BIOL\*3300 Lab11 F21

**Differential Gene Expression Analysis (DGE)**

# Background review

In the last lab, we processed read counts and investigated how samples cluster using PCA and hierarchical clustering methods in R. In addition to performing such exploratory analyses based on normalized measures of expression levels, numerous efforts have been dedicated to optimize statistical tests to decide whether a given gene’s expression varies between two (or more) conditions based on the information gleaned from as little as two or three replicates per condition.

The two basic tasks of all DGE tools are:

1. Estimate the magnitude of differential expression between two or more conditions based on read counts from replicated samples, i.e., calculate the fold change of read counts, taking into account the differences in sequencing depth and variability.
2. Estimate the significance of the difference and correct for multiple testing.

In this lab, we assess the difference in expression between the control and over-expressed groups on a gene-by-gene basis and depict the KEGG pathway.

# Estimating the difference between read counts for a given gene

To determine whether the read count differences between different conditions for a given gene are greater than expected by chance, DGE tools must find a way to estimate that difference using the information from the replicates of each condition. The R package, DESeq2, uses regression models that are applied to every single gene. Linear regression models usually take the following typical form: Y = b0 +b1 \* x+e.

Here, Y will entail all read counts (from all conditions) for a given gene. b0 is called the intercept; x is the condition (for RNA-seq, this is very often a discrete factor, e.g., "WT" or

"mutant", or, in mathematical terms, 0 or 1), e is a term capturing the error or uncertainty, and b1 is the coefficient that captures the difference between the samples of different conditions.

The very simple model shown above could be fitted in R using the function lm(rlog.normgenotype) which will return estimates for both b0 and b1, so that the average expression values of the baseline genotype (e.g., WT = 0) would correspond to Y = b0 + b1 \* 0 + e. This is equivalent to Y = b0 (assuming that e is very small), thereby demonstrating why the intercept (b0) can be interpreted as the average of our baseline group. b1, on the other hand, will be the coefficient whose closeness to zero will be evaluated during the statistical testing step.

# Testing the null hypothesis

The null hypothesis is that there is no systematic difference between the average read count values of the different conditions for a given gene. Which test is used to assign a p-value again depends on the tool, but generally you can think of them as some variation of the well- known t−test (How dissimilar are the means of two populations?) or ANOVAs (How well does a reduced model capture the data when compared to the full model with all coefficients?).

Once you’ve obtained a list of p-values for all the genes of your data set, it is important to realize that you just performed the same type of test for thousands and thousands of genes. That means, that even if you decide to focus on genes with a p-value smaller than 0.05, if you’ve looked at 10,000 genes your final list may contain 0.0510, 000 = 500 false positive hits. To guard yourself against this, all the tools will offer some sort of correction for the multiple hypotheses you tested, e.g. in the form of the Benjamini-Hochberg formula. You should definitely rely on the adjusted p-values rather than the original ones to zoom into possible candidates for downstream analyses and follow-up studies.

# Running DGE analysis tools

## DESeq2 workflow

For our example data set, we would like to compare the effect of the Mov10 over expressed versus the control samples, with the control values used as the denominator for the fold change calculation.

* str(DESeq.ds$condition)
* DESeq.ds$condition <- relevel(DESeq.ds$condition, ref="Control ")

## Running the DE analysis

* DESeq.ds <- DESeq(DESeq.ds)

Extract the base means across samples, log2 fold changes, standard errors, test statistics, p- values and adjusted p-values for every gene using results().

# tells you which types of values can be extracted with results()

* resultsNames(DESeq.ds)
* DGE.results <- results(DESeq.ds,

independentFiltering = TRUE, alpha = 0.05)

# first line indicates which comparison was done for the log2FC

* head(DGE.results)
* summary(DGE.results)

# the DESeqResult object can basically be handled like a data.frame

* table(DGE.results$padj < 0.05)

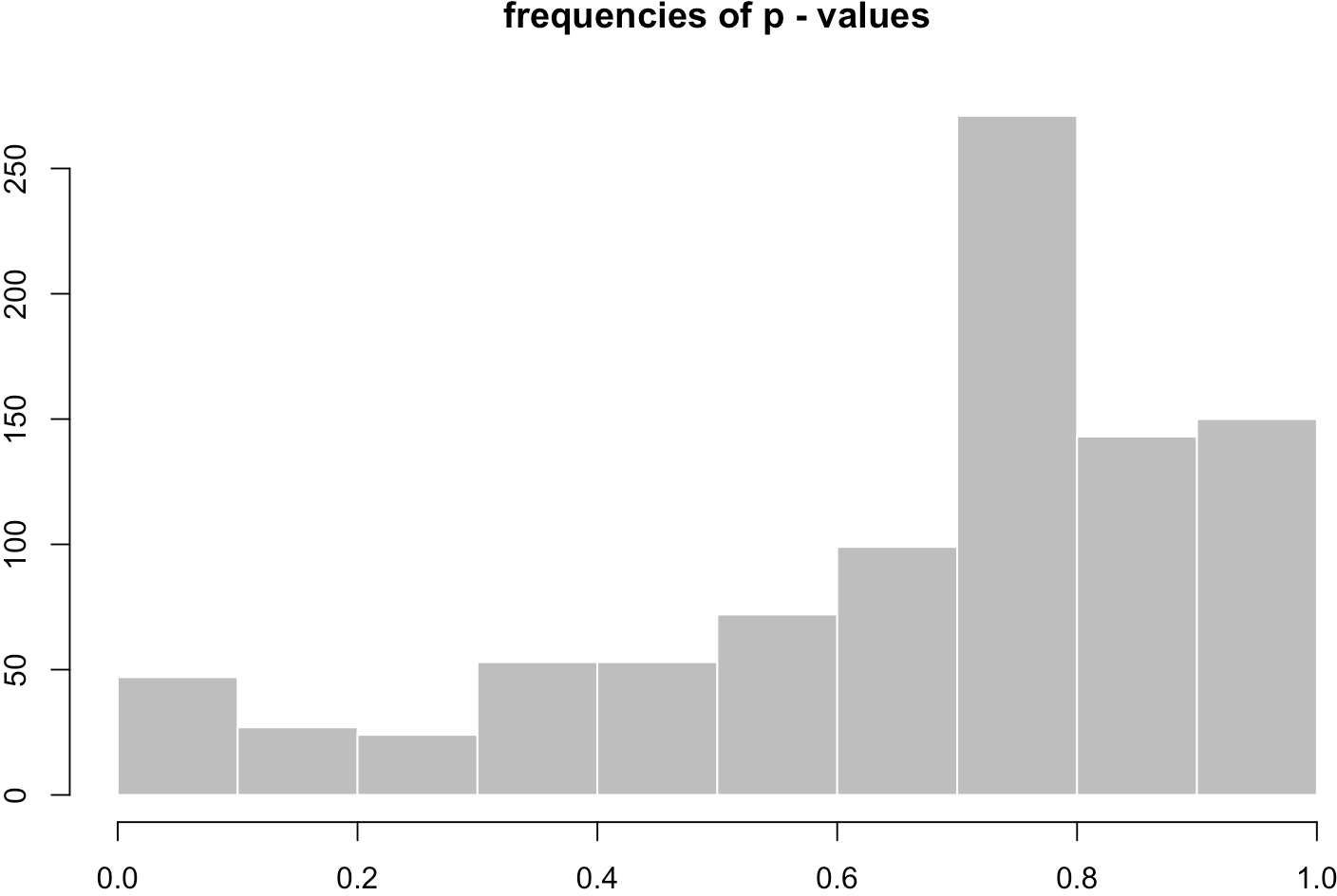
NAs in the padj column (but values in both log2FC and pvalue) are indicative of that gene being filtered out by the independent filtering [because it was very lowly expressed].

## Exploratory plots following DGE analysis

Histograms are a simple and fast way of getting a feeling for how frequently certain values are present in a data set. A common example is a histogram of p-values.

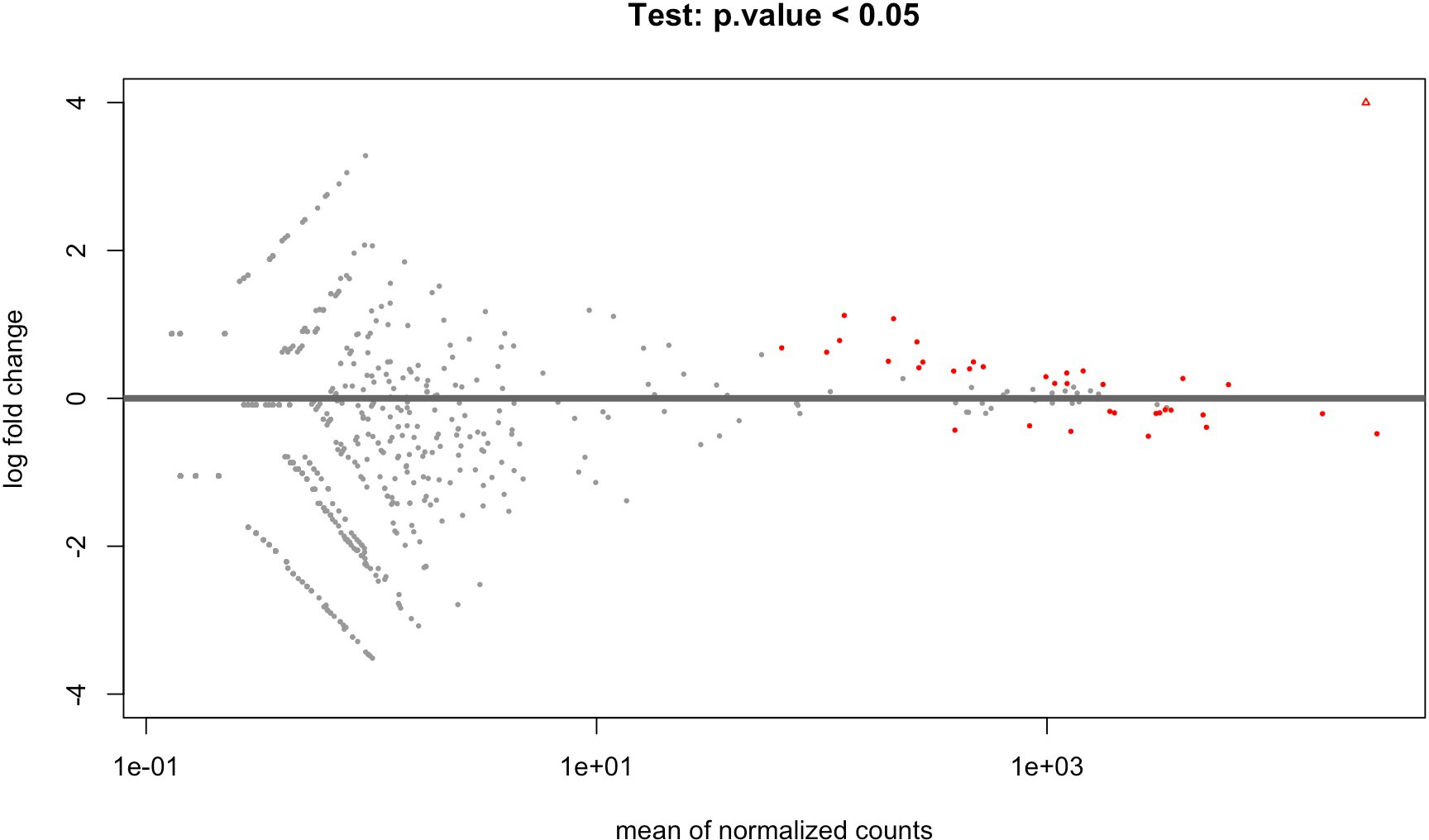
* hist(DGE.results$pvalue,

col = "grey" , border = "white" , xlab = " " , ylab = " " , main = "frequencies of p - values")



MA plots were originally developed for visualizing cDNA microarray results, but they are also useful for RNA-seq analyses. The MA plot provides a global view of the relationship between the expression change between conditions (log ratios, M), the average expression strength of the genes (average mean, A). The MA-plot provides a global view of the differential genes, with the log2 fold change on the y-axis over the mean of normalized counts. Genes that pass the significance threshold (adjusted p.value <0.05) are colored in red.

* plotMA(DGE.results, alpha = 0.05, colNonSig = "gray60", colSig = "red", main = "Test: p.value < 0.05", ylim = c(-4,4))



Heatmaps are a popular means to visualize the expression values across the individual samples. The following commands can be used to obtain heatmaps for rlog-normalized read counts for genes that show differential expression with adjusted p-values <0.05.

# load the library with the aheatmap () function

* install.packages("NMF")
* install.packages("pheatmap")
* library(NMF)
* library(pheatmap)

# aheatmap needs a matrix of values , e.g. , a matrix of DE genes with the # transformed read counts for each replicate

# sort the results according to the adjusted p- value

* DGE.results.sorted <- DGE.results[order(DGE.results$padj), ]

# identify genes with the desired adjusted p- value cut -off

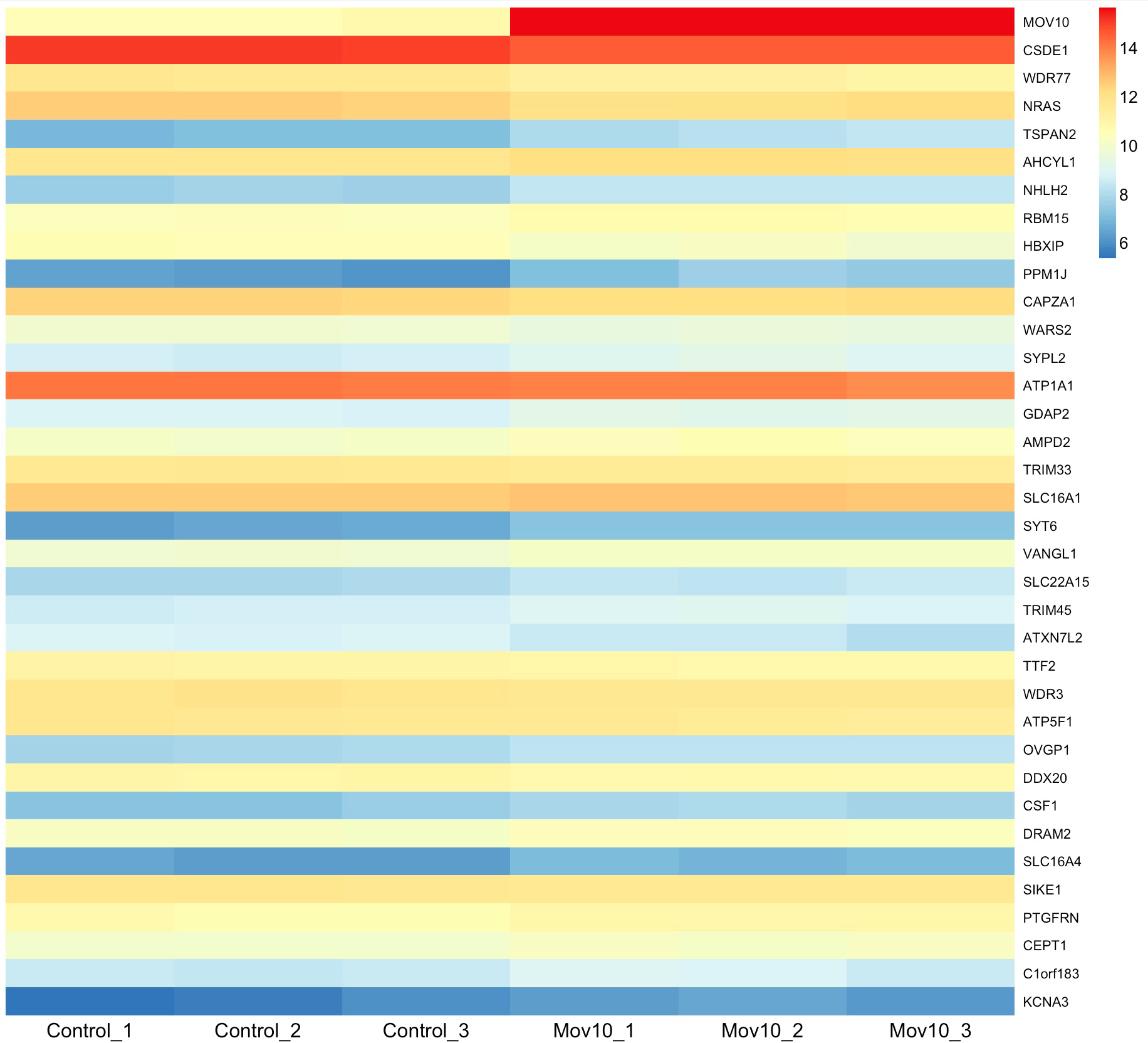
* DGEgenes <- rownames(subset(DGE.results.sorted , padj < 0.05))

# extract the normalized read counts for DE genes into a matrix

* hm.mat\_DGEgenes <- log.norm.counts[DGEgenes, ]

# plot the normalized read counts of DE genes sorted by the adjusted p-value

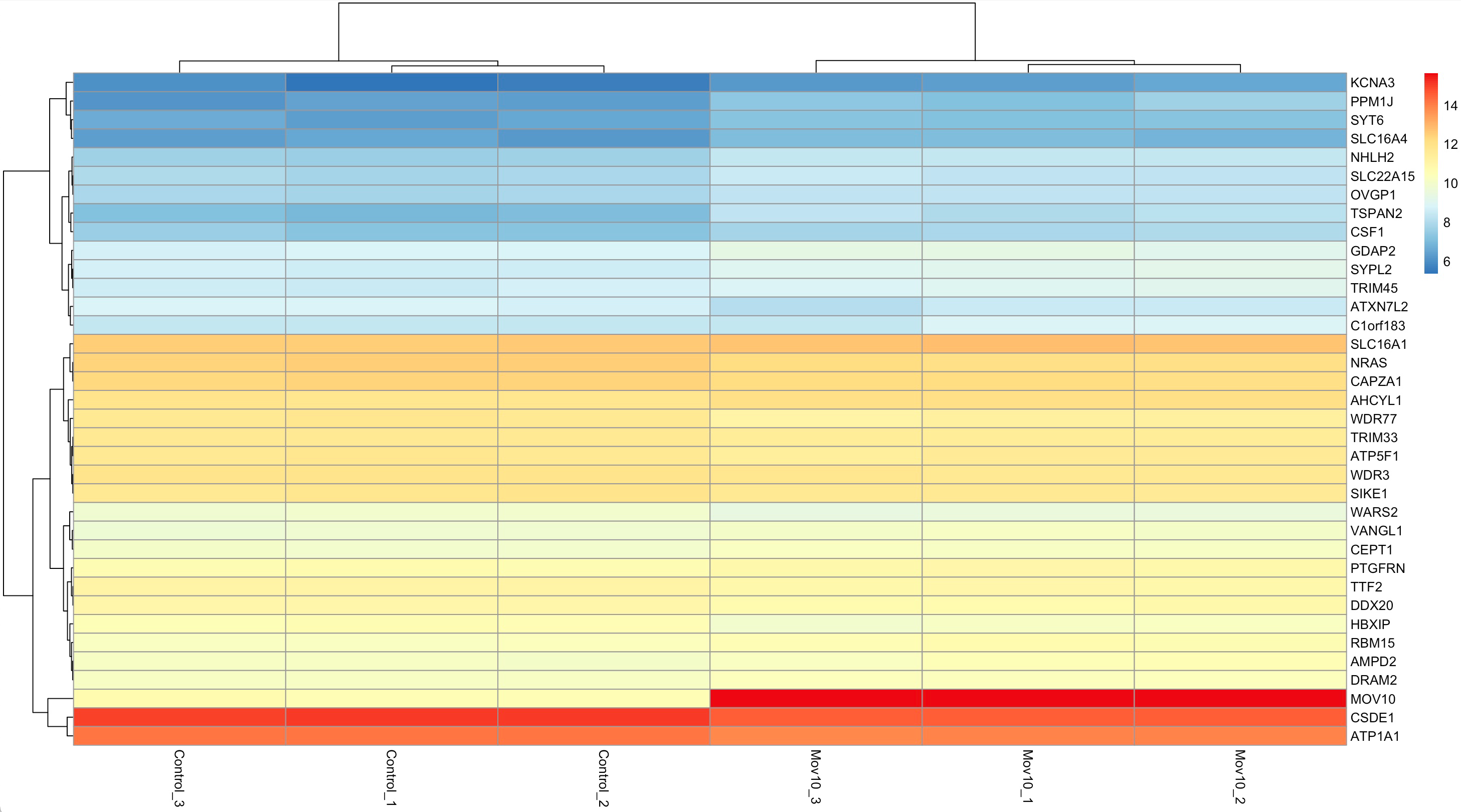
* pheatmap(hm.mat\_DGEgenes , Rowv = NA , Colv = NA)



# combine the heatmap with hierarchical clustering # add dendrograms to rows and column

* pheatmap(hm.mat\_DGEgenes, Rowv = TRUE , Colv = TRUE,

distfun = "euclidean" , hclustfun = "average")



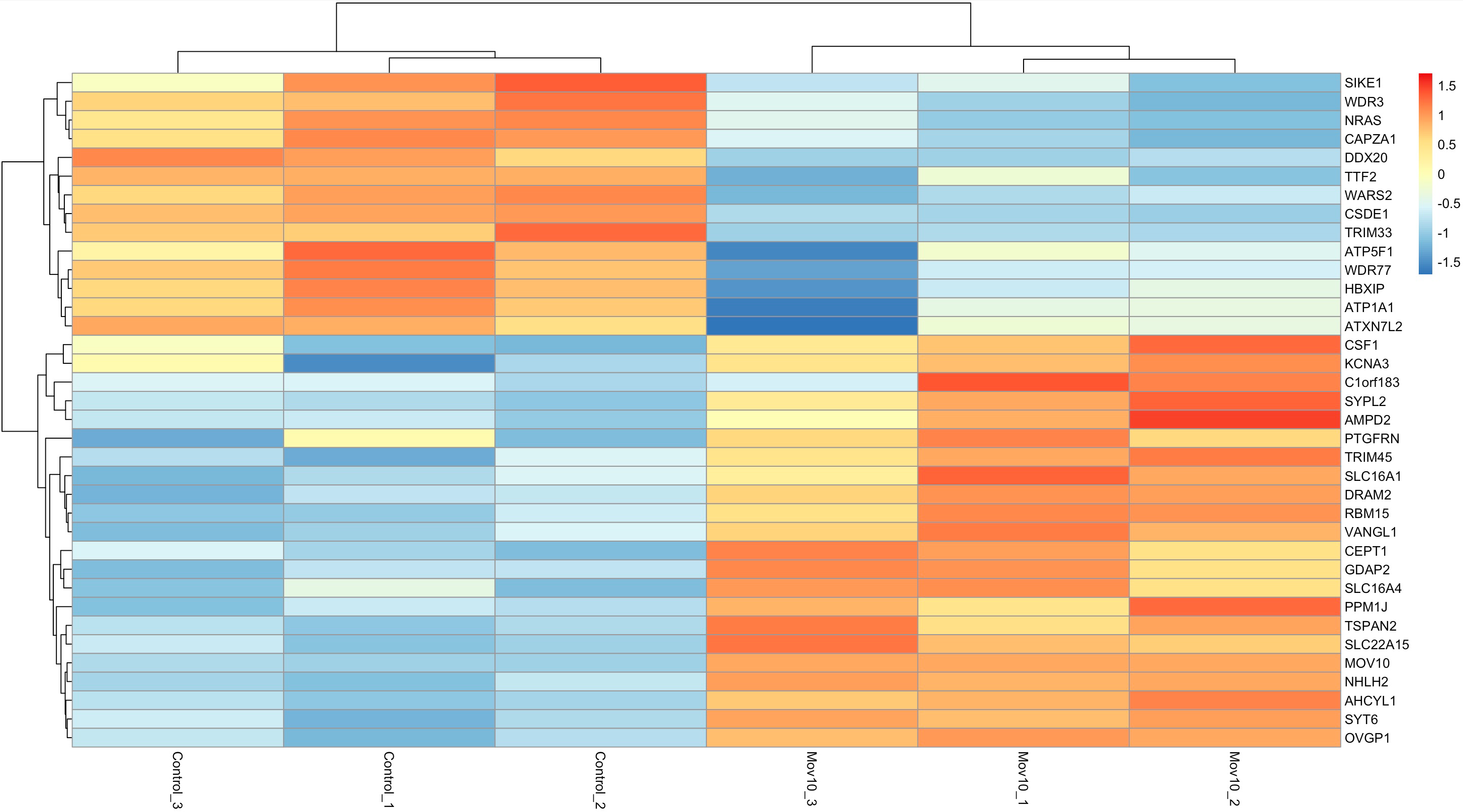
# scale the read counts per gene to emphasize the sample-type-specific # differences

* pheatmap (hm.mat\_DGEgenes , Rowv = TRUE , Colv = TRUE ,

distfun = "euclidean" , hclustfun = "average", scale = "row" )

# values are transformed into distances from the center

# of the row - specific average : ( actual value - mean of the group ) / # standard deviation



## Read counts of single genes

An important sanity check of your data and the DGE analysis is to see whether genes about which you have prior knowledge behave as expected. For example, the samples named "Mov10" were generated from over expressed the mov10 gene, so mov10 should be among the most strongly upregulated genes in this DGE analysis.

To check whether mov10 expression is abundant in the contrast group, we first need to map the gene "symbol" that we used for generating the read count matrix to the full gene name and gene ontology (GO) terms so that we can retrieve the rlog-transformed read counts and the moderated log2 fold changes. There is more than one way to obtain annotation data, here we will use a data base that can be directly accessed from within R. The website <https://www.bioconductor.org/packages/release/data/annotation/>lists all annotation packages that are available through bioconductor. For human data, we will use org.Hs.eg.db.

* BiocManager::install("org.Hs.eg.db")
* library(org.Hs.eg.db)

# list columns that can be retrieved from the annotation data base

* columns(org.Hs.eg.db)

# make a batch retrieval for all DE genes

* anno <- AnnotationDbi::select(org.Hs.eg.db,

keys = DGEgenes,keytype ="SYMBOL", columns = c("GENENAME","GO"))

# check whether Mov10 pops up among the top upregulated genes

* DGE.results.sorted\_logFC<-DGE.results[order(DGE.results$log2FoldChange), ]
* DGEgenes\_logFC <- rownames (subset(DGE.results.sorted\_logFC, padj < 0.05))
* anno[match(DGEgenes\_logFC, anno$SYMBOL), ]

# Gene set enrichment of KEGG pathways using

**ClusterProfiler**

ClusterProfiler actually offers access to a fairly sprawling set of R packages that all deal with over-representation analyses and gene set enrichment analyses. Now we focus on highlighting a function that allows you to perform GSEA using the pathways annotated and curated by [KEGG](https://www.genome.jp/kegg/pathway.html).

To this end, you will need to provide a full list of all genes that you analyzed. Those genes should be sorted by some sort of metric that you feel is a meaningful representation of the signal you're after, e.g. log2FC or a combination of log2FC and p-value or the like. This sorted vector of gene-value pairs is then compared to the KEGG pathways and those pathways that seem to have a consistent enrichment for positive (or negative) fold changes across all of the pathway's genes will be highlighted.

The main advantage of using clusterProfiler comes from its implementation of [numerous](https://yulab-smu.github.io/clusterProfiler-book/chapter12.html) [helpful plots](https://yulab-smu.github.io/clusterProfiler-book/chapter12.html). In addition, it takes care of downloading the appropriate annotation from KEGG (or any other data base for which it offers functionalities).

* BiocManager::install("clusterProfiler")
* BiocManager::install("ggnewscale")
* library(clusterProfiler)

# clusterProfiler requires a sorted vector where the values correspond # to the measure used for the sorting

* DGE.results <- DGE.results[order(-1\*DGE.results$log2FoldChange),]
* genes\_for\_cp <- DGE.results$log2FoldChange
* names(genes\_for\_cp) <- row.names(DGE.results)

## run the gene set enrichment analysis

* gse <- gseGO(geneList=genes\_for\_cp,

ont ="ALL",

keyType = "SYMBOL", nPerm = 10000,

minGSSize = 3,

maxGSSize = 800,

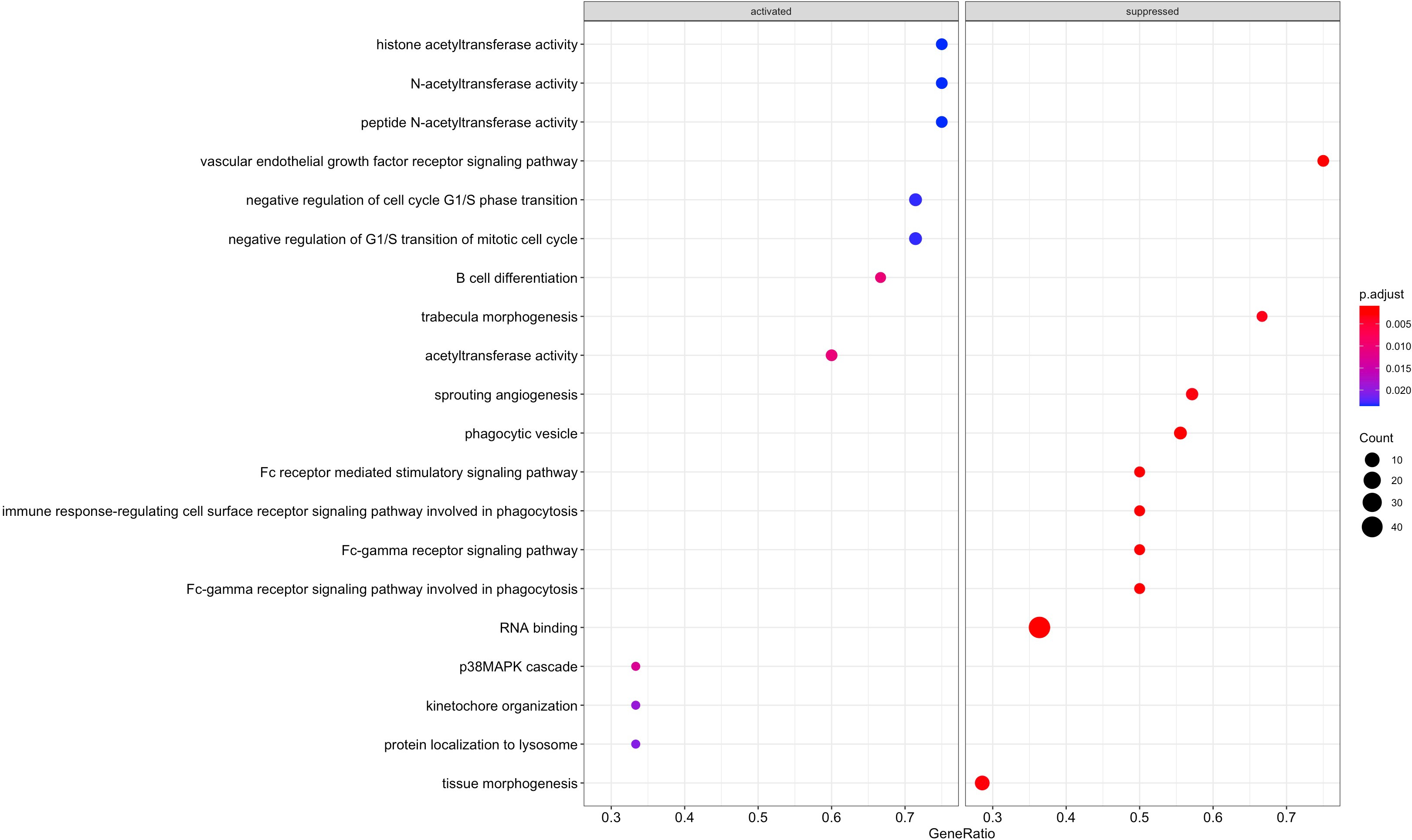
pvalueCutoff = 1, verbose = TRUE,

OrgDb = org.Hs.eg.db , pAdjustMethod = "none" )

### Dot plots:

Dot plots depict the enrichment scores and gene counts per gene set (for the most significant gene sets).

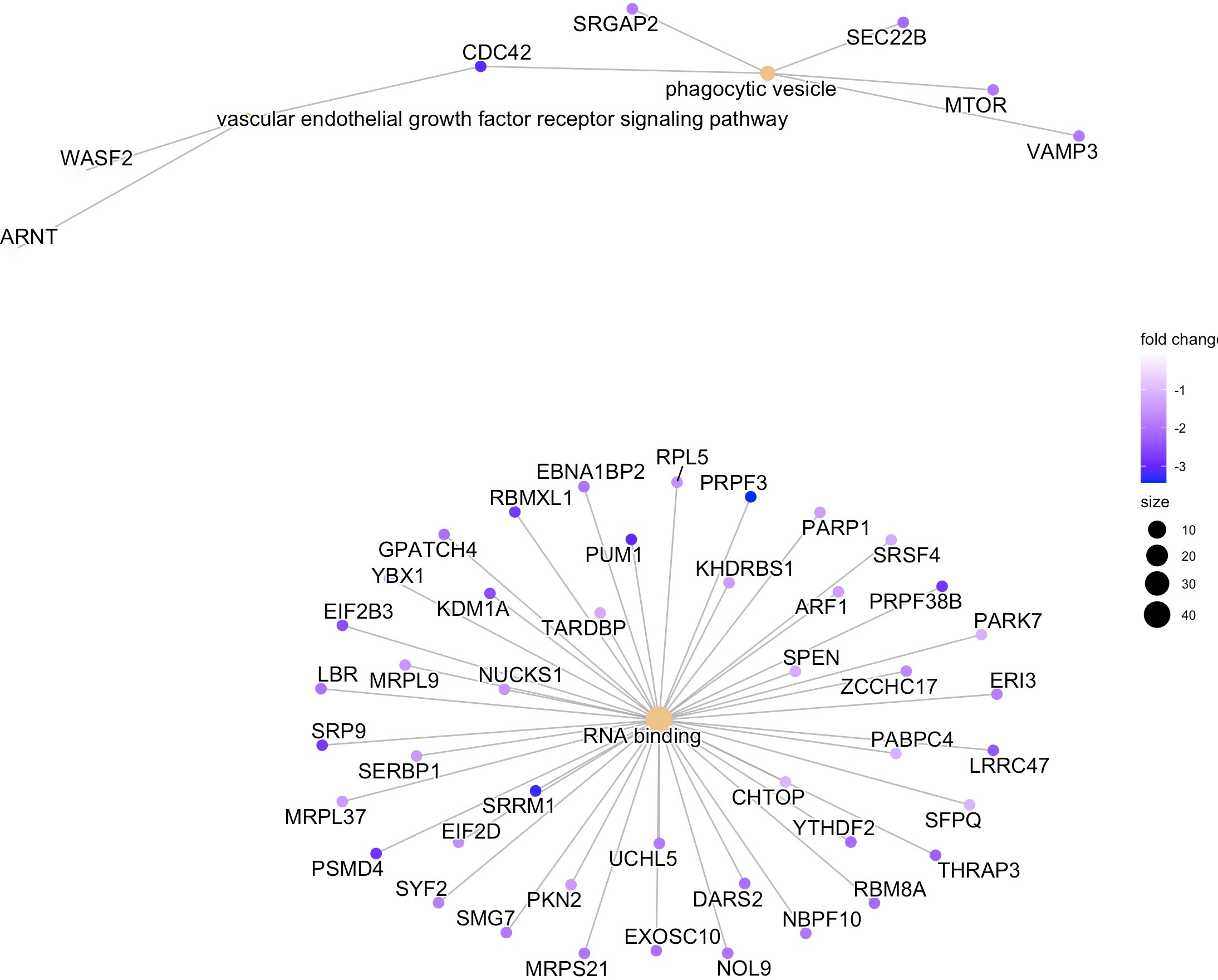
* require(DOSE)
* dotplot(gse, showCategory=10, split=".sign") + facet\_grid(~.sign)



### Cnetplots:

Cnetplots depict the linkages of genes and biological concepts (e.g. GO terms or KEGG pathways) as a network.

* + nodes = genes (of the top 5 most sign. GO terms by default)
  + edges = indicate whether a gene belongs to a given gene set
  + size of the GO terms = number of genes belong to a given category
* cnetplot(gse, categorySize="pvalue", foldChange=genes\_for\_cp, showCategory = 3)



When you finish this lab session, you can delete these packages to save your storage space in your computer:

* remove.packages(c("DEseq2","ggplot2","pheatmap","NMF","clusterProfiler","ggnewscale"))
* remove.packages(c("org.Hs.eg,db"))

# Assignment for Lab 11

### Please answer the following questions:

Combined with the study described in [Kenny PJ et al., Cell Rep 2014](https://www.ncbi.nlm.nih.gov/pubmed/25464849) to analysis the generated dot plot and cnetplots, and explain why the path of "RNA binding" is suppressed.

(Answer: This result shows genes related to the process of "RNA binding" are suppressed, which makes sense, as RNA helicase MOV10 has been implicated in miRNA-mediated translational suppression and has an effect on the fate of its bound RNAs: usually facilitating translation suppression, reflecting its role in the miRNA pathway. )

# Reference

These lab materials are derived from the following website:

1. <http://www.ncbi.nlm.nih.gov/pubmed/25464849>
2. <https://chagall.med.cornell.edu/RNASEQcourse/Intro2RNAseq.pdf>
3. <https://github.com/friedue/course_RNA-seq2019/tree/master/Day04>