EW Sarcoma

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
rm(list = ls())
```

```r
# Load libraries
library(pheatmap)
library(recount)
library(DESeq2)
library(tidyverse)
library(ggpubr)
library(EnhancedVolcano)
library(EnsDb.Hsapiens.v86)
library(msigdbr)
library(clusterProfiler)
library(enrichplot)
library(cowplot)
library(org.Hs.eg.db)
library(gridExtra)
library(ggridges)
```

```r
#The same as going the recount2 and download them manually
project_info <- abstract_search(query = "Ewing sarcoma")
```

```r
# Download the study data
download_study("SRP015989")
```

```
## 2026-01-25 18:52:22.096893 downloading file rse_gene.Rdata to SRP015989
```

```r
# Load the data
load("SRP015989/rse_gene.Rdata")
```

```r
# Fix the colData to give a column with the appropriate groups
rse_gene$condition <- c(rep("shCTR", 3), rep("shEF1", 4))
```

```r
rownames(rse_gene) <- gsub("\\..*$", "", rownames(rse_gene))
head(rownames(rse_gene))
```

```
## [1] "ENSG00000000003" "ENSG00000000005" "ENSG00000000419" "ENSG00000000457"
## [5] "ENSG00000000460" "ENSG00000000938"
```

```r
# Create DESeq2 data
dds <- DESeqDataSet(rse_gene, design = ~condition)
```

```
## converting counts to integer mode

## Warning in DESeqDataSet(rse_gene, design = ~condition): 45 duplicate rownames
## were renamed by adding numbers

## Warning in DESeqDataSet(rse_gene, design = ~condition): some variables in
## design formula are characters, converting to factors
```

```r
#filter out low count genes across all samples
dds <- dds[rowSums(counts(dds)) > 10, ]
```

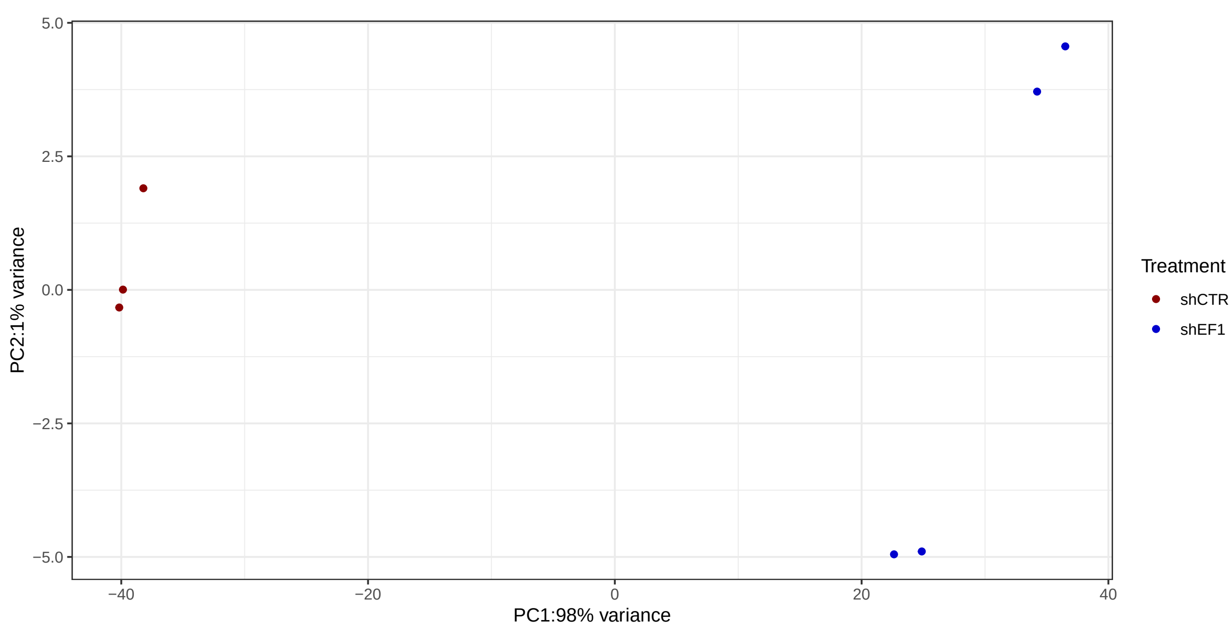
```r
# relevel so that the control experiment is the base condition
dds$condition <- relevel(dds$condition, ref = "shCTR")
```

```r
# transform data into log scale to make the variance more uniform
logdds <- rlog(dds)
logdata <- as.data.frame(assay(logdds))
```

```r
#PCA is performed to confirm replicates cluster correctly, detect batch effects or outliers, and ensure
PCA_data <- plotPCA(logdds, intgroup = "condition", returnData = TRUE)
```

```
## using ntop=500 top features by variance
```

```r
percentVar <- round(100 * attr(PCA_data, "percentVar"))
ggplot(PCA_data, aes(x = PC1, y = PC2, color = condition)) + geom_point() + theme_bw() +
  xlab(paste0("PC1:",percentVar[1], "% variance")) +
  ylab(paste0("PC2:",percentVar[2], "% variance")) +
  scale_color_manual(name = "Treatment", values = c("red4","blue3"))
```



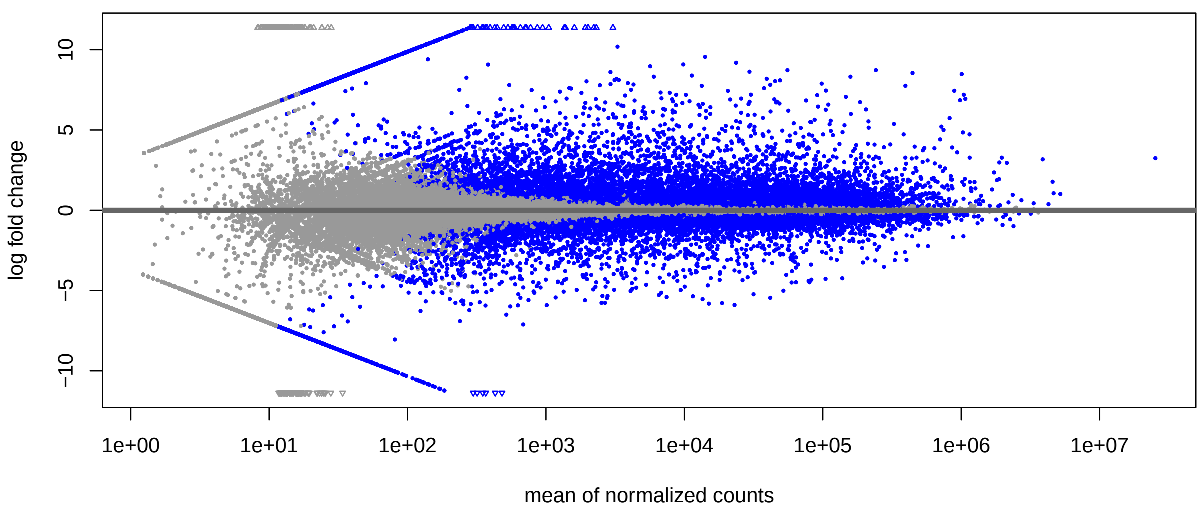
This PCA plot shows a very strong separation between shCTR and shEF1 along PC1, indicating that EF1 knockdown causes a dominant transcriptional change. Replicates cluster tightly within each group, suggesting good sample quality and no major batch effects.

```r
# Perform DESeq2 analysis
dds <- DESeq(dds)
```

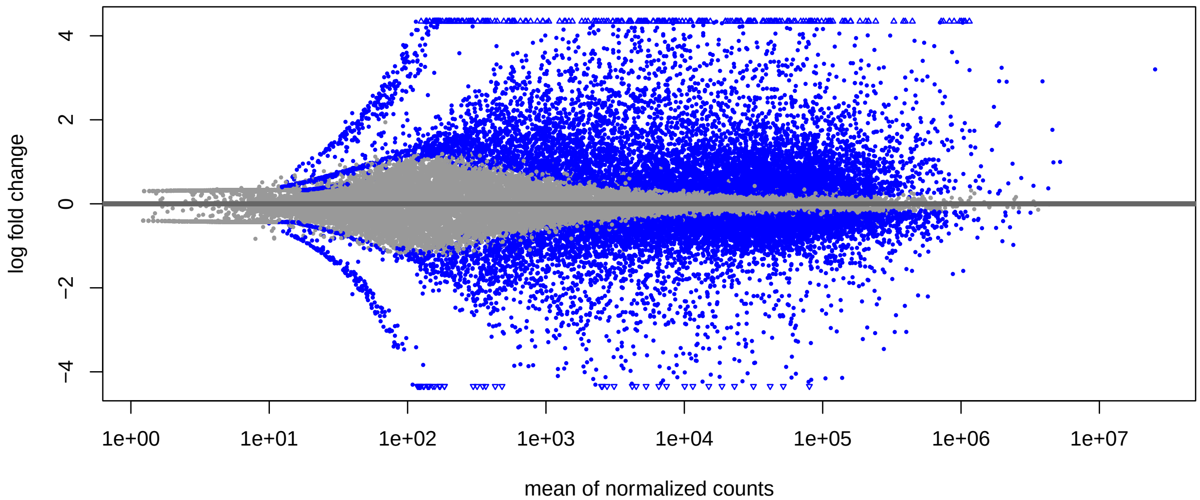
```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```r
res <- results(dds)
```

```r
# -- plotMA
plotMA(res)
```




```r
# LFC shrink
#find the fold change is exagerrated and shrink into right
resNorm <- lfcShrink(dds = dds, res = res, type = "normal", coef = 2)
#coef = 2 means change the second results
# -- plotMA with resNorm
plotMA(resNorm)
```




```r
#Create a result dataframe called resdf
resdf <- as.data.frame(resNorm) %>%
  rownames_to_column() %>%
  rename(ENSEMBL = rowname)
```

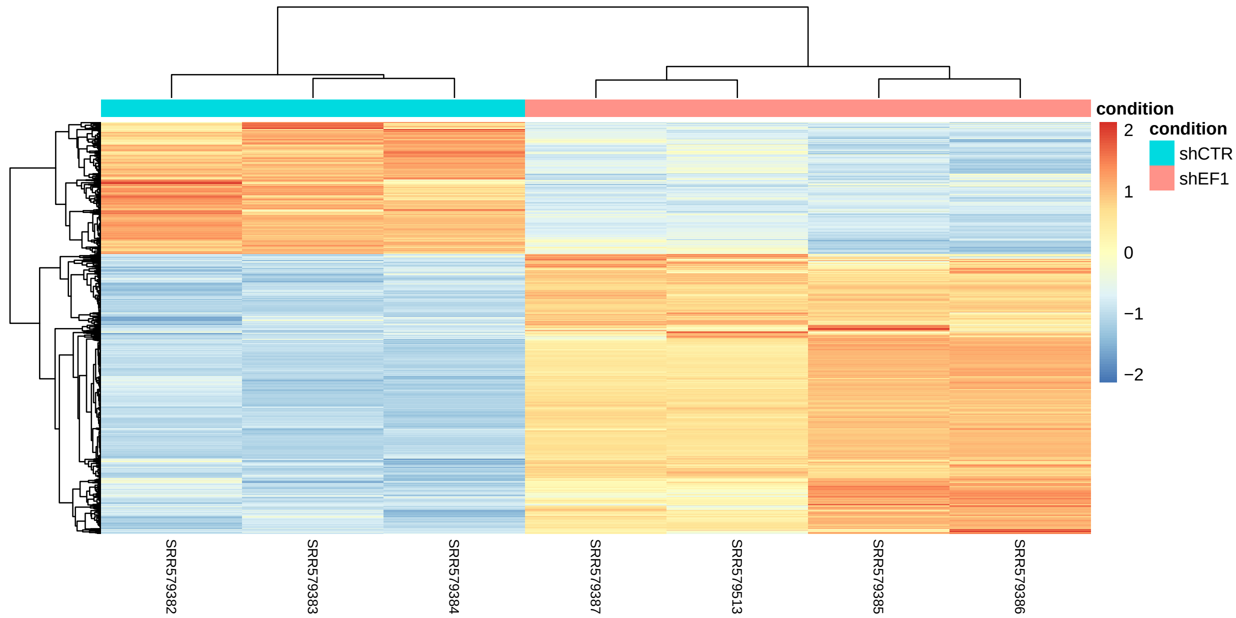
```r
#extract gene lists based on ENSEMBL
genelist <- AnnotationDbi::select(org.Hs.eg.db, keys = rownames(resNorm), keytype = "ENSEMBL",
                                  columns = c("ENTREZID", "SYMBOL", "GENETYPE","GENENAME"))
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```r
#filter genes with no padj, no symbol, or duplicated symbol
resdf <- resdf %>%
  left_join(genelist, by = "ENSEMBL") %>%
  filter(!is.na(padj)) %>%
  filter(!is.na(SYMBOL)) %>%
  filter(!duplicated(SYMBOL))
```

```r
#filter out significant genes
sig_gene_resdata <- resdf[resdf$baseMean >= 20 & abs(resdf$log2FoldChange) >=1 &
                            resdf$padj <= 0.05, ]
sig_gene <- sig_gene_resdata$ENSEMBL
```

```r
#heatmap for significant genes
sig_gene_logdata <- logdata[sig_gene,]

heatmap_anno <- as.data.frame(colData(dds)) %>%
  dplyr::select(condition)

pheatmap(sig_gene_logdata, scale = "row", clustering_distance_rows = "correlation",
         annotation_col = heatmap_anno, title = "Differenly expressed gene",
         show_rownames = FALSE, fontsize_col = 8)
```




```r
#count the number of up-regulated genes
nrow(resdf[resdf$baseMean >= 20 & resdf$log2FoldChange >=1 & resdf$padj <= 0.05, ])
```

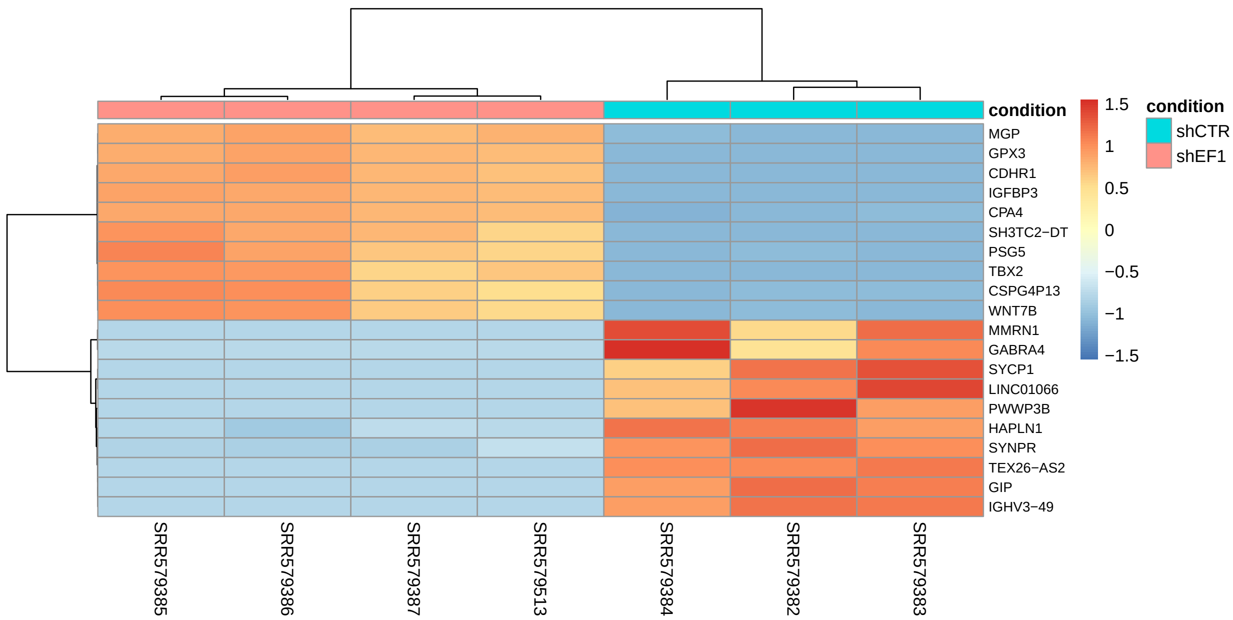
```
## [1] 3086
```

```r
#count the number of down-regulated genes
nrow(resdf[resdf$baseMean >= 20 & resdf$log2FoldChange <=-1 & resdf$padj <= 0.05, ])
```

```
## [1] 1455
```

```r
# extract data for the top 10-upregulated genes
top10_ensembl <- sig_gene_resdata %>%
  arrange(desc(log2FoldChange)) %>%
  head(10) %>%
  pull(ENSEMBL)

top10_geneid <- sig_gene_resdata %>%
  arrange(desc(log2FoldChange)) %>%
  head(10) %>%
  pull(SYMBOL)

upregulated_heatdata <- sig_gene_logdata[top10_ensembl,]
rownames(upregulated_heatdata) <- top10_geneid

# extract data for the top 10-downregulated genes
down10_ensembl <- sig_gene_resdata %>%
  arrange(log2FoldChange) %>%
  head(10) %>%
  pull(ENSEMBL)

down10_geneid <- sig_gene_resdata %>%
  arrange(log2FoldChange) %>%
```

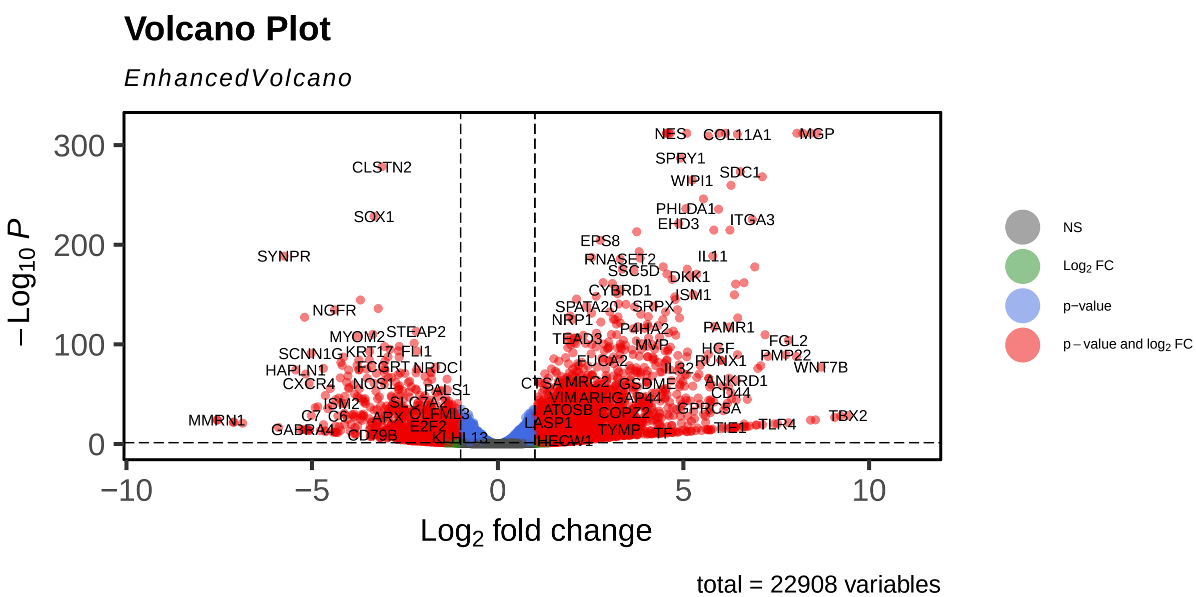
```r
  head(10) %>%
  pull(SYMBOL)

downregulated_heatdata <- sig_gene_logdata[down10_ensembl,]
rownames(downregulated_heatdata) <- down10_geneid

#Heatmap of the top and down regulated genes
mergeheatdata <- rbind(downregulated_heatdata, upregulated_heatdata)
pheatmap(mergeheatdata, scale = "row", clustering_distance_rows = "correlation",
         annotation_col = heatmap_anno, fontsize_row = 8)
```




```r
# Volcano plot
EnhancedVolcano(resdf, x = "log2FoldChange", y = "padj", lab = resdf$SYMBOL,
                FCcutoff = 1, pCutoff = 0.05, pointSize = 1, labSize = 3,
                title = "Volcano Plot", titleLabSize = 20, legendPosition = "right",
                legendIconSize = 8, legendLabSize = 8, gridlines.major = FALSE,
                gridlines.minor = FALSE, border = "full")
```

```
## Warning: One or more p-values is 0. Converting to 10^-1 * current lowest
## non-zero p-value...

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the EnhancedVolcano package.
##   Please report the issue to the authors.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
```

```
## i The deprecated feature was likely used in the EnhancedVolcano package.
##   Please report the issue to the authors.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
#Prepare gene list with decreasing log2FC for GSEA analysis (both GO and KEGG)
GSEAlist <- resdf$log2FoldChange
names(GSEAlist) <- resdf$ENTREZID

#remove NAs & remove duplicated IDs
GSEAlist <- GSEAlist[!is.na(GSEAlist)]
GSEAlist <- GSEAlist[!duplicated(names(GSEAlist))]

# sort for GSEA
GSEAlist <- sort(GSEAlist, decreasing = TRUE)
```

```r
#Gene Set Enrichment Analysis for GO/KEGG
gseaGO <- gseGO(geneList = GSEAlist, ont = "ALL", minGSSize = 30, keyType = "ENTREZID",
                pvalueCutoff = 0.05, pAdjustMethod = "BH", verbose = TRUE,
                OrgDb = org.Hs.eg.db)
```

```
## using 'fgsea' for GSEA analysis, please cite Korotkevich et al (2019).
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
```

```
## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize, gseaParam, : There are ties in
## The order of those tied genes will be arbitrary, which may produce unexpected results.
```

```
## Warning in fgseaMultilevel(pathways = pathways, stats = stats, minSize =
## minSize, : There were 31 pathways for which P-values were not calculated
## properly due to unbalanced (positive and negative) gene-level statistic values.
## For such pathways pval, padj, NES, log2err are set to NA. You can try to
## increase the value of the argument nPermSimple (for example set it nPermSimple
## = 10000)

## Warning in fgseaMultilevel(pathways = pathways, stats = stats, minSize =
## minSize, : For some of the pathways the P-values were likely overestimated. For
## such pathways log2err is set to NA.

## Warning in fgseaMultilevel(pathways = pathways, stats = stats, minSize =
## minSize, : For some pathways, in reality P-values are less than 1e-10. You can
## set the `eps` argument to zero for better estimation.

## leading edge analysis...

## done...
```

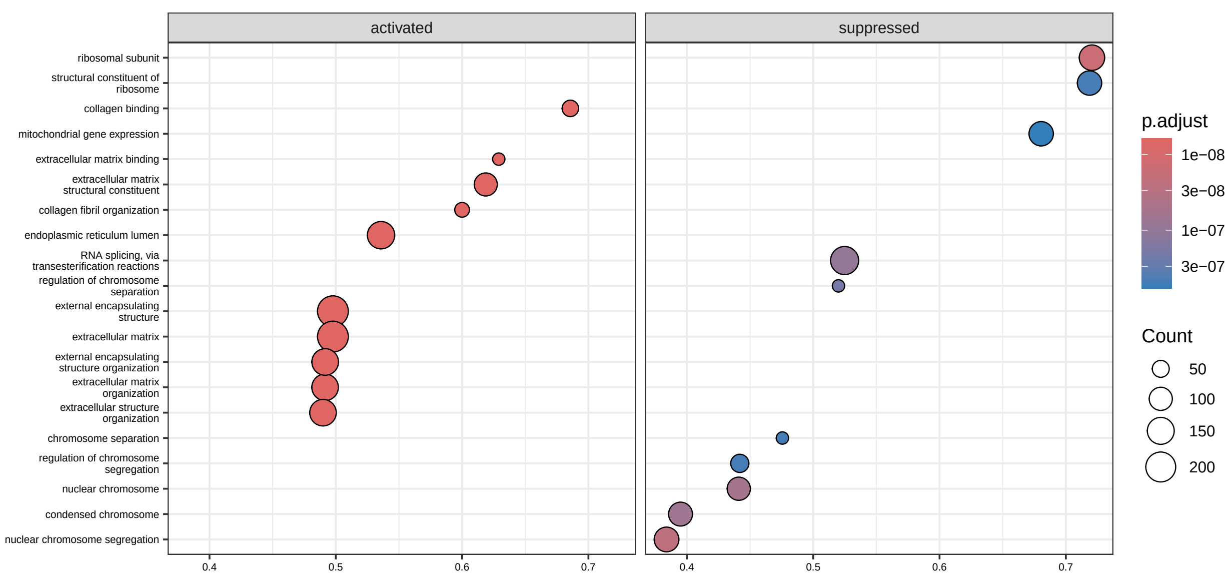
```r
gseaKEGG <- gseKEGG(geneList = GSEAlist, organism = "hsa", minGSSize = 30,
                    pvalueCutoff = 0.05, pAdjustMethod = "BH", verbose = TRUE)
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/link/hsa/pathway"...

## Reading KEGG annotation online: "https://rest.kegg.jp/list/pathway/hsa"...

## using 'fgsea' for GSEA analysis, please cite Korotkevich et al (2019).

## preparing geneSet collections...

## GSEA analysis...

## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize, gseaParam, : There are ties in
## The order of those tied genes will be arbitrary, which may produce unexpected results.

## Warning in fgseaMultilevel(pathways = pathways, stats = stats, minSize =
## minSize, : There were 1 pathways for which P-values were not calculated
## properly due to unbalanced (positive and negative) gene-level statistic values.
## For such pathways pval, padj, NES, log2err are set to NA. You can try to
## increase the value of the argument nPermSimple (for example set it nPermSimple
## = 10000)

## Warning in fgseaMultilevel(pathways = pathways, stats = stats, minSize =
## minSize, : For some of the pathways the P-values were likely overestimated. For
## such pathways log2err is set to NA.

## Warning in fgseaMultilevel(pathways = pathways, stats = stats, minSize =
## minSize, : For some pathways, in reality P-values are less than 1e-10. You can
## set the `eps` argument to zero for better estimation.

## leading edge analysis...

## done...
```

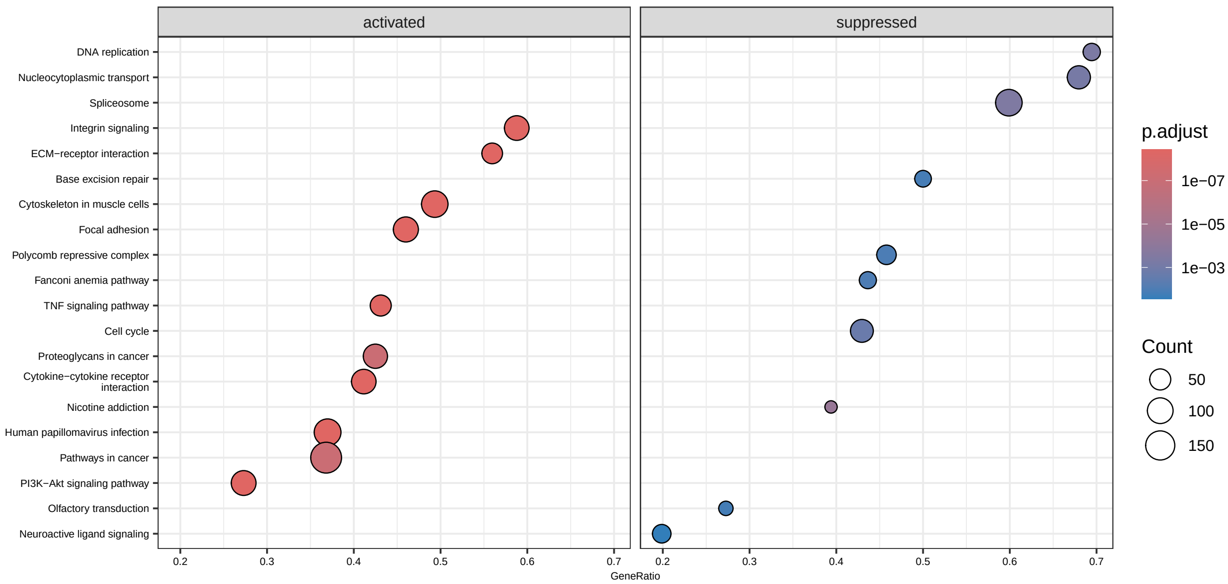
```
#Make gene IDs readable (converts Entrez IDs to gene symbols)
gseaGO <- setReadable(gseaGO, OrgDb = org.Hs.eg.db, keyType = 'ENTREZID')
gseaKEGG <- setReadable(gseaKEGG, OrgDb = org.Hs.eg.db, keyType = 'ENTREZID')
```

```
#function to find the associated pathways with gene
gene_pathway_gsea <- function(gene, pathwaylist) {
  down_list <- list()
  for (i in seq(1, nrow(pathwaylist), by = 1)) {
    seq = pathwaylist$core_enrichment[i]
    genes <- strsplit(seq, "/")[[1]] # split string into vector
    down_list[[i]] <- genes
  }

  for (a in seq_len(nrow(pathwaylist))) {
    list_gene = down_list[[a]]
    for (b in seq_len(length(list_gene))) {
      if (list_gene[b] == gene) {
        print(paste0("the pathway called ",pathwaylist$Description[[a]],
                     " contains gene ", gene, " with the NES score: ", pathwaylist$NES[[a]]))
      }
    }
  }
}
```

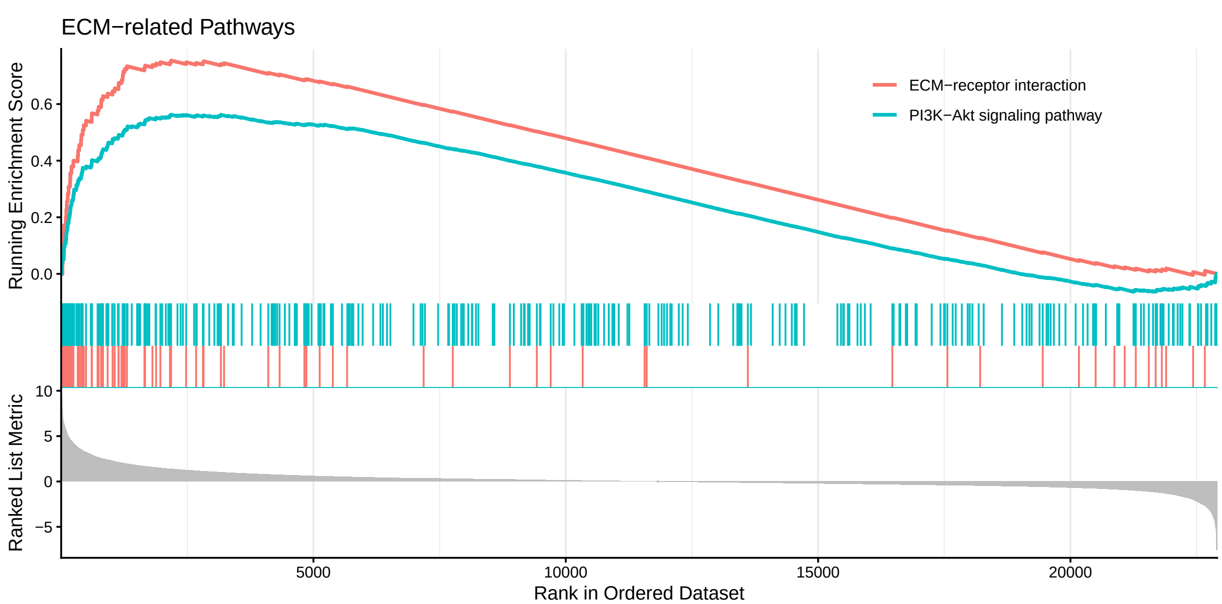
```
#Perform dotplot using GO and KEGG
dotplot(gseaGO, showCategory = 10, font.size = 6, split = ".sign") + facet_grid(.~.sign)
```



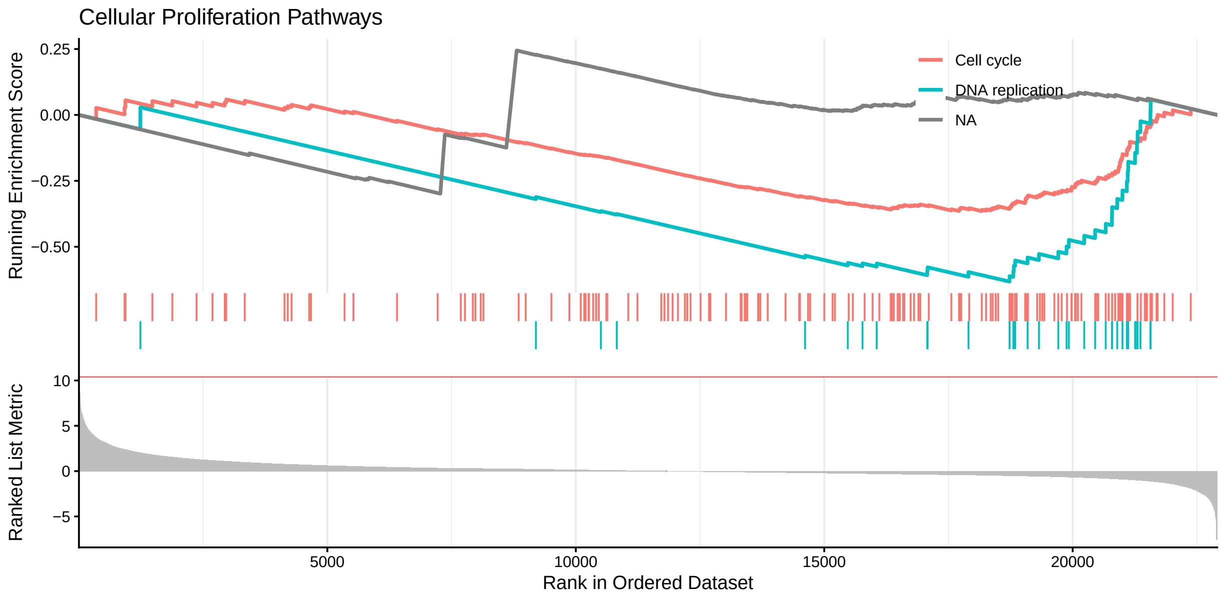
```r
dotplot(gseaKEGG, showCategory = 10, font.size = 6, split = ".sign") + facet_grid(.~.sign)
```




```r
#gseaplot for ECM-related Pathways
gseaplot2(gseaKEGG, geneSetID = c("hsa04512","hsa04151"), title = "ECM-related Pathways")
```

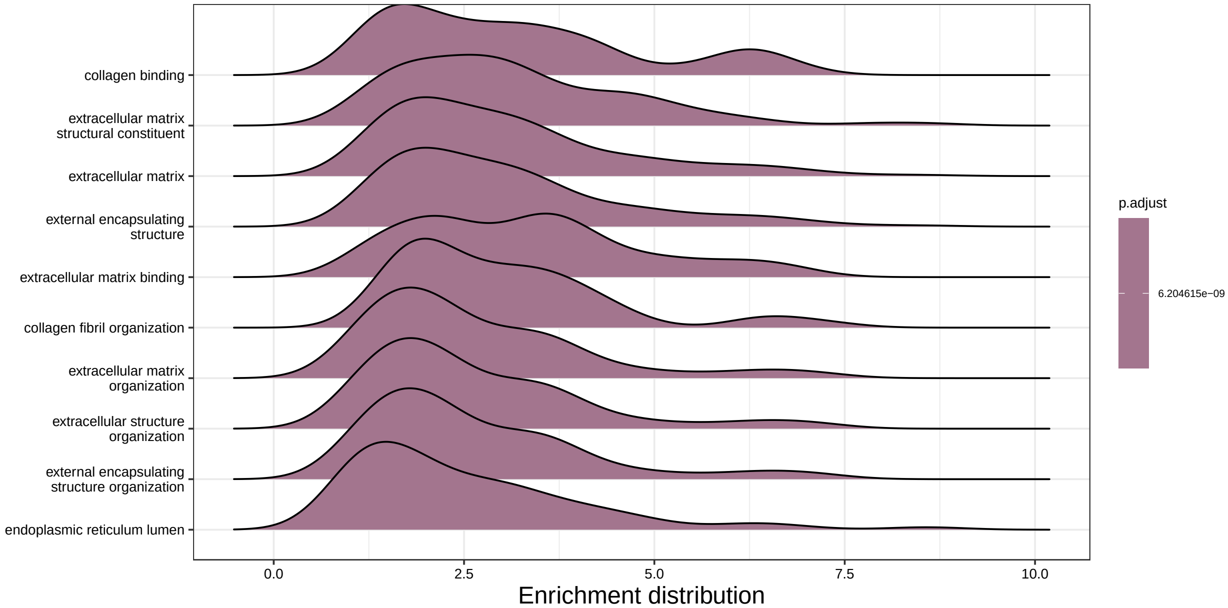



```r
#gseaplot for Cellular Proliferation Pathways
gseaplot2(gseaKEGG, geneSetID = c("hsa03010","hsa03030","hsa04110"), title = "Cellular Proliferation Pa
```



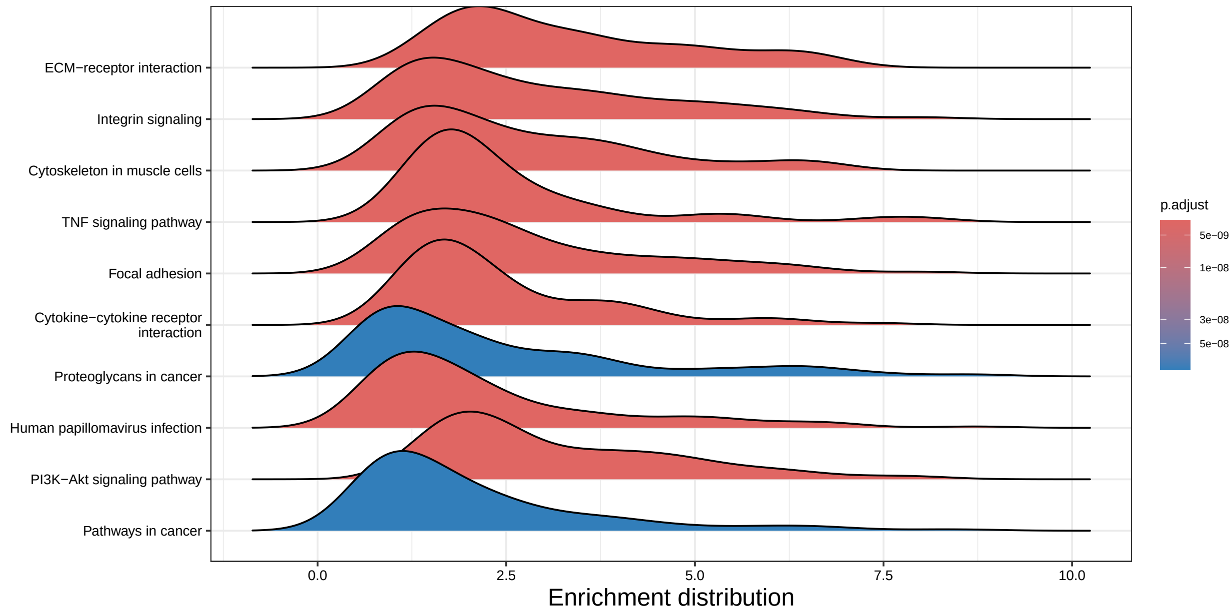

```r
#Perform ridgeplot using GO and KEGG
ridgeplot(gseaGO, showCategory = 10) + labs(x = "Enrichment distribution") +
  theme(text = element_text(size = 8),
        axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8))
```

```
## Picking joint bandwidth of 0.494
```

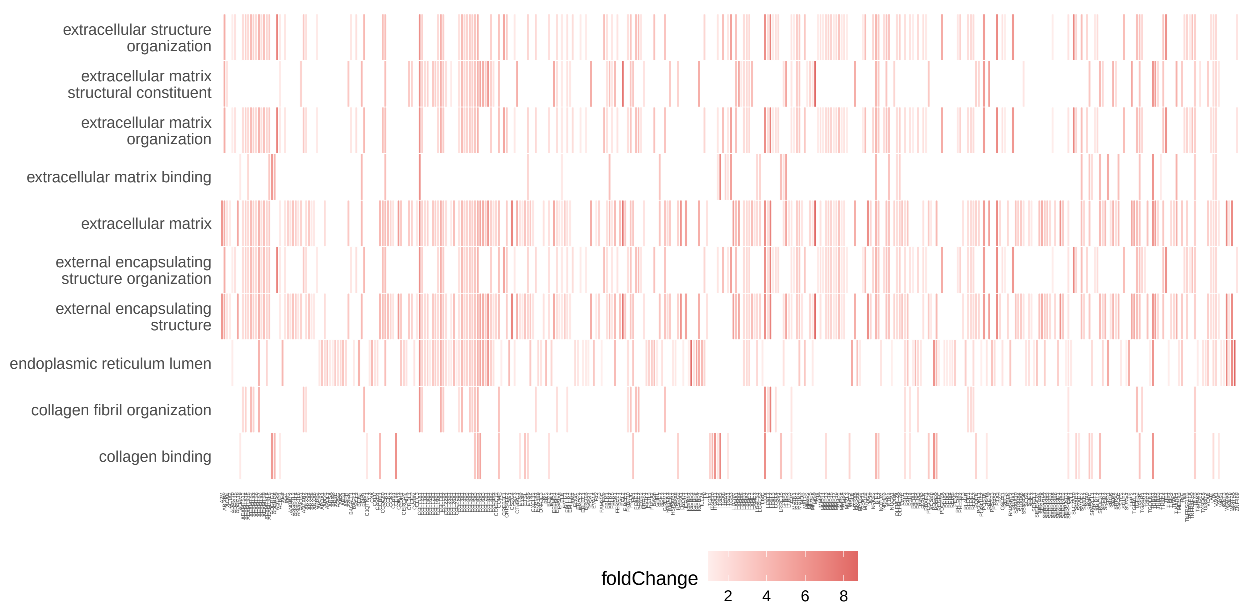



```r
ridgeplot(gseaKEGG, showCategory = 10) + labs(x = "Enrichment distribution") +
  theme(text = element_text(size = 8),
        axis.text.x = element_text(size = 8),
        axis.text.y = element_text(size = 8))
```

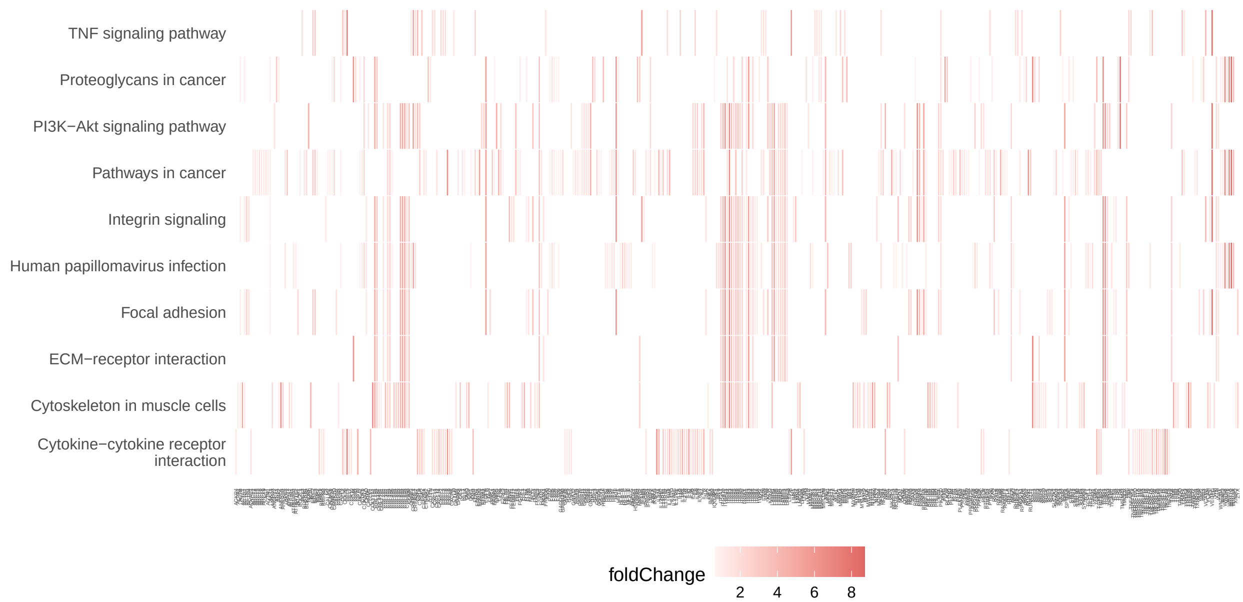
```
## Picking joint bandwidth of 0.51
```




```r
#Perform heatplot using GO and KEGG
heatplot(gseaGO, showCategory = 10, foldChange = GSEAlist) +
  theme(axis.text.x = element_text(size = 3, angle = 90, hjust = 1, vjust = 0.5)) +
  theme(legend.position = "bottom")
```




```r
heatplot(gseaKEGG, showCategory = 10, foldChange = GSEAlist) +
  theme(axis.text.x = element_text(size = 3, angle = 90, hjust = 1, vjust = 0.5)) +
  theme(legend.position = "bottom")
```

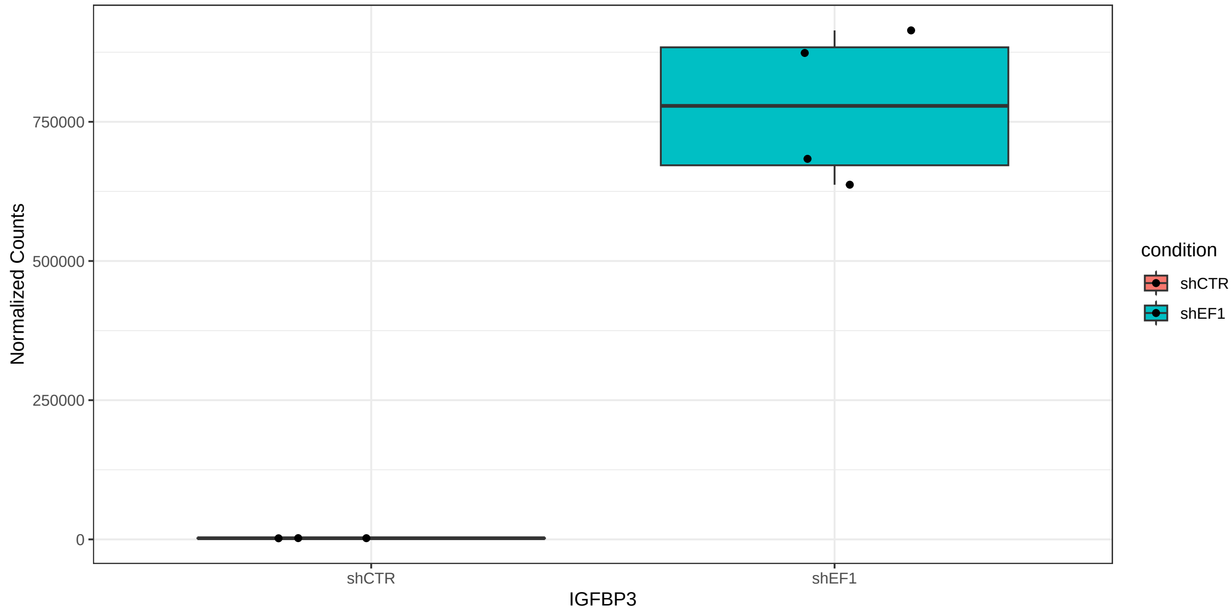



```
#Perform category plot using GO and KEGG
#categorySize can be either 'pvalue' or 'geneNum'
#GO_cnet <- cnetplot(gseaGO,font.size=4, categorySize="geneNum", foldChange=GSEAlist,max.overlaps=50,no
#KEGG_cnet <- cnetplot(gseaKEGG,font.size=4, categorySize="geneNum", foldChange=GSEAlist,max.overlaps=5
#cowplot::plot_grid(GO_cnet, KEGG_cnet)
```

```
#draw countplot for IGFBP3
resdf %>%
  filter(SYMBOL == "IGFBP3")
```

```
##           ENSEMBL baseMean log2FoldChange    lfcSE     stat pvalue padj
## 1 ENSG00000146674 444947.3       8.435331 0.144721 58.27098      0    0
##   ENTREZID SYMBOL       GENETYPE                                   GENENAME
## 1     3486 IGFBP3 protein-coding insulin like growth factor binding protein 3
```

```
IGFBP3 <- plotCounts(dds, gene = "ENSG00000146674", intgroup = "condition", returnData = TRUE)
ggplot(IGFBP3, aes(x = condition, y = count, fill = condition)) + geom_boxplot() +
  geom_jitter(width = 0.2) + theme_bw() + ylab("Normalized Counts") + xlab("IGFBP3")
```




```r
#extract count data for SYNPR
resdf %>%
  filter(SYMBOL == "SYNPR")
```

```
##           ENSEMBL baseMean log2FoldChange     lfcSE      stat        pvalue
## 1 ENSG00000163630  23045.2      -5.755599 0.1944637 -29.58401 2.399955e-192
##            padj ENTREZID SYMBOL       GENETYPE     GENENAME
## 1 2.537449e-189   132204  SYNPR protein-coding synaptoporin
```

```r
SYNPR <- plotCounts(dds, gene = "ENSG00000163630", intgroup = "condition", returnData = TRUE)
SYNPRplot <- ggplot(SYNPR, aes(x = condition, y = count, fill = condition)) + geom_boxplot() +
  geom_jitter(width = 0.2) + theme_bw() + ylab("Normalized Counts") + xlab("SYNPR")
```

```r
#extract count data for GABRA4
resdf %>%
  filter(SYMBOL == "GABRA4")
```

```
##           ENSEMBL baseMean log2FoldChange     lfcSE      stat       pvalue
## 1 ENSG00000109158 169.5765      -5.249731 0.6053175 -8.209414 2.222705e-16
##           padj ENTREZID SYMBOL       GENETYPE
## 1 2.979606e-15     2557 GABRA4 protein-coding
##                                                  GENENAME
## 1 gamma-aminobutyric acid type A receptor subunit alpha4
```

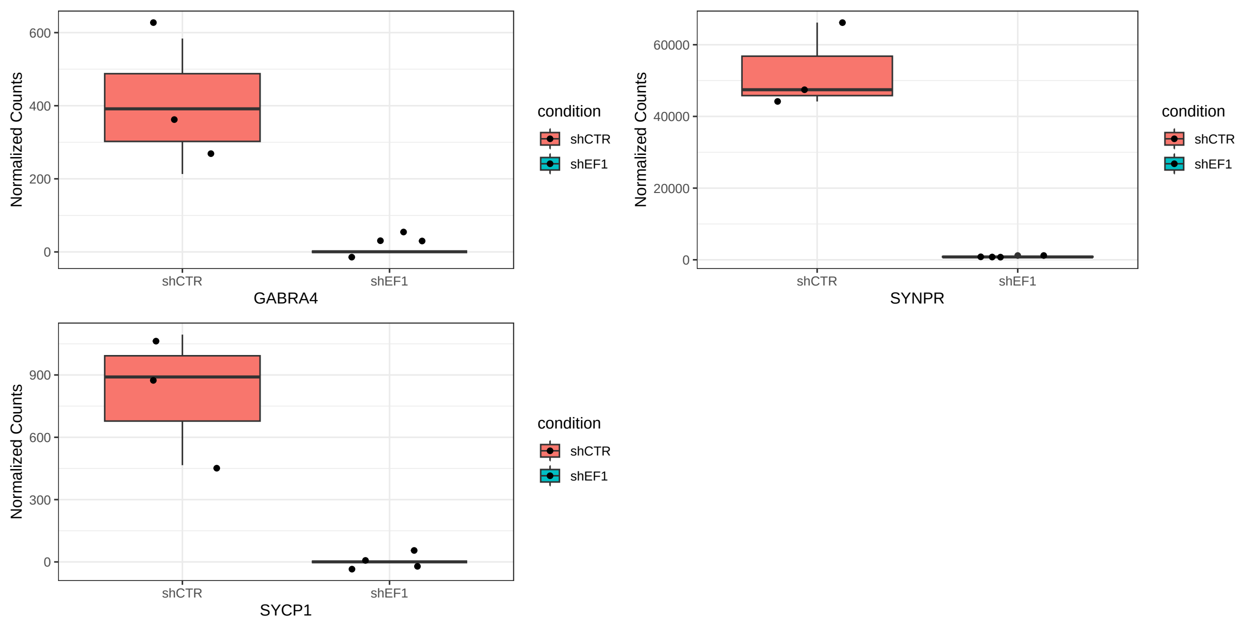
```r
GABRA4 <- plotCounts(dds, gene = "ENSG00000109158", intgroup = "condition", returnData = TRUE)
GABRA4plot <- ggplot(GABRA4, aes(x = condition, y = count, fill = condition)) + geom_boxplot() +
  geom_jitter(width = 0.2) + theme_bw() + ylab("Normalized Counts") + xlab("GABRA4")
```

```r
#extract count data for SYCP1
resdf %>%
  filter(SYMBOL == "SYCP1")
```

```
##           ENSEMBL baseMean log2FoldChange     lfcSE      stat       pvalue
## 1 ENSG00000198765 349.8534      -7.104489 0.5533138 -10.08841 6.216952e-24
##           padj ENTREZID SYMBOL       GENETYPE                      GENENAME
## 1 1.324882e-22     6847  SYCP1 protein-coding synaptonemal complex protein 1
```

```
SYCP1 <- plotCounts(dds, gene = "ENSG00000198765", intgroup = "condition", returnData = TRUE)
SYCP1plot <- ggplot(SYCP1, aes(x = condition, y = count, fill = condition)) + geom_boxplot() +
  geom_jitter(width = 0.2) + theme_bw() + ylab("Normalized Counts") + xlab("SYCP1")
```

```
#Draw countplot for GABRA4,SYNPR, and SYCP1
cowplot::plot_grid(GABRA4plot, SYNPRplot, SYCP1plot)
```




```
#extract count data for MGP
resdf %>%
  filter(SYMBOL == "MGP")
```

```
##           ENSEMBL baseMean log2FoldChange     lfcSE     stat pvalue padj
## 1 ENSG00000111341 241845.3       8.599807 0.1508039 56.98821      0    0
##   ENTREZID SYMBOL       GENETYPE           GENENAME
## 1     4256    MGP protein-coding matrix Gla protein
```

```
MGP <- plotCounts(dds, gene = "ENSG00000111341", intgroup = "condition", returnData = TRUE)
MGPplot <- ggplot(MGP, aes(x = condition, y = count, fill = condition)) + geom_boxplot() +
  geom_jitter(width = 0.2) + theme_bw() + ylab("Normalized Counts") +  xlab("MGP")
```

```
#extract count data for WNT7B
resdf %>%
  filter(SYMBOL == "WNT7B")
```

```
##           ENSEMBL baseMean log2FoldChange     lfcSE     stat       pvalue
## 1 ENSG00000188064 3287.612       8.707041 0.4053024 18.94608 4.757167e-80
##           padj ENTREZID SYMBOL       GENETYPE              GENENAME
## 1 6.960755e-78     7477  WNT7B protein-coding Wnt family member 7B
```

```r
WNT7B <- plotCounts(dds, gene = "ENSG00000188064", intgroup = "condition", returnData = TRUE)
WNT7Bplot <- ggplot(WNT7B, aes(x = condition, y = count, fill = condition)) + geom_boxplot() +
  geom_jitter(width = 0.2) + theme_bw() + ylab("Normalized Counts") +  xlab("WNT7B")
```

```r
#extract count data for CDHR1
resdf %>%
  filter(SYMBOL == "CDHR1")
```

```
##           ENSEMBL baseMean log2FoldChange     lfcSE     stat       pvalue
## 1 ENSG00000148600  865.966       8.556644 0.4695745 10.60554 2.808296e-26
##           padj ENTREZID SYMBOL       GENETYPE                         GENENAME
## 1 6.738266e-25    92211  CDHR1 protein-coding cadherin related family member 1
```

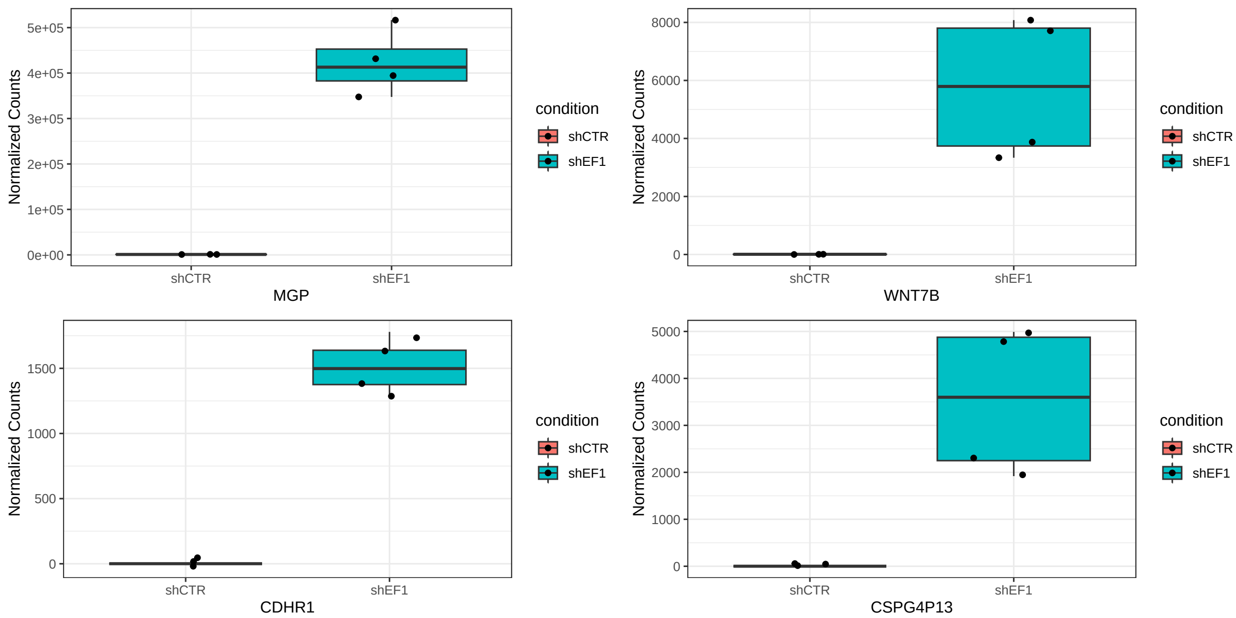
```r
CDHR1 <- plotCounts(dds, gene = "ENSG00000148600", intgroup = "condition", returnData = TRUE)
CDHR1plot <- ggplot(CDHR1, aes(x = condition, y = count, fill = condition)) + geom_boxplot() +
  geom_jitter(width = 0.2) + theme_bw() + ylab("Normalized Counts") +  xlab("CDHR1")
```

```r
#extract count data for CSPG4P13
resdf %>%
  filter(SYMBOL == "CSPG4P13")
```

```
##           ENSEMBL baseMean log2FoldChange     lfcSE     stat       pvalue
## 1 ENSG00000260139 2015.051       9.230152 0.4713314 11.39572 4.391879e-30
##           padj  ENTREZID   SYMBOL GENETYPE
## 1 1.266035e-28 100302666 CSPG4P13   pseudo
##                                         GENENAME
## 1 chondroitin sulfate proteoglycan 4 pseudogene 13
```

```r
CSPG4P13 <- plotCounts(dds, gene = "ENSG00000260139", intgroup = "condition", returnData = TRUE)
CSPG4P13plot <- ggplot(CSPG4P13, aes(x = condition, y = count, fill = condition)) + geom_boxplot() +
  geom_jitter(width = 0.2) + theme_bw() + ylab("Normalized Counts") +  xlab("CSPG4P13")
```

```r
# Draw countplot for MGP, WNT7B, HAPLN1, and CSPG4P13
cowplot::plot_grid(MGPplot, WNT7Bplot, CDHR1plot, CSPG4P13plot)
```


Normalized Counts
5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
shCTR
shEF1
MGP
condition
shCTR
shEF1
8000
6000
4000
2000
0
WNT7B
1500
1000
500
CDHR1
5000
3000
CSPG4P13