Genes regulados por p53-SET

Ulises Gaspar

Se importan librerías:

```
library(limma)
library(edgeR)
library("pheatmap")
library(ggplot2)
library(gplots)
```

```
## Warning: package ’gplots’ was built under R version 4.2.3
```

```
##
## Attaching package: ’gplots’
```

```
## The following object is masked from ’package:stats’:
##
##     lowess
```

Se cargan los datos de fc:

```
setwd("/Users/hecto/Downloads")
fc <- read.table("counts.txt", header = T)
```

Cargar información a edgeR:

```
samples <- factor(c("WildType", "WildType", "WildTypeKnockDown", "WildTypeKnockDown", "p53KnockOut", "p5

DGEList_p53 = DGEList(counts=fc[,3:10], group=samples, genes=fc[,1:2])
dim(DGEList_p53)
```

```
## [1] 55385     8
```

Se eliminan genes con baja expresión

```
keep = rowSums(cpm(DGEList_p53)>1) >= 3
DGEList_p53 = DGEList_p53[keep,]
dim(DGEList_p53)
```

```
## [1] 8476    8
```

Se genera la matriz de diseño

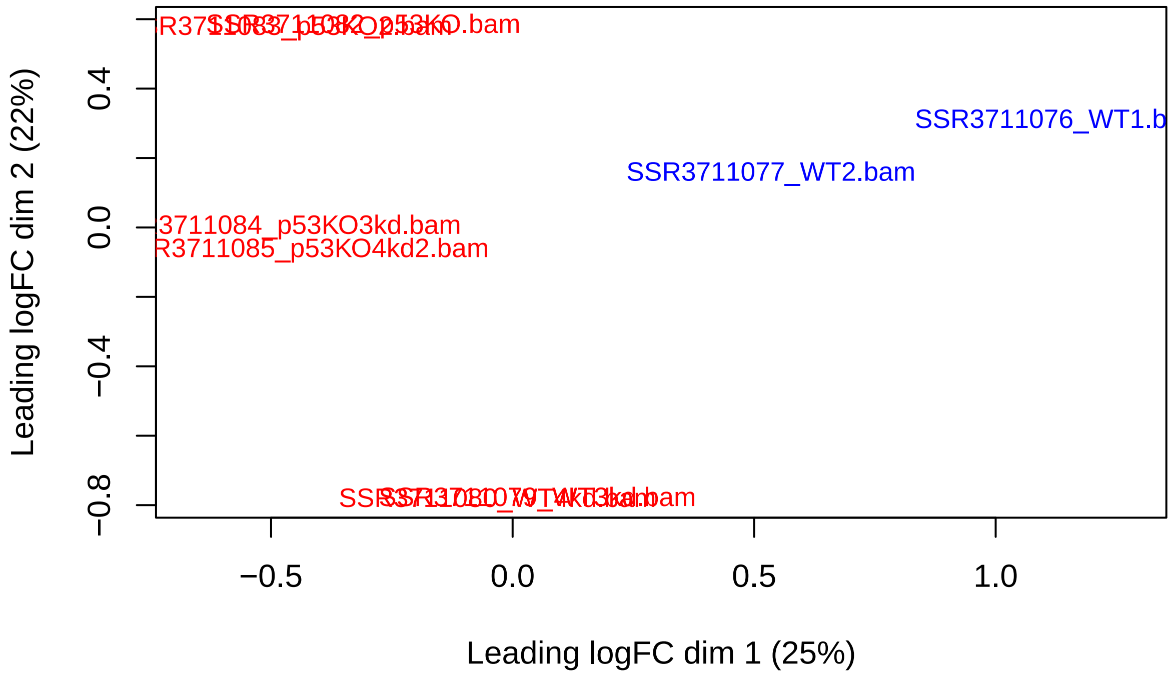
```
design <- model.matrix(~0+samples)
colnames(design) <- levels(samples)
```

Factores de normalización y variación de las muestras:

```
DGEList_p53 = calcNormFactors(DGEList_p53)
DGEList_p53 = estimateDisp(DGEList_p53, design = design)
```

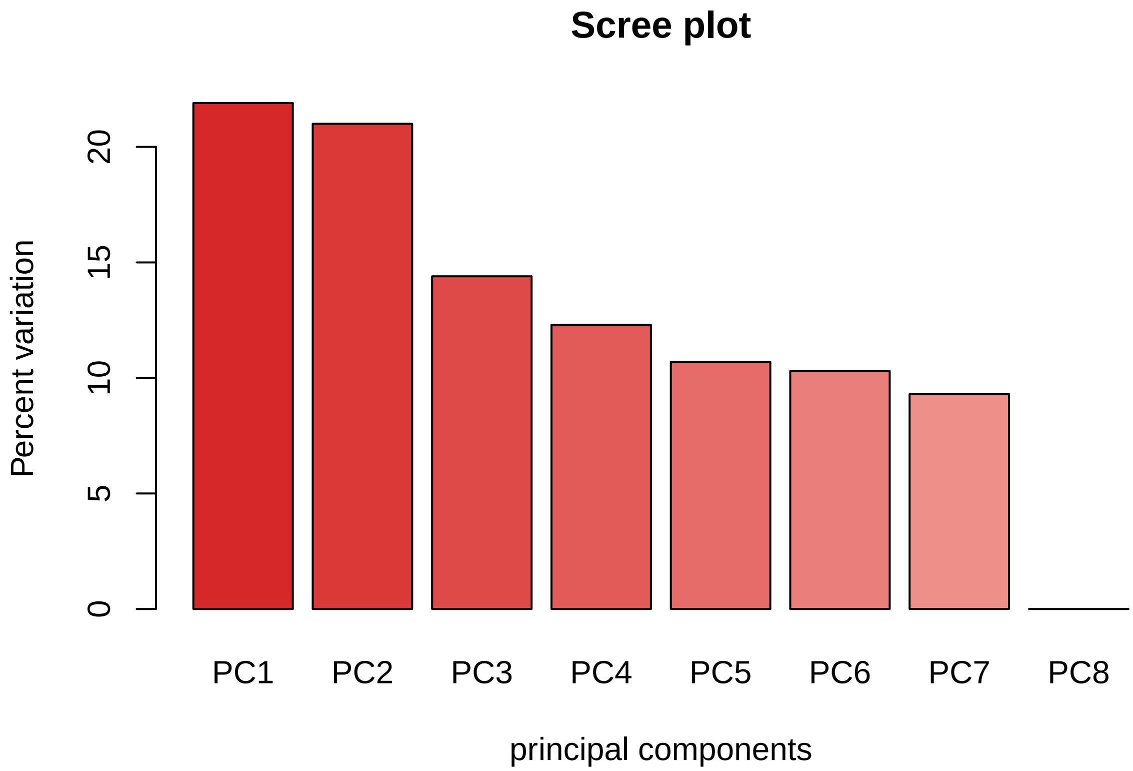
Se genera un PCA. Se consideran aquellos con p53 funcional como control (solo las primeras dos muestras)

```
colors = c("blue", "blue", "red", "red", "red", "red", "red", "red")
plotMDS(DGEList_p53, cex = 0.8, col = colors, )
```



Se observa como los datos se agrupan en su grupo correspondiente.

```
PCA_log2CPM = prcomp(t(cpm(DGEList_p53, log = T)), center = T, scale = T)
PCA_perc_var = round(((PCA_log2CPM$sdev^2/sum(PCA_log2CPM$sdev^2))*100), 1)

color_range <- colorRampPalette(c("#d62828", "#FED4CC"))

barplot(PCA_perc_var, names = colnames(PCA_log2CPM$x), main = "Scree plot", col = color_range(11), xlab
```



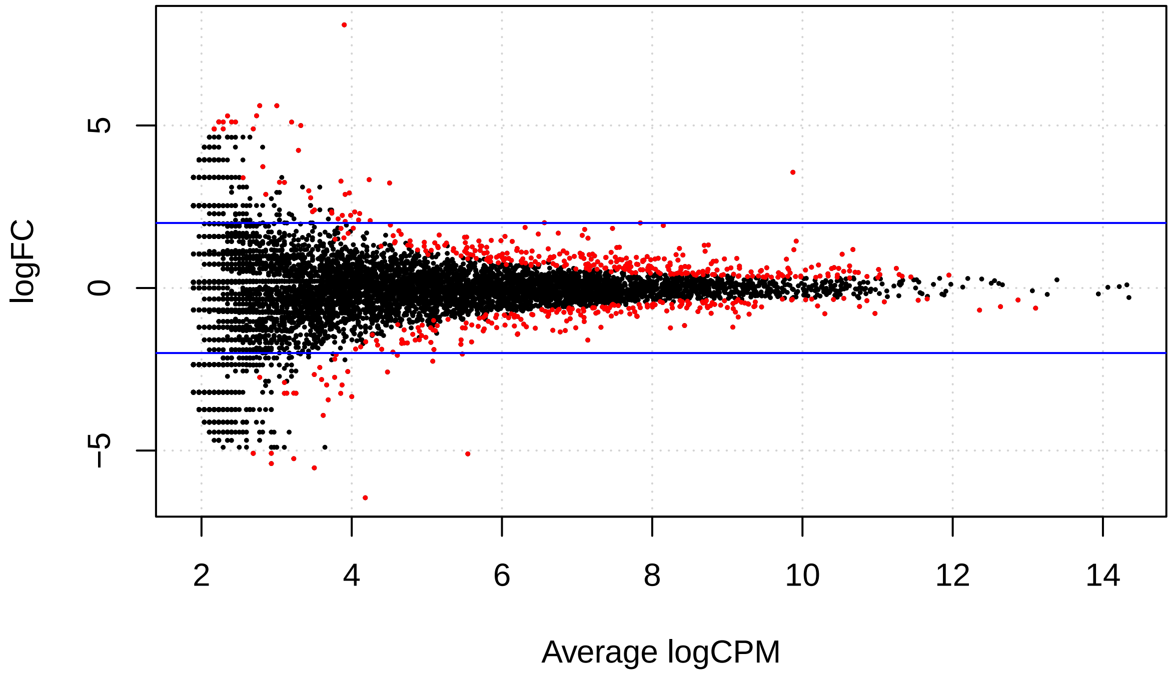
Los 2 primeros componentes parecen explicar mucha de la variabilidad (~20% cada uno), pero aún así no es tan fuerte.

Se procede a calcular los genes expresados diferencialmente Como tenemos 4 grupos, y solo 1 tiene p53 funcional, consideré que solo 2 conrastes son necesarios, replicando el paper: CRISPR Ctr (Wt) vs CRISPR Crt/SET kd, y CRISPR Ctr/SET kd vs CRISPR p55 KO/SET kd

```
fit <- glmFit(DGEList_p53, design)

#contrastWTSETkd_v_WT <- glmLRT(fit, contrast = makeContrasts(WildTypeKnockDown - WildType, levels = de
contrastWT_v_WTSETkd <- glmLRT(fit, contrast = makeContrasts(WildType - WildTypeKnockDown, levels = des

#contrastWTSETkd_v_p53KOSETkd <- glmLRT(fit, contrast = makeContrasts(WildTypeKnockDown - p53KnockOutKn
contrastp53KOSETkd_v_WTSETkd <- glmLRT(fit, contrast = makeContrasts(p53KnockOutKnockDown - WildTypeKno
```

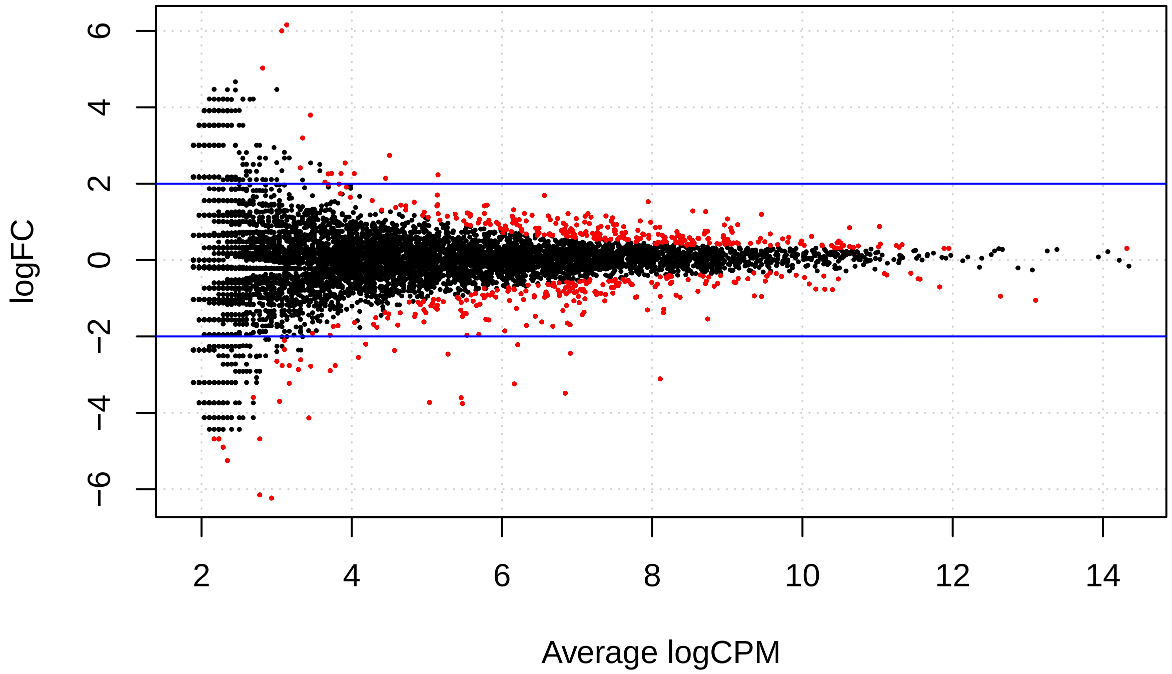
Se visualizan los genes del contraste entre CRISPR ctr vs CRISPR ctr/SET kd (aquellos con):

```
dt_significant1 <- decideTestsDGE(contrastWT_v_WTSETkd, adjust.method = "BH", p.value = 0.05)
vctr_names_sig1 <- rownames(DGEList_p53)[as.logical(dt_significant1)]
plotSmear(contrastWT_v_WTSETkd, de.tags = vctr_names_sig1)
abline(h = c(-2,2), col = "blue")
```



Se visualizan los genes del contraste entre CRISPR Ctr/SETkd vs CRISPR p55 KO/SET kd:

```
dt_significant2 <- decideTestsDGE(contrastp53KOSETkd_v_WTSETkd, adjust.method = "BH", p.value = 0.05)
vctr_names_sig2 <- rownames(DGEList_p53)[as.logical(dt_significant2)]
plotSmear(contrastp53KOSETkd_v_WTSETkd, de.tags = vctr_names_sig2)
abline(h = c(-2,2), col = "blue")
```



Heatmap herárquico de genes expresados diferencialmente:

```
vctr_sig <- as.logical(decideTestsDGE(contrastWT_v_WTSETkd, adjust.method = "BH", p.value = 0.000005))
vctr_sig<- vctr_sig | as.logical(decideTestsDGE(contrastp53KOSETkd_v_WTSETkd, adjust.method = "BH", p.va
vctr_names_hcl <- rownames(DGEList_p53)[vctr_sig]
length(vctr_names_hcl)
```

```
## [1] 133
```

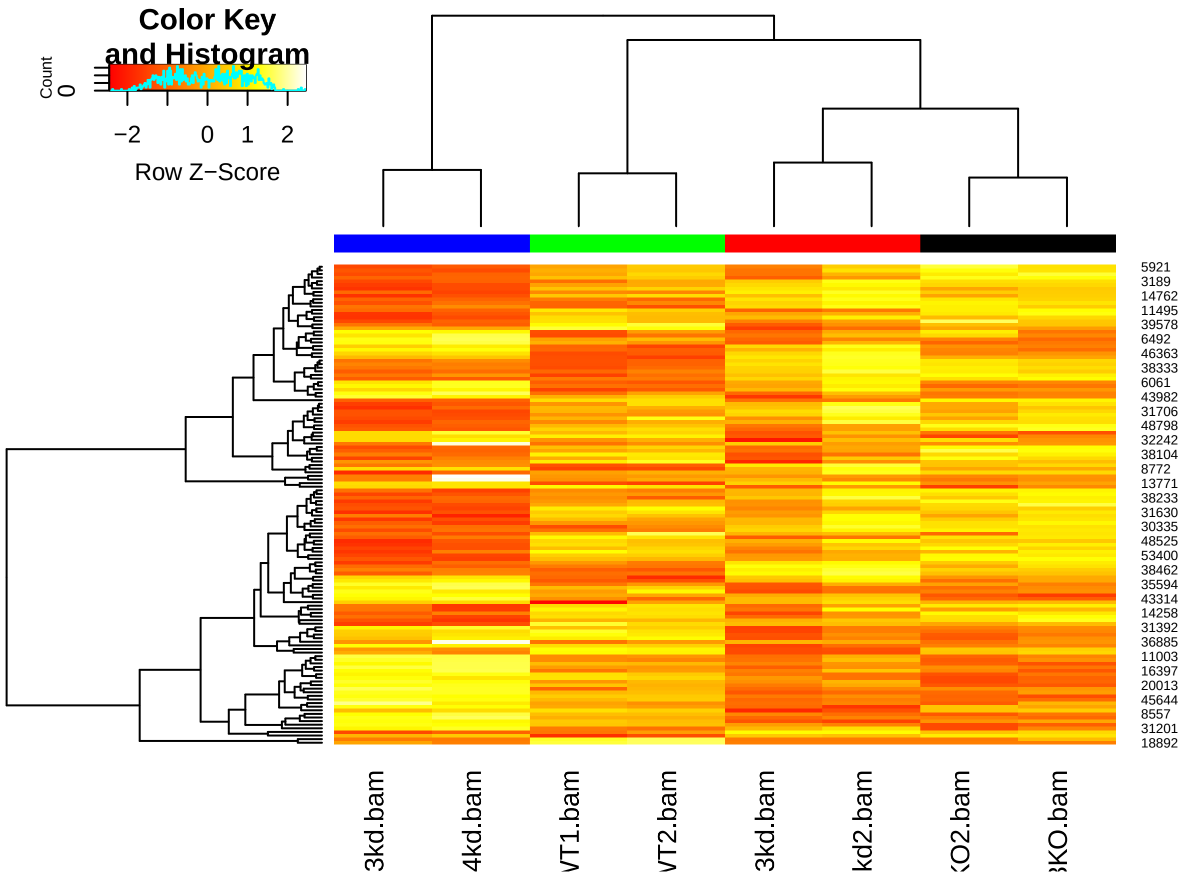
```
vctr_sig_WT_v_WTSETkd <- as.logical(decideTestsDGE(contrastWT_v_WTSETkd, adjust.method = "BH", p.value =
vctr_names_1 <- rownames(DGEList_p53)
vctr_sig_p53KOSETkd_v_WTSETkd <- as.logical(decideTestsDGE(contrastp53KOSETkd_v_WTSETkd, adjust.method =
vctr_names_2 <- rownames(DGEList_p53)
```

```
mtrx_significant <- DGEList_p53$counts[vctr_names_hcl, ]

Genes_significant_1 <- DGEList_p53$genes[vctr_sig_WT_v_WTSETkd, ]
Genes_significant_2 <- DGEList_p53$genes[vctr_sig_p53KOSETkd_v_WTSETkd, ]

vctr_colors =  as.factor(c("black", "red", "green", "blue"))
```

```
vctr_sample_colors <- as.character(vctr_colors[as.numeric(DGEList_p53$samples$group)])
vctr_sample_colors
```

```
## [1] "green" "green" "blue"  "blue"  "black" "black" "red"   "red"
```

```
heatmap.2(log2(mtrx_significant + 1), ColSideColors = vctr_sample_colors, key = TRUE, trace = "none", c
```



Se procede a hacer el análisis de ontología de genes:

```
library(clusterProfiler)
```

```
##
```

```
## clusterProfiler v4.6.2  For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X Bo, and G Yu.
```

```
##
## Attaching package: 'clusterProfiler'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(org.Mm.eg.db)
```

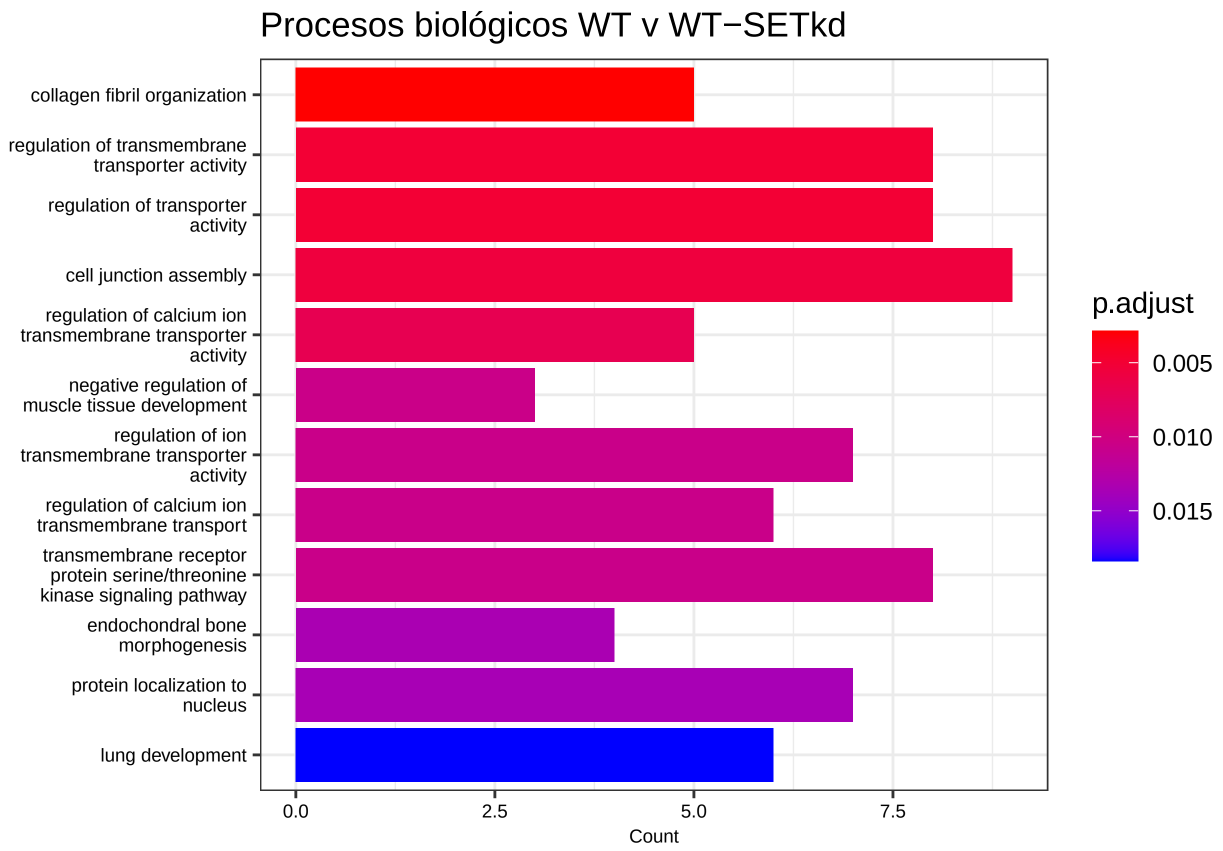
```
## Loading required package: AnnotationDbi
## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following object is masked from 'package:limma':
##
##     plotMA

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:clusterProfiler':
##
##     rename

## The following object is masked from 'package:gplots':
##
##     space

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
##
## Attaching package: 'IRanges'

## The following object is masked from 'package:clusterProfiler':
##
##     slice

## The following object is masked from 'package:grDevices':
##
##     windows

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:clusterProfiler':
##
##     select

##
```

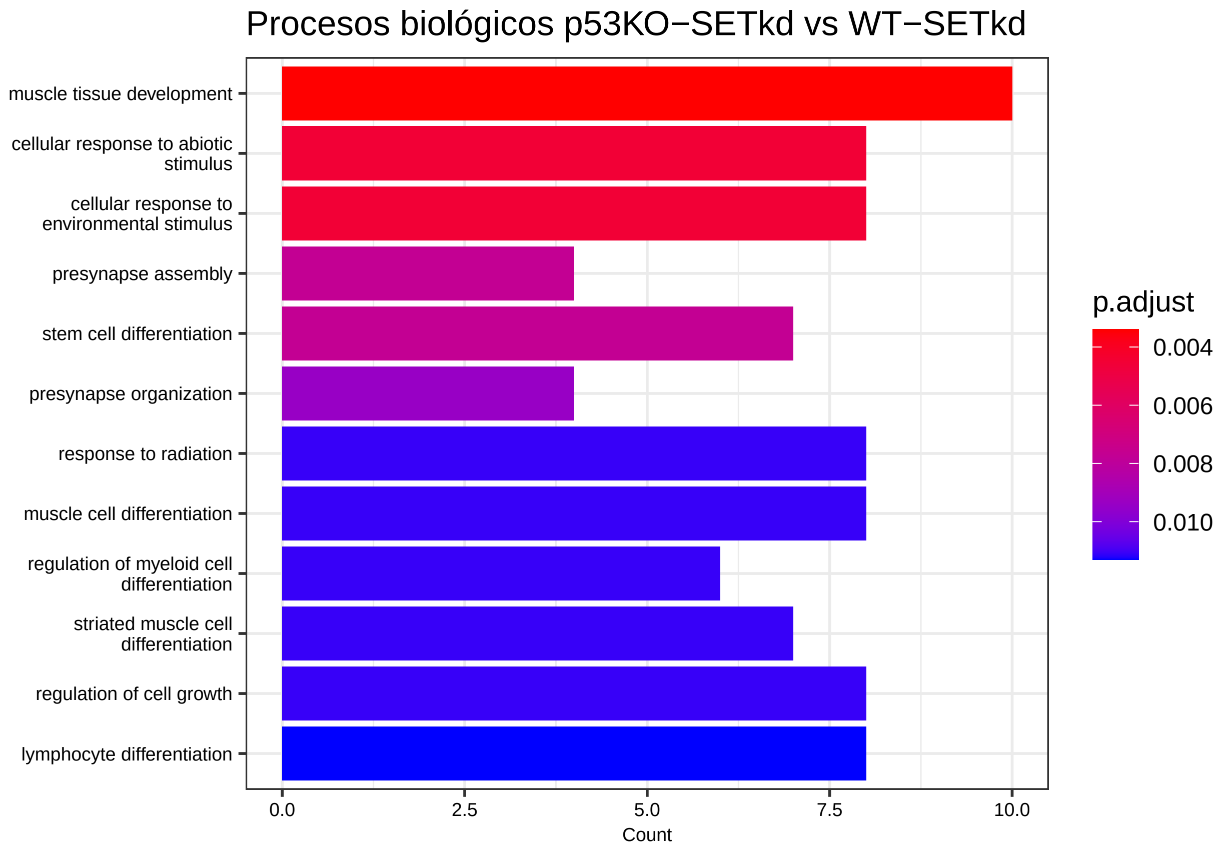
```
library(AnnotationDbi)
```

Proceso biológico

```
GO_results1 = enrichGO(gene = Genes_significant_1$GeneID, OrgDb = org.Mm.eg.db, keyType = "ENSEMBL", on
plot(barplot(GO_results1, showCategory = 12, font.size = 7, title = "Procesos biológicos WT v WT-SETkd")
```

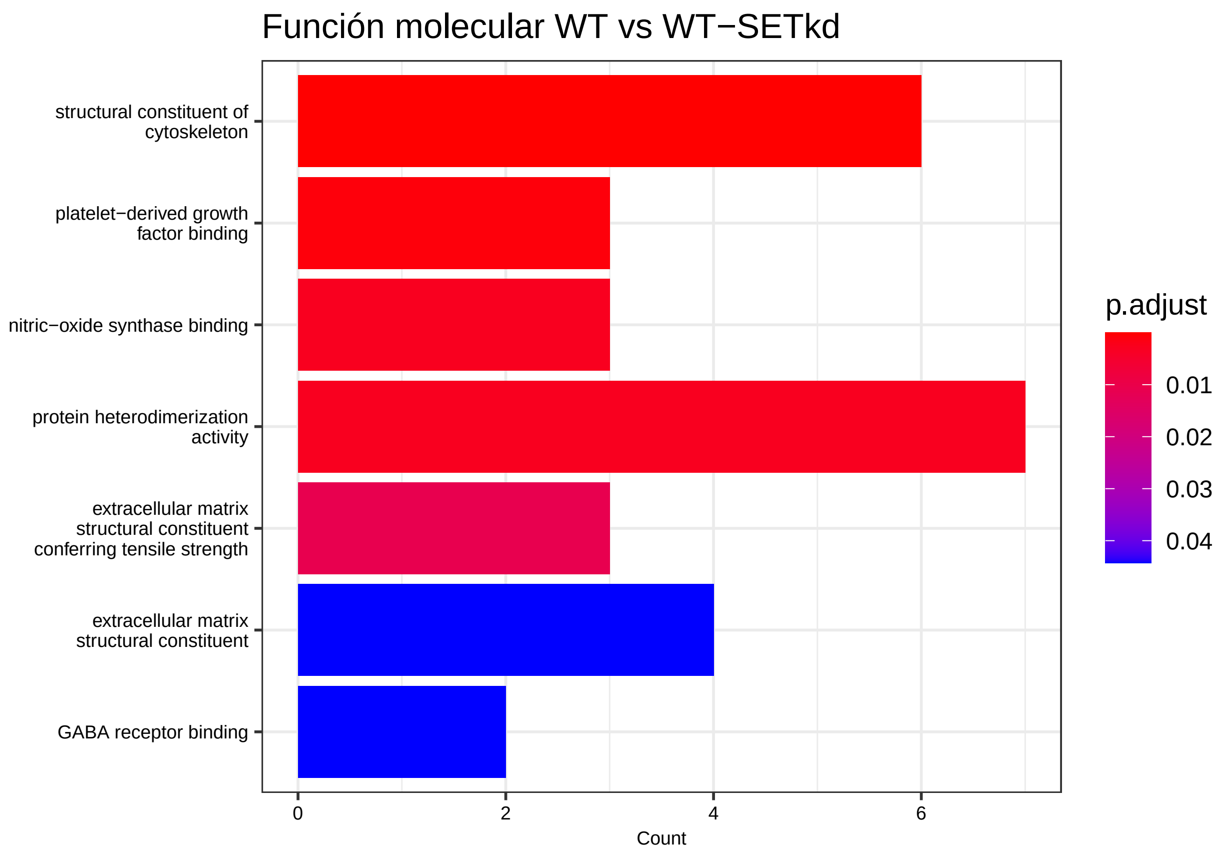


```
GO_results2 = enrichGO(gene = Genes_significant_2$GeneID, OrgDb = org.Mm.eg.db, keyType = "ENSEMBL", on
plot(barplot(GO_results2, showCategory = 12, font.size = 7, title = "Procesos biológicos p53KO-SETkd vs
```

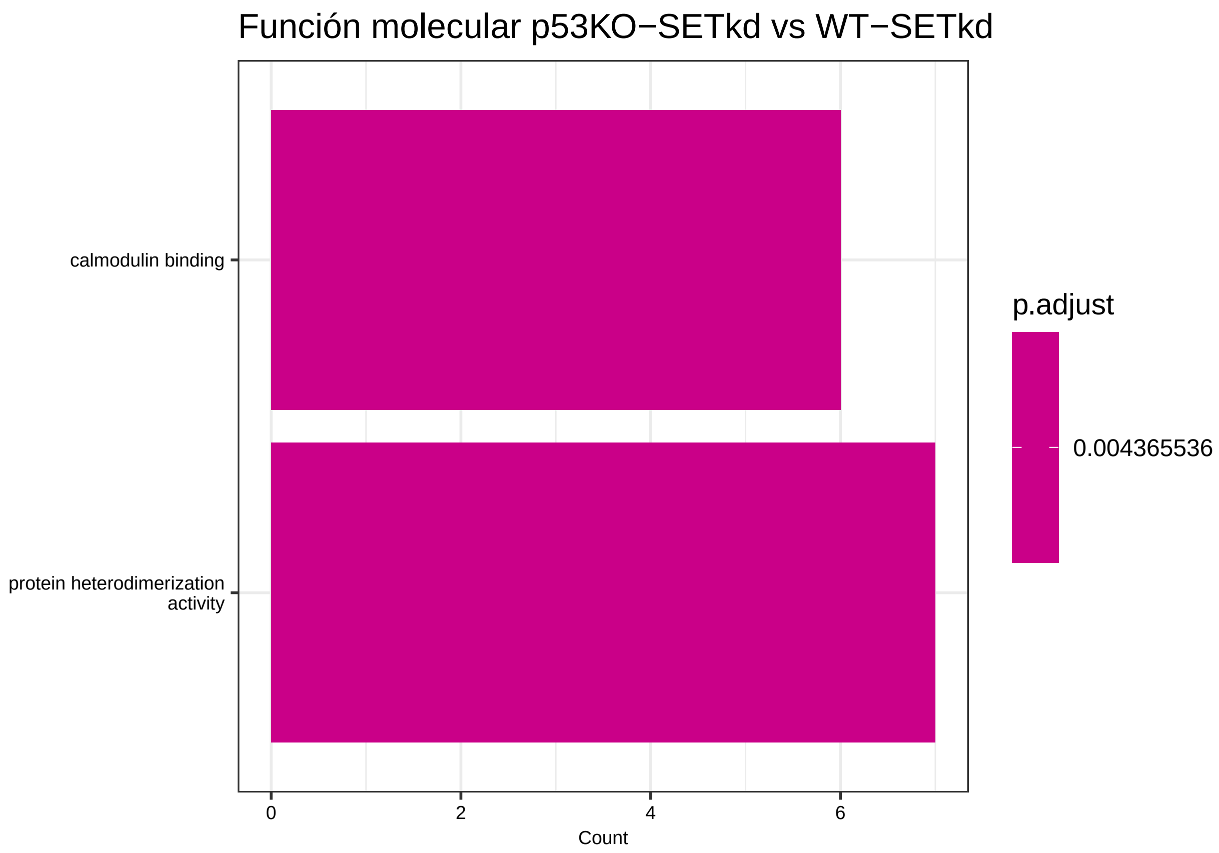


Función molecular:

```
GO_resultsMF1 = enrichGO(gene = Genes_significant_1$GeneID, OrgDb = org.Mm.eg.db, keyType = "ENSEMBL", 
plot(barplot(GO_resultsMF1, showCategory = 12, font.size = 7, title = "Función molecular WT vs WT-SETkd
```

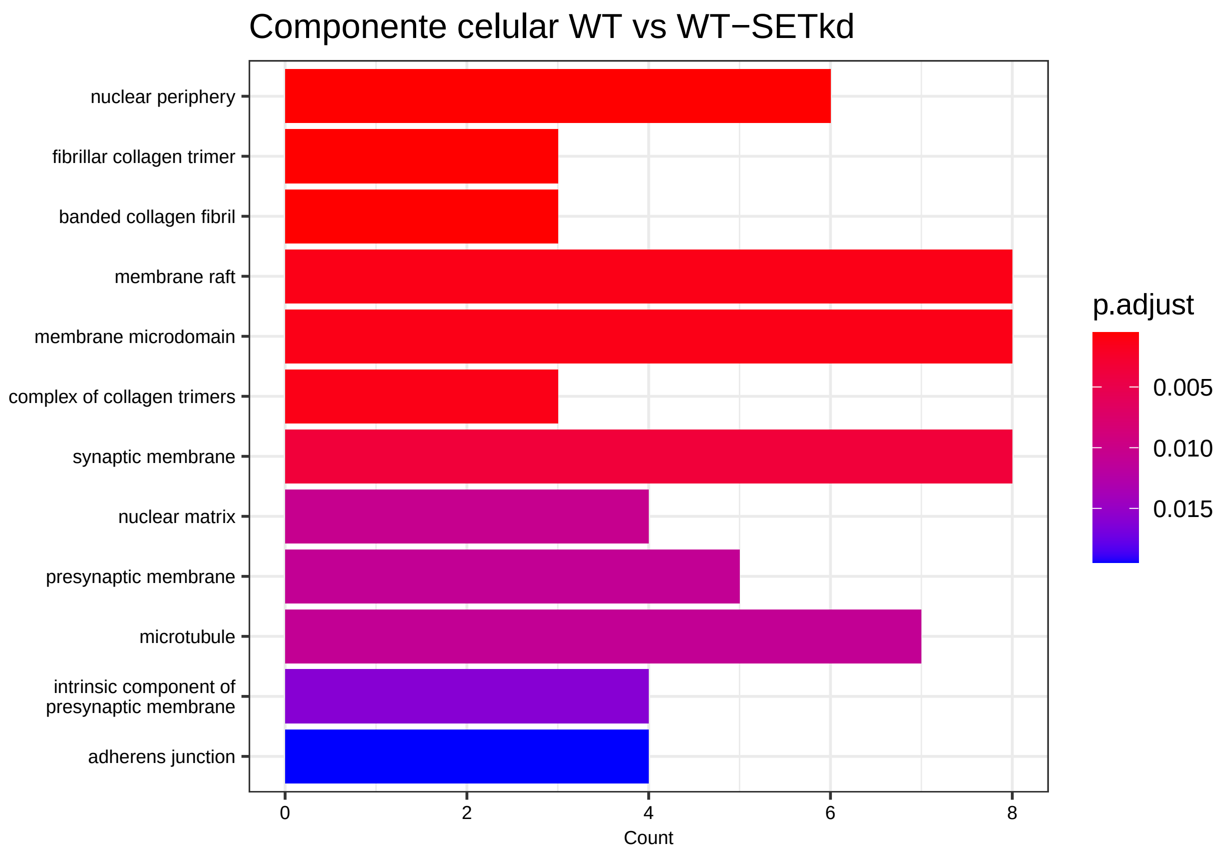


```
GO_resultsMF2 = enrichGO(gene = Genes_significant_2$GeneID, OrgDb = org.Mm.eg.db, keyType = "ENSEMBL",
plot(barplot(GO_resultsMF2, showCategory = 12, font.size = 7, title = "Función molecular p53KO-SETkd vs
```

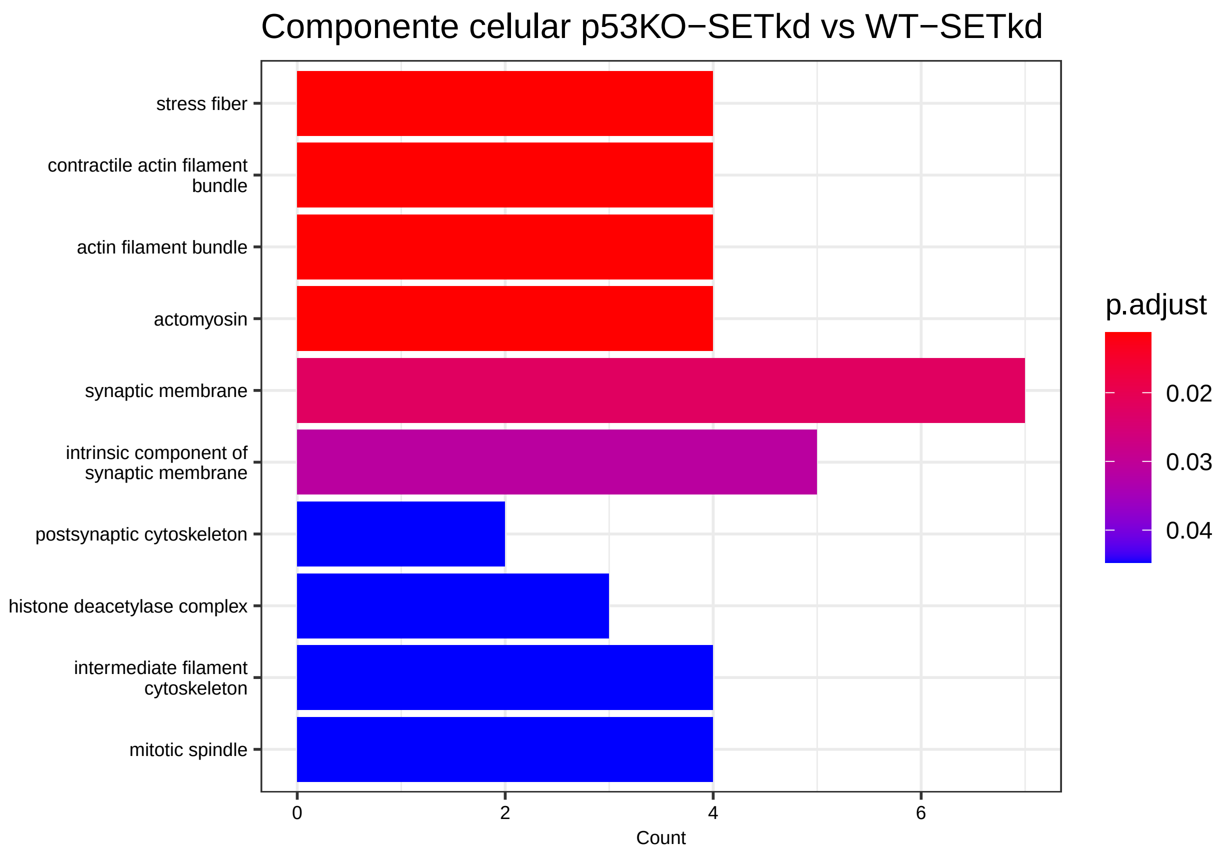


Componente celular

```
GO_resultsCC1 = enrichGO(gene = Genes_significant_1$GeneID, OrgDb = org.Mm.eg.db, keyType = "ENSEMBL", o
plot(barplot(GO_resultsCC1, showCategory = 12, font.size = 7, title = "Componente celular WT vs WT-SETkd
```



```
GO_resultsCC2 = enrichGO(gene = Genes_significant_2$GeneID, OrgDb = org.Mm.eg.db, keyType = "ENSEMBL", o
plot(barplot(GO_resultsCC2, showCategory = 12, font.size = 7, title = "Componente celular p53KO-SETkd vs
```


Componente celular p53KO−SETkd vs WT−SETkd
stress fiber
contractile actin filament bundle
actin filament bundle
actomyosin
synaptic membrane
intrinsic component of synaptic membrane
postsynaptic cytoskeleton
histone deacetylase complex
intermediate filament cytoskeleton
mitotic spindle
0
2
4
6
Count
p.adjust
0.02
0.03
0.04