

# **Comparative Transcriptomic Analysis of Endothelial Cells Exposed to Unidirectional Versus Bidirectional Flow**

Negar Vahdani

2024-06-01

## **Table of contents**

<b>1 Clean Memory, set working directory and load libraries</b>	<b>2</b>
<b>2 Import the data and prepare the metadata</b>	<b>2</b>
<b>3 Make the Deseq2 object and filter the low expressed genes</b>	<b>4</b>
<b>4 Run Desq2 analysis</b>	<b>5</b>
<b>5 Plot PCA</b>	<b>6</b>
<b>6 Save the differential expressed genes (DEGs) in each comparison</b>	<b>9</b>
<b>7 Add ENSMBL gene ids to the results</b>	<b>10</b>
<b>8 Plot upregulated and downregulated genes in volcano plots</b>	<b>12</b>
<b>9 GO analysis for DEGs</b>	<b>18</b>
9.1 GO plots for selected pathways . . . . .	26
<b>10 Plot Heatmap of top 500 genes across all samples</b>	<b>31</b>
10.1 Plot Heatmap of selected pathways . . . . .	33

## 1 Clean Memory, set working directory and load libraries

```
#Clear Memory and set the working directory
rm(list=ls())
gc()
setwd("/Users/negarvahdani/Deseq2")
dir.create("results", recursive = TRUE)

#load libraries
library(dplyr)
library(readxl)
library(DESeq2)
library(biomaRt)
library(ggplot2)
library(ggrepel)
library(ComplexHeatmap)
library(clusterProfiler)
library(tibble)
library(tidyverse)
library(matrixStats)
```

## 2 Import the data and prepare the metadata

```
#import data
counts110_trimmed <-
  read.delim("/Users/negarvahdani/gene_counts_ref110_trimmed.txt",
            sep = "\t", stringsAsFactor = FALSE, header= TRUE,
            fill = TRUE, row.names = 1)

#remove the columns and the rows of data containing sequencing information
counts110_trimmed_colremove <- counts110_trimmed %>% select(-c(1:5))

#remove the prefix and suffix of the the column name
colnames(counts110_trimmed_colremove) <-
  gsub("X.data.users.nvahdani.flow_project.bamsort110.output_trimmed.",
        "", colnames(counts110_trimmed_colremove))
colnames(counts110_trimmed_colremove) <-
  sub(".sorted.bam", "", colnames(counts110_trimmed_colremove))
```

```
#check the column names
head(counts110_trimmed_colremove)
```

	e3_24h_p	e3_24h_t	e3_4h_p	e3_4h_t	e4_24h_p	e4_24h_t	e4_4h_p
ENSG00000279928	0	0	0	0	0	0	0
ENSG00000228037	0	0	0	0	0	0	6
ENSG00000142611	0	1	0	0	0	0	0
ENSG00000284616	0	0	0	0	0	0	0
ENSG00000157911	292	447	115	495	296	386	377
ENSG00000269896	0	21	18	11	7	6	14
	e4_4h_t	e6_24h_p	e6_24h_t	e6_4h_p	e6_4h_t		
ENSG00000279928	0	2	0	0	0		
ENSG00000228037	0	0	0	0	0		
ENSG00000142611	0	0	0	5	0		
ENSG00000284616	0	0	0	0	0		
ENSG00000157911	407	377	467	383	388		
ENSG00000269896	23	14	12	6	4		

```
colnames(counts110_trimmed_colremove)
```

```
[1] "e3_24h_p" "e3_24h_t" "e3_4h_p" "e3_4h_t" "e4_24h_p" "e4_24h_t"
[7] "e4_4h_p" "e4_4h_t" "e6_24h_p" "e6_24h_t" "e6_4h_p" "e6_4h_t"
```

```
#load the metadata and check the column names
new_metadata <- read.delim("/Users/negarvahdani/new_metadata.txt",
                           sep = "\t", stringsAsFactor = FALSE, header= TRUE,
                           fill = TRUE, row.names = 1)

# check if the column names are the same as the row names of the metadata
colnames(counts110_trimmed_colremove) %in% rownames(new_metadata)
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
# check if the column names are the same as the row names of the metadata
identical(colnames(counts110_trimmed_colremove), rownames(new_metadata))
```

```
[1] TRUE
```

```

#create a new column with the flow profile and time
new_metadata$flow_time <- paste0(new_metadata$flow_profile, "_",
                                new_metadata$time)

#convert the columns to a factor
new_metadata$flow_time <- factor(new_metadata$flow_time)
new_metadata$flow_profile <- factor(new_metadata$flow_profile)
head(new_metadata)

```

	experiment	time	flow_profile	flow_time
e3_24h_p	3	24	unidirectional	unidirectional_24
e3_24h_t	3	24	bidirectional	bidirectional_24
e3_4h_p	3	4	unidirectional	unidirectional_4
e3_4h_t	3	4	bidirectional	bidirectional_4
e4_24h_p	4	24	unidirectional	unidirectional_24
e4_24h_t	4	24	bidirectional	bidirectional_24

### 3 Make the Deseq2 object and filter the low expressed genes

```

#make DESeq2 object
dds_flow_time <-
  DESeqDataSetFromMatrix(countData = counts110_trimmed_colremove,
                        colData = new_metadata,
                        design= ~ flow_time)

#remove the low expressed genes and check the no. of the removed columns
#set threshold and no. of samples
threshold <- 5
min_samples <- 3

#calculate the sum of counts across all samples for each gene
gene_counts <- rowSums(counts(dds_flow_time))

#create a logical vector indicating whether each gene meets the criteria
keep_genes <- gene_counts >= threshold

#subset the DESeqDataSet
dds_filtered <- dds_flow_time[keep_genes,]

#DESeq2 analysis
dds_flow_time <- DESeq(dds_flow_time)

```

```
estimating size factors  
  
estimating dispersions  
  
gene-wise dispersion estimates  
  
mean-dispersion relationship  
  
final dispersion estimates  
  
fitting model and testing
```

```
#check the lists the coefficients  
print(resultsNames(dds_flow_time))
```

```
[1] "Intercept"  
[2] "flow_time_bidirectional_4_vs_bidirectional_24"  
[3] "flow_time_unidirectional_24_vs_bidirectional_24"  
[4] "flow_time_unidirectional_4_vs_bidirectional_24"
```

## 4 Run Desq2 analysis

```
#extract the results for the comparison of the flow profile and time  
#tilter 24h vs pump 24h  
res_uni_bidirectional_24 <-  
  lfcShrink(dds_flow_time,parallel = TRUE,  
            contrast=c("flow_time","bidirectional_24","unidirectional_24"),  
            type="ashr")
```

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>

```
#convert the results to a data frame  
res_uni_bidirectional_24 <- as.data.frame(res_uni_bidirectional_24 )  
head(res_uni_bidirectional_24) #check the first rows
```

	baseMean	log2FoldChange	lfcSE	pvalue	padj
ENSG00000279928	0.1400055	-0.012117556	0.3593873	0.6798435	NA
ENSG00000228037	0.4547908	0.000000000	0.3580922	1.0000000	NA
ENSG00000142611	0.4967602	0.006178389	0.3584660	0.8319448	NA
ENSG00000284616	0.0000000	0.000000000	0.3692715	NA	NA
ENSG00000157911	354.5700642	0.158482844	0.1867622	0.1225302	0.4983356
ENSG00000269896	11.8541232	0.058353799	0.3059987	0.4419554	0.8178249

```
#tilter 4h vs pump 4h
res_uni_bidirectional_4 <-
  lfcShrink(dds_flow_time,parallel = TRUE,
             contrast=c("flow_time", "bidirectional_4","unidirectional_4"),
             type="ashr")
```

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>

```
#convert the results to a data frame and check the first rows
res_uni_bidirectional_4 <- as.data.frame(res_uni_bidirectional_4 )
head(res_uni_bidirectional_4)
```

	baseMean	log2FoldChange	lfcSE	pvalue	padj
ENSG00000279928	0.1400055	0.000000000	0.2344532	1.0000000	NA
ENSG00000228037	0.4547908	-0.009906163	0.2362416	0.4399685	NA
ENSG00000142611	0.4967602	-0.009524368	0.2361023	0.4571412	NA
ENSG00000284616	0.0000000	0.000000000	0.2380879	NA	NA
ENSG00000157911	354.5700642	0.114227151	0.1988798	0.0852675	0.5495606
ENSG00000269896	11.8541232	-0.018984809	0.2063680	0.6158550	0.9694326

## 5 Plot PCA

```
## PCA function

PCA <- function(mat,color_pca="",shape_pca= "", label_pca= "",
                 save_plot= "no", name_of_plot= "PCA", comp1=1,
                 comp2=2, pdf_width=12, pdf_height=12){
  #Get the differential expressed values from the comparrison interested,
```

```

#extract the normalized values from the assay of vsd and plot them.
#Giving condition and group from your design table

#1. Extract the counts.
dt <- mat

#2. Perform pca
pca_dt <- prcomp(t(dt))
cat("PCA running...\n")
# Sys.sleep(0.2)

#3. Extract percentVar data.
percentVar_dt <- pca_dt$sdev^2/sum(pca_dt$sdev^2)
cat("Percents calculated...\n")
# Sys.sleep(0.2)

#4. Create the new dataframe to plot.
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")
# Sys.sleep(0.2)

#5. Plot it
cat("Plotting...\n")
# Sys.sleep(0.2)
print(save_plot)
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 = 5) +
    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")) +
# coord_fixed()+
NULL
}
if (save_plot== "yes"){

  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 = 5) +
  xlab(paste0("PC", comp1,": ", round(percentVar_dt[comp1] * 100),
              "% variance")) +
  ylab(paste0("PC",comp2,": ", round(percentVar_dt[comp2] * 100),
              "% variance")) +
# coord_fixed()+
NULL
print(pca_p)
dev.copy(pdf, paste0(name_of_plot,".pdf"),
        width = pdf_width,height = pdf_height)
dev.off()
}
# Sys.sleep(0.2)
cat("Done")
print(pca_p)

#return(pca_p)
}

#run the PCA function
PCA(counts110_trimmed_colremove, color_pca = factor(new_metadata$time),
    shape_pca = factor(new_metadata$flow_profile), save_plot = "no",
    name_of_plot = "PCA_nolog")

```

PCA running...  
Percents calculated...

```
Data frame built...
```

```
Plotting...
```

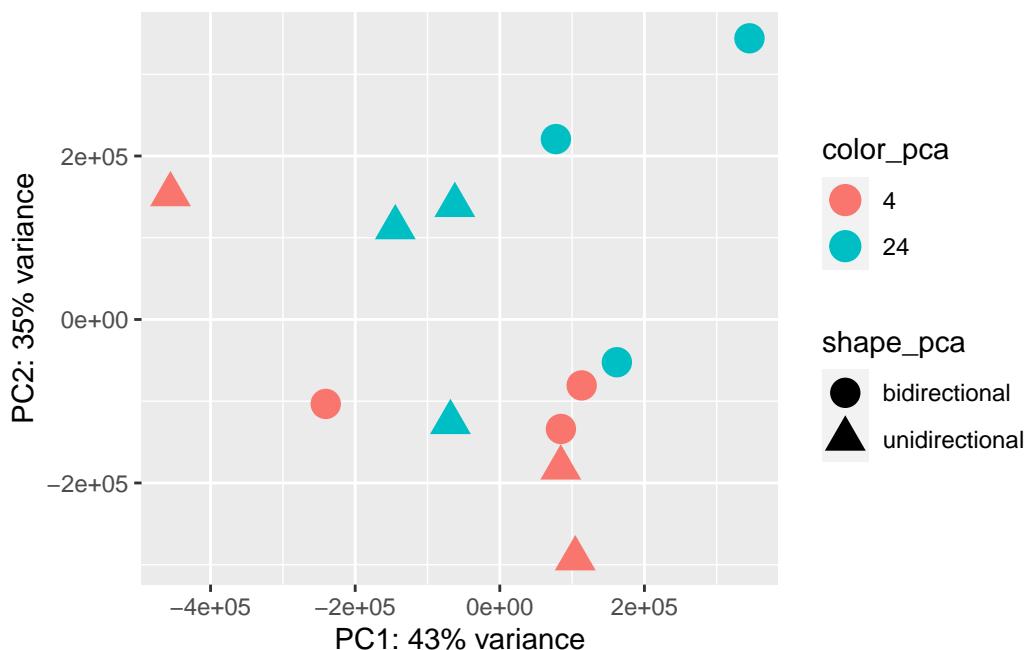
```
[1] "no"
```

```
Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
```

```
i Please use tidy evaluation idioms with `aes()``.
```

```
i See also `vignette("ggplot2-in-packages")` for more information.
```

```
Done
```



## 6 Save the differential expressed genes (DEGs) in each comparison

```
#put 1 instead of the NAs because it causes problems in the volcano plot
res_uni_bidirectional_24$padj[is.na(res_uni_bidirectional_24$padj)] <- 1
res_uni_bidirectional_4$padj[is.na(res_uni_bidirectional_4$padj)] <- 1

#save the differential expressed genes in each comparison
res_uni_bidirectional_24_padj0.05 <- subset(res_uni_bidirectional_24,
                                             padj < 0.05)
```

```

write.csv(as.data.frame(res_uni_bidirectional_24_padj0.05),
          file="unidirectional_vs_bidirectional_24_0.05_nopb.csv")
head(res_uni_bidirectional_24_padj0.05)

```

	baseMean	log2FoldChange	lfcSE	pvalue	padj
ENSG00000130766	745.104354	-0.6042284	0.3080410	5.534883e-04	1.943932e-02
ENSG00000092853	1317.929292	1.4715801	0.3163299	2.375180e-07	3.497268e-05
ENSG00000236098	9.162552	-0.7006854	0.9759735	4.108929e-04	1.554073e-02
ENSG00000023909	2084.137196	-1.1834854	0.2824740	7.122420e-07	9.086322e-05
ENSG00000198198	1489.214895	-0.4057124	0.1879218	1.424821e-03	3.655861e-02
ENSG00000117525	308.979163	-2.6614545	0.4370338	2.789364e-12	1.561144e-09

```

res_uni_bidirectional_4_padj0.05 <- subset(res_uni_bidirectional_4,
                                              padj < 0.05)
write.csv(as.data.frame(res_uni_bidirectional_4_padj0.05),
          file="unidirectional_vs_bidirectional_4_0.05_nopb.csv")
head(res_uni_bidirectional_4_padj0.05)

```

	baseMean	log2FoldChange	lfcSE	pvalue	padj
ENSG00000132881	57.76324	0.8568570	0.5505004	2.173710e-04	1.487556e-02
ENSG00000162407	1978.62537	-0.5549861	0.2070021	3.110924e-04	1.865012e-02
ENSG00000127124	368.14958	-1.2764537	0.3464764	2.095402e-07	7.600025e-05
ENSG00000023909	2084.13720	-1.4496330	0.2912663	3.083979e-10	6.114980e-07
ENSG00000143315	275.00192	0.7384807	0.3265591	2.064510e-04	1.426281e-02
ENSG00000132763	117.11548	0.6430754	0.4443382	8.166160e-04	3.859109e-02

## 7 Add ENSEMBL gene ids to the results

```

#convert gene ids to names using biomart
symbol_to_ensembl_human <- function(){
  require(biomart) # load biomart package
  ensembl <- useMart(biomart = "ensembl", dataset = "hsapiens_gene_ensembl",
                      host = "https://www.ensembl.org")
  # use the ensembl mart
  # query biomart
  annotations <- getBM(attributes = c("ensembl_gene_id","hgnc_symbol"),
                        # get the ensembl gene id and the hgnc symbol
                        mart = ensembl)# the ensembl mart

```

```

    return(annotations) # return the results
}

#get the gene ids and convert to data frame
gene_id <- symbol_to_ensml_human()
gene_id_df <- as.data.frame(gene_id)
head(gene_id_df)

ensembl_gene_id hgnc_symbol
1 ENSG00000210049      MT-TF
2 ENSG00000211459      MT-RNR1
3 ENSG00000210077      MT-TV
4 ENSG00000210082      MT-RNR2
5 ENSG00000209082      MT-TL1
6 ENSG00000198888      MT-ND1

#add the ensembl gene id as a column, merge with the data and save the csv
res_uni_bidirectional_24$ensembl_gene_id <- rownames(res_uni_bidirectional_24)
res_uni_bidirectional_4$ensembl_gene_id <- rownames(res_uni_bidirectional_4)
res_uni_bidirectional_24merged <- merge(res_uni_bidirectional_24,
                                         gene_id_df, by.x = "ensembl_gene_id",
                                         all.x=TRUE)
res_uni_bidirectional_4hmerged <- merge(res_uni_bidirectional_4,
                                         gene_id_df, by.x = "ensembl_gene_id",
                                         all.x=TRUE)

res_uni_bidirectional_24merged_padj0.05 <-
  subset(res_uni_bidirectional_24merged, padj < 0.05)
write.csv(as.data.frame(res_uni_bidirectional_24merged_padj0.05),
          file="unidirectional_vs_bidirectional_24_0.05_nopb_withsymbol.csv")
head(res_uni_bidirectional_24merged_padj0.05)

ensembl_gene_id  baseMean log2FoldChange      lfcSE      pvalue
9   ENSG00000001084  502.9601     -0.5804792 0.3502721 1.356347e-03
13  ENSG00000001497  857.7391      0.5631373 0.2315451 1.766710e-04
40  ENSG00000003402 7649.1939     -0.4074632 0.1312337 1.380003e-04
82  ENSG00000005108  624.6087      1.1664631 0.2512031 1.196775e-07
85  ENSG00000005187  167.0716      2.2175175 0.3117089 4.535044e-14
137 ENSG00000006459  851.4950     -0.7895024 0.3706497 2.368339e-04

      padj hgnc_symbol
9   3.550911e-02       GCLC

```

```

13 8.356567e-03      LAS1L
40 6.840872e-03      CFLAR
82 1.977528e-05      THSD7A
85 4.628413e-11      ACSM3
137 1.040271e-02     KDM7A

res_uni_bidirectional_4hmerged_padj0.05 <-
  subset(res_uni_bidirectional_4hmerged, padj < 0.05)
write.csv(as.data.frame(res_uni_bidirectional_4hmerged_padj0.05)
  , file="unidrectional_vs_bidirectional_4_0.05_nopb_withsymbol.csv")
head(res_uni_bidirectional_4hmerged_padj0.05)

  ensembl_gene_id  baseMean log2FoldChange      lfcSE      pvalue
20  ENSG00000002016  168.1894    0.8707179 0.3757943 8.314925e-05
67  ENSG00000004866  604.4218   -0.5734751 0.2106566 2.636006e-04
135 ENSG00000006451  8337.3423   0.4658322 0.2073872 9.509839e-04
137 ENSG00000006459  851.4950   -1.0084843 0.3069049 3.948719e-06
152 ENSG00000006704  666.1753   -0.7424482 0.3083939 1.627811e-04
219 ENSG00000008405  751.4543   -0.9217799 0.2374767 1.045986e-06
               padj hgnc_symbol
20  0.0072670439      RAD52
67  0.0169412009      ST7
135 0.0423870125      RALA
137 0.0006739766      KDM7A
152 0.0119192070      GTF2IRD1
219 0.0002529193      CRY1

```

## 8 Plot upregulated and downregulated genes in volcano plots

```

draw_volcano<- function(fileinput, title) {

  require(ggplot2)
  require(ggrepel)
  require(clusterProfiler)
  require(tidyverse)

  # read input file
  # drawing plots

```

```

ggplot(data =fileinput , aes(x = log2FoldChange, y = -log10(padj))) +
  # draw lines
  geom_hline(yintercept = -log10(0.05), linetype = "dashed",
             col = "chartreuse4") +
  geom_hline(yintercept = -log10(0.05), linetype = "dotted",
             col = "darkgoldenrod") +
  geom_vline(xintercept = 0, linetype = "dashed")+
  # draw points
  geom_point(x = fileinput$log2FoldChange, y = -log10(fileinput$padj),
             alpha = 0.5,size = 2,color="grey51") +
  # draw coloured points
  geom_point(data = fileinput[which(fileinput$padj < 0.05 &
                                    fileinput$log2FoldChange < -0.585),],
             aes(x=log2FoldChange, y = -log10(padj)), shape = 21,
             color = "royalblue2", fill = "royalblue2",
             alpha = 0.5, size = 2) +
  geom_point(data = fileinput[which(fileinput$padj < 0.05 &
                                    fileinput$log2FoldChange <= - 1),],
             aes(x=log2FoldChange, y = -log10(padj)),
             shape = 21, color = "royalblue4", fill = "royalblue4",
             alpha = 0.5, size = 2) +
  geom_point(data = fileinput[which(fileinput$padj < 0.05 &
                                    fileinput$log2FoldChange > 0.585),],
             aes(x=log2FoldChange, y = -log10(padj)),
             shape = 21, color = "red2", fill = "red2",
             alpha = 0.5, size = 2) +
  geom_point(data = fileinput[which(fileinput$padj < 0.05 &
                                    fileinput$log2FoldChange >= 1),],
             aes(x=log2FoldChange, y = -log10(padj)),
             shape = 21, color = "red4", fill = "red4",
             alpha = 0.5, size = 2) +
  # x axis scale
  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))) +
  xlab("log2FoldChange") + #ylab("-Log10(p.value)") +
  scale_y_continuous(breaks = seq(0,round(-log10(min(fileinput$padj))+1),
                                 by = 4),
                     limits = c(0,round(-log10(min(fileinput$padj))+1))) +

```

```

ylab("-Log10(pAdjusted)") + # ylab("-log10(p.value)")+

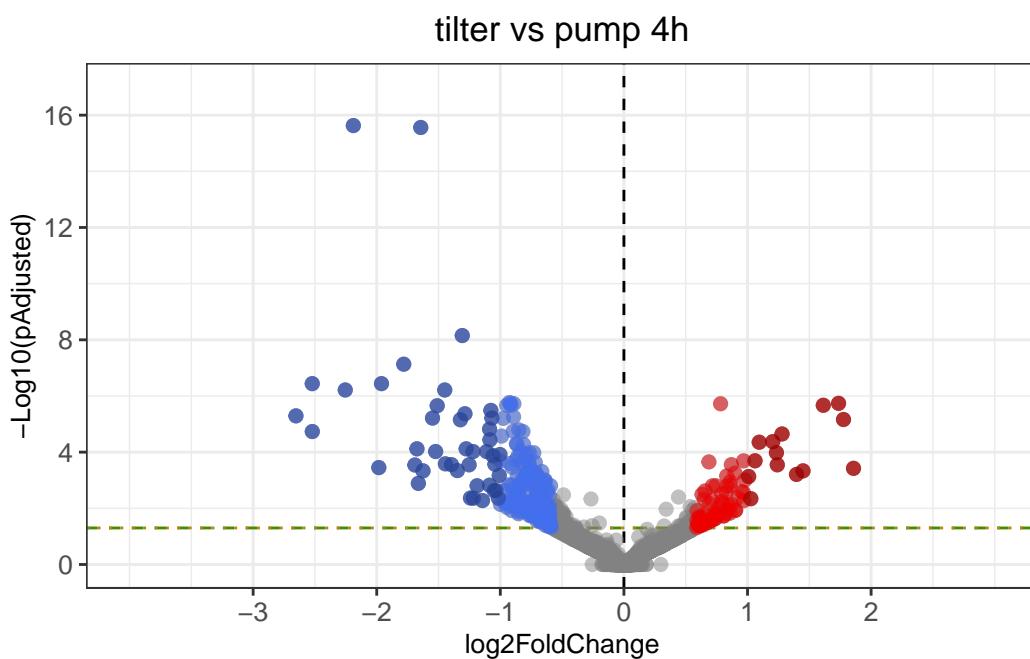
# set title
ggtitle(title)+

# x and y axis limits
# black and white theme
theme_bw() +
# center title
theme(plot.title = element_text(hjust = 0.5),
      axis.text = element_text(size = 10),
      axis.title.x = element_text(size = 10),
      axis.title.y = element_text(size = 10))

}

##volcano plot : tilter vs pump at 4h
a = draw_volcano(res_uni_bidirectional_4hmerged, "tilter vs pump 4h")
a

```



```

# add labels: make a list of top 25 genes and bottom 25 genes
options(ggrepel.max.overlaps=50) # set the maximum number of overlaps
label_set_top <- res_uni_bidirectional_4hmerged %>%

```

```
    arrange(desc(log2FoldChange)) %>% head(25) %>% pull(hgnc_symbol)
label_set_top
```

```
[1] ""
[7] "SH2D3C"
[13] ""
[19] "GIMAP8"
[25] "MAP3K12"
```

```
"NRROS"      "SPAAR"       "PDE7B"       "DLL4"        "NAGS"
"BMP4"        "GIMAP1"      "SOX7"        "CAVIN2"      "GIMAP6"
"DHODH"       "IMP3"        "NUDT18"      "HTR1B"       "PTP4A2P2"
"TMEM144"     "LPP-AS2"     "EHD3"        "SPHK2"       "RAD52"
```

```
label_set_bottom <- res_uni_bidirectional_4hmerged %>%
  arrange(desc(log2FoldChange)) %>% tail(25) %>% pull(hgnc_symbol)
label_set_bottom
```

```
[1] "PITPN1C"
[7] "NOG"
[13] "GLI2"
[19] "ATF3"
[25] "IL11"
```

```
"HIVEP3"      "TXNRD1"      "GABARAPL1"   "CHSY3"       "SMARCD3"
"KLF4"        "GCLM"        "EGFR"        "EGR1"        "NR4A1"
"MOAP1"       "NR4A3"       "HEY1"        "F2RL3"      "ZFPM1"
"DLX2"        "GLIS3"       "OSGIN1"      "HMOX1"      ""
"NRROS"       "SPAAR"       "PDE7B"       "DLL4"        "
```

```
# combine the top and bottom genes altered genes
label_set <- c(label_set_bottom, label_set_top)
label_set
```

```
[1] "PITPN1C"
[7] "NOG"
[13] "GLI2"
[19] "ATF3"
[25] "IL11"
[31] "NAGS"
[37] "GIMAP6"
[43] "PTP4A2P2"
[49] "RAD52"
```

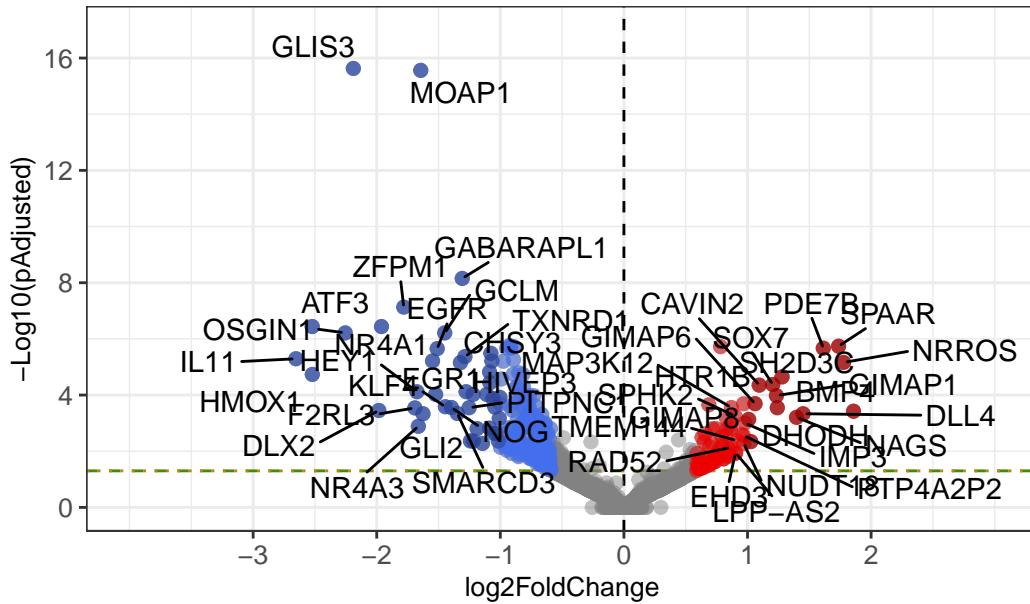
```
"HIVEP3"      "TXNRD1"      "GABARAPL1"   "CHSY3"       "SMARCD3"
"KLF4"        "GCLM"        "EGFR"        "EGR1"        "NR4A1"
"MOAP1"       "NR4A3"       "HEY1"        "F2RL3"      "ZFPM1"
"DLX2"        "GLIS3"       "OSGIN1"      "HMOX1"      ""
"NRROS"       "SPAAR"       "PDE7B"       "DLL4"        "
```

```
# add the labels to the plot
a2= a +
  geom_text_repel(data = res_uni_bidirectional_4hmerged
                  [res_uni_bidirectional_4hmerged$hgnc_symbol %in% label_set,],
                  aes(label=res_uni_bidirectional_4hmerged
                      [res_uni_bidirectional_4hmerged$hgnc_symbol %in%
                      label_set,]$hgnc_symbol),
```

)

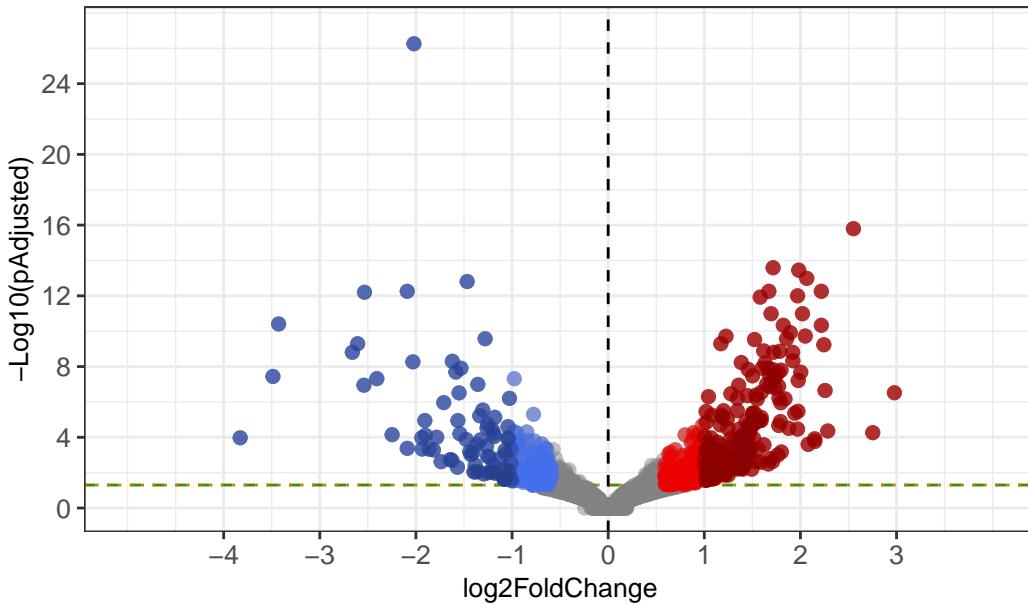
a2

tilter vs pump 4h



```
##volcano plot : tilter vs pump at 24h
b = draw_volcano(res_uni_bidirectional_24merged,"tilter vs pump 24h")
b
```

tilter vs pump 24h



```
# add labels: make a list of top 25 genes and bottom 25 genes
options(ggrepel.max.overlaps=50) # set the maximum number of overlaps
label_set_top <- res_uni_bidirectional_24merged %>%
  arrange(desc(log2FoldChange)) %>% head(25) %>% pull(hgnc_symbol)
label_set_top
```

```
[1] "BMPER"      ""          "SELE"       "VCAM1"      "CENPA"      "CDCA3"      "ACSM3"
[8] "CDCA7"      "SLFN13"     "SEPTIN4"    "E2F2"       "GINS2"      "MCM10"      "CDCA8"
[15] "KIF18B"     "HTR1B"      "AURKB"      "ZNF367"     "GTSE1"      "CXCL5"      "HMMR"
[22] "RRM2"       "TROAP"      "CDC45"      "EME1"
```

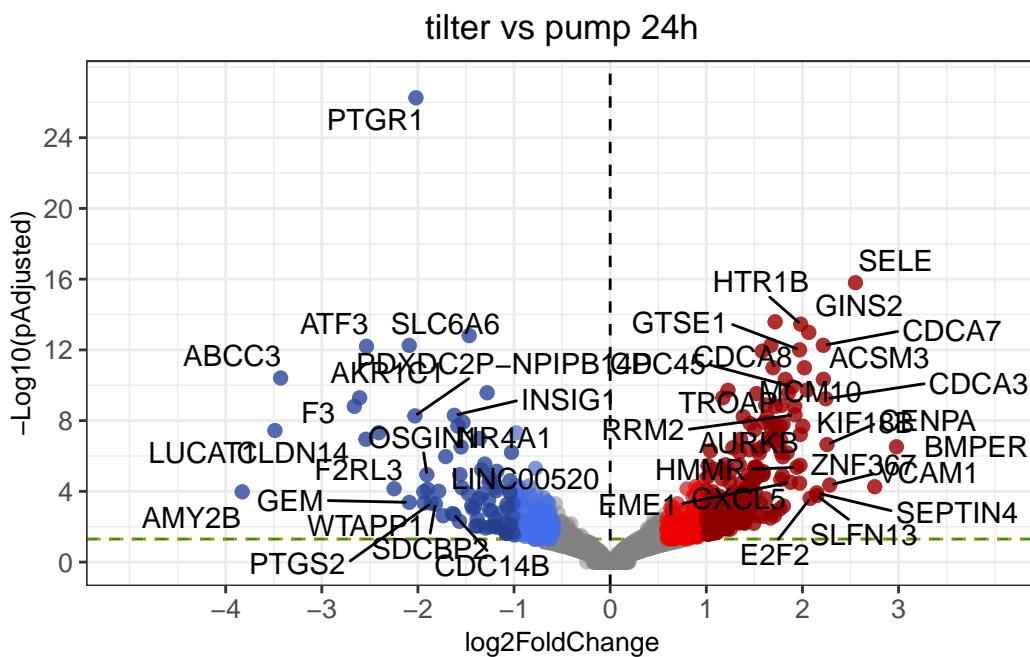
```
label_set_bottom <- res_uni_bidirectional_24merged %>%
  arrange(desc(log2FoldChange)) %>% tail(25) %>% pull(hgnc_symbol)
label_set_bottom
```

```
[1] "INSIG1"       "CDC14B"      """           "NR4A1"
[5] ""             "LINC00520"   "SDCBP2"      "PTGS2"
[9] "F2RL3"        "OSGIN1"      "WTAPP1"      ""
[13] "PTGR1"        "PDXDC2P-NPIPBP14P" "GEM"         "SLC6A6"
[17] ""             ""             "ATF3"        "CLDN14"
[21] "AKR1C1"       "F3"           "ABCC3"      "LUCAT1"
[25] "AMY2B"
```

```

# combine the top and bottom genes altered genes
label_set <- c(label_set_bottom, label_set_top)
# add the labels to the plot
b2= b +
  geom_text_repel(data = res_uni_bidirectional_24merged
                  [res_uni_bidirectional_24merged$hgnc_symbol %in% label_set,],
                  aes(label=res_uni_bidirectional_24merged
                      [res_uni_bidirectional_24merged$hgnc_symbol %in%
                       label_set,]$hgnc_symbol),
  )
b2

```



## 9 GO analysis for DEGs

```

##gene ontology

GO_function <- function(gene_list, pval  = 0.05, onto= "MF",
                         prefix="", org="human"){
  require(clusterProfiler)
  require(ReactomePA)
}

```

```

if(org=="mouse"){
  orgdb <- "org.Mm.eg.db"
  org_reactome <- "mouse"
}else if(org=="human"){
  orgdb <- "org.Hs.eg.db"
  org_reactome <- "human"
}else{
  message("Please enter a valid organism (mouse or human)")
}

if(onto %in% c("MF", "CC", "BP")){

  compGO <- enrichGO(gene = gene_list, pvalueCutoff  = pval,
                      keyType = "SYMBOL",
                      pAdjustMethod = "BH", OrgDb = orgdb, ont = onto)
}else if(onto=="reactome"){
  gene_list <- bitr(gene_list, fromType = "SYMBOL",
                     toType = "ENTREZID", OrgDb = orgdb) #SYMBOL
  gene_list <- gene_list$ENTREZID
  compGO <- enrichPathway(gene = gene_list, pvalueCutoff  = 0.05,
                           organism= org_reactome,      readable = TRUE)

}else{
  message("Please enter a valid GO term")
}

if(is.null(compGO)){
  message(paste0("No GO:", onto, " obtained"))
  message(paste0("*****"))
  message(paste0("\n"))

}else {

  compGO_df <- as.data.frame(compGO)
  compGO_df$GeneRatio_decimal <- compGO_df$GeneRatio
  compGO_df$GeneRatio_decimal <- sapply(compGO_df$GeneRatio_decimal,
                                         function(x)
                                           (eval(parse(text
                                         = as.character(x)))))

}

```

```

compGO_df$BgRatio_decimal <- compGO_df$BgRatio
compGO_df$BgRatio_decimal <- sapply(compGO_df$BgRatio_decimal,
                                      function(x)
                                         (eval(parse(text = as.character(x)))))
compGO_df <- compGO_df %>% tidyr::separate_rows(geneID,
                                                    sep = "/",
                                                    convert = FALSE) %>%
  arrange(desc(GeneRatio_decimal))
compGO_df %>% head

if(nrow(compGO_df)==0){
  message(paste0("No GO:",onto, " obtained"))
  message(paste0("*****"))
  message(paste0("\n"))

} else{

  write.csv(compGO_df, paste0(prefix,"_GO_",onto, "_pathways.csv"))

  full_name= switch(onto,
                    MF= "Molecular Function",
                    CC= "Cellular Components",
                    BP= "Biological Pathways"
  )

  print(dotplot(compGO, showCategory = 15,
                title = paste0("GO Pathway Enrichment Analysis \n",
                               full_name),
                font.size = 6))
  dev.copy(
    pdf,
    file = paste0(prefix,"_GO_",onto, "_pathways.pdf"),
    width = 12,
    height = 16
  )
  dev.off ()

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

```

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

}

}

}

# run the GO function : tilter vs pump at 4h
dir.create("results/GO_analysis/analysis_withoutpb", recursive = TRUE)
```

Warning in dir.create("results/GO\_analysis/analysis\_withoutpb", recursive = TRUE): 'results/GO\_analysis/analysis\_withoutpb' already exists

```
GO_function((res_uni_bidirectional_4hmerged %>% filter(padj<0.05)
             %>% pull(hgnc_symbol)), onto="MF",
             prefix="results/GO_analysis/pump_tilter_4h" )
```

Loading required package: ReactomePA

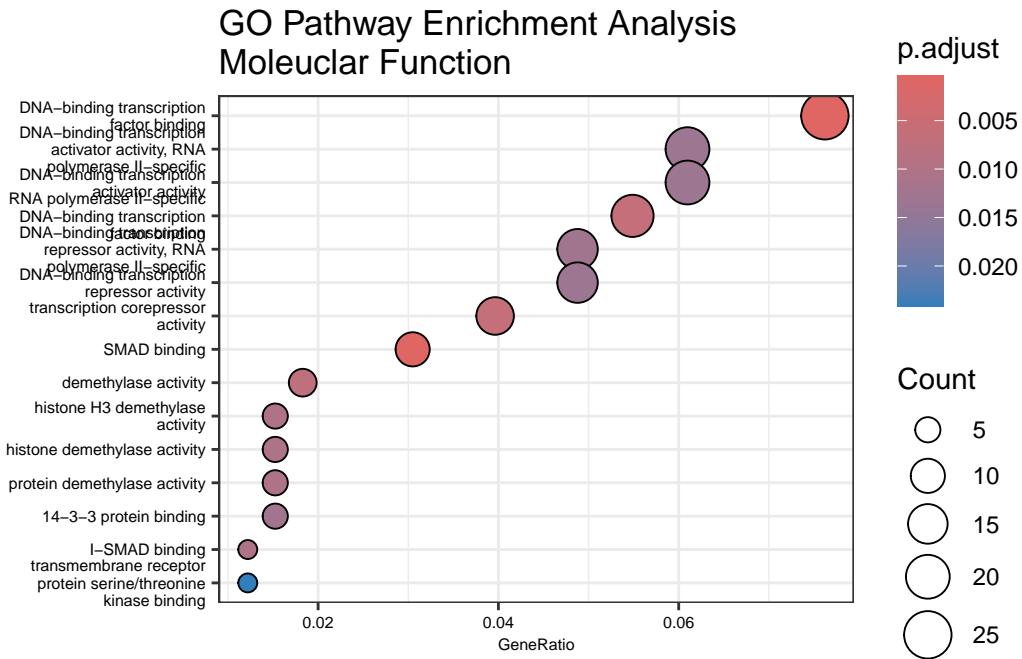
ReactomePA v1.46.0 For help: <https://yulab-smu.top/biomedical-knowledge-mining-book/>

If you use ReactomePA in published research, please cite:

Guangchuang Yu, Qing-Yu He. ReactomePA: an R/Bioconductor package for reactome pathway analysis.

Pathway analysis GO:MF done

\*\*\*\*\*



```
GO_function((res_uni_bidirectional_4hmerged %>% filter(padj<0.05)
             %>% pull(hgnc_symbol)), onto="CC",
             prefix="results/GO_analysis/pump_tilter_4h",org = "human" )
```

No GO:CC obtained

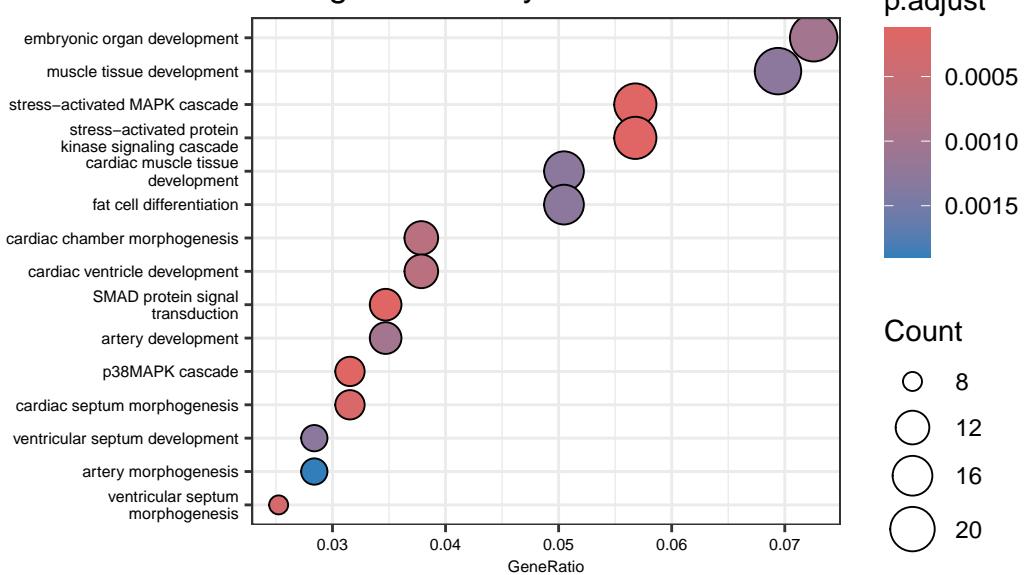
\*\*\*\*\*

```
GO_function((res_uni_bidirectional_4hmerged %>% filter(padj<0.05)
             %>% pull(hgnc_symbol)), onto="BP",
             prefix="results/GO_analysis/pump_tiliter_4h" )
```

Pathway analysis GO:BP done

\*\*\*\*\*

## GO Pathway Enrichment Analysis Biological Pathways

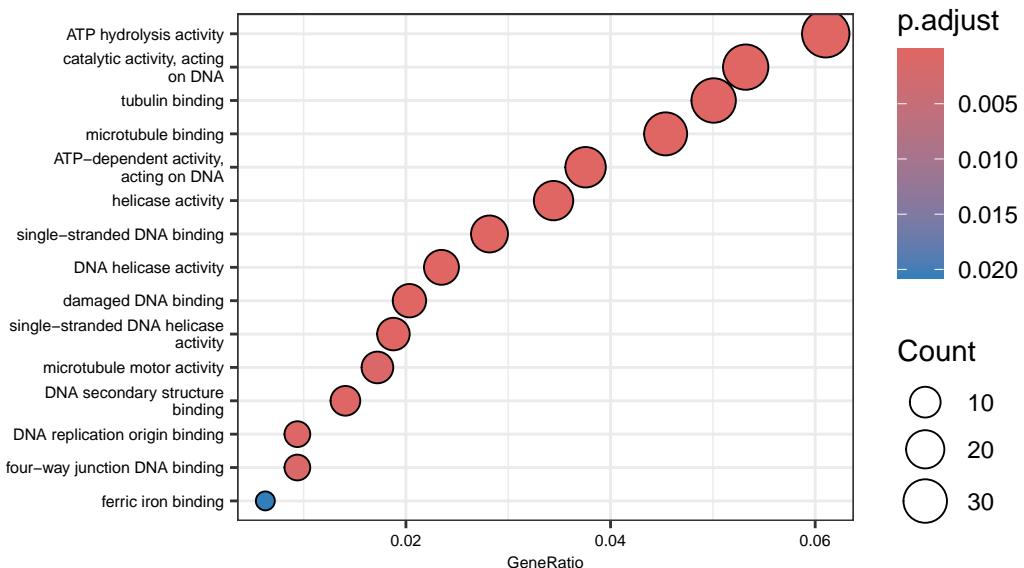


```
##volcano plot : tilter vs pump at 24h
GO_function((res_uni_bidirectional_24merged %>% filter(padj<0.05)
             %>% pull(hgnc_symbol)), onto="MF",
             prefix="results/GO_analysis/analysis_withoutpb/pump_tilter_24h" )
```

Pathway analysis GO:MF done

\*\*\*\*\*

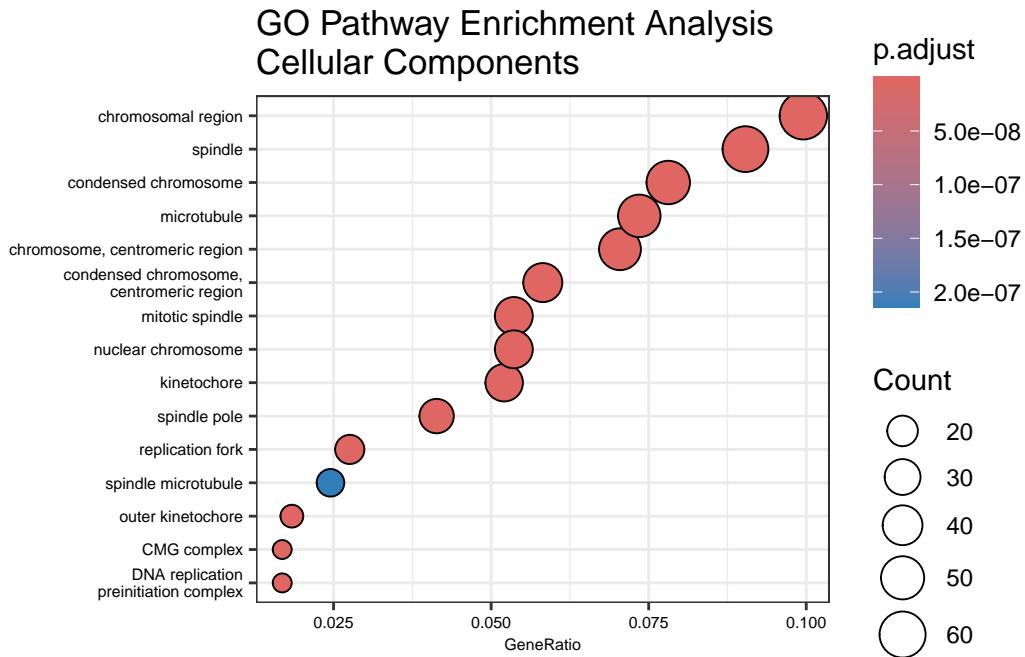
## GO Pathway Enrichment Analysis Molecular Function



```
GO_function((res_uni_bidirectional_24merged %>% filter(padj<0.05)
             %>% pull(hgnc_symbol)), onto="CC",
            prefix="results/GO_analysis/analysis_withoutpb/pump_tilfter_24h",
            org = "human" )
```

Pathway analysis GO:CC done

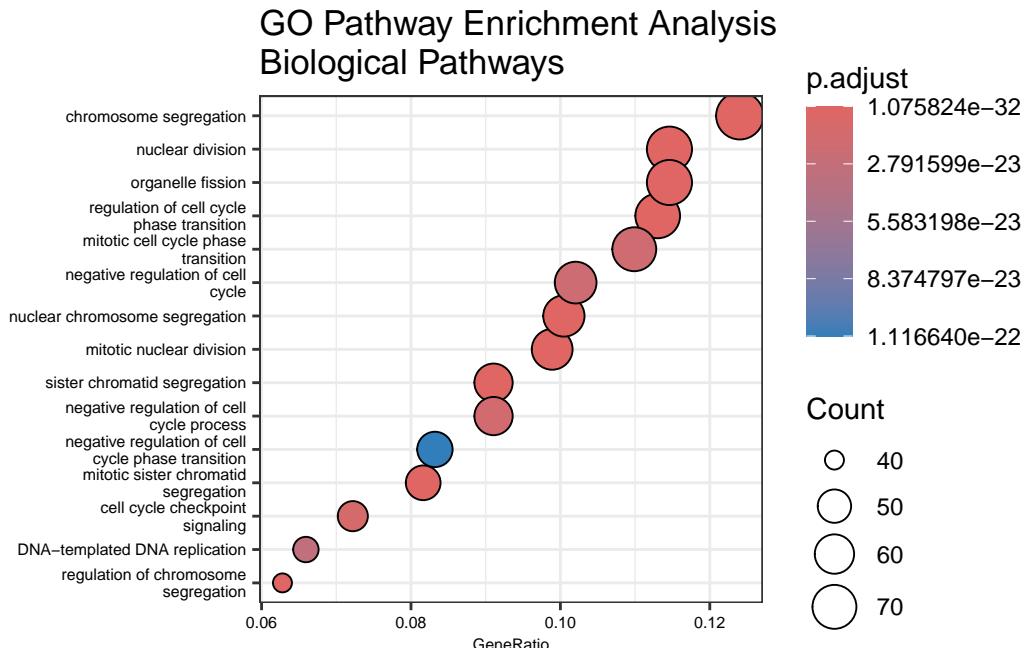
\*\*\*\*\*



```
GO_function((res_uni_bidirectional_24merged %>% filter(padj<0.05)
             %>% pull(hgnc_symbol)), onto="BP",
            prefix="results/GO_analysis/analysis_withoutpb/pump_tilfter_24h" )
```

Pathway analysis GO:BP done

\*\*\*\*\*



## 9.1 GO plots for selected pathways

```
# GO analysis for the selected pathways for tilter and pump at 24h time point
# read the GO analysis results obtained in the previous step
GO_res_uni_bidirectional_24_ <- read.csv("pump_tilter_24h_GO_BP_pathways.csv")
head(GO_res_uni_bidirectional_24_)
```

X	ID	Description	GeneRatio	BgRatio	pvalue
1	1	GO:0007059 chromosome segregation	79/637	424/18870	2.302705e-36
2	2	GO:0007059 chromosome segregation	79/637	424/18870	2.302705e-36
3	3	GO:0007059 chromosome segregation	79/637	424/18870	2.302705e-36
4	4	GO:0007059 chromosome segregation	79/637	424/18870	2.302705e-36
5	5	GO:0007059 chromosome segregation	79/637	424/18870	2.302705e-36
6	6	GO:0007059 chromosome segregation	79/637	424/18870	2.302705e-36
		p.adjust	qvalue	geneID	Count
				GeneRatio_decimal	BgRatio_decimal
1	1.075824e-32	9.555014e-33	NCAPD2	79	0.1240188
2	1.075824e-32	9.555014e-33	DPF1	79	0.1240188
3	1.075824e-32	9.555014e-33	BRCA1	79	0.1240188
4	1.075824e-32	9.555014e-33	TACC3	79	0.1240188
5	1.075824e-32	9.555014e-33	RAD18	79	0.1240188
6	1.075824e-32	9.555014e-33	TRIP13	79	0.1240188

```

# make a list of the select the pathways of interest
selected_pathways_24h <- c('ERK1 and ERK2 cascade',
                           'epithelial cell migration','leukocyte migration',
                           'sprouting angiogenesis','regeneration',
                           'regulation of microtubule cytoskeleton organization',
                           'substrate adhesion-dependent cell spreading',
                           'chemokine production',
                           'microtubule depolymerization',
                           'positive regulation of vasculature development')

# check the selected pathways
selected_pathways_24h

```

```

[1] "ERK1 and ERK2 cascade"
[2] "epithelial cell migration"
[3] "leukocyte migration"
[4] "sprouting angiogenesis"
[5] "regeneration"
[6] "regulation of microtubule cytoskeleton organization"
[7] "substrate adhesion-dependent cell spreading"
[8] "chemokine production"
[9] "microtubule depolymerization"
[10] "positive regulation of vasculature development"

```

```

# subset the GO analysis results to the selected pathways
GO_res_uni_bidirectional_24_ <-
  GO_res_uni_bidirectional_24_[GO_res_uni_bidirectional_24_$Description
                            %in% selected_pathways_24h,]

# arrange the results by GeneRatio_decimal
GO_res_uni_bidirectional_24_ <- GO_res_uni_bidirectional_24_ %>%
  arrange((GeneRatio_decimal))

# set the levels of the Description column
GO_res_uni_bidirectional_24_$Description <-
  factor(GO_res_uni_bidirectional_24_$Description,
         levels=unique(GO_res_uni_bidirectional_24_$Description))

# check the dimensions of the data and the first rows
GO_res_uni_bidirectional_24_ %>% dim

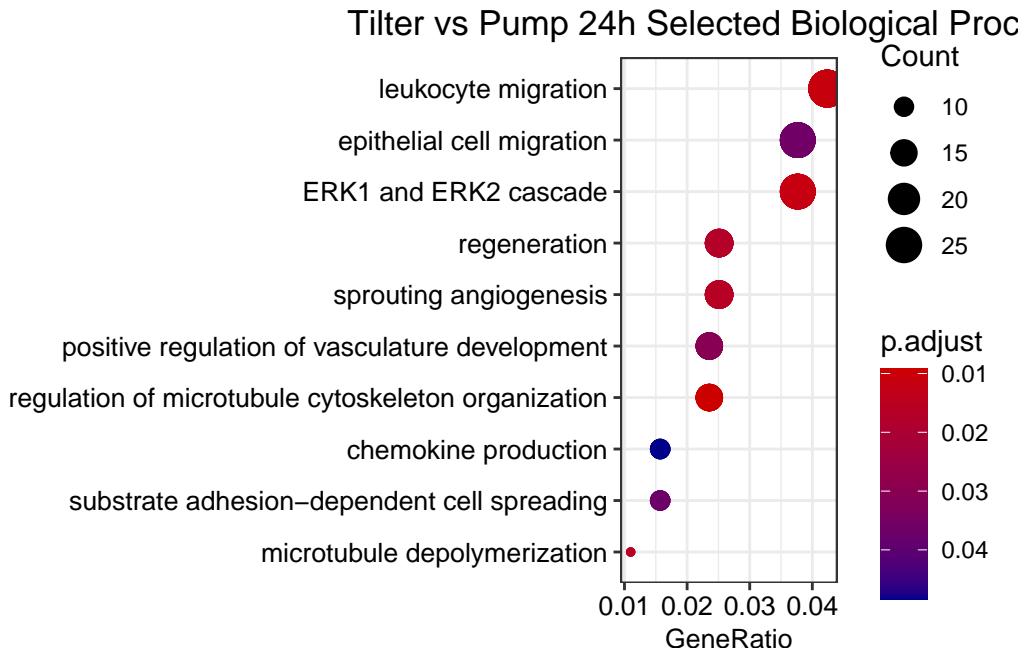
```

```
[1] 164 12
```

```
GO_res_uni_bidirectional_24_ %>% head
```

X	ID	Description	GeneRatio	BgRatio	pvalue
1	4776	GO:0007019 microtubule depolymerization	7/637	45/18870	0.0007183456
2	4777	GO:0007019 microtubule depolymerization	7/637	45/18870	0.0007183456
3	4778	GO:0007019 microtubule depolymerization	7/637	45/18870	0.0007183456
4	4779	GO:0007019 microtubule depolymerization	7/637	45/18870	0.0007183456
5	4780	GO:0007019 microtubule depolymerization	7/637	45/18870	0.0007183456
6	4781	GO:0007019 microtubule depolymerization	7/637	45/18870	0.0007183456
p.adjust	qvalue	geneID	Count	GeneRatio_decimal	BgRatio_decimal
1	0.01574125	0.01398072	NAV3	7	0.01098901
2	0.01574125	0.01398072	TPX2	7	0.01098901
3	0.01574125	0.01398072	STMN1	7	0.01098901
4	0.01574125	0.01398072	KIF18A	7	0.01098901
5	0.01574125	0.01398072	KIF2C	7	0.01098901
6	0.01574125	0.01398072	NCKAP5	7	0.01098901

```
# set the maximum number of overlaps
options(ggrepel.max.overlaps = 50)
ggplot(data = (GO_res_uni_bidirectional_24_ ) , # 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 = "red3", high = "blue4",
                      guide=guide_colourbar(reverse = TRUE)) +
# set color gradient to blue to red
theme_bw() + # set theme to black and white
labs(title = paste0("Tilter vs Pump 24h Selected Biological Processes"),
      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")) +
# change color of Counts in legend to black
guides(size = guide_legend(override.aes = list(color = "black", alpha=1)))
```



```
# GO analysis for the selected pathways for tilter and pump at 4h time point
# read the GO analysis results obtained in the previous step
GO_res_uni_bidirectional_4_ <- read.csv("pump_tilter_4h_GO_BP_pathways.csv")
#make a list of the select the pathways of interest
selected_pathways_4h <-
  c('transmembrane receptor protein serine/threonine kinase signaling pathway',
    'ERK1 and ERK2 cascade','p38MAPK cascade','stress-activated MAPK cascade',
    'BMP signaling pathway','SMAD protein signal transduction','JNK cascade',
    'canonical NF-kappaB signal transduction')
# check the selected pathways
selected_pathways_4h
```

```
[1] "transmembrane receptor protein serine/threonine kinase signaling pathway"
[2] "ERK1 and ERK2 cascade"
[3] "p38MAPK cascade"
[4] "stress-activated MAPK cascade"
[5] "BMP signaling pathway"
[6] "SMAD protein signal transduction"
[7] "JNK cascade"
[8] "canonical NF-kappaB signal transduction"
```

```

# subset the GO analysis results to the selected pathways
GO_res_uni_bidirectional_4_ <-
  GO_res_uni_bidirectional_4_[GO_res_uni_bidirectional_4_$Description
                            %in% selected_pathways_4h,]
GO_res_uni_bidirectional_4_ <- GO_res_uni_bidirectional_4_ %>%
  arrange(GeneRatio_decimal) # arrange the results by GeneRatio_decimal
# set the levels of the Description column
GO_res_uni_bidirectional_4_$Description <-
  factor(GO_res_uni_bidirectional_4_$Description,
         levels=unique(GO_res_uni_bidirectional_4_$Description))
GO_res_uni_bidirectional_4_ %>% dim # check the dimensions of the data

```

[1] 90 12

```
GO_res_uni_bidirectional_4_ %>% head # check the first rows
```

X	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust
1	931	GO:0038066 p38MAPK cascade	10/317	57/18870	3.335848e-08	0.0001227592
2	932	GO:0038066 p38MAPK cascade	10/317	57/18870	3.335848e-08	0.0001227592
3	933	GO:0038066 p38MAPK cascade	10/317	57/18870	3.335848e-08	0.0001227592
4	934	GO:0038066 p38MAPK cascade	10/317	57/18870	3.335848e-08	0.0001227592
5	935	GO:0038066 p38MAPK cascade	10/317	57/18870	3.335848e-08	0.0001227592
6	936	GO:0038066 p38MAPK cascade	10/317	57/18870	3.335848e-08	0.0001227592
			qvalue	geneID	Count	GeneRatio_decimal
1	0.0001045349	MAP2K3	10		0.03154574	0.003020668
2	0.0001045349	PHLPP1	10		0.03154574	0.003020668
3	0.0001045349	BMP4	10		0.03154574	0.003020668
4	0.0001045349	ZFP36	10		0.03154574	0.003020668
5	0.0001045349	DAB2IP	10		0.03154574	0.003020668
6	0.0001045349	MFHAS1	10		0.03154574	0.003020668

```

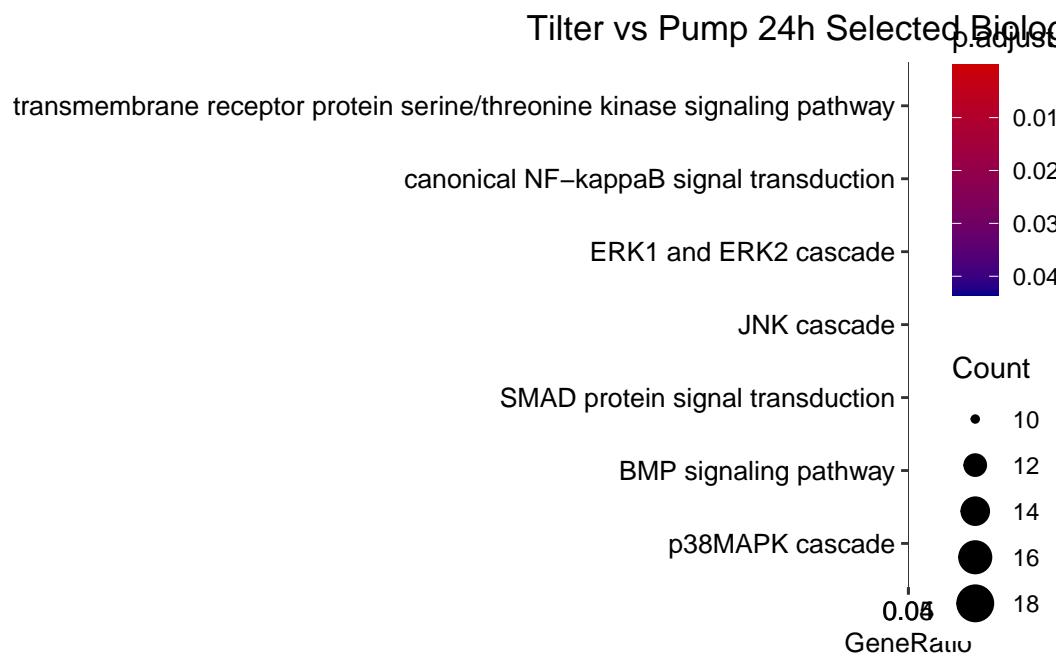
# create a dotplot for the GO terms with GeneRatio_decimal on x axis decreasing
# gene ratio , and GO term on y axis, color by p.adjust size of dot by Count
# set the maximum number of overlaps
options(ggrepel.max.overlaps = 50)
ggplot(data = (GO_res_uni_bidirectional_4_), # make ggplot object
       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 = "red3", high = "blue4",
                      guide=guide_colourbar(reverse = TRUE)) +
# set color gradient to blue to red
theme_bw() + # set theme to black and white
labs(title = paste0("Tilter vs Pump 24h Selected Biological Processes"),
     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")) +
# change color of Counts in legend to black
guides(size = guide_legend(override.aes = list(color = "black", alpha=1)))

```



## 10 Plot Heatmap of top 500 genes across all samples

```

##heatmap
#create a normalized data and select number of genes for heatmap

```

```

vst_data <- vst(dds_flow_time, blind = FALSE) # create a variance stabilized data
heatmap_matrix <- assay(vst_data) # create a matrix of the variance stabilized data

#create a normalized data and select the top 500 for heatmap across all conditions
top_genes_500 <- head(order(rowVars(heatmap_matrix), decreasing = TRUE), 500) # select the top 500 genes
heatmap_matrix_all <- heatmap_matrix[top_genes_500, ] # subset the heatmap matrix to the top 500 genes
heatmap_matrix_all_subset_col <-
  heatmap_matrix_all[, c("e3_4h_t", "e4_4h_t", "e6_4h_t", "e3_4h_p",
                        "e4_4h_p", "e6_4h_p", "e3_24h_t", "e4_24h_t", "e6_24h_t",
                        "e3_24h_p", "e4_24h_p", "e6_24h_p)] # subset the heatmap matrix to the top 500 genes
colnames(heatmap_matrix_all_subset_col) # check the column names

[1] "e3_4h_t"   "e4_4h_t"   "e6_4h_t"   "e3_4h_p"   "e4_4h_p"   "e6_4h_p"
[7] "e3_24h_t"  "e4_24h_t"  "e6_24h_t"  "e3_24h_p"  "e4_24h_p"  "e6_24h_p"

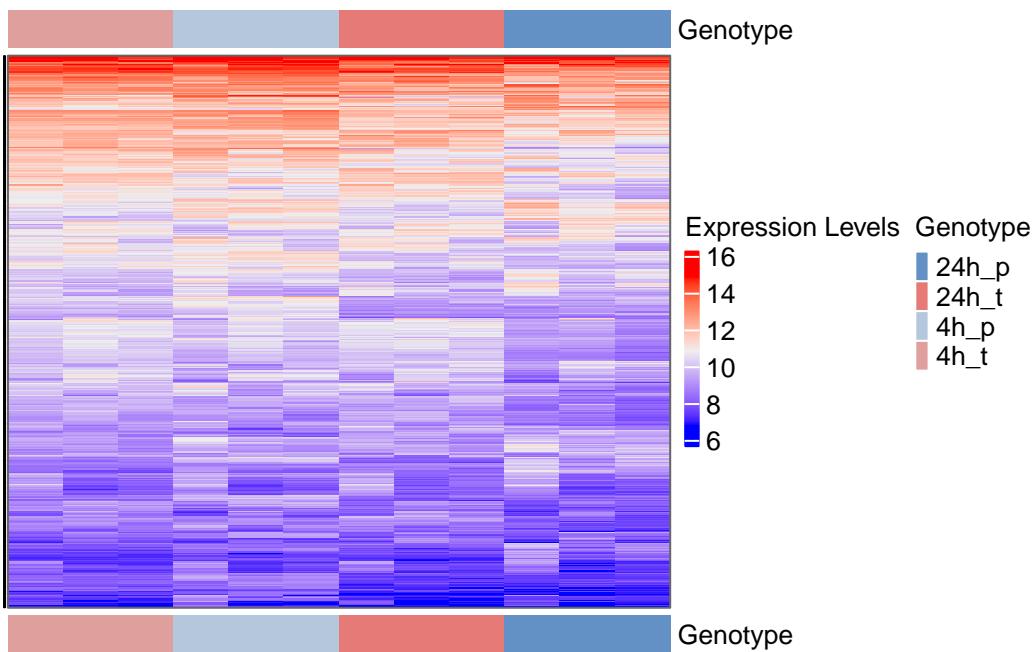
# create the heatmap annotation
ha = HeatmapAnnotation(
  Genotype = rep(c("4h_t", "4h_p", "24h_t", "24h_p"), each=3),
  col = list(Genotype = c(
    "4h_t" = "#DE9F9D", "4h_p" = "#B5C7DC", "24h_t" = "#E57A77", "24h_p" = "#6391C5"
  ))
),
# change font size of the annotation
annotation_name_gp = grid::gpar(fontsize = 10),
annotation_legend_param = list(
  # title = "Expression Levels",
  title_gp = gpar(fontsize = 10),
  labels_gp = gpar(fontsize = 10),
  grid_width = unit(5, "point"),
  grid_height = unit(5, "point")
)
)
Heatmap(heatmap_matrix_all_subset_col, name="Expression Levels",
        cluster_columns = FALSE,
        cluster_rows = TRUE,
        row_dend_width = unit(0, "cm"),
        show_column_names = FALSE,
        show_row_names = FALSE,
        top_annotation = ha,
        gap=unit(2, "point"),

```

```

border= TRUE,
border_gp = gpar(col = "grey40"),
row_title_gp = gpar(fontsize = 5), # change row split font size
row_title_rot = 0,
bottom_annotation = ha,
heatmap_legend_param = list(
  title = "Expression Levels",
  title_gp = gpar(fontsize = 10),
  labels_gp = gpar(fontsize = 10),
  grid_width    = unit(5,"point"),
  grid_height   = unit(5,"point")
),
row_names_gp = grid::gpar(fontsize = 14)
)

```



## 10.1 Plot Heatmap of selected pathways

```

#selected pathway heatmaps comparing tilter and pump at 24h time point
# convert the matrix to a data frame
heatmap_matrix <- as.data.frame(heatmap_matrix)

```

```

#add the row names as a new column
heatmap_matrix$ensembl_gene_id <- rownames(heatmap_matrix)
# merge the data with the gene names
heatmap_matrix_merged <- merge(heatmap_matrix, gene_id_df,
                                by.x = "ensembl_gene_id", all.x=TRUE)
# subset the data to the selected genes
heatmap_matrix_merged_selectedpathwyas_24_all <- heatmap_matrix_merged[
  heatmap_matrix_merged$hgnc_symbol %in% c('SELE', 'KITLG', 'STK10', 'TGFB2',
                                             'HMOX1', 'FLT1', 'IL1A', 'PGF',
                                             'TNFSF18', 'EDNRB', 'IL33', 'CH25H',
                                             'LYST', 'ADAM17', 'VCAM1', 'CXCL5',
                                             'ITGA2', 'LGALS9', 'SELP', 'AN06',
                                             'GBA1', 'CALR', 'CSF1', 'SELL',
                                             'HMGB1', 'CCL23', 'CCL16', 'CFLAR',
                                             'CEACAM1', 'NOX4', 'FAM83D', 'HMGCR',
                                             'GBP1', 'DUSP4', 'BMP4', 'SPRY2',
                                             'ACKR3', 'MFHAS1', 'CNKSR3', 'ATF3',
                                             'NEK10', 'BMPER', 'CDK1', 'TNFAIP8L3',
                                             'IQGAP3', 'MIR126', 'ANLN', 'PTGS2',
                                             'ANGPT2', 'RGCC', 'SASH1', 'PIK3R3',
                                             'PROX1', 'NR4A1', 'EFNB2', 'MACF1',
                                             'KLF4', 'HAS2', 'NANOS1', 'FGF16',
                                             'E2F2', 'RECK', 'E2F8', 'TSPAN18',
                                             'E2F7', 'AURKA', 'PCNA', 'POSTN',
                                             'LPIN1', 'NREP', 'ATIC', 'CCNA2',
                                             'ADM', 'NNMT', 'TYMS', 'DHFR',
                                             'SPAAR', 'TACC3', 'NAV3', 'SPAG5',
                                             'MAP2', 'TPX2', 'STMN1', 'KIF18A',
                                             'STIL', 'HAUS8', 'PSRC1', 'SKA1',
                                             'SKA3', 'PLK1', 'FAM107A', 'NUP62',
                                             'BRCA1', 'HIPK2', 'F3', 'AGO2',
                                             'ADM2', 'ST6GAL1', 'UNC13D', 'MERTK',
                                             'AKIP1', 'PEAK1', 'S100A10', 'OAS1',
                                             'OAS3', 'IL1RL1', 'KIF2C', 'NCKAP5',
                                             'KIF18B'),]

#prepare the selected data for a plotting a heatmap
# subset the data to remove the ensembl gene id
heatmap_matrix_merged_selectedpathwyas_24_all <-
  subset(heatmap_matrix_merged_selectedpathwyas_24_all,
         select = -ensembl_gene_id)
# set the row names to the hgnc symbol

```

```

rownames(heatmap_matrix_merged_selectedpathwyas_24_all) <-
  heatmap_matrix_merged_selectedpathwyas_24_all$hgnc_symbol
# convert the data to a data frame
heatmap_matrix_merged_selectedpathwyas_24_all <-
  as.data.frame(heatmap_matrix_merged_selectedpathwyas_24_all)
# subset the data to remove the hgnc symbol
heatmap_matrix_merged_selectedpathwyas_24_all <-
  subset(heatmap_matrix_merged_selectedpathwyas_24_all, select = -hgnc_symbol)
# convert the data to a matrix
heatmap_matrix_merged_selectedpathwyas_24_all <-
  as.matrix(heatmap_matrix_merged_selectedpathwyas_24_all)
# select the columns of interest
heatmap_matrix_merged_selectedpathwyas_24_all <-
  heatmap_matrix_merged_selectedpathwyas_24_all[, c("e3_24h_t", "e4_24h_t",
                                                    "e6_24h_t", "e3_24h_p",
                                                    "e4_24h_p", "e6_24h_p")]
# check the column names
colnames(heatmap_matrix_merged_selectedpathwyas_24_all)

```

[1] "e3\_24h\_t" "e4\_24h\_t" "e6\_24h\_t" "e3\_24h\_p" "e4\_24h\_p" "e6\_24h\_p"

```

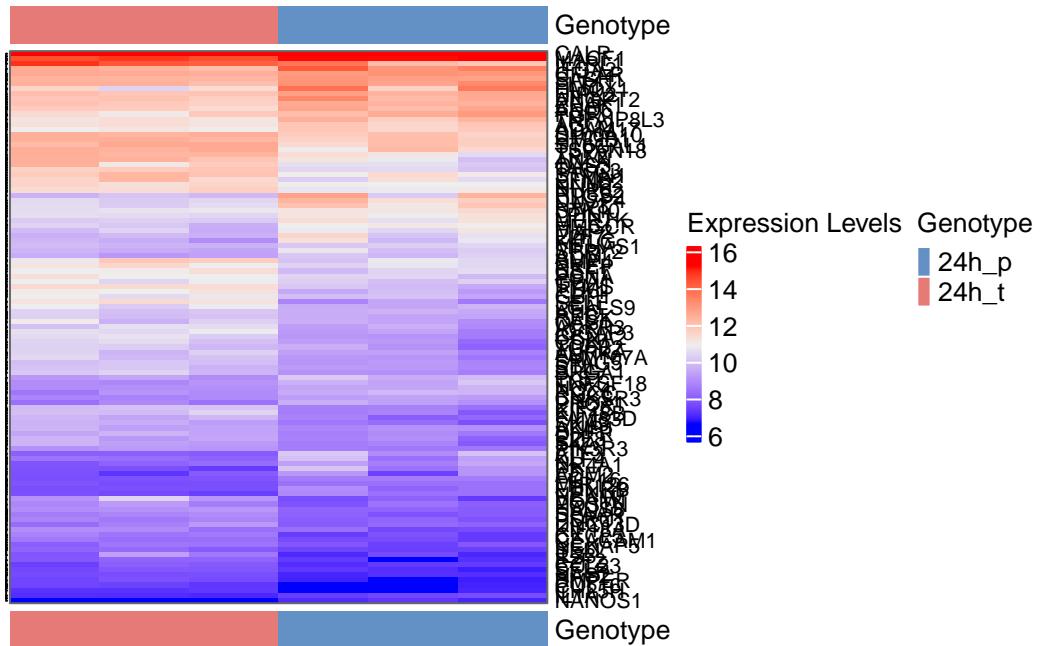
##heatmap of the selected genes comparing tilter and pump at 24h time point
# create the heatmap annotation
ha = HeatmapAnnotation(
  Genotype = rep(c("24h_t", "24h_p"), each=3),
  col = list(Genotype = c(
    "24h_t" ="#E57A77", "24h_p"="#6391C5"
  )),
  # change font size of the annotation
  annotation_name_gp = grid::gpar(fontsize = 10),
  annotation_legend_param = list(
    # title = "Expression Levels",
    title_gp = gpar(fontsize = 10),
    labels_gp = gpar(fontsize = 10),
    grid_width = unit(5,"point"),
    grid_height = unit(5,"point")
  )
)

```

```

Heatmap(heatmap_matrix_merged_selectedpathwyas_24_all, name="Expression Levels",
        cluster_columns = FALSE,
        cluster_rows = TRUE,
        row_dend_width = unit(0, "cm"),
        show_column_names = FALSE,
        show_row_names = TRUE,
        top_annotation = ha,
        gap=unit(2, "point"),
        border= TRUE,
        border_gp = gpar(col = "grey40"),
        row_title_gp = gpar(fontsize = 5), # change row split font size
        row_title_rot = 0,
        bottom_annotation = ha,
        heatmap_legend_param = list(
            title = "Expression Levels",
            title_gp = gpar(fontsize = 10),
            labels_gp = gpar(fontsize = 10),
            grid_width    = unit(5,"point"),
            grid_height   = unit(5,"point")
        ),
        row_names_gp = grid::gpar(fontsize = 8)
)

```



```

##subset the selected genes heatmaps comparing titter and pump at 4h timepoint
heatmap_matrix_merged_selectedpathwyas_4_all <- heatmap_matrix_merged[
  heatmap_matrix_merged$hgnc_symbol %in% c("TAB2", "SPHK2", "HMOX1", "NFKB1",
                                             "TRIM25", "EDNRB", "DAB2IP", "PELI2",
                                             "BRD4", "SLC20A1", "CXXC5",
                                             "C18orf32", "PER1", "ROR1", "PELI1",
                                             "RAPGEF2", "BTN2A2", "BMP4", "LIF",
                                             "SPRY2", "MAP3K12", "ACKR3", "EGFR",
                                             "MFHAS1", "EPHB1", "ATF3", "BMPER",
                                             "PDGFA", "HIPK2", "PHLPP1", "SH2D3C",
                                             "SH3RF1", "SH3RF3", "SMAD7", "TGFBR1",
                                             "ZMIZ1", "PMEPA1", "SMAD6", "SKI",
                                             "LDLRAD4", "BMPR2", "MAP2K3", "ZFP36",
                                             "MAPKAPK2", "SPHK1", "RELL1", "EGR1",
                                             "NOG", "SMURF1", "INTS9", "APPL2",
                                             "NRROS"),]

# subset the data to remove the ensembl gene id
heatmap_matrix_merged_selectedpathwyas_4_all <-
  subset(heatmap_matrix_merged_selectedpathwyas_4_all,
         select = -ensembl_gene_id)
# set the row names to the hgnc symbol
rownames(heatmap_matrix_merged_selectedpathwyas_4_all) <-
  heatmap_matrix_merged_selectedpathwyas_4_all$hgnc_symbol
# convert the data to a data frame
heatmap_matrix_merged_selectedpathwyas_4_all <-
  as.data.frame(heatmap_matrix_merged_selectedpathwyas_4_all)
# subset the data to remove the hgnc symbol
heatmap_matrix_merged_selectedpathwyas_4_all <-
  subset(heatmap_matrix_merged_selectedpathwyas_4_all, select = -hgnc_symbol)
# convert the data to a matrix
heatmap_matrix_merged_selectedpathwyas_4_all <-
  as.matrix(heatmap_matrix_merged_selectedpathwyas_4_all)
# select the columns of interest
heatmap_matrix_merged_selectedpathwyas_4_all <-
  heatmap_matrix_merged_selectedpathwyas_4_all[, c("e3_4h_t", "e4_4h_t", "e6_4h_t",
                                                   "e3_4h_p", "e4_4h_p",
                                                   "e6_4h_p")]
# check the column names
colnames(heatmap_matrix_merged_selectedpathwyas_4_all)

```

```
[1] "e3_4h_t" "e4_4h_t" "e6_4h_t" "e3_4h_p" "e4_4h_p" "e6_4h_p"
```

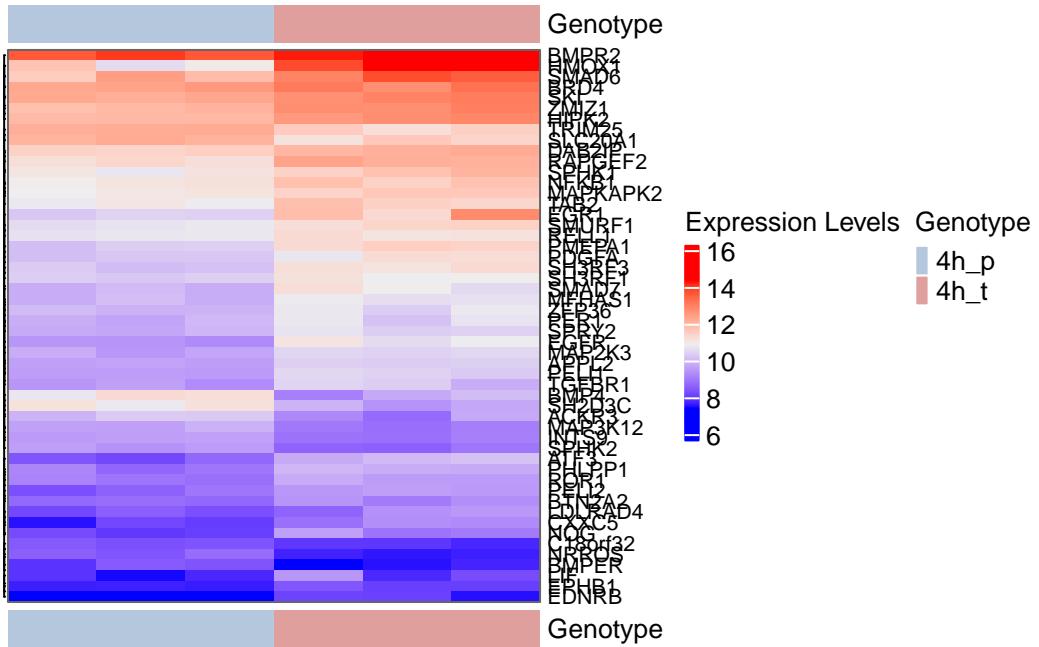
```

# create the heatmap annotation

ha = HeatmapAnnotation(
  Genotype = rep(c("4h_p","4h_t" ),each=3),
  col = list(Genotype = c(
    "4h_p"="#B5C7DC","4h_t"="#DE9F9D"
  )
),
# change font size of the annotation
annotation_name_gp = grid::gpar(fontsize = 10),
annotation_legend_param = list(
  # title = "Expression Levels",
  title_gp = gpar(fontsize = 10),
  labels_gp = gpar(fontsize = 10),
  grid_width = unit(5,"point"),
  grid_height = unit(5,"point")
)
)

Heatmap(heatmap_matrix_merged_selectedpathwyas_4_all, name="Expression Levels",
        cluster_columns = FALSE,
        cluster_rows = TRUE,
        row_dend_width = unit(0, "cm"),
        show_column_names = FALSE,
        show_row_names = TRUE,
        top_annotation = ha,
        gap=unit(2, "point"),
        border= TRUE,
        border_gp = gpar(col = "grey40"),
        row_title_gp = gpar(fontsize = 5), # change row split font size
        row_title_rot = 0,
        bottom_annotation = ha,
        heatmap_legend_param = list(
          title = "Expression Levels",
          title_gp = gpar(fontsize = 10),
          labels_gp = gpar(fontsize = 10),
          grid_width = unit(5,"point"),
          grid_height = unit(5,"point")
        ),
        row_names_gp = grid::gpar(fontsize = 8)
)

```



```
# save.image("workspace/Deseq_cleanedup_analysis_160224.Rdata") # save the workspace
```