maketitle

Contents

1 General QC figures from DE analysis 2

2 Report from DESeq2 analysis 6

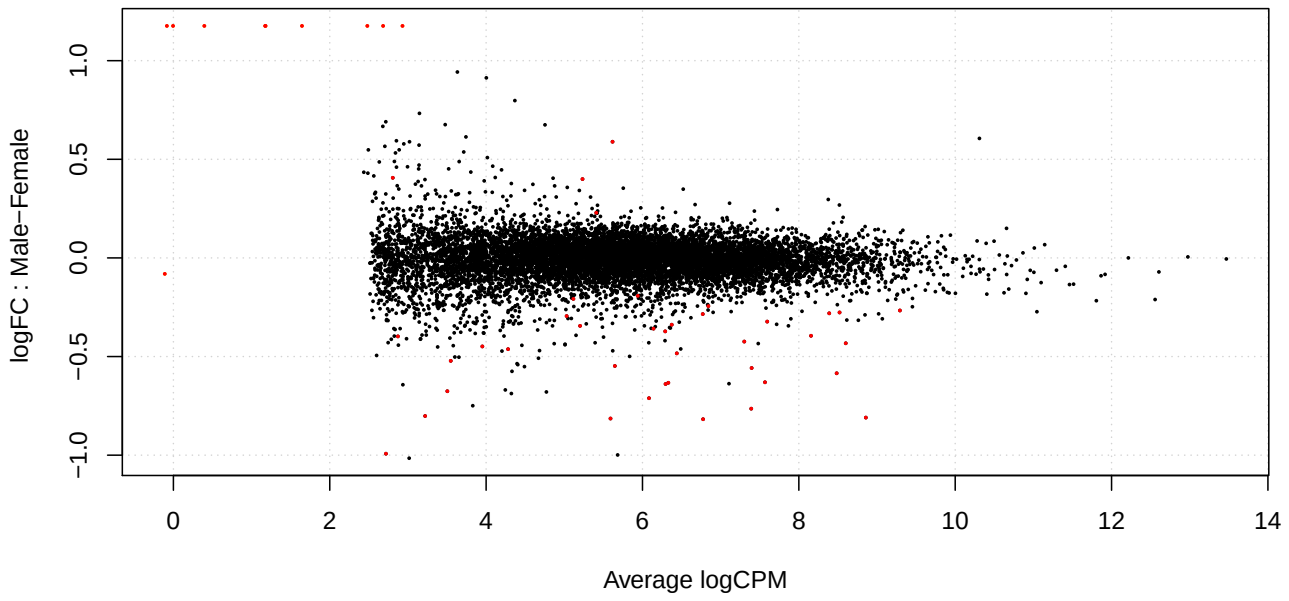
3 Detect patterns of expression 13

```
library(DEGreport)
data(humanSexDEedgeR)
library(edgeR)
```

1 General QC figures from DE analysis

We are going to do a differential expression analysis with edgeR. We have an object that is comming from the edgeR package. It countains a gene count matrix for 85 TSI HapMap individuals, and the gender information. With that, we are going to apply the 'glmFit' function to get genes differentially expressed between males and females.

```
des<-humanSexDEedgeR$design
fit <- glmFit(humanSexDEedgeR,des)
lrt <- glmLRT(fit)
tab<-cbind(lrt$table,p.adjust(lrt$table$PValue,method="BH"))
detags <- rownames(tab[tab[,5]<=0.1,])
plotSmear(humanSexDEedgeR, de.tags=detags)
```



We need to extract the experiment design data.frame where the condition is Male or Female.

```
counts<-cpm(humanSexDEedgeR,log=FALSE)
g1<-colnames(counts)[1:41]
g2<-colnames(counts)[42:85]
design<-data.frame(condition=sub("1","Male",sub("0","Female",des[,2])), other=1, row.names=colnames(counts
```

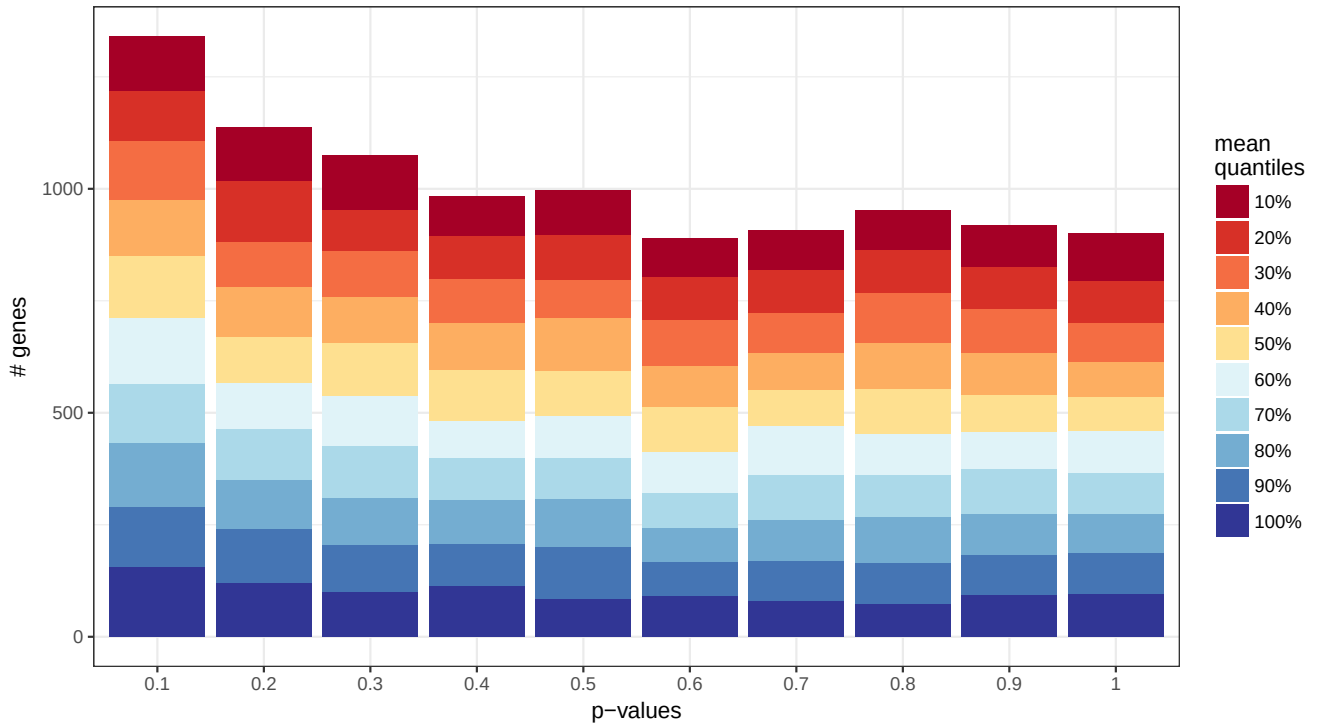
We are getting the chromosome information for each gene. This way we can colour genes according autosomic,X or Y chromosomes.

```
data(geneInfo)
```

p-value distribution gives an idea on how well you model is capturing the input data and as well whether it could be some

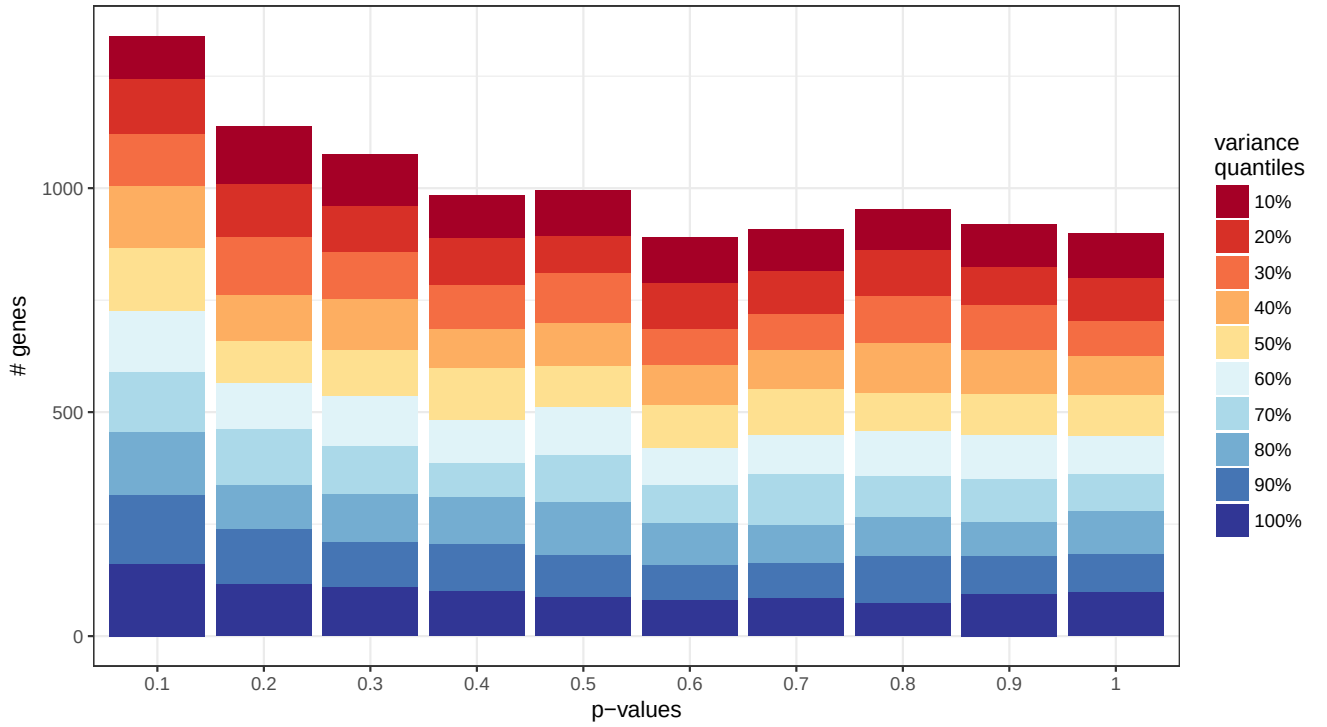
problem for some set of genes. In general, you expect to have a flat distribution with peaks at 0 and 1. In this case, we add the mean count information to check if any set of genes are enriched in any specific p-value range.

```
degMean(lrt$table$PValue, counts)
```



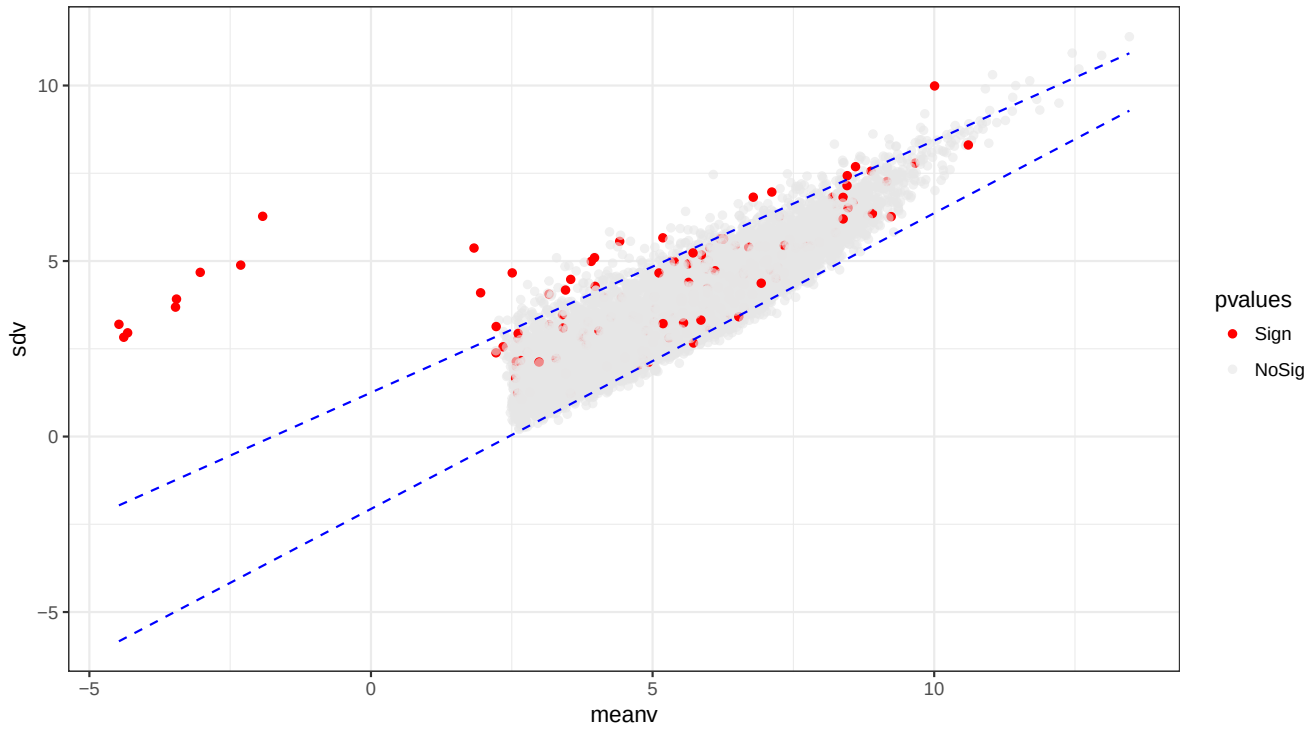
The same idea can be done with the gene variation.

```
degVar(lrt$table$PValue, counts)
```



Variation (dispersion) and average expresion relationship shouldn't be a factor among the differentialy expressed genes. When plotting average mean and standard desviation, significant genes should be randomly distributed.

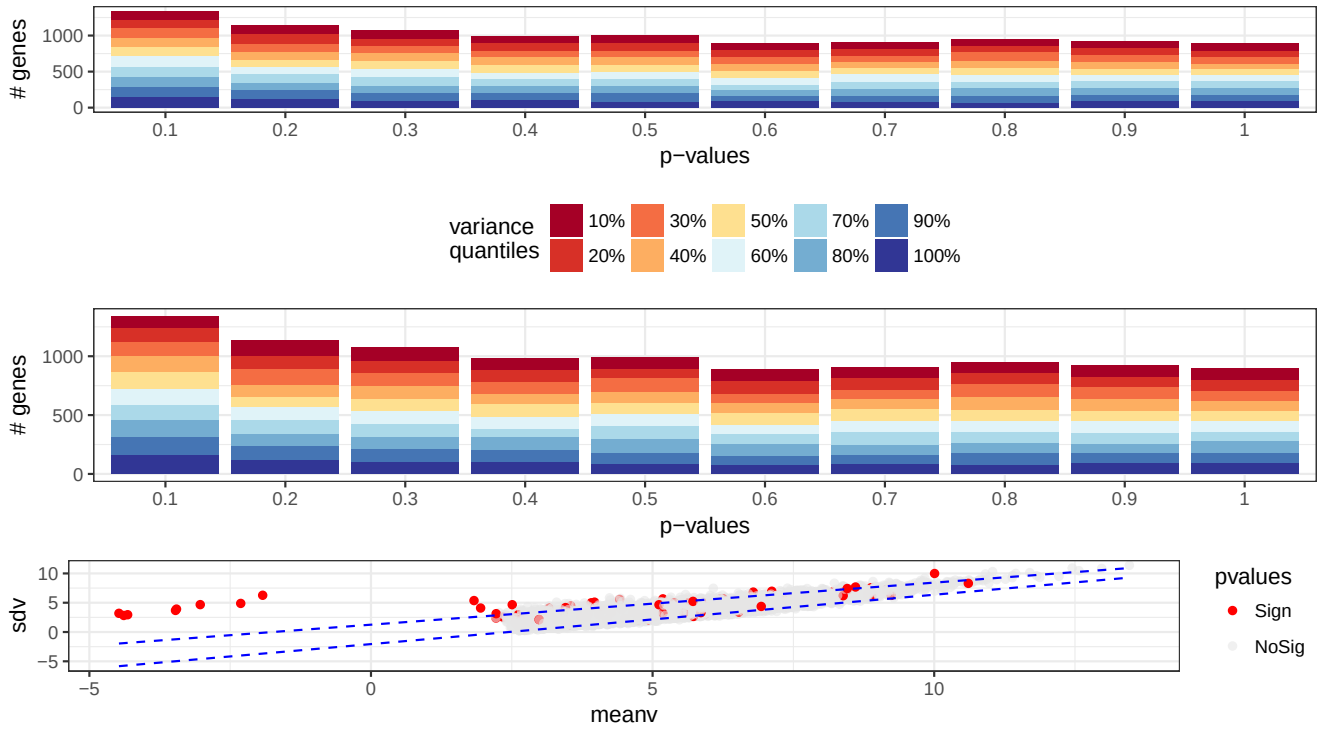
```
degMV(humanSexDEedgeR$samples$group, lrt$table$PValue, counts)
```



In this case, it would be good to look at the ones that are totally outside the expected correlation.

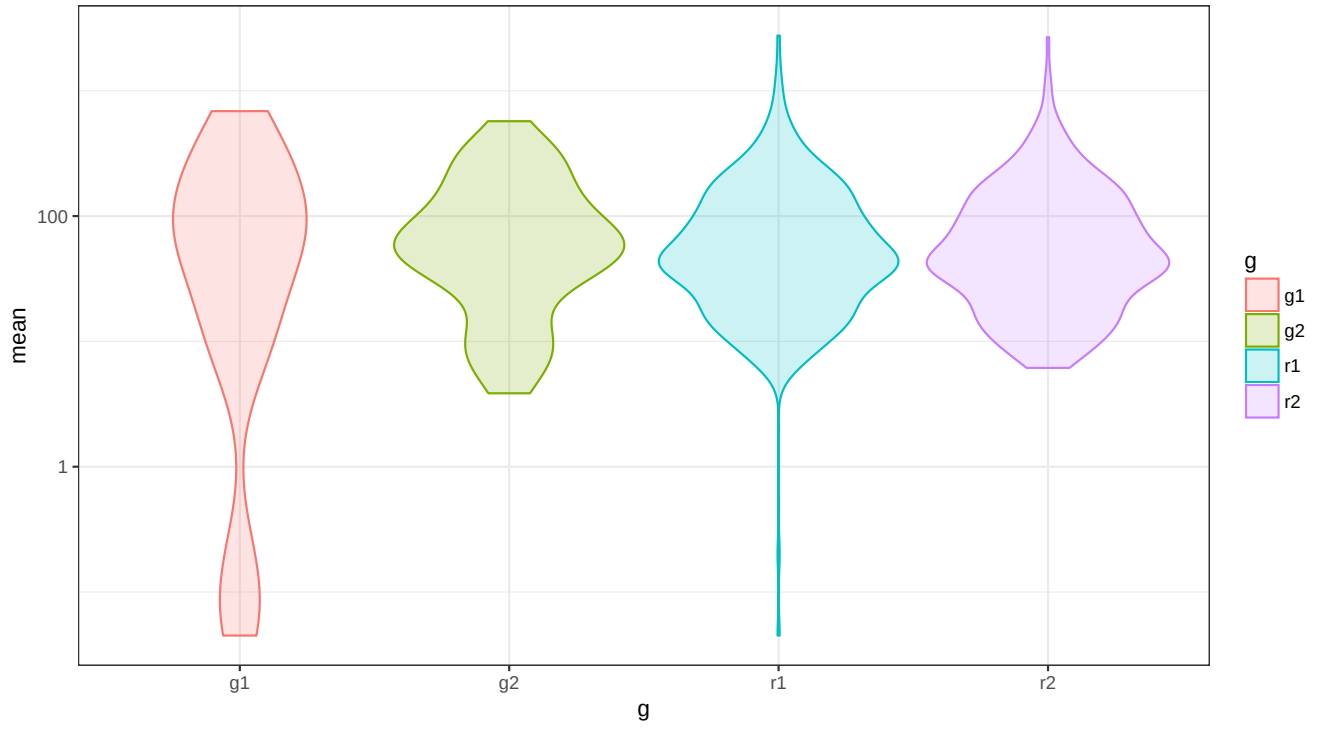
You can put this tree plots together using `degQC`.

```
degQC(lrt$table$PValue, counts, humanSexDEedgeR$samples$group)
```



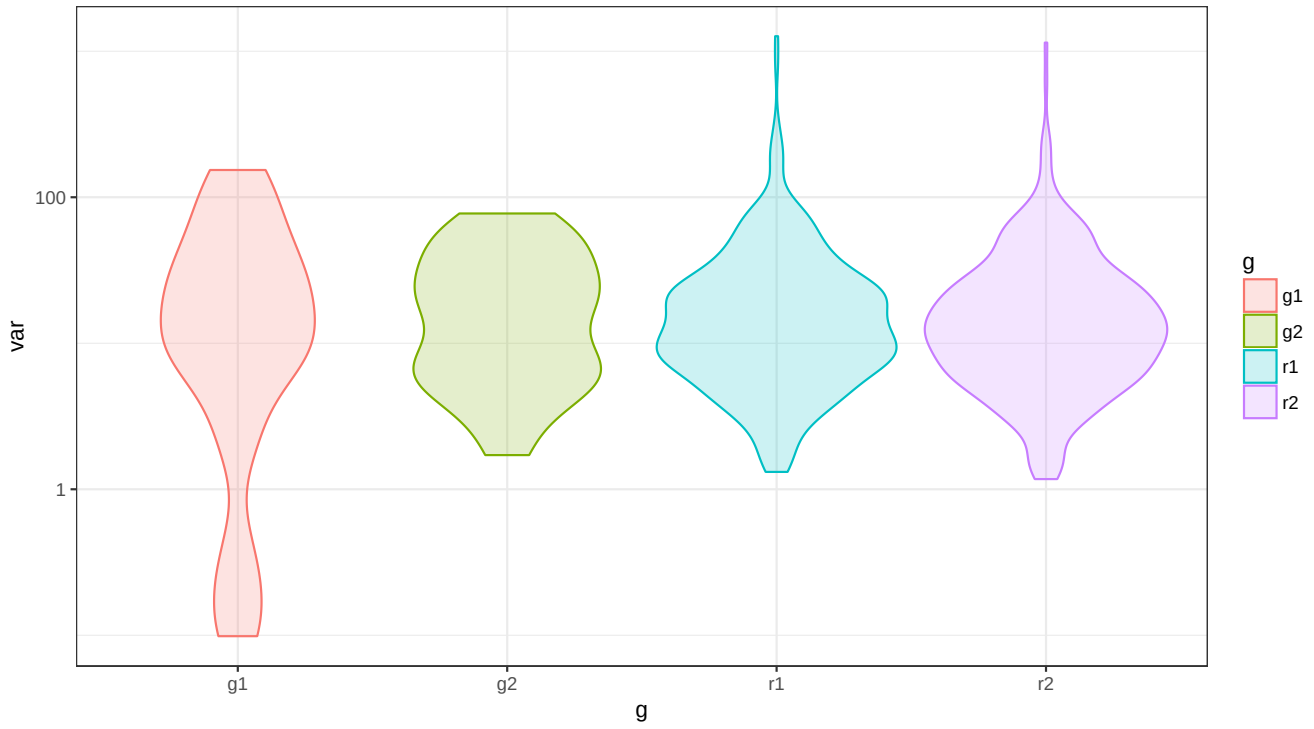
Other way to look at this is showing the mean count distribution among groups.

```
degMB(detags,g1,g2,counts)
```



The same idea can be applied to gene variation.

```
degVB(detags,g1,g2,counts)
```



Browsing gene expression can help to validate results or select some gene for donwstream analysis. Run the fowlling lines if you want to visualize your expression values by condition:

```
degObj(counts,design,"degObj.rda")
library(shiny)
shiny::runGitHub("lpantano/shiny", subdir="expression")
```

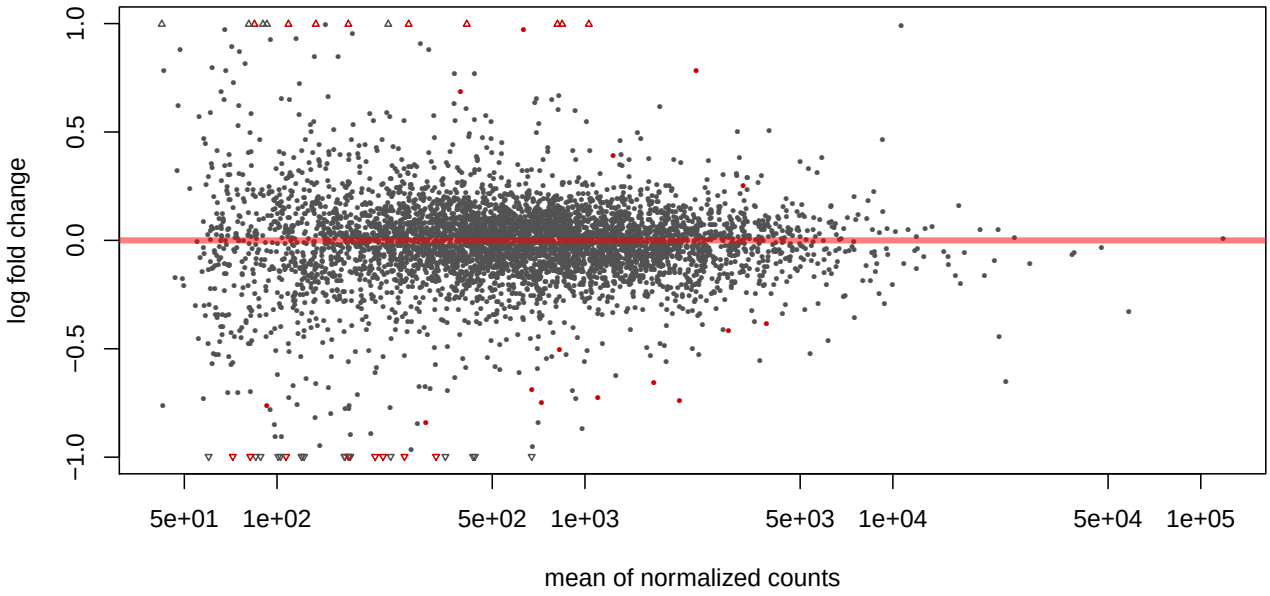
2 Report from DESeq2 analysis

In this section, we show how to use DESeq2 output to create a full report, including figures and tbale with top de-regulated genes, GO enrichment analysis and heatmaps and PCA plots. If you set `path_results`, different files will be saved there.

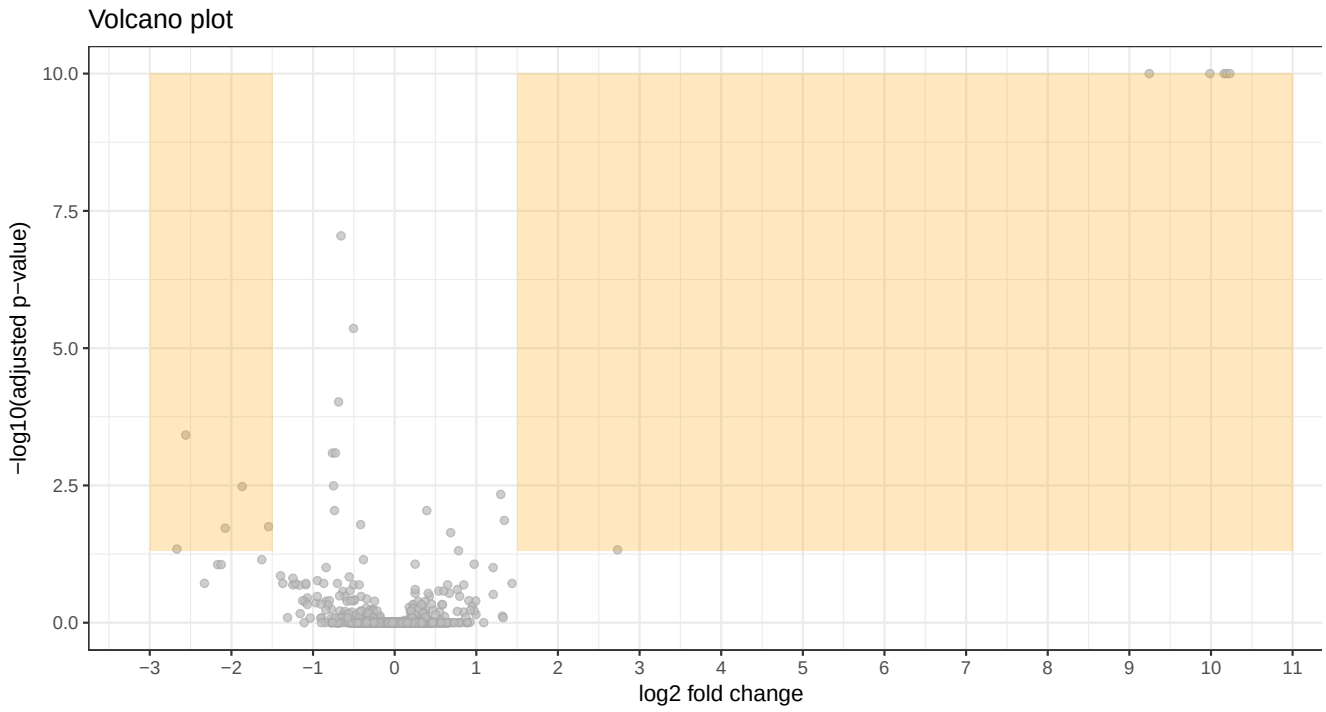
```
data(humanSexDEedgeR)
library(DESeq2)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(humanSexDEedgeR$counts[1:5000, idx],
                              humanSexDEedgeR$samples[idx,],
                              design=~group)
dse <- DESeq(dse)
res <- results(dse)
resreport <- degResults(dds=dse, name="test", org=NULL,
do_go=FALSE, group="group", xs="group", path_results = NULL)
```

```
## ## Comparison:  test {.tabset}
##
##
##  <br>out of 5000 with nonzero total read count<br>adjusted p-value < 0.1<br>LFC > 0 (up)      : 14, 0.28
##
##
```

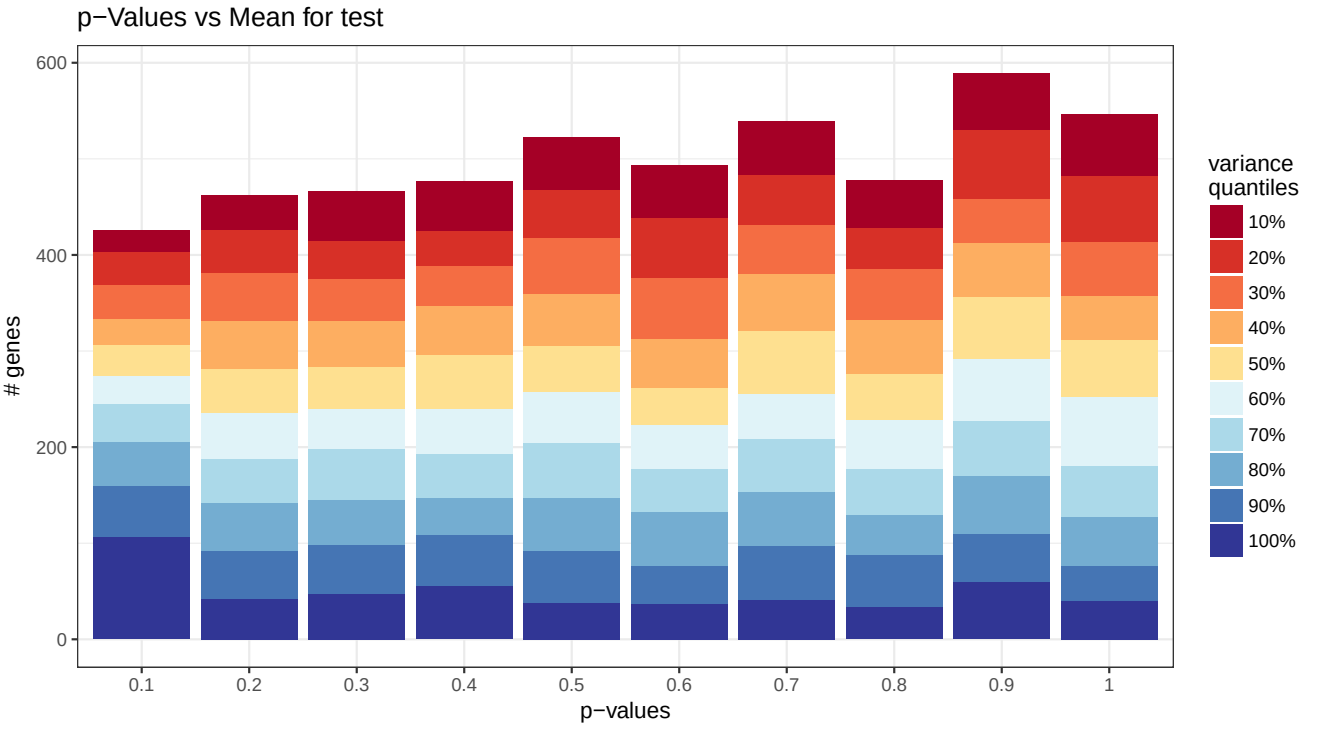
```
## Differential expression file at:  test_de.csv
##
## Normalized counts matrix file at:  test_log2_counts.csv
##
## ### MA plot plot
```




```
##
##
## ### Volcano plot
```

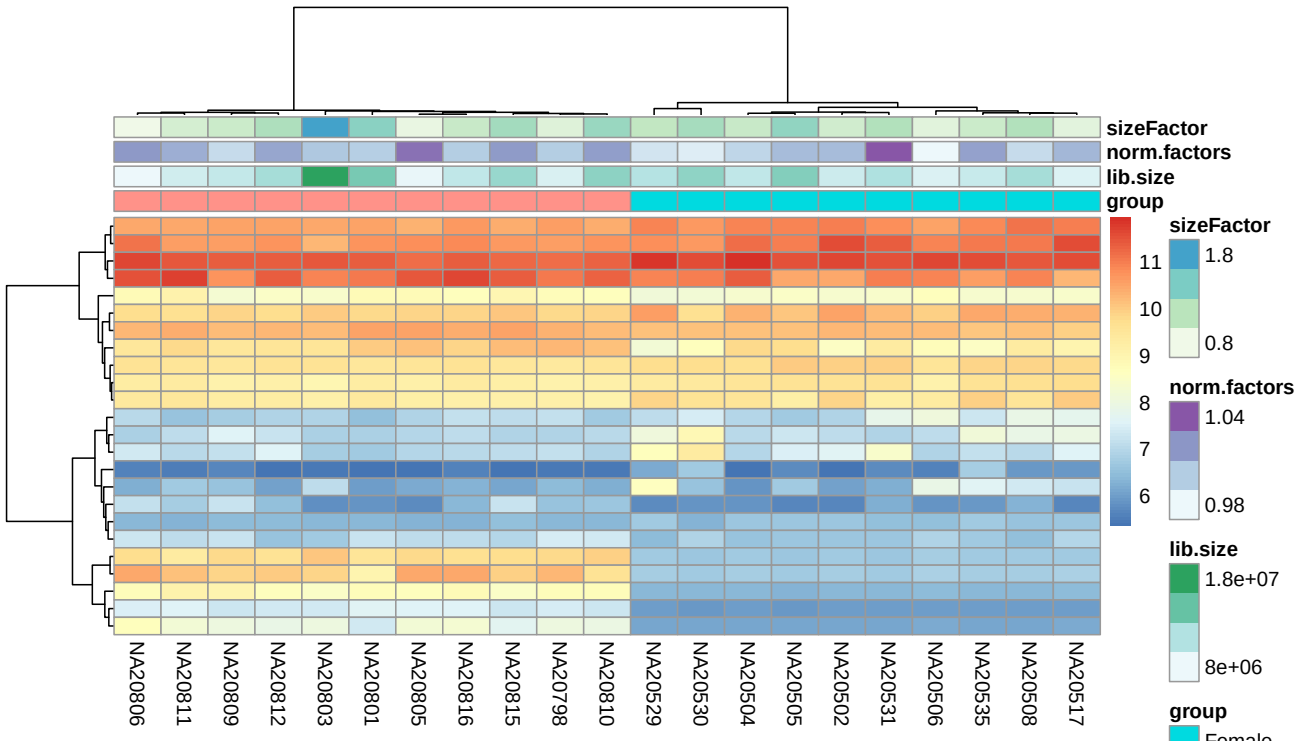


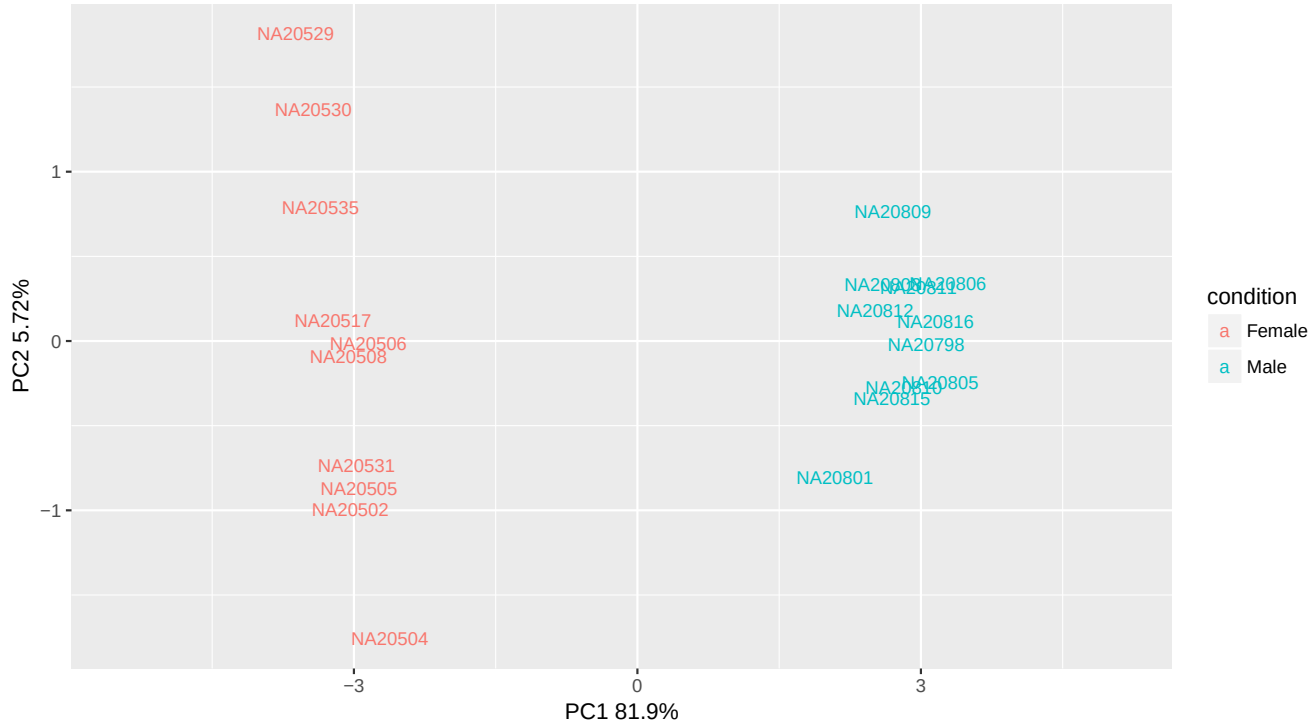

```
##
##
## ### QC for DE genes p-values/variance
```




```
##
##
## ### Most significand, FDR< 0.05  and log2FC >  0.1 :  24
```

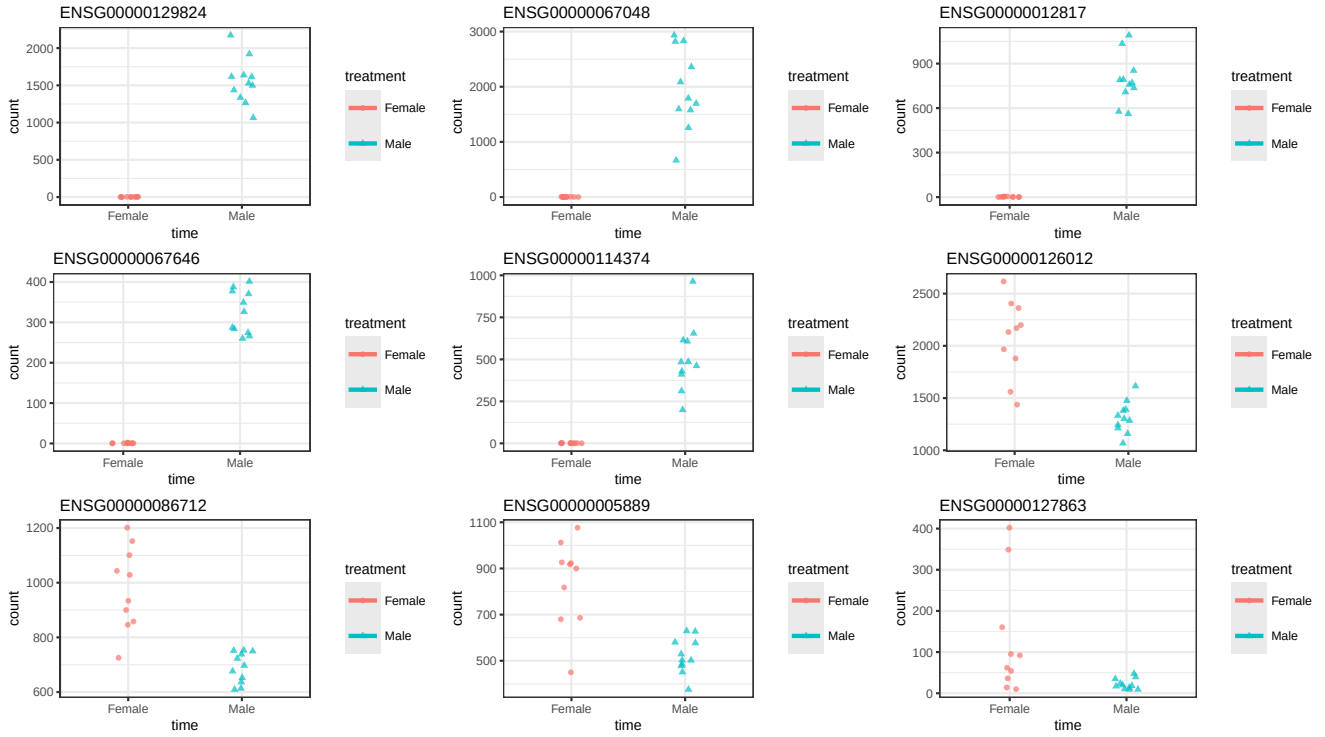
```
##
##
## ### Plots most significand
```






```
##
##
```

```
##
## Plot top 9 genes
```

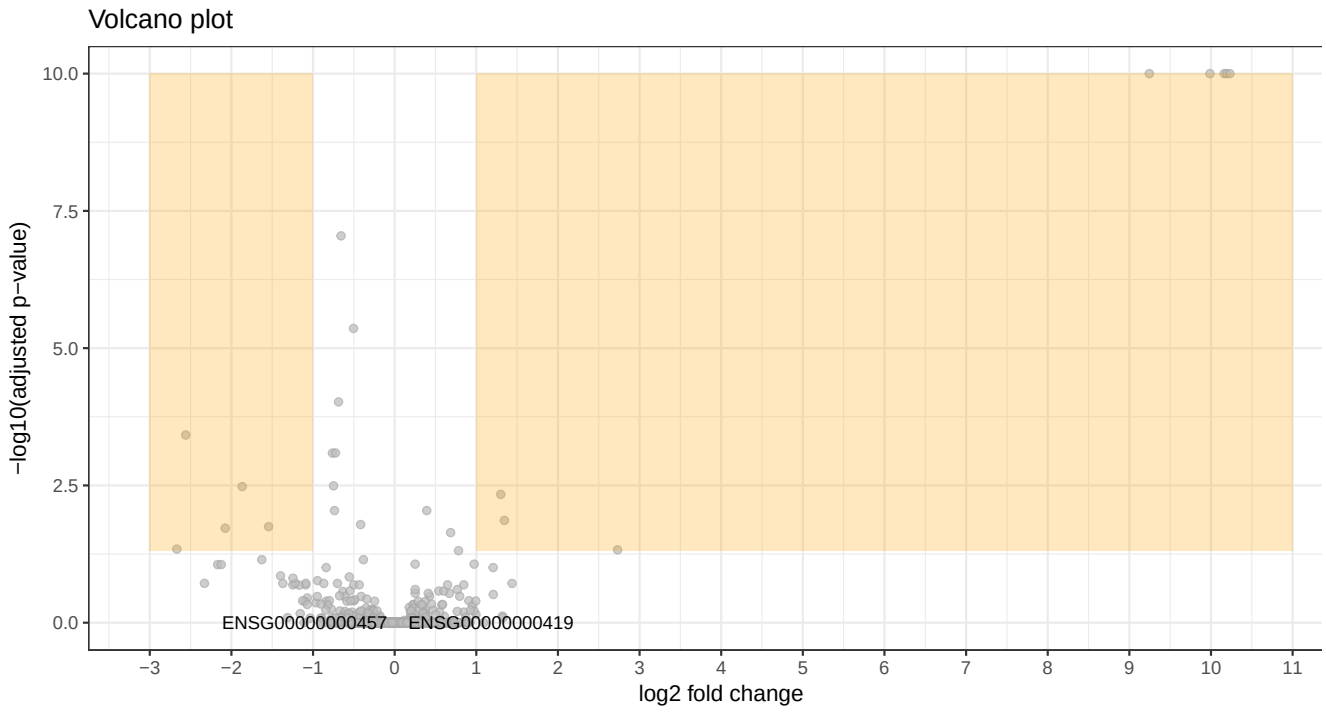



```
##
##
##
## ### Top DE genes
##
##
## \begin{tabular}{l|r|r|r|r|r|r|r}
## \hline
##   & baseMean & log2FoldChange & lfcSE & stat & pvalue & padj & absMaxLog2FC\\
## \hline
## ENSG00000129824 & 814.13416 & 10.2310169 & 0.4212221 & 24.288888 & 0.00e+00 & 0.0000000 & 10.2310169\\
## \hline
## ENSG00000067048 & 1029.66302 & 10.1630142 & 0.4227486 & 24.040324 & 0.00e+00 & 0.0000000 & 10.1630142\\
## \hline
## ENSG00000012817 & 413.38974 & 9.2448052 & 0.4239169 & 21.808062 & 0.00e+00 & 0.0000000 & 9.2448052\\
## \hline
## ENSG00000067646 & 170.50547 & 10.1920183 & 0.6578309 & 15.493370 & 0.00e+00 & 0.0000000 & 10.1920183\\
## \hline
## ENSG00000114374 & 267.68367 & 9.9877320 & 0.6644527 & 15.031516 & 0.00e+00 & 0.0000000 & 9.9877320\\
## \hline
## ENSG00000126012 & 1675.22034 & -0.6567918 & 0.1017438 & -6.455349 & 0.00e+00 & 0.0000001 & 0.6567918\\
## \hline
## ENSG00000086712 & 827.54826 & -0.5044764 & 0.0867767 & -5.813498 & 0.00e+00 & 0.0000044 & 0.5044764\\
## \hline
## ENSG00000005889 & 672.24906 & -0.6872559 & 0.1309146 & -5.249652 & 2.00e-07 & 0.0000952 & 0.6872559\\
## \hline
## ENSG00000127863 & 71.82765 & -2.5597667 & 0.5155504 & -4.965115 & 7.00e-07 & 0.0003814 & 2.5597667\\
```

```
## \hline
## ENSG00000130021 & 1102.21826 & -0.7249755 & 0.1511677 & -4.795836 & 1.60e-06 & 0.0008100 & 0.7249755\\
## \hline
## ENSG00000006757 & 92.81863 & -0.7620411 & 0.1595675 & -4.775665 & 1.80e-06 & 0.0008142 & 0.7620411\\
## \hline
## ENSG00000101846 & 723.09247 & -0.7488453 & 0.1674185 & -4.472894 & 7.70e-06 & 0.0032153 & 0.7488453\\
## \hline
## ENSG00000073282 & 220.95671 & -1.8676530 & 0.4197992 & -4.448919 & 8.60e-06 & 0.0033194 & 1.8676530\\
## \hline
## ENSG00000124256 & 133.65411 & 1.2990932 & 0.2978179 & 4.362039 & 1.29e-05 & 0.0046020 & 1.2990932\\
## \hline
## ENSG00000005020 & 1236.88298 & 0.3929158 & 0.0939787 & 4.180901 & 2.90e-05 & 0.0090736 & 0.3929158\\
## \hline
## ENSG00000005302 & 2030.87268 & -0.7371289 & 0.1758354 & -4.192154 & 2.76e-05 & 0.0090736 & 0.7371289\\
## \hline
## ENSG00000112486 & 843.04215 & 1.3443022 & 0.3301377 & 4.071944 & 4.66e-05 & 0.0137125 & 1.3443022\\
## \hline
## ENSG00000130741 & 2925.78447 & -0.4165810 & 0.1036856 & -4.017733 & 5.88e-05 & 0.0163224 & 0.4165810\\
## \hline
## ENSG00000137285 & 170.75604 & -1.5447348 & 0.3876995 & -3.984361 & 6.77e-05 & 0.0178058 & 1.5447348\\
## \hline
## ENSG00000134775 & 259.33782 & -2.0755273 & 0.5245926 & -3.956455 & 7.61e-05 & 0.0190175 & 2.0755273\\
## \hline
## \end{tabular}
```

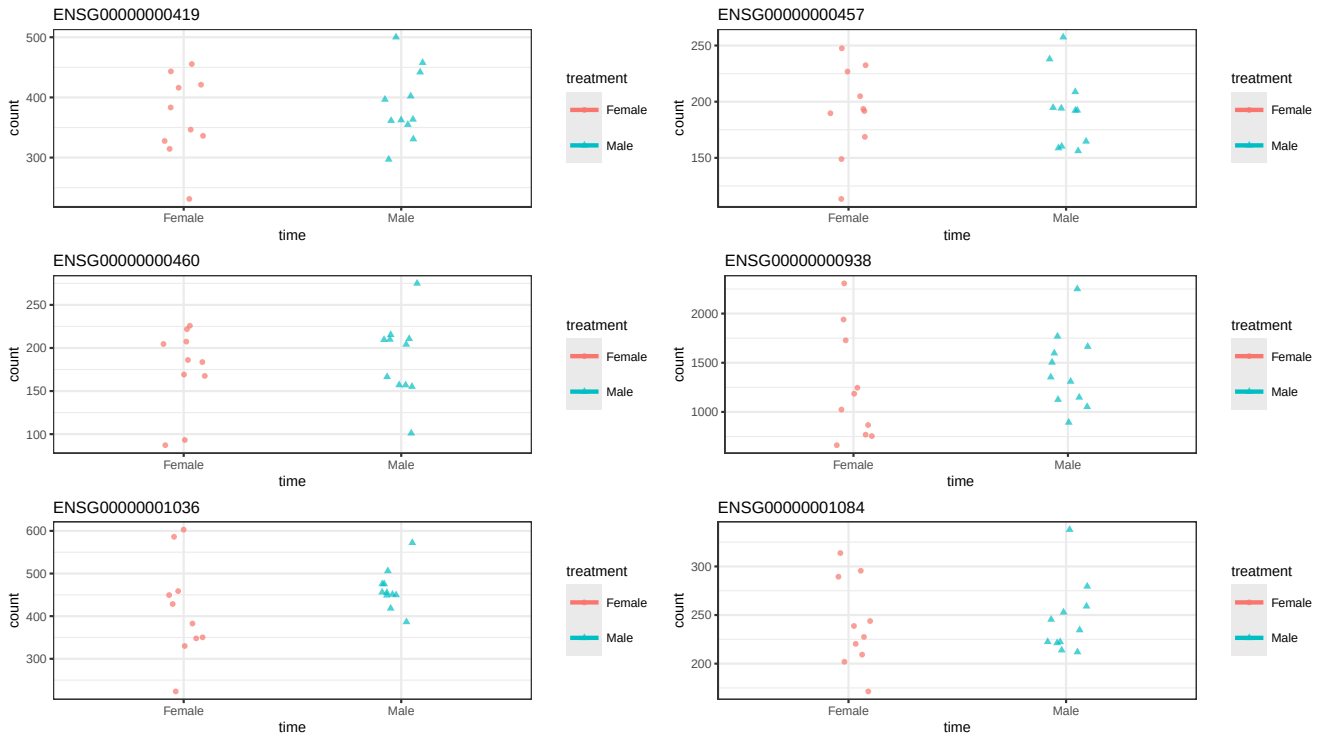
Volcano plot using the output of DESeq2. It mainly needs data.frame with two columns (logFC and pVal). Specific genes can be plot using the option `plot_text` (subset of the previous data.frame with a 3rd column to be used to plot the gene name).

```
res$id <- row.names(res)
show = as.data.frame(res[1:2, c("log2FoldChange", "padj", "id")])
degVolcano(as.data.frame(res[,c("log2FoldChange", "padj")]), plot_text = show)
```



Plot top genes coloring by group. Very useful for experiments with nested groups. ‘xs‘ can be ‘time‘ or ‘WT‘/‘KO‘, and ‘group‘ can be ‘treated‘/‘untreated‘. Another classification can be added, like ‘batch‘ that will plot points with different shapes.

```
degPlot(dds=dse, res = res, n = 6, xs = "group", group = "group")
```



3 Detect patterns of expression

In this section, we show how to detect pattern of expression. Mainly useful when data is a time course experiment. `degPatterns` needs a expression matrix, the design experiment and the column used to group samples.

```
ma = assay(rlog(dse))[row.names(res)[1:100],]
res <- degPatterns(ma, as.data.frame(colData(dse)), time="group", col=NULL)
```

```
##
##
## Working with  100  genes
##
##
##
##  Working with  100 genes after filtering: minc >  15
```