Cox7a1 7dpi RNASeq Garcia-Poyatos et al 2024

Code

AUTHOR
Prateek Arora

PUBLISHED
January 24, 2024

1 Clean memory and set working directory

▼ Code

```
rm(list = ls()) # clear workspace
gc() # clear memory
setwd("Carolina_data/RNASeq") # set working directory
getwd() # check working directory
```

A matrix: 2 × 6 of type dbl

	used	(Mb)	gc trigger	(Mb)	max used	(Mb)
Ncells	631329	33.8	1411245	75.4	985811	52.7
Vcells	1169048	9.0	8388608	64.0	1815676	13.9

'/mnt/Data_8TB/Carolina_data/Cell_paper/RNASeq'

2 load libraries

▼ Code

```
library(tidyverse) # load tidyverse for data manipulation and plotting
library(biomaRt) # load biomaRt for ensembl
library(DESeq2) # load DESeq2 for differential expression analysis
library(clusterProfiler) # load clusterProfiler for GO analysis
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.2     ✔ readr     2.1.4
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ ggplot2   3.4.2     ✔ tibble    3.2.1
✔ lubridate 1.9.2     ✔ tidyr     1.3.0
✔ purrr     1.0.1
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to
become errors
Loading required package: S4Vectors

Loading required package: stats4

Loading required package: BiocGenerics
```

```
Attaching package: ‘BiocGenerics’

The following objects are masked from ‘package:lubridate’:

    intersect, setdiff, union

The following objects are masked from ‘package:dplyr’:

    combine, intersect, setdiff, union

The following objects are masked from ‘package:stats’:

    IQR, mad, sd, var, xtabs

The following objects are masked from ‘package:base’:

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


Attaching package: ‘S4Vectors’

The following objects are masked from ‘package:lubridate’:

    second, second<-

The following objects are masked from ‘package:dplyr’:

    first, rename

The following object is masked from ‘package:tidyr’:

    expand

The following object is masked from ‘package:utils’:

    findMatches
```

```
The following objects are masked from ‘package:base’:

    expand.grid, I, unname


Loading required package: IRanges


Attaching package: ‘IRanges’


The following object is masked from ‘package:lubridate’:

    %within%


The following objects are masked from ‘package:dplyr’:

    collapse, desc, slice


The following object is masked from ‘package:purrr’:

    reduce


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 object is masked from ‘package:dplyr’:

    count



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


Loading required package: Biobase

Welcome to Bioconductor

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



Attaching package: ‘Biobase’


The following object is masked from ‘package:MatrixGenerics’:

    rowMedians


The following objects are masked from ‘package:matrixStats’:

    anyMissing, rowMedians




clusterProfiler v4.8.1  For help: https://yulab-smu.top/biomedical-knowledge-mining-
book/

If you use clusterProfiler in published research, please cite:
T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X
Bo, and G Yu. clusterProfiler 4.0: A universal enrichment tool for interpreting omics
data. The Innovation. 2021, 2(3):100141


Attaching package: ‘clusterProfiler’


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:biomaRt’:

    select


The following object is masked from ‘package:purrr’:

    simplify


The following object is masked from ‘package:stats’:

    filter
```

3 Load data

▼ Code

```
counts_file <- read.table("09_feature_counts/feature_counts.txt", header = T,
        row.names = 1) # read in counts file
colnames(counts_file) <- gsub("X08_star_alignment_2nd_pass.", "",
        colnames(counts_file)) # clean column names
colnames(counts_file) <-
        gsub("__NadiaMercader_RNA_Seq_Directional_S._R1_001.Aligned.sortedByCoord.out.
        "", colnames(counts_file)) # clean column names
colnames(counts_file) # check column names
counts_file <- counts_file %>% dplyr::select(-
        c('Chr','Start','End','Strand','Length')) # remove unnecessary columns
head(counts_file) # check file
```

'Chr' · 'Start' · 'End' · 'Strand' · 'Length' · 'KO_10' · 'KO_7' · 'KO_8' · 'KO_9' · 'WT_1' · 'WT_2' · 'WT_3' · 'WT_5'

A data.frame: 6 × 8

	KO_10	KO_7	KO_8	KO_9	WT_1	WT_2	WT_3	WT_5
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
ENSDARG00000103202	0	0	0	4	0	0	0	0
ENSDARG00000009657	1327	811	919	759	1396	1325	1570	1258
ENSDARG00000096472	0	0	0	2	0	0	0	0
ENSDARG00000096156	6	4	8	5	5	6	5	18
ENSDARG00000076160	2	0	6	0	9	7	0	0

	KO_10	KO_7	KO_8	KO_9	WT_1	WT_2	WT_3	WT_5
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
ENSDARG00000117163	11	10	3	10	23	11	11	16

▼ Code

```
# make metadata file

metadata <- data.frame(Sample_Name = colnames(counts_file), row.names =
        colnames(counts_file)) # make metadata file
rownames(metadata) <- metadata$Sample_Name # set rownames
metadata$condition <- c( rep("WT", 4),rep("Cox7a_KO", 4)) # add condition column
metadata$condition <- factor(metadata$condition, levels = c("Cox7a_KO", "WT")) # set
        factor levels
metadata <- metadata %>% dplyr::arrange(condition) # arrange by condition
metadata
```

A data.frame: 8 × 2

	Sample_Name	condition
	<chr>	<fct>
KO_10	KO_10	Cox7a_KO
KO_7	KO_7	Cox7a_KO
KO_8	KO_8	Cox7a_KO
KO_9	KO_9	Cox7a_KO
WT_1	WT_1	WT
WT_2	WT_2	WT
WT_3	WT_3	WT
WT_5	WT_5	WT

▼ Code

```
counts_file <- counts_file[,metadata$Sample_Name] # reorder columns
head(counts_file) # check file
identical(colnames(counts_file), rownames(metadata)) # check if column names and
        rownames are identical
```

A data.frame: 6 × 8

	KO_10	KO_7	KO_8	KO_9	WT_1	WT_2	WT_3	WT_5
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
ENSDARG00000103202	0	0	0	4	0	0	0	0
ENSDARG00000009657	1327	811	919	759	1396	1325	1570	1258

	KO_10	KO_7	KO_8	KO_9	WT_1	WT_2	WT_3	WT_5
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
ENSDARG00000096472	0	0	0	2	0	0	0	0
ENSDARG00000096156	6	4	8	5	5	6	5	18
ENSDARG00000076160	2	0	6	0	9	7	0	0
ENSDARG00000117163	11	10	3	10	23	11	11	16

TRUE

▼ Code

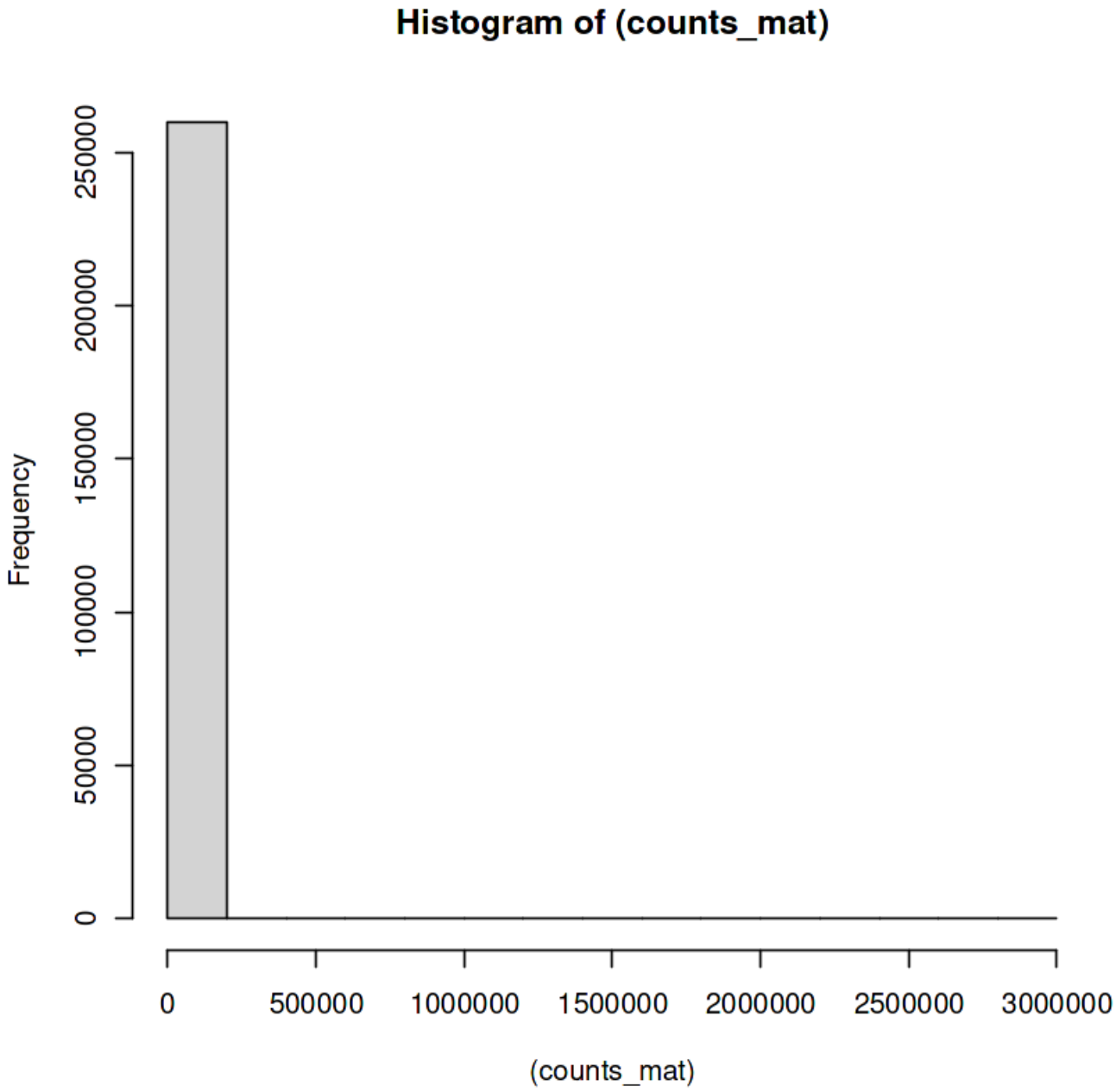
```
counts_mat <- as.matrix(counts_file) # convert to matrix for DESeq2
head(counts_mat)
```

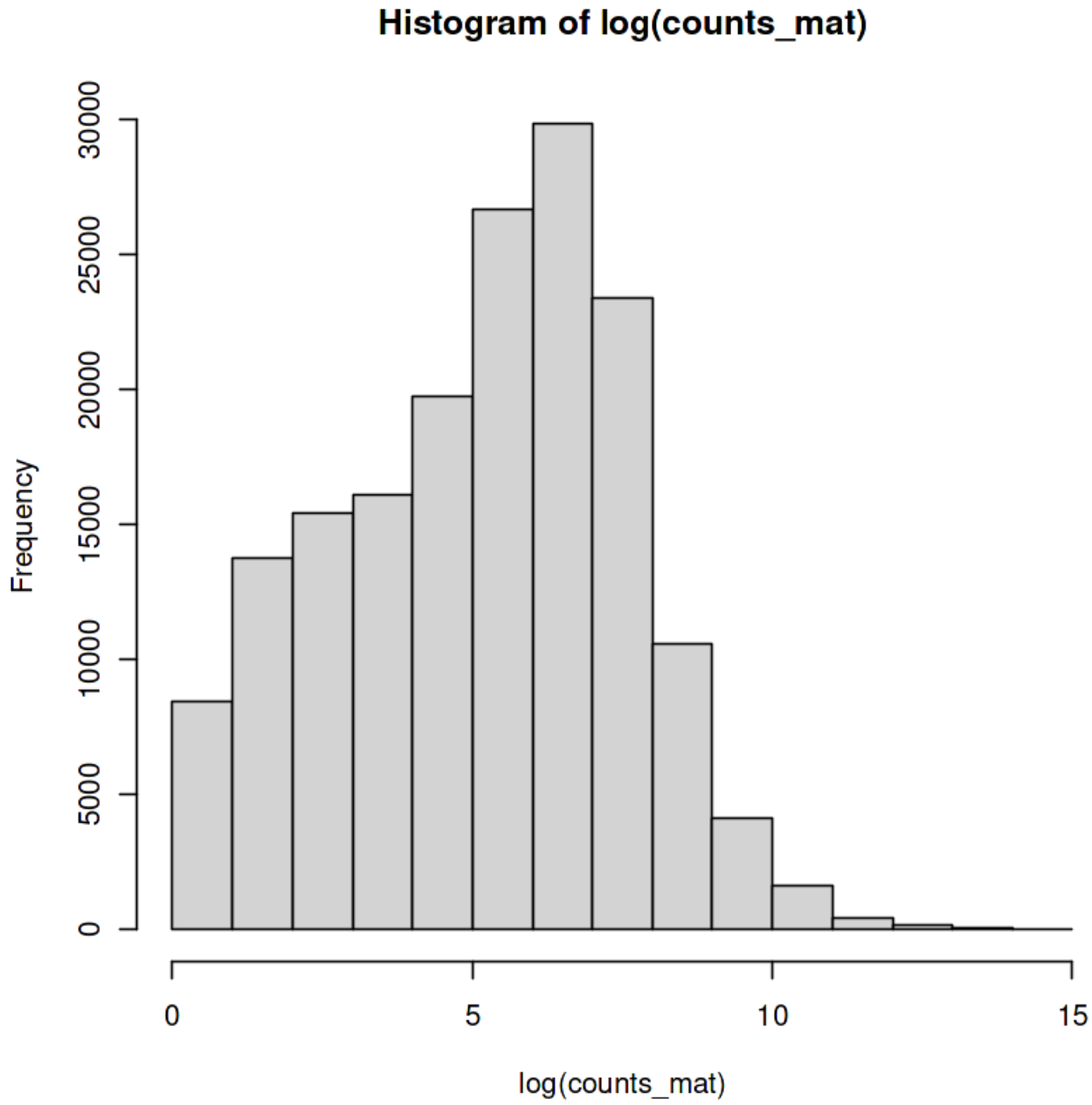
A matrix: 6 × 8 of type int

	KO_10	KO_7	KO_8	KO_9	WT_1	WT_2	WT_3	WT_5
ENSDARG00000103202	0	0	0	4	0	0	0	0
ENSDARG00000009657	1327	811	919	759	1396	1325	1570	1258
ENSDARG00000096472	0	0	0	2	0	0	0	0
ENSDARG00000096156	6	4	8	5	5	6	5	18
ENSDARG00000076160	2	0	6	0	9	7	0	0
ENSDARG00000117163	11	10	3	10	23	11	11	16

▼ Code

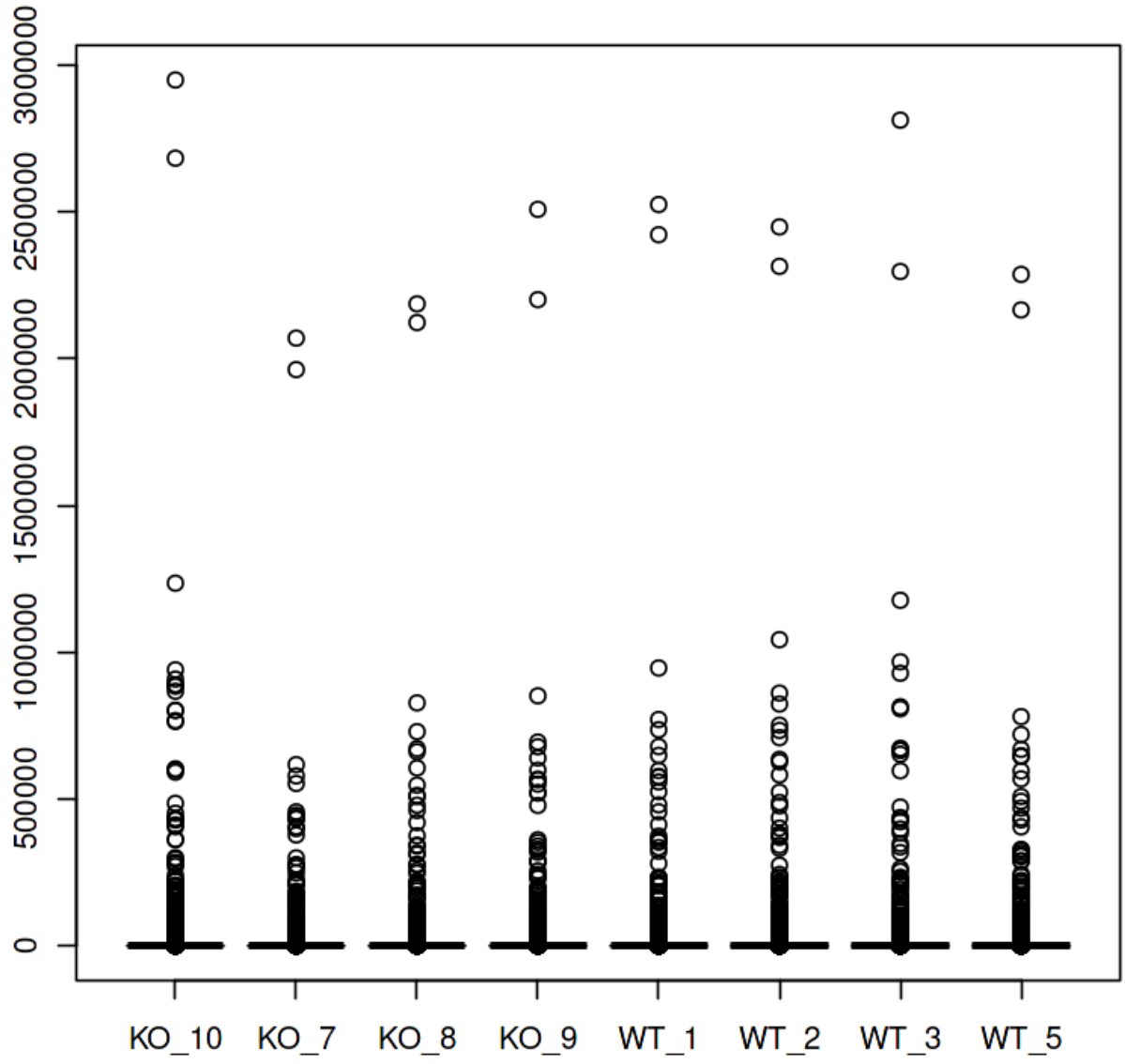
```
hist((counts_mat)) # check distribution of counts
hist(log(counts_mat)) # check distribution of log counts
```

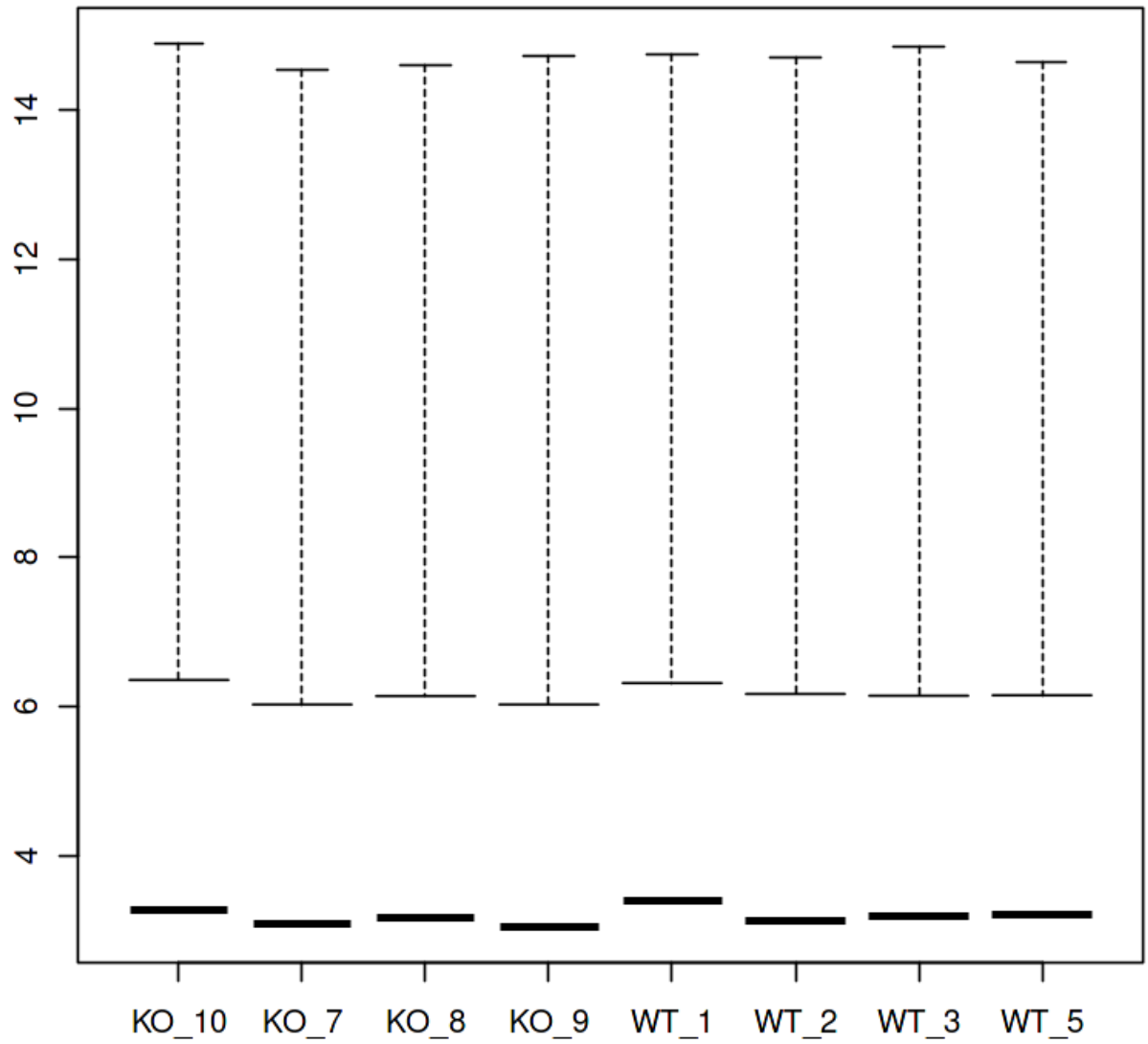

Histogram of (counts_mat)
Frequency
0
50000
100000
150000
200000
250000
0
500000
1000000
1500000
2000000
2500000
3000000
(counts_mat)



▼ Code

```
boxplot(counts_mat) # check distribution of counts per sample
boxplot(log(counts_mat)) # check distribution of log counts per sample
```


0
500000
1000000
1500000
2000000
2500000
3000000
KO_10
KO_7
KO_8
KO_9
WT_1
WT_2
WT_3
WT_5



▼ Code

```
sum(is.na(counts_mat)) # check for missing values
```

0

4 Rum PCA to check for trend and outliers

▼ Code

```
# Make a PCA function
PCA <- function(mat, color_pca = "", shape_pca = "", label_pca = "", save_plot =
        "no", name_of_plot = "PCA", comp1 = 1, comp2 = 2, pdf_height=12,
        pdf_width=12) {
    dt <- mat
    pca_dt <- prcomp(t(dt))
    cat("PCA running...\n")
```

```
    percentVar_dt <- pca_dt$sdev^2 / sum(pca_dt$sdev^2)
    cat("Percents calculated...\n")
    dt_f <- data.frame(PC1 = pca_dt$x[, comp1],
                       PC2 = pca_dt$x[, comp2],
                       color_pca = color_pca,
                       shape_pca = shape_pca,
                       label_pca = label_pca)
    cat("Data frame built...\n")
    cat("Plotting...\n")
    require(ggplot2)
    require(ggrepel)
    if (save_plot == "no") {
        pca_p <- ggplot(data = dt_f, aes_string(x = paste0("PC1"),
                                                y = paste0("PC2"),
                                                color = "color_pca",
                                                shape = "shape_pca", label =
        "label_pca")) +
                geom_point(size = 3) +
                geom_text_repel(size = 3, max.overlaps = 50,
                                box.padding = 1.5, point.padding = 0.5, force = 50) +
                xlab(paste0("PC", comp1, ": ", round(percentVar_dt[comp1] * 100), "%
        variance")) +
                ylab(paste0("PC", comp2, ": ", round(percentVar_dt[comp2] * 100), "%
        variance")) +
            NULL
    }
    if (save_plot == "yes") {
        pdf(paste0(name_of_plot, ".pdf"), width = pdf_width, height = pdf_height)
        cat("Saving plot as: ", paste0(name_of_plot, "...\n"))
        pca_p <- ggplot(data = dt_f, aes_string(x = paste0("PC", comp1),
                                                y = paste0("PC", comp2),
                                                color = "color_pca",
                                                shape = "shape_pca", label =
        "label_pca")) +
                geom_text_repel(size = 3, max.overlaps = 50,
                                box.padding = 1.5,
                                point.padding = 0.5, force = 50) +
                geom_point(size = 3) +
                xlab(paste0("PC", comp1, ": ", round(percentVar_dt[comp1] * 100), "%
        variance")) +
                ylab(paste0("PC", comp2, ": ", round(percentVar_dt[comp2] * 100), "%
        variance")) +
            NULL
        print(pca_p)
        dev.off()
    }
    cat("Done")
    print(pca_p)
}
```

▼ Code

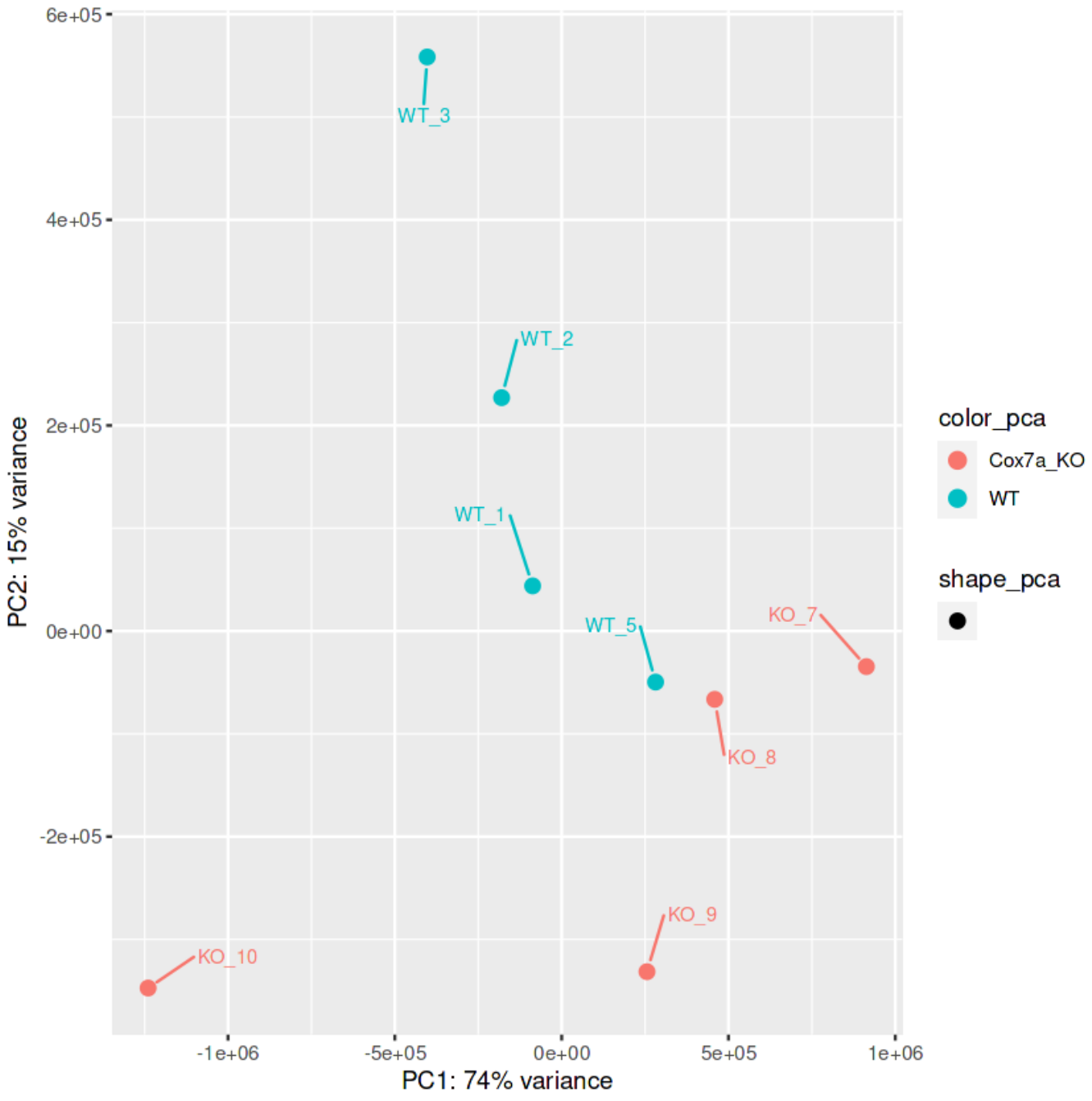
```
# create a Results directory to store results
dir.create("Results", recursive = T)
```

```
Warning message in dir.create("Results", recursive = T):
“'Results' already exists”
```

▼ Code

```
# run PCA on counts matrix
 PCA(counts_mat, color_pca = metadata$condition, shape_pca = "", label_pca =
        metadata$Sample_Name, save_plot = "yes", name_of_plot = "Results/PCA", comp1
        = 1, comp2 = 2, pdf_height=12, pdf_width=12)
```

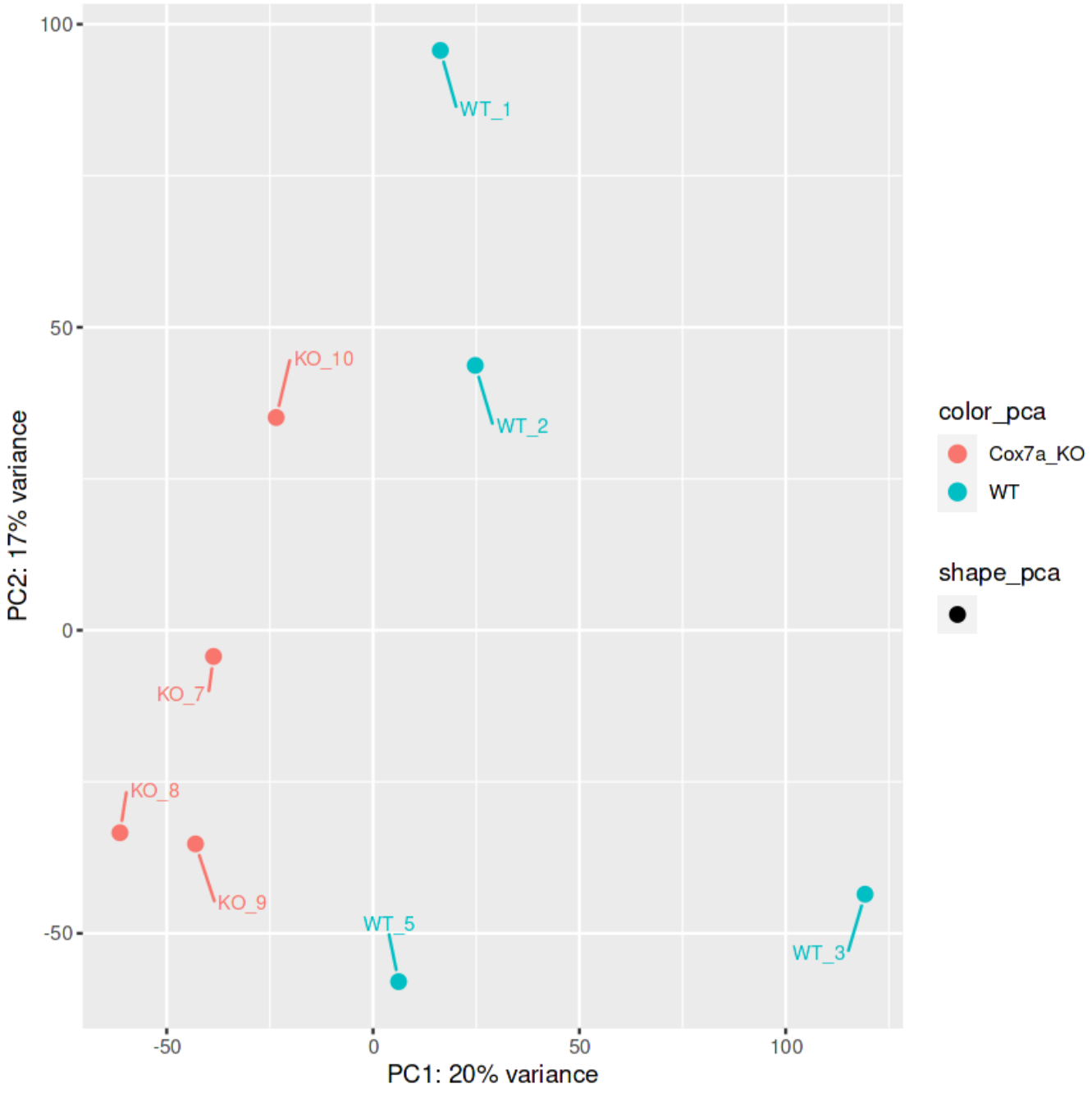
```
PCA running...
Percents calculated...
Data frame built...
Plotting...
Saving plot as:  Results/PCA...
Done
```



▼ Code

```
# Run PCA on log counts matrix
PCA(log2(counts_mat+1), color_pca = metadata$condition, shape_pca = "", label_pca =
        metadata$Sample_Name, save_plot = "yes", name_of_plot = "Results/PCA_log",
        comp1 = 1, comp2 = 2, pdf_height=12, pdf_width=12)
```

```
PCA running...
Percents calculated...
Data frame built...
Plotting...
Saving plot as:  Results/PCA_log...
Done
```



5 Run Desq2 analysis

▼ Code

```
dds <- DESeqDataSetFromMatrix(countData=counts_mat,
                              colData=metadata,
                              design=~condition) # make DESeq2 object
```

▼ Code

```
keep <- rowSums(counts(dds)) >= 10 # keep genes with at least 10 counts
dds_filtered <- dds[keep,] # filter based on keep
dds
dds_filtered
```

```
class: DESeqDataSet
dim: 32520 8
metadata(1): version
assays(1): counts
rownames(32520): ENSDARG00000103202 ENSDARG00000009657 ...
  ENSDARG00000101098 ENSDARG00000103574
rowData names(0):
colnames(8): KO_10 KO_7 ... WT_3 WT_5
colData names(2): Sample_Name condition

class: DESeqDataSet
dim: 22343 8
metadata(1): version
assays(1): counts
rownames(22343): ENSDARG00000009657 ENSDARG00000096156 ...
  ENSDARG00000104659 ENSDARG00000103574
rowData names(0):
colnames(8): KO_10 KO_7 ... WT_3 WT_5
colData names(2): Sample_Name condition
```

▼ Code

```
dds_filtered <- DESeq(dds_filtered, parallel = T) # run DESeq2
```

```
estimating size factors

estimating dispersions

gene-wise dispersion estimates: 10 workers

mean-dispersion relationship

final dispersion estimates, fitting model and testing: 10 workers
```

▼ Code

```
resultsNames(dds_filtered) # check results names
```

'Intercept' · 'condition_WT_vs_Cox7a_KO'

▼ Code

```r
results_WT_KO <- lfcShrink(dds_filtered, contrast = c("condition","Cox7a_KO","WT"),
                           parallel=TRUE, type = "ashr") # shrink log fold changes
        using ashr method for Cox7a_KO vs WT
print("lfcshrinakge done...")
results_WT_KO_df <- as.data.frame(results_WT_KO) # convert to data frame
head(results_WT_KO_df)
plotMA(results_WT_KO, ylim=c(-10,10)) # plot MA plot
results_WT_KO_df <- results_WT_KO_df %>% dplyr::arrange(padj) # arrange by padj
head(results_WT_KO_df)
```

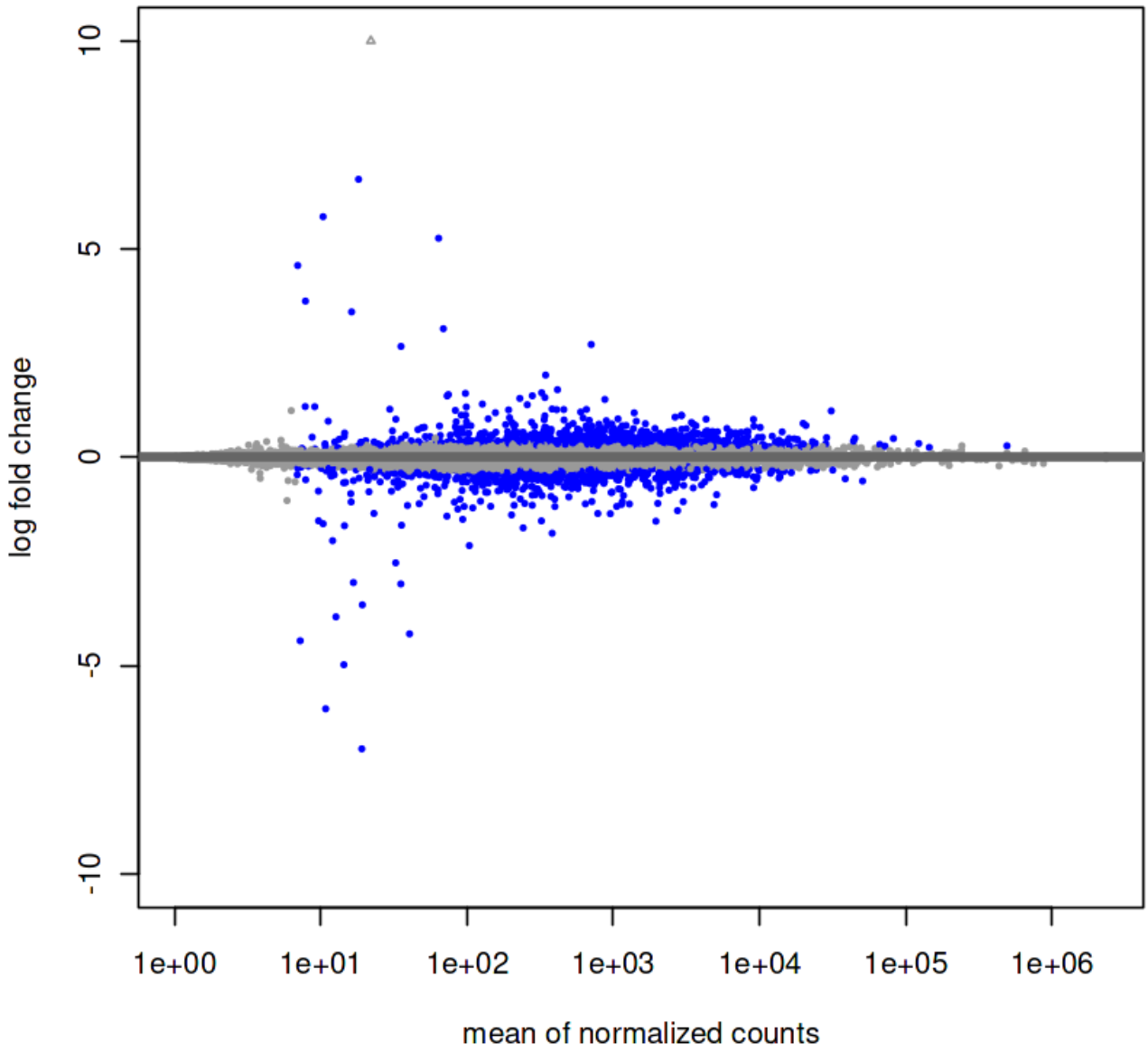
```
using 'ashr' for LFC shrinkage. If used in published research, please cite:
    Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.
    https://doi.org/10.1093/biostatistics/kxw041

[1] "lfcshrinakge done..."
```

A data.frame: 6 × 5

	baseMean	log2FoldChange	lfcSE	pvalue	padj
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
ENSDARG00000009657	1148.306489	-0.36641431	0.1427850	0.0006226866	0.0174343
ENSDARG00000096156	7.096952	-0.03537140	0.2289743	0.5484592047	0.8261124
ENSDARG00000076160	2.820730	-0.01481899	0.2532079	0.6445563313	NA
ENSDARG00000117163	11.589063	-0.07815191	0.2323184	0.2686859935	0.6245806
ENSDARG00000096187	5.993731	0.03185551	0.2452809	0.4728926837	NA
ENSDARG00000076014	275.688504	0.04408028	0.1413125	0.6193406672	0.8594960

A data.frame: 6 × 5

	baseMean	log2FoldChange	lfcSE	pvalue	padj
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
ENSDARG00000089791	958.0652	-1.347831	0.1313111	7.575781e-27	1.427504e-22
ENSDARG00000035519	1963.6138	-1.527732	0.1533885	2.001634e-25	1.885839e-21
ENSDARG00000063307	879.0429	1.395107	0.1581055	7.882223e-21	4.950824e-17
ENSDARG00000025788	400.3032	-1.175105	0.1511315	1.770101e-17	8.338505e-14
ENSDARG00000068374	2936.7863	1.009402	0.1341592	9.013761e-17	3.396926e-13
ENSDARG00000055723	325.0561	1.555549	0.2051782	1.627645e-16	5.111620e-13



▼ Code

```
results_WT_KO_df %>% filter(padj < 0.05) %>% dim  # check number of DEGs with padj < 0.05
```

1070 · 5

▼ Code

```
gtf <- rtracklayer::import('/mnt/Data_8TB/Genomes/v109/Danio_rerio.GRCz11.109.gtf') # import gtf file for zebrafish to get gene names
gtf_df <- as.data.frame(gtf)
head(gtf_df)
gtf_df %>% colnames
```

	seqnames	start	end	width	strand	source	type	score	phase	gene_id
	<fct>	<int>	<int>	<int>	<fct>	<fct>	<fct>	<dbl>	<int>	<chr>
1	4	30402837	30403763	927	+	havana	gene	NA	NA	ENSDARG
2	4	30402837	30403763	927	+	havana	transcript	NA	NA	ENSDARG
3	4	30402837	30402893	57	+	havana	exon	NA	NA	ENSDARG
4	4	30403203	30403350	148	+	havana	exon	NA	NA	ENSDARG
5	4	30403546	30403763	218	+	havana	exon	NA	NA	ENSDARG
6	4	1722899	1730920	8022	+	ensembl_havana	gene	NA	NA	ENSDARG

'seqnames' · 'start' · 'end' · 'width' · 'strand' · 'source' · 'type' · 'score' · 'phase' · 'gene_id' · 'gene_version' · 'gene_name' · 'gene_source' · 'gene_biotype' · 'transcript_id' · 'transcript_version' · 'transcript_name' · 'transcript_source' · 'transcript_biotype' · 'tag' · 'exon_number' · 'exon_id' · 'exon_version' · 'protein_id' · 'protein_version'

▼ Code

```
cox_genes <- gtf_df %>% filter(grepl("cox", gene_name)) %>% filter(type=="gene") %>%
        dplyr::select(gene_id,
gene_name) # get cox genes
cox_genes
```

A data.frame: 30 × 2

gene_id	gene_name
<chr>	<chr>
ENSDARG00000038865	acox3
ENSDARG00000104537	cox7c
ENSDARG00000054588	cox6a2
ENSDARG00000038577	cox6c
ENSDARG00000014727	acox1
ENSDARG00000069920	cox17
ENSDARG00000037860	cox6b2
ENSDARG00000022438	cox6a1
ENSDARG00000053217	cox7a2a
ENSDARG00000102463	cox18
ENSDARG00000039136	cox16

gene_id	gene_name
<chr>	<chr>
ENSDARG00000020149	acoxl
ENSDARG00000099997	cox20
ENSDARG00000068738	cox5b2
ENSDARG00000075933	cox15
ENSDARG00000032970	cox4i1
ENSDARG00000099663	cox5ab
ENSDARG00000034309	cox10
ENSDARG00000061004	cox11
ENSDARG00000054907	cox7a2l
ENSDARG00000069464	cox7a1
ENSDARG00000022509	cox4i2
ENSDARG00000095273	cox8a
ENSDARG00000115557	cox7b
ENSDARG00000012388	cox4i1l
ENSDARG00000045230	cox6b1
ENSDARG00000063882	cox19
ENSDARG00000092124	cox14
ENSDARG00000097209	cox8b
ENSDARG00000088383	cox5aa

▼ Code

```
gtf_df <- gtf_df %>% dplyr::select(gene_id, gene_name) # select gene_id and gene_name
        columns
gtf_df  <- gtf_df %>% dplyr::distinct() # remove duplicates
gtf_df$gene_id %>% duplicated() %>% sum() # check for duplicates
head(gtf_df)  # check file
```

0

A data.frame: 6 × 2

	gene_id	gene_name
	<chr>	<chr>
1	ENSDARG00000103202	CR383668.1
2	ENSDARG00000009657	fgfr1op2
3	ENSDARG00000096472	AL845295.2

	gene_id	gene_name
	<chr>	<chr>
4	ENSDARG00000096156	si:dkey-21h14.12
5	ENSDARG00000076160	si:dkey-285e18.2
6	ENSDARG00000117163	znf1114

▼ Code

```
# convert gene_id to gene_name
symbol_to_ensmbl_danio <- function(x) {
    require(biomaRt) # load biomaRt package
    zf_mart <- useMart(biomart = "ensembl", dataset = "drerio_gene_ensembl") # use
        the ensembl mart
    results <- getBM(attributes = c("ensembl_gene_id", "zfin_id_symbol",
        "description"), # get ensembl gene id, zfin id symbol and description
                     filters = "ensembl_gene_id", # filter by ensembl gene id
                     values = x,
                     mart = zf_mart)
    results
}
description <- symbol_to_ensmbl_danio(gtf_df$gene_id)
description <- as.data.frame(description)
head(description)
```

A data.frame: 6 × 3

	ensembl_gene_id	zfin_id_symbol	description
	<chr>	<chr>	<chr>
1	ENSDARG00000000018	nrf1	nuclear respiratory factor 1 [Source:NCBI gene;Acc:64604]
2	ENSDARG00000000019	ube2h	ubiquitin-conjugating enzyme E2H (UBC8 homolog, yeast) [Source:ZFIN;Acc:ZDB-GENE-030616-67]
3	ENSDARG00000000423	si:ch73-314g15.3	si:ch73-314g15.3 [Source:ZFIN;Acc:ZDB-GENE-030616-19]
4	ENSDARG00000000442	slc39a13	solute carrier family 39 member 13 [Source:NCBI gene;Acc:368686]
5	ENSDARG00000000460	nitr2b	novel immune-type receptor 2b [Source:ZFIN;Acc:ZDB-GENE-001106-6]
6	ENSDARG00000000767	spi1b	Spi-1 proto-oncogene b [Source:ZFIN;Acc:ZDB-GENE-980526-164]

▼ Code

```r
results_WT_KO_df$ensembl_gene_id <- rownames(results_WT_KO_df) # add ensembl gene id
        column
results_WT_KO_df %>% head # check file
# convert ensemble to symbol by merging with gtf_df
results_WT_KO_df_gene <- merge(results_WT_KO_df, gtf_df, by.x = "ensembl_gene_id",
        by.y = "gene_id", all.x = T)
results_WT_KO_df_gene %>% head()
results_WT_KO_df_gene$gene_name %>% duplicated() %>% sum()
results_WT_KO_df_gene$gene_name %>% is.na() %>% sum()
results_WT_KO_df_gene <- results_WT_KO_df_gene %>% dplyr::arrange(padj)
write.csv(results_WT_KO_df_gene, "Results/DEGs_KO_vs_WT.csv")
```

A data.frame: 6 × 6

	baseMean	log2FoldChange	lfcSE	pvalue	padj	ensembl_g
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
ENSDARG00000089791	958.0652	-1.347831	0.1313111	7.575781e-27	1.427504e-22	ENSDARG
ENSDARG00000035519	1963.6138	-1.527732	0.1533885	2.001634e-25	1.885839e-21	ENSDARG
ENSDARG00000063307	879.0429	1.395107	0.1581055	7.882223e-21	4.950824e-17	ENSDARG
ENSDARG00000025788	400.3032	-1.175105	0.1511315	1.770101e-17	8.338505e-14	ENSDARG
ENSDARG00000068374	2936.7863	1.009402	0.1341592	9.013761e-17	3.396926e-13	ENSDARG
ENSDARG00000055723	325.0561	1.555549	0.2051782	1.627645e-16	5.111620e-13	ENSDARG

A data.frame: 6 × 7

	ensembl_gene_id	baseMean	log2FoldChange	lfcSE	pvalue	padj	gene_na
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	ENSDARG00000000001	163.3757	0.00501928	0.1534130	9.547782e-01	9.883977e-01	slc35a5
2	ENSDARG00000000002	2654.8789	0.97021886	0.1397340	6.984866e-15	1.880226e-11	ccdc80
3	ENSDARG00000000018	723.0188	-0.08979923	0.1318007	3.106098e-01	6.650933e-01	nrf1
4	ENSDARG00000000019	4076.2952	-0.21684774	0.1074539	1.142926e-02	1.197784e-01	ube2h
5	ENSDARG00000000068	558.9280	0.05265276	0.1018394	4.851408e-01	7.935337e-01	slc9a3r1

	ensembl_gene_id	baseMean	log2FoldChange	lfcSE	pvalue	padj	gene_na
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
6	ENSDARG00000000069	848.1649	0.07789406	0.1166314	3.470132e-01	6.974199e-01	dap

426

63

▼ Code

```
results_WT_KO_df_gene_description <- merge(results_WT_KO_df_gene, description, by.x =
        "ensembl_gene_id", by.y = "ensembl_gene_id", all.x = T) # merge with
        description
results_WT_KO_df_gene_description %>% head
write.csv(results_WT_KO_df_gene_description, "Results/DEGs_KO_vs_WT_description.csv")
```

A data.frame: 6 × 9

	ensembl_gene_id	baseMean	log2FoldChange	lfcSE	pvalue	padj	gene_na
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	ENSDARG00000000001	163.3757	0.00501928	0.1534130	9.547782e-01	9.883977e-01	slc35a5
2	ENSDARG00000000002	2654.8789	0.97021886	0.1397340	6.984866e-15	1.880226e-11	ccdc80
3	ENSDARG00000000018	723.0188	-0.08979923	0.1318007	3.106098e-01	6.650933e-01	nrf1
4	ENSDARG00000000019	4076.2952	-0.21684774	0.1074539	1.142926e-02	1.197784e-01	ube2h
5	ENSDARG00000000068	558.9280	0.05265276	0.1018394	4.851408e-01	7.935337e-01	slc9a3r1
6	ENSDARG00000000069	848.1649	0.07789406	0.1166314	3.470132e-01	6.974199e-01	dap

6 plot volcano plot

▼ Code

```
# prepare data for volcano plot
results_WT_KO_df_gene_volcano <- results_WT_KO_df_gene
results_WT_KO_df_gene_volcano$padj[is.na(results_WT_KO_df_gene_volcano$padj)] <- 1 #
        replace NA with 1 because as NA is non significant
results_WT_KO_df_gene_volcano$ens_gene <- results_WT_KO_df_gene_volcano$gene_name
results_WT_KO_df_gene_volcano$ens_gene[is.na(results_WT_KO_df_gene_volcano$ens_gene)]
        <- results_WT_KO_df_gene_volcano$ensembl_gene_id
results_WT_KO_df_gene_volcano <- results_WT_KO_df_gene_volcano %>%
        dplyr::arrange(padj)
results_WT_KO_df_gene_volcano %>% head
```

```
Warning message in
results_WT_KO_df_gene_volcano$ens_gene[is.na(results_WT_KO_df_gene_volcano$ens_gene)]
<- results_WT_KO_df_gene_volcano$ensembl_gene_id:
“number of items to replace is not a multiple of replacement length”
```

A data.frame: 6 × 8

	ensembl_gene_id	baseMean	log2FoldChange	lfcSE	pvalue	padj	gene_na
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	ENSDARG00000089791	958.0652	-1.347831	0.1313111	7.575781e-27	1.427504e-22	slc25a3
2	ENSDARG00000035519	1963.6138	-1.527732	0.1533885	2.001634e-25	1.885839e-21	hsth1l
3	ENSDARG00000063307	879.0429	1.395107	0.1581055	7.882223e-21	4.950824e-17	sgsm2
4	ENSDARG00000025788	400.3032	-1.175105	0.1511315	1.770101e-17	8.338505e-14	chp2
5	ENSDARG00000068374	2936.7863	1.009402	0.1341592	9.013761e-17	3.396926e-13	si:ch211-132b12
6	ENSDARG00000055723	325.0561	1.555549	0.2051782	1.627645e-16	5.111620e-13	hsp70l

▼ Code

```
# function to draw volcano plot
require(ggplot2)
require(ggrepel)
require(clusterProfiler)
require(tidyverse)
options(ggrepel.max.overlaps = z`50)
draw_volcano <- function(fileinput, title, lfcthres=1) {
    # fileinput: data frame with log2FoldChange and padj columns
```

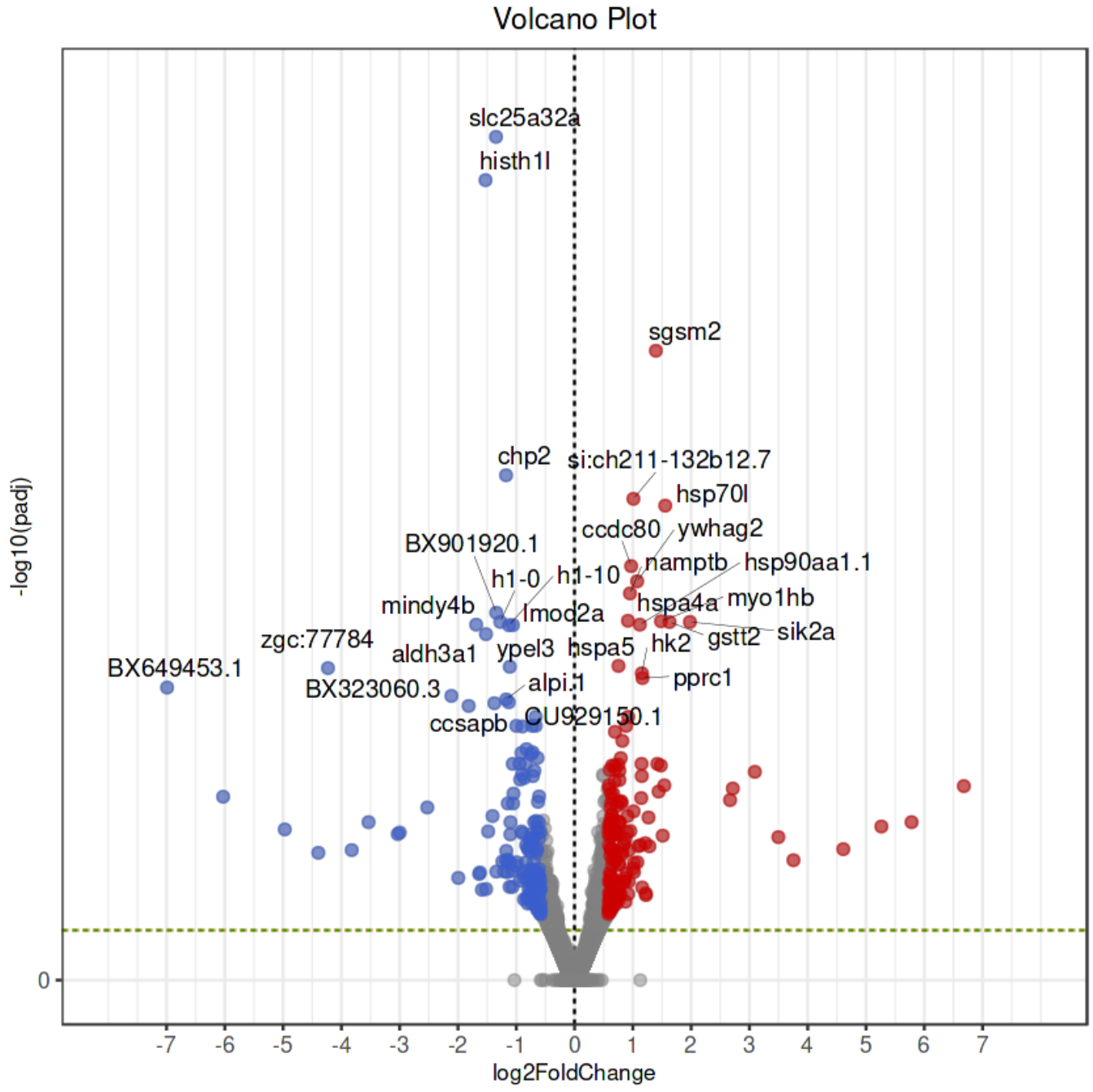
```
    # title: title of the plot
    # lfcthres: log2 fold change threshold
    ggplot(data = fileinput , aes(x = log2FoldChange, y = -log10(padj))) + # make
        ggplot object with log2FoldChange and -log10(padj)
        geom_hline(yintercept = -log10(0.05), linetype = "dotted", col =
        "darkgoldenrod") + # add horizontal line for padj = 0.05
        geom_vline(xintercept = 0, linetype = "dashed") + # add vertical line for
        log2FoldChange = 0
        geom_point(x = fileinput$log2FoldChange, y = -log10(fileinput$padj),alpha =
        0.5, size = 2,color="grey51") + # add points with color grey
        geom_point(data = fileinput[which(fileinput$padj < 0.05 &
        fileinput$log2FoldChange < -lfcthres),],# add points with padj < 0.05 and
        log2FoldChange < -lfcthres with color blue
                  aes(x=log2FoldChange, y = -log10(padj)), shape = 21, color =
        "royalblue3", fill = "royalblue3",
                  alpha = 0.5, size = 2) +
        geom_point(data = fileinput[which(fileinput$padj < 0.05 &
        fileinput$log2FoldChange > lfcthres),], # add points with padj < 0.05 and
        log2FoldChange > lfcthres with color red
                  aes(x=log2FoldChange, y = -log10(padj)), shape = 21, color =
        "red3", fill = "red3",
                  alpha = 0.5, size = 2) +
        scale_x_continuous(breaks = seq(round(min(fileinput$log2FoldChange)- 0.5),
                                        round(max(fileinput$log2FoldChange)+ 0.5),by
        = 1),
                           limits =
        c(round(min(fileinput$log2FoldChange)-1),round(max(fileinput$log2FoldChange)+1
        + # set x axis limits
        scale_y_continuous(breaks = seq(0,round(-log10(min(fileinput$padj)+1)),by =
        4),
                           limits = c(0,round(-log10(min(fileinput$padj))+1))) + #
        set y axis limits
        ggtitle(title) +# add title based on input
        theme_bw() + # set theme to black and white
        theme(plot.title = element_text(hjust = 0.5), axis.text = element_text(size =
        10),# set theme for plot title and axis text
              axis.title.x = element_text(size = 10), axis.title.y =
        element_text(size = 10))
}
# Draw volcano plot
b = draw_volcano(results_WT_KO_df_gene_volcano, "Volcano Plot",lfcthres = 0.58)
# Add labels for significant genes:
gene_name <- results_WT_KO_df_gene_volcano %>% arrange(padj) %>% pull(ens_gene) %>%
        as.character() %>% head(30) # select top 30 genes for labeling
print(gene_name)
c = b + geom_text_repel(data =
        results_WT_KO_df_gene_volcano[results_WT_KO_df_gene_volcano$ens_gene %in%
        gene_name,], aes(label = gene_name),
                        nudge_x = 0.5, nudge_y = 0.5, segment.size = 0.1) # add
        labels for significant genes to the plot
# Save the plot with labels:
print(c)
dev.copy(pdf, file = "Results/Volcano_plot.pdf", width = 12, height = 12)
dev.off()
```

```
[1] "slc25a32a"         "histh1l"           "sgsm2"
[4] "chp2"              "si:ch211-132b12.7" "hsp70l"
[7] "ccdc80"            "ywhag2"            "namptb"
```

```
 [10] "BX901920.1"        "hspa4a"            "myo1hb"
 [13] "h1-0"              "sik2a"             "gstt2"
 [16] "hsp90aa1.1"        "mindy4b"           "lmod2a"
 [19] "h1-10"             "aldh3a1"           "hspa5"
 [22] "ypel3"             "zgc:77784"         "hk2"
 [25] "pprc1"             "BX649453.1"        "BX323060.3"
 [28] "alpi.1"            "CU929150.1"        "ccsapb"
```

pdf: 4

png: 2



7 Perform Pathway analysis

▼ Code

```
# Basic function to convert zebrafish to mouse gene names
```

```r
zgGenes <- results_WT_KO_df_gene$ensembl_gene_id

# This function uses biomaRt to convert zebrafish gene names to mouse gene names.
# This is done by using the zebrafish and mouse ensembl mart datasets.
# The function takes a vector of zebrafish gene names as input and returns
# a data frame of zebrafish and mouse gene names.

convertDanioGeneList_Mouse <- function(x){
  require("biomaRt") # load biomaRt package
  mouse = useMart("ensembl", dataset = "mmusculus_gene_ensembl", host =
        "https://dec2021.archive.ensembl.org/") # use mouse mart
  danio = useMart("ensembl", dataset = "drerio_gene_ensembl", host =
        "https://dec2021.archive.ensembl.org/") # use zebrafish mart

  genesV2 = getLDS(attributes = c("ensembl_gene_id", "zfin_id_symbol"),
                   filters = "ensembl_gene_id", # get zebrafish gene names
                   values = x , # use the zebrafish gene names
                   mart = danio,  # use the zebrafish mart
                   attributesL = c("mgi_symbol", "ensembl_gene_id", "description"), #
        get mouse gene names
                   martL = mouse, uniqueRows=T) # use the mouse mart

  colnames(genesV2)[colnames(genesV2)== "Gene.stable.ID"] <- "EnsmblID_Zebrafish" #
        rename columns
  colnames(genesV2)[colnames(genesV2)== "Gene.stable.ID.1"] <- "EnsmblID_Mouse" #
        rename columns

  # Check if the gene is not found
  if (length(genesV2) == 0) {
    print("No gene found for this input")
  } else {
    return(genesV2) # return the genes
  }
}

# Run the function
Mouse_Genes <- convertDanioGeneList_Mouse(zgGenes)
# print the first 6 genes
print(head(Mouse_Genes))
```

```
  EnsmblID_Zebrafish ZFIN.symbol MGI.symbol     EnsmblID_Mouse
1 ENSDARG00000063908      mt-co2     mt-Co2 ENSMUSG00000064354
2 ENSDARG00000013438       sycp3    Gm20817 ENSMUSG00000100032
3 ENSDARG00000013438       sycp3    Gm28490 ENSMUSG00000094789
4 ENSDARG00000013438       sycp3    Gm21094 ENSMUSG00000095263
5 ENSDARG00000013438       sycp3    Gm20838 ENSMUSG00000095011
6 ENSDARG00000013438       sycp3    Gm20888 ENSMUSG00000094616
                                                    Gene.description
1 mitochondrially encoded cytochrome c oxidase II [Source:MGI Symbol;Acc:MGI:102503]
2                        predicted gene, 20817 [Source:MGI Symbol;Acc:MGI:5434173]
3                         predicted gene 28490 [Source:MGI Symbol;Acc:MGI:5579196]
4                        predicted gene, 21094 [Source:MGI Symbol;Acc:MGI:5434449]
5                        predicted gene, 20838 [Source:MGI Symbol;Acc:MGI:5434194]
6                        predicted gene, 20888 [Source:MGI Symbol;Acc:MGI:5434244]
```

▼ Code

```
# merge with results_WT_KO_df_gene to get mouse gene names
 results_WT_KO_df_gene_mouse <- merge(results_WT_KO_df, Mouse_Genes, by.x =
        "ensembl_gene_id", by.y = "EnsmblID_Zebrafish", all.x = T)
results_WT_KO_df_gene_mouse %>% head
```

A data.frame: 6 × 10

	ensembl_gene_id	baseMean	log2FoldChange	lfcSE	pvalue	padj	ZFIN.sy
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	ENSDARG00000000001	163.3757	0.00501928	0.1534130	9.547782e-01	9.883977e-01	slc35a5
2	ENSDARG00000000002	2654.8789	0.97021886	0.1397340	6.984866e-15	1.880226e-11	NA
3	ENSDARG00000000018	723.0188	-0.08979923	0.1318007	3.106098e-01	6.650933e-01	nrf1
4	ENSDARG00000000019	4076.2952	-0.21684774	0.1074539	1.142926e-02	1.197784e-01	ube2h
5	ENSDARG00000000068	558.9280	0.05265276	0.1018394	4.851408e-01	7.935337e-01	slc9a3r1
6	ENSDARG00000000069	848.1649	0.07789406	0.1166314	3.470132e-01	6.974199e-01	dap

▼ Code

```
# GO analysis function for DEGs
GO_function <- function(gene_list, pval = 0.05, onto = "MF", prefix = "", org =
        "mouse") { # gene_list: vector of gene names, pval: pvalue cutoff, onto: GO
        term, prefix: prefix for output file, org: organism
    require(clusterProfiler) # load clusterProfiler
    require(org.Mm.eg.db)# load org.Mm.eg.db
    require(org.Hs.eg.db)# load org.Hs.eg.db
    require(ReactomePA) # load ReactomePA
    if (org == "mouse") { # check organism
        orgdb <- "org.Mm.eg.db" # set orgdb
        org_reactome <- "mouse" # set org_reactome
    } else if (org == "human") { # check organism
        orgdb <- "org.Hs.eg.db" # set orgdb
        org_reactome <- "human" # set org_reactome
    } else { # if organism is not mouse or human
```

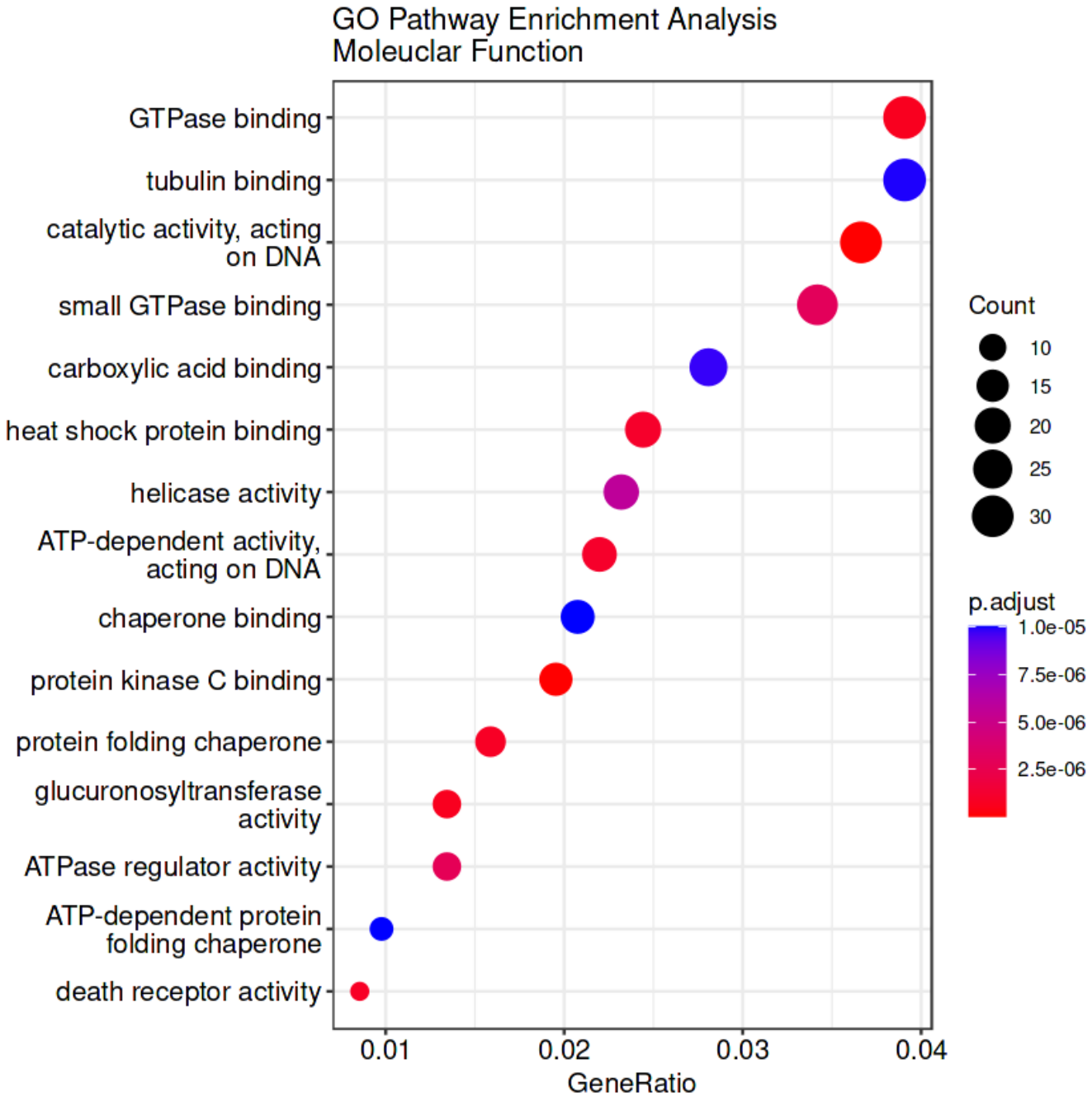
```
    message("Please enter a valid organism (mouse or human)")
}

if (onto %in% c("MF", "CC", "BP")) { # check if GO term is MF, CC or BP
    compGO <- enrichGO(gene = gene_list, pvalueCutoff = pval, keyType = "SYMBOL",
                       pAdjustMethod = "BH", OrgDb = orgdb, ont = onto) # run GO
    analysis
} else if (onto == "reactome") { # check if GO term is reactome
    gene_list <- bitr(gene_list, fromType = "SYMBOL", toType = "ENTREZID", OrgDb
    = orgdb)  # convert gene names to entrez ids
    gene_list <- gene_list$ENTREZID # get entrez ids
    compGO <- enrichPathway(gene = gene_list, pvalueCutoff = 0.05, organism =
    org_reactome, readable = TRUE) # run reactome analysis
} else { # if GO term is not MF, CC, BP or reactome
    message("Please enter a valid GO term")
}

if (is.null(compGO)) { # check if compGO is null
    message(paste0("No GO:", onto, " obtained")) # print message
    message(paste0
    ("********************************************************************************
    message(paste0("\n"))
} else { # if compGO is not null
    compGO_df <- as.data.frame(compGO) # convert to data frame
    compGO_df$GeneRatio_decimal <- compGO_df$GeneRatio # convert GeneRatio to
    decimal
    compGO_df$GeneRatio_decimal <- sapply(compGO_df$GeneRatio_decimal,
                                          function(x) (eval(parse(text =
    as.character(x))))) # convert GeneRatio to decimal
    compGO_df$BgRatio_decimal <- compGO_df$BgRatio # convert BgRatio to decimal
    compGO_df$BgRatio_decimal <- sapply(compGO_df$BgRatio_decimal,
                                        function(x) (eval(parse(text =
    as.character(x))))  # convert BgRatio to decimal
    compGO_df <- compGO_df %>% tidyr::separate_rows(geneID, sep = "/", convert =
    FALSE) %>%
        arrange(desc(GeneRatio_decimal)) # separate geneID column by / and
    arrange by GeneRatio_decimal
    compGO_df %>% head # check file

    if (nrow(compGO_df) == 0) { # check if compGO_df is empty
        message(paste0("No GO:", onto, " obtained"))

    message(paste0("****************************************************************
        message(paste0("\n"))
    } else { # if compGO_df is not empty
        write.csv(compGO_df, paste0(prefix, "_GO_", onto, "_pathways.csv"))

        full_name = switch(onto, # get full name of GO term

            MF = "Moleuclar Function",
            CC = "Cellular Components",
            BP = "Biological Processes",
            reactome = "Reactome Pathways"
        )

        print(dotplot(compGO, showCategory = 15, title = paste0("GO Pathway
    Enrichment Analysis \n", full_name),
```

```
                font.size = 12)) # plot dotplot
            dev.copy( # save plot
                pdf,
                file = paste0(prefix, "_GO_", onto, "_pathways.pdf"),
                width = 10,
                height = 12
            )
            dev.off ()

            message(paste0("Pathway analysis GO:", onto, " done"))

        message(paste0("**********************************************************************
            message(paste0("\n"))
        }
    }
}
```

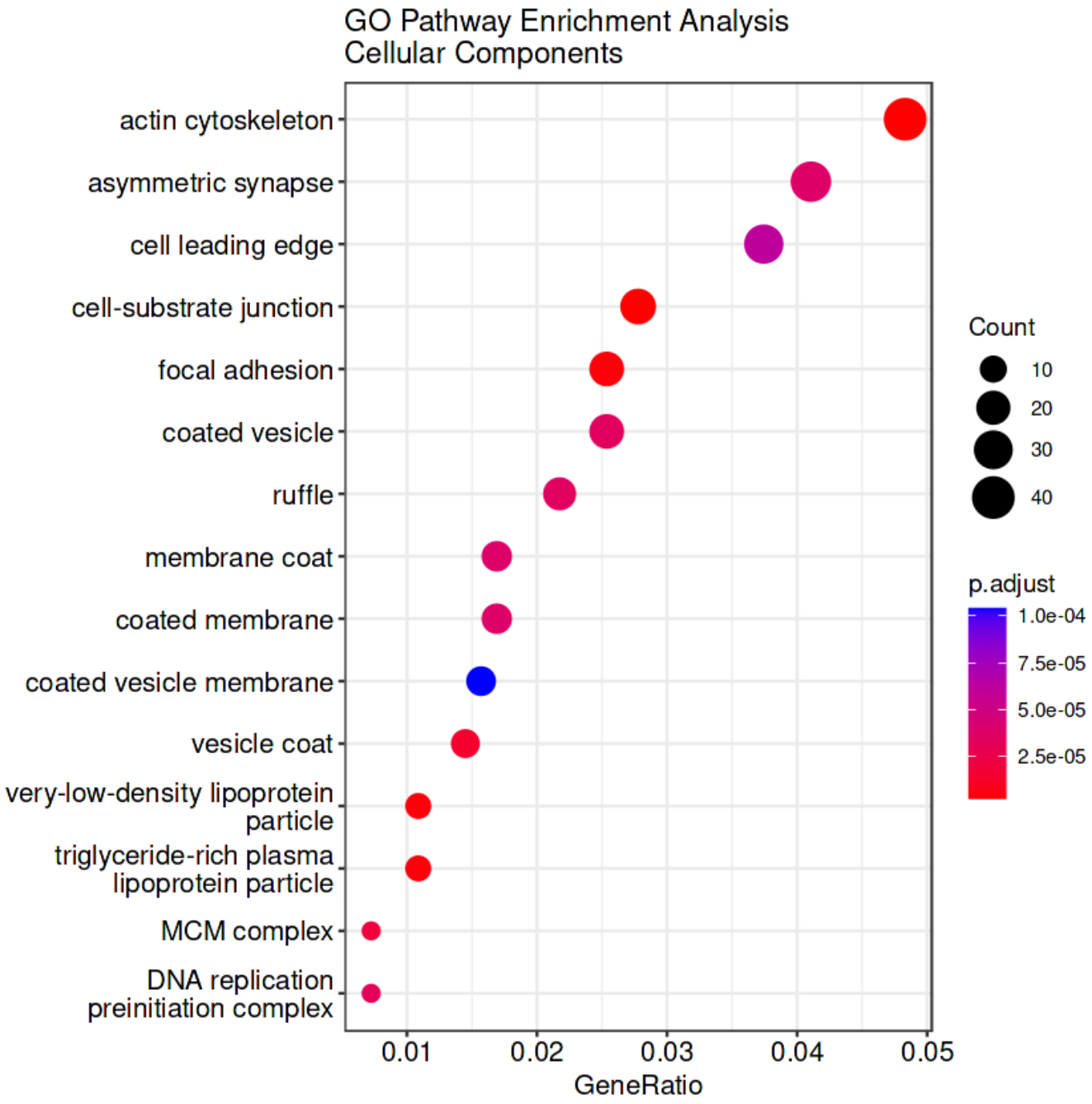
▼ Code

```
# Perform GO analysis on DEGs
dir.create("Results/GO", recursive = T) # create a GO directory to store results
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>% pull(MGI.symbol)),
        pval = 0.05, onto = "MF", prefix = "Results/GO/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>% pull(MGI.symbol)),
        pval = 0.05, onto = "CC", prefix = "Results/GO/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>% pull(MGI.symbol)),
        pval = 0.05, onto = "BP", prefix = "Results/GO/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05) %>% pull(MGI.symbol)),
        pval = 0.05, onto = "reactome", prefix = "Results/GO/KO_vs_WT", org =
        "mouse")
```

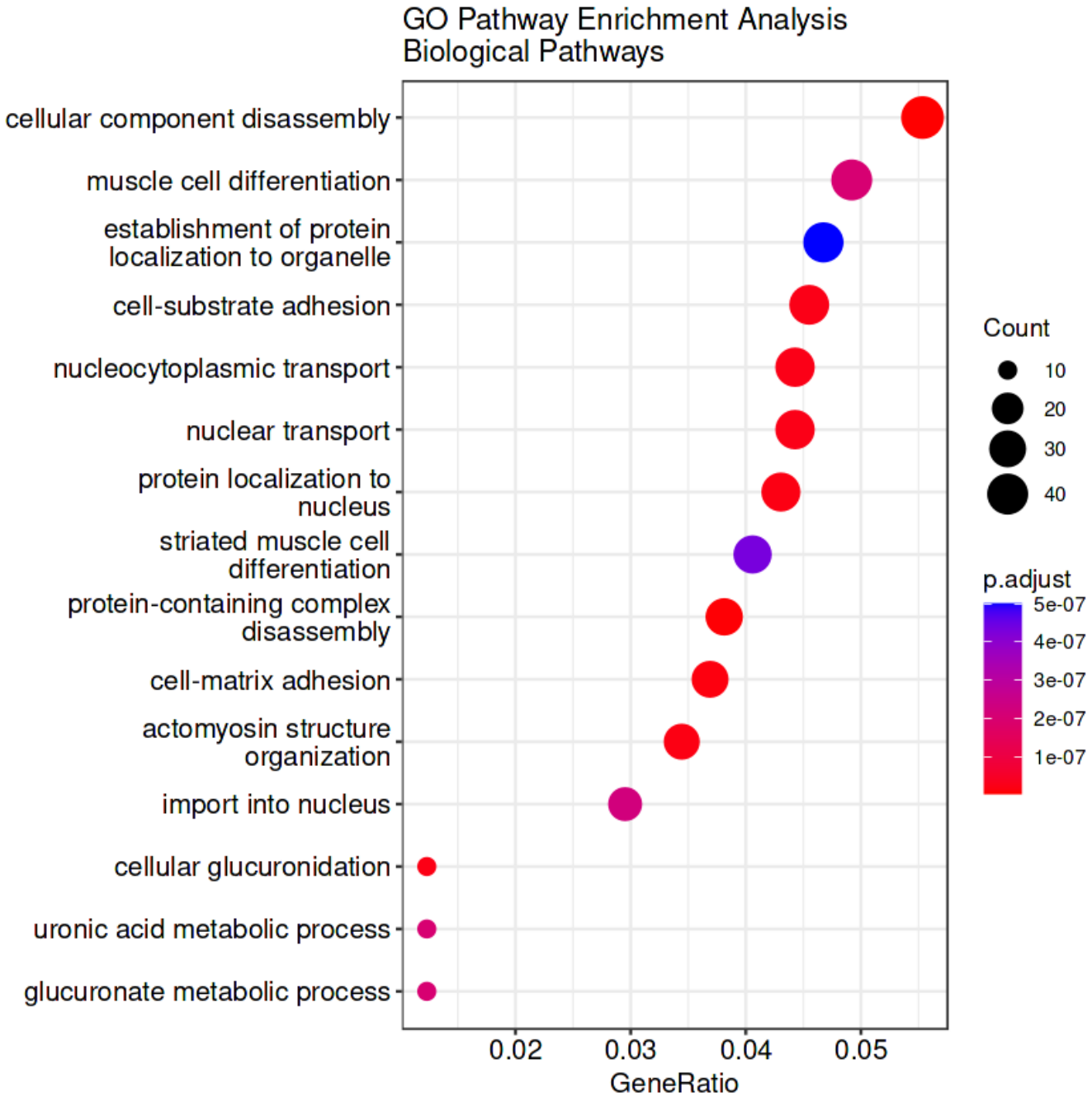
```
Pathway analysis GO:MF done

**********************************************************************************
**



Pathway analysis GO:CC done

**********************************************************************************
**
```

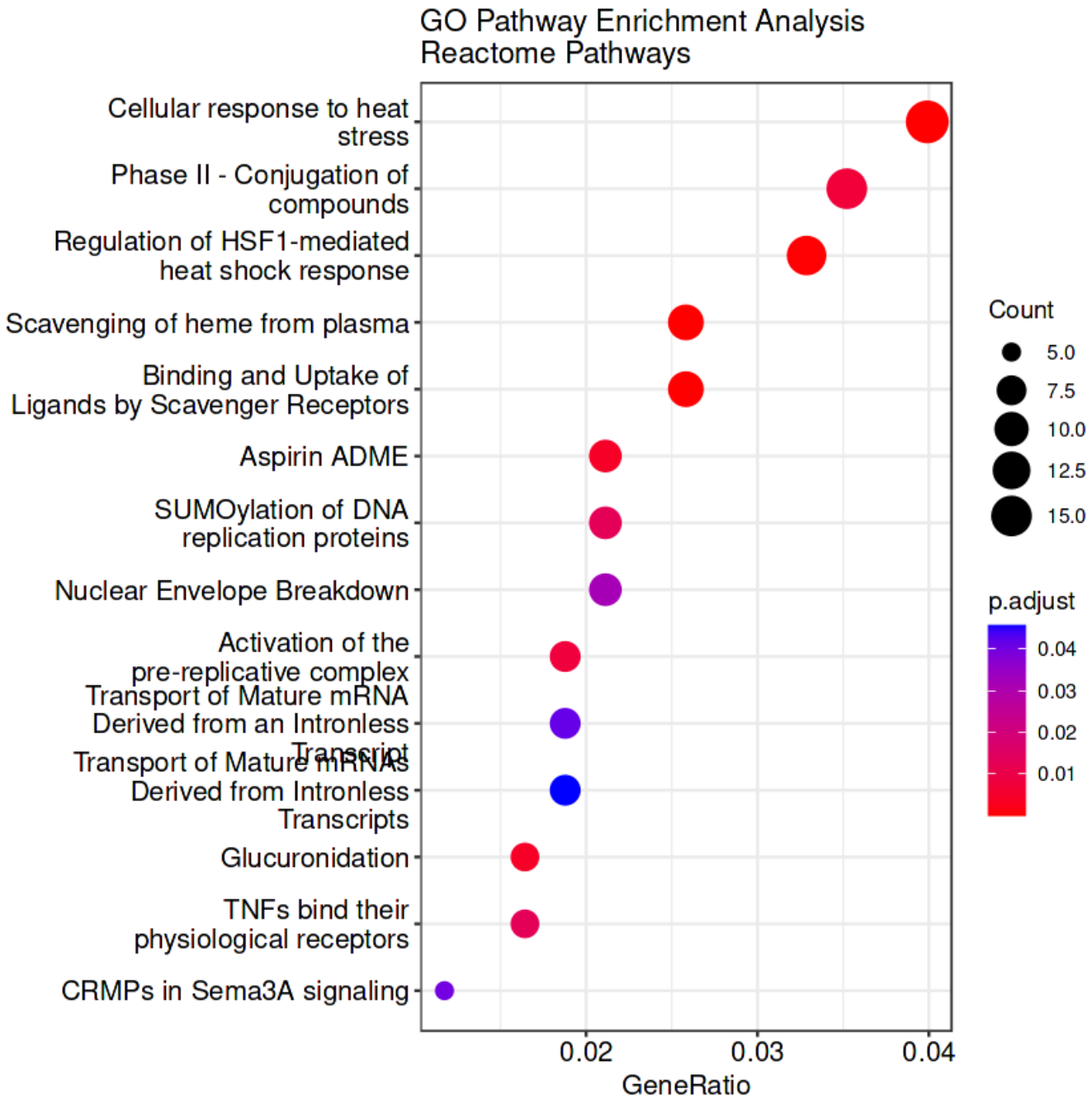



```
Pathway analysis GO:BP done

**********************************************************************************
**
```




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

Warning message in bitr(gene_list, fromType = "SYMBOL", toType = "ENTREZID", OrgDb =
orgdb):
“1.74% of input gene IDs are fail to map...”
Pathway analysis GO:reactome done

**********************************************************************************
**
```


GO Pathway Enrichment Analysis
Biological Pathways
cellular component disassembly
muscle cell differentiation
establishment of protein localization to organelle
cell-substrate adhesion
nucleocytoplasmic transport
nuclear transport
protein localization to nucleus
striated muscle cell differentiation
protein-containing complex disassembly
cell-matrix adhesion
actomyosin structure organization
import into nucleus
cellular glucuronidation
uronic acid metabolic process
glucuronate metabolic process
0.02
0.03
0.04
0.05
GeneRatio
Count
10
20
30
40
p.adjust
5e-07
4e-07
3e-07
2e-07
1e-07



▼ Code

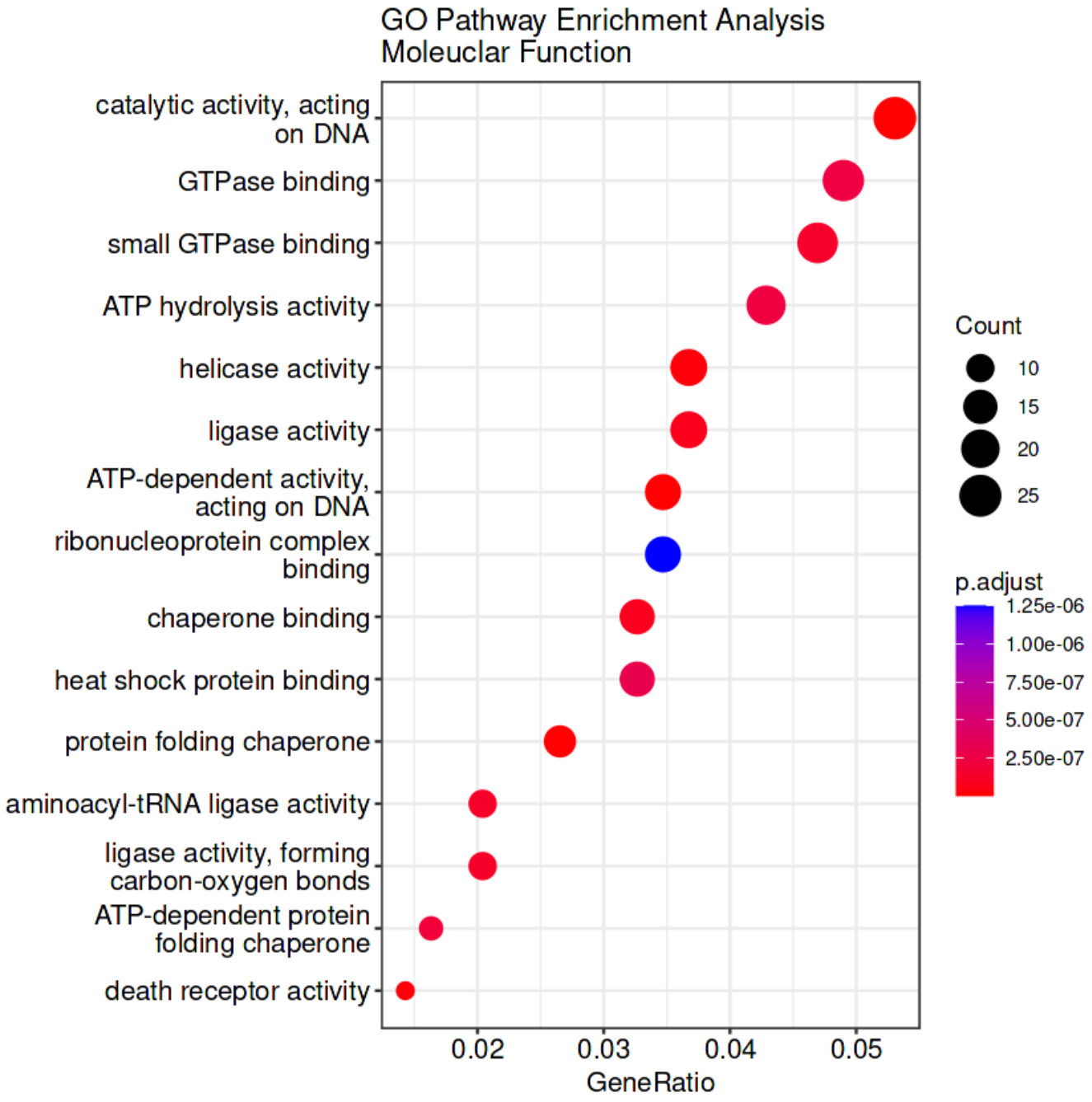
```
# Perform GO analysis on upregulated DEGs
dir.create("Results/GO/up/", recursive = T)
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>%
        filter(log2FoldChange > 0) %>% pull(MGI.symbol)), pval = 0.05, onto = "MF",
        prefix = "Results/GO/up/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>%
        filter(log2FoldChange > 0) %>% pull(MGI.symbol)), pval = 0.05, onto = "CC",
        prefix = "Results/GO/up/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>%
        filter(log2FoldChange > 0) %>% pull(MGI.symbol)), pval = 0.05, onto = "BP",
        prefix = "Results/GO/up/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05) %>%
        filter(log2FoldChange > 0) %>% pull(MGI.symbol)), pval = 0.05, onto =
        "reactome", prefix = "Results/GO/up/KO_vs_WT", org = "mouse")
```

```
Pathway analysis GO:MF done
```

****
**

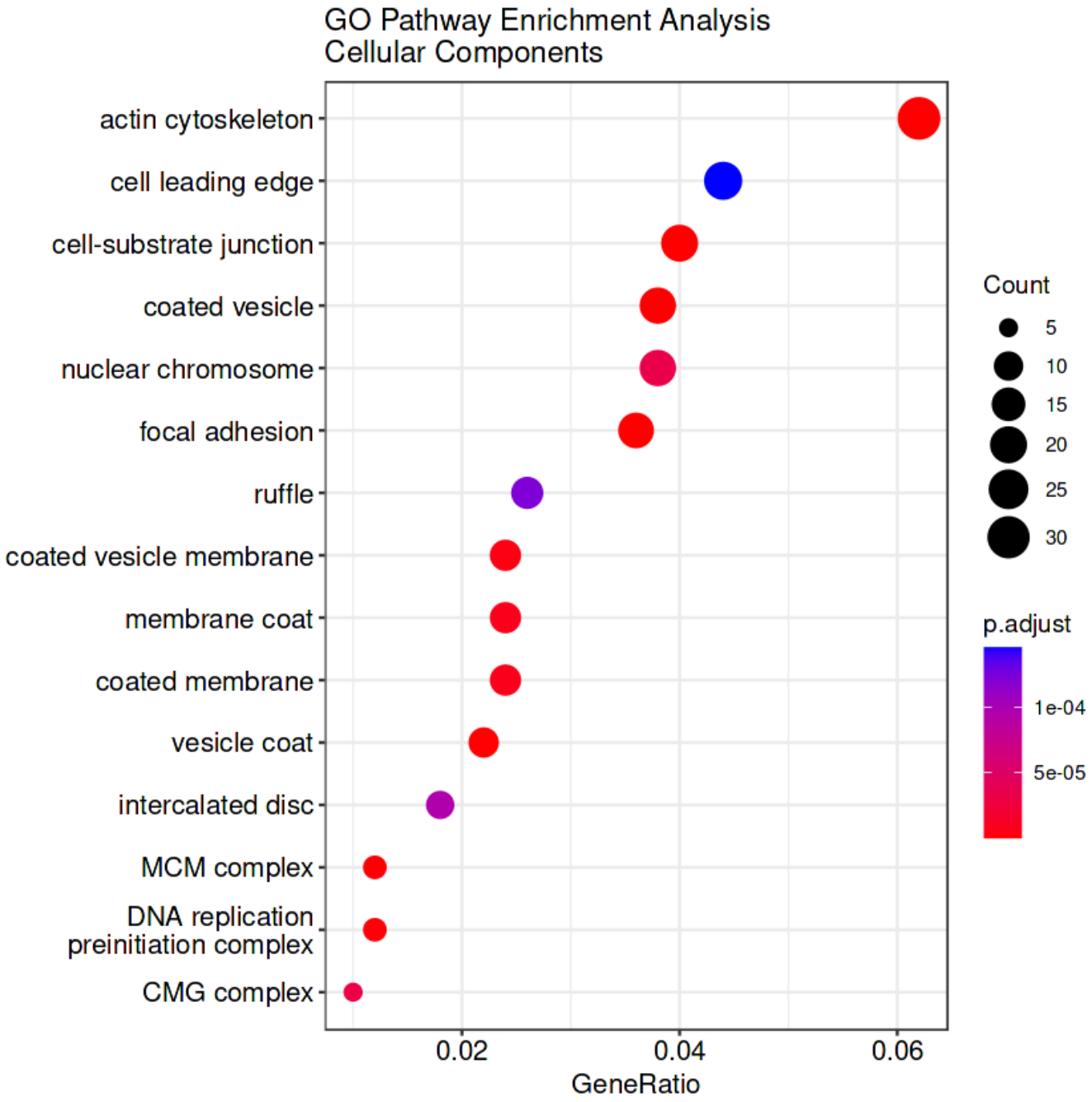
Pathway analysis GO:CC done

****
**

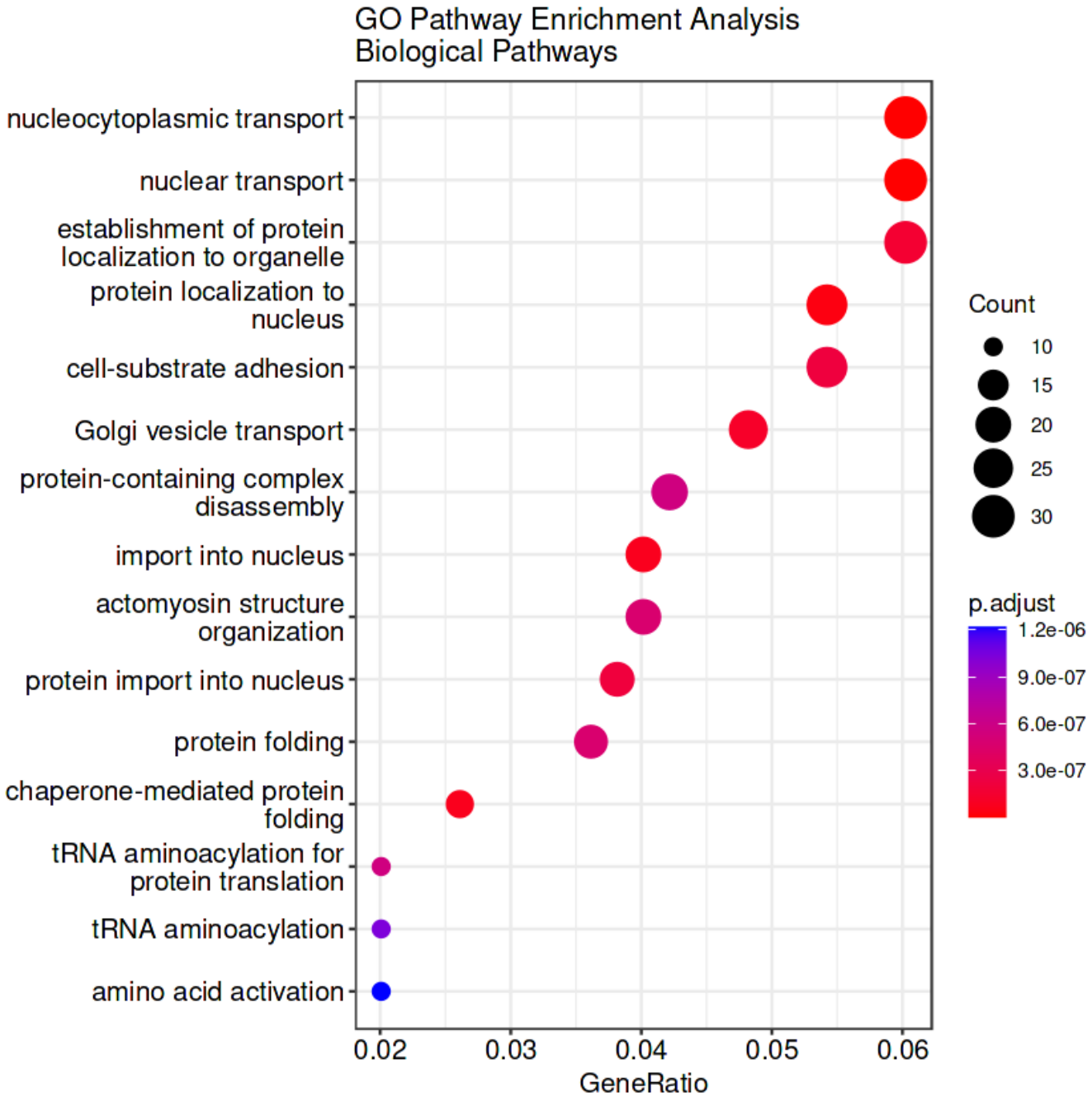


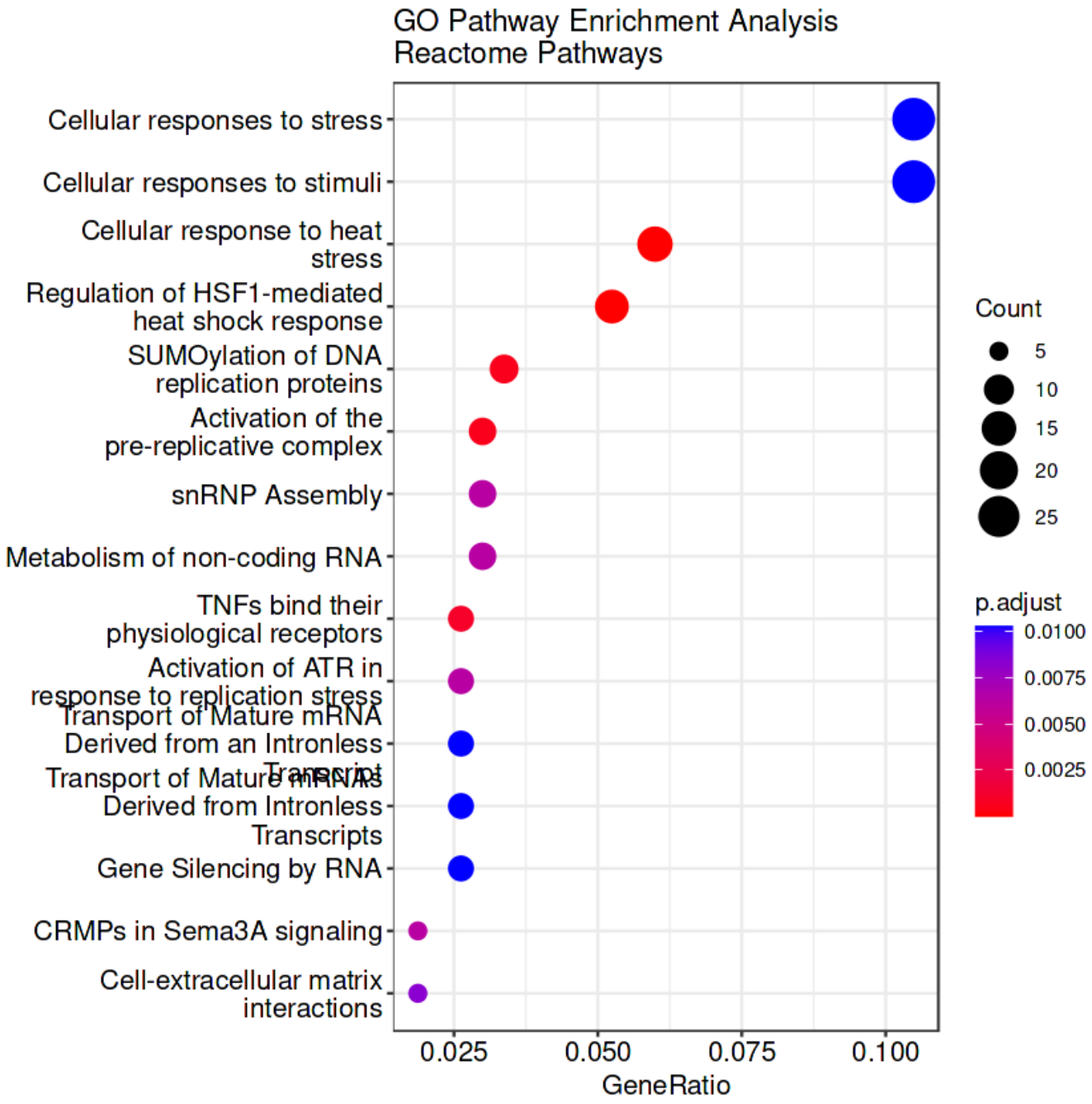
Pathway analysis GO:BP done

****
**




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

Warning message in bitr(gene_list, fromType = "SYMBOL", toType = "ENTREZID", OrgDb =
orgdb):
“1.55% of input gene IDs are fail to map...”
Pathway analysis GO:reactome done

**********************************************************************************
**
```


GO Pathway Enrichment Analysis
Biological Pathways
nucleocytoplasmic transport
nuclear transport
establishment of protein localization to organelle
protein localization to nucleus
cell-substrate adhesion
Golgi vesicle transport
protein-containing complex disassembly
import into nucleus
actomyosin structure organization
protein import into nucleus
protein folding
chaperone-mediated protein folding
tRNA aminoacylation for protein translation
tRNA aminoacylation
amino acid activation
0.02
0.03
0.04
0.05
0.06
GeneRatio
Count
10
15
20
25
30
p.adjust
1.2e-06
9.0e-07
6.0e-07
3.0e-07



▼ Code

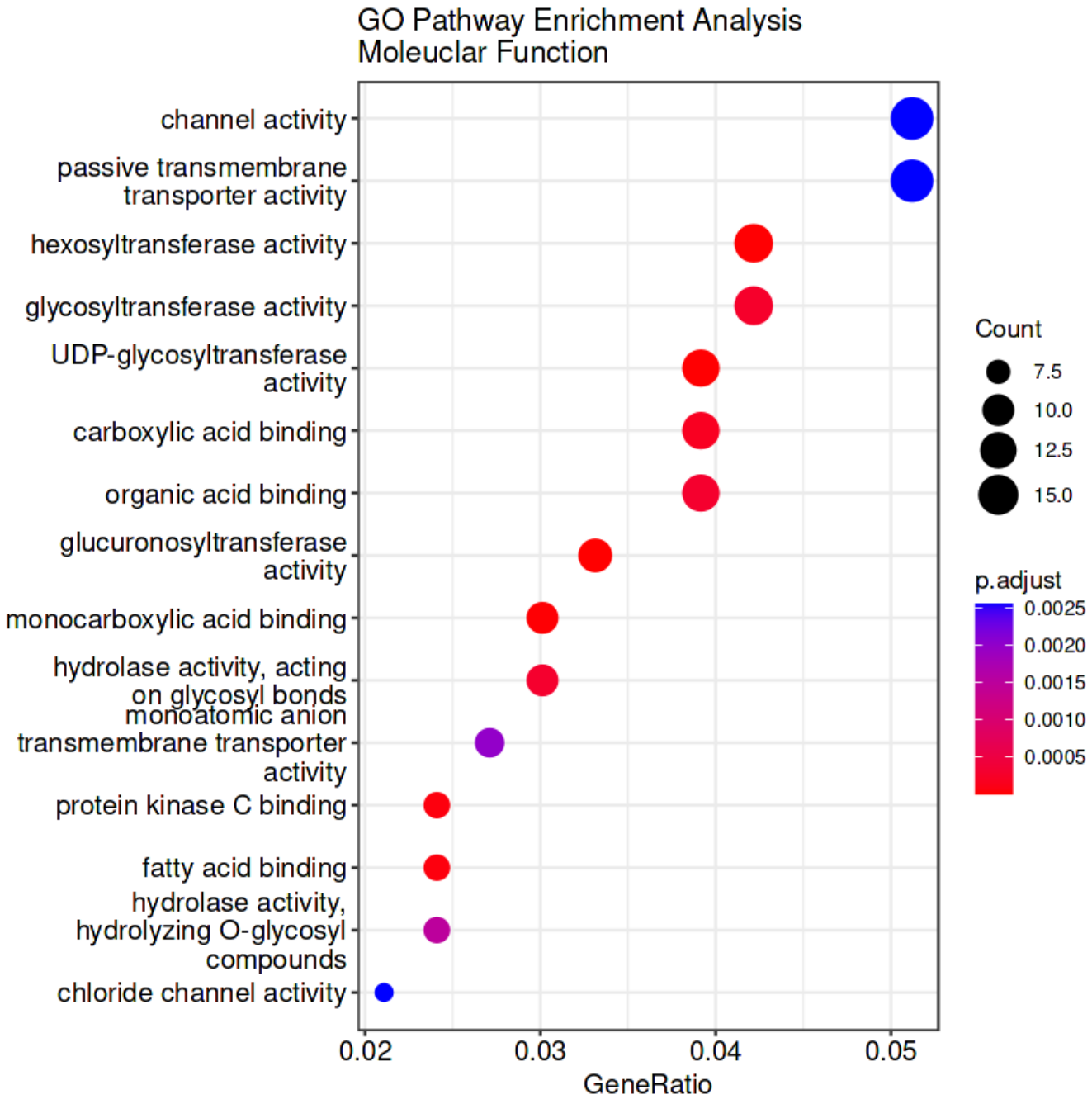
```
#  Perform GO analysis on downregulated DEGs
dir.create("Results/GO/down/", recursive = T)
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>%
        filter(log2FoldChange <= 0) %>% pull(MGI.symbol)), pval = 0.05, onto = "MF",
        prefix = "Results/GO/down/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>%
        filter(log2FoldChange <= 0) %>% pull(MGI.symbol)), pval = 0.05, onto = "CC",
        prefix = "Results/GO/down/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05)%>%
        filter(log2FoldChange <= 0) %>% pull(MGI.symbol)), pval = 0.05, onto = "BP",
        prefix = "Results/GO/down/KO_vs_WT", org = "mouse")
GO_function((results_WT_KO_df_gene_mouse %>% filter(padj<0.05) %>%
        filter(log2FoldChange <= 0) %>% pull(MGI.symbol)), pval = 0.05, onto =
        "reactome", prefix = "Results/GO/down/KO_vs_WT", org = "mouse")
```

```
Pathway analysis GO:MF done
```

****
**

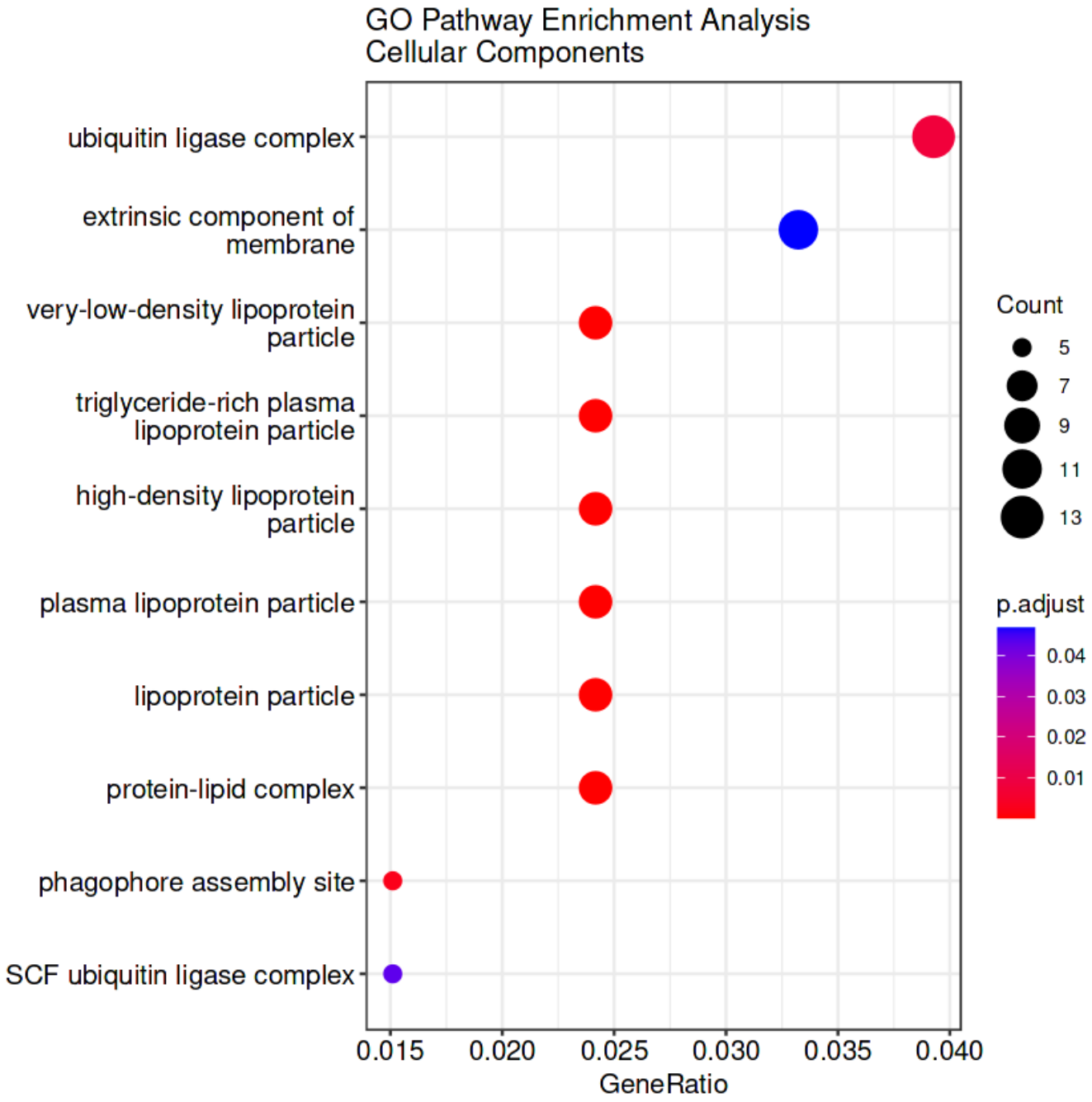
Pathway analysis GO:CC done

****
**

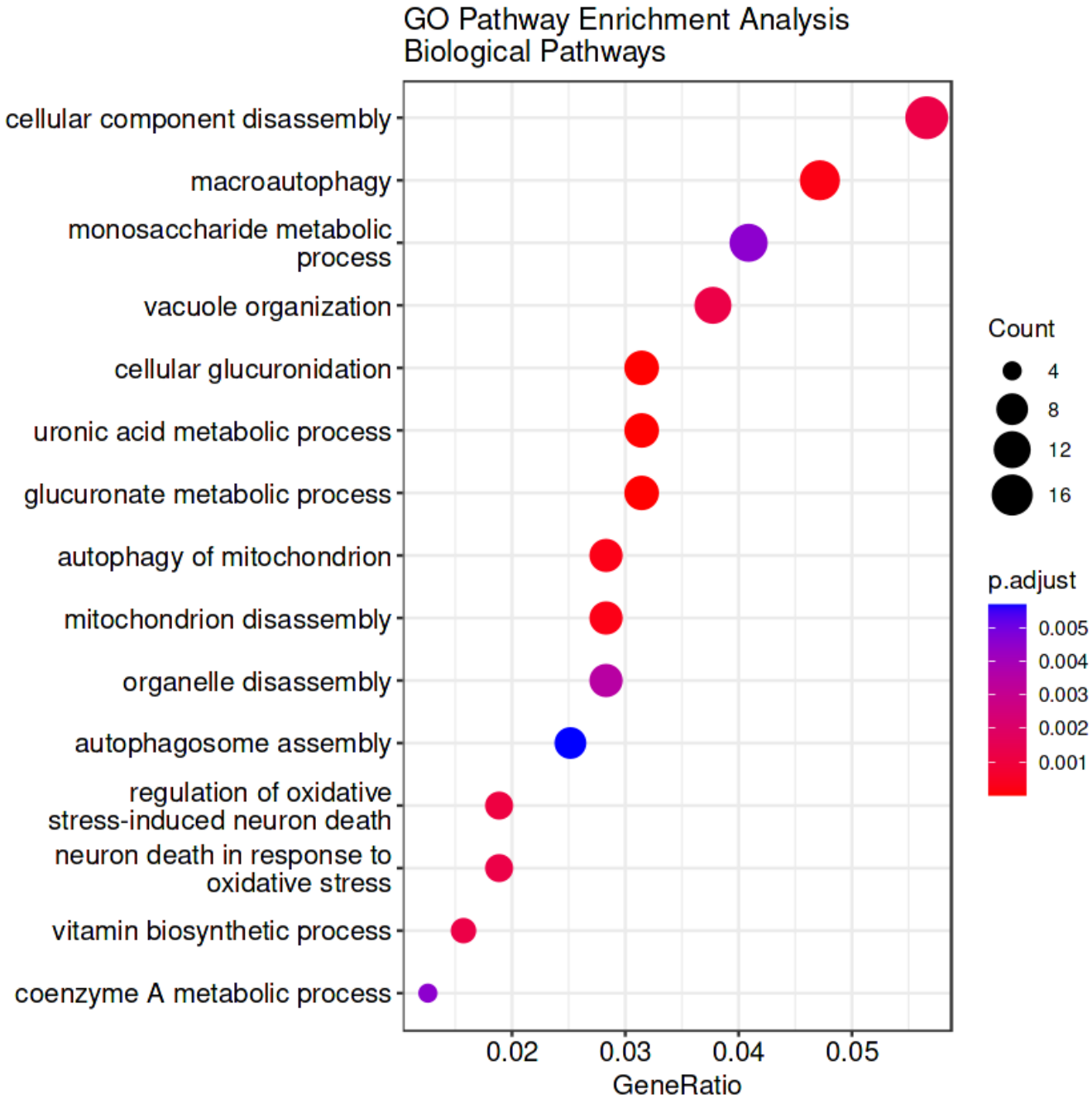


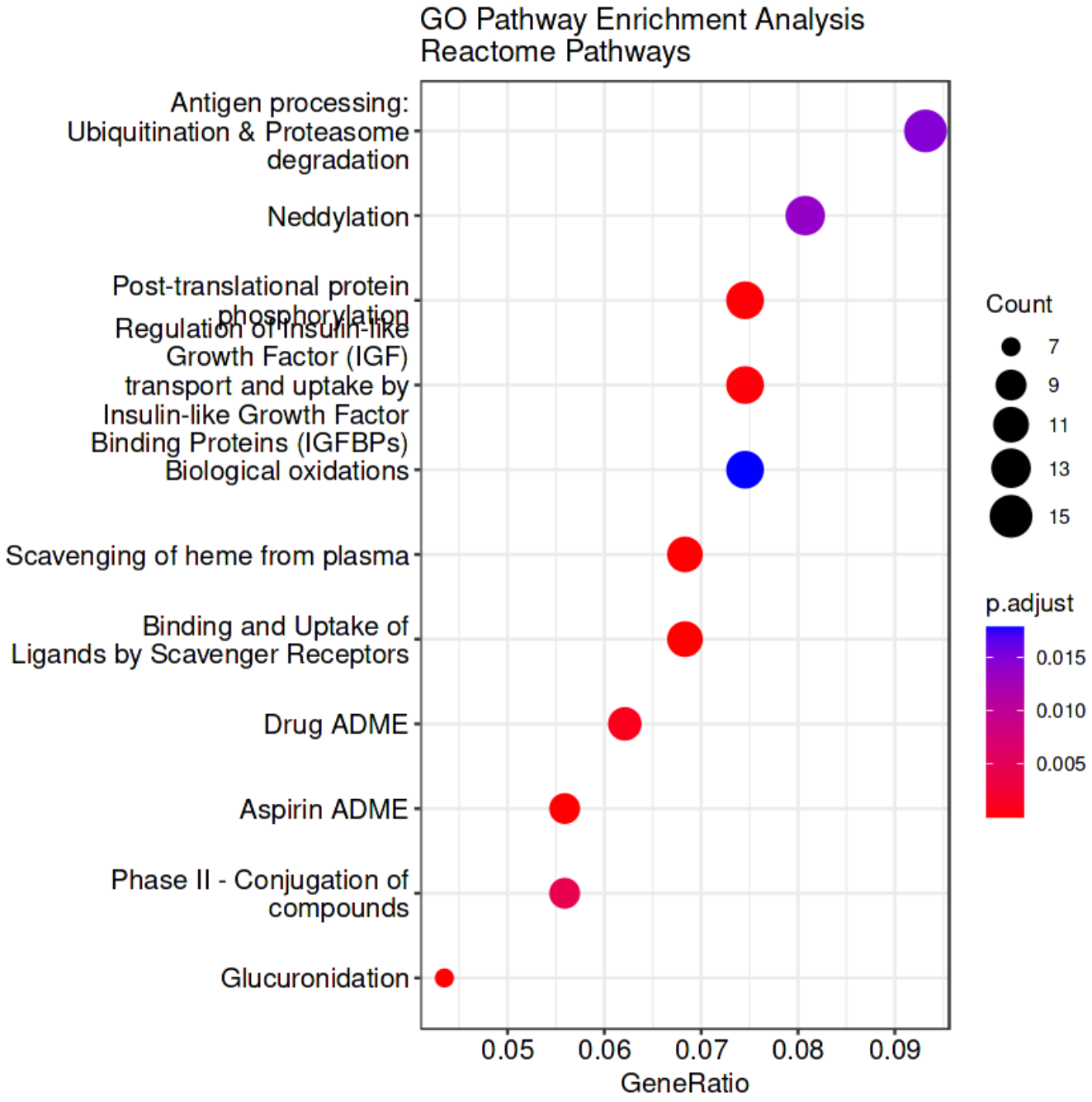
Pathway analysis GO:BP done

****
**




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

Warning message in bitr(gene_list, fromType = "SYMBOL", toType = "ENTREZID", OrgDb =
orgdb):
“2.28% of input gene IDs are fail to map...”
Pathway analysis GO:reactome done

**********************************************************************************
**
```


GO Pathway Enrichment Analysis
Biological Pathways
cellular component disassembly
macroautophagy
monosaccharide metabolic process
vacuole organization
cellular glucuronidation
uronic acid metabolic process
glucuronate metabolic process
autophagy of mitochondrion
mitochondrion disassembly
organelle disassembly
autophagosome assembly
regulation of oxidative stress-induced neuron death
neuron death in response to oxidative stress
vitamin biosynthetic process
coenzyme A metabolic process
0.02
0.03
0.04
0.05
GeneRatio
Count
4
8
12
16
p.adjust
0.005
0.004
0.003
0.002
0.001



▼ Code

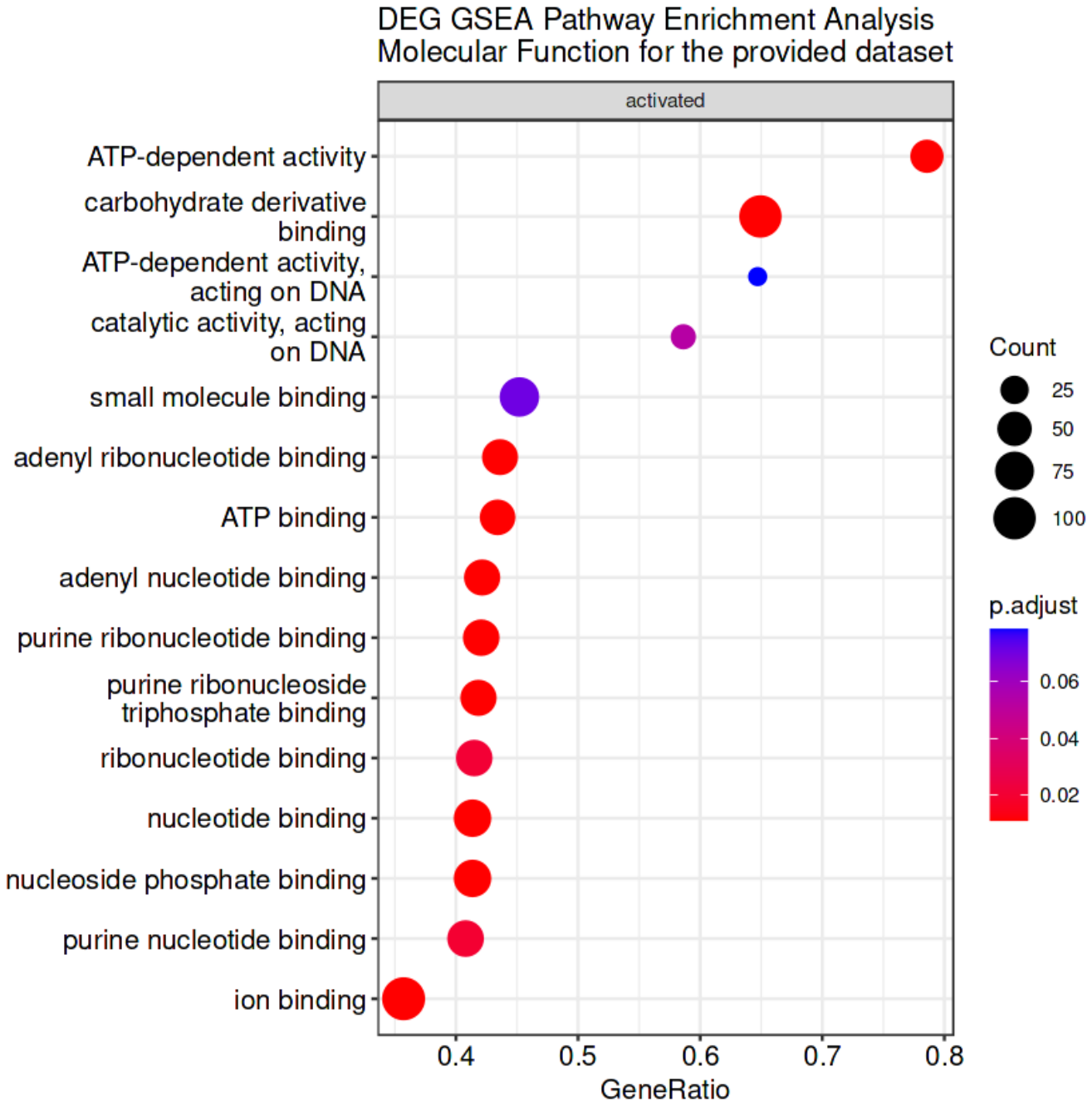
```
# GSEA function for bulk RNASeq data
GSEA_function_bulk <- function(df, pval_deg = 0.05, pval_enrich = 0.1, onto = "MF",
        prefix = "", pdf_width = 12, pdf_height = 12) {
    # df: data frame with log2FoldChange and padj columns
    # pval_deg: pvalue cutoff for DEGs
    # pval_enrich: pvalue cutoff for enrichment
    # onto: GO term
    # prefix: prefix for output file
    # pdf_width: width of pdf
    # pdf_height: height of pdf

    gene_list_df <- df[df$padj <= pval_deg, ] # get DEGs
    gene_list_df <- gene_list_df %>% arrange(desc(log2FoldChange)) # arrange by
        log2FoldChange
    gene_list_df <- gene_list_df[!is.na(gene_list_df$MGI.symbol), ] # remove NA
    gene_list_df <- gene_list_df[!duplicated(gene_list_df$MGI.symbol), ] # remove
        duplicates
```

```
    gene_list <- gene_list_df %>% pull(log2FoldChange) # get log2FoldChange
    names(gene_list) <- gene_list_df %>% pull(MGI.symbol) # set gene names as names
        of log2FoldChange
    gene_list <- gene_list[!duplicated(gene_list)] # remove duplicates
    print(head(gene_list)) # check file

    compGO <- gseGO(gene = gene_list, pvalueCutoff = pval_enrich, keyType = "SYMBOL",
                    pAdjustMethod = "BH", OrgDb = "org.Mm.eg.db", ont = onto) # run
        GSEA
    if (is.null(compGO) | nrow(compGO@result) == 0) { # check if compGO is null or if
        nrow of compGO@result is 0
        message(paste0("No GP:", onto, " obtained for the provided dataset"))
        message(paste0("**********************************************************
        message(paste0("\n"))
    } else { # if compGO is not null
        compGO_df <- as.data.frame(compGO) # convert to data frame
        compGO_df <- compGO_df %>% tidyr::separate_rows(core_enrichment, sep = "/",
        convert = FALSE) %>% arrange((p.adjust)) # separate core_enrichment column
        by / and arrange by p.adjust

        if (nrow(compGO_df) == 0) { # check if compGO_df is empty
            message(paste0("No GP:", onto, " obtained for the provided dataset"))

        message(paste0("**********************************************************
            message(paste0("\n"))
        } else { # if compGO_df is not empty
            write.csv(compGO_df, paste0(prefix, "_GSEA_", onto, "_pathways.csv")) #
        write to csv

            full_name = switch(onto, # get full name of GO term
                MF = "Molecular Function",
                CC = "Cellular Components",
                BP = "Biological Pathways"
            )

            print(dotplot(compGO, showCategory = 15, title = paste0("DEG GSEA Pathway
        Enrichment Analysis \n", full_name, " for the provided dataset"),
                      font.size = 12) + facet_grid(.~.sign)) # plot dotplot
            dev.copy( # save plot
                pdf,
                file = paste0(prefix, "_GSEA_", onto, "_pathways.pdf"),
                width = pdf_width,
                height = pdf_height
            )
            dev.off ()

            message(paste0("DEG Pathway analysis GSEA:", onto, " for the provided
        dataset done"))

        message(paste0("**********************************************************
            message(paste0("\n"))
        }
    }
}
# Run GSEA on DEGs
dir.create("Results/GSEA", recursive = T)
```

```r
GSEA_function_bulk(df = (results_WT_KO_df_gene_mouse %>% filter(padj<0.05)), pval_deg
        = 0.05, pval_enrich = 0.1, onto = "MF", prefix = "Results/GSEA/KO_vs_WT")
GSEA_function_bulk(df = (results_WT_KO_df_gene_mouse %>% filter(padj<0.05)), pval_deg
        = 0.05, pval_enrich = 0.1, onto = "CC", prefix = "Results/GSEA/KO_vs_WT")
GSEA_function_bulk(df = (results_WT_KO_df_gene_mouse %>% filter(padj<0.05)), pval_deg
        = 0.05, pval_enrich = 0.1, onto = "BP", prefix = "Results/GSEA/KO_vs_WT")
```

```
    Oit3     Chd5   Npffr2    Gstt2    Hspa2    Smoc1
6.678322 2.714058 2.667678 1.629230 1.555549 1.540089

preparing geneSet collections...

GSEA analysis...

leading edge analysis...

done...

DEG Pathway analysis GSEA:MF for the provided dataset done

**********************************************************************************
**



    Oit3     Chd5   Npffr2    Gstt2    Hspa2    Smoc1
6.678322 2.714058 2.667678 1.629230 1.555549 1.540089

preparing geneSet collections...

GSEA analysis...

no term enriched under specific pvalueCutoff...

No GP:CC obtained for the provided dataset

**********************************************************************************
**



    Oit3     Chd5   Npffr2    Gstt2    Hspa2    Smoc1
6.678322 2.714058 2.667678 1.629230 1.555549 1.540089

preparing geneSet collections...

GSEA analysis...

no term enriched under specific pvalueCutoff...

No GP:BP obtained for the provided dataset
```

****
**



▼ Code

```
# Read in mitocarta file

mitocarta_mouse_pathways <-
        readxl::read_excel("/mnt/Data_8TB/Carolina_data/Cell_paper/Mouse.MitoCarta3.0.
        sheet = 4)

mitocarta_mouse_pathways <- mitocarta_mouse_pathways %>% dplyr::select(MitoPathway,
        Genes)
# mitocarta_mouse_pathways %>% head
mitocarta_mouse_pathways_rows_list <- mitocarta_mouse_pathways %>%
        tidyr::separate_rows(Genes, sep = ",", convert = FALSE) %>%
        arrange(MitoPathway) %>% dplyr::rename(term= MitoPathway,  gene= Genes)
```

```
# mitocarta_mouse_pathways_rows_list <- mitocarta_mouse_pathways_rows_list
# remove trailing and leading spaces
mitocarta_mouse_pathways_rows_list$gene <- gsub("^\\s+|\\s+$", "",
        mitocarta_mouse_pathways_rows_list$gene)
mitocarta_mouse_pathways_rows_list %>% head
```

A tibble: 6 × 2

term	gene
<chr>	<chr>
ABC transporters	Abca9
ABC transporters	Abcb10
ABC transporters	Abcb6
ABC transporters	Abcb7
ABC transporters	Abcb8
ABC transporters	Abcd1

▼ Code

```
# prepare data for GSEA on mitocarta pathways
gene_list_df <- results_WT_KO_df_gene_mouse[results_WT_KO_df_gene_mouse$padj<=0.05,]
        # get DEGs
gene_list_df <- gene_list_df %>% arrange(desc(log2FoldChange)) # arrange by
        log2FoldChange
gene_list_df <- gene_list_df[!is.na(gene_list_df$MGI.symbol),] # remove NA
gene_list_df <- gene_list_df[!duplicated(gene_list_df$MGI.symbol),] # remove
        duplicates
gene_list <- gene_list_df %>% pull(log2FoldChange) # get log2FoldChange
names(gene_list) <- gene_list_df %>% pull(MGI.symbol) # set gene names as names of
        log2FoldChange

gene_list <- gene_list[!duplicated(gene_list)] # remove duplicates
print(head(gene_list))  # check file
```

```
    Oit3     Chd5   Npffr2    Gstt2    Hspa2    Smoc1
6.678322 2.714058 2.667678 1.629230 1.555549 1.540089
```

▼ Code

```
names(gene_list)[names(gene_list) %in% mitocarta_mouse_pathways_rows_list$gene] #
        check if gene names are in mitocarta pathways
names(gene_list)[grepl("Cox7", names(gene_list))] # check if gene names are in
        mitocarta pathways
(mitocarta_mouse_pathways_rows_list$gene)[grepl("Cox7",
        (mitocarta_mouse_pathways_rows_list$gene))] # check if gene names are in
        mitocarta pathways
```

'Mpv17l2' · 'Kmo' · 'Tbrg4' · 'Endog' · 'Lonp1' · 'Hspd1' · 'Yars2' · 'Acaa2' · 'Acp6' · 'Slc25a47' · 'Pdss2' · 'Afg3l1' · 'Akap1' · 'Alas1' · 'Ecsit' · 'Hspa9' · 'Abcb10' · 'Gfm1' · 'Iars2' · 'Acaca' · 'Ppm1k' · 'Acss1' ·

'Marchf5' · 'Amacr' · 'Isca2' · 'Mtx3' · 'Gls2' · 'Bnip3' · 'Crot' · 'Aldh5a1' · 'Mterf4' · 'Ngrn' · 'Oxr1' · 'Oxsm' · 'Cyp27a1' · 'Coq8a' · 'Ak4' · 'Slc25a32' · 'Aldh3a2'
'Cox7a1' · 'Cox7a2' · 'Cox7a2l' · 'Cox7b' · 'Cox7c' · 'Cox7a1' · 'Cox7a2' · 'Cox7a2l' · 'Cox7b' · 'Cox7c' · 'Cox7a1' · 'Cox7a2' · 'Cox7a2l' · 'Cox7b' · 'Cox7c' · 'Cox7a2l' · 'Cox7a1' · 'Cox7a2' · 'Cox7a2l' · 'Cox7b' · 'Cox7c' · 'Cox7a2l'

▼ Code

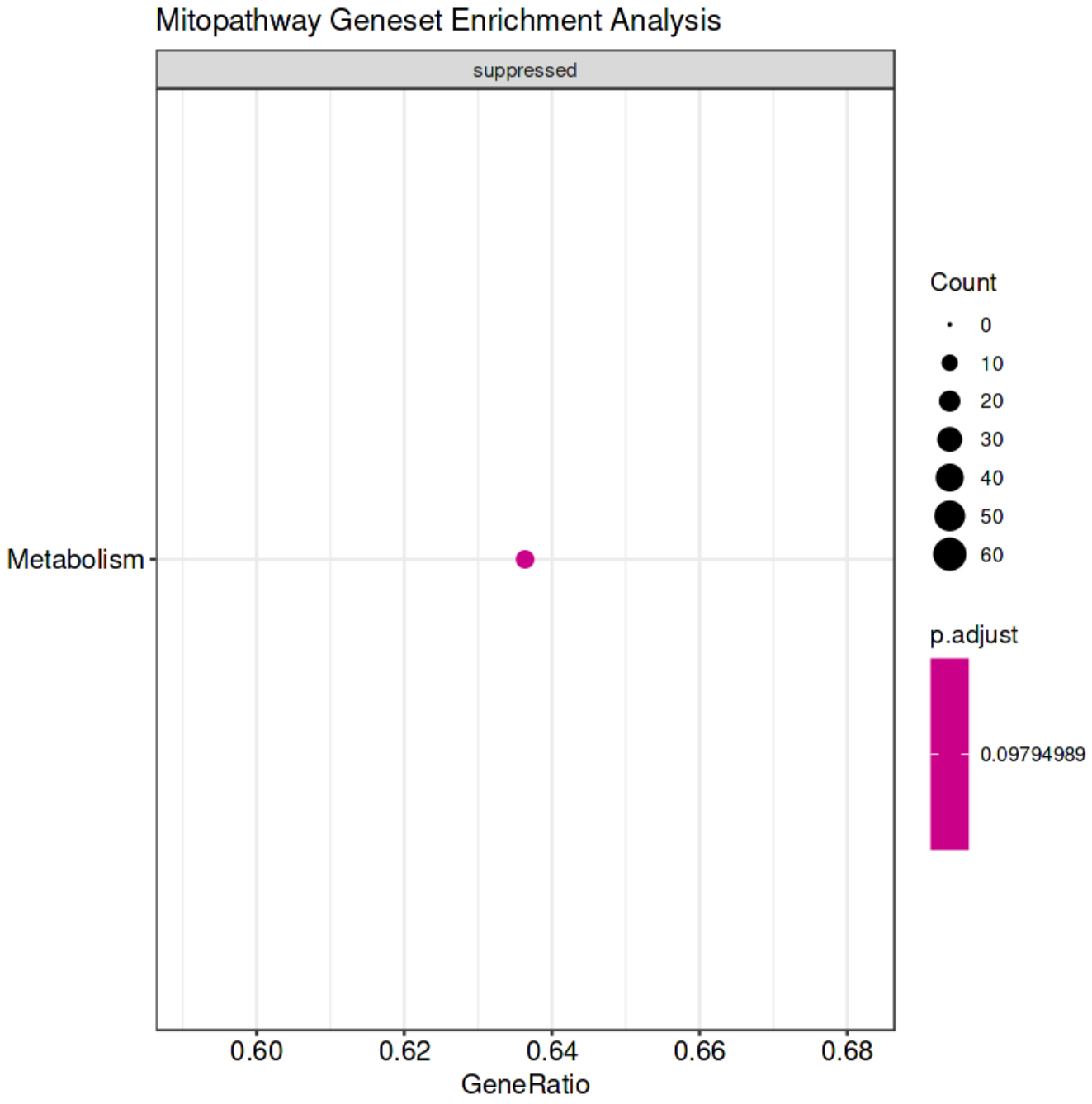
```
set.seed(123) # set seed
gsea_mito <- GSEA(gene_list, TERM2GENE = mitocarta_mouse_pathways_rows_list,
        pvalueCutoff = 0.2)
 dotplot(gsea_mito, showCategory = 15, title = paste0("Mitopathway Geneset Enrichment
        Analysis"),
            font.size = 12) + facet_grid(.~.sign)+ scale_size_area(limits = c(0,60))+
        # plot dotplot
            NULL # plot dotplot
    # Save dotplot as pdf file
    dev.copy(
    pdf,
    file = paste0("Results/GSEA/KO_vs_WT_Mitopathway_GSEA.pdf"),
    width = 22,
    height = 8
    )
    dev.off ()

    gsea_mito_df <- as.data.frame(gsea_mito)
        gsea_mito_df

    gsea_mito_df <- gsea_mito_df %>% tidyr::separate_rows(core_enrichment, sep = "/",
        convert = FALSE) %>%
      arrange((p.adjust))

    # Save enriched pathways data frame as CSV file
    write.csv(gsea_mito_df, paste0("Results/GSEA/KO_vs_WT_Mitopathway_GSEA.csv"))

    # Print message indicating that analysis for the current cell type is complete
#     message(paste0("Cell type: ", i, " done"))
    message(paste0("**********************************************************************
    message(paste0("\n"))
```

```
preparing geneSet collections...

GSEA analysis...

leading edge analysis...

done...

Scale for size is already present.
Adding another scale for size, which will replace the existing scale.
```

pdf: 4
png: 2

	ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust
	<chr>	<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
Metabolism	Metabolism	Metabolism	22	-0.3414702	-1.474444	0.09794989	0.09794989

****
**



8 Prepare figure for Paper

▼ Code

```
BP <- read.csv("Results/GO/KO_vs_WT_GO_BP_pathways.csv") # read in GO BP file
# BP %>% head
MF <- read.csv("Results/GO/KO_vs_WT_GO_MF_pathways.csv") # read in GO MF file
# MF %>% head

# row bind all the data frames and add a column to indicate the GO term
BP$GO_class <- "Biological Process"
MF$GO_class <- "Molecular Functions"

# row bind all the data frames
GO_all <- rbind(BP, MF)
GO_all %>% head
```

A data.frame: 6 × 13

	X	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue	gene
	<int>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<chr
1	1	GO:0022411	cellular component disassembly	45/813	450/28564	3.408267e-13	1.666642e-09	1.271104e-09	Grw
2	2	GO:0022411	cellular component disassembly	45/813	450/28564	3.408267e-13	1.666642e-09	1.271104e-09	Atp2
3	3	GO:0022411	cellular component disassembly	45/813	450/28564	3.408267e-13	1.666642e-09	1.271104e-09	Nav3
4	4	GO:0022411	cellular component disassembly	45/813	450/28564	3.408267e-13	1.666642e-09	1.271104e-09	Hif1
5	5	GO:0022411	cellular component disassembly	45/813	450/28564	3.408267e-13	1.666642e-09	1.271104e-09	Abce
6	6	GO:0022411	cellular component disassembly	45/813	450/28564	3.408267e-13	1.666642e-09	1.271104e-09	Gsn

▼ Code

```
# Select pathways to plot
terms_for_figure <- c("cell-substrate adhesion",
"striated muscle cell differentiation",
```

```
"regulation of vasculature development",
"positive regulation of cell projection organization",
"regulation of actin filament-based process",
"regulation of actin cytoskeleton organization",
"mitotic cell cycle phase transition",
"cellular response to oxidative stress",
"striated muscle tissue development",
"actin binding",
"histone binding",
"DNA helicase activity",
"calcium ion transmembrane transporter activity",
"UDP-glycosyltransferase activity")
```

▼ Code

```
terms_for_figure
```

'cell-substrate adhesion' · 'striated muscle cell differentiation' · 'regulation of vasculature development' · 'positive regulation of cell projection organization' · 'regulation of actin filament-based process' · 'regulation of actin cytoskeleton organization' · 'mitotic cell cycle phase transition' · 'cellular response to oxidative stress' · 'striated muscle tissue development' · 'actin binding' · 'histone binding' · 'DNA helicase activity' · 'calcium ion transmembrane transporter activity' · 'UDP-glycosyltransferase activity'

▼ Code

```
GO_all_figure <- GO_all %>% dplyr::filter(Description %in% terms_for_figure) # filter
        based on terms_for_figure
GO_all_figure$Description <- str_to_title(GO_all_figure$Description) # make
        description oin title case
GO_all_figure$Description[GO_all_figure$Description=="Dna Helicase Activity"] <- "DNA
        Helicase Activity" # correct spelling
GO_all_figure$Description[GO_all_figure$Description=="Udp-Glycosyltransferase
        Activity"] <- "UDP-Glycosyltransferase Activity" # correct spelling

GO_all_figure <- GO_all_figure %>% group_by(GO_class) %>%
        arrange((GeneRatio_decimal)) %>% ungroup() # arrange by GeneRatio_decimal
GO_all_figure %>% head # check file
GO_all_figure$Description <- factor(GO_all_figure$Description,
        levels=unique(GO_all_figure$Description)) # set levels for description

GO_all_figure %>% str
GO_all_figure$Description %>% unique()
GO_all_figure$GeneRatio_decimal %>% max
```

A tibble: 6 × 13

X	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue	ge
<int>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<c
1157	GO:0003678	DNA Helicase Activity	9/819	56/28171	3.162251e-05	0.0008110243	0.0006069955	D

X	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue	ge
<int>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<c
1158	GO:0003678	DNA Helicase Activity	9/819	56/28171	3.162251e-05	0.0008110243	0.0006069955	D
1159	GO:0003678	DNA Helicase Activity	9/819	56/28171	3.162251e-05	0.0008110243	0.0006069955	G
1160	GO:0003678	DNA Helicase Activity	9/819	56/28171	3.162251e-05	0.0008110243	0.0006069955	M
1161	GO:0003678	DNA Helicase Activity	9/819	56/28171	3.162251e-05	0.0008110243	0.0006069955	M
1162	GO:0003678	DNA Helicase Activity	9/819	56/28171	3.162251e-05	0.0008110243	0.0006069955	M

```
tibble [329 × 13] (S3: tbl_df/tbl/data.frame)
 $ X                : int [1:329] 1157 1158 1159 1160 1161 1162 1163 1164 1165 1095
...
 $ ID               : chr [1:329] "GO:0003678" "GO:0003678" "GO:0003678" "GO:0003678"
...
 $ Description      : Factor w/ 14 levels "DNA Helicase Activity",..: 1 1 1 1 1 1 1 1
1 2 ...
 $ GeneRatio        : chr [1:329] "9/819" "9/819" "9/819" "9/819" ...
 $ BgRatio          : chr [1:329] "56/28171" "56/28171" "56/28171" "56/28171" ...
 $ pvalue           : num [1:329] 3.16e-05 3.16e-05 3.16e-05 3.16e-05 3.16e-05 ...
 $ p.adjust         : num [1:329] 0.000811 0.000811 0.000811 0.000811 0.000811 ...
 $ qvalue           : num [1:329] 0.000607 0.000607 0.000607 0.000607 0.000607 ...
 $ geneID           : chr [1:329] "Ddx3x" "D1Pas1" "G3bp1" "Mcm5" ...
 $ Count            : int [1:329] 9 9 9 9 9 9 9 9 11 ...
 $ GeneRatio_decimal: num [1:329] 0.011 0.011 0.011 0.011 0.011 ...
 $ BgRatio_decimal  : num [1:329] 0.00199 0.00199 0.00199 0.00199 0.00199 ...
 $ GO_class         : chr [1:329] "Molecular Functions" "Molecular Functions"
"Molecular Functions" "Molecular Functions" ...
```

DNA Helicase Activity · Calcium Ion Transmembrane Transporter Activity · UDP-Glycosyltransferase Activity · Histone Binding · Striated Muscle Tissue Development · Cellular Response To Oxidative Stress · Regulation Of Vasculature Development · Mitotic Cell Cycle Phase Transition · Regulation Of Actin Cytoskeleton Organization · Regulation Of Actin Filament-Based Process · Actin Binding · Striated Muscle Cell Differentiation · Positive Regulation Of Cell Projection Organization · Cell-Substrate Adhesion

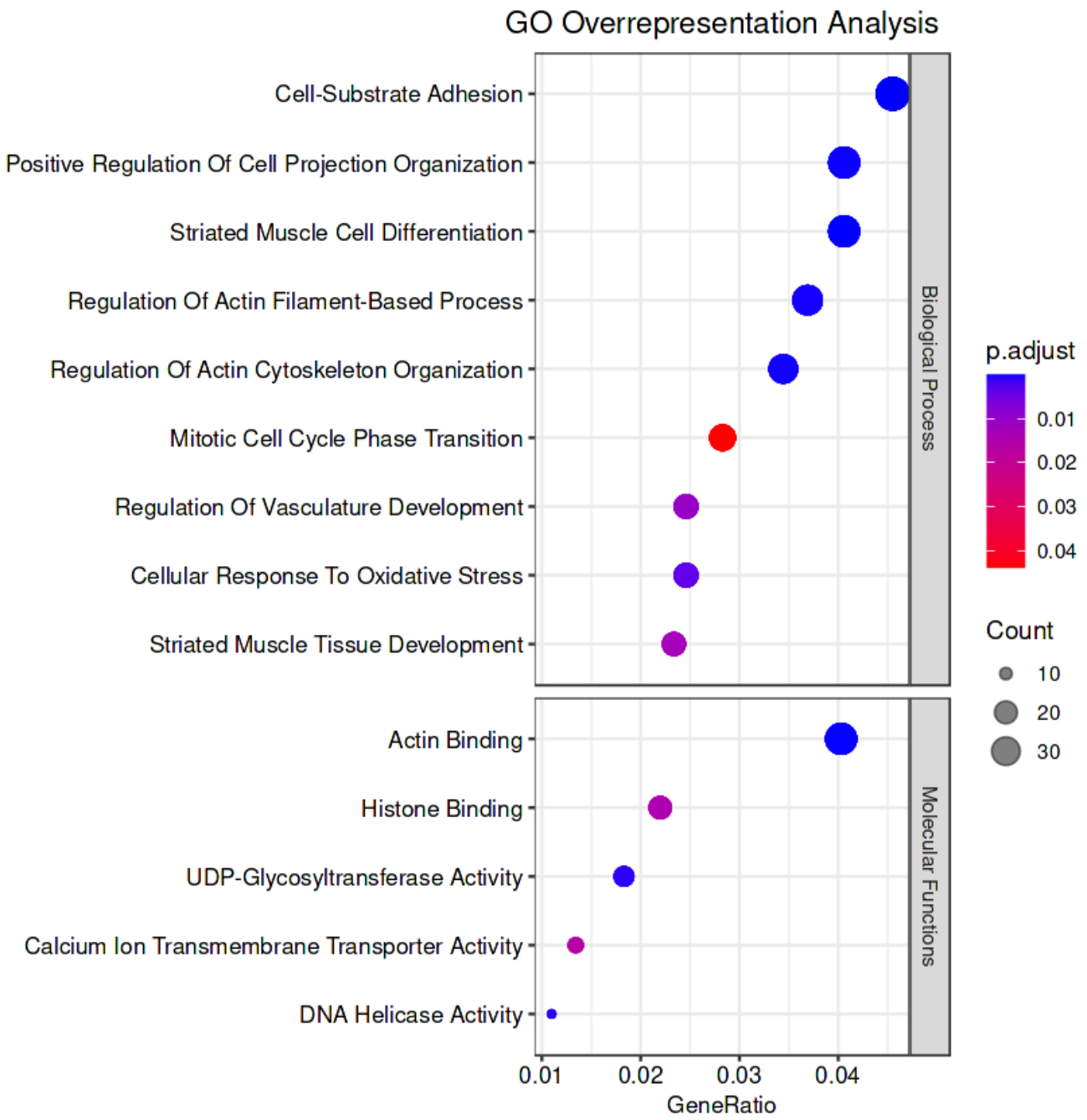
▶ **Levels**:

0.045510455104551

▼ Code

```r
# create dotplot for the GO terms with GeneRatio_decimal on x axis decreasing
        generatio , and GO term on y axis, color by p.adjust size of dot by Count

options(ggrepel.max.overlaps = 50)

    ggplot(data = (GO_all_figure  ) , # make ggplot object
    aes(x = GeneRatio_decimal, y = Description,  # x axis is GeneRatio_decimal, y
        axis is Description
    color = p.adjust, size = Count)) + # color by p.adjust, size of dot by Count
        geom_point(alpha = 0.5) + # add points with alpha 0.5
        scale_color_gradient(low = "blue", high = "red",guide=guide_colourbar(reverse
        = TRUE)) + # set color gradient to blue to red
        facet_grid(vars(GO_class), scales = "free",  space = "free_y") + # facet by
        GO_class with free scales and free y axis
        theme_bw() + # set theme to black and white
        labs(title = "GO Overrepresentation Analysis", x = "GeneRatio", y = "") + #
        set title and axis labels
        theme(plot.title = element_text(hjust = 0.5), axis.text = element_text(size =
        10), # set theme for plot title and axis text
              axis.title.x = element_text(size = 10), axis.title.y =
        element_text(size = 10),
              axis.text.x=element_text(colour="black"),
              axis.text.y=element_text(colour="black")) +
        NULL
    # Save dotplot as pdf file
    dev.copy(
    pdf,
    file = paste0("Results/GO/Figure_KO_vs_WT_GO_pathways.pdf"),
    width = 7,
    height = 6
    )
    dev.off ()
```

pdf: 4
png: 2



▼ Code

```
```

9 save data and session info

▼ Code

```
save.image("DEG_Cox7aKO_7dpi.RData")
```

▼ Code

```
load("DEG_Cox7aKO_7dpi.RData")
```

▼ Code

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 22.04.3 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=de_CH.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=de_CH.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=de_CH.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=de_CH.UTF-8 LC_IDENTIFICATION=C

time zone: Europe/Zurich
tzcode source: system (glibc)

attached base packages:
[1] stats4    stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] clusterProfiler_4.8.1       DESeq2_1.40.1
 [3] SummarizedExperiment_1.30.2 Biobase_2.60.0
 [5] MatrixGenerics_1.12.2       matrixStats_1.0.0
 [7] GenomicRanges_1.52.0        GenomeInfoDb_1.36.1
 [9] IRanges_2.34.1              S4Vectors_0.38.1
[11] BiocGenerics_0.46.0         biomaRt_2.56.1
[13] lubridate_1.9.2             forcats_1.0.0
[15] stringr_1.5.0               dplyr_1.1.2
[17] purrr_1.0.1                 readr_2.1.4
[19] tidyr_1.3.0                 tibble_3.2.1
[21] ggplot2_3.4.2               tidyverse_2.0.0

loaded via a namespace (and not attached):
  [1] RColorBrewer_1.1-3      jsonlite_1.8.7          magrittr_2.0.3
  [4] farver_2.1.1            zlibbioc_1.46.0         vctrs_0.6.3
  [7] memoise_2.0.1           RCurl_1.98-1.12         ggtree_3.8.0
 [10] base64enc_0.1-3         htmltools_0.5.5         S4Arrays_1.0.4
 [13] progress_1.2.2          curl_5.0.1              gridGraphics_0.5-1
 [16] plyr_1.8.8              cachem_1.0.8            uuid_1.1-0
 [19] igraph_1.5.0            lifecycle_1.0.3         pkgconfig_2.0.3
 [22] gson_0.1.0              Matrix_1.5-3            R6_2.5.1
 [25] fastmap_1.1.1           GenomeInfoDbData_1.2.10 digest_0.6.32
 [28] aplot_0.1.10            enrichplot_1.20.0       colorspace_2.1-0
 [31] patchwork_1.1.2         AnnotationDbi_1.62.1    RSQLite_2.3.1
 [34] filelock_1.0.2          fansi_1.0.4             timechange_0.2.0
 [37] httr_1.4.6              polyclip_1.10-4         compiler_4.3.2
 [40] bit64_4.0.5             withr_2.5.0             downloader_0.4
```

```
 [43] BiocParallel_1.34.2     viridis_0.6.3           DBI_1.1.3
 [46] ggforce_0.4.1           MASS_7.3-60             rappdirs_0.3.3
 [49] DelayedArray_0.26.6     HDO.db_0.99.1           tools_4.3.2
 [52] scatterpie_0.2.1        ape_5.7-1               glue_1.6.2
 [55] nlme_3.1-162            GOSemSim_2.26.0         shadowtext_0.1.2
 [58] grid_4.3.2              pbdZMQ_0.3-9            reshape2_1.4.4
 [61] fgsea_1.26.0            generics_0.1.3          gtable_0.3.3
 [64] tzdb_0.4.0              data.table_1.14.8       hms_1.1.3
 [67] tidygraph_1.2.3         xml2_1.3.4              utf8_1.2.3
 [70] XVector_0.40.0          ggrepel_0.9.3           pillar_1.9.0
 [73] yulab.utils_0.0.6       IRdisplay_1.1           splines_4.3.2
 [76] tweenr_2.0.2            treeio_1.24.1           BiocFileCache_2.8.0
 [79] lattice_0.22-5          bit_4.0.5               tidyselect_1.2.0
 [82] GO.db_3.17.0            locfit_1.5-9.8          Biostrings_2.68.1
 [85] gridExtra_2.3           graphlayouts_1.0.0      stringi_1.7.12
 [88] lazyeval_0.2.2          ggfun_0.0.9             evaluate_0.21
 [91] codetools_0.2-19        ggraph_2.1.0            qvalue_2.32.0
 [94] ggplotify_0.1.0         cli_3.6.1               IRkernel_1.3.2
 [97] repr_1.1.6              munsell_0.5.0           Rcpp_1.0.11
[100] dbplyr_2.3.2            png_0.1-8               XML_3.99-0.14
[103] parallel_4.3.2          blob_1.2.4              prettyunits_1.1.1
[106] DOSE_3.26.1             bitops_1.0-7            tidytree_0.4.2
[109] viridisLite_0.4.2       scales_1.2.1            crayon_1.5.2
[112] rlang_1.1.1             cowplot_1.1.1           fastmatch_1.1-3
[115] KEGGREST_1.40.0
```