2(gsa_C57BL, geneSetID = 1:3, subplots = 1:2)
```



In deze plot kan je zien waar de genen begeven in de genlijst en waar de ES zich begeeft. De ES begeeft zich op het meest hoogste punt of , net als in dit geval het laagst getal. Hier kan je dus zien dat keratinization de grootste afwijking heeft van de drie. De lijntjes er onder geven aan waar de genen zich begeven in de genlijst dus links boven in de genlijst en em rechts onder in de genlijst die gesorteerd is op log2FoldChange.

Volcano

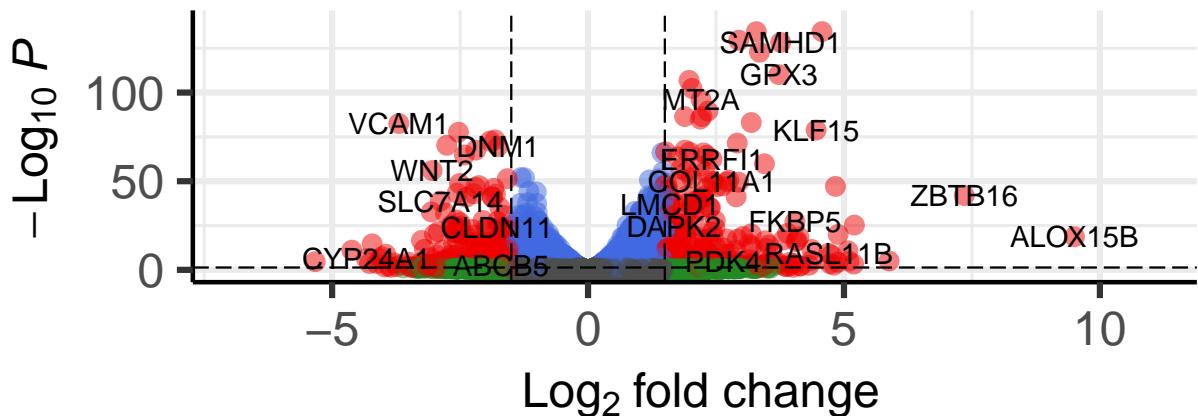
Hier zie eerste een volcano plot van NSG_bladder tumor_Vehicle tegen C57BL/6_bladder tumor_Baseline. Dit had ik uit enthousiasme al gemaakt zonder te kijken wat er echt in de data stond.

```
EnhancedVolcano(res,
  lab = rownames(res),
  x = 'log2FoldChange',
  y = 'pvalue',
  title = "behandeld tegen onbehandeld",
  pCutoff = 0.05,
  FCCcutoff = 1.5,
  pointSize = 3.0,
  labSize = 4.0,)
```

behandeld tegen onbehandeld

EnhancedVolcano

● NS ● Log₂ FC ● p-value ● p – value and log₂ FC



total = 35329 variables

Hier zie je dus dat gene- Enolb een grote invloed heeft maar een midere verandering heeft dan gen-Csn3.

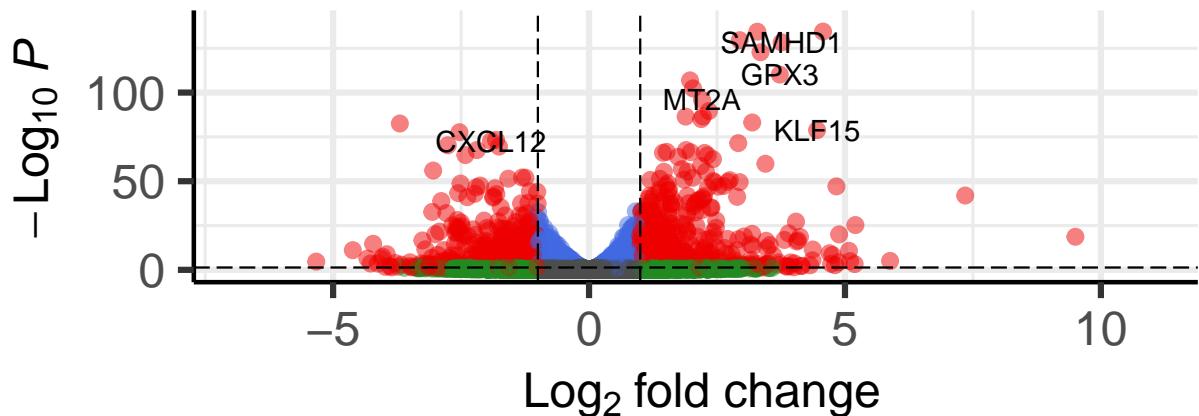
Hier onder kijk ik welke hier de hoogste p waarde hebben en zet ik die in de grsfiek het zijn de 20 hoogste genen.

```
resultaat_oder <- res[order(res$pvalue), ]
top20genen <- rownames(head(resultaat_oder, 20))
EnhancedVolcano(res,
  lab = rownames(res),
  x = 'log2FoldChange',
  y = 'pvalue',
  selectLab = top20genen,
  title = "behandeld tegen onbehandeld",
  pCutoff = 0.05,
  pointSize = 2.5,
  labSize = 4
)
```

behandeld tegen onbehandeld

EnhancedVolcano

● NS ● Log₂ FC ● p-value ● p – value and log₂ FC



total = 35329 variables

Hier kan je dus in 1 keer zien wat de stippen zijn met de hoogste p waarde en kan een makkelijker een conclusie trekken.

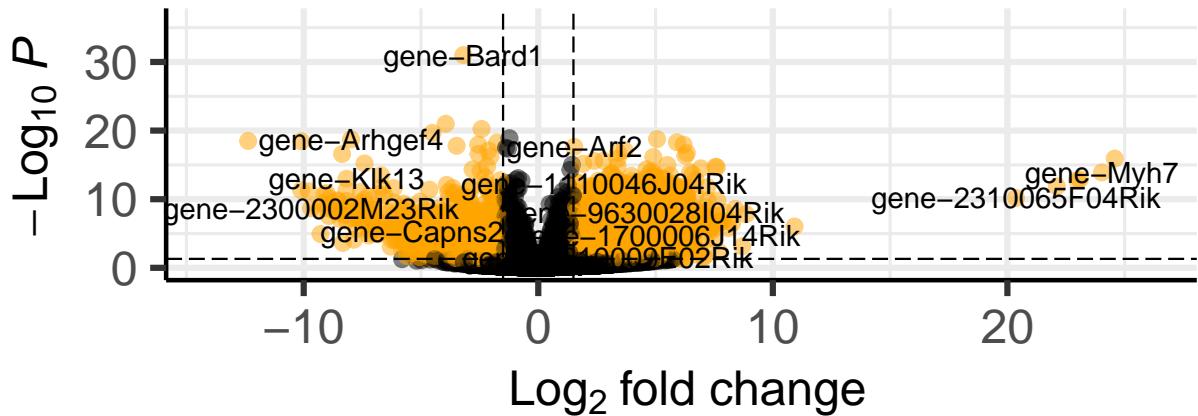
Hier onder word de goeie vergelijking gedaan wat ik zou moeten uitwerken.

```
EnhancedVolcano(resultaat_C57BL,
  lab = rownames(resultaat_C57BL),
  x = 'log2FoldChange',
  y = 'pvalue',
  title = "behandeld tegen onbehandeld C57BL ",
  pCutoff = 0.05,
  FCCcutoff = 1.5,
  pointSize = 2.5,
  labSize = 4.0,
  col=c('black', 'black', 'black', 'orange'))
```

behandeld tegen onbehandeld C57BL

EnhancedVolcano

● NS ● Log₂ FC ● p-value ○ p – value and log₂ FC



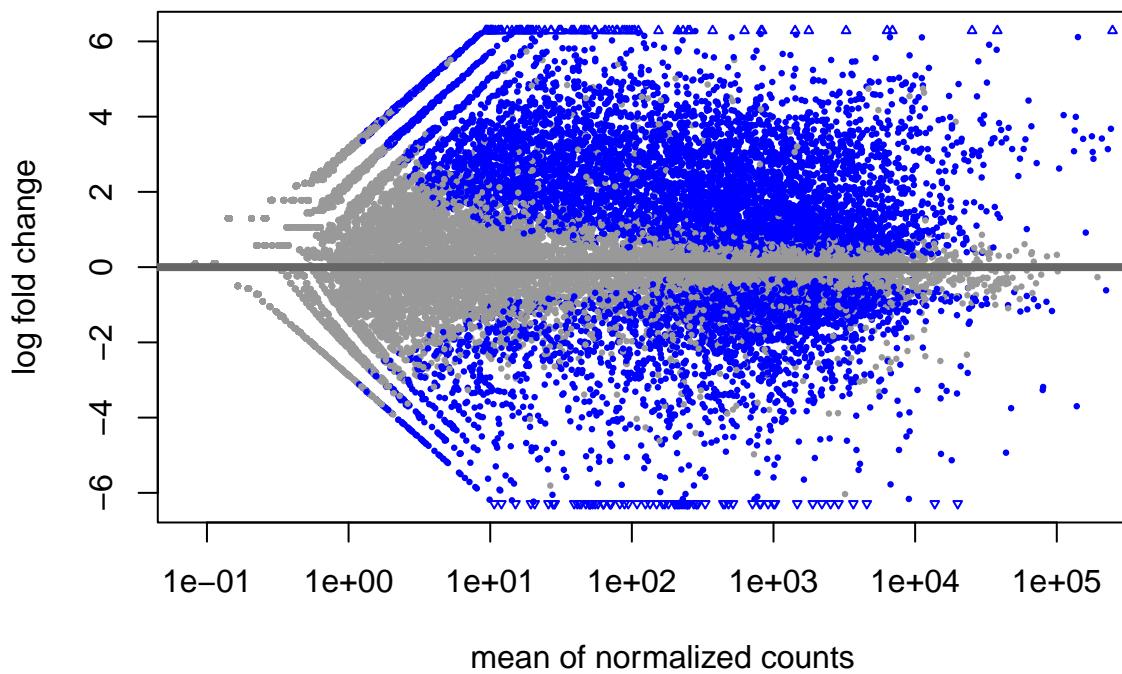
total = 41641 variables

Hier zie je dus dat er een grote verandering is bij het gen Myh7 dus is interessant om te kijken. Dit geld ook voor Bard1 want die heeft een ergere invloed er op.

MA

Hier ga ik en MA plot maken zodat je dan kan zien de gemiddelde expressie niveaus over de de expressie van de genen tussen de twee situaties.

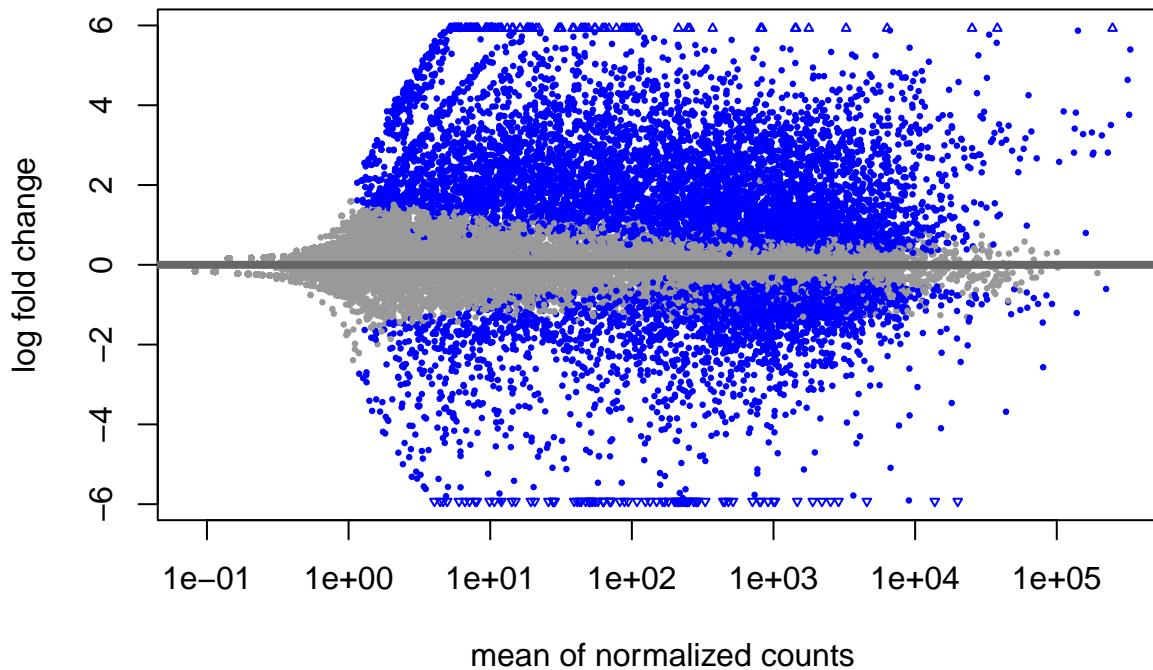
```
p3 <-plotMA(resultaat_C57BL)
```



Deze plot ziet er nog al raar uit dus ik ga even kijken wat er anders kan

Ik ben er achter gekomen dat de data eerst moet Shrinken zodat je consistent een betere uitput krijgt

```
lfcshrink_subset <- lfcShrink(C57BL_subset, coef="treatment_Entinostat_vs_Baseline", type="apeglm")
p4 <plotMA(lfcshrink_subset )
```



Dit ziet er beter uit en kan je concluderen dat er best wel een groot verschil tussen de genen die tot uiting komen bij de verschillende omstandigheden.

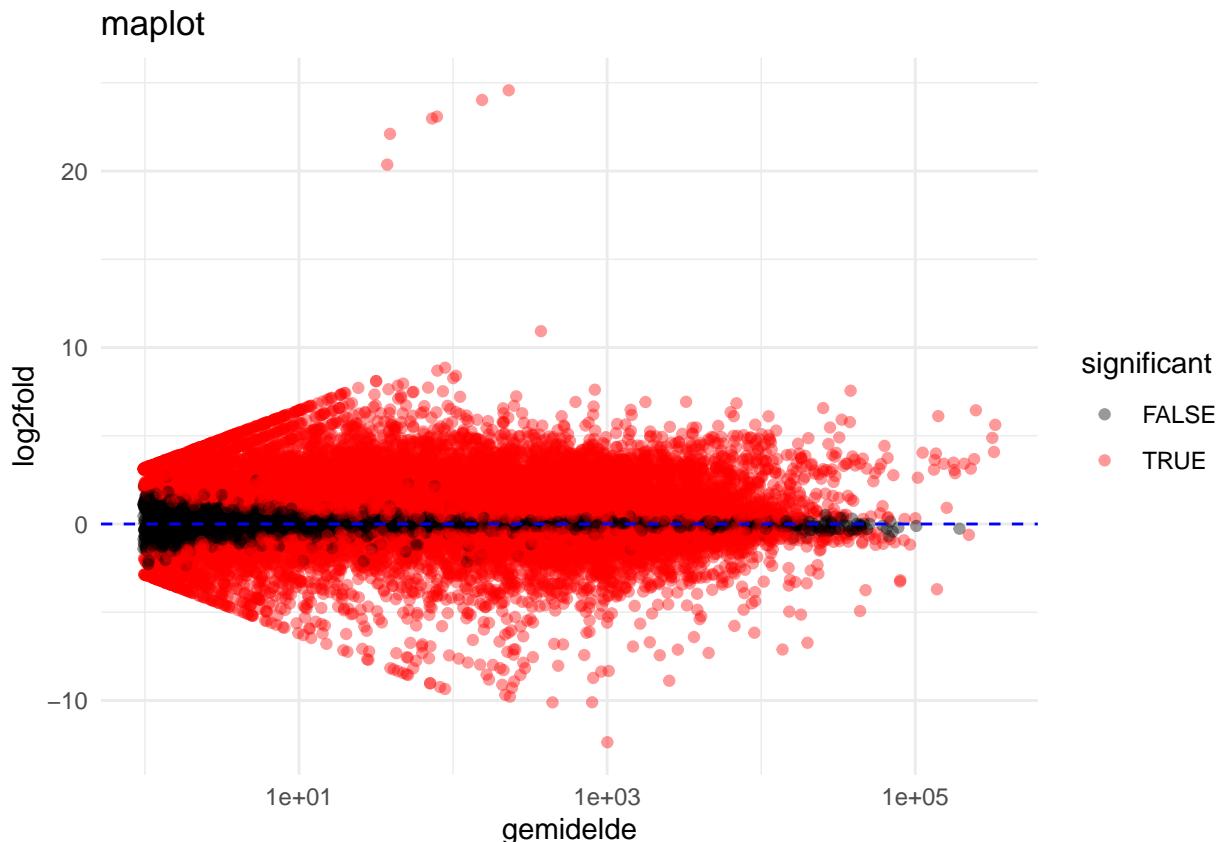
Ik ga hier nog proberen om het mooier te laten zien met ggplot

```

data_frame_resultaat_C57B <- as.data.frame(resultaat_C57BL)
data_frame_resultaat_C57B <- data_frame_resultaat_C57B[!is.na(data_frame_resultaat_C57B$padj),]
data_frame_resultaat_C57B$significant <- data_frame_resultaat_C57B$padj < 0.5

ggplot(data_frame_resultaat_C57B,
       aes(x=baseMean, y=log2FoldChange))+
  geom_point(aes(color = significant), alpha = 0.4, size = 1.5)+
  scale_x_log10()+
  geom_hline(yintercept = 0, color = "blue", linetype = "dashed")+
  labs(title = 'maplot',
       x = 'gemidelde',
       y = "log2fold")+
  scale_color_manual(values = c("TRUE" = "red", "FALSE" = "black"))+
  theme_minimal()

```



Het verschil is er niet echt en het is naar mijn idee ook wel minder duidelijk omdat het nu lijkt dat er best wel weinig niet anders is maar bij de MA plot functie juist wel te zien is.

pheatmap

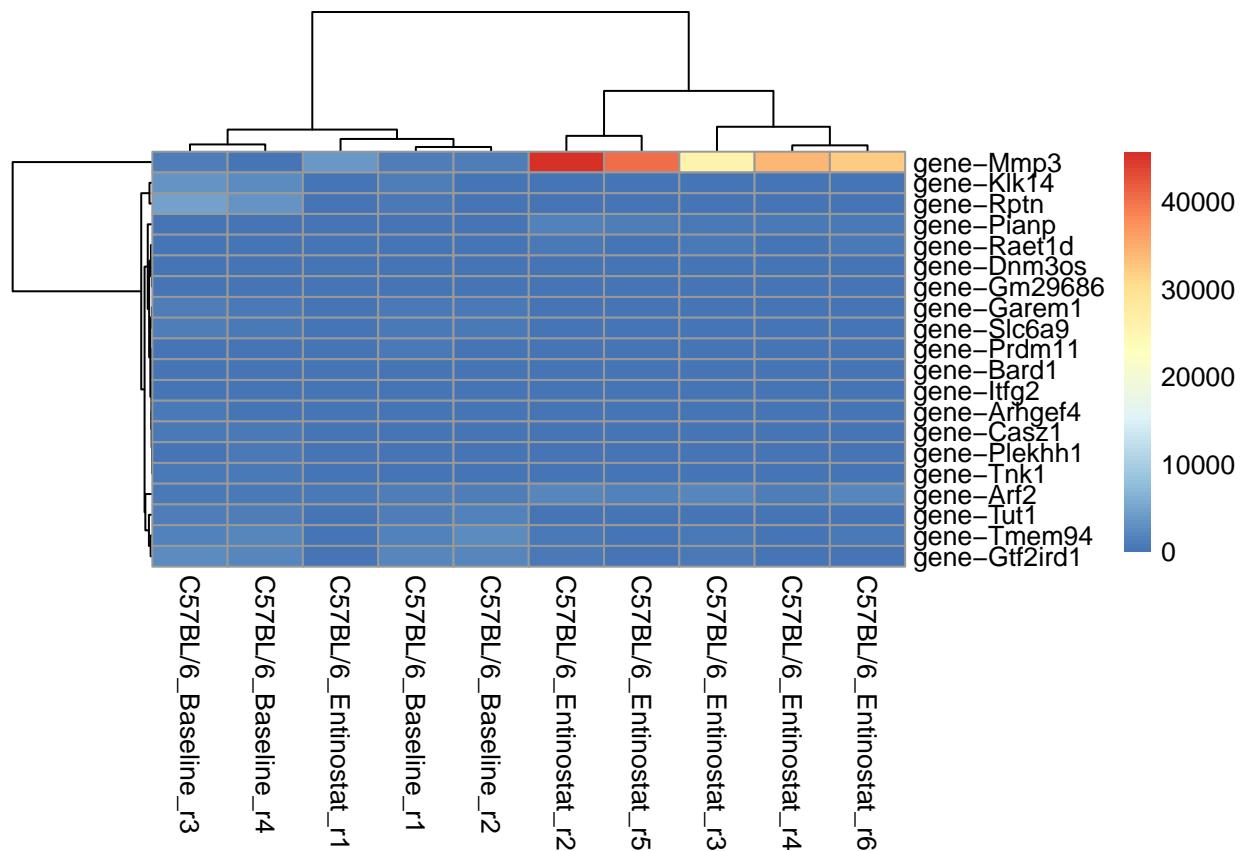
Hier ga ik een heat map maken

```

library(pheatmap)
topgene <- head(order(resultaat_C57BL$padj), 20)
#mat <- assay(dds)[topgene,]
mat_2 <- assay(C57BL_subset)[topgene,]
#pheatmap(mat, cluster_rows=TRUE, cluster_cols = TRUE)

```

```
pheatmap(mat_2, cluster_rows=TRUE, cluster_cols = TRUE)
```

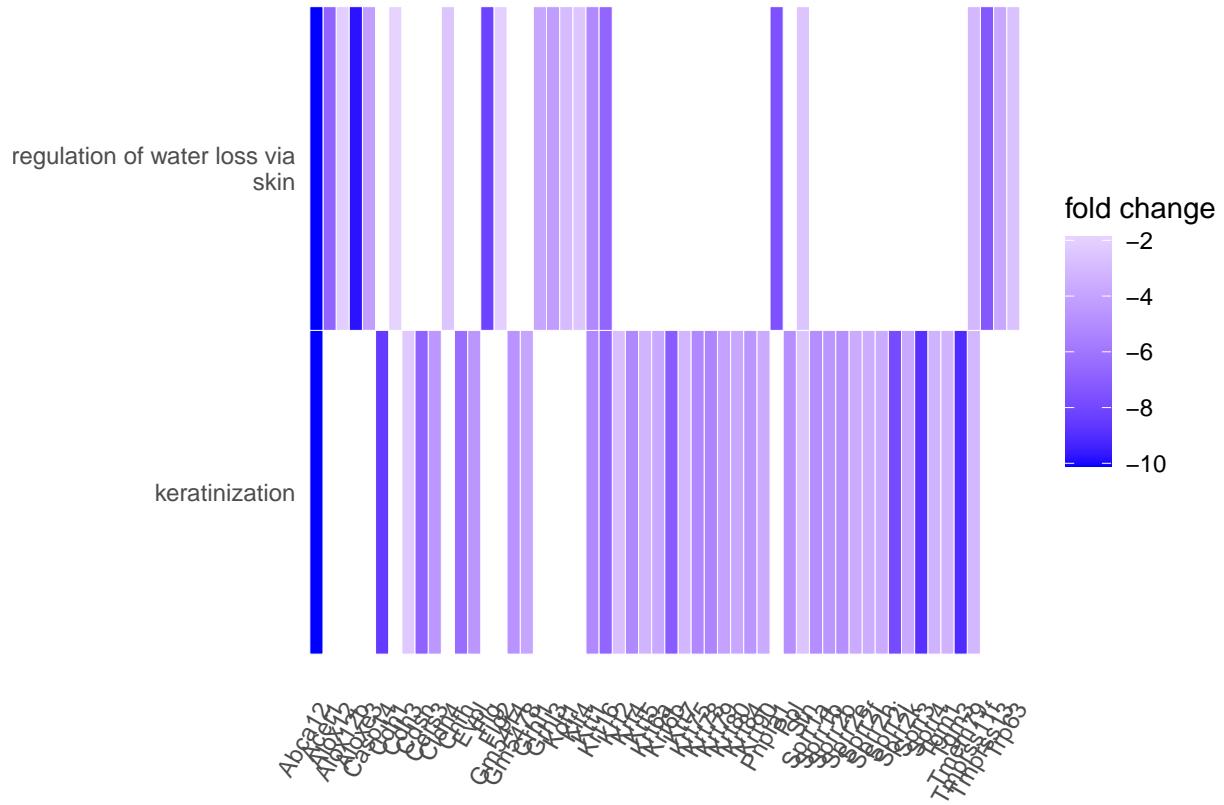


Deze vind ik nog al leijk ik ga eerst even kijken naar een andere.

Ik heb een andere manier gevonden.

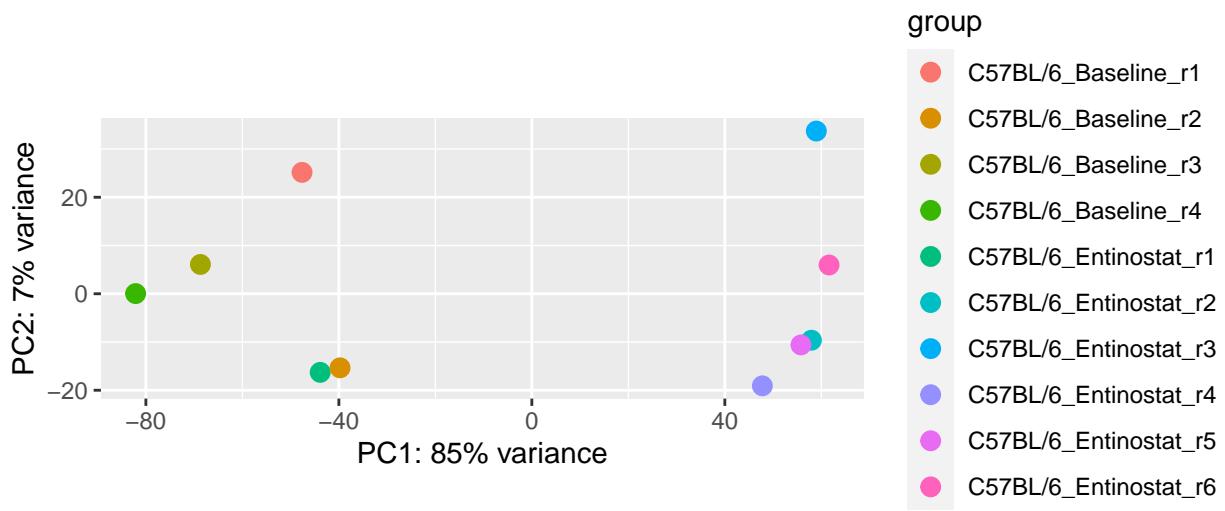
In de Heatplot kan je ook makkelijk de Discription meegeven die je graag zou willen zien dus laat ik hier weer keratinization en regulation of water loss via skin.

```
heatplot(gsa_C57BL, foldChange=gen_lijst, showCategory=c("keratinization", "regulation of water loss via skin"))
```



Hier kan je dus bij elk gen de invloed op de gegeven Disciptie. In dit geval allemaal negatieve ### PCA
Hier maak ik een pca plot om te kijken of onze data een beetje goed verdeeld is

```
vsd_C57BL <- vst(C57BL_subset, blind = FALSE)
DESeq2::plotPCA(vsd_C57BL, intgroup = "condition")
```



Op eerste oog opslag lijkt het goed totdat je bedenkt dat er van de basline er maar 4 zijn en van de entinostat er 6 zijn hierdoor is het gemiddelde van de entinostat een stuk meer naar het midden is dan de basline.

annotatie met biomart

Wij wouden gaan anoteren om dat het ons interessant lijkt maar na overleg met de docenten hebben we besloten om het niet te doen ook omdat het niet werkte dus met de code uit de meegegeven tutorial laat ik wel staan maar doen we niks meer mee.

```
col_data <- col_data %>%
  dplyr::group_by(strain, treatment) %>%
  dplyr::mutate(r_num = row_number()) %>%
  dplyr::ungroup() %>%
  mutate(condition = paste0(strain, "_", treatment, "_r", r_num))
head(col_data)

## # A tibble: 6 x 6
##   Run           source_name     strain treatment r_num condition
##   <chr>          <chr>        <chr>    <chr>    <int> <chr>
## 1 SRR12129014_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Vehicle      1 C57BL/6_~
## 2 SRR12129015_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Entinost~      1 C57BL/6_~
## 3 SRR12129016_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Vehicle      2 C57BL/6_~
## 4 SRR12129017_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Vehicle      3 C57BL/6_~
## 5 SRR12129018_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Vehicle      4 C57BL/6_~
## 6 SRR12129019_star_ReadsPerGene C57BL/6_bladde~ C57BL~ Entinost~      2 C57BL/6_~

for (i in 1:nrow(col_data)) {
  idx <- grep(col_data$Run[i], names(counts))
  names(counts)[idx] <- col_data$condition[i]
}
head(counts)

##           C57BL/6_Vehicle_r1 C57BL/6_Entinostat_r1 C57BL/6_Vehicle_r2
## MissingGeneID            5859                  4387                7820
## gene-0610005C13Rik         3                      0                  0
## gene-0610006L08Rik         0                      1                  0
## gene-0610009E02Rik         0                      1                  1
## gene-0610009L18Rik        53                     29                 25
## gene-0610010K14Rik       1024                     790                881
##           C57BL/6_Vehicle_r3 C57BL/6_Vehicle_r4 C57BL/6_Entinostat_r2
## MissingGeneID            8707                  3405                3821
## gene-0610005C13Rik         0                      0                  0
## gene-0610006L08Rik         0                      0                  2
## gene-0610009E02Rik         1                      0                  1
## gene-0610009L18Rik        93                     43                 34
## gene-0610010K14Rik       1408                     897                523
##           C57BL/6_Entinostat_r3 C57BL/6_Vehicle_r5 C57BL/6_Baseline_r1
## MissingGeneID            3777                  6823                4673
## gene-0610005C13Rik         1                      0                  0
## gene-0610006L08Rik         0                      0                  0
## gene-0610009E02Rik         1                      4                  0
## gene-0610009L18Rik        44                     93                 25
## gene-0610010K14Rik       492                    1304                975
##           C57BL/6_Baseline_r2 C57BL/6_Entinostat_r4
## MissingGeneID            7162                  2160
## gene-0610005C13Rik         0                      1
## gene-0610006L08Rik         0                      1
## gene-0610009E02Rik         0                      1
```

```

## gene-0610009L18Rik      55          15
## gene-0610010K14Rik     1390         419
## C57BL/6_Baseline_r3 C57BL/6_Entinostat_r5
## MissingGeneID          5096        2718
## gene-0610005C13Rik      0           0
## gene-0610006L08Rik      0           0
## gene-0610009E02Rik      0           4
## gene-0610009L18Rik      24          49
## gene-0610010K14Rik     972          676
## C57BL/6_Entinostat_r6 C57BL/6_Vehicle_r6 C57BL/6_Baseline_r4
## MissingGeneID          3870        9562        5163
## gene-0610005C13Rik      0           0           0
## gene-0610006L08Rik      0           0           0
## gene-0610009E02Rik      6           0           2
## gene-0610009L18Rik      34          43          45
## gene-0610010K14Rik     431          964         876
## NSG_Baseline_r1 NSG_Baseline_r2 NSG_Baseline_r3
## MissingGeneID          8453        8045        5803
## gene-0610005C13Rik      4           1           2
## gene-0610006L08Rik      0           0           0
## gene-0610009E02Rik      7           4           7
## gene-0610009L18Rik      38          47          31
## gene-0610010K14Rik     967          1298        664
## NSG_Baseline_r4 NSG_Baseline_r5 NSG_Entinostat_r1
## MissingGeneID          6777        8639        6767
## gene-0610005C13Rik      1           3           1
## gene-0610006L08Rik      1           0           0
## gene-0610009E02Rik      8           10          6
## gene-0610009L18Rik      45          48          95
## gene-0610010K14Rik    1001          1336        1470
## NSG_Entinostat_r2 NSG_Entinostat_r3 NSG_Entinostat_r4
## MissingGeneID          9366        9204        3295
## gene-0610005C13Rik      6           2           0
## gene-0610006L08Rik      0           0           0
## gene-0610009E02Rik      5           6           0
## gene-0610009L18Rik     139          124          64
## gene-0610010K14Rik    1528          1339        387
## NSG_Entinostat_r5 NSG_Vehicle_r1 NSG_Vehicle_r2
## MissingGeneID          12470       5393        7122
## gene-0610005C13Rik      2           2           1
## gene-0610006L08Rik      1           0           0
## gene-0610009E02Rik     13           6          11
## gene-0610009L18Rik     113          41          50
## gene-0610010K14Rik    1393          735        1145
## NSG_Vehicle_r3 NSG_Vehicle_r4 NSG_Vehicle_r5
## MissingGeneID          8495        8854        7730
## gene-0610005C13Rik      2           4           2
## gene-0610006L08Rik      0           0           1
## gene-0610009E02Rik      5           8           4
## gene-0610009L18Rik     122          150          160
## gene-0610010K14Rik    1618          1689        1511
counts$Ensembl <- mapIds(x = org.Mm.eg.db,
                           keys=gsub("gene-", "", row.names(counts)),

```

```

            column="ENSEMBL",
            keytype="SYMBOL",
            multiVals="first")

library(biomaRt)
ensembl=useMart("ENSEMBL_MART_ENSEMBL", host="https://www.ensembl.org")
ensembl <- useMart("ensembl")
mart.datasets <- listDatasets(ensembl)
ensembl <- useDataset('mmusculus_gene_ensembl', mart = ensembl)
filters <- listFilters(ensembl)
attributes <- listAttributes(ensembl)

# Set the 'attributes' values
attrs.get <- c("ensembl_gene_id", "chromosome_name",
              "start_position", "end_position", "description")

Perform a biomaRt query using 'getBM'
results <- getBM(attributes = attrs.get,
                  filters = "ensembl_gene_id",
                  values = counts$Ensembl[12:15],
                  mart = ensembl, verbose = TRUE)

results$gene_length <- abs(results$end_position - results$start_position)

merge(x = counts, y = results, by.x = 'Ensembl', by.y = 'ensembl_gene_id')

```