Analysis Of Senescence Transcriptome

Xiaoyan Wen

Abstract

This report includes quality control and normalization for raw count data (part 1). The results of differential expression analysis were selected using the criteria of fold change > 2 and adjusted p-value < 0.01. To validate the DE analysis, plot count for top genes that have minimum adjusted p-value in each comparison, MA plot and volcano plot were drawn (part 2). For the list of DEGs, we generated a dendrogram and heatmap for visualization. And use both supervised and unsupervised methods to further cluster the data (part 3). The top 10 enrichment elements of each cluster are reported in part 4.

Introduction

Cellular senescence is an important phenomenon that contributes not only to neoplastic transformation but also to aging and pathophysiological processes associated with cellular phenotype transition/transformation during tissue injury and repair. It is an interesting topic in the mechanism of sepsis-associated organ injury and early diagnostic markers for prognosis. Senescence is one of the hot study focuses, characterized by an irreversible cell cycle arrested, associated with special phenotype changes in the post-injury cells upon certain extracellular stimuli. Studies have shown that senescent cell-secreted proteins not only are sensitive markers for early injury but also could mediate multiple signal cascades involved in subsequence pathophysiologic processes. Despite its importance, the mechanism of senescence and its regulation are largely unknown. This report will hopefully provide a novel insight on senescence signal cascades through analyzing RNAseq data (Series GSE153921)[1] obtained from senescence inducible human fibroblast [2],

Results

Part 1. Data quality control and normalization

Method

After dropping rows with criteria of rowsum < 10 & gene symbol == “NA”, size factor estimation and dispersion estimation were performed, and three methods log2, VST (the variance stabilizing transformation), rlog (the regularized logarithm transformation) were used for normalization.

Results

To reduce the size of the nonsense data and increase the speed, we removed the rows with rowsum < 10 & with no information about the gene. We thus reduced the row number from 62,161 to 17,227. Due to the possibility of different sequencing depths across samples and variability across biological replicates, size factor estimation and dispersion estimation were performed using estimateSizeFactors() and estimateDispersions() functions in the DESeq2 package. The dispersion plot shows that given gene expression mean value, the variances are associated with estimate means (figure 1). We used three methods log2, VST, and rlog to normalize the count data, with the last two performed using vst() and rlog() function offered by the DESeq2 package. The side-by-side boxplot and meanSD plot show normalization soothe out the variation (figure 2-3).

Figures

Figure 1. Dispersion plot for gene count matrix

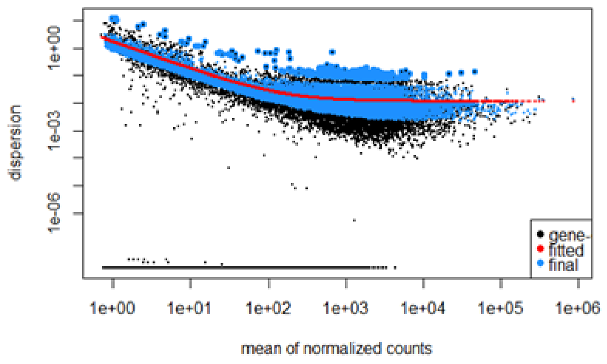


Figure 2. boxplots comparing normalization methods.

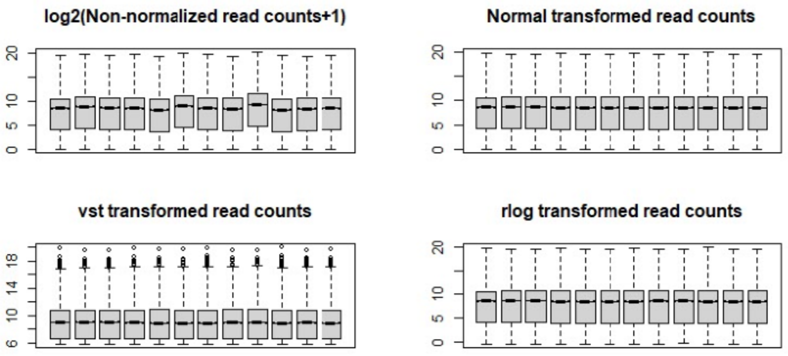
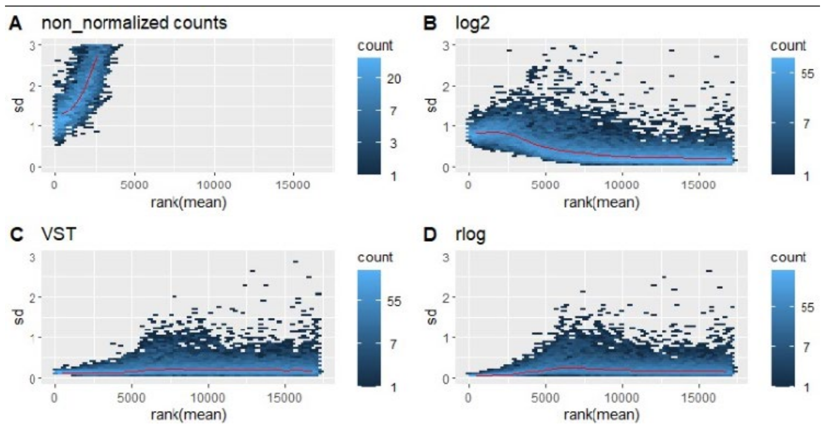


Figure 3. meanSD plot comparing normalization methods



Part 2. Identify Genes differentially expressed

Method

Differential expression analysis was performed using function DEseq() in the DESeq2 package. Because we want to explore the group difference of "ko" given senescence "induced" and "non_induced" condition, an interactive ~group were added to the model design, and thus six paired groups were able to be compared (Table 1).

Results

The results of differential expression were picked using the criteria padj<.01 & log2FoldChange > 1 and ended with 6684 significant rows (2774 unique genes) in total for six paired comparisons. To validate the results, count plots for top genes that have minimum adjusted p-value, MA plot, and volcano plot were drawn for three of all comparisons (figure 4-6).

Table 1: Interactive comparisons

Group compare	Wald test FDR_adjusted p-value < 0.1	
group inducedko shXPO7 5 vs inducedko shXPO7 3	LFC > 0 (up) : 21, 0.12%	LFC < 0 (down) : 13, 0.075%
group inducedno ko vs inducedko shXPO7 3	LFC > 0 (up) : 28, 0.16%	LFC < 0 (down) : 86, 0.5%
group non inducedcontrol vs inducedko shXPO7 3	LFC > 0 (up) : 3986, 23%	LFC < 0 (down) : 4074, 24%
group inducedno_ko vs non_inducedcontrol	LFC > 0 (up) : 4037, 23%	LFC < 0 (down) : 4106, 24%
group inducedko_shXPO7_5 vs non_inducedcontrol	LFC > 0 (up) : 4158, 24%	LFC < 0 (down) : 4177, 24%
group inducedko_top 10 vs inducedno_ko	LFC > 0 (up) : 556, 3.2%	LFC < 0 (down) : 213, 1.2%

Figures

Figure 4. representative count plot for top DE gene that has minimal adjp

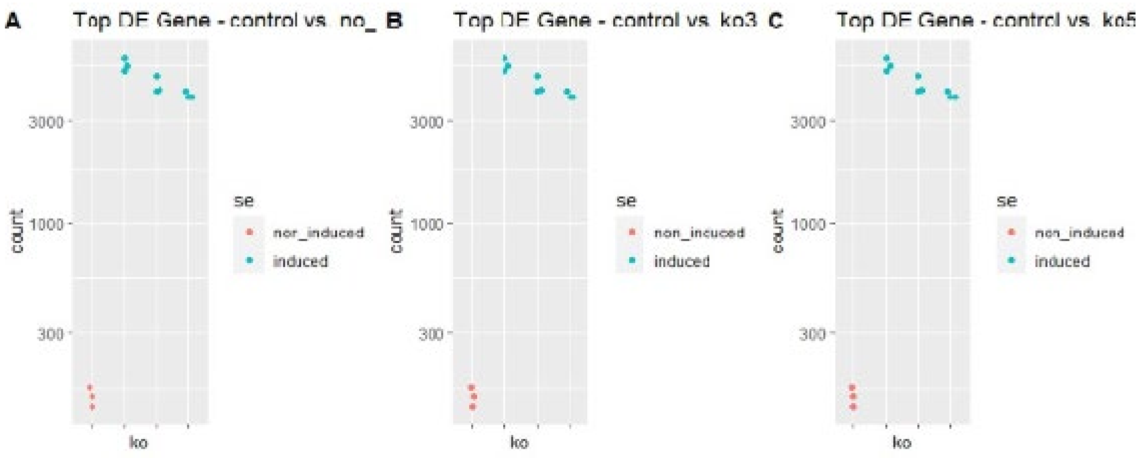


Figure 5. representative MA plot for paired comparisons

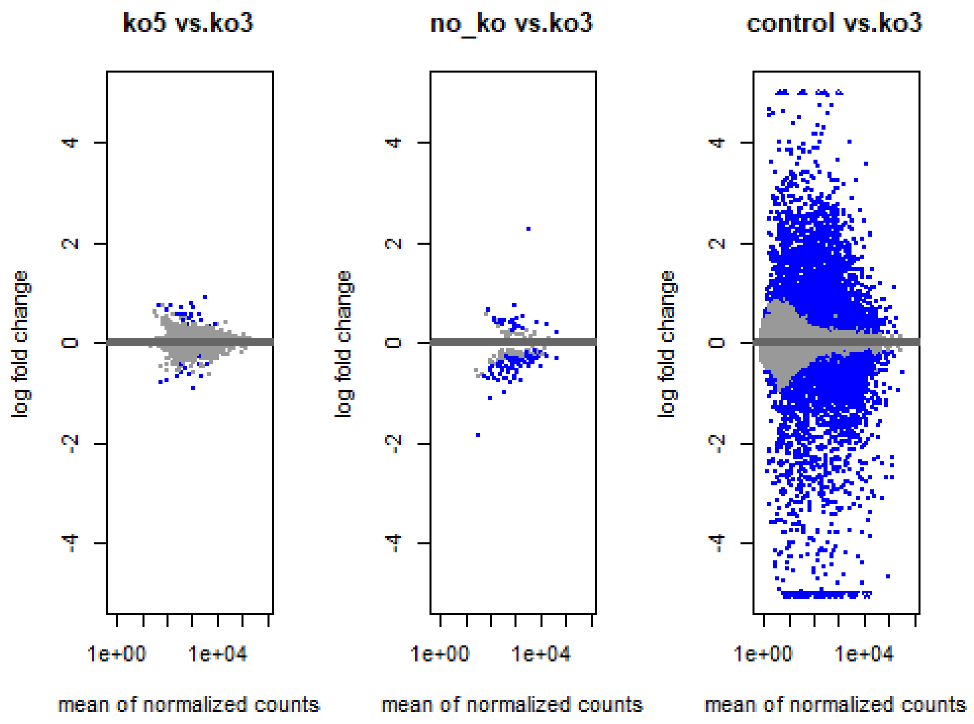
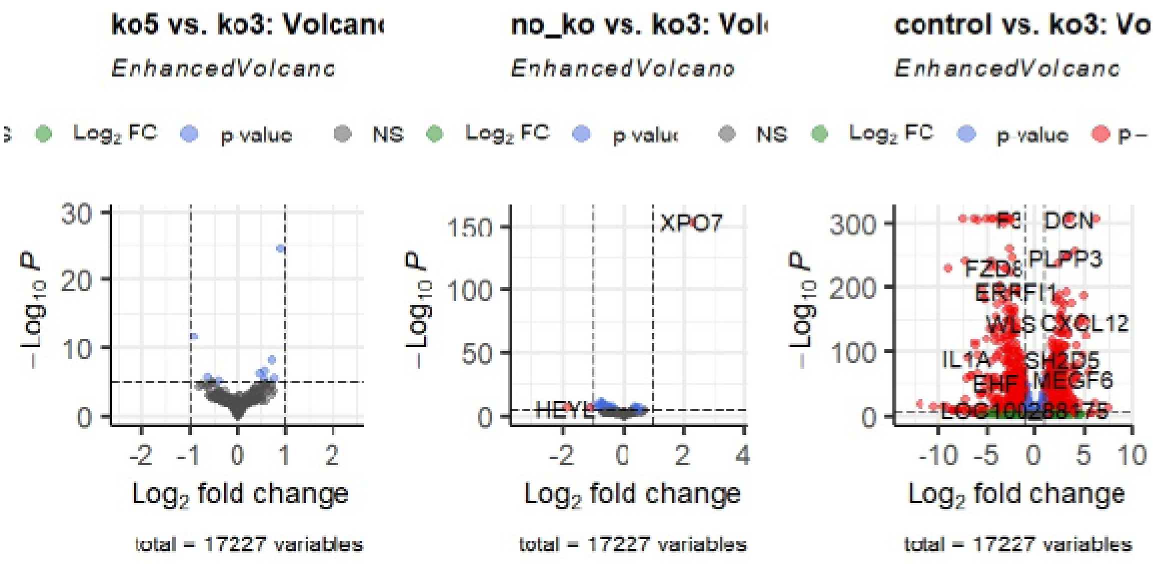


Figure 6. representative volcano plot for paired comparisons



Part 3. Cluster differentially expressed genes

Method

For the DEGs list, dendrogram and heatmap were built for samples and gene-rows based on Pearson correlation. Cluster number was determined by both supervised k-mean (with the k number computed using silhouette-wide value) and unsupervised PCA method.

Results

From dendrogram and PCA for samples, we found a clear separation between senescence-induced and non-induced samples, but not among XPO7 knock-out and non-knockout ones (figure 7, 9). Both silhouette-wide value guided k-mean and PCA results in cluster number as two (figure 8-9). The cluster results could be visualized on a dendrogram and heatmap graph (figure 10-11).

Figures

Figure 7. dendrogram for samples

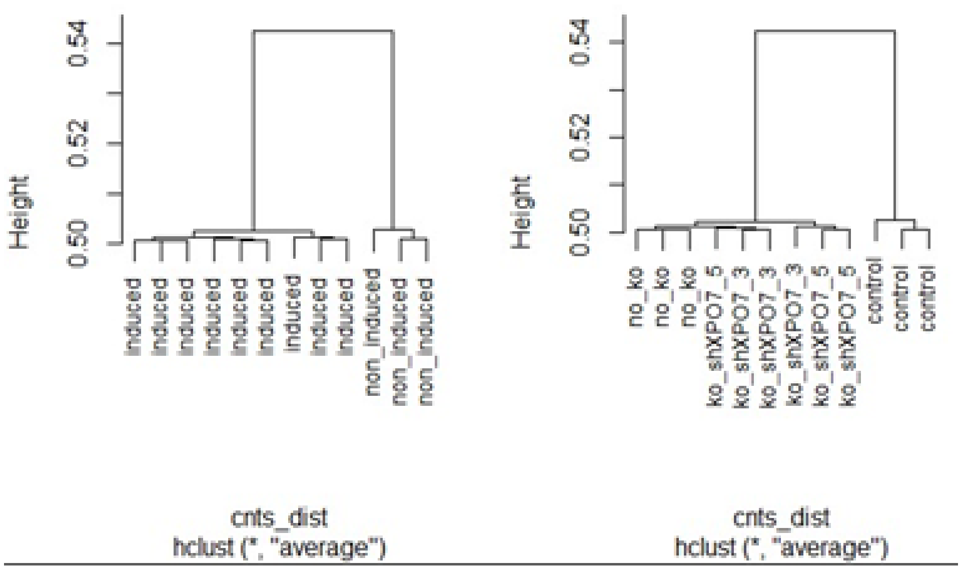


Figure 8. silhouette-wide value guided k-mean cluster number = 2

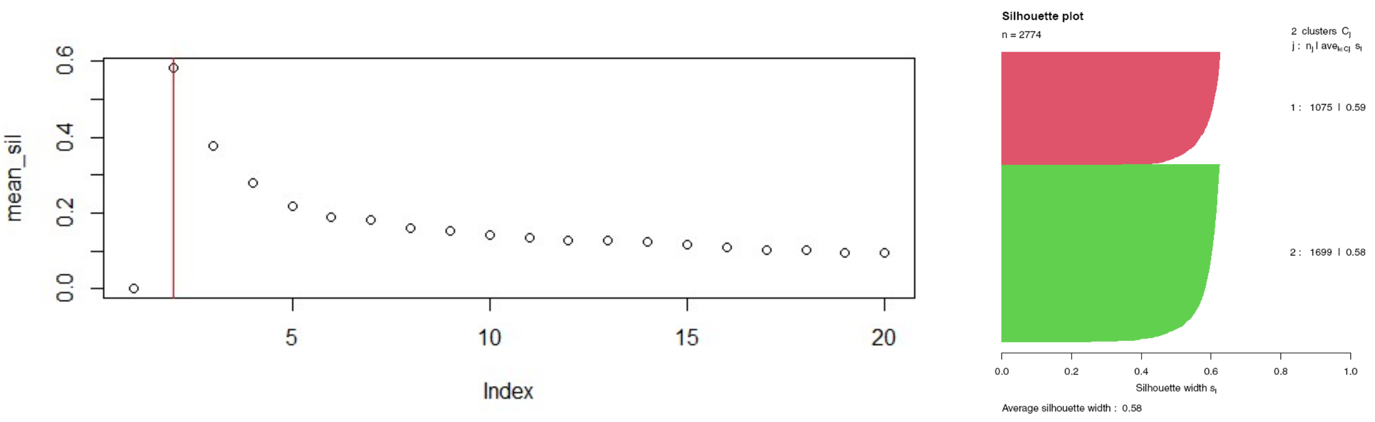


Figure 9. PCA clustering for samples and genes

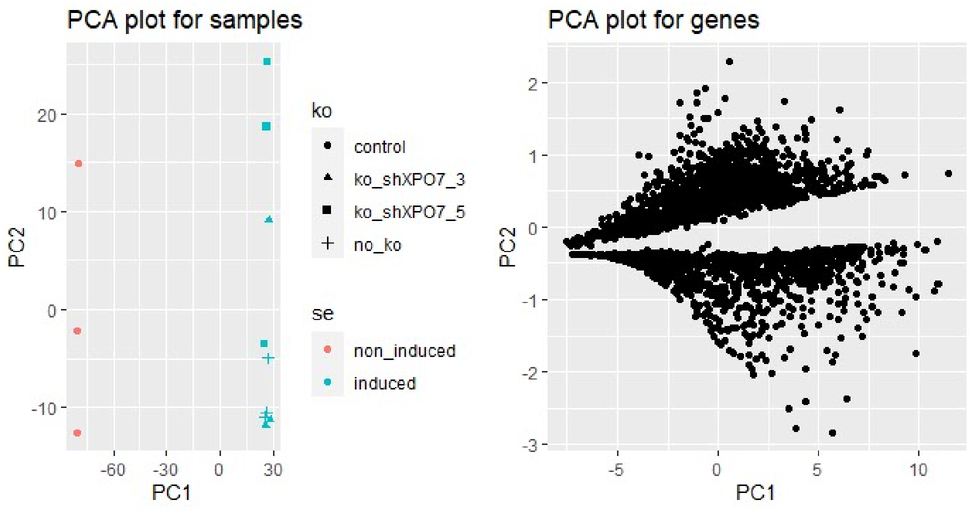


Figure 10. visualize clusters using dendrogram

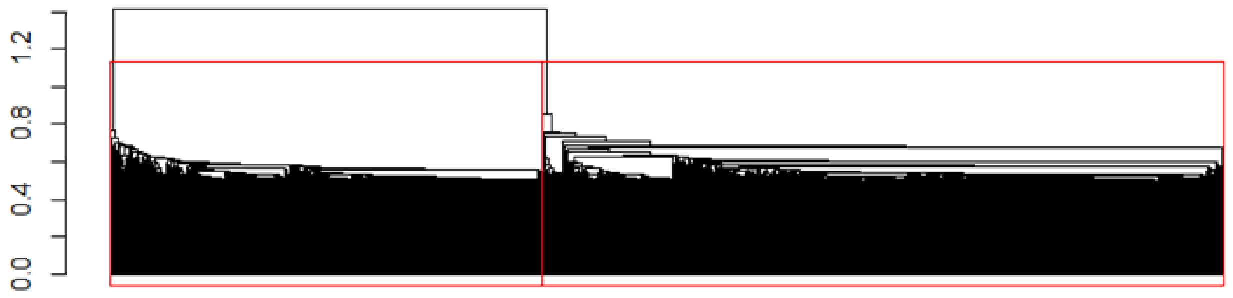
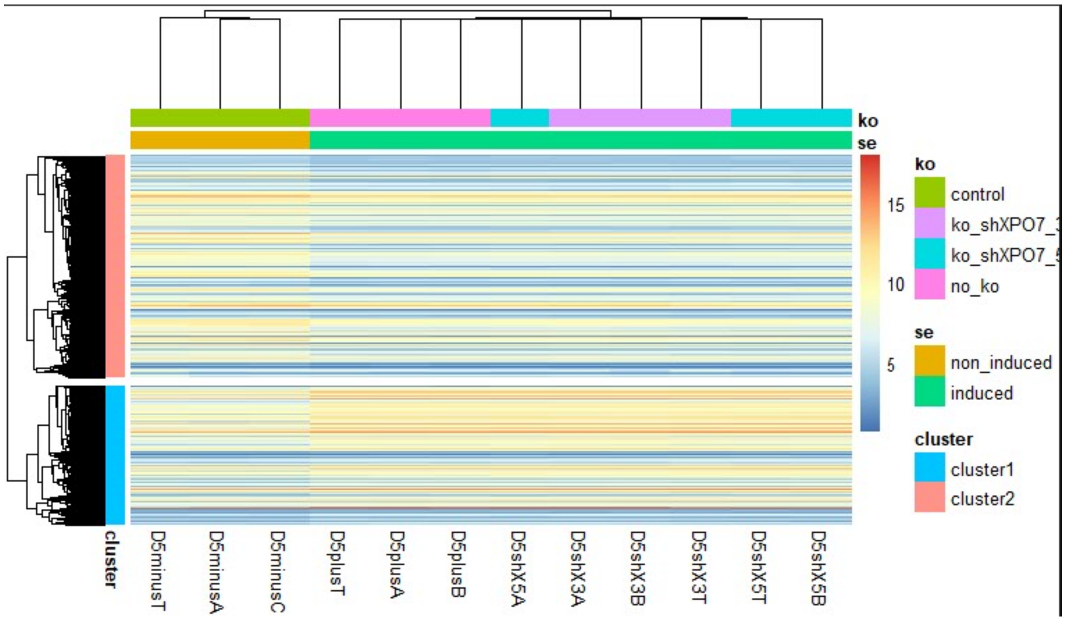


Figure 11. visualize clusters using heatmap



Part 4. GO-term analysis of differentially expressed genes and clusters

Method

The gene ontology enrichment analysis and GSEA for DEGs and clusters were performed using the clusterProfiler package. To do that, genes were first sorted by fold-increase in decreasing order. Because it performs multiple comparisons, p values were adjusted by the FDR method.

Results

GO-term enrichment analysis

Dot plots representing the top 10 enriched gene sets result from the enrichment analysis and enrichment map network for DEGs and clusters were shown (figure 12-17).

GSEA analysis

For DEGs, the activated gene sets and pathway enrichment distribution with adjusted p-value were shown in Figures 18-19. DSEA plot shows the top-ranked genes were enriched for the reproduction elements (figure 20).

For cluster 1, the activated gene sets and pathway enrichment distribution with adjusted p-value were shown in Figures 21-22. DSEA plot shows the top-ranked genes were enriched for the immune system process elements (figure 23).

For cluster 2, the activated gene sets and pathway enrichment distribution with adjusted p-value were shown in figure 24-25. DSEA plot shows the ranked genes were enriched for the reproduction elements (figure 26).

Figures

Figure 12. top 10 enriched gene sets result from the enrichment analysis for DEGs

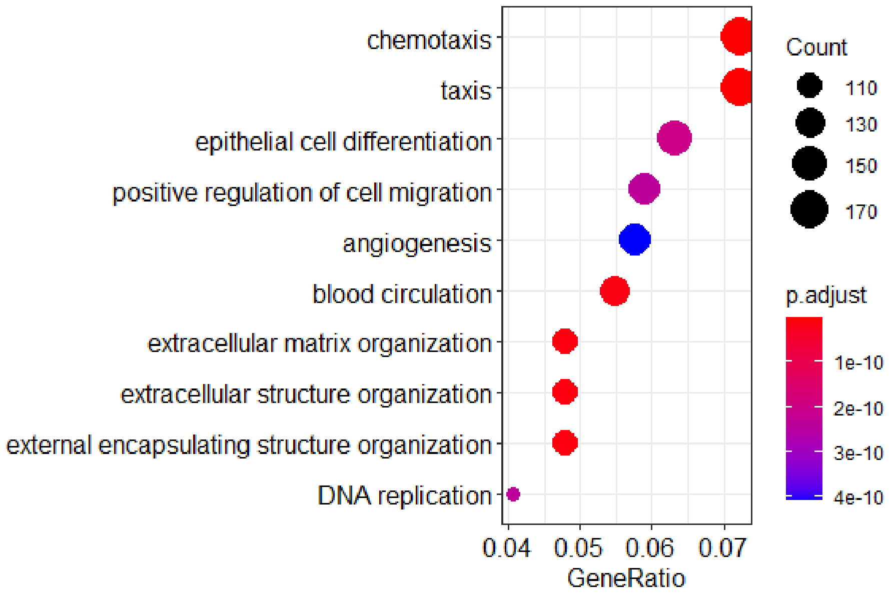


Figure 13. enrichment map network for DEGS

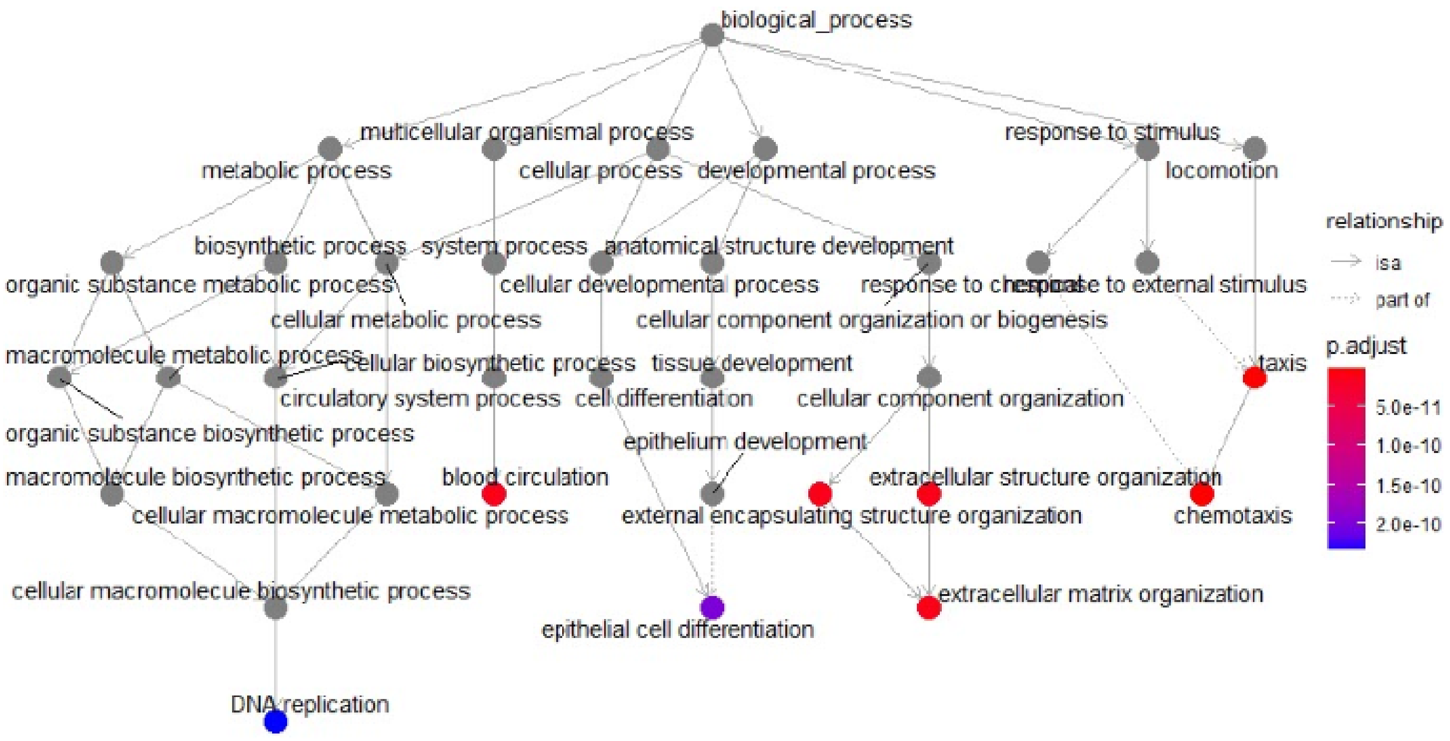

biological_process
multicellular organismal process
response to stimulus
metabolic process
cellular process
developmental process
locomotion
biosynthetic process
system process
anatomical structure development
organic substance metabolic process
cellular developmental process
response to chemical
response to external stimulus
cellular metabolic process
cellular component organization or biogenesis
macromolecule metabolic process
cellular biosynthetic process
tissue development
taxis
circulatory system process
cell differentiation
cellular component organization
organic substance biosynthetic process
epithelium development
macromolecule biosynthetic process
blood circulation
extracellular structure organization
cellular macromolecule metabolic process
external encapsulating structure organization
chemotaxis
cellular macromolecule biosynthetic process
extracellular matrix organization
epithelial cell differentiation
DNA replication
relationship
isa
part of
p.adjust
5.0e-11
1.0e-10
1.5e-10
2.0e-10

Figure 14. top 10 enriched gene sets result from the enrichment analysis for cluster 1

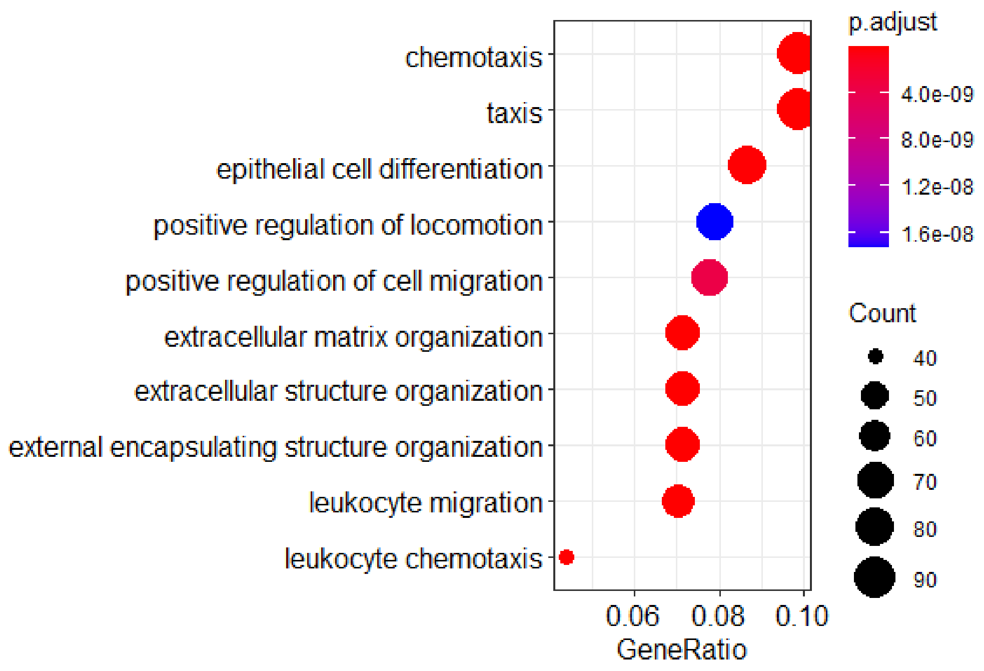


Figure 15. enrichment map network for cluster 1

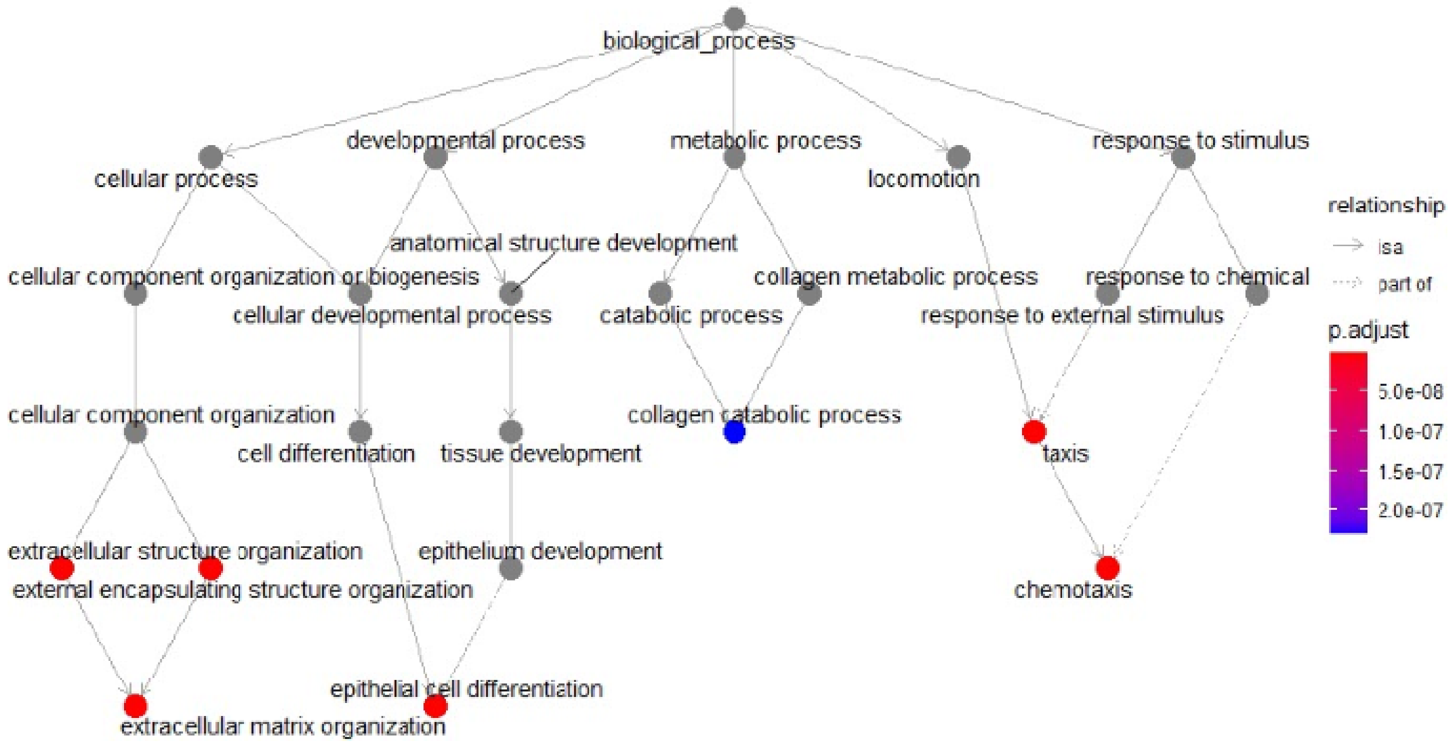

biological_process
cellular process
developmental process
metabolic process
locomotion
response to stimulus
anatomical structure development
cellular component organization or biogenesis
cellular developmental process
collagen metabolic process
catabolic process
response to chemical
response to external stimulus
cellular component organization
cell differentiation
tissue development
collagen catabolic process
taxis
extracellular structure organization
external encapsulating structure organization
epithelium development
chemotaxis
epithelial cell differentiation
extracellular matrix organization
relationship
isa
part of
p.adjust
5.0e-08
1.0e-07
1.5e-07
2.0e-07

Figure 16. top 10 enriched gene sets result from the enrichment analysis for cluster 2

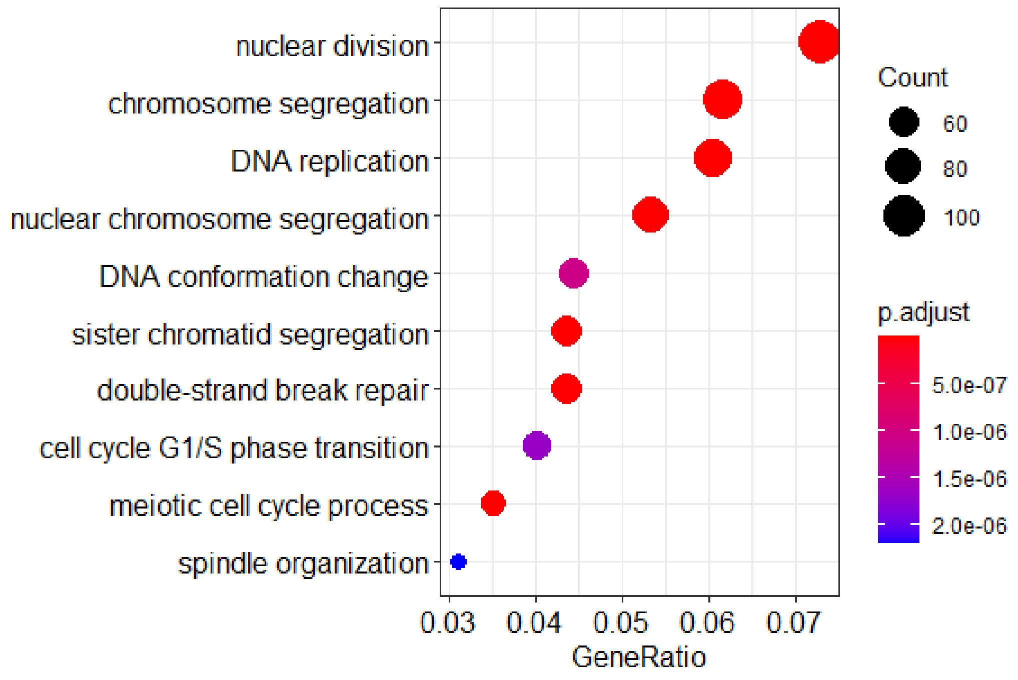


Figure 17. enrichment map network for cluster 2

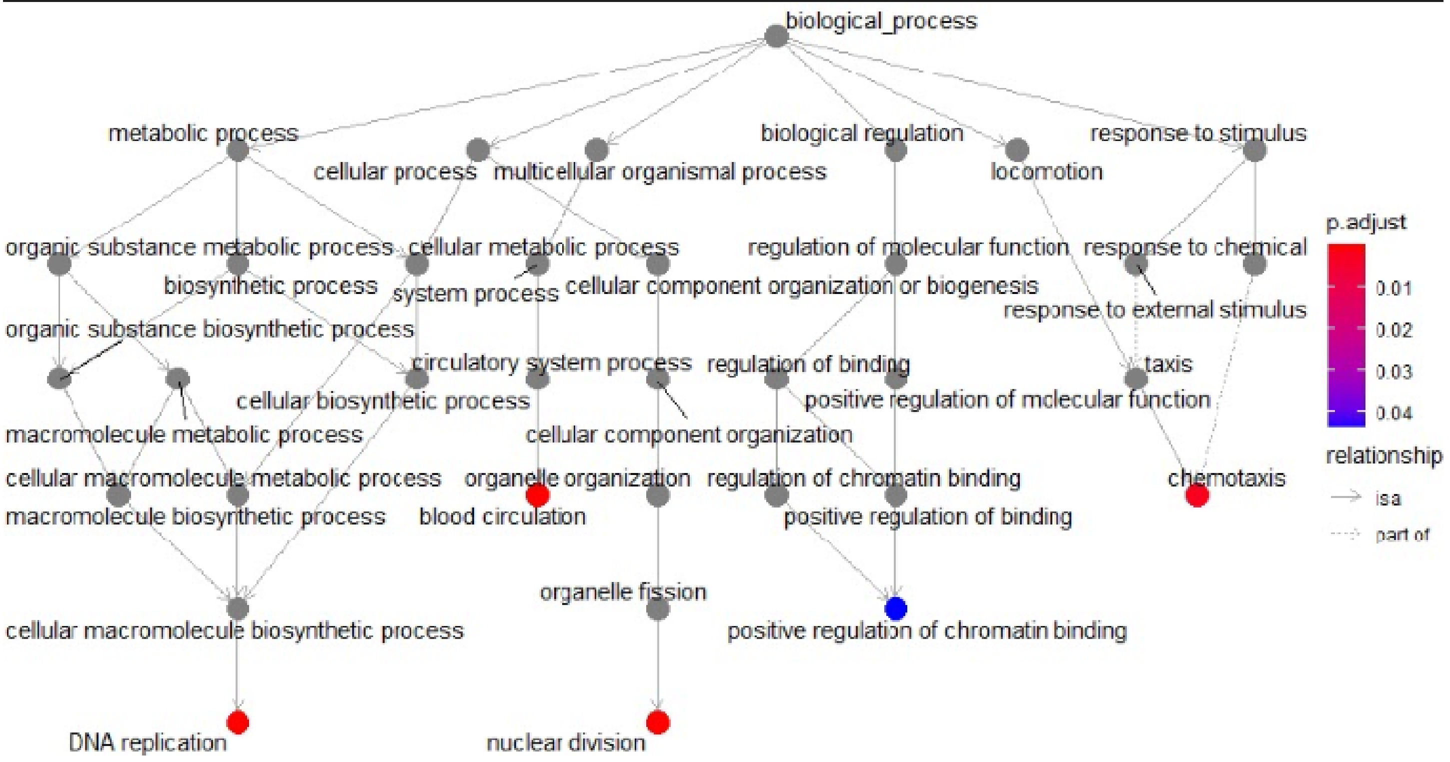

biological_process
metabolic process
cellular process
multicellular organismal process
biological regulation
locomotion
response to stimulus
organic substance metabolic process
cellular metabolic process
regulation of molecular function
response to chemical
biosynthetic process
system process
cellular component organization or biogenesis
response to external stimulus
organic substance biosynthetic process
circulatory system process
regulation of binding
taxis
cellular biosynthetic process
positive regulation of molecular function
macromolecule metabolic process
cellular component organization
cellular macromolecule metabolic process
organelle organization
regulation of chromatin binding
chemotaxis
macromolecule biosynthetic process
blood circulation
positive regulation of binding
organelle fission
cellular macromolecule biosynthetic process
positive regulation of chromatin binding
DNA replication
nuclear division
p.adjust
0.01
0.02
0.03
0.04
relationship
isa
part of

Figure 18. activate gene sets for DEGs

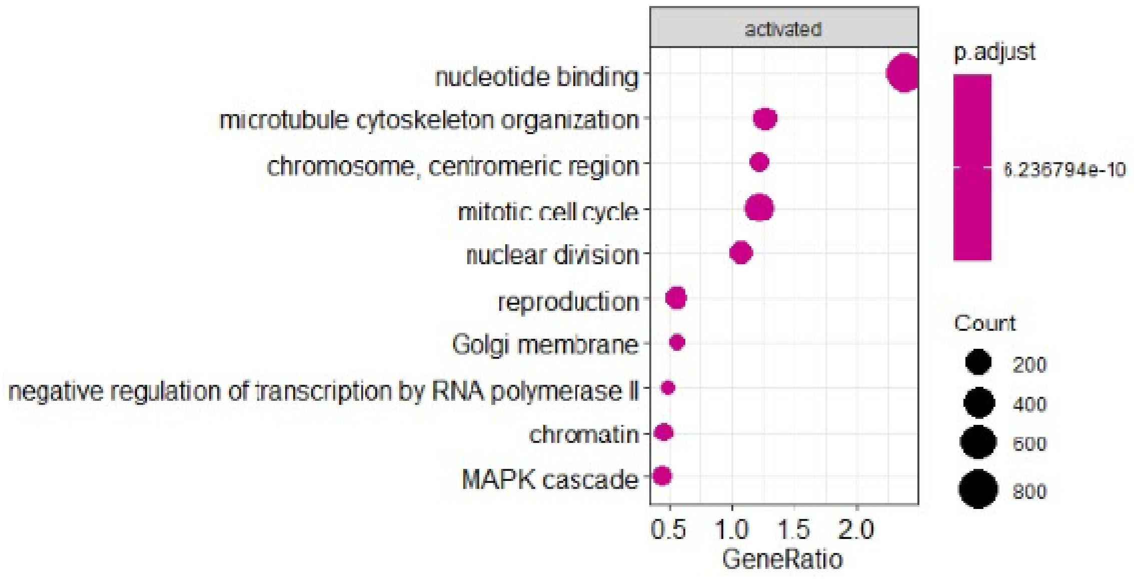


Figure 19. pathway enrichment distribution for DEGs

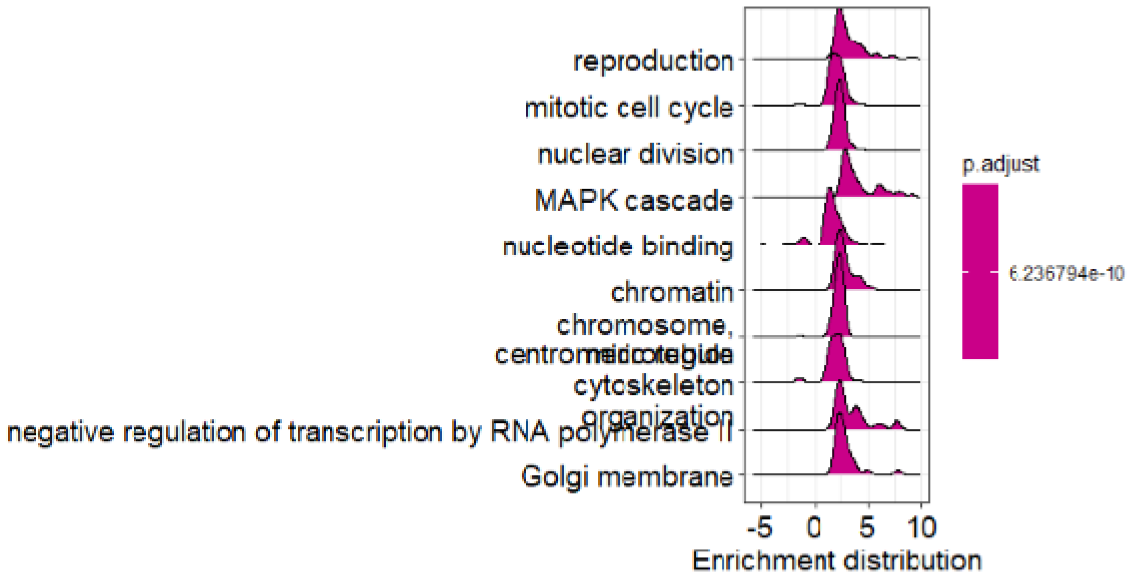


Figure 20. GSEA plot for DEGs

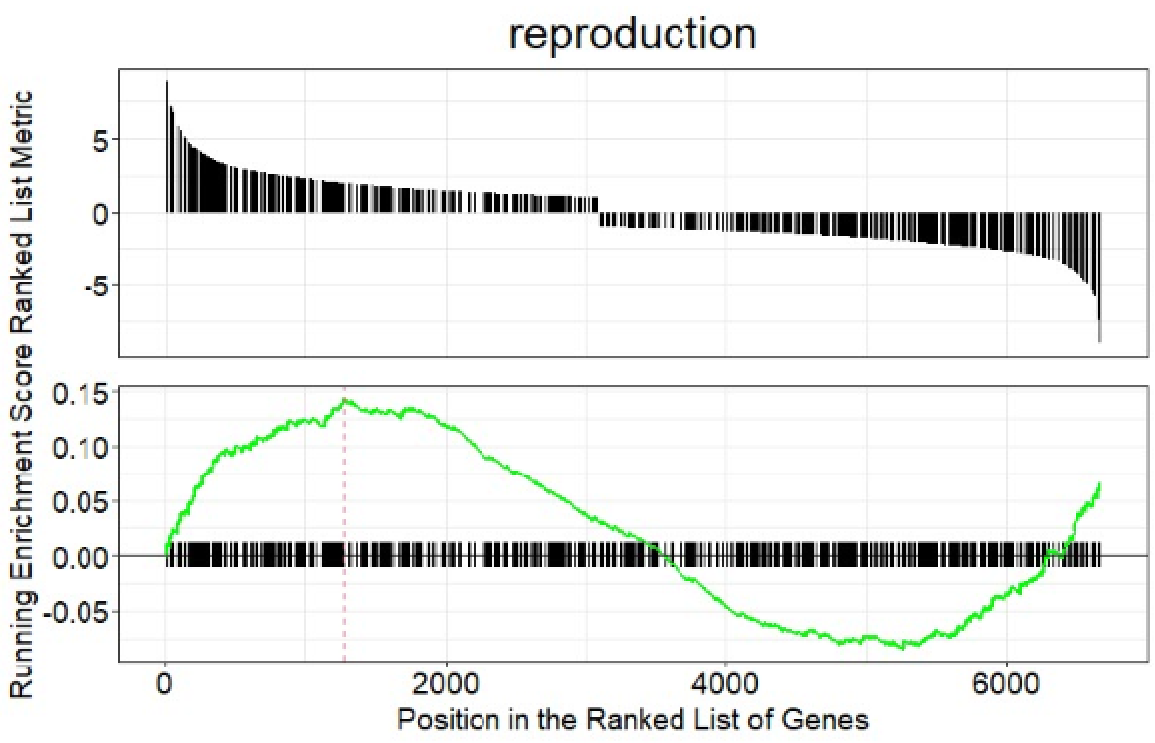


Figure 21. activated gene sets for cluster 1

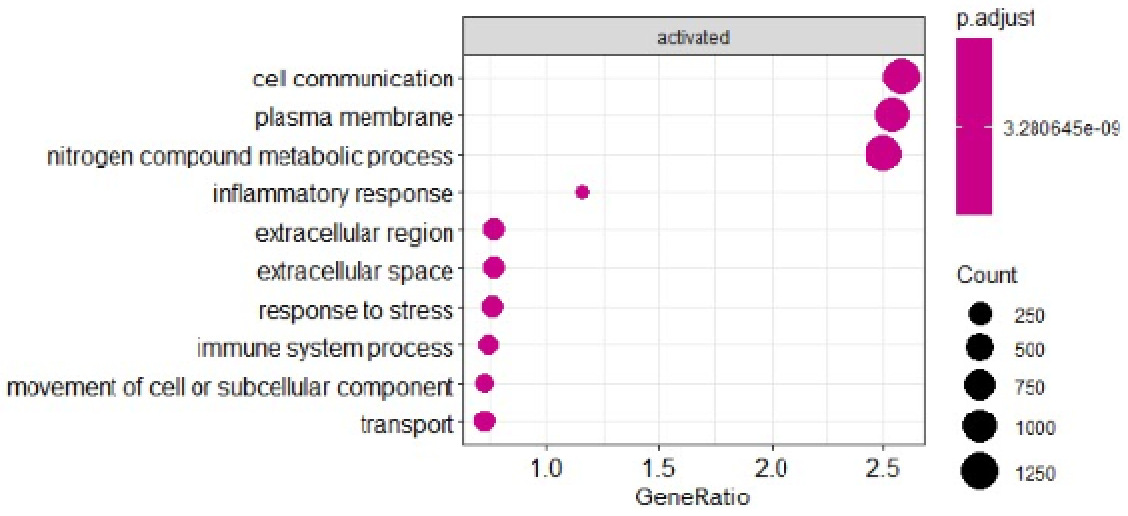


Figure 22. pathway enrichment distribution for cluster 1

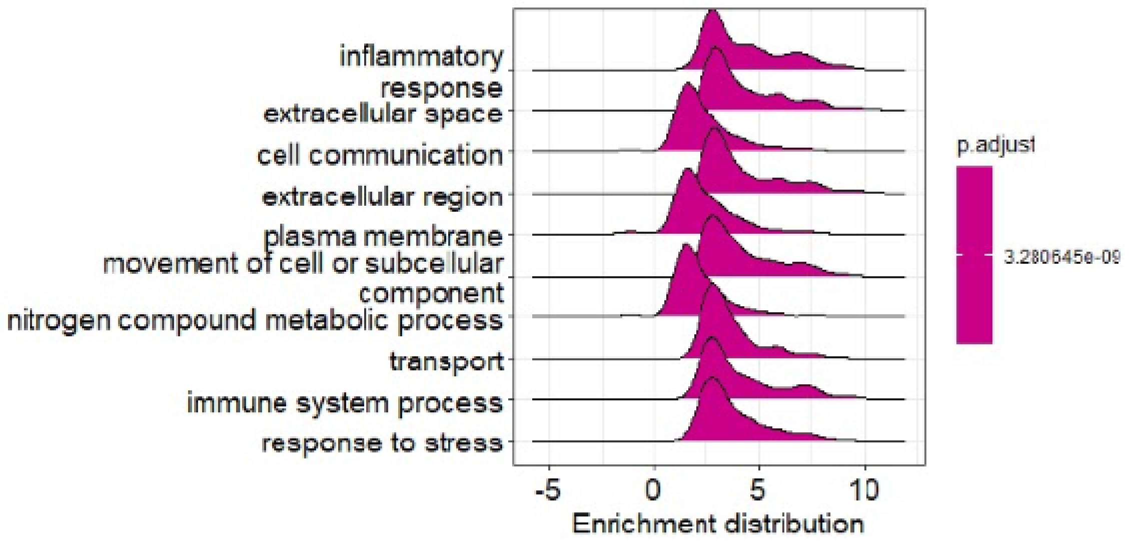


Figure 23. GSEA plot for cluster 1

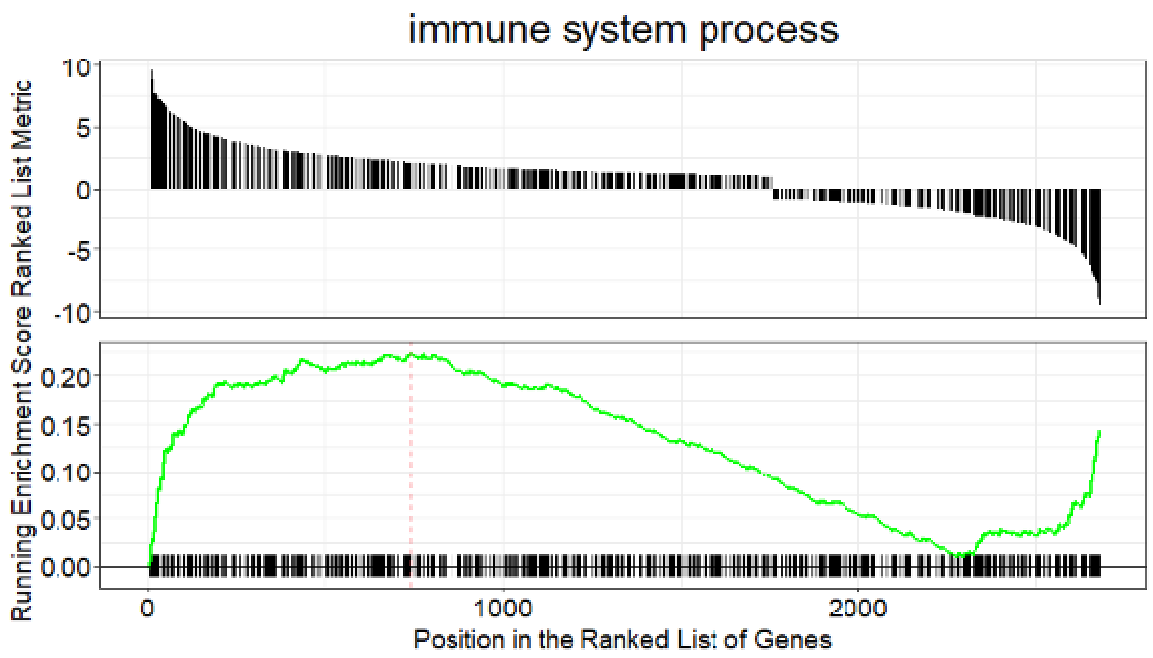


Figure 24. activated gene sets for cluster 2

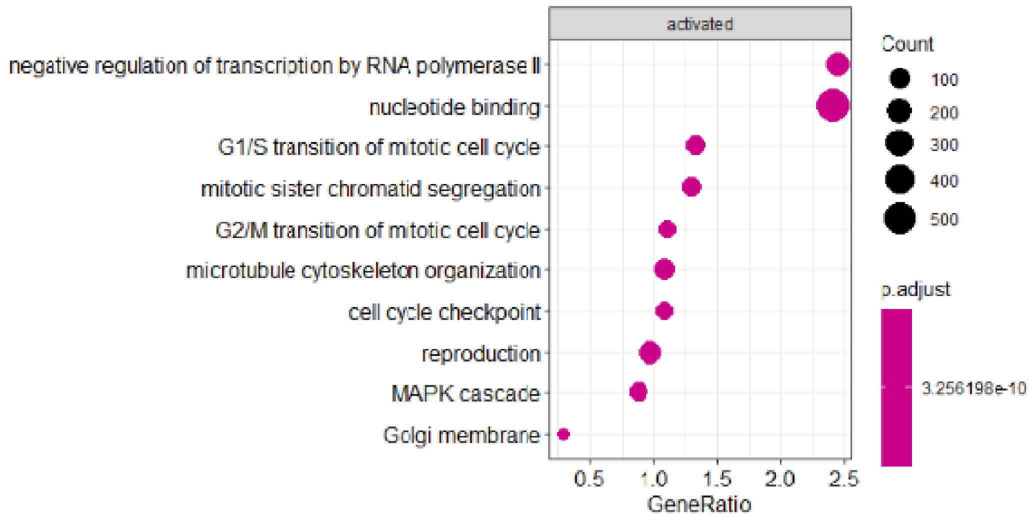


Figure 25. pathway enrichment distribution for cluster 2

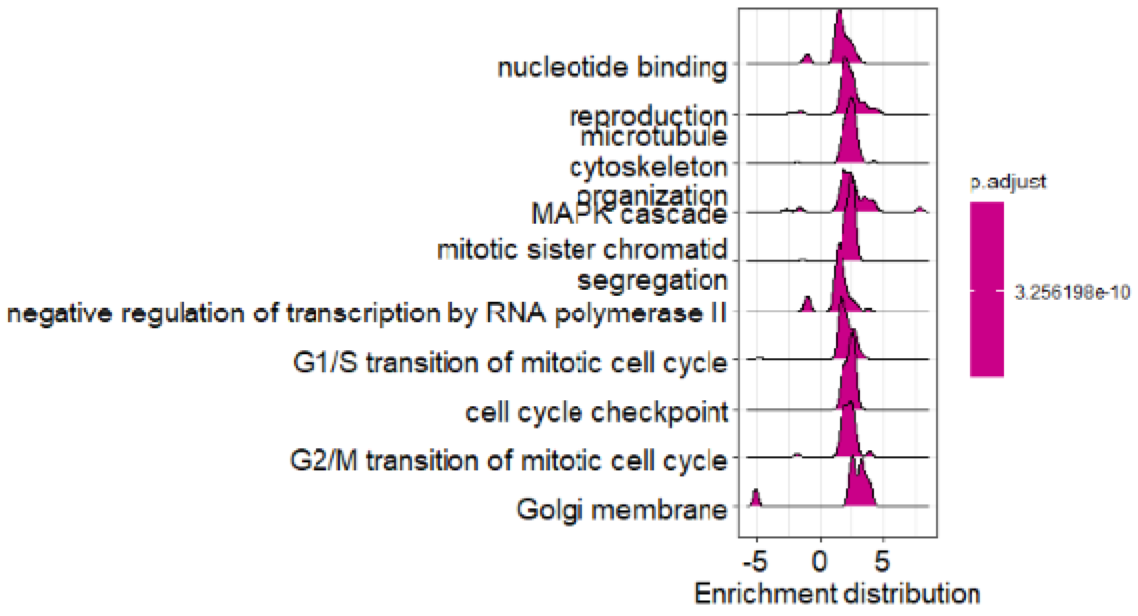
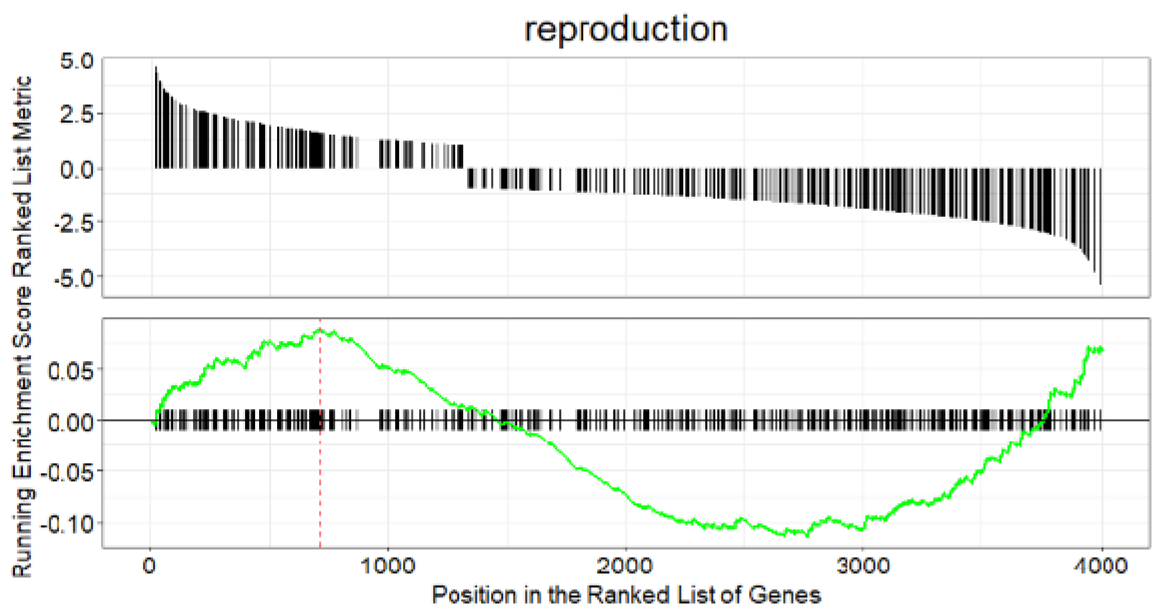


Figure 26. GSEA plot for cluster 2



Discussion

From this analysis I learned that 1) high-throughput RNAseq data needs proper quality control and normalization before proceeding to analyses; 2) for DEGs list, we were able to build correlation hierarchies and draw heatmap to observe the associations among samples and/or genes; 3) using supervised k-mean and unsupervised PCA method, we clustered the DEGs into two clusters that behave differently in un-senescent and senescent cells; 4) GO-term analysis suggested immune system processes related and reproduction cell cycle-related enrichment elements for cluster 1 and cluster 2.

Compared to the paper that published this dataset [3] I got different results because this analysis did not find XPO7 knock-out was very influential in terms of RNA expression whereas the authors were mainly focused on GO-term enrichment for XPO7 knock-out and did not present any results on actual expression analysis.

There are 2774 differentially expressed genes in this dataset. It is no surprise that reproduction cell cycle-related BPs are enriched for one of the clusters because senescence is a cell cycle-related disorder. The immune system processes related to cluster 1 is of most interest. It is known that senescence is involved not only in aging, but also in cancer, organoid development, and injury [4-5]. Does it mean that inflammatory elements play broader roles than what we believe, and could they be used as biomarkers for diagnosis and progress monitoring? In a hope that genes in the top enriched pathways could be further verified in wet-lab experiments, I created top 10 lists for DEGs, cluster 1, and cluster 2 at the end of the coding. It is interesting to see that most of the genes are not the typical markers used for cell cycle identification and inflammation diagnosis.

References

1. Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res. 2002 Jan 1;30(1):207-10. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE153921
2. Innes A.J., Gil J. (2019) IMR90 ER:RAS: A Cell Model of Oncogene-Induced Senescence. In: Demaria M. (eds) Cellular Senescence. Methods in Molecular Biology, vol 1896. Humana Press, New York, NY. https://doi.org/10.1007/978-1-4939-8931-7_9 https://pubmed.ncbi.nlm.nih.gov/30474842/
3. Innes AJ, Sun B, Wagner V, Brookes S, McHugh D, Pombo J, Porreca RM, Dharmalingam G, Vernia S, Zuber J, Vannier JB, García-Escudero R, Gil J. XPO7 is a tumor suppressor regulating p21CIP1-dependent senescence. Genes Dev. 2021 Mar 1;35(5-6):379-391. doi: 10.1101/gad.343269.120. Epub 2021 Feb 18. PMID: 33602872; PMCID: PMC7919420. https://pubmed.ncbi.nlm.nih.gov/33602872/
4. Calcinotto A, Kohli J, Zagato E, Pellegrini L, Demaria M, Alimonti A. Cellular Senescence: Aging, Cancer, and Injury. Physiol Rev. 2019 Apr 1;99(2):1047-1078. doi: 10.1152/physrev.00020.2018. PMID: 30648461. https://pubmed.ncbi.nlm.nih.gov/30648461/
5. Campisi, J., d'Adda di Fagagna, F. Cellular senescence: when bad things happen to good cells. Nat Rev Mol Cell Biol 8, 729–740 (2007). https://doi.org/10.1038/nrm2233 https://www.nature.com/articles/nrm2233

FinalProject.R

wen_x

2021-08-22

```
library(GEOquery)

## Loading required package: Biobase

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

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

## The following objects are masked from 'package:base':
##
##     anyDuplicated, 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

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

## Setting options('download.file.method.GEOquery'='auto')

## Setting options('GEOquery.inmemory.gpl'=FALSE)

library(GOstats)

## Loading required package: Category

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

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

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

##
## Attaching package: 'IRanges'

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

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:S4Vectors':
##
##     expand

## Loading required package: graph

##

##
## Attaching package: 'GOstats'

## The following object is masked from 'package:AnnotationDbi':
##
##     makeGOGraph

library(GO.db)
library(Category)

library(AnnotationDbi)
library(annotate)

## Loading required package: XML

##
## Attaching package: 'XML'

## The following object is masked from 'package:graph':
##
##     addNode

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

```
##
```

```
library(DESeq2)
```

```
## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## The following object is masked from 'package:Biobase':
##
##     rowMedians
```

```
library(clusterProfiler)
```

```
## clusterProfiler v4.0.3  For help: https://yulab-smu.top/biomedical-knowledge-mining-bo
ok/
##
## 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. clusterProfiler 4.0: A universal enrichment tool for interpreting omics da
ta. The Innovation. 2021, 2(3):100141. doi: 10.1016/j.xinn.2021.100141
```

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

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

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

## The following object is masked from 'package:S4Vectors':
##
##     rename

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

library(enrichplot)
library(DOSE)

## DOSE v3.18.1  For help: https://guangchuangyu.github.io/software/DOSE
##
## If you use DOSE in published research, please cite:
## Guangchuang Yu, Li-Gen Wang, Guang-Rong Yan, Qing-Yu He. DOSE: an R/Bioconductor packa
ge for Disease Ontology Semantic and Enrichment analysis. Bioinformatics 2015, 31(4):608-
609

library(ggridges)
library(ggupset)

library(dendextend)

##
## ---------------------
## Welcome to dendextend version 1.15.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendexte
nd/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------

##
## Attaching package: 'dendextend'
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library(vsn)
library(pheatmap)
library(ggplot2)
library(EnhancedVolcano)
```

```
## Loading required package: ggrepel
```

```
## Registered S3 methods overwritten by 'ggalt':
##   method                  from
##   grid.draw.absoluteGrob  ggplot2
##   grobHeight.absoluteGrob ggplot2
##   grobWidth.absoluteGrob  ggplot2
##   grobX.absoluteGrob      ggplot2
##   grobY.absoluteGrob      ggplot2
```

```
library(ggbeeswarm)
library(apeglm)

library(PoiClaClu)
library(glmpca)
library(M3C)
library(Rtsne)
library(cluster)
library(ggbiplot)
```

```
## Loading required package: plyr
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:clusterProfiler':
##
##     arrange, mutate, rename, summarise
```

```
## The following object is masked from 'package:matrixStats':
##
##     count
```

```
## The following object is masked from 'package:graph':
##
##     join
```

```
## The following object is masked from 'package:IRanges':
##
##     desc
```

```
## The following object is masked from 'package:S4Vectors':
##
##     rename
```

```
## Loading required package: scales
## Loading required package: grid

library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:Biobase':
##
##     combine

## The following object is masked from 'package:BiocGenerics':
##
##     combine

library(devtools)

## Loading required package: usethis

library(rgl)

library(ggpubr)

##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:plyr':
##
##     mutate

## The following object is masked from 'package:dendextend':
##
##     rotate

## The following object is masked from 'package:enrichplot':
##
##     color_palette

# =====================================================
# Data prepare
# =====================================================

# load the count data
fileURL <- paste("https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE153921&format=file&fi
le=GSE153921%5FAndrew%5FXPO7%5Fmerged%5Fgene%5Fcounts%2Ecsv%2Egz")

download.file(fileURL,"GSE153921_Andrew_XPO7_merged_gene_counts.csv.gz")

gene_counts <- read.csv("GSE153921_Andrew_XPO7_merged_gene_counts.csv.gz", row.names=1)

dim(gene_counts)

## [1] 62161    12
```

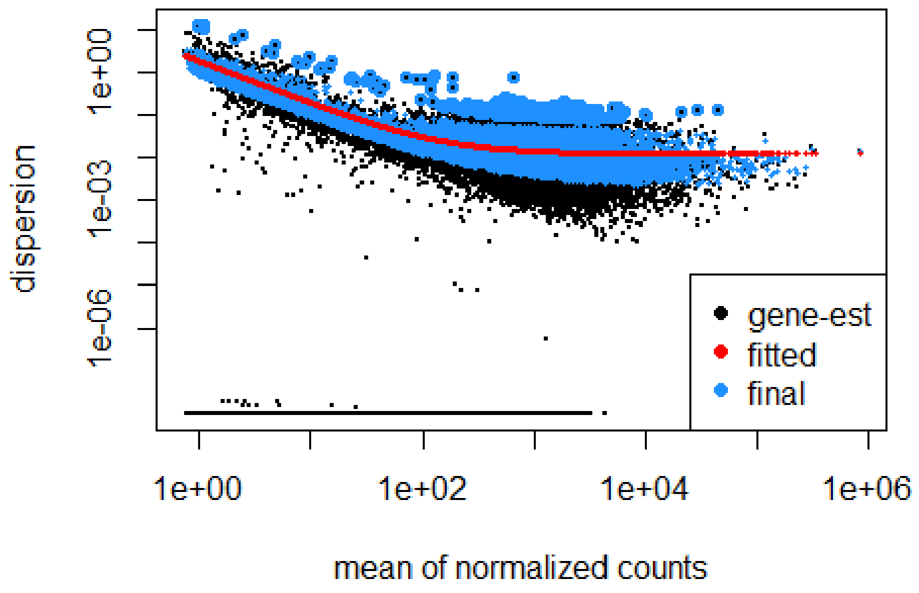
```
# Part 1
# filtering out low counts
keep <- apply(gene_counts, 1, sum) > 10
gene_counts <- gene_counts[keep,]

dim(gene_counts)

## [1] 21743    12

# Obtain corresponding gene symbol from ENSEMBL IDs (rownames)
keytypes(org.Hs.eg.db)

##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
##  [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
## [11] "GENETYPE"     "GO"           "GOALL"        "IPI"          "MAP"
## [16] "OMIM"         "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"
## [21] "PMID"         "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
## [26] "UNIPROT"

# create a data frame for gene annotation, with ensembl id as row name
gene_anno <- data.frame(symbol = mapIds(org.Hs.eg.db, rownames(gene_counts),
                                        keytype = "ENSEMBL",
                                       column = "SYMBOL"
)
)

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

rownames(gene_anno) <- rownames(gene_counts)

# add genename column
gene_anno$geneName <- mapIds(org.Hs.eg.db, rownames(gene_counts),
                              keytype = "ENSEMBL",
                              column = "GENENAME")

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

# add entrez id column
gene_anno$ENTREZID <- mapIds(org.Hs.eg.db, rownames(gene_counts),
                             keytype = "ENSEMBL",
                             column = "ENTREZID")

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

head(gene_anno)

##                       symbol                               geneName  ENTREZID
## ENSG00000227232       WASH7P    WASP family homolog 7, pseudogene    653635
## ENSG00000238009 LOC100996442         uncharacterized LOC100996442 100996442
## ENSG00000237683         <NA>                                 <NA>      <NA>
## ENSG00000241860         <NA>                                 <NA>      <NA>
## ENSG00000228463    RPL23AP21 ribosomal protein L23a pseudogene 21    728481
## ENSG00000237094         <NA>                                 <NA>      <NA>
```

```
# update gene_counts and gene_anno with non-NA symbol
keep_anno <- which(gene_anno$symbol != "NA")

gene_counts <- gene_counts[keep_anno,]
head(gene_counts)

##                 D5minusT D5minusA D5minusC D5plusT D5plusA D5plusB D5shX3T
## ENSG00000227232        3        4        6       3       4      13       4
## ENSG00000238009        5        6        6       3       5      11       2
## ENSG00000228463       21       43       41      38      25      45      31
## ENSG00000230021        3        1        1       7       3       6       2
## ENSG00000225972        4        8        7      10       2      18       6
## ENSG00000225630     1168     1561     1395    2073    1439    2718    2045
##                 D5shX3A D5shX3B D5shX5T D5shX5A D5shX5B
## ENSG00000227232       3      11       4       7       8
## ENSG00000238009       4       3       3       2       3
## ENSG00000228463      25      53      27      39      16
## ENSG00000230021       3       5       3       5       0
## ENSG00000225972       2      14       2       7       7
## ENSG00000225630    1854    3165    1742    1729    1814

gene_anno <- gene_anno[keep_anno,]
head(gene_anno)

##                       symbol                            geneName  ENTREZID
## ENSG00000227232       WASH7P    WASP family homolog 7, pseudogene    653635
## ENSG00000238009 LOC100996442         uncharacterized LOC100996442 100996442
## ENSG00000228463    RPL23AP21 ribosomal protein L23a pseudogene 21    728481
## ENSG00000230021 LOC101928626         uncharacterized LOC101928626 101928626
## ENSG00000225972     MTND1P23                 MT-ND1 pseudogene 23 100887749
## ENSG00000225630     MTND2P28                 MT-ND2 pseudogene 28 100652939

# =====================================================
# study design
# =====================================================
#
gse <- getGEO("GSE153921")

## Found 1 file(s)

## GSE153921_series_matrix.txt.gz

## Rows: 0 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (13): ID_REF, GSM4658433, GSM4658434, GSM4658435, GSM4658436, GSM4658437...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

## File stored at:

## C:\Users\wen_x\AppData\Local\Temp\RtmpewPX03/GPL16791.soft
```

```
gsedta <- gse$GSE153921_series_matrix.txt.gz


coldata <- data.frame(Expgroup = gsedta$source_name_ch1)
rownames(coldata) <- names(gene_counts)
coldata$GEO <- gsedta$geo_accession
coldata$se <- relevel(factor(rep(c("non_induced", "induced"),
                                 c(3, 9))),
                      "non_induced")
coldata$ko <- relevel(factor(rep(c("control", "no_ko", "ko_shXPO7_3","ko_shXPO7_5"),
                                 c(3,3,3,3))),
                      "control")



#creating the DESeq2 object from the matrix of counts
dds <- DESeqDataSetFromMatrix(countData = gene_counts,
                              colData = coldata,
                              design = ~ ko)

colData(dds)

## DataFrame with 12 rows and 4 columns
##                         Expgroup         GEO          se          ko
##                      <character> <character>    <factor>    <factor>
## D5minusT ER-RAS Non-induced D..  GSM4658433 non_induced     control
## D5minusA ER-RAS Non-induced D..  GSM4658434 non_induced     control
## D5minusC ER-RAS Non-induced D..  GSM4658435 non_induced     control
## D5plusT   ER-RAS Induced D5plus  GSM4658436 induced         no_ko
## D5plusA   ER-RAS Induced D5plus  GSM4658437 induced         no_ko
## ...                          ...         ...         ...         ...
## D5shX3A  ER-RAS Induced + shX..  GSM4658440     induced ko_shXPO7_3
## D5shX3B  ER-RAS Induced + shX..  GSM4658441     induced ko_shXPO7_3
## D5shX5T  ER-RAS Induced + shX..  GSM4658442     induced ko_shXPO7_5
## D5shX5A  ER-RAS Induced + shX..  GSM4658443     induced ko_shXPO7_5
## D5shX5B  ER-RAS Induced + shX..  GSM4658444     induced ko_shXPO7_5

# =====================================================
# Data prepare
# =====================================================
# make a copy for dds:

# cnt for normalization; dds for clustering
cnts <- dds
```

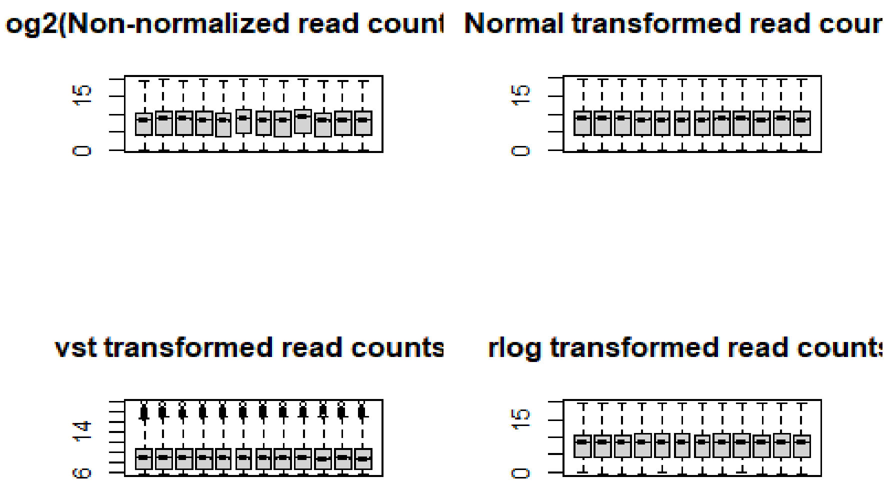
```
# Part 2
# ==============================================================================

# Normalization of dataset
# before normalization, perform size factor estimation
cnts <- estimateSizeFactors(cnts)
cnts <- estimateDispersions(cnts)
```

```
## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

plotDispEsts(cnts)
```

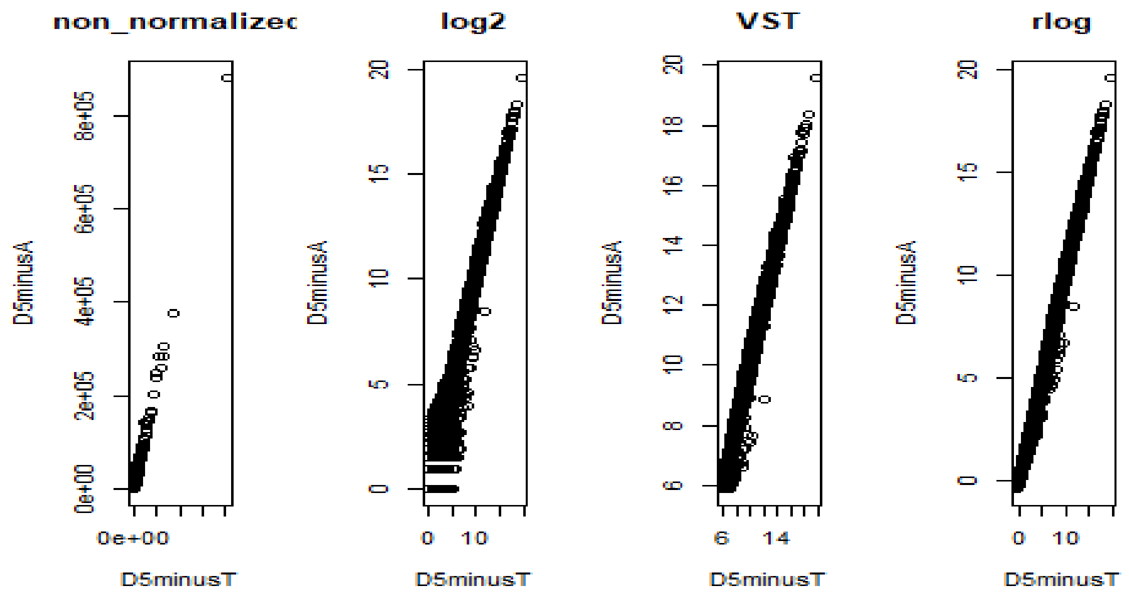


```
# three normalization methods: vst, rlog, and log2
# ntd
ntd_cnts <- normTransform(cnts)
# head(assay(ntd_cnts))

# vst
vst_cnts <- vst(cnts, blind = FALSE)
# head(assay(vst_cnts))

# rlog
rld_cnts <- rlog(cnts, blind = FALSE)
# head(assay(rld_cnts))


# boxplot shows difference between normalized and non-normalized read counts
par(mfrow=c(2,2))
boxplot(log2(counts(cnts)+1), notch=TRUE,
        main="log2(Non-normalized read counts+1)",
        cex=.6, xaxt="n")
boxplot(assay(ntd_cnts), notch=TRUE,
        main="Normal transformed read counts",
        cex=.6, xaxt="n")
```

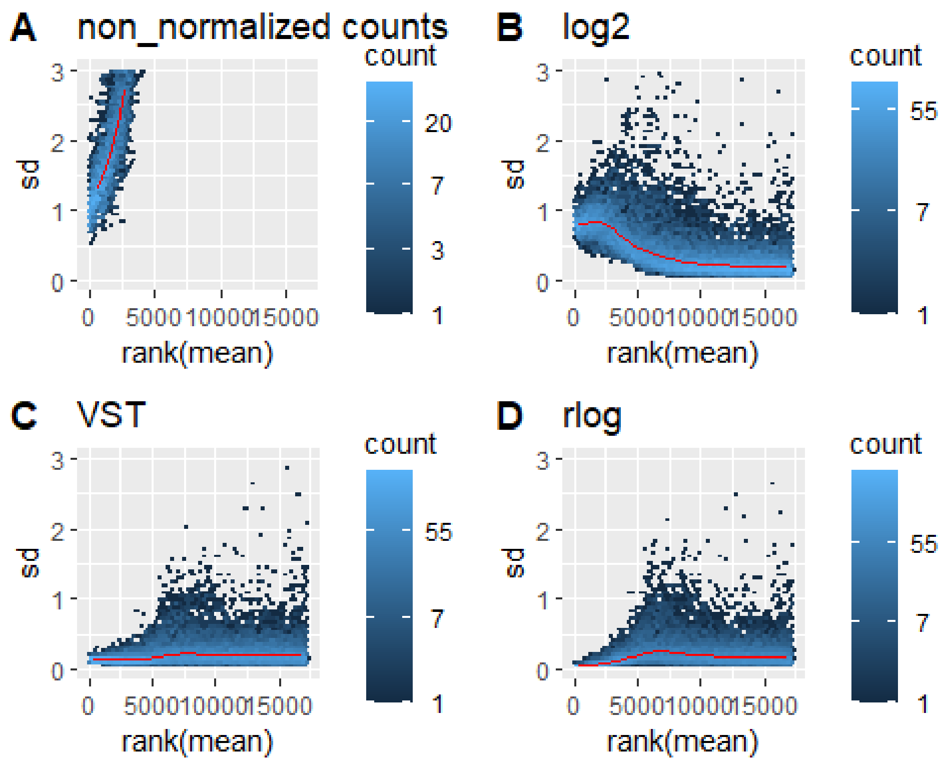
```
boxplot(assay(vst_cnts), notch=TRUE,
        main="vst transformed read counts",
        cex=.6, xaxt="n")
boxplot(assay(rld_cnts), notch=TRUE,
        main="rlog transformed read counts",
        cex=.6, xaxt="n")
```



```
# plot the expression measurements for all 3 methods
par(mfrow=c(1,4))
plot(counts(cnts), main="non_normalized")
plot(assay(ntd_cnts), main="log2")
plot(assay(vst_cnts), main="VST")
plot(assay(rld_cnts), main="rlog")
```



```
# plot meanSD
cntp <- meanSdPlot(counts(cnts))$gg + ggtitle("non_normalized counts")+ scale_y_continuou
s(limits = c(0, 3))

ntdp <- meanSdPlot(assay(ntd_cnts))$gg + ggtitle("log2")+ scale_y_continuous(limits = c
(0, 3))

vstp <- meanSdPlot(assay(vst_cnts))$gg + ggtitle("VST")+ scale_y_continuous(limits = c(0,
 3))

rldp <- meanSdPlot(assay(rld_cnts))$gg + ggtitle("rlog")+ scale_y_continuous(limits = c
(0, 3))

ggarrange(cntp, ntdp, vstp, rldp,
          labels = c("A", "B", "C","D"),
          ncol = 2, nrow = 2)

## Warning: Removed 14479 rows containing non-finite values (stat_binhex).

## Warning: Removed 5 rows containing missing values (geom_hex).

## Warning: Removed 33 row(s) containing missing values (geom_path).

## Warning: Removed 10 rows containing non-finite values (stat_binhex).

## Warning: Removed 1 rows containing missing values (geom_hex).

## Warning: Removed 1 rows containing missing values (geom_hex).

## Warning: Removed 5 rows containing missing values (geom_hex).
```



```
# Part 3
# =========================================================================
# differential expressed analysis (one factor within ko, compared to control)
dds <- DESeq(dds)

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

colData(dds)

## DataFrame with 12 rows and 5 columns
##                          Expgroup         GEO          se          ko sizeFactor
##                       <character> <character>    <factor>    <factor>  <numeric>
## D5minusT ER-RAS Non-induced D..  GSM4658433 non_induced     control   0.882447
## D5minusA ER-RAS Non-induced D..  GSM4658434 non_induced     control   1.133855
## D5minusC ER-RAS Non-induced D..  GSM4658435 non_induced     control   0.966584
## D5plusT    ER-RAS Induced D5plus  GSM4658436 induced         no_ko     1.003402
## D5plusA    ER-RAS Induced D5plus  GSM4658437 induced         no_ko     0.773841
## ...                           ...         ...         ...         ...        ...
## D5shX3A   ER-RAS Induced + shX..  GSM4658440     induced ko_shXPO7_3   0.898675
## D5shX3B   ER-RAS Induced + shX..  GSM4658441     induced ko_shXPO7_3   1.610331
## D5shX5T   ER-RAS Induced + shX..  GSM4658442     induced ko_shXPO7_5   0.800913
## D5shX5A   ER-RAS Induced + shX..  GSM4658443     induced ko_shXPO7_5   0.905299
## D5shX5B   ER-RAS Induced + shX..  GSM4658444     induced ko_shXPO7_5   0.995622

resultsNames(dds)

## [1] "Intercept"                 "ko_ko_shXPO7_3_vs_control"
## [3] "ko_ko_shXPO7_5_vs_control" "ko_no_ko_vs_control"

# To count-in the influence of senescence induce, we perform a group-interaction analysis

# modify design
paste0(dds$se, dds$ko)

##  [1] "non_inducedcontrol" "non_inducedcontrol" "non_inducedcontrol"
##  [4] "inducedno_ko"       "inducedno_ko"       "inducedno_ko"
##  [7] "inducedko_shXPO7_3" "inducedko_shXPO7_3" "inducedko_shXPO7_3"
## [10] "inducedko_shXPO7_5" "inducedko_shXPO7_5" "inducedko_shXPO7_5"

dds$group <- factor(paste0(dds$se, dds$ko))
design(dds) <- ~group

# differential expression analysis
dds <- DESeq(dds)
```

```
## using pre-existing size factors

## estimating dispersions

## found already estimated dispersions, replacing these

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

colData(dds)

## DataFrame with 12 rows and 6 columns
##                         Expgroup         GEO          se          ko sizeFactor
##                      <character> <character>    <factor>    <factor>  <numeric>
## D5minusT ER-RAS Non-induced D..   GSM4658433 non_induced     control   0.882447
## D5minusA ER-RAS Non-induced D..   GSM4658434 non_induced     control   1.133855
## D5minusC ER-RAS Non-induced D..   GSM4658435 non_induced     control   0.966584
## D5plusT   ER-RAS Induced D5plus   GSM4658436 induced         no_ko     1.003402
## D5plusA   ER-RAS Induced D5plus   GSM4658437 induced         no_ko     0.773841
## ...                          ...         ...         ...         ...        ...
## D5shX3A  ER-RAS Induced + shX..   GSM4658440     induced ko_shXPO7_3   0.898675
## D5shX3B  ER-RAS Induced + shX..   GSM4658441     induced ko_shXPO7_3   1.610331
## D5shX5T  ER-RAS Induced + shX..   GSM4658442     induced ko_shXPO7_5   0.800913
## D5shX5A  ER-RAS Induced + shX..   GSM4658443     induced ko_shXPO7_5   0.905299
## D5shX5B  ER-RAS Induced + shX..   GSM4658444     induced ko_shXPO7_5   0.995622
##                        group
##                     <factor>
## D5minusT non_inducedcontrol
## D5minusA non_inducedcontrol
## D5minusC non_inducedcontrol
## D5plusT  inducedno_ko
## D5plusA  inducedno_ko
## ...                      ...
## D5shX3A  inducedko_shXPO7_3
## D5shX3B  inducedko_shXPO7_3
## D5shX5T  inducedko_shXPO7_5
## D5shX5A  inducedko_shXPO7_5
## D5shX5B  inducedko_shXPO7_5

resultsNames(dds)

## [1] "Intercept"
## [2] "group_inducedko_shXPO7_5_vs_inducedko_shXPO7_3"
## [3] "group_inducedno_ko_vs_inducedko_shXPO7_3"
## [4] "group_non_inducedcontrol_vs_inducedko_shXPO7_3"

# Results
res_ko5ko3 <- results(dds,
                      name = "group_inducedko_shXPO7_5_vs_inducedko_shXPO7_3")
res_nkoko3 <- results(dds,
                     name = "group_inducedno_ko_vs_inducedko_shXPO7_3")
res_ctrko3 <- results(dds,
                     name = "group_non_inducedcontrol_vs_inducedko_shXPO7_3")

res_ctrnko <- results(dds,
                     contrast = c("group", "inducedno_ko", "non_inducedcontrol"))
res_ctrko5 <- results(dds,
                     contrast = c("group", "inducedko_shXPO7_5", "non_inducedcontrol"))
res_nkoko5 <- results(dds,
                     contrast = c("group", "inducedko_shXPO7_5", "inducedno_ko"))

summary(res_ko5ko3, na.rm=T); head(res_ko5ko3)

##
## out of 17227 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 21, 0.12%
## LFC < 0 (down)     : 13, 0.075%
## outliers [1]       : 3, 0.017%
## low counts [2]     : 2672, 16%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

## log2 fold change (MLE): group inducedko shXPO7 5 vs inducedko shXPO7 3
## Wald test p-value: group inducedko shXPO7 5 vs inducedko shXPO7 3
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE       stat    pvalue
##                 <numeric>      <numeric> <numeric>  <numeric> <numeric>
## ENSG00000227232    5.48765     0.50095761 0.7070282  0.7085398  0.478610
## ENSG00000238009    4.33544     0.16078216 0.8848280  0.1817101  0.855810
## ENSG00000228463   32.70594    -0.00681019 0.3499621 -0.0194598  0.984474
## ENSG00000230021    3.19388     0.09328207 0.9875806  0.0944551  0.924748
## ENSG00000225972    6.57474    -0.01842959 0.7340190 -0.0251078  0.979969
## ENSG00000225630 1839.91302    -0.04714463 0.0935097 -0.5041685  0.614143
##                     padj
##                <numeric>
## ENSG00000227232  0.702373
## ENSG00000238009        NA
## ENSG00000228463  0.993004
## ENSG00000230021        NA
## ENSG00000225972  0.990864
## ENSG00000225630  0.794895

summary(res_nkoko3, na.rm=T); head(res_nkoko3)

##
## out of 17227 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 28, 0.16%
## LFC < 0 (down)     : 86, 0.5%
## outliers [1]       : 3, 0.017%
## low counts [2]     : 3340, 19%
## (mean count < 7)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

## log2 fold change (MLE): group inducedno ko vs inducedko shXPO7 3
## Wald test p-value: group inducedno ko vs inducedko shXPO7 3
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat    pvalue
##                 <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000227232   5.48765      0.3095703 0.7076774  0.437446  0.661788
## ENSG00000238009   4.33544      1.1567787 0.7989148  1.447938  0.147635
## ENSG00000228463  32.70594      0.1586911 0.3423872  0.463484  0.643017
## ENSG00000230021   3.19388      0.8476516 0.9243259  0.917048  0.359117
## ENSG00000225972   6.57474      0.5963785 0.6950862  0.857992  0.390897
## ENSG00000225630 1839.91302     -0.0406381 0.0932857 -0.435630  0.663105
##                      padj
##                 <numeric>
## ENSG00000227232        NA
## ENSG00000238009        NA
## ENSG00000228463  0.999995
## ENSG00000230021        NA
## ENSG00000225972        NA
## ENSG00000225630  0.999995

summary(res_ctrko3, na.rm=T); head(res_ctrko3)

##
## out of 17227 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 3986, 23%
## LFC < 0 (down)     : 4074, 24%
## outliers [1]       : 3, 0.017%
## low counts [2]     : 334, 1.9%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

## log2 fold change (MLE): group non inducedcontrol vs inducedko shXPO7 3
## Wald test p-value: group non inducedcontrol vs inducedko shXPO7 3
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat      pvalue
##                 <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000227232   5.48765     -0.1732871 0.7412234 -0.233785 8.15152e-01
## ENSG00000238009   4.33544      1.1041475 0.8035699  1.374053 1.69425e-01
## ENSG00000228463  32.70594      0.1764470 0.3426736  0.514913 6.06614e-01
## ENSG00000230021   3.19388     -0.7285644 1.0625502 -0.685675 4.92918e-01
## ENSG00000225972   6.57474      0.0961393 0.7192726  0.133662 8.93670e-01
## ENSG00000225630 1839.91302     -0.5576192 0.0941261 -5.924173 3.13874e-09
##                        padj
##                   <numeric>
## ENSG00000227232 8.84392e-01
## ENSG00000238009 2.81570e-01
## ENSG00000228463 7.25103e-01
## ENSG00000230021 6.27431e-01
## ENSG00000225972 9.35249e-01
## ENSG00000225630 2.30593e-08
```

```
summary(res_ctrnko, na.rm=T); head(res_ctrnko)
```

```
##
## out of 17227 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 4037, 23%
## LFC < 0 (down)     : 4106, 24%
## outliers [1]       : 3, 0.017%
## low counts [2]     : 334, 1.9%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
## log2 fold change (MLE): group inducedno_ko vs non_inducedcontrol
## Wald test p-value: group inducedno_ko vs non_inducedcontrol
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE       stat      pvalue
##                 <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## ENSG00000227232   5.48765      0.4828574 0.7321282  0.6595258 5.09558e-01
## ENSG00000238009   4.33544      0.0526312 0.7285356  0.0722425 9.42409e-01
## ENSG00000228463  32.70594     -0.0177559 0.3423373 -0.0518668 9.58635e-01
## ENSG00000230021   3.19388      1.5762160 1.0226031  1.5413762 1.23225e-01
## ENSG00000225972   6.57474      0.5002391 0.6995258  0.7151118 4.74540e-01
## ENSG00000225630 1839.91302      0.5169811 0.0943446  5.4797128 4.26017e-08
##                        padj
##                   <numeric>
## ENSG00000227232 6.40694e-01
## ENSG00000238009 9.66970e-01
## ENSG00000228463 9.77089e-01
## ENSG00000230021 2.17026e-01
## ENSG00000225972 6.08948e-01
## ENSG00000225630 2.75265e-07
```

```
summary(res_ctrko5, na.rm=T); head(res_ctrko5)
```

```
##
## out of 17227 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 4158, 24%
## LFC < 0 (down)     : 4177, 24%
## outliers [1]       : 3, 0.017%
## low counts [2]     : 334, 1.9%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
## log2 fold change (MLE): group inducedko_shXPO7_5 vs non_inducedcontrol
## Wald test p-value: group inducedko_shXPO7_5 vs non_inducedcontrol
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat      pvalue
##                   <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000227232    5.48765       0.674245  0.731501  0.921728 3.56670e-01
## ENSG00000238009    4.33544      -0.943365  0.821839 -1.147871 2.51022e-01
## ENSG00000228463   32.70594      -0.183257  0.349913 -0.523722 6.00472e-01
## ENSG00000230021    3.19388       0.821846  1.080118  0.760886 4.46725e-01
## ENSG00000225972    6.57474      -0.114569  0.738225 -0.155195 8.76667e-01
## ENSG00000225630 1839.91302       0.510475  0.094566  5.398075 6.73596e-08
##                         padj
##                    <numeric>
## ENSG00000227232 4.95979e-01
## ENSG00000238009 3.79261e-01
## ENSG00000228463 7.21836e-01
## ENSG00000230021 5.86398e-01
## ENSG00000225972 9.26243e-01
## ENSG00000225630 4.31111e-07
```

```
summary(res_nkoko5, na.rm=T); head(res_nkoko5)
```

```
##
## out of 17227 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 556, 3.2%
## LFC < 0 (down)     : 213, 1.2%
## outliers [1]       : 3, 0.017%
## low counts [2]     : 5008, 29%
## (mean count < 32)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

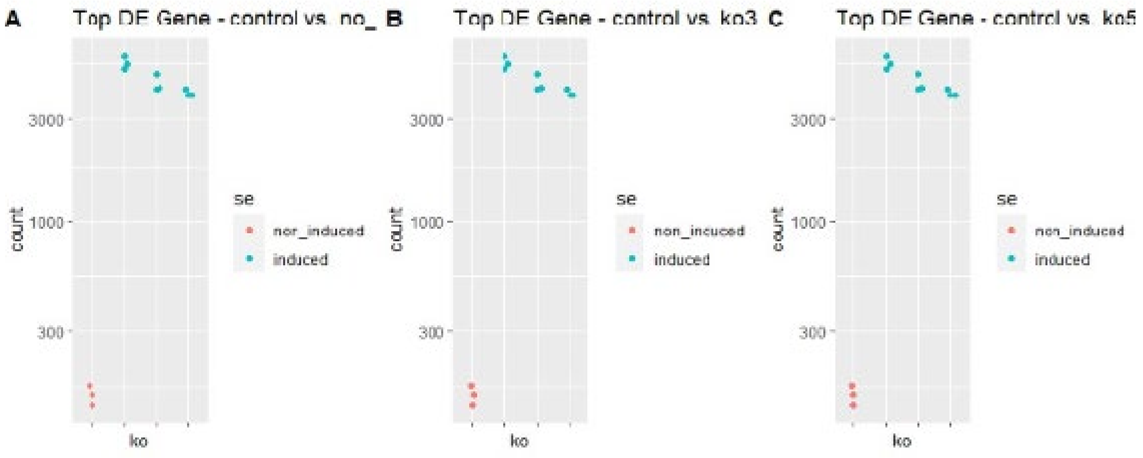
```
## log2 fold change (MLE): group inducedko_shXPO7_5 vs inducedno_ko
## Wald test p-value: group inducedko_shXPO7_5 vs inducedno_ko
## DataFrame with 6 rows and 6 columns
##                    baseMean log2FoldChange     lfcSE      stat    pvalue
##                   <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000227232    5.48765     0.19138727 0.6974871  0.2743954  0.783781
## ENSG00000238009    4.33544    -0.99599656 0.8172885 -1.2186597  0.222973
## ENSG00000228463   32.70594    -0.16550128 0.3496329 -0.4733574  0.635958
## ENSG00000230021    3.19388    -0.75436955 0.9444677 -0.7987246  0.424450
## ENSG00000225972    6.57474    -0.61480805 0.7146798 -0.8602567  0.389648
## ENSG00000225630 1839.91302    -0.00650657 0.0937296 -0.0694185  0.944656
##                       padj
##                  <numeric>
## ENSG00000227232         NA
## ENSG00000238009         NA
## ENSG00000228463   0.747294
## ENSG00000230021         NA
## ENSG00000225972         NA
## ENSG00000225630   0.965800
```

```
# select significant genes based on criteria (p < .01, abs(log2FoldChange) >1)
resSig_ko5ko3_up <- subset(res_ko5ko3, padj<.01 & log2FoldChange > 1)
resSig_ko5ko3_down <- subset(res_ko5ko3, padj<.01 & log2FoldChange < -1)
```

```
resSig_nkoko3_up <- subset(res_nkoko3, padj<.01 & log2FoldChange > 1)
resSig_nkoko3_down <- subset(res_nkoko3, padj<.01 & log2FoldChange < -1)

resSig_ctrko3_up <- subset(res_ctrko3, padj<.01 & log2FoldChange > 1)
resSig_ctrko3_down <- subset(res_ctrko3, padj<.01 & log2FoldChange < -1)

resSig_ctrnko_up <- subset(res_ctrnko, padj<.01 & log2FoldChange > 1)
resSig_ctrnko_down <- subset(res_ctrnko, padj<.01 & log2FoldChange < -1)

resSig_ctrko5_up <- subset(res_ctrko5, padj<.01 & log2FoldChange > 1)
resSig_ctrko5_down <- subset(res_ctrko5, padj<.01 & log2FoldChange < -1)

resSig_nkoko5_up <- subset(res_nkoko5, padj<.01 & log2FoldChange > 1)
resSig_nkoko5_down <- subset(res_nkoko5, padj<.01 & log2FoldChange < -1)


# significant gene IDs
geneSig <- c( rownames(resSig_ko5ko3_up) ,
              rownames(resSig_ko5ko3_down) ,
              rownames(resSig_nkoko3_up) ,
              rownames(resSig_nkoko3_down) ,
              rownames(resSig_ctrko3_up) ,
              rownames(resSig_ctrko3_down) ,
              rownames(resSig_ctrnko_up) ,
              rownames(resSig_ctrnko_down) ,
              rownames(resSig_ctrko5_up) ,
              rownames(resSig_ctrko5_down) ,
              rownames(resSig_nkoko5_up) ,
              rownames(resSig_nkoko5_down) )
length(geneSig)
```

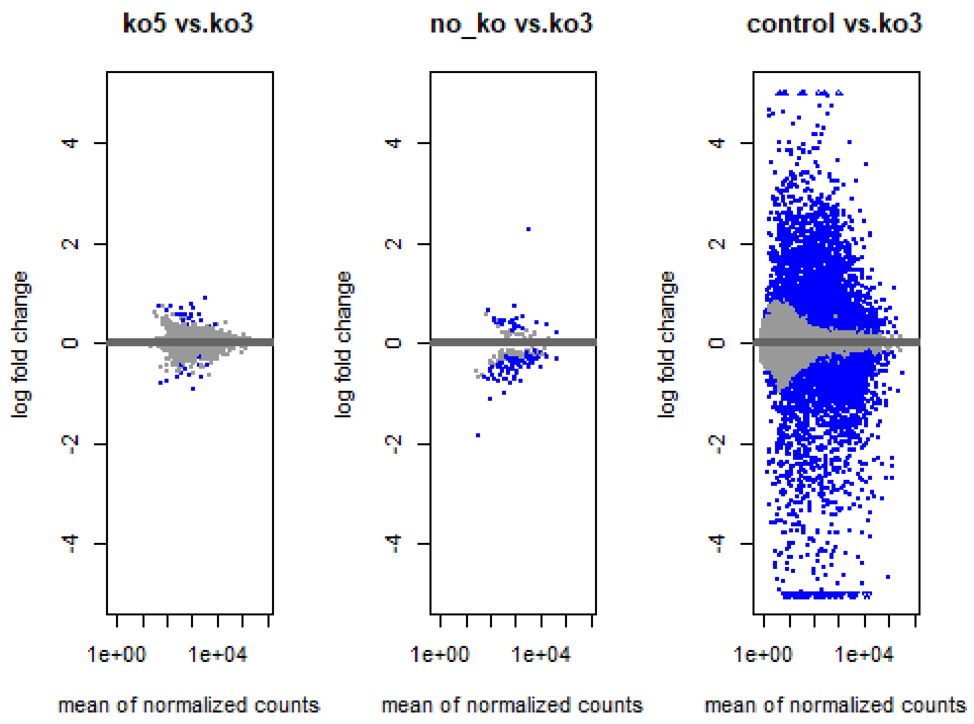
```
## [1] 6684
```

```
# Significant genes statistic data for GO enrichment analysis

log2foldchangeSig <- c(
  resSig_ko5ko3_up$log2FoldChange ,
  resSig_ko5ko3_down$log2FoldChange ,
  resSig_nkoko3_up$log2FoldChange ,
  resSig_nkoko3_down$log2FoldChange ,
  resSig_ctrko3_up$log2FoldChange ,
  resSig_ctrko3_down$log2FoldChange ,
  resSig_ctrnko_up$log2FoldChange ,
  resSig_ctrnko_down$log2FoldChange ,
  resSig_ctrko5_up$log2FoldChange ,
  resSig_ctrko5_down$log2FoldChange ,
  resSig_nkoko5_up$log2FoldChange ,
  resSig_nkoko5_down$log2FoldChange
)


padjSig <- c(
  resSig_ko5ko3_up$padj ,
  resSig_ko5ko3_down$padj ,
  resSig_nkoko3_up$padj ,
  resSig_nkoko3_down$padj ,
  resSig_ctrko3_up$padj ,
  resSig_ctrko3_down$padj ,
  resSig_ctrnko_up$padj ,
  resSig_ctrnko_down$padj ,
  resSig_ctrko5_up$padj ,
  resSig_ctrko5_down$padj ,
  resSig_nkoko5_up$padj ,
  resSig_nkoko5_down$padj
)

geneSig_sta <- as.data.frame(cbind(geneSig,log2foldchangeSig, padjSig))
geneSig_sta$cgrp <- c(
  rep("ko5ko3", length(rownames(resSig_ko5ko3_up))) ,
  rep("ko5ko3", length(rownames(resSig_ko5ko3_down))) ,
  rep("nkoko3", length(rownames(resSig_nkoko3_up))) ,
  rep("nkoko3", length(rownames(resSig_nkoko3_down))) ,
  rep("ctrko3", length(rownames(resSig_ctrko3_up))) ,
  rep("ctrko3", length(rownames(resSig_ctrko3_down))) ,
  rep("ctrnko", length(rownames(resSig_ctrnko_up))) ,
  rep("ctrnko", length(rownames(resSig_ctrnko_down))) ,
  rep("ctrko5", length(rownames(resSig_ctrko5_up))) ,
  rep("ctrko5", length(rownames(resSig_ctrko5_down))) ,
  rep("nkoko5", length(rownames(resSig_nkoko5_up))) ,
  rep("nkoko5", length(rownames(resSig_nkoko5_down)))
)

geneSig_sta$updown <- c(
  rep("up", length(rownames(resSig_ko5ko3_up))) ,
  rep("down", length(rownames(resSig_ko5ko3_down))) ,
  rep("up", length(rownames(resSig_nkoko3_up))) ,
  rep("down", length(rownames(resSig_nkoko3_down))) ,
  rep("up", length(rownames(resSig_ctrko3_up))) ,
  rep("down", length(rownames(resSig_ctrko3_down))) ,
  rep("up", length(rownames(resSig_ctrnko_up))) ,
  rep("down", length(rownames(resSig_ctrnko_down))) ,
  rep("up", length(rownames(resSig_ctrko5_up))) ,
  rep("down", length(rownames(resSig_ctrko5_down))) ,
  rep("up", length(rownames(resSig_nkoko5_up))) ,
  rep("down", length(rownames(resSig_nkoko5_down)))
)

head(geneSig_sta)

##           geneSig log2foldchangeSig                padjSig   cgrp updown
## 1 ENSG00000130227  2.28743724467855 2.17026537082881e-150 nkoko3     up
## 2 ENSG00000163909 -2.04280207199295  0.000369561579468369 nkoko3   down
## 3 ENSG00000117600 -1.05109866144401   4.08279615064362e-06 nkoko3   down
## 4 ENSG00000169129 -1.21971498067099  0.000129016128013861 nkoko3   down
## 5 ENSG00000162576   1.0475918603143  1.20003594902264e-15 ctrko3     up
## 6 ENSG00000179403   1.2062334711963  3.41349878914502e-08 ctrko3     up

# check DE results by plotting counts
#
#ctrnko
top_ctrnko <- rownames(res_ctrnko)[which.min(res_ctrnko$padj)]
d1 <- plotCounts(dds, gene = top_ctrnko,
                 intgroup = c("se", "ko"),
                 returnData = TRUE)

#ctrko3
top_ctrko3 <- rownames(res_ctrko3)[which.min(res_ctrko3$padj)]
d2 <- plotCounts(dds, gene = top_ctrko3,
                 intgroup = c("se", "ko"),
                 returnData = TRUE)

#ctrko5
top_ctrko5 <- rownames(res_ctrko5)[which.min(res_ctrko5$padj)]
d3 <- plotCounts(dds, gene = top_ctrko5,
                 intgroup = c("se", "ko"),
                 returnData = TRUE)


# plotting
ggplot(d1, aes(x=ko, y=count, color=se), x.text.angle = 90 )+
  scale_y_log10()+
  geom_beeswarm(cex=3)+
  labs(title = "Top DE Gene - control vs. no_ko") -> tp1
ggplot(d2, aes(x=ko, y=count, color=se), x.text.angle = 90)+
  scale_y_log10()+
  geom_beeswarm(cex=3)+
  labs(title = "Top DE Gene - control vs. ko3") -> tp2
ggplot(d3, aes(x=ko, y=count, color=se), x.text.angle = 90)+
  scale_y_log10()+
  geom_beeswarm(cex=3)+
  labs(title = "Top DE Gene - control vs. ko5") -> tp3

ggarrange(tp1 + rremove("x.text"),
          tp2 + rremove("x.text"),
          tp3 + rremove("x.text"),
          labels = c("A", "B", "C"),
          ncol = 3, nrow = 1)
```



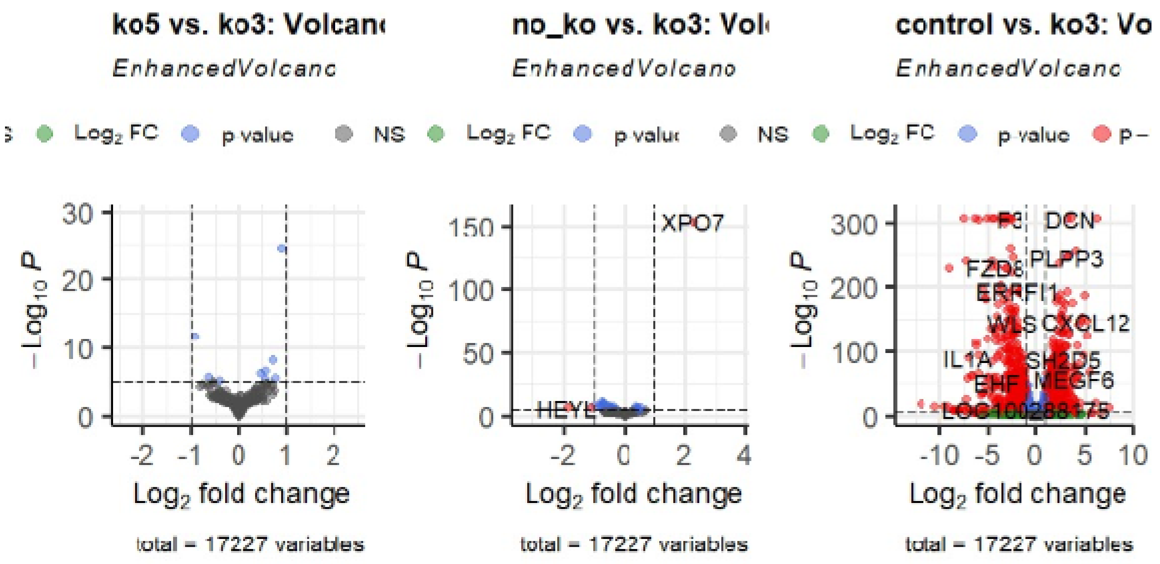
```
# plotMA
resultsNames(dds)

## [1] "Intercept"
## [2] "group_inducedko_shXPO7_5_vs_inducedko_shXPO7_3"
## [3] "group_inducedno_ko_vs_inducedko_shXPO7_3"
## [4] "group_non_inducedcontrol_vs_inducedko_shXPO7_3"

res_ko5ko3_lfcs <- lfcShrink(dds, coef = "group_inducedko_shXPO7_5_vs_inducedko_shXPO7_3
", type = "apeglm")

## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895

res_nkoko3_lfcs <- lfcShrink(dds, coef = "group_inducedno_ko_vs_inducedko_shXPO7_3", type
 = "apeglm")

## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895

res_ctrko3_lfcs <- lfcShrink(dds, coef = "group_non_inducedcontrol_vs_inducedko_shXPO7_3
", type = "apeglm")

## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895

par(mfrow=c(1,3))
plotMA(res_ko5ko3_lfcs, ylim=c(-5, 5))+title("ko5 vs.ko3")

## integer(0)

plotMA(res_nkoko3_lfcs, ylim=c(-5, 5))+title("no_ko vs.ko3")

## integer(0)
```

```
plotMA(res_ctrko3_lfcs, ylim=c(-5, 5))+title("control vs.ko3")
```



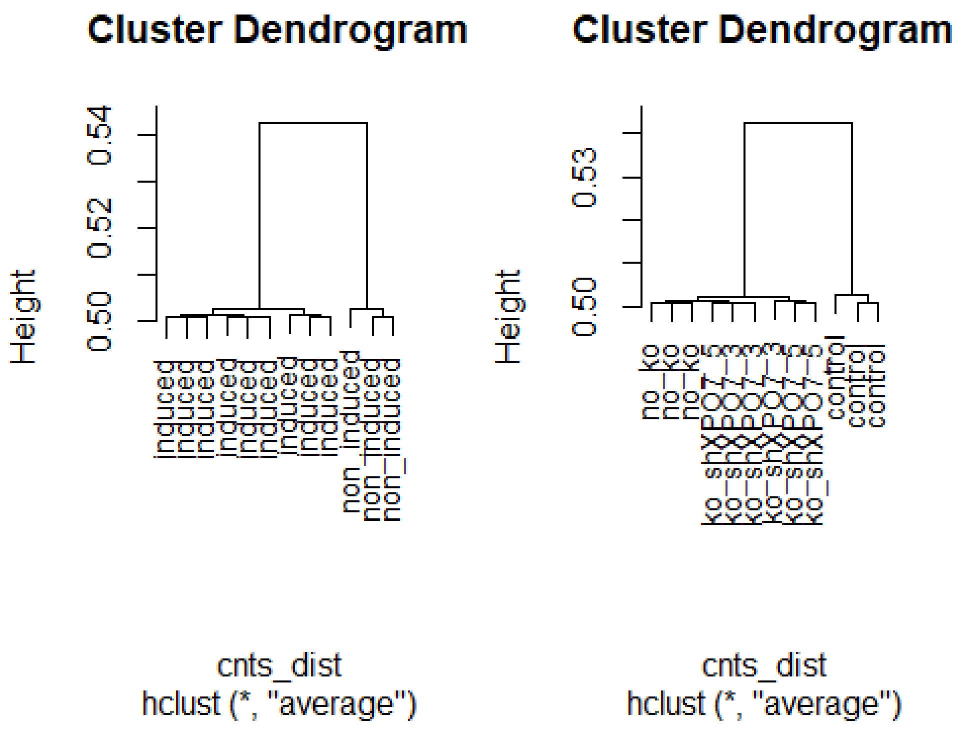
```
## integer(0)

# check DE results by volcano plot

# volcano plot
# obtain gene symbols
 symb_ko5ko3 <- gene_anno[which(rownames(gene_anno)==rownames(res_ko5ko3_lfcs)),1]

symb_nkoko3 <- gene_anno[which(rownames(gene_anno)==rownames(res_nkoko3_lfcs)),1]

symb_ctrko3 <- gene_anno[which(rownames(gene_anno)==rownames(res_ctrko3_lfcs)),1]

# volcano plotting

EnhancedVolcano(res_ko5ko3_lfcs,
                # lab = rownames(res_ko5ko3_lfcs),
                lab = symb_ko5ko3,
                x = 'log2FoldChange',
                y = 'pvalue', title = "ko5 vs. ko3: Volcano Plot") -> vp1

## Warning: Ignoring unknown parameters: xlim, ylim

EnhancedVolcano(res_nkoko3_lfcs,
                # lab = rownames(res_nkoko3_lfcs),
                lab = symb_nkoko3,
                x = 'log2FoldChange',
                y = 'pvalue', title = "no_ko vs. ko3: Volcano Plot") -> vp2
```

```
## Warning: Ignoring unknown parameters: xlim, ylim

EnhancedVolcano(res_ctrko3_lfcs,
                # lab = rownames(res_nkoko3_lfcs),
                lab = symb_ctrko3,
                x = 'log2FoldChange',
                y = 'pvalue', title = "control vs. ko3: Volcano Plot") -> vp3

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

## Warning: Ignoring unknown parameters: xlim, ylim

grid.arrange(vp1, vp2, vp3, nrow = 1)
```



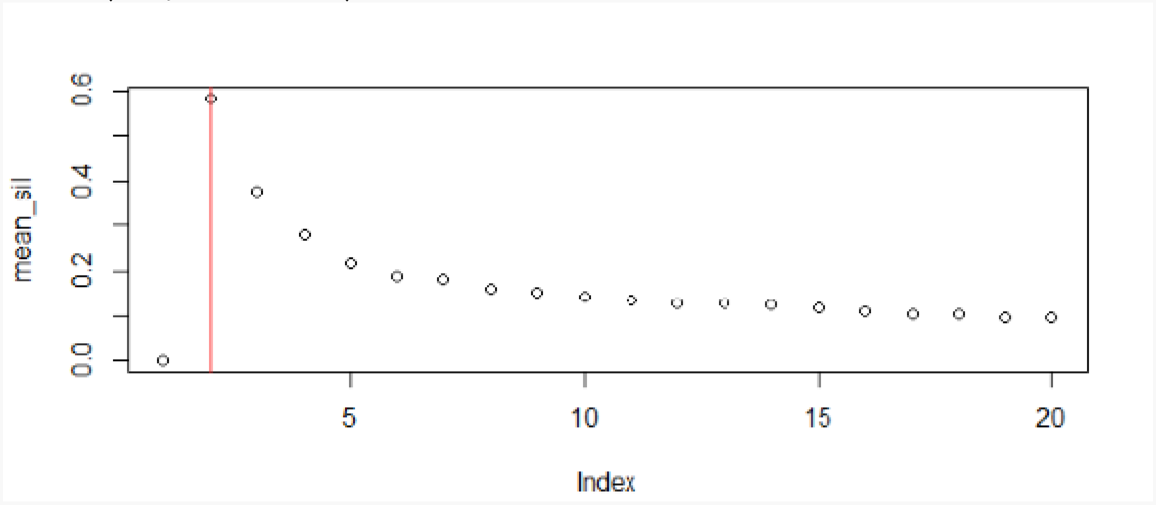
```
# Part 4
# ======================================================================
# clustering of significant genes
# using rlog normalized counts
rld_cnts_Sig <- subset(assay(rld_cnts),
                       rownames(assay(rld_cnts)) %in% geneSig)
dim(rld_cnts_Sig)

## [1] 2774   12

# Calculate distance and cluster

# define distance calculate function
calc_dist <- function(df){
  tempcor <- cor(df, method = "pearson")
  tempdit <- as.dist(1-tempcor/2)
  return(tempdit)
}

# distance for samples
cnts_dist <- calc_dist(rld_cnts_Sig)
```

```
# clustering for samples
cnts_clust <- hclust(cnts_dist, method = "ave")

par(mfrow=c(1,2))
cnts_clust$labels <- coldata$se
plot(cnts_clust, cex = .9)

cnts_clust$labels <- coldata$ko
plot(cnts_clust, cex = .9)
```

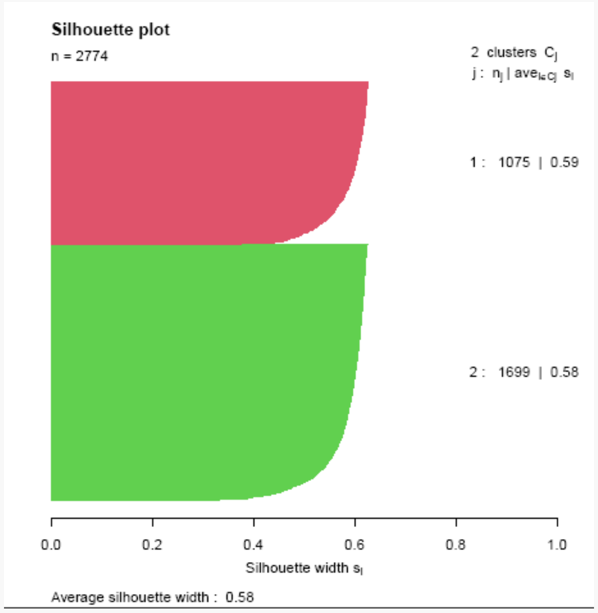


```
# distance for genes
cnts_g_dist <- calc_dist(t(rld_cnts_Sig))

# cluster for genes
cnts_g_clust <- hclust(cnts_g_dist, method = "ave")


# define function to compute the number of clustering-cut using silhouetteWide values
calic_k4clust <- function(in_clust, in_dist){
  mean_sil <- NULL
  mean_sil[1] <- 0
  for (i in 2:20){
    t_clusters <- cutree(in_clust, k=i)
    t_clusters_sil <- silhouette(t_clusters, dist=in_dist)
    mean_sil[i] <- mean(t_clusters_sil[,"sil_width"])
  }
  plot(mean_sil)
}
```

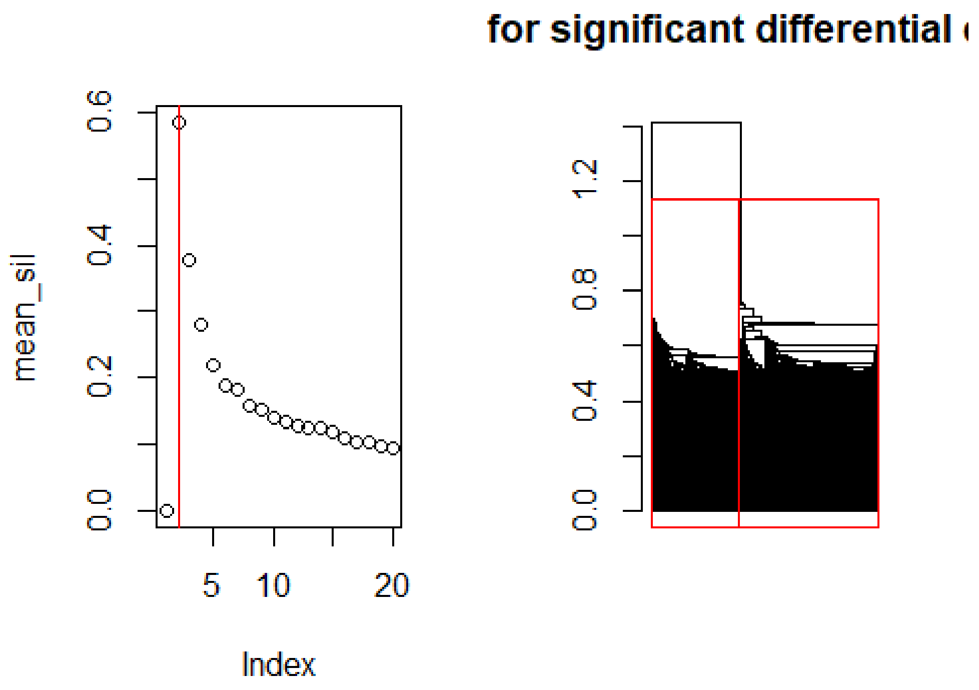
```
# use silhouette width values to guide the number of clustering
calic_k4clust(cnts_g_clust, cnts_g_dist)
abline(v=2, col="red")
```



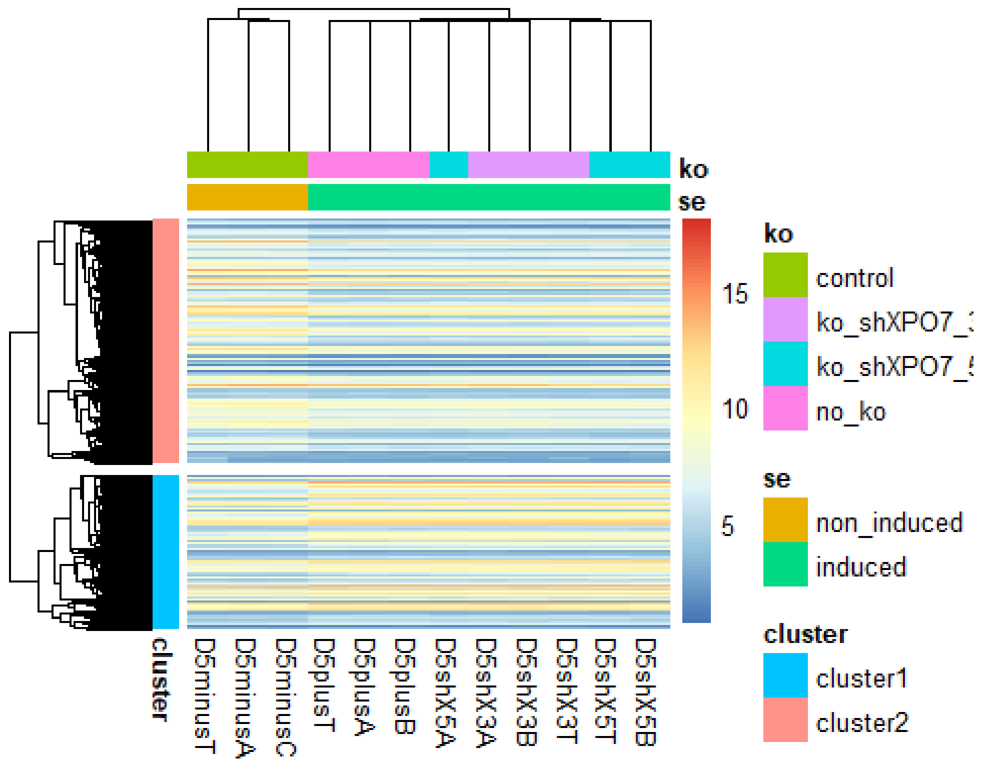
```
# validation of the model for k=2
k <- 2
cnts_g_cutg <- cutree(as.dendrogram(cnts_g_clust), k = k)
sil <- silhouette(cnts_g_cutg, dist = cnts_g_dist)
rownames(sil) <- row.names(rld_cnts_Sig)


pdf()
plot(sil, main = "Silhouette plot",
     cex.names = 0.8, col = 2:(k + 1), nmax = 100)
dev.off()

## png
##   2
```



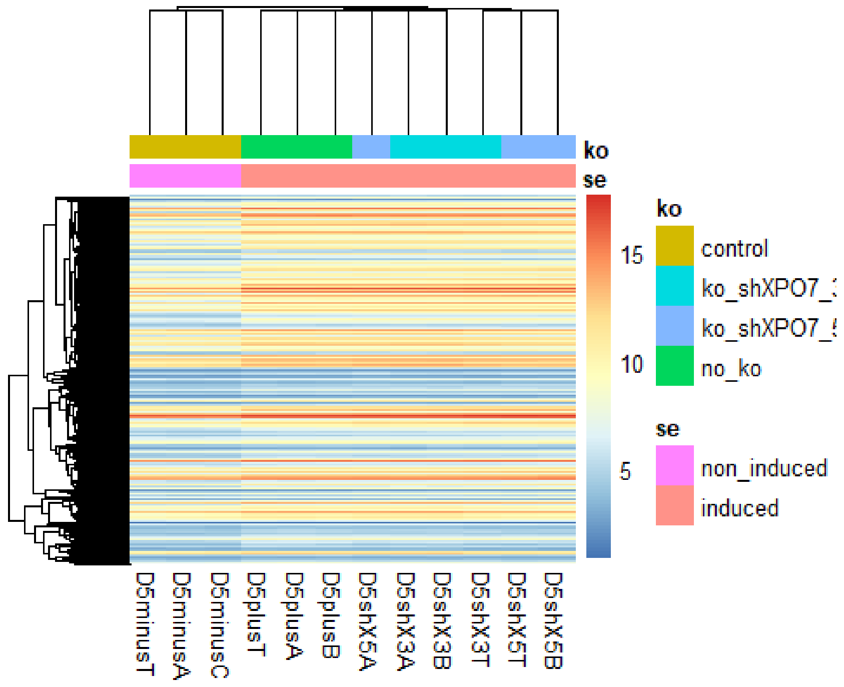
```
# gene number for each cluster
table(cnts_g_cutg)

## cnts_g_cutg
##    1    2
## 1075 1699


# Correspond clusters with gene ids for later usages

cnts_g_cutgr <- data.frame(cnts_g_cutg, row.names(rld_cnts_Sig))
names(cnts_g_cutgr)  <- c("cluster", "ENSEMBL")
rownames(cnts_g_cutgr) <- cnts_g_cutgr$ENSEMBL
head(cnts_g_cutgr)

##                 cluster         ENSEMBL
## ENSG00000223764       1 ENSG00000223764
## ENSG00000217801       1 ENSG00000217801
## ENSG00000162576       2 ENSG00000162576
## ENSG00000179403       2 ENSG00000179403
## ENSG00000205090       2 ENSG00000205090
## ENSG00000169885       1 ENSG00000169885


# dendrogram with k-mean clustering
plot(as.dendrogram(cnts_g_clust), leaflab = "none",
     main ="Dendrogram for significant differential expressed genes")
rect.hclust(cnts_g_clust, k = 2, border = "red")
```



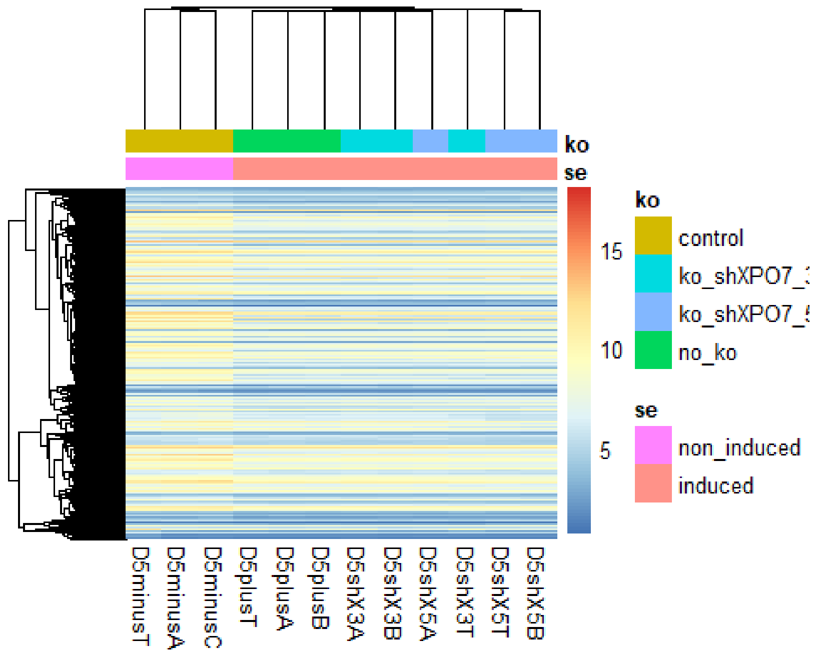
```
# visualize clusters using heatmap

clustgrp <- data.frame(cluster = ifelse(test = cnts_g_cutg == 1, yes = "cluster1", no = "
cluster2"))

pheatmap(rld_cnts_Sig,
         clustering_distance_cols = cnts_dist,
         clustering_distance_rows = cnts_g_dist,
         annotation_col = coldata[ ,c(3:4)],
         annotation_row = clustgrp,
         cutree_rows = 2,
         show_rownames = F)
```



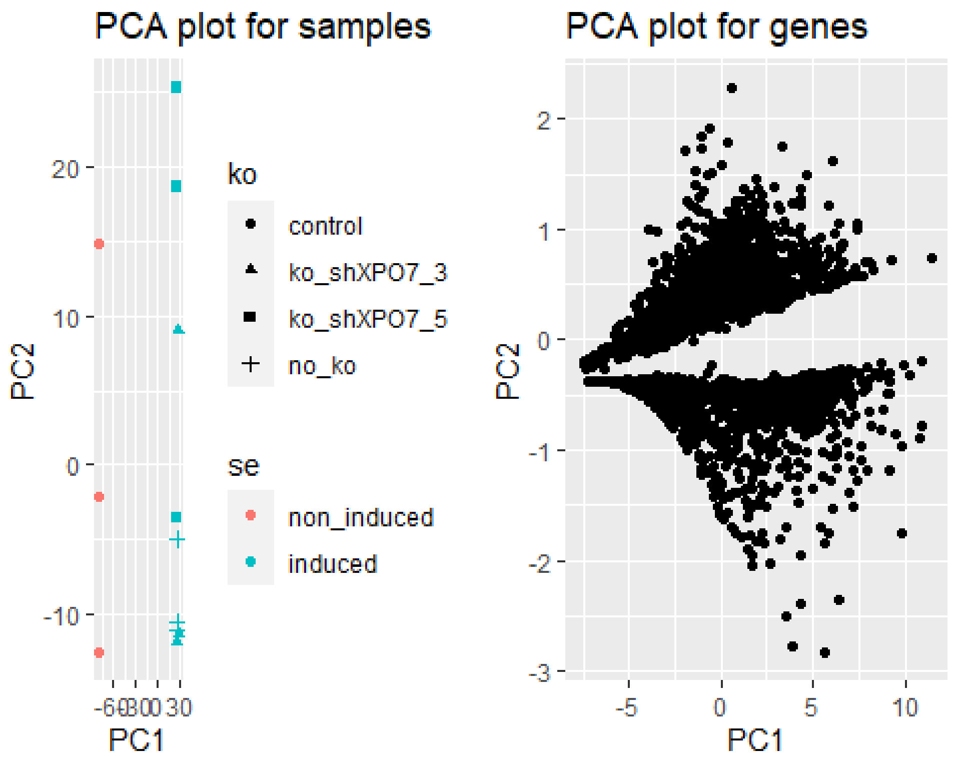
```
# heatmap for cluster
# cluster1
# distance for cluster1
cnts_clust1_dist <- calc_dist(rld_cnts_Sig[cnts_g_cutg==1,])
cnts_clust1_g_dist <- calc_dist(t(rld_cnts_Sig[cnts_g_cutg==1,]))

pheatmap(rld_cnts_Sig[cnts_g_cutg==1,],
         clustering_distance_cols = cnts_clust1_dist,
         clustering_distance_rows = cnts_clust1_g_dist,
         annotation_col = coldata[ ,c(3:4)],
         show_rownames = F)
```



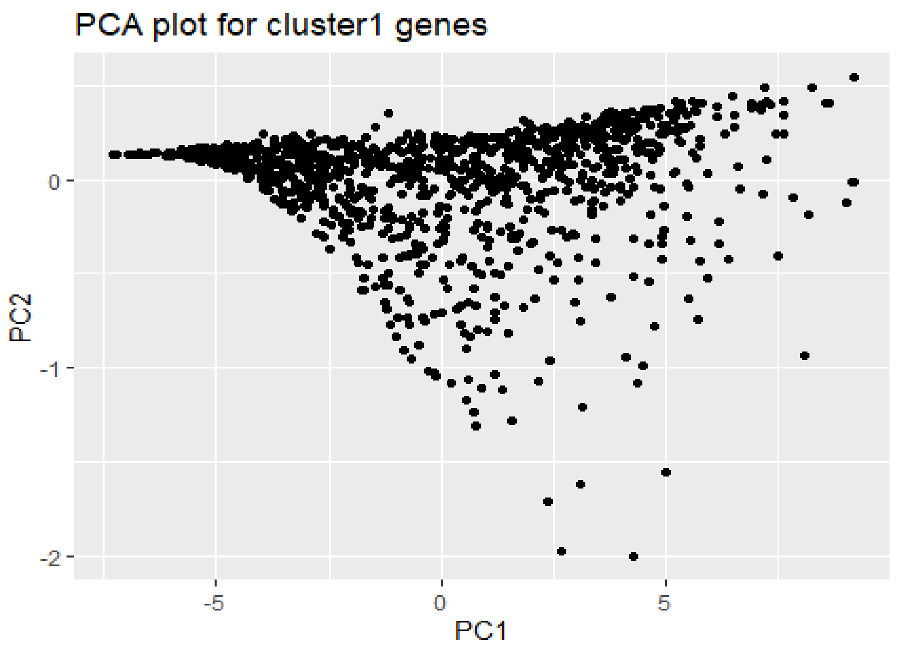
```
# cluster2
# distance for cluster2
cnts_clust2_dist <- calc_dist(rld_cnts_Sig[cnts_g_cutg==2,])
cnts_clust2_g_dist <- calc_dist(t(rld_cnts_Sig[cnts_g_cutg==2,]))

pheatmap(rld_cnts_Sig[cnts_g_cutg==2,],
         clustering_distance_cols = cnts_clust2_dist,
         clustering_distance_rows = cnts_clust2_g_dist,
         annotation_col = coldata[ ,c(3:4)],
         show_rownames = F)
```



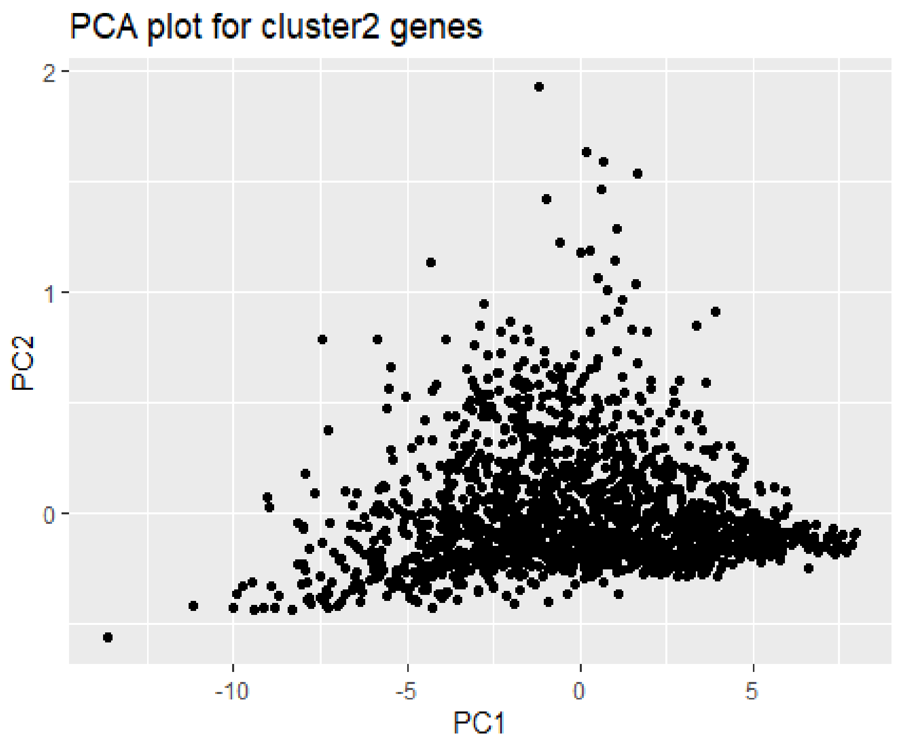
```r
# perform PCA

# for samples
dataProcomp <- prcomp(t(rld_cnts_Sig), center = T, scale = T)
summary(dataProcomp)

## Importance of components:
##                            PC1      PC2     PC3     PC4     PC5    PC6     PC7
## Standard deviation     48.2342 13.44162 7.84865 7.05681 5.38021 4.8846 4.78865
## Proportion of Variance  0.8387  0.06513 0.02221 0.01795 0.01043 0.0086 0.00827
## Cumulative Proportion   0.8387  0.90383 0.92603 0.94399 0.95442 0.9630 0.97129
##                           PC8     PC9    PC10    PC11      PC12
## Standard deviation     4.7692 4.45030 4.33059 4.28246 4.919e-14
## Proportion of Variance 0.0082 0.00714 0.00676 0.00661 0.000e+00
## Cumulative Proportion  0.9795 0.98663 0.99339 1.00000 1.000e+00

pcadata <- cbind(as.data.frame(dataProcomp$x), coldata)
ggplot(pcadata, aes(PC1, PC2, col=se, shape=ko))+
  geom_point()+
  ggtitle("PCA plot for samples") -> pca_sp

# for genes
dataProcomp_gene <- prcomp(rld_cnts_Sig, center = T, scale = T)
summary(dataProcomp_gene)

## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.4058 0.60539 0.13243 0.07161 0.05947 0.03875 0.03635
## Proportion of Variance 0.9666 0.03054 0.00146 0.00043 0.00029 0.00013 0.00011
## Cumulative Proportion  0.9666 0.99716 0.99862 0.99905 0.99934 0.99947 0.99958
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.03457 0.03307 0.03103 0.03029 0.02968
## Proportion of Variance 0.00010 0.00009 0.00008 0.00008 0.00007
## Cumulative Proportion  0.99968 0.99977 0.99985 0.99993 1.00000

# anno_Sig <- gene_anno[rownames(rld_cnts_Sig),1]
pcadata_gene <- as.data.frame(dataProcomp_gene$x)

ggplot(pcadata_gene, aes(PC1, PC2))+
  geom_point()+
  ggtitle("PCA plot for genes") -> pca_gp

grid.arrange(pca_sp,pca_gp, nrow = 1)
```



```
# cluster1
dataProcomp_clust1_gene <- prcomp(rld_cnts_Sig[cnts_g_cutg==1,], center = T, scale = T)

pcadata_clust1_gene <- as.data.frame(dataProcomp_clust1_gene$x)

ggplot(pcadata_clust1_gene, aes(PC1, PC2))+
  geom_point()+
  ggtitle("PCA plot for cluster1 genes")
```



```
# cluster2
dataProcomp_clust2_gene <- prcomp(rld_cnts_Sig[cnts_g_cutg==2,], center = T, scale = T)

pcadata_clust2_gene <- as.data.frame(dataProcomp_clust2_gene$x)

ggplot(pcadata_clust2_gene, aes(PC1, PC2))+
  geom_point()+
  ggtitle("PCA plot for cluster2 genes")
```



```
# Part 5
################################################################################
# Gene Enrichment Analysis

# === GO-term analysis using GOstats package
# significant genes enrichment analysis
# for all significant genes
geneSig_param <- new("GOHyperGParams",
                       geneIds = gene_anno[geneSig,3],
                       universeGeneIds=gene_anno$ENTREZID,
                       annotation="org.Hs.eg.db",
                       ontology="BP",
                       pvalueCutoff=0.001,
                       testDirection="over")

## Warning in makeValidParams(.Object): removing duplicate IDs in geneIds

## Warning in makeValidParams(.Object): removing duplicate IDs in universeGeneIds

geneSig_overRepresented <- hyperGTest(geneSig_param)
head(summary(geneSig_overRepresented),10)
```

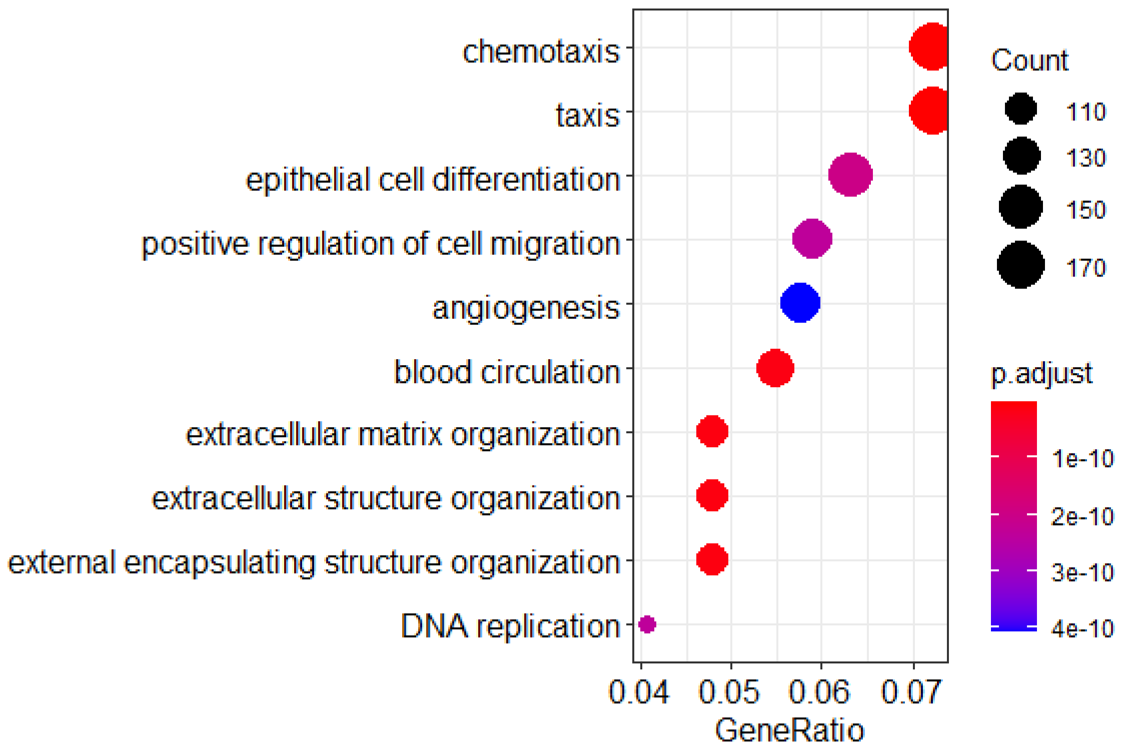
```
##        GOBPID       Pvalue OddsRatio  ExpCount Count Size
## 1  GO:0032501 4.228145e-32  1.710208  956.4758  1213 5491
## 2  GO:0048856 7.609839e-28  1.663482  793.4342  1024 4555
## 3  GO:0048731 2.395769e-26  1.673295  651.6438   865 3741
## 4  GO:0050896 2.962750e-26  1.619612 1158.7101  1391 6652
## 5  GO:0051239 3.661596e-26  1.843687  362.8372   538 2083
## 6  GO:0007275 1.511102e-25  1.639370  727.2421   943 4175
## 7  GO:0032502 7.974447e-25  1.604682  861.0198  1081 4943
## 8  GO:0048513 3.713889e-23  1.699309  462.4737   641 2655
## 9  GO:0009888 1.431934e-22  1.894751  261.9814   404 1504
## 10 GO:0023052 2.439066e-22  1.566653  834.5430  1041 4791
##                                              Term
## 1               multicellular organismal process
## 2               anatomical structure development
## 3                             system development
## 4                           response to stimulus
## 5  regulation of multicellular organismal process
## 6             multicellular organism development
## 7                          developmental process
## 8                        animal organ development
## 9                             tissue development
## 10                                     signaling
```

```
# clust1
geneSig_clust1_param <- new("GOHyperGParams",
                     geneIds = gene_anno[geneSig_clust1,3],
                     universeGeneIds=gene_anno$ENTREZID,
                     annotation="org.Hs.eg.db",
                     ontology="BP",
                     pvalueCutoff=0.001,
                     testDirection="over")
```

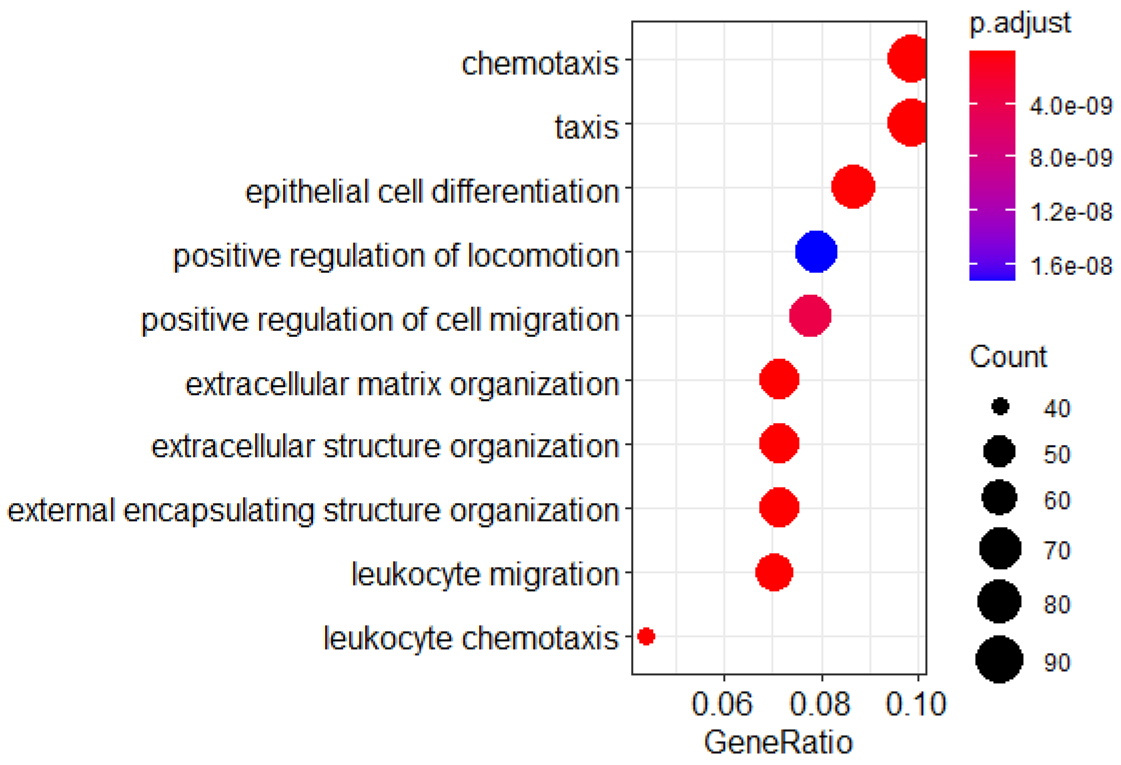
```
## Warning in makeValidParams(.Object): removing duplicate IDs in universeGeneIds
```

```
geneSig_clust1_overRepresented <- hyperGTest(geneSig_clust1_param)
head(summary(geneSig_clust1_overRepresented),10)
```

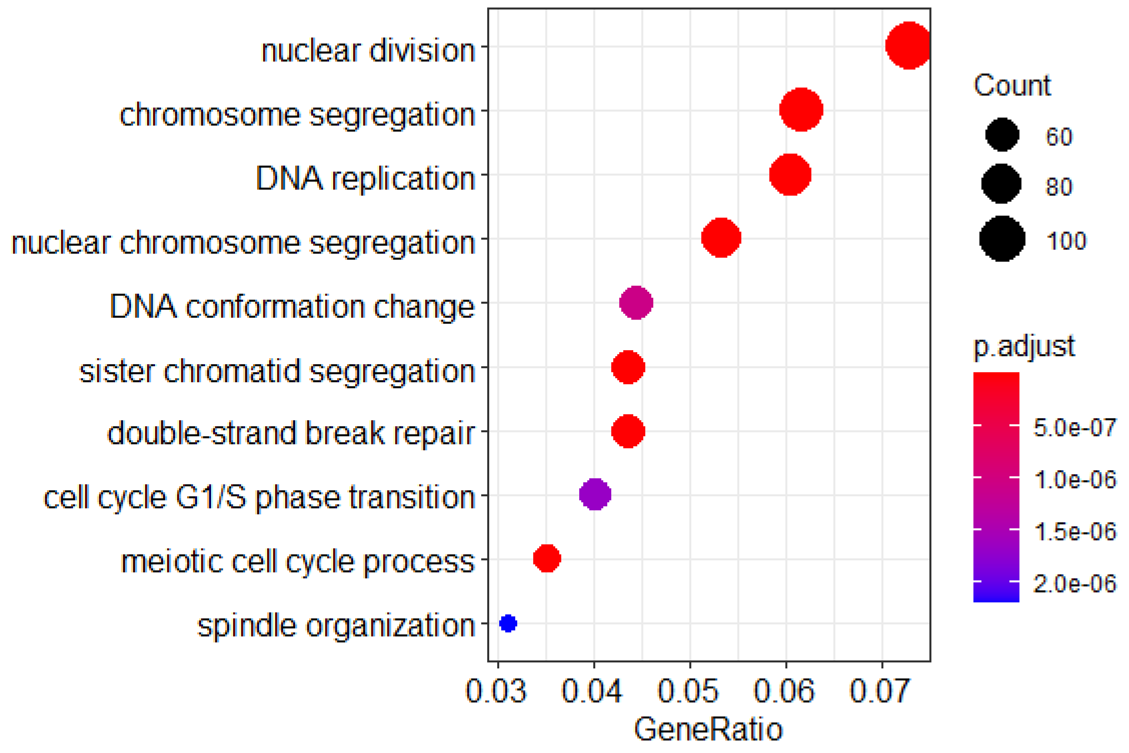
```
##        GOBPID       Pvalue OddsRatio ExpCount Count Size
## 1  GO:0032501 1.050736e-30  2.210726 370.5633   537 5491
## 2  GO:0023052 1.484721e-30  2.209231 323.3234   487 4791
## 3  GO:0007154 1.348626e-29  2.179839 324.0657   485 4802
## 4  GO:0050896 6.121526e-27  2.125107 448.9140   604 6652
## 5  GO:0007165 9.776974e-26  2.072640 297.8813   445 4414
## 6  GO:0048856 2.211560e-24  2.025746 307.3968   451 4555
## 7  GO:0042221 4.964571e-24  2.091775 222.5674   355 3298
## 8  GO:0048731 1.202684e-23  2.042498 252.4635   388 3741
## 9  GO:0007275 1.895799e-23  2.011270 281.7523   420 4175
## 10 GO:0051239 3.630128e-23  2.259854 140.5724   253 2083
##                                              Term
## 1               multicellular organismal process
## 2                                      signaling
## 3                             cell communication
## 4                           response to stimulus
## 5                             signal transduction
## 6                   anatomical structure development
## 7                            response to chemical
## 8                              system development
## 9                 multicellular organism development
## 10 regulation of multicellular organismal process

# clust2
geneSig_clust2_param <- new("GOHyperGParams",
                            geneIds = gene_anno[geneSig_clust2,3],
                            universeGeneIds=gene_anno$ENTREZID,
                            annotation="org.Hs.eg.db",
                            ontology="BP",
                            pvalueCutoff=0.001,
                            testDirection="over")

## Warning in makeValidParams(.Object): removing duplicate IDs in universeGeneIds

geneSig_clust2_overRepresented <- hyperGTest(geneSig_clust2_param)
head(summary(geneSig_clust2_overRepresented),10)

##        GOBPID       Pvalue OddsRatio  ExpCount Count Size
## 1  GO:0000278 1.560270e-23  2.483846 100.30191   201  940
## 2  GO:1903047 1.781540e-23  2.605924  87.07059   182  816
## 3  GO:0007049 2.043632e-23  2.097059 177.34231   303 1662
## 4  GO:0022402 2.715214e-23  2.260091 133.70031   246 1253
## 5  GO:0006260 2.754887e-22  4.171327  28.81012    87  270
## 6  GO:0051301 2.166016e-20  2.832766  58.79399   133  551
## 7  GO:0000280 7.333819e-20  3.236371  41.72133   105  391
## 8  GO:0006261 9.148723e-20  5.436086  16.32574    59  153
## 9  GO:0007059 3.989668e-19  3.543734  32.97159    89  309
## 10 GO:0048285 7.630328e-19  2.995368  46.73642   111  438
##                              Term
## 1                mitotic cell cycle
## 2        mitotic cell cycle process
## 3                        cell cycle
## 4                cell cycle process
## 5                   DNA replication
## 6                     cell division
## 7                  nuclear division
## 8  DNA-dependent DNA replication
## 9            chromosome segregation
## 10               organelle fission

# ============= GO-term Over-Representation Analysis using clusterProfiler ==============
# for significant genes

# obtain ENTREZID list for significant genes
# geneSig_entz <- gene_anno[unique(geneSig),3]

geneSig_enrichgo <- enrichGO(gene = gene_anno[unique(geneSig),3],
                             universe = gene_anno$ENTREZID,
                             OrgDb = org.Hs.eg.db,
                              keyType = "ENTREZID",
                              readable = T,
                              ont = "BP",
                              pvalueCutoff = 0.05,
                              qvalueCutoff = 0.10)

# === cluster1

geneSig_clust1_enrichgo <- enrichGO(gene = gene_anno[geneSig_clust1,3],
                             universe = gene_anno$ENTREZID,
                             OrgDb = org.Hs.eg.db,
                             keyType = "ENTREZID",
                             readable = T,
                             ont = "BP",
                             pvalueCutoff = 0.05,
                             qvalueCutoff = 0.10)

# === cluster2

geneSig_clust2_enrichgo <- enrichGO(gene = gene_anno[geneSig_clust2,3],
                                    universe = gene_anno$ENTREZID,
                                    OrgDb = org.Hs.eg.db,
                                    keyType = "ENTREZID",
                                    readable = T,
                                    ont = "BP",
                                    pvalueCutoff = 0.05,
                                    qvalueCutoff = 0.10)

## plotting
dotplot(geneSig_enrichgo)
```



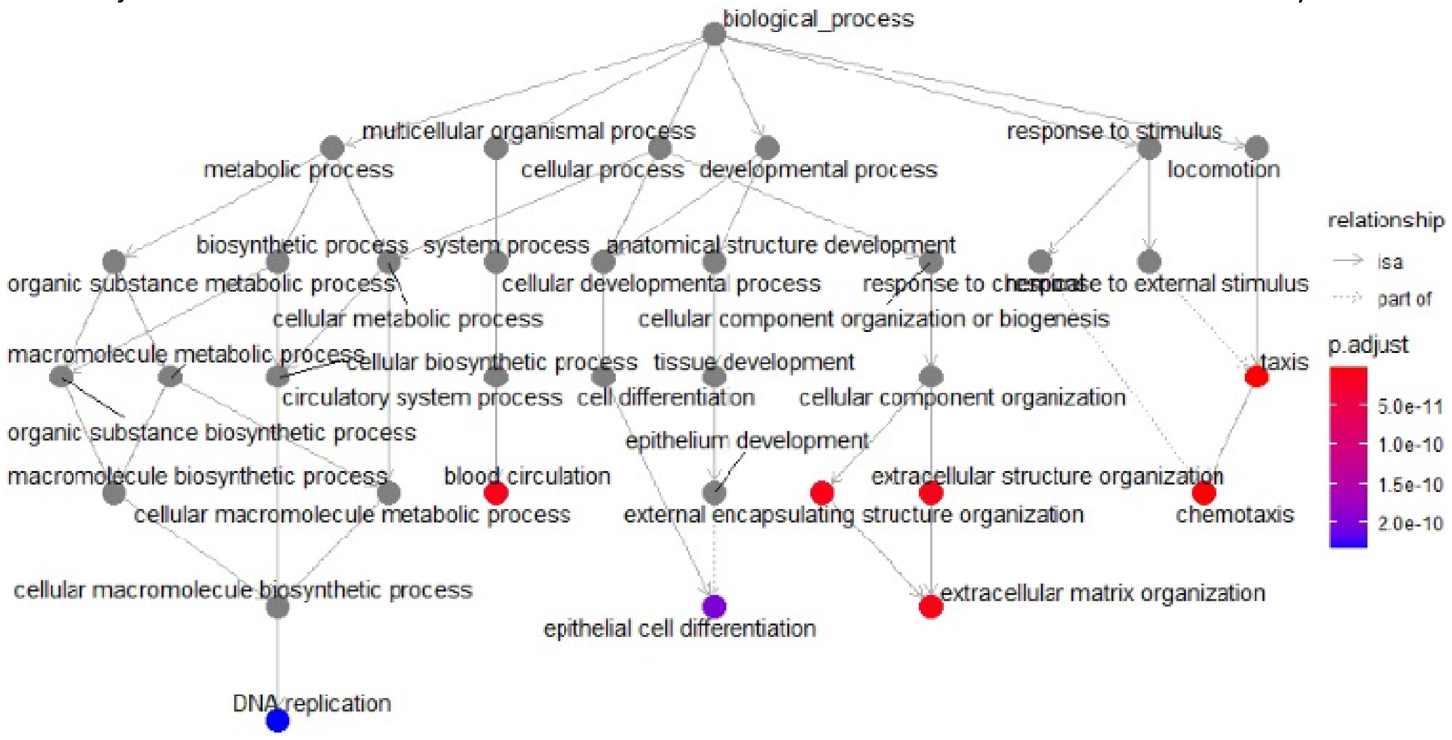
```
dotplot(geneSig_clust1_enrichgo)
```



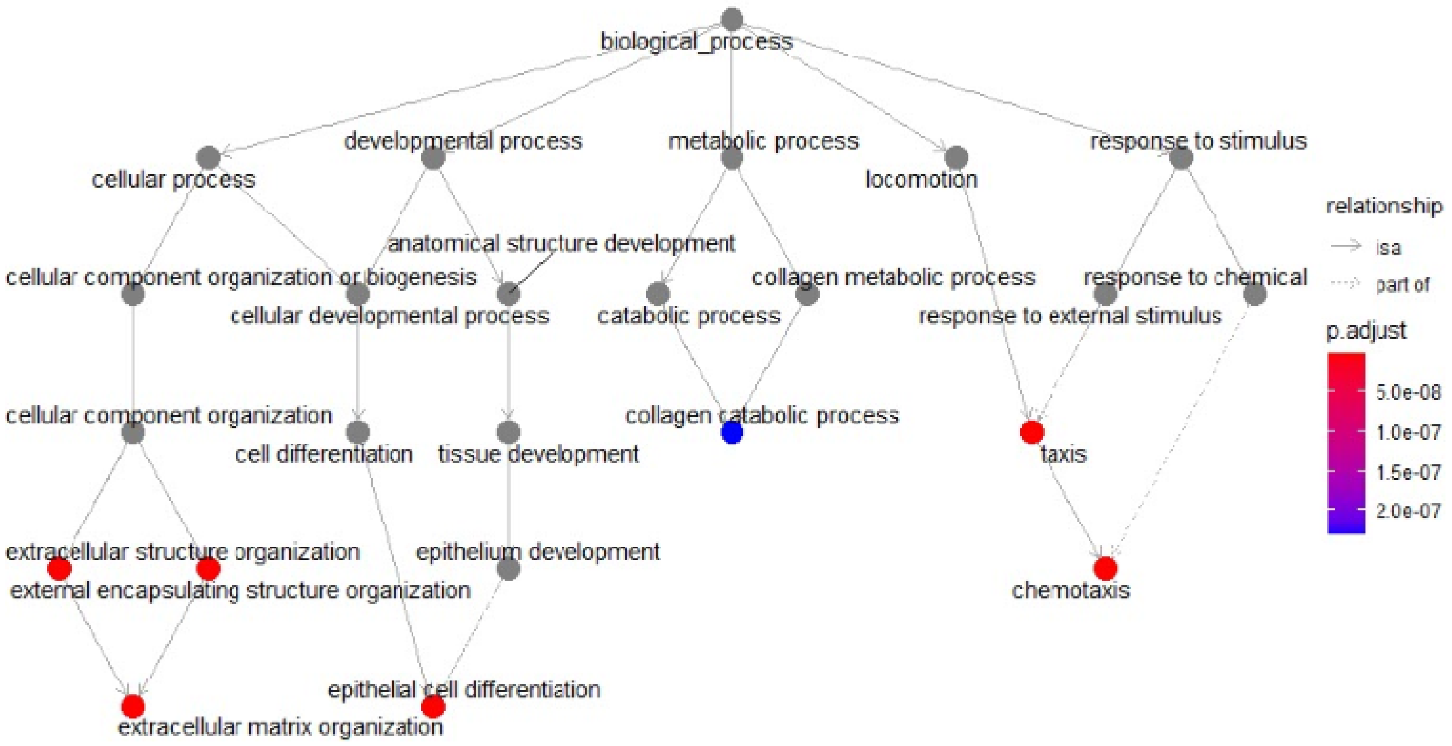
```
dotplot(geneSig_clust2_enrichgo)
```



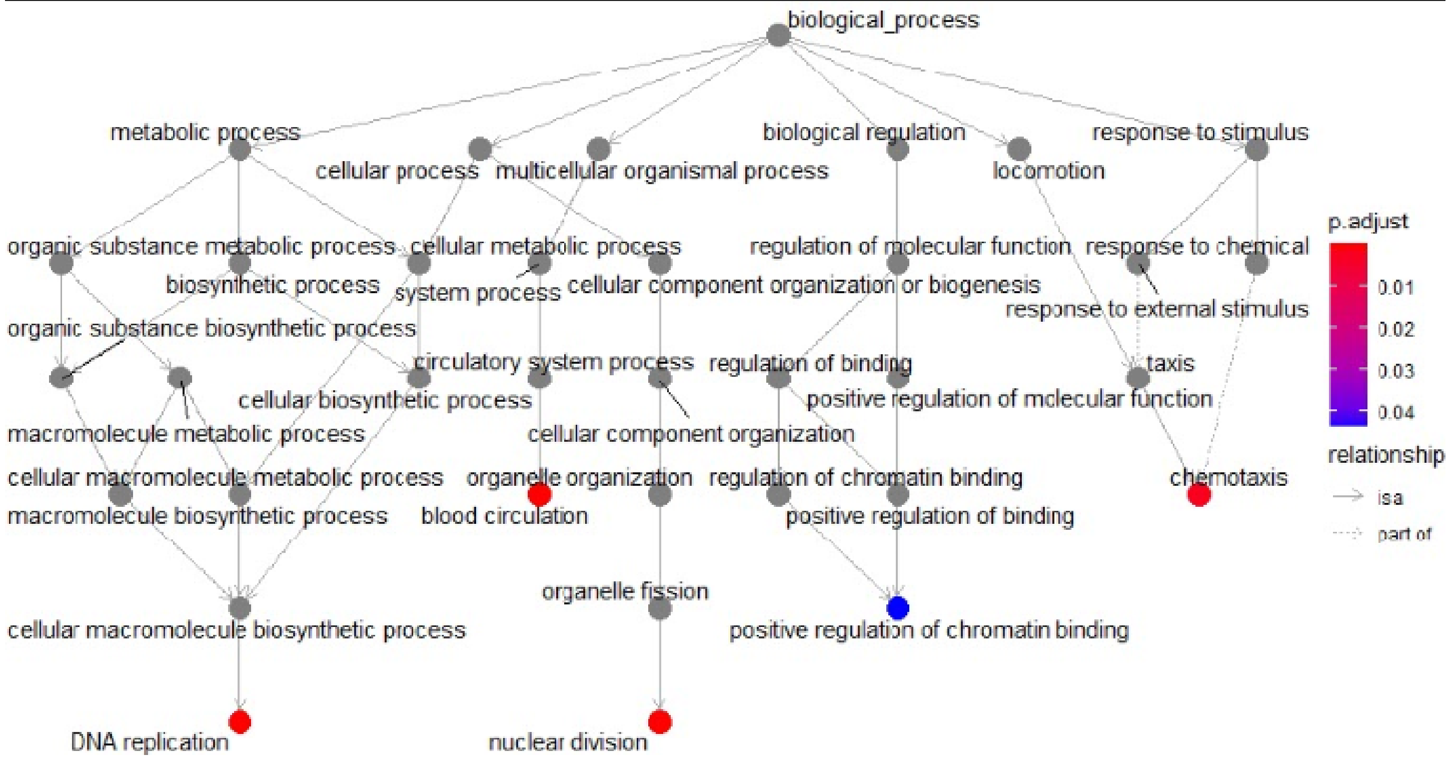
```
## use simplify to remove redundant terms
geneSig_enrichgo <- simplify(geneSig_enrichgo,
                             cutoff=0.3, by="p.adjust",
                             select_fun=min)


goplot(geneSig_enrichgo,showCategory = 10)

## Warning: ggrepel: 44 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```


biological_process
multicellular organismal process
metabolic process
cellular process
developmental process
response to stimulus
locomotion
biosynthetic process
system process
anatomical structure development
organic substance metabolic process
cellular developmental process
response to chemical
response to external stimulus
cellular metabolic process
cellular component organization or biogenesis
macromolecule metabolic process
cellular biosynthetic process
tissue development
taxis
circulatory system process
cell differentiation
cellular component organization
organic substance biosynthetic process
epithelium development
macromolecule biosynthetic process
blood circulation
extracellular structure organization
cellular macromolecule metabolic process
external encapsulating structure organization
chemotaxis
cellular macromolecule biosynthetic process
extracellular matrix organization
epithelial cell differentiation
DNA replication
relationship
isa
part of
p.adjust
5.0e-11
1.0e-10
1.5e-10
2.0e-10

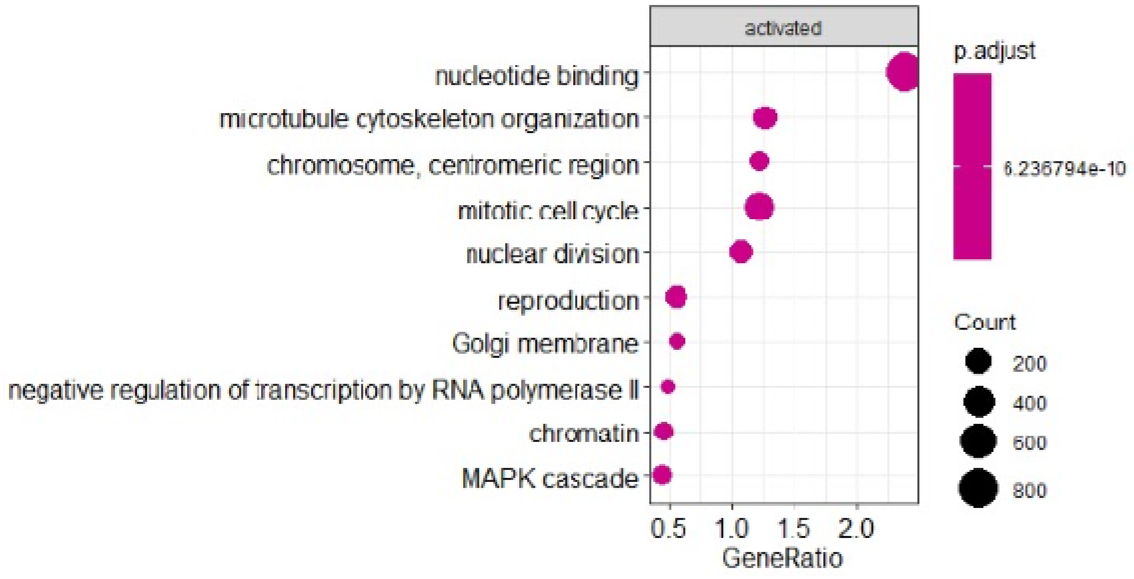
```
## use simplify to remove redundant terms
geneSig_clust1_enrichgo <- simplify(geneSig_clust1_enrichgo,
                                    cutoff=0.3, by="p.adjust",
                                    select_fun=min)


goplot(geneSig_clust1_enrichgo,showCategory = 10)

## Warning: ggrepel: 14 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```


biological_process
cellular process
developmental process
metabolic process
locomotion
response to stimulus
anatomical structure development
cellular component organization or biogenesis
collagen metabolic process
response to chemical
cellular developmental process
catabolic process
response to external stimulus
cellular component organization
collagen catabolic process
cell differentiation
tissue development
taxis
extracellular structure organization
epithelium development
external encapsulating structure organization
chemotaxis
epithelial cell differentiation
extracellular matrix organization
relationship
isa
part of
p.adjust
5.0e-08
1.0e-07
1.5e-07
2.0e-07

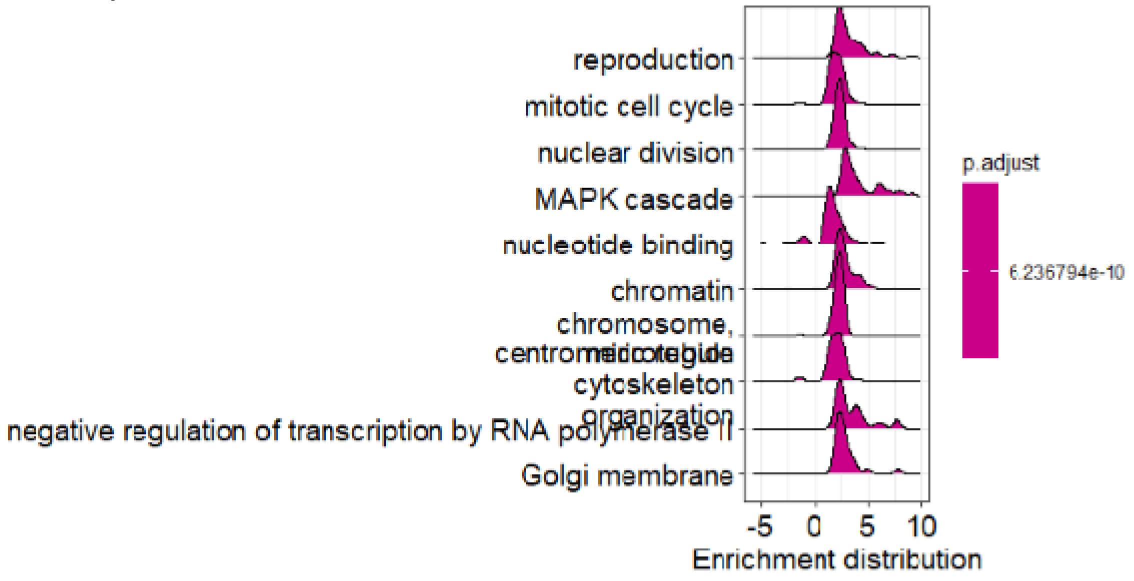
```
## use simplify to remove redundant terms
geneSig_clust2_enrichgo <- simplify(geneSig_clust2_enrichgo,
                                    cutoff=0.3, by="p.adjust",
                                    select_fun=min)


goplot(geneSig_clust2_enrichgo, showCategory = 10)

## Warning: ggrepel: 20 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```


biological_process
metabolic process
cellular process
multicellular organismal process
biological regulation
locomotion
response to stimulus
organic substance metabolic process
cellular metabolic process
regulation of molecular function
response to chemical
biosynthetic process
system process
cellular component organization or biogenesis
response to external stimulus
organic substance biosynthetic process
circulatory system process
regulation of binding
taxis
cellular biosynthetic process
positive regulation of molecular function
macromolecule metabolic process
cellular component organization
cellular macromolecule metabolic process
organelle organization
regulation of chromatin binding
chemotaxis
macromolecule biosynthetic process
blood circulation
positive regulation of binding
organelle fission
cellular macromolecule biosynthetic process
positive regulation of chromatin binding
DNA replication
nuclear division
p.adjust
0.01
0.02
0.03
0.04
relationship
isa
part of

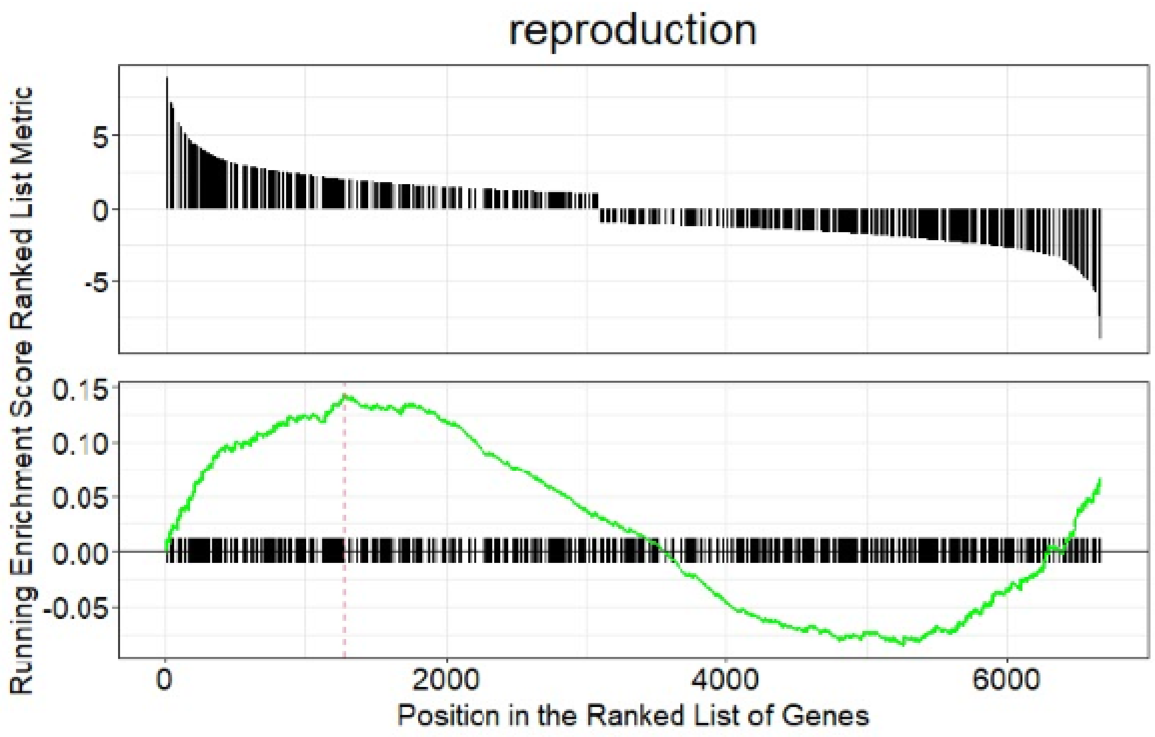
```
# ===
# GSEA plotting

# get a gene list(entrez id) with sorted foldchanges in decreasing order
# for significant genes
geneSig_foldchag_sort <- geneSig_sta[order(as.numeric(geneSig_sta$log2foldchangeSig), dec
reasing = T), c(1:2)]
geneSig_foldchag_sort$entrezid <- gene_anno[geneSig_foldchag_sort$geneSig,3]


geneSig_foldchang_list <- as.numeric(geneSig_foldchag_sort$log2foldchangeSig)
names(geneSig_foldchang_list) <- geneSig_foldchag_sort$entrezid

geneSig_gse <- gseGO(geneList = geneSig_foldchang_list,
                     ont = "ALL",
                     keyType = "ENTREZID",
                     minGSSize = 10,
                     maxGSSize = 500,
                     pvalueCutoff = .05,
                     verbose = TRUE,
                     OrgDb = org.Hs.eg.db,
                     pAdjustMethod = "fdr")

## preparing geneSet collections...

## GSEA analysis...

## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize,
## gseaParam, : There are duplicate gene names, fgsea may produce unexpected
## results.
## Warning in fgseaMultilevel(...): 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...

# output
require(DOSE)
dotplot(geneSig_gse, showCategory=10, split=".sign") +
  facet_grid(.~.sign)
```



```
# pathway enrichment distribution with p-value
ridgeplot(geneSig_gse, label_format = 20, showCategory = 10)+
  labs(x="Enrichment distribution")

## Picking joint bandwidth of 0.284
```

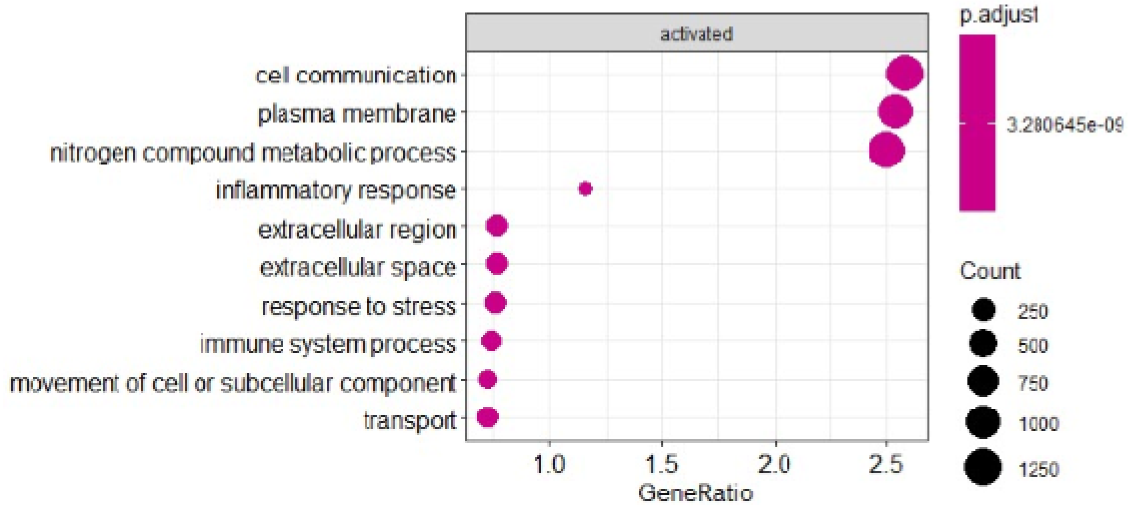


```
# GSEA plot
gseaplot(geneSig_gse, by="all",
         title=geneSig_gse$Description[1], geneSetID = 1)
```

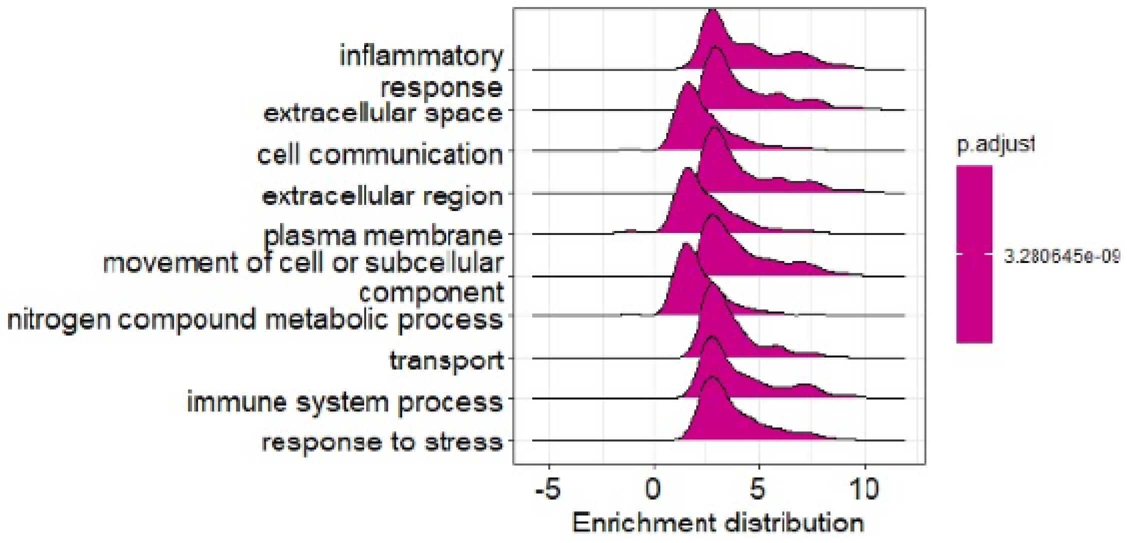


```
# === cluster 1
# get a gene list(entrez id) of decreasing sorted foldchanges
# for cluster1 genes
geneSig_clust1_sta <- geneSig_sta[geneSig_sta$geneSig %in% geneSig_clust1,]

geneSig_clust1_foldchag_sort <-
  geneSig_clust1_sta[order(as.numeric(geneSig_clust1_sta$log2foldchangeSig), decreasing =
 T), c(1:2)]
```

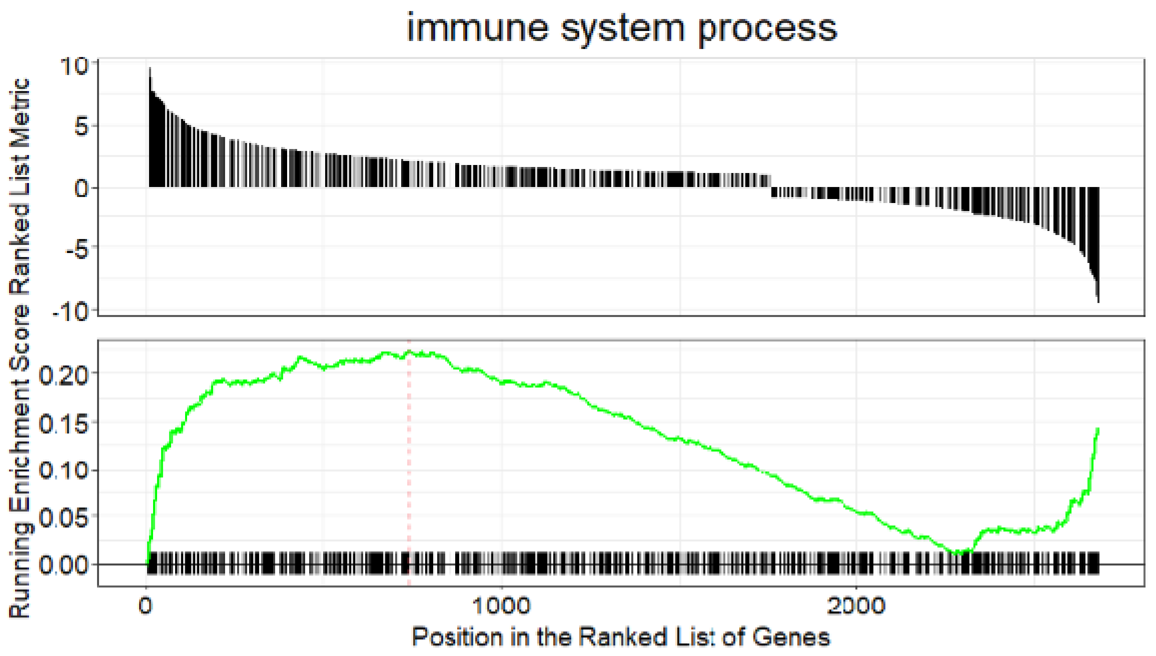
```
geneSig_clust1_foldchag_sort$entrezid <- gene_anno[geneSig_clust1_foldchag_sort$geneSig,
3]

geneSig_clust1_foldchang_list <- as.numeric(geneSig_clust1_foldchag_sort$log2foldchangeSi
g)

names(geneSig_clust1_foldchang_list) <- geneSig_clust1_foldchag_sort$entrezid

geneSig_clust1_gse <- gseGO(geneList = geneSig_clust1_foldchang_list,
                     ont = "ALL",
                     keyType = "ENTREZID",
                     minGSSize = 10,
                     maxGSSize = 500,
                     pvalueCutoff = .05,
                     verbose = TRUE,
                     OrgDb = org.Hs.eg.db,
                     pAdjustMethod = "fdr")

## preparing geneSet collections...

## GSEA analysis...

## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize,
## gseaParam, : There are duplicate gene names, fgsea may produce unexpected
## results.

## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize,
## gseaParam, : 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...

# output
dotplot(geneSig_clust1_gse, showCategory=10, split=".sign") +
  facet_grid(.~.sign)
```



```
# pathway enrichment distribution with p-value
ridgeplot(geneSig_clust1_gse, label_format = 20, showCategory = 10)+
  labs(x="Enrichment distribution")

## Picking joint bandwidth of 0.416
```

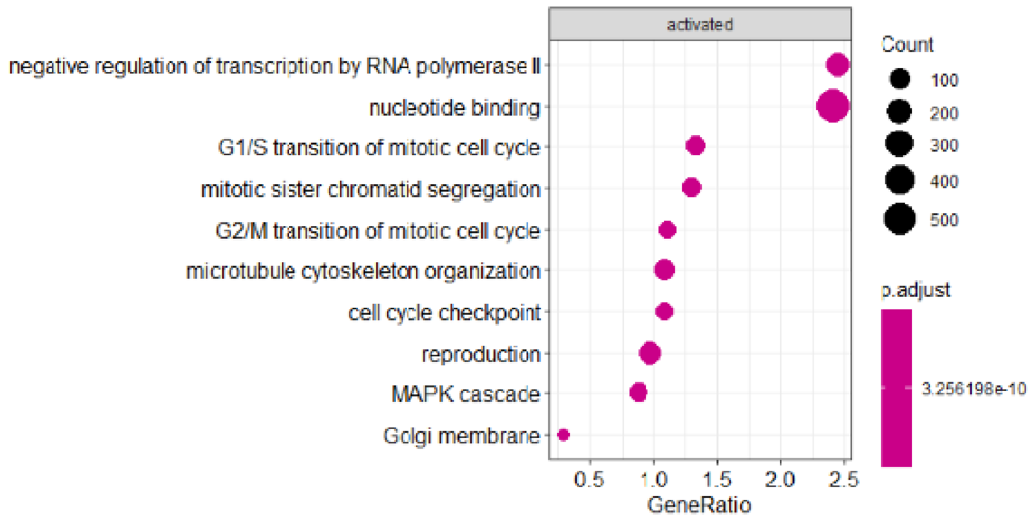


```
# GSEA plot
gseaplot(geneSig_clust1_gse, by="all",
         title=geneSig_clust1_gse$Description[1], geneSetID = 1)
```

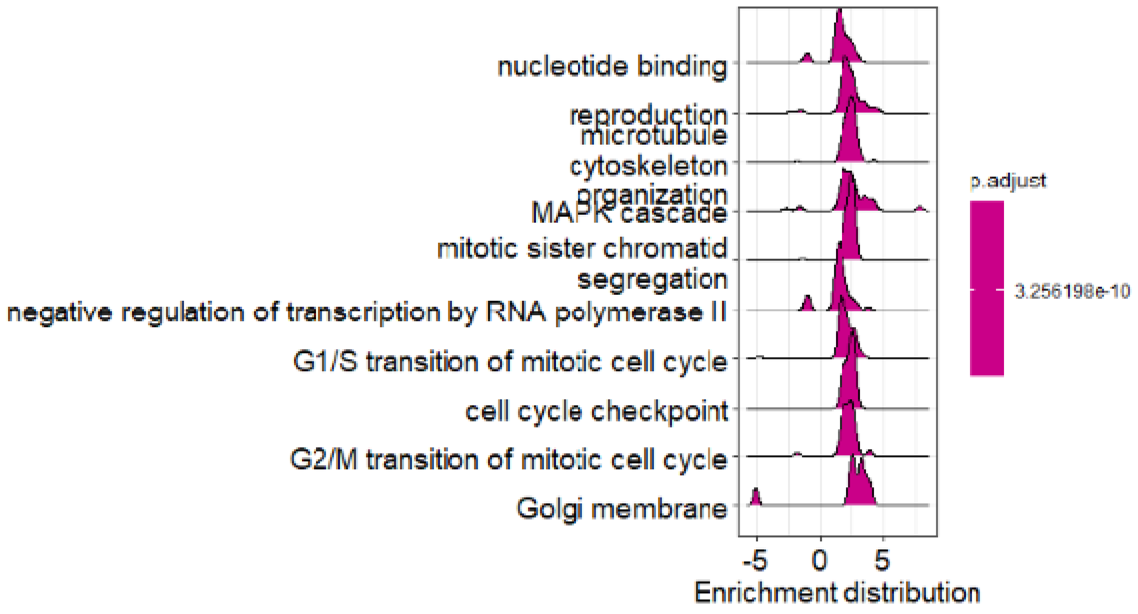


```
#
#  === cluster 2
#
geneSig_clust2_sta <- geneSig_sta[geneSig_sta$geneSig %in% geneSig_clust2,]

geneSig_clust2_foldchag_sort <-
```

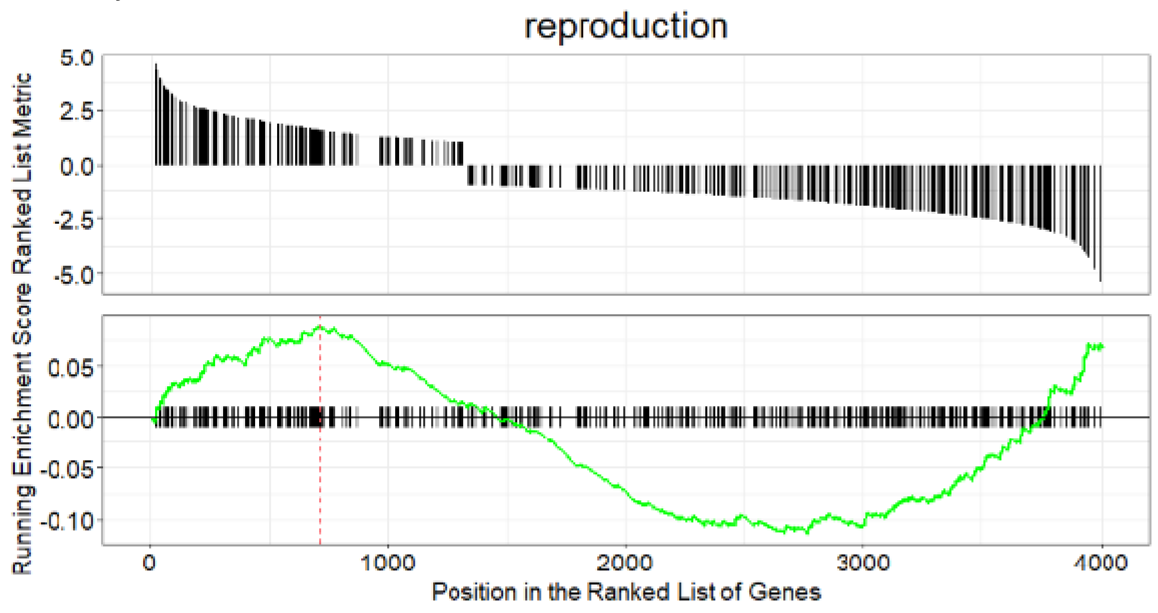
```
  geneSig_clust2_sta[order(as.numeric(geneSig_clust2_sta$log2foldchangeSig), decreasing =
 T), c(1:2)]

geneSig_clust2_foldchag_sort$entrezid <- gene_anno[geneSig_clust2_foldchag_sort$geneSig,
3]

geneSig_clust2_foldchang_list <- as.numeric(geneSig_clust2_foldchag_sort$log2foldchangeSi
g)

names(geneSig_clust2_foldchang_list) <- geneSig_clust2_foldchag_sort$entrezid

geneSig_clust2_gse <- gseGO(geneList = geneSig_clust2_foldchang_list,
                            ont = "ALL",
                            keyType = "ENTREZID",
                            minGSSize = 10,
                            maxGSSize = 500,
                            pvalueCutoff = .05,
                            verbose = TRUE,
                            OrgDb = org.Hs.eg.db,
                            pAdjustMethod = "fdr")

## preparing geneSet collections...

## GSEA analysis...

## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize,
## gseaParam, : There are duplicate gene names, fgsea may produce unexpected
## results.

## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize,
## gseaParam, : 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...

# output
#
dotplot(geneSig_clust2_gse, showCategory=10, split=".sign") +
  facet_grid(.~.sign)
```



```
# pathway enrichment distribution with p-value
ridgeplot(geneSig_clust2_gse, label_format = 20, showCategory = 10)+
  labs(x="Enrichment distribution")

## Picking joint bandwidth of 0.203
```



```
# GSEA plot
gseaplot(geneSig_clust2_gse, by="all",
         title=geneSig_clust2_gse$Description[1], geneSetID = 1)
```


reproduction
Ranked List Metric
5.0
2.5
0.0
-2.5
-5.0
Running Enrichment Score
0.05
0.00
-0.05
-0.10
0
1000
2000
3000
4000
Position in the Ranked List of Genes

```
# top 10 genes
# for geneSig, cluster1, and cluster2
print("Top fold change genes in significant gene list")

## [1] "Top fold change genes in significant gene list"

head((gene_anno[unique(geneSig_foldchag_sort$geneSig), c(1:2)]),10)

##                 symbol                                 geneName
## ENSG00000170373   CST1                               cystatin SN
## ENSG00000240386  LCE1F                late cornified envelope 1F
## ENSG00000005001 PRSS22                        serine protease 22
## ENSG00000108342   CSF3                colony stimulating factor 3
## ENSG00000125538   IL1B                        interleukin 1 beta
## ENSG00000115009  CCL20              C-C motif chemokine ligand 20
## ENSG00000163207    IVL                                 involucrin
## ENSG00000101441   CST4                                 cystatin S
## ENSG00000114654  EFCC1 EF-hand and coiled-coil domain containing 1
## ENSG00000135914  HTR2B             5-hydroxytryptamine receptor 2B

print("Top fold change genes in cluster 1 gene list")

## [1] "Top fold change genes in cluster 1 gene list"

head((gene_anno[unique(geneSig_clust1_foldchag_sort$geneSig), c(1:2)]),10)

##                 symbol                                 geneName
## ENSG00000170373   CST1                               cystatin SN
## ENSG00000240386  LCE1F                late cornified envelope 1F
## ENSG00000005001 PRSS22                        serine protease 22
## ENSG00000108342   CSF3                colony stimulating factor 3
## ENSG00000125538   IL1B                        interleukin 1 beta
## ENSG00000115009  CCL20              C-C motif chemokine ligand 20
## ENSG00000163207    IVL                                 involucrin
## ENSG00000101441   CST4                                 cystatin S
## ENSG00000114654  EFCC1 EF-hand and coiled-coil domain containing 1
## ENSG00000109321   AREG                               amphiregulin

print("Top fold change genes in cluster 2 gene list")

## [1] "Top fold change genes in cluster 2 gene list"

head((gene_anno[unique(geneSig_clust2_foldchag_sort$geneSig), c(1:2)]),10)

##                       symbol
## ENSG00000135914        HTR2B
## ENSG00000169903       TM4SF4
## ENSG00000234546     LNCTAM34A
## ENSG00000103888        CEMIP
## ENSG00000196616        ADH1B
## ENSG00000127951         FGL2
## ENSG00000157766         ACAN
## ENSG00000267288 LOC105371795
## ENSG00000254607 LOC101929427
## ENSG00000107562       CXCL12
##                                                              geneName
## ENSG00000135914                      5-hydroxytryptamine receptor 2B
## ENSG00000169903                    transmembrane 4 L six family member 4
## ENSG00000234546  long non coding transcriptional activator of miR34a
## ENSG00000103888             cell migration inducing hyaluronidase 1
## ENSG00000196616 alcohol dehydrogenase 1B (class I), beta polypeptide
## ENSG00000127951                                    fibrinogen like 2
## ENSG00000157766                                             aggrecan
## ENSG00000267288                          uncharacterized LOC105371795
## ENSG00000254607                          uncharacterized LOC101929427
## ENSG00000107562                       C-X-C motif chemokine ligand 12
```

session_info()

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.1.0 (2021-05-18)
##  os       Windows 10 x64
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  ctype    English_United States.1252
##  tz       America/New_York
##  date     2021-08-22
##
## - Packages -------------------------------------------------------------------
##  package               * version    date       lib source
##  abind                   1.4-5      2016-07-21 [1] CRAN (R 4.1.0)
##  affy                    1.70.0     2021-05-19 [1] Bioconductor
##  affyio                  1.62.0     2021-05-19 [1] Bioconductor
##  annotate              * 1.70.0     2021-05-19 [1] Bioconductor
##  AnnotationDbi         * 1.54.1     2021-06-08 [1] Bioconductor
##  AnnotationForge         1.34.0     2021-05-19 [1] Bioconductor
##  ape                     5.5        2021-04-25 [1] CRAN (R 4.1.0)
##  apeglm                * 1.14.0     2021-05-19 [1] Bioconductor
##  aplot                   0.0.6      2020-09-03 [1] CRAN (R 4.1.0)
##  ash                     1.0-15     2015-09-01 [1] CRAN (R 4.1.0)
##  askpass                 1.1        2019-01-13 [1] CRAN (R 4.1.0)
##  assertthat              0.2.1      2019-03-21 [1] CRAN (R 4.1.0)
##  backports               1.2.1      2020-12-09 [1] CRAN (R 4.1.0)
##  bbmle                   1.0.24     2021-08-05 [1] CRAN (R 4.1.0)
##  bdsmatrix               1.3-4      2020-01-13 [1] CRAN (R 4.1.0)
##  beeswarm                0.4.0      2021-06-01 [1] CRAN (R 4.1.0)
##  Biobase               * 2.52.0     2021-05-19 [1] Bioconductor
##  BiocGenerics          * 0.38.0     2021-05-19 [1] Bioconductor
##  BiocManager             1.30.16    2021-06-15 [1] CRAN (R 4.1.0)
##  BiocParallel            1.26.1     2021-07-04 [1] Bioconductor
##  Biostrings              2.60.2     2021-08-05 [1] Bioconductor
##  bit                     4.0.4      2020-08-04 [1] CRAN (R 4.1.0)
##  bit64                   4.0.5      2020-08-30 [1] CRAN (R 4.1.0)
##  bitops                  1.0-7      2021-04-24 [1] CRAN (R 4.1.0)
##  blob                    1.2.2      2021-07-23 [1] CRAN (R 4.1.0)
##  broom                   0.7.9      2021-07-27 [1] CRAN (R 4.1.0)
##  cachem                  1.0.5      2021-05-15 [1] CRAN (R 4.1.0)
##  callr                   3.7.0      2021-04-20 [1] CRAN (R 4.1.0)
##  car                     3.0-11     2021-06-27 [1] CRAN (R 4.1.0)
##  carData                 3.0-4      2020-05-22 [1] CRAN (R 4.1.0)
##  Category              * 2.58.0     2021-05-19 [1] Bioconductor
##  cellranger              1.1.0      2016-07-27 [1] CRAN (R 4.1.0)
##  cli                     3.0.1      2021-07-17 [1] CRAN (R 4.1.0)
##  cluster              * 2.1.2      2021-04-17 [1] CRAN (R 4.1.0)
##  clusterProfiler      * 4.0.3      2021-08-15 [1] Bioconductor
##  coda                   0.19-4     2020-09-30 [1] CRAN (R 4.1.0)
##  codetools              0.2-18     2020-11-04 [1] CRAN (R 4.1.0)
##  colorspace             2.0-2      2021-06-24 [1] CRAN (R 4.1.0)
##  corpcor                1.6.9      2017-04-01 [1] CRAN (R 4.1.0)
##  cowplot                1.1.1      2020-12-30 [1] CRAN (R 4.1.0)
##  crayon                 1.4.1      2021-02-08 [1] CRAN (R 4.1.0)
##  crosstalk              1.1.1      2021-01-12 [1] CRAN (R 4.1.0)
##  curl                   4.3.2      2021-06-23 [1] CRAN (R 4.1.0)
##  data.table             1.14.0     2021-02-21 [1] CRAN (R 4.1.0)
##  DBI                    1.1.1      2021-01-15 [1] CRAN (R 4.1.0)
##  DelayedArray           0.18.0     2021-05-19 [1] Bioconductor
##  dendextend           * 1.15.1     2021-05-08 [1] CRAN (R 4.1.0)
##  desc                   1.3.0      2021-03-05 [1] CRAN (R 4.1.0)
##  DESeq2               * 1.32.0     2021-05-19 [1] Bioconductor
##  devtools             * 2.4.2      2021-06-07 [1] CRAN (R 4.1.0)
##  digest                 0.6.27     2020-10-24 [1] CRAN (R 4.1.0)
##  DO.db                  2.9        2021-08-17 [1] Bioconductor
##  doParallel             1.0.16     2020-10-16 [1] CRAN (R 4.1.0)
##  DOSE                 * 3.18.1     2021-06-22 [1] Bioconductor
##  doSNOW                 1.0.19     2020-10-16 [1] CRAN (R 4.1.0)
##  downloader             0.4        2015-07-09 [1] CRAN (R 4.1.0)
##  dplyr                  1.0.7      2021-06-18 [1] CRAN (R 4.1.0)
##  ellipsis               0.3.2      2021-04-29 [1] CRAN (R 4.1.0)
##  emdbook                1.3.12     2020-02-19 [1] CRAN (R 4.1.0)
##  EnhancedVolcano      * 1.10.0     2021-05-19 [1] Bioconductor
##  enrichplot           * 1.12.2     2021-07-01 [1] Bioconductor
##  evaluate               0.14       2019-05-28 [1] CRAN (R 4.1.0)
##  extrafont              0.17       2014-12-08 [1] CRAN (R 4.1.0)
##  extrafontdb            1.0        2012-06-11 [1] CRAN (R 4.1.0)
##  fansi                  0.5.0      2021-05-25 [1] CRAN (R 4.1.0)
##  farver                 2.1.0      2021-02-28 [1] CRAN (R 4.1.0)
##  fastmap                1.1.0      2021-01-25 [1] CRAN (R 4.1.0)
##  fastmatch              1.1-3      2021-07-23 [1] CRAN (R 4.1.0)
##  fgsea                  1.18.0     2021-05-19 [1] Bioconductor
##  forcats                0.5.1      2021-01-27 [1] CRAN (R 4.1.0)
##  foreach                1.5.1      2020-10-15 [1] CRAN (R 4.1.0)
##  foreign                0.8-81     2020-12-22 [1] CRAN (R 4.1.0)
##  fs                     1.5.0      2020-07-31 [1] CRAN (R 4.1.0)
##  genefilter             1.74.0     2021-05-19 [1] Bioconductor
##  geneplotter            1.70.0     2021-05-19 [1] Bioconductor
##  generics               0.1.0      2020-10-31 [1] CRAN (R 4.1.0)
##  GenomeInfoDb         * 1.28.1     2021-07-01 [1] Bioconductor
##  GenomeInfoDbData       1.2.6      2021-08-14 [1] Bioconductor
##  GenomicRanges        * 1.44.0     2021-05-19 [1] Bioconductor
##  GEOquery             * 2.60.0     2021-05-20 [1] Bioconductor
##  ggalt                  0.4.0      2017-02-15 [1] CRAN (R 4.1.0)
##  ggbeeswarm           * 0.6.0      2017-08-07 [1] CRAN (R 4.1.0)
##  ggbiplot             * 0.55       2021-08-22 [1] Github (vqv/ggbiplot@7325e88)
##  ggforce                0.3.3      2021-03-05 [1] CRAN (R 4.1.0)
##  ggplot2              * 3.3.5      2021-06-25 [1] CRAN (R 4.1.0)
##  ggpubr               * 0.4.0      2020-06-27 [1] CRAN (R 4.1.0)
##  ggraph                 2.0.5      2021-02-23 [1] CRAN (R 4.1.0)
##  ggrastr                0.2.3      2021-03-01 [1] CRAN (R 4.1.0)
##  ggrepel              * 0.9.1      2021-01-15 [1] CRAN (R 4.1.0)
##  ggridges             * 0.5.3      2021-01-08 [1] CRAN (R 4.1.0)
##  ggsignif               0.6.2      2021-06-14 [1] CRAN (R 4.1.0)
##  ggtree                 3.0.3      2021-08-12 [1] Bioconductor
##  ggupset              * 0.3.0      2020-05-05 [1] CRAN (R 4.1.0)
##  glmpca               * 0.2.0      2020-07-18 [1] CRAN (R 4.1.0)
##  glue                   1.4.2      2020-08-27 [1] CRAN (R 4.1.0)
##  GO.db                * 3.13.0     2021-08-17 [1] Bioconductor
##  GOSemSim               2.18.1     2021-07-29 [1] Bioconductor
##  GOstats              * 2.58.0     2021-05-19 [1] Bioconductor
##  graph                * 1.70.0     2021-05-19 [1] Bioconductor
##  graphlayouts           0.7.1      2020-10-26 [1] CRAN (R 4.1.0)
##  gridExtra            * 2.3        2017-09-09 [1] CRAN (R 4.1.0)
##  GSEABase               1.54.0     2021-05-19 [1] Bioconductor
##  gtable                 0.3.0      2019-03-25 [1] CRAN (R 4.1.0)
##  haven                  2.4.3      2021-08-04 [1] CRAN (R 4.1.0)
##  hexbin                 1.28.2     2021-01-08 [1] CRAN (R 4.1.0)
##  highr                  0.9        2021-04-16 [1] CRAN (R 4.1.0)
##  hms                    1.1.0      2021-05-17 [1] CRAN (R 4.1.0)
##  htmltools              0.5.1.1    2021-01-22 [1] CRAN (R 4.1.0)
##  htmlwidgets            1.5.3      2020-12-10 [1] CRAN (R 4.1.0)
##  httr                   1.4.2      2020-07-20 [1] CRAN (R 4.1.0)
##  igraph                 1.2.6      2020-10-06 [1] CRAN (R 4.1.0)
##  IRanges              * 2.26.0     2021-05-19 [1] Bioconductor
##  iterators              1.0.13     2020-10-15 [1] CRAN (R 4.1.0)
##  jsonlite               1.7.2      2020-12-09 [1] CRAN (R 4.1.0)
##  KEGGREST               1.32.0     2021-05-19 [1] Bioconductor
##  KernSmooth             2.23-20    2021-05-03 [1] CRAN (R 4.1.0)
##  knitr                  1.33       2021-04-24 [1] CRAN (R 4.1.0)
##  labeling               0.4.2      2020-10-20 [1] CRAN (R 4.1.0)
##  lattice                0.20-44    2021-05-02 [1] CRAN (R 4.1.0)
##  lazyeval               0.2.2      2019-03-15 [1] CRAN (R 4.1.0)
##  lifecycle              1.0.0      2021-02-15 [1] CRAN (R 4.1.0)
##  limma                  3.48.3     2021-08-10 [1] Bioconductor
##  locfit                 1.5-9.4    2020-03-25 [1] CRAN (R 4.1.0)
##  M3C                  * 1.14.0     2021-05-19 [1] Bioconductor
##  magrittr               2.0.1      2020-11-17 [1] CRAN (R 4.1.0)
##  maps                   3.3.0      2018-04-03 [1] CRAN (R 4.1.0)
##  MASS                   7.3-54     2021-05-03 [1] CRAN (R 4.1.0)
##  Matrix               * 1.3-4      2021-06-01 [1] CRAN (R 4.1.0)
##  matrixcalc             1.0-5      2021-07-28 [1] CRAN (R 4.1.0)
##  MatrixGenerics       * 1.4.2      2021-08-08 [1] Bioconductor
##  matrixStats          * 0.60.0     2021-07-26 [1] CRAN (R 4.1.0)
##  memoise                2.0.0      2021-01-26 [1] CRAN (R 4.1.0)
##  munsell                0.5.0      2018-06-12 [1] CRAN (R 4.1.0)
##  mvtnorm                1.1-2      2021-06-07 [1] CRAN (R 4.1.0)
##  nlme                   3.1-152    2021-02-04 [1] CRAN (R 4.1.0)
##  numDeriv               2016.8-1.1 2019-06-06 [1] CRAN (R 4.1.0)
##  openssl                1.4.4      2021-04-30 [1] CRAN (R 4.1.0)
##  openxlsx               4.2.4      2021-06-16 [1] CRAN (R 4.1.0)
##  org.Hs.eg.db         * 3.13.0     2021-08-17 [1] Bioconductor
##  patchwork              1.1.1      2020-12-17 [1] CRAN (R 4.1.0)
##  pheatmap             * 1.0.12     2019-01-04 [1] CRAN (R 4.1.0)
##  pillar                 1.6.2      2021-07-29 [1] CRAN (R 4.1.0)
##  pkgbuild               1.2.0      2020-12-15 [1] CRAN (R 4.1.0)
##  pkgconfig              2.0.3      2019-09-22 [1] CRAN (R 4.1.0)
##  pkgload                1.2.1      2021-04-06 [1] CRAN (R 4.1.0)
##  plyr                 * 1.8.6      2020-03-03 [1] CRAN (R 4.1.0)
##  png                    0.1-7      2013-12-03 [1] CRAN (R 4.1.0)
##  PoiClaClu            * 1.0.2.1    2019-01-04 [1] CRAN (R 4.1.0)
##  polyclip               1.10-0     2019-03-14 [1] CRAN (R 4.1.0)
##  preprocessCore         1.54.0     2021-05-19 [1] Bioconductor
##  prettyunits            1.1.1      2020-01-24 [1] CRAN (R 4.1.0)
##  processx               3.5.2      2021-04-30 [1] CRAN (R 4.1.0)
##  proj4                  1.0-10.1   2021-01-26 [1] CRAN (R 4.1.0)
##  ps                     1.6.0      2021-02-28 [1] CRAN (R 4.1.0)
##  purrr                  0.3.4      2020-04-17 [1] CRAN (R 4.1.0)
##  qvalue                 2.24.0     2021-05-19 [1] Bioconductor
##  R6                     2.5.1      2021-08-19 [1] CRAN (R 4.1.0)
##  RBGL                   1.68.0     2021-05-19 [1] Bioconductor
##  RColorBrewer           1.1-2      2014-12-07 [1] CRAN (R 4.1.0)
##  Rcpp                   1.0.7      2021-07-07 [1] CRAN (R 4.1.0)
##  RCurl                  1.98-1.3   2021-03-16 [1] CRAN (R 4.1.0)
##  readr                  2.0.1      2021-08-10 [1] CRAN (R 4.1.0)
##  readxl                 1.3.1      2019-03-13 [1] CRAN (R 4.1.0)
##  remotes                2.4.0      2021-06-02 [1] CRAN (R 4.1.0)
##  reshape2               1.4.4      2020-04-09 [1] CRAN (R 4.1.0)
##  reticulate             1.20       2021-05-03 [1] CRAN (R 4.1.0)
##  rgl                  * 0.107.10   2021-07-22 [1] CRAN (R 4.1.0)
##  Rgraphviz              2.36.0     2021-05-19 [1] Bioconductor
##  rio                    0.5.27     2021-06-21 [1] CRAN (R 4.1.0)
##  rlang                  0.4.11     2021-04-30 [1] CRAN (R 4.1.0)
##  rmarkdown              2.10       2021-08-06 [1] CRAN (R 4.1.0)
##  rprojroot              2.0.2      2020-11-15 [1] CRAN (R 4.1.0)
##  RSpectra               0.16-0     2019-12-01 [1] CRAN (R 4.1.0)
##  RSQLite                2.2.7      2021-04-22 [1] CRAN (R 4.1.0)
##  rstatix                0.7.0      2021-02-13 [1] CRAN (R 4.1.0)
##  rstudioapi             0.13       2020-11-12 [1] CRAN (R 4.1.0)
##  Rtsne                * 0.15       2018-11-10 [1] CRAN (R 4.1.0)
##  Rttf2pt1                 1.3.9      2021-07-22 [1] CRAN (R 4.1.0)
##  rvcheck                  0.1.8      2020-03-01 [1] CRAN (R 4.1.0)
##  S4Vectors              * 0.30.0     2021-05-19 [1] Bioconductor
##  scales                 * 1.1.1      2020-05-11 [1] CRAN (R 4.1.0)
##  scatterpie               0.1.6      2021-04-23 [1] CRAN (R 4.1.0)
##  sessioninfo              1.1.1      2018-11-05 [1] CRAN (R 4.1.0)
##  shadowtext               0.0.8      2021-04-23 [1] CRAN (R 4.1.0)
##  snow                     0.4-3      2018-09-14 [1] CRAN (R 4.1.0)
##  stringi                  1.7.3      2021-07-16 [1] CRAN (R 4.1.0)
##  stringr                  1.4.0      2019-02-10 [1] CRAN (R 4.1.0)
##  SummarizedExperiment   * 1.22.0     2021-05-19 [1] Bioconductor
##  survival                 3.2-12     2021-08-13 [1] CRAN (R 4.1.0)
##  testthat                 3.0.4      2021-07-01 [1] CRAN (R 4.1.0)
##  tibble                   3.1.3      2021-07-23 [1] CRAN (R 4.1.0)
##  tidygraph                1.2.0      2020-05-12 [1] CRAN (R 4.1.0)
##  tidyr                    1.1.3      2021-03-03 [1] CRAN (R 4.1.0)
##  tidyselect               1.1.1      2021-04-30 [1] CRAN (R 4.1.0)
##  tidytree                 0.3.4      2021-05-22 [1] CRAN (R 4.1.0)
##  treeio                   1.16.1     2021-05-23 [1] Bioconductor
##  tweenr                   1.0.2      2021-03-23 [1] CRAN (R 4.1.0)
##  tzdb                     0.1.2      2021-07-20 [1] CRAN (R 4.1.0)
##  umap                     0.2.7.0    2020-11-04 [1] CRAN (R 4.1.0)
##  usethis                * 2.0.1      2021-02-10 [1] CRAN (R 4.1.0)
##  utf8                     1.2.2      2021-07-24 [1] CRAN (R 4.1.0)
##  vctrs                    0.3.8      2021-04-29 [1] CRAN (R 4.1.0)
##  vipor                    0.4.5      2017-03-22 [1] CRAN (R 4.1.0)
##  viridis                  0.6.1      2021-05-11 [1] CRAN (R 4.1.0)
##  viridisLite              0.4.0      2021-04-13 [1] CRAN (R 4.1.0)
##  vroom                    1.5.4      2021-08-05 [1] CRAN (R 4.1.0)
##  vsn                    * 3.60.0     2021-05-19 [1] Bioconductor
##  withr                    2.4.2      2021-04-18 [1] CRAN (R 4.1.0)
##  xfun                     0.25       2021-08-06 [1] CRAN (R 4.1.0)
##  XML                    * 3.99-0.6   2021-03-16 [1] CRAN (R 4.1.0)
##  xml2                     1.3.2      2020-04-23 [1] CRAN (R 4.1.0)
##  xtable                   1.8-4      2019-04-21 [1] CRAN (R 4.1.0)
##  XVector                  0.32.0     2021-05-19 [1] Bioconductor
##  yaml                     2.2.1      2020-02-01 [1] CRAN (R 4.1.0)
##  zip                      2.2.0      2021-05-31 [1] CRAN (R 4.1.0)
##  zlibbioc                 1.38.0     2021-05-19 [1] Bioconductor
##
## [1] C:/Program Files/R/R-4.1.0/library
```