Meta-analysis using data from this study and from GSE158081

Negar Vahdani

2024-06-01

Table of contents

1 Clean Memory, set working directory and load libraries

```
##Clear Memory and set the working directory
#| label: load_libraries
#| echo: true
#| results: 'hide'
#| message: false
#| warning: false
rm(list=ls())
gc()
```

```
          used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
Ncells  583831 31.2    1333382 71.3         NA   669411 35.8
Vcells 1079305  8.3    8388608 64.0      16384  1851664 14.2
```

```
setwd("/Users/negarvahdani/Deseq2")
dir.create("results", recursive = TRUE)
```

```
Warning in dir.create("results", recursive = TRUE): 'results' already exists
```

```
#load libraries
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(DESeq2)
```

```
Warning: package 'DESeq2' was built under R version 4.3.3
```

```
Loading required package: S4Vectors
```

```
Loading required package: stats4
```

```
Loading required package: BiocGenerics
```

```

Attaching package: 'BiocGenerics'


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:dplyr':

    first, rename


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 objects are masked from 'package:dplyr':

    collapse, desc, slice

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
```

```r
library(biomaRt)
library(ggplot2)
library(ggrepel)
library(ComplexHeatmap)
```

```
Loading required package: grid
```

```
========================================
ComplexHeatmap version 2.18.0
Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
Github page: https://github.com/jokergoo/ComplexHeatmap
Documentation: http://jokergoo.github.io/ComplexHeatmap-reference

If you use it in published research, please cite either one:
- Gu, Z. Complex Heatmap Visualization. iMeta 2022.
- Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
    genomic data. Bioinformatics 2016.


The new InteractiveComplexHeatmap package can directly export static
complex heatmaps into an interactive Shiny app with zero effort. Have a try!
```

```
This message can be suppressed by:
  suppressPackageStartupMessages(library(ComplexHeatmap))
========================================
```

```r
library(clusterProfiler)
```

```
clusterProfiler v4.10.0  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, an


Attaching package: 'clusterProfiler'


The following object is masked from 'package:biomaRt':

    select


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

    slice


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

    rename


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

    filter
```

```r
library(tibble)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v readr     2.1.4
v lubridate 1.9.3     v stringr   1.5.1
v purrr     1.0.2     v tidyr     1.3.0
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x lubridate::%within%()     masks IRanges::%within%()
x IRanges::collapse()       masks dplyr::collapse()
x Biobase::combine()        masks BiocGenerics::combine(), dplyr::combine()
x matrixStats::count()      masks dplyr::count()
x IRanges::desc()           masks dplyr::desc()
x tidyr::expand()           masks S4Vectors::expand()
x clusterProfiler::filter() masks dplyr::filter(), stats::filter()
x S4Vectors::first()        masks dplyr::first()
x dplyr::lag()              masks stats::lag()
x ggplot2::Position()       masks BiocGenerics::Position(), base::Position()
x purrr::reduce()           masks GenomicRanges::reduce(), IRanges::reduce()
x clusterProfiler::rename() masks S4Vectors::rename(), dplyr::rename()
x lubridate::second()       masks S4Vectors::second()
x lubridate::second<-()     masks S4Vectors::second<-()
x clusterProfiler::select() masks biomaRt::select(), dplyr::select()
x purrr::simplify()         masks clusterProfiler::simplify()
x clusterProfiler::slice()  masks IRanges::slice(), dplyr::slice()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(matrixStats)
library(readxl)
library(ggVennDiagram)
```

```
Attaching package: 'ggVennDiagram'

The following object is masked from 'package:tidyr':

    unite
```

```r
library(sva)
```

```
Loading required package: mgcv
Loading required package: nlme

Attaching package: 'nlme'

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

    collapse
```

```
The following object is masked from 'package:dplyr':

    collapse

This is mgcv 1.9-0. For overview type 'help("mgcv-package")'.
Loading required package: genefilter

Attaching package: 'genefilter'

The following object is masked from 'package:readr':

    spec

The following object is masked from 'package:ComplexHeatmap':

    dist2

The following objects are masked from 'package:MatrixGenerics':

    rowSds, rowVars

The following objects are masked from 'package:matrixStats':

    rowSds, rowVars

Loading required package: BiocParallel
```

2 Import the data and metadata from the paaper and pervious study

```r
##import the data for meta-analysis and add edit the column names
count_elife <-
  read.delim("/Users/negarvahdani/elife_reverse_strand_gene_counts_inforemoved.txt",
             sep = "\t", stringsAsFactor = FALSE, header= TRUE,
             fill = TRUE, row.names = 1)
colnames(count_elife) <-
  gsub("X.data.users.nvahdani.flow_project.meta.analysis.bamsort.", "",
       colnames(count_elife))
colnames(count_elife) <-sub(".sorted.bam", "", colnames(count_elife))
```

```
count_elife <- subset(count_elife, select = -End)
sample_names_elife <- paste0("cord_",seq(1,7)) # create column names
colnames(count_elife) <- sample_names_elife # assign the column names
colnames(count_elife) # check the column names
```

```
[1] "cord_1" "cord_2" "cord_3" "cord_4" "cord_5" "cord_6" "cord_7"
```

```
sample_name_elife_df <- data.frame(sample_name=colnames(count_elife)) # create a data frame

#load the metadata
metadata_combined <- read.csv("results/metadata_combined_elife.csv",
                              row.names = 1) # available on github

#import the data from the flow study for further analysis
#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
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"
```

```
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
```

3 Bind the count tables and batch correct the data

```
# find the rows that are not in both count tables and remove them
missing_rows <- rownames(count_elife)[!rownames(count_elife) %in%
                                        rownames(counts110_trimmed_colremove)]
count_elife <- count_elife[!rownames(count_elife)%in% missing_rows,]

##bind the count tables and batch correct the data
counts_combined <- bind_cols(counts110_trimmed_colremove,count_elife)
identical(rownames(metadata_combined), colnames(counts_combined)) # check if the rownames of
```

```
[1] TRUE
```

```
head(counts_combined)# check the head of the counts
```

```
                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 cord_1 cord_2 cord_3
ENSG00000279928       0        2        0       0       0      0      0      0
ENSG00000228037       0        0        0       0       0      0      1      0
ENSG00000142611       0        0        0       5       0      5      3     22
ENSG00000284616       0        0        0       0       0      0      0      0
```

```
ENSG00000157911     407      377      467     383     388     48     56     77
ENSG00000269896      23       14       12       6       4      6      6      5
                cord_4 cord_5 cord_6 cord_7
ENSG00000279928      0      0      0      0
ENSG00000228037      1      0      0      0
ENSG00000142611     28     11      9     25
ENSG00000284616      0      0      0      0
ENSG00000157911    150     97     39     57
ENSG00000269896     13      9      8      2
```

```r
class(counts_combined) # check the class of the counts
```

```
[1] "data.frame"
```

```r
counts_combined <- as.matrix(counts_combined) # convert the counts to a matrix
counts_combined[is.na(counts_combined)] <- 0 # replace NAs with 0 for further analysis

# batch correct the counts using ComBat_seq function
batchcorrected_counts <- ComBat_seq(as.matrix(counts_combined),
                                    batch = metadata_combined$batch)
```

```
Found 2 batches
Using null model in ComBat-seq.
Adjusting for 0 covariate(s) or covariate level(s)
Estimating dispersions
Fitting the GLM model
Shrinkage off - using GLM estimates for parameters
Adjusting the data
```

4 Make the Deseq2 object and filter the low expressed genes

```r
#check for NAs
max(counts_combined[!is.na(counts_combined)])
```

```
[1] 633514
```

```
sum(is.na(counts_combined))
```

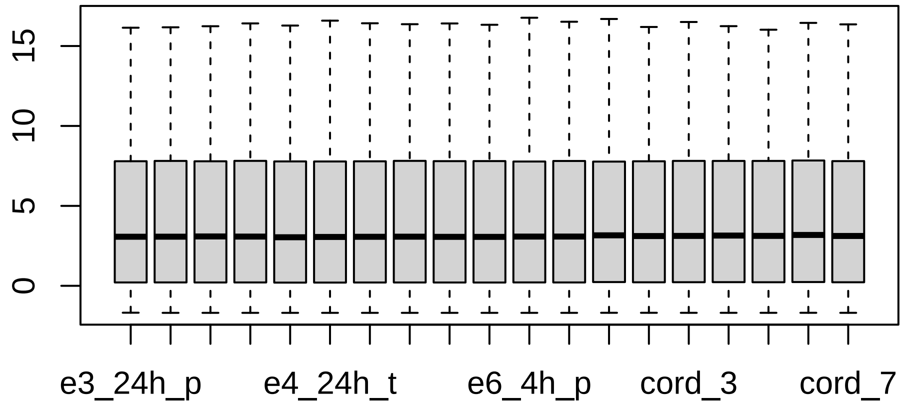
```
[1] 0
```

```
#create the Deseq2 object
dds_combined <- DESeqDataSetFromMatrix(countData = batchcorrected_counts,
                                       colData = metadata_combined,
                                       design= ~ flow_time)
```

```
converting counts to integer mode
```

```
Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
design formula are characters, converting to factors
```

```
#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_combined))

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

#subset the DESeqDataSet
dds_combined_filtered <- dds_combined[keep_genes,]

# check the batch correction
dds_combined_filtered_rlog <- rlog(dds_combined_filtered)
boxplot(assay(dds_combined_filtered_rlog))
```



5 PCA plot

```
## 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)
```

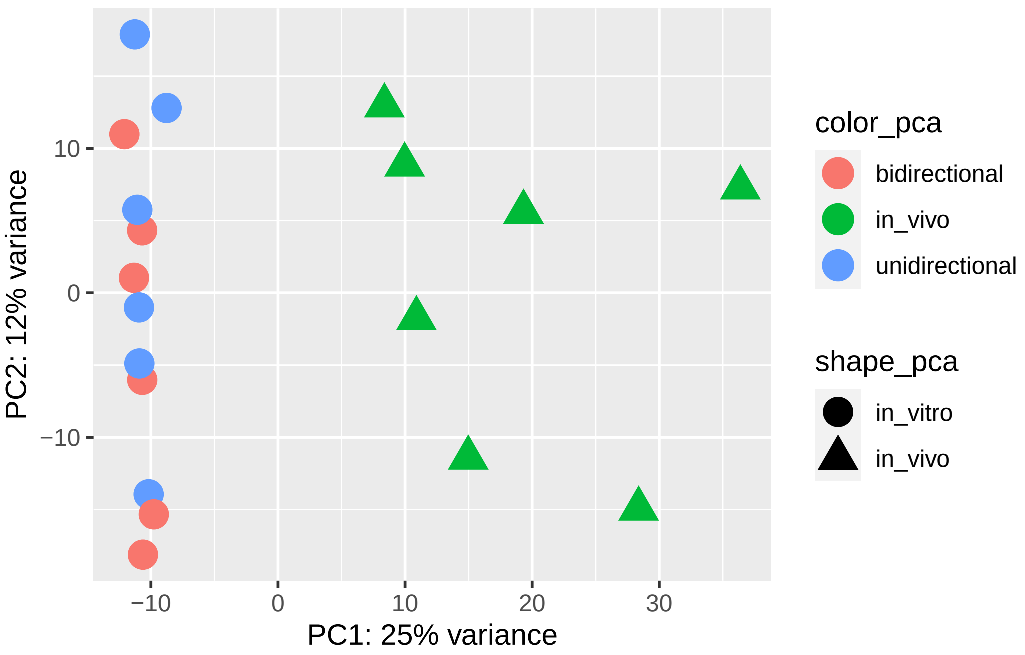
```
cat("Percents calculated...\n")
# Sys.sleep(0.2)

#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")) +
```

```r
      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)
}

PCA(assay(dds_combined_filtered_rlog),
    color_pca = factor(metadata_combined$flow_profile),
    shape_pca = factor(metadata_combined$batch),
    save_plot = "no",
    name_of_plot = "PCA_counts_corrected_flow_profile_rlog_afterfiltering")
```

```
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 Run Deseq2 analysis

```
## run DESeq2 analysis
dds_combined_filtered <- DESeq(dds_combined_filtered)
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

```
-- replacing outliers and refitting for 92 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)
```

```
estimating dispersions
```

```
fitting model and testing
```

```
#run DEGs, compare in vivo data with tilter and pump in 2 time points. Convert the results i
##tilter 4h vs cord
res_tilter4_cord <- lfcShrink(dds_combined_filtered,
                              parallel = TRUE,
                              contrast=c("flow_time",
                                         "bidirectional_4","in_vivo_in_vivo"),
                              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
```

```
res_tilter4_cord <- as.data.frame(res_tilter4_cord)
head(res_tilter4_cord)
```

```
                   baseMean log2FoldChange       lfcSE     pvalue      padj
ENSG00000228037   0.5893282  -0.0683821500 0.739574374 0.38261558        NA
ENSG00000142611   2.7148644  -1.6429503900 2.762428845 0.00430657 0.1172559
ENSG00000157911 194.5542076   0.0008040096 0.016549861 0.23302595 0.9999433
ENSG00000269896  10.7625682  -0.0011530149 0.055823704 0.73688903 0.9999433
ENSG00000228463   1.5224704  -0.0059604852 0.206791566 0.75439209        NA
ENSG00000142655 267.9577538  -0.0002799315 0.009572694 0.54378078 0.9999433
```

```
res_tilter4_cord$ensembl_gene_id <- rownames(res_tilter4_cord)

##tilter 24h vs cord
res_tilter24_cord <- lfcShrink(dds_combined_filtered,
                               parallel = TRUE,
                               contrast=c("flow_time", "bidirectional_24",
                                          "in_vivo_in_vivo"), 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
```

```
res_tilter24_cord <- as.data.frame(res_tilter24_cord)
head(res_tilter24_cord)
```

```
                   baseMean log2FoldChange       lfcSE     pvalue      padj
ENSG00000228037   0.5893282  -1.157016e-01 0.941501081 0.36014738        NA
ENSG00000142611   2.7148644  -1.067048e+00 2.198964747 0.01278252 0.1738865
ENSG00000157911 194.5542076   2.526878e-04 0.012016538 0.67998817 0.9995752
ENSG00000269896  10.7625682  -2.232108e-03 0.074950062 0.71389759 0.9995752
ENSG00000228463   1.5224704  -5.795360e-02 0.483776460 0.29556129 0.9501662
ENSG00000142655 267.9577538   1.151449e-05 0.009253435 0.98138853 0.9995752
```

```
res_tilter24_cord$ensembl_gene_id <- rownames(res_tilter24_cord)

##pump 4h vs cord
res_pump4_cord <- lfcShrink(dds_combined_filtered,
                            parallel = TRUE,
                            contrast=c("flow_time", "unidirectional_4",
                                       "in_vivo_in_vivo"), 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
```

```
res_pump4_cord <- as.data.frame(res_pump4_cord)
head(res_pump4_cord)
```

```
                   baseMean log2FoldChange       lfcSE    pvalue      padj
ENSG00000228037   0.5893282   7.952826e-04 0.510322122 0.9891835        NA
ENSG00000142611   2.7148644  -1.971296e-03 0.160329880 0.8897757 0.9999782
ENSG00000157911 194.5542076  -3.430993e-04 0.011643244 0.5014258 0.9999782
ENSG00000269896  10.7625682   5.781174e-04 0.052837245 0.8617099 0.9999782
ENSG00000228463   1.5224704  -7.284686e-03 0.215523845 0.7142378        NA
ENSG00000142655 267.9577538   5.865377e-05 0.008724545 0.8922843 0.9999782
```

```r
res_pump4_cord$ensembl_gene_id <- rownames(res_pump4_cord)


##pump 24h vs cord
res_pump24_cord <- lfcShrink(dds_combined_filtered,
                             parallel = TRUE,
                             contrast=c("flow_time", "unidirectional_24",
                                        "in_vivo_in_vivo"), 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
```

```r
res_pump24_cord <- as.data.frame(res_pump24_cord)
head(res_pump24_cord)
```

```
                   baseMean log2FoldChange      lfcSE      pvalue       padj
ENSG00000228037   0.5893282   -0.123705458 0.95953406 0.362901138         NA
ENSG00000142611   2.7148644   -2.677964689 3.05362593 0.003343573 0.06931042
ENSG00000157911 194.5542076   -0.006330381 0.03813977 0.194440790 0.70646502
ENSG00000269896  10.7625682   -0.021389522 0.18491990 0.208242668 0.72777298
ENSG00000228463   1.5224704    0.016695308 0.30674768 0.682366981 0.97962935
ENSG00000142655 267.9577538    0.002285510 0.02916195 0.633998541 0.97230069
```

```r
res_pump24_cord$ensembl_gene_id <- rownames(res_pump24_cord)


#check for duplicates
table(duplicated(res_tilter4_cord$column1))
```

```
< table of extent 0 >
```

```r
table(duplicated(res_tilter24_cord$column1))
```

```
< table of extent 0 >
```

```r
table(duplicated(res_pump4_cord$column1))
```

```
< table of extent 0 >
```

```r
table(duplicated(res_pump24_cord$column1))
```

```
< table of extent 0 >
```

7 Add ENSMBL gene ids to the result tables

```r
# convert gene ids to names using biomart
symbol_to_ensmbl_human <- function() {
  require(biomaRt) # load biomaRt package
  ensembl <- useMart(biomart = "ensembl", dataset = "hsapiens_gene_ensembl")
  # use the main Ensembl mart host
  # query biomart
  annotations <- getBM(attributes = c("ensembl_gene_id", "hgnc_symbol"),
                       mart = ensembl)
  # get the ensembl gene id and the hgnc symbol
  return(annotations) # return the results
}

gene_id <- symbol_to_ensmbl_human() # get the gene ids
gene_id_df <- as.data.frame(gene_id) # convert the gene ids to a data frame
head(gene_id_df) # check the head of the data frame
```

```
  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
```

```r
##merge the ensembl gene id to the data frame
res_tilter4_cord <- merge(res_tilter4_cord, gene_id_df,
                          by.x = "ensembl_gene_id",all.x=TRUE)
res_tilter24_cord <- merge(res_tilter24_cord, gene_id_df,
                           by.x = "ensembl_gene_id",all.x=TRUE)
res_pump4_cord <- merge(res_pump4_cord, gene_id_df,
                        by.x = "ensembl_gene_id",all.x=TRUE)
res_pump24_cord <- merge(res_pump24_cord, gene_id_df,
                            by.x = "ensembl_gene_id",all.x=TRUE)
## remove NAs
res_tilter4_cord$padj[is.na(res_tilter4_cord$padj)] <- 1
res_tilter24_cord$padj[is.na(res_tilter24_cord$padj)] <- 1
res_pump4_cord$padj[is.na(res_pump4_cord$padj)] <- 1
res_pump24_cord$padj[is.na(res_pump24_cord$padj)] <- 1
```

8 Export the DEGs with padj $<$ 0.05

```
res_tilter4_cord_padj0.05 <- subset(res_tilter4_cord, padj < 0.05)
res_tilter24_cord_padj0.05 <- subset(res_tilter24_cord, padj < 0.05)
head(res_tilter4_cord_padj0.05)
```

```
    ensembl_gene_id  baseMean log2FoldChange     lfcSE       pvalue         padj
2   ENSG00000000005 21.750133      -9.165948 1.354568 5.601772e-12 1.423680e-09
29  ENSG00000002933 26.297965      -9.405525 1.409727 1.022849e-11 2.477219e-09
159 ENSG00000007264  3.210779      -3.543474 3.153673 8.750481e-04 3.646506e-02
217 ENSG00000009790 37.828165      -9.583033 1.897880 1.499627e-07 1.442586e-05
283 ENSG00000011600 75.109634     -10.669652 1.683327 7.092408e-11 1.431415e-08
341 ENSG00000016391  3.703080      -3.615176 3.132470 8.159064e-04 3.406947e-02
    hgnc_symbol
2          TNMD
29     TMEM176A
159        MATK
217    TRAF3IP3
283      TYROBP
341        CHDH
```

```
head(res_tilter24_cord_padj0.05)
```

```
    ensembl_gene_id  baseMean log2FoldChange     lfcSE       pvalue         padj
2   ENSG00000000005 21.750133      -9.273978 1.348992 2.090275e-12 5.453812e-10
29  ENSG00000002933 26.297965      -9.504552 1.403444 4.018956e-12 1.011150e-09
48  ENSG00000004468  4.155997      -3.786025 3.299989 1.466970e-03 4.305964e-02
101 ENSG00000005844  7.199565      -5.570936 2.963381 3.644626e-04 1.495641e-02
109 ENSG00000006016 11.817305      -7.284142 1.460448 2.792482e-07 2.469302e-05
187 ENSG00000008226  3.428860      -3.650852 3.150647 1.425206e-03 4.212631e-02
    hgnc_symbol
```

```
2          TNMD
29     TMEM176A
48         CD38
101       ITGAL
109       CRLF1
187       DLEC1
```

```
write.csv(as.data.frame(res_tilter4_cord_padj0.05),
          file="tilter_4_cord_0.05.csv")
write.csv(as.data.frame(res_tilter24_cord_padj0.05),
          file="tilter_24_cord_0.05.csv")


res_pump4_cord_padj0.05 <- subset(res_pump4_cord, padj < 0.05)
res_pump24_cord_padj0.05 <- subset(res_pump24_cord, padj < 0.05)
head(res_pump4_cord_padj0.05)
```

```
    ensembl_gene_id  baseMean log2FoldChange    lfcSE       pvalue         padj
2   ENSG00000000005 21.750133      -8.914139 1.354471 2.068000e-11 4.932673e-09
29  ENSG00000002933 26.297965      -9.154332 1.409617 3.529215e-11 7.945096e-09
217 ENSG00000009790 37.828165      -9.336276 1.902829 3.053582e-07 2.872374e-05
283 ENSG00000011600 75.109634     -10.421781 1.683147 1.906463e-10 3.569896e-08
354 ENSG00000018625 16.951451      -8.561871 1.328933 5.544828e-11 1.194582e-08
411 ENSG00000025423  7.736663      -3.496824 3.450453 1.175505e-03 4.609082e-02
    hgnc_symbol
2          TNMD
29     TMEM176A
217    TRAF3IP3
283      TYROBP
354      ATP1A2
411     HSD17B6
```

```
head(res_pump24_cord_padj0.05)
```

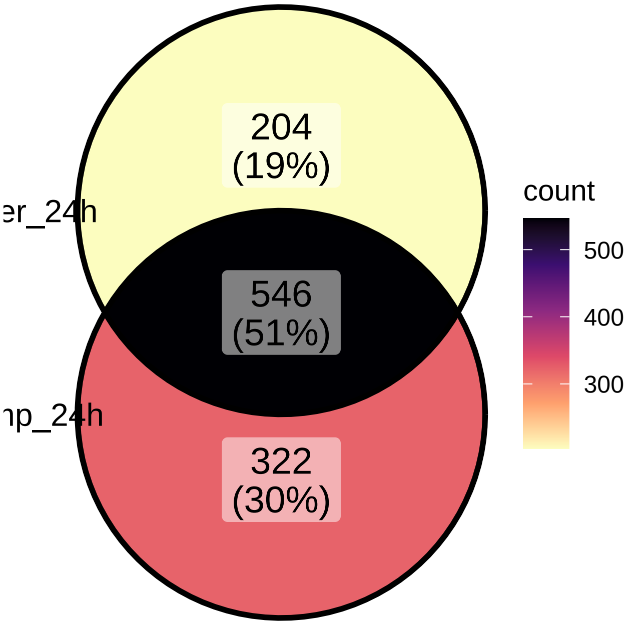
```
   ensembl_gene_id    baseMean log2FoldChange     lfcSE       pvalue
2  ENSG00000000005   21.750133      -9.161015 1.3517195 2.366078e-12
29 ENSG00000002933   26.297965      -9.385729 1.4073204 4.519745e-12
42 ENSG00000003989 1332.427065      -0.834547 1.0225241 1.969381e-03
48 ENSG00000004468    4.155997      -3.761451 3.1987372 1.526005e-03
76 ENSG00000005108  383.568272      -1.083551 0.4335631 5.391987e-05
79 ENSG00000005187  194.760889      -1.072248 0.4871026 8.392392e-05
```

```
           padj hgnc_symbol
2  6.181164e-10        TNMD
29 1.138060e-09    TMEM176A
42 4.948099e-02      SLC7A2
48 4.141221e-02        CD38
76 2.908320e-03      THSD7A
79 4.231912e-03       ACSM3
```

```
write.csv(as.data.frame(res_pump4_cord_padj0.05),
          file="pump_4_cord_0.05.csv")
write.csv(as.data.frame(res_pump24_cord_padj0.05),
          file="pump_24_cord_0.05.csv")
```

9 Make venn diagram between the two comparisons

```
venn_list <- list(Pump_24h=res_pump24_cord_padj0.05$ensembl_gene_id,
                    Tilter_24h=res_tilter24_cord_padj0.05$ensembl_gene_id

) # create a list of the gene ids


plot <- ggVennDiagram(venn_list,     label_size = 5,
                      # set_size=8
) +
  scale_fill_viridis_c(option = "magma", direction = -1) +
  theme(plot.title = element_text(hjust = 0.2)) # change the title position
print(plot)
```


204
(19%)
er_24h
546
(51%)
np_24h
322
(30%)
count
500
400
300