Marques et al 2021 SMARTSeq Analysis

Contents

Analyzing RNASeq from Proepicardium, Pericardium and Heart tube

#Clear Memory and load libraries

```
rm(list = ls())
gc()
```

```
##          used (Mb) gc trigger (Mb) max used (Mb)
## Ncells 424579 22.7     880310 47.1   665545 35.6
## Vcells 809193  6.2    8388608 64.0  1819096 13.9
```

```
# setwd("/home/prateek/Mercader_Lab/Ines_Data/SMART_Laura/final")
getwd()
```

```
## [1] "/home/prateek/Mercader_Lab/Ines_Data/SMART_Laura/final"
```

```
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(ggplot2)
library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

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

## The following objects are masked from 'package: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, 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 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
```

```
library(clusterProfiler)
```

```
##

## clusterProfiler v4.0.0  For help: https://guangchuangyu.github.io/software/clusterProfiler
##
## If you use clusterProfiler in published research, please cite:
## Guangchuang Yu, Li-Gen Wang, Yanyan Han, Qing-Yu He. clusterProfiler: an R package for comparing biological themes among gene clusters.

##
## 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:stats':
##
##     filter
```

```
library(enrichplot)
library(openxlsx)
library(cmapR)
```

Read counts file

```
counts_all <- read.table("SMART_Laura_featurecounts_counts_all.txt", header = T, check.names = F)
colnames(counts_all)[7:16] <- gsub(pattern = "./star_alignments_and_counts/", "", colnames(counts_all)[7:16])
colnames(counts_all)[7:16] <- gsub(pattern = "__.*", "", colnames(counts_all)[7:16])
rownames(counts_all) <- counts_all$Geneid
counts <- counts_all[,7:16]
counts <- counts %>% dplyr::select(-S3_3)
head(counts)
```

```
##                    S1_1 S1_2 S1_4 S1_5 S3_1 S3_2 S3_4 S5_1 S5_2
## ENSDARG00000102141    2    0    3    5    5   13    3    9    3
## ENSDARG00000102123    0    0    0    0    1    4   10    0    0
## ENSDARG00000114503    0    0    1    3    0    2    3    4    4
## ENSDARG00000115971    0    0    0    0    0    0    0    1    0
## ENSDARG00000098311    1    0    1    1    1    1    2    7    4
## ENSDARG00000104839    3    2    3    1    2    1    0    0    0
```

```
colnames(counts)
```

```
## [1] "S1_1" "S1_2" "S1_4" "S1_5" "S3_1" "S3_2" "S3_4" "S5_1" "S5_2"
```

Read Metadata

```
metadata_smart <- read.xlsx("Metadata.xlsx")
rownames(metadata_smart) <- metadata_smart$Sample_Name

#sample removed as it was outlier according to PCA
```

```
metadata_smart <- metadata_smart %>% filter(!(Sample_Name == "S3_3"))
(metadata_smart)
```

```
##      Sample_Name        Tissue
## S1_1        S1_1 Proepicardium
## S1_2        S1_2 Proepicardium
## S1_4        S1_4 Proepicardium
## S1_5        S1_5 Proepicardium
## S3_1        S3_1    Heart_Tube
## S3_2        S3_2    Heart_Tube
## S3_4        S3_4    Heart_Tube
## S5_1        S5_1   Pericardium
## S5_2        S5_2   Pericardium
```

Convert counts dataframe to matrix

```
counts_mat <- as.matrix(counts)
```

Function to convert Ensmbl ID to Zebrafish gene symbol

```
Genes <- rownames(counts_mat)
ensmbl_to_gene_danio <- function(x){
  require(biomaRt)
  mart <- useMart(biomart = "ensembl", dataset = "drerio_gene_ensembl")
  # query biomart
  results <- getBM(attributes = c("ensembl_gene_id", "zfin_id_symbol"),
                                  filters = "ensembl_gene_id", values = x,
                                  mart = mart)
  results
                                      }
gene_symbols_ensmbl_df <- ensmbl_to_gene_danio(Genes)
```

```
## Loading required package: biomaRt
```

```
head(gene_symbols_ensmbl_df)
```

```
##      ensembl_gene_id   zfin_id_symbol
## 1 ENSDARG00000000018             nrf1
## 2 ENSDARG00000000019            ube2h
## 3 ENSDARG00000000423 si:ch73-314g15.3
## 4 ENSDARG00000000442          slc39a13
## 5 ENSDARG00000000460            nitr2b
## 6 ENSDARG00000000767             spi1b
```

Function to convert Zebrafish Ensmbl ID to Mouse genes

```
zgGenes <-  rownames(counts_mat)
# Basic function to convert zebrafish to human gene names


convertDanioGeneList_Mouse <- function(x){
  require("biomaRt")
  require("curl")
  mouse = useMart("ensembl", dataset = "mmusculus_gene_ensembl")
  danio = useMart("ensembl", dataset = "drerio_gene_ensembl")
  genesV2 = getLDS(attributes = c("ensembl_gene_id", "zfin_id_symbol"), filters = "ensembl_gene_id",
                                values = x , mart = danio, attributesL = c("mgi_symbol", "ensembl_gene_id", "description"),
                                martL = mouse, uniqueRows=T)

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

  # Print the first 6 genes found to the screen
  return(genesV2)
}

Mouse_Genes <- convertDanioGeneList_Mouse(zgGenes)
```

```
## Loading required package: curl
```

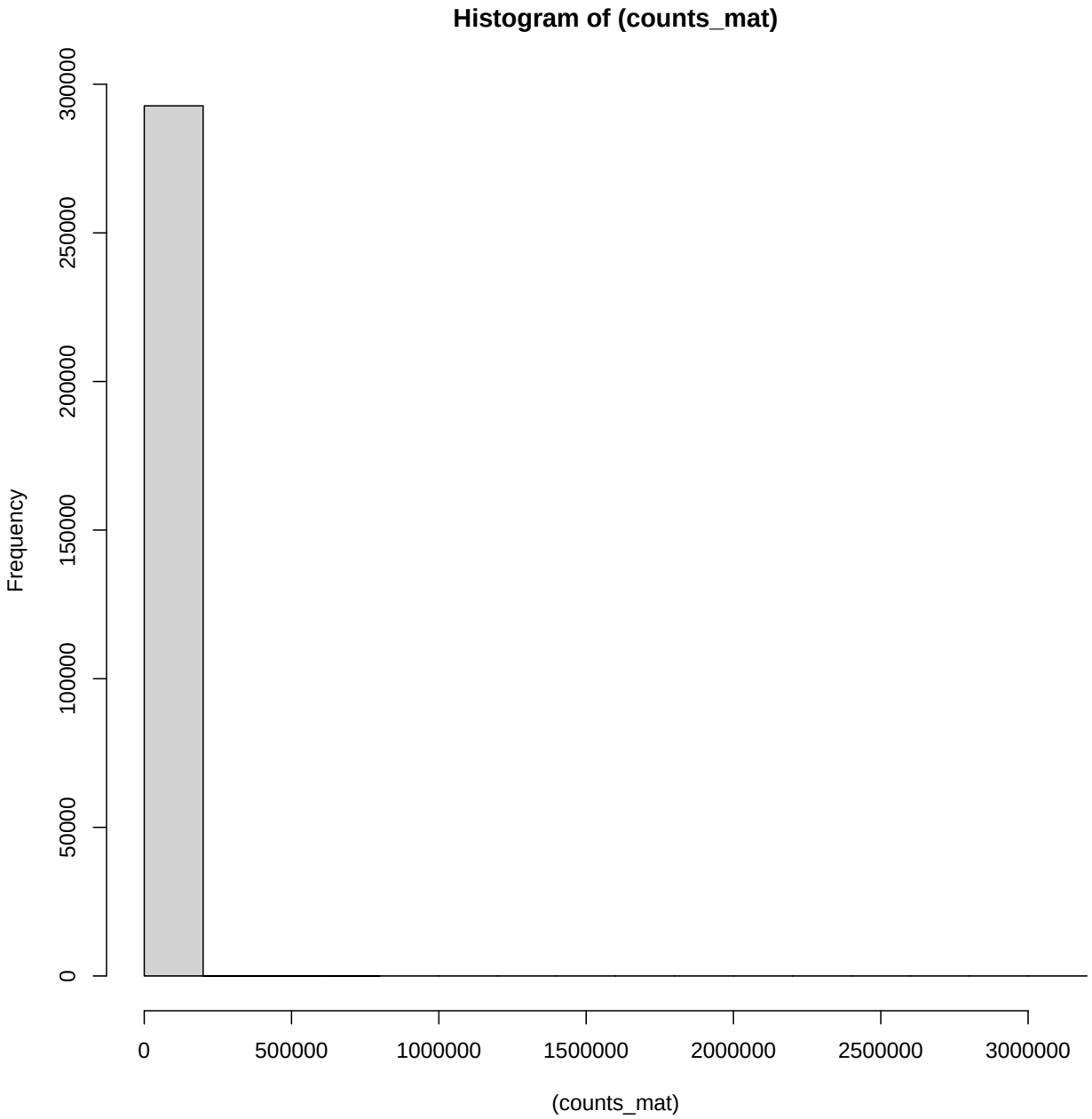
```
## Using libcurl 7.68.0 with OpenSSL/1.1.1f
```

```
head(Mouse_Genes)
```

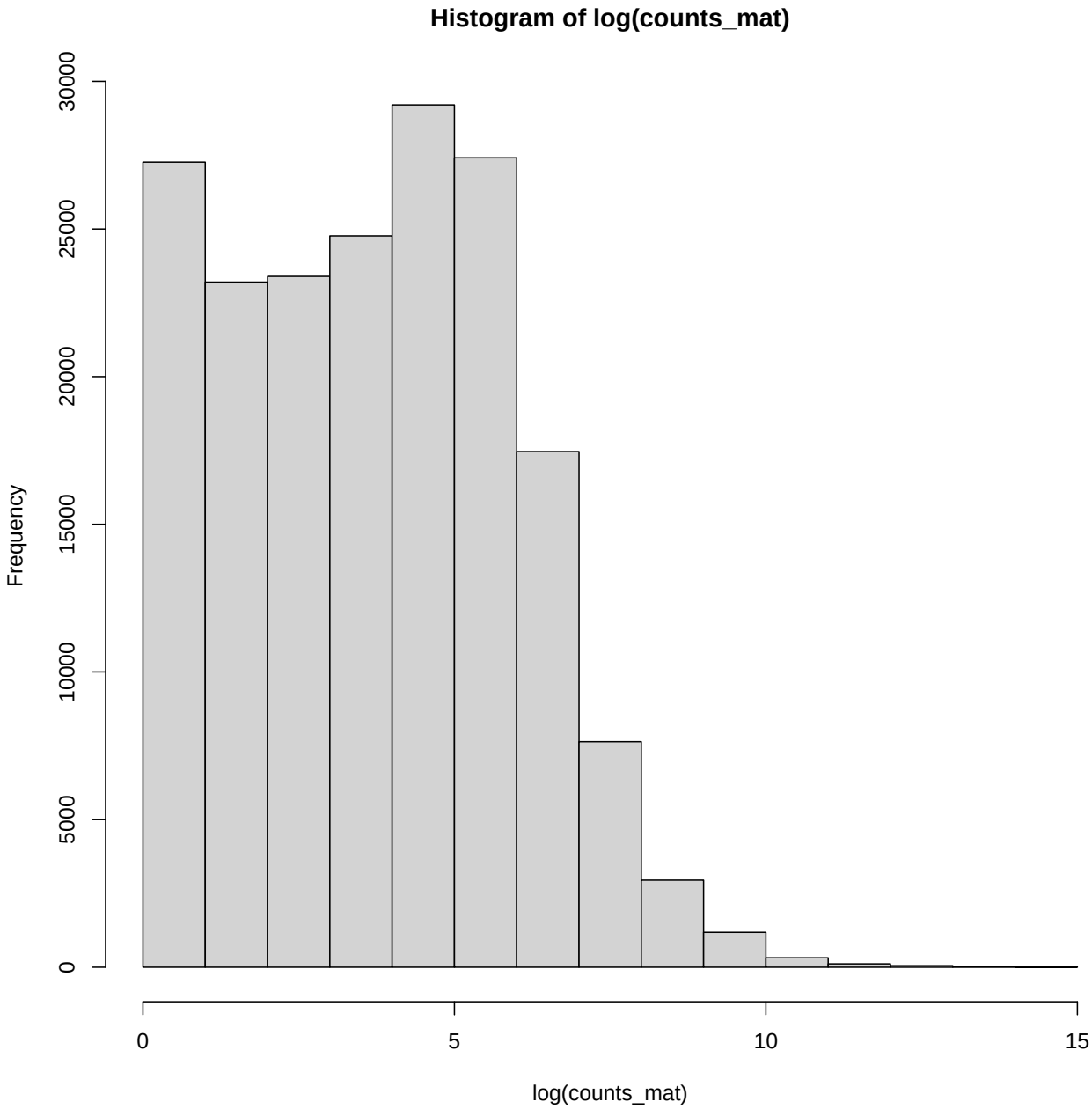
```
##   EnsmblID_Zebrafish ZFIN.symbol MGI.symbol     EnsmblID_Mouse
## 1 ENSDARG00000035697       spinb    Gm20816 ENSMUSG00000095634
## 2 ENSDARG00000035697       spinb    Gm33815 ENSMUSG00000102388
## 3 ENSDARG00000035697       spinb    Gm28079 ENSMUSG00000099550
## 4 ENSDARG00000035697       spinb    Gm29644 ENSMUSG00000099530
## 5 ENSDARG00000035697       spinb    Gm21118 ENSMUSG00000094052
## 6 ENSDARG00000063908      mt-co2     mt-Co2 ENSMUSG00000064354
##                                                                Gene.description
## 1                           predicted gene, 20816 [Source:MGI Symbol;Acc:MGI:5434172]
## 2                           predicted gene, 33815 [Source:MGI Symbol;Acc:MGI:5592974]
## 3                            predicted gene 28079 [Source:MGI Symbol;Acc:MGI:5578785]
## 4                            predicted gene 29644 [Source:MGI Symbol;Acc:MGI:5580350]
## 5                           predicted gene, 21118 [Source:MGI Symbol;Acc:MGI:5434473]
## 6 mitochondrially encoded cytochrome c oxidase II [Source:MGI Symbol;Acc:MGI:102503]
```

Histogram for counts ditribution

```
hist((counts_mat))
```


Histogram of (counts_mat)
Frequency
0
50000
100000
150000
200000
250000
300000
0
500000
1000000
1500000
2000000
2500000
3000000
(counts_mat)

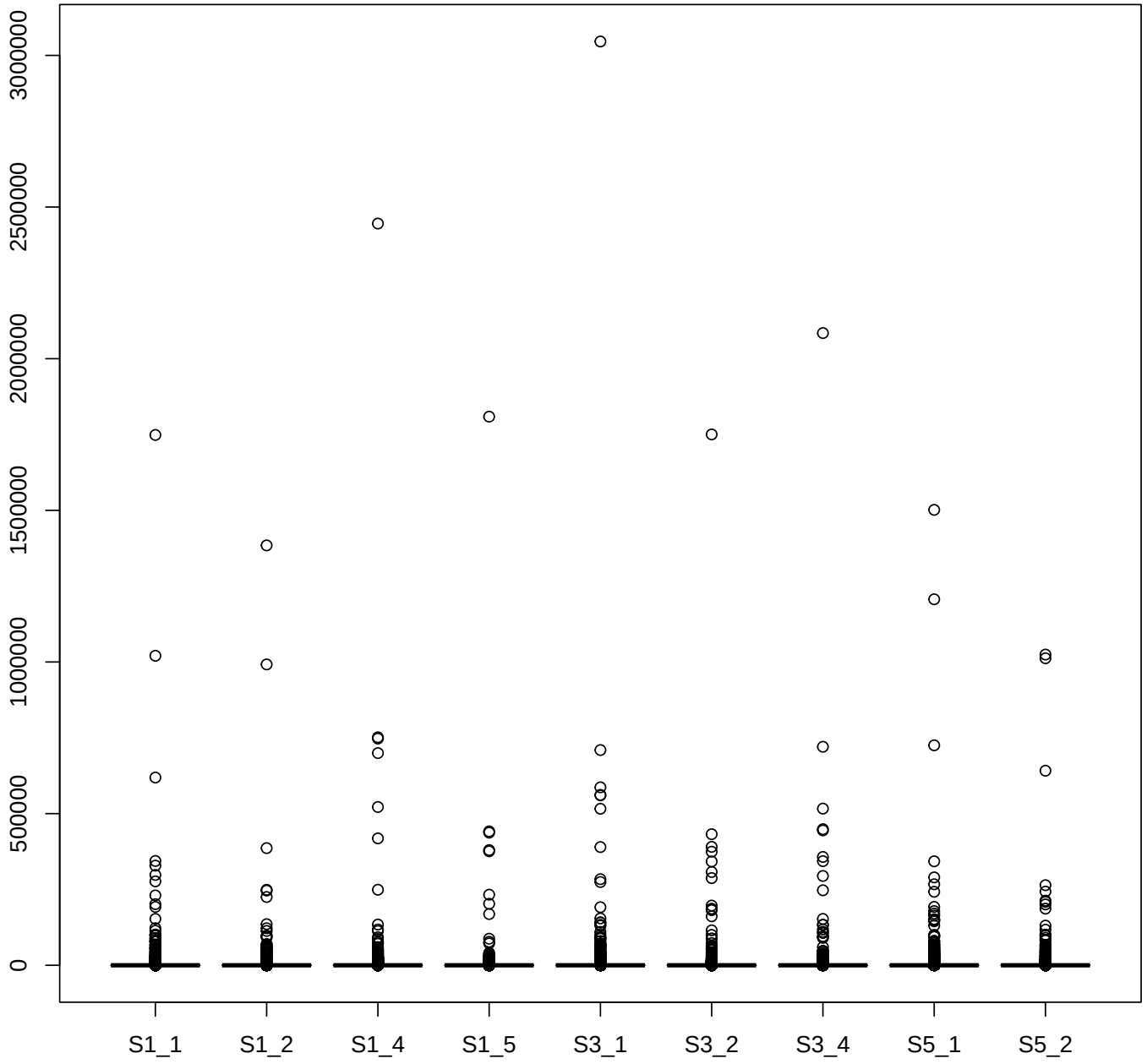
```
hist(log(counts_mat))
```



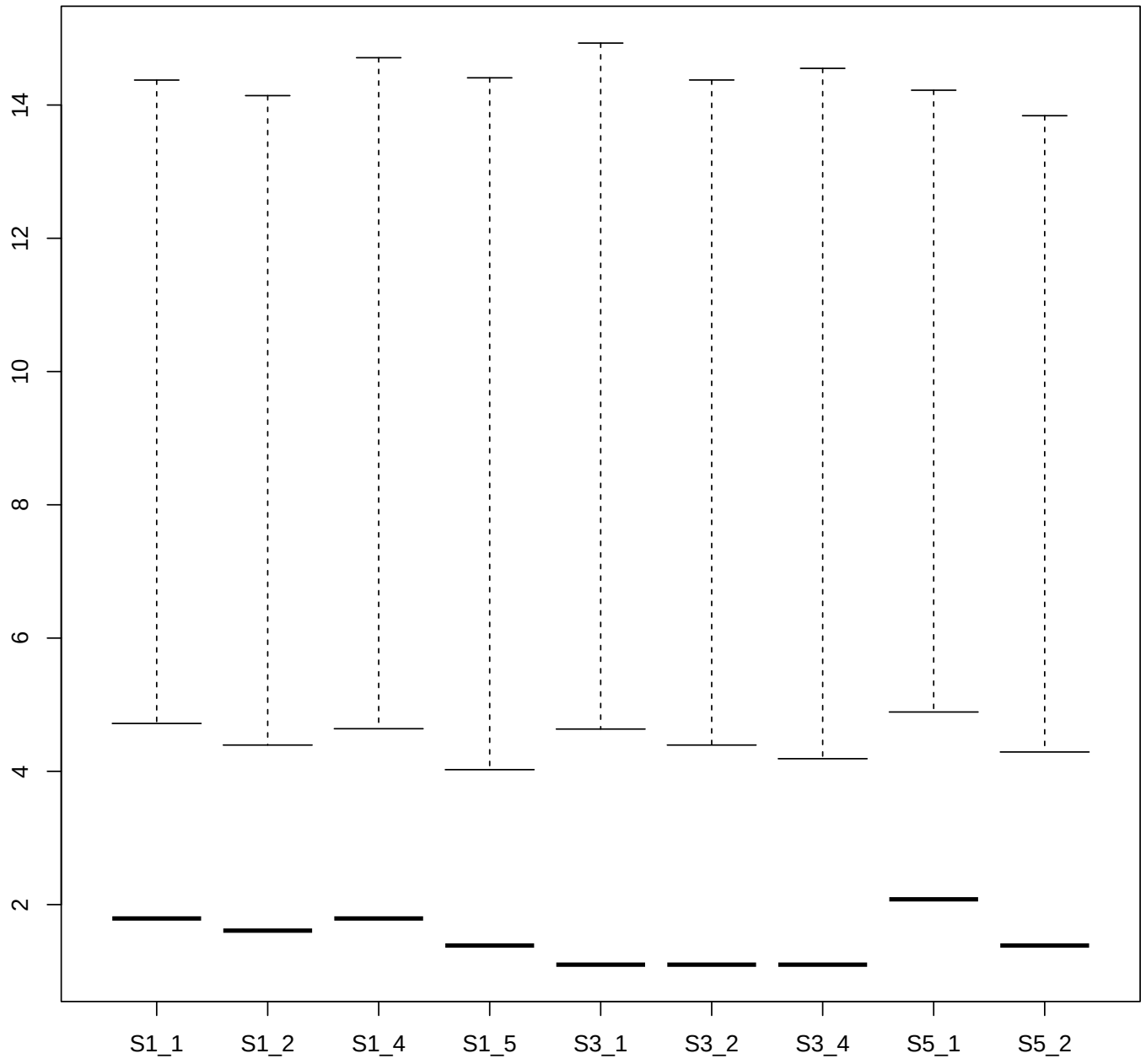
boxplot to visualize the counts ditribu-

tion

```
boxplot(counts_mat)
```


0
500000
1000000
1500000
2000000
2500000
3000000
S1_1
S1_2
S1_4
S1_5
S3_1
S3_2
S3_4
S5_1
S5_2

```
boxplot(log(counts_mat))
```



check if any valuse is NA

```
sum(is.na(counts_mat))
```

```
## [1] 0
```

#Function for PCA

```
PCA <- function(mat,color_pca="",shape_pca= "", label_pca= "",save_plot= "no", name_of_plot= "PCA", comp1=1, comp2=2){
  #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 = 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")) +
          # coord_fixed()+
    NULL
}
if (save_plot== "yes"){
png(filename =paste0(name_of_plot,".png"),res = 300,width = 2560,height = 1440)
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")) +
          # coord_fixed()+
  NULL
print(pca_p)
dev.off()
 }
# Sys.sleep(0.2)
cat("Done")
print(pca_p)
```

```
  #return(pca_p)
}
```

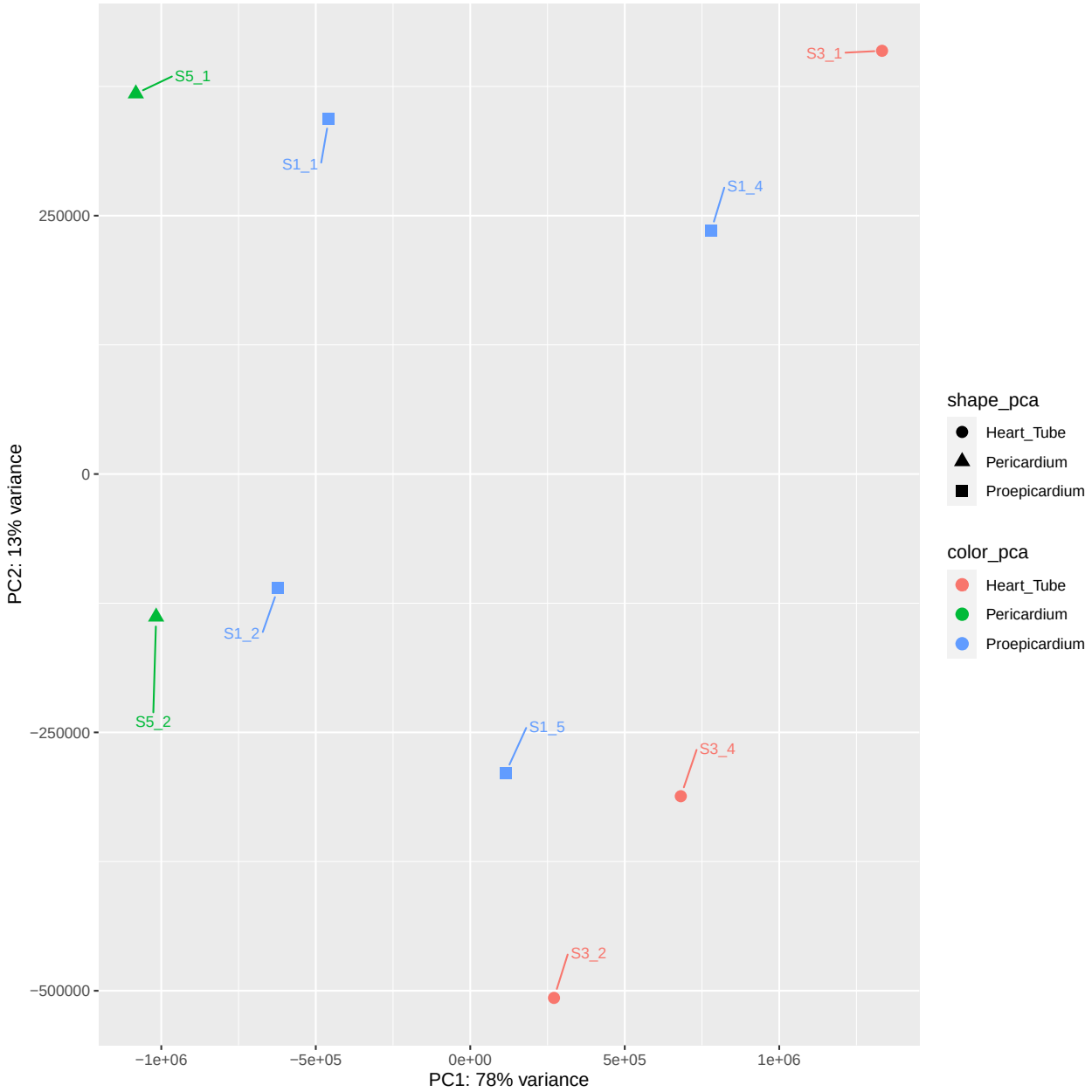
Draw PCA and check the distribution(without normalization)

```
PCA(mat = counts_mat, color_pca = metadata_smart$Tissue,
    shape_pca =   metadata_smart$Tissue, label_pca = rownames(metadata_smart), save_plot = "no")
```

```
## PCA running...
## Percents calculated...
## Data frame built...
## Plotting...
## [1] "no"
```

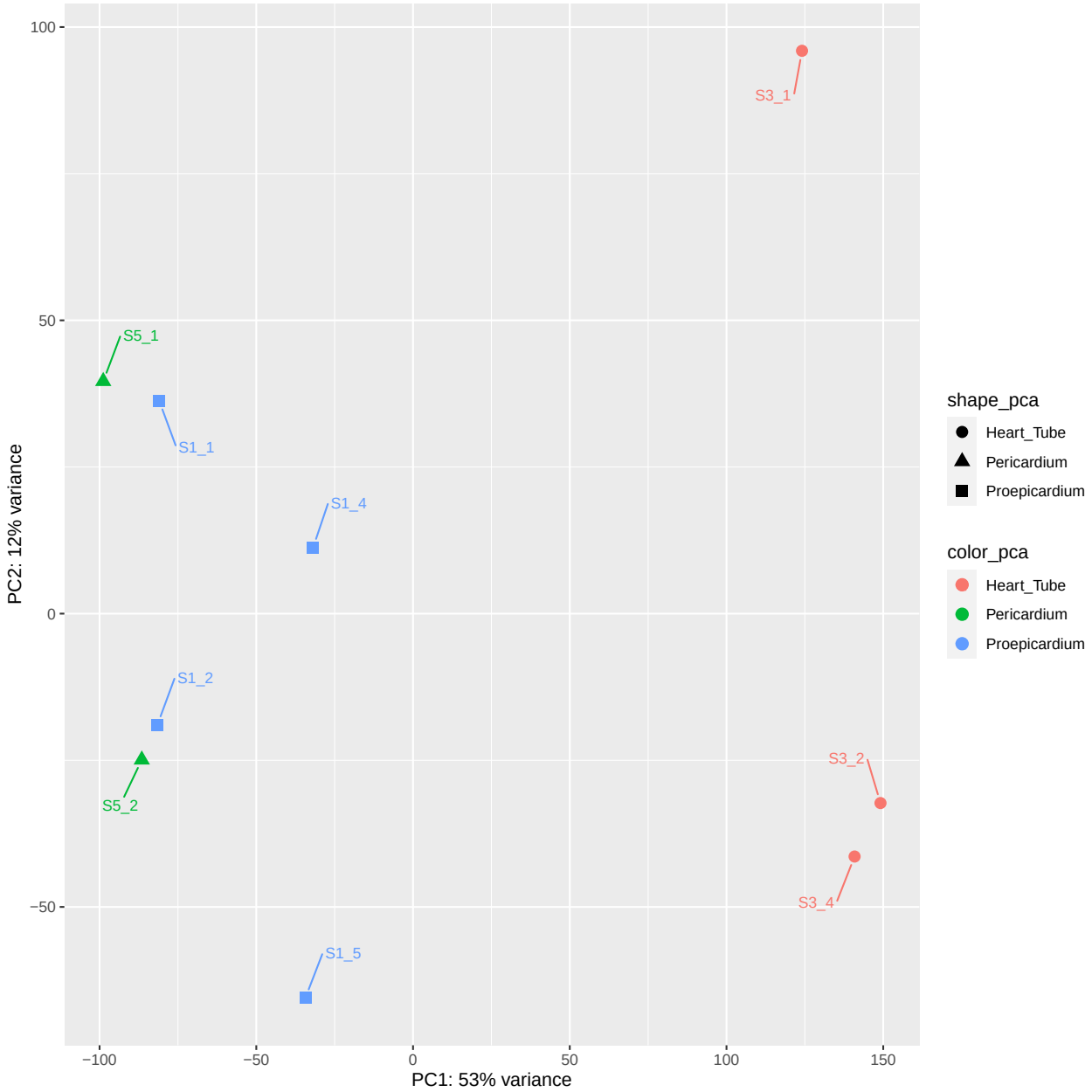
```
## Loading required package: ggrepel
```

```
## Done
```


S3_1
S5_1
S1_1
S1_4
250000
0
S1_2
S5_2
S1_5
S3_4
−250000
S3_2
−500000
−1e+06
−5e+05
0e+00
5e+05
1e+06
PC1: 78% variance
PC2: 13% variance
shape_pca
Heart_Tube
Pericardium
Proepicardium
color_pca
Heart_Tube
Pericardium
Proepicardium

```
PCA(mat = log(counts_mat+1), color_pca = metadata_smart$Tissue,
    shape_pca =   metadata_smart$Tissue, label_pca = rownames(metadata_smart), save_plot = "no")
```

```
## PCA running...
## Percents calculated...
## Data frame built...
## Plotting...
## [1] "no"
## Done
```



Make DESeq2 object

```
dds <- DESeqDataSetFromMatrix(countData=counts_mat,
                              colData=metadata_smart,
                              design=~Tissue)
```

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

Remove low expressing genes

```
keep <- rowSums(counts(dds)) >= 10
dds_filtered <- dds[keep,]
dds
```

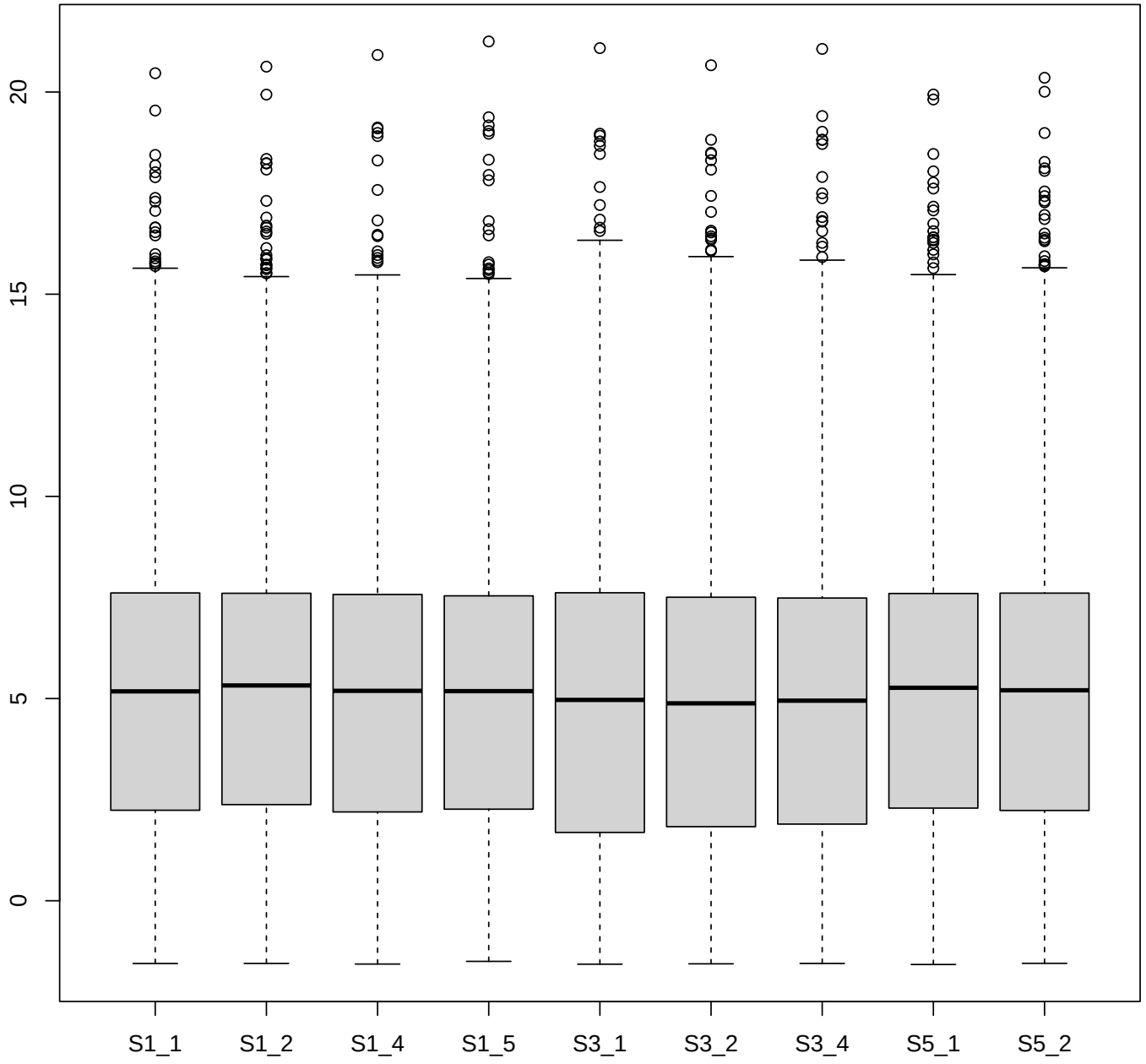
```
## class: DESeqDataSet
## dim: 32536 9
## metadata(1): version
## assays(1): counts
## rownames(32536): ENSDARG00000102141 ENSDARG00000102123 ... Turquoise
##   YFP
## rowData names(0):
## colnames(9): S1_1 S1_2 ... S5_1 S5_2
## colData names(2): Sample_Name Tissue
```

```
dds_filtered
```

```
## class: DESeqDataSet
## dim: 21486 9
## metadata(1): version
## assays(1): counts
## rownames(21486): ENSDARG00000102141 ENSDARG00000102123 ...
##   Egfp_Kozak_3_UTR mCherry
## rowData names(0):
## colnames(9): S1_1 S1_2 ... S5_1 S5_2
## colData names(2): Sample_Name Tissue
```

rlog normalization and check with boxplot

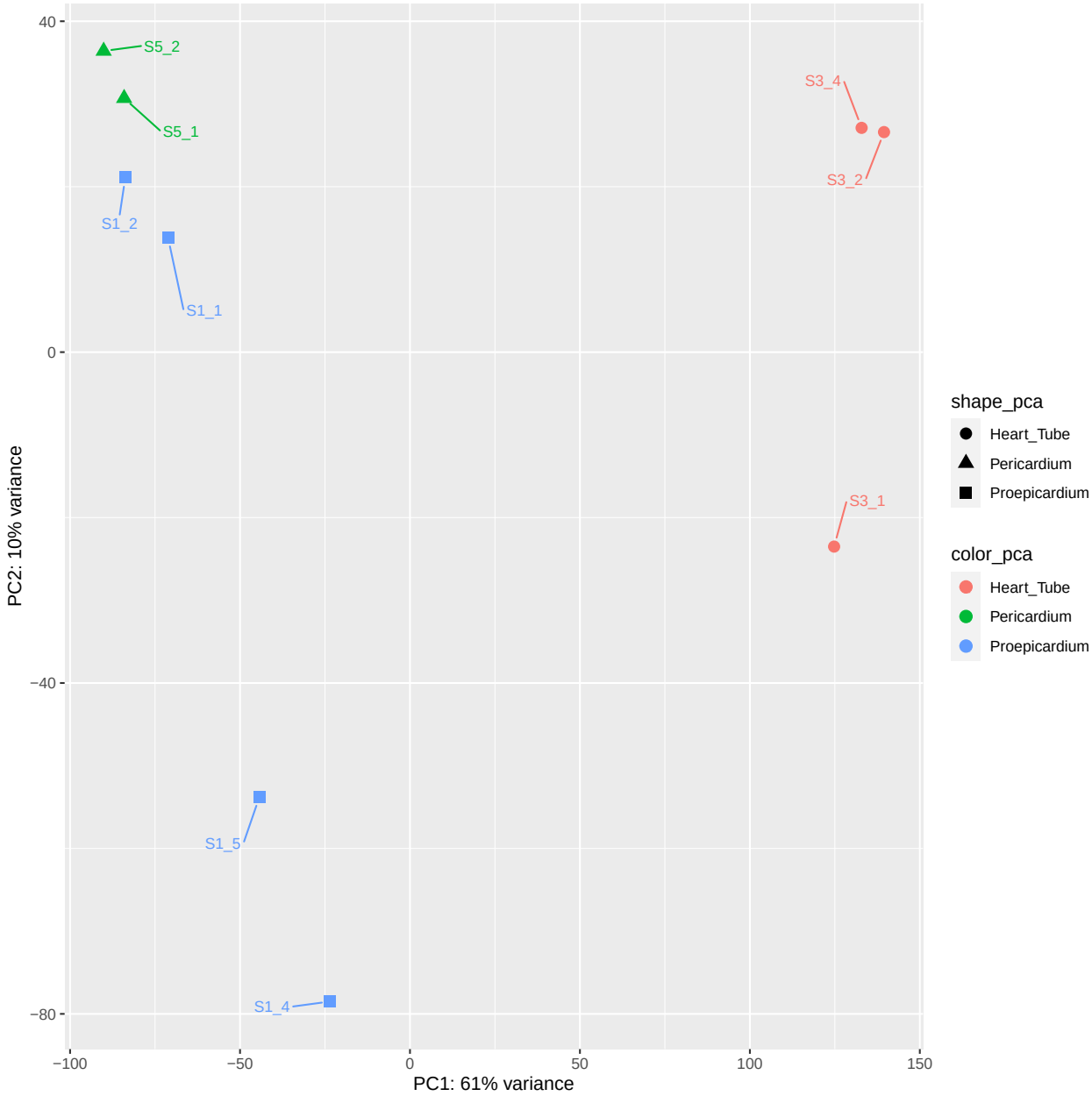
```
dds_filtered_rlog <- rlog(dds_filtered)
boxplot(assay(dds_filtered_rlog))
```



PCA After normalization

```
PCA(mat = assay(dds_filtered_rlog), color_pca = metadata_smart$Tissue,
    shape_pca =   metadata_smart$Tissue, label_pca = rownames(metadata_smart), save_plot = "no")
```

```
## PCA running...
## Percents calculated...
## Data frame built...
## Plotting...
## [1] "no"
## Done
```



PCA function with circles

```
library(ggbiplot)
```

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

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

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

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

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

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

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

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

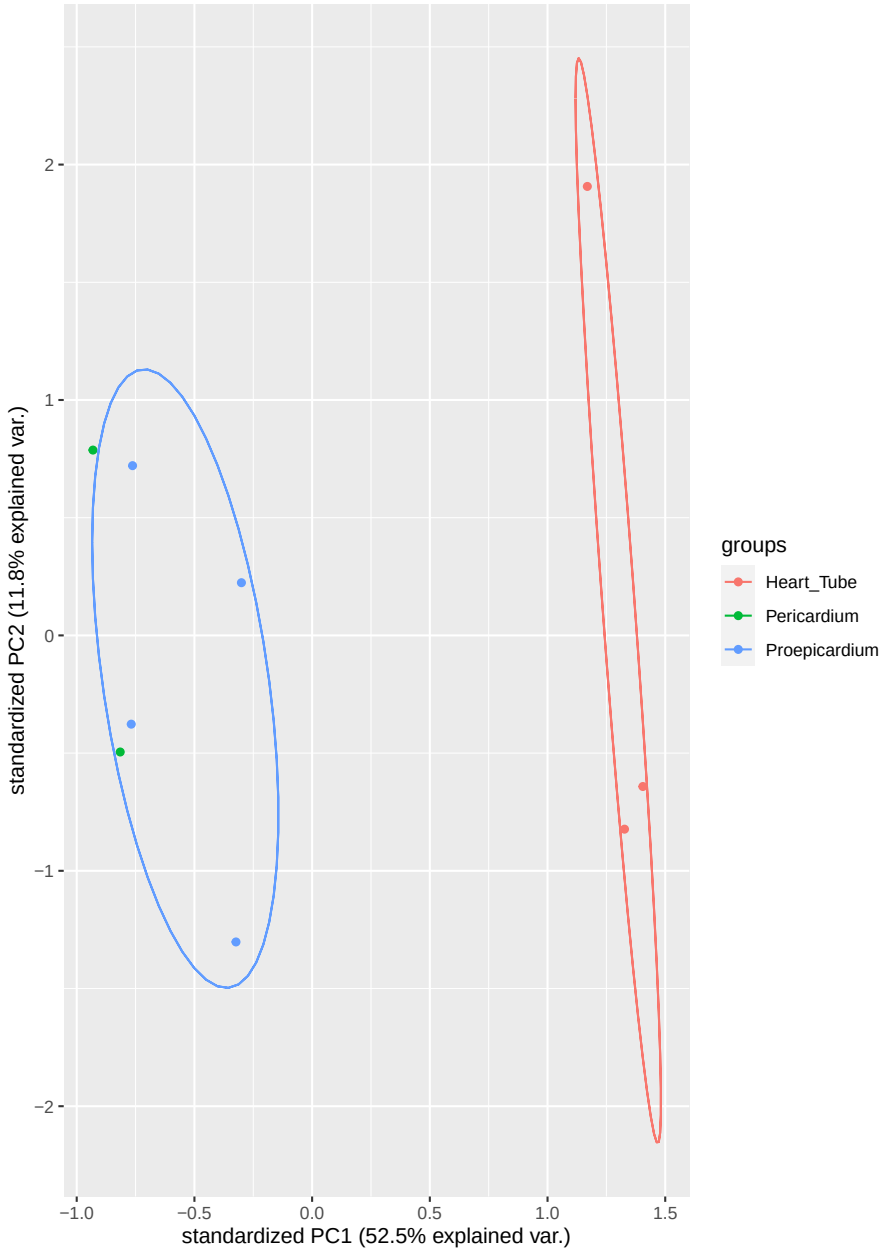
```
## Loading required package: scales

## Loading required package: grid
```

```
PCA_biplot <- function(mat,color_pca="",shape_pca= "", label_pca= "",
                       save_plot= "no", name_of_plot= "PCA", comp1=1, comp2=2){
  #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")


  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)
  require(ggbiplot)

    pca_p <- ggbiplot(pcobj = pca_dt, ellipse = T, groups = color_pca, var.axes = F)

  cat("Done")
  print(pca_p)

}
```

PCA with cirlces after normalization

```
PCA_biplot(mat = log(counts_mat+1), color_pca = metadata_smart$Tissue,
           shape_pca = metadata_smart$Tissue, label_pca = metadata_smart$Sample_Name, save_plot = "no")
```

```
## PCA running...
## Data frame built...
## Plotting...
## [1] "no"
## Done
```


standardized PC2 (11.8% explained var.)
2
1
0
−1
−2
−1.0
−0.5
0.0
0.5
1.0
1.5
standardized PC1 (52.5% explained var.)
groups
Heart_Tube
Pericardium
Proepicardium

```
design(dds_filtered)
```

```
## ~Tissue
```

Run DEseq2

```
dds_filtered <- DESeq(dds_filtered, parallel = T)
```

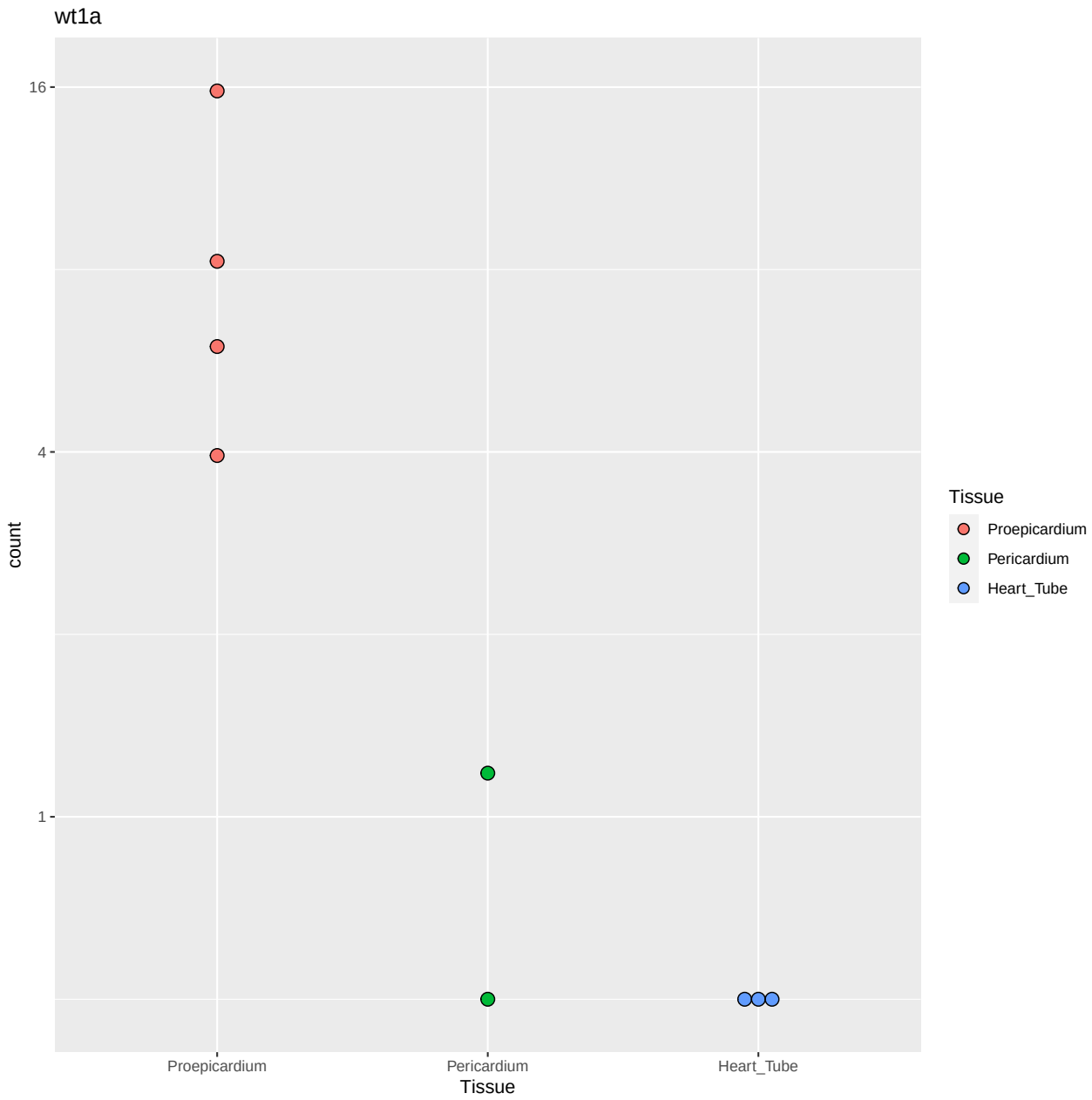
```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates: 10 workers

## mean-dispersion relationship

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

```
resultsNames(dds_filtered)
```

```
## [1] "Intercept"                           "Tissue_Pericardium_vs_Heart_Tube"
## [3] "Tissue_Proepicardium_vs_Heart_Tube"
```

```
df <- plotCounts(dds_filtered, gene=c("ENSDARG00000031420"), intgroup = "Tissue",returnData = T)
df$Tissue <- factor(df$Tissue, levels=c("Proepicardium","Pericardium","Heart_Tube"))

ggplot(df, aes(x= Tissue, y= count, fill= Tissue, shape= Tissue))+
  geom_dotplot(binaxis = "y", binwidth = 0.075, stackdir = "center", size= 1)+

    scale_y_continuous(trans='log2')+ labs(title = "wt1a")+
    NULL
```

```
## Warning: Ignoring unknown parameters: size
```


wt1a
16
4
1
count
Proepicardium
Pericardium
Heart_Tube
Tissue
Tissue
Proepicardium
Pericardium
Heart_Tube

```
dev.copy(
svg,
file = paste0("wt1a_expression_outlier_removed.svg"),
width = 10,
height = 8
)
```

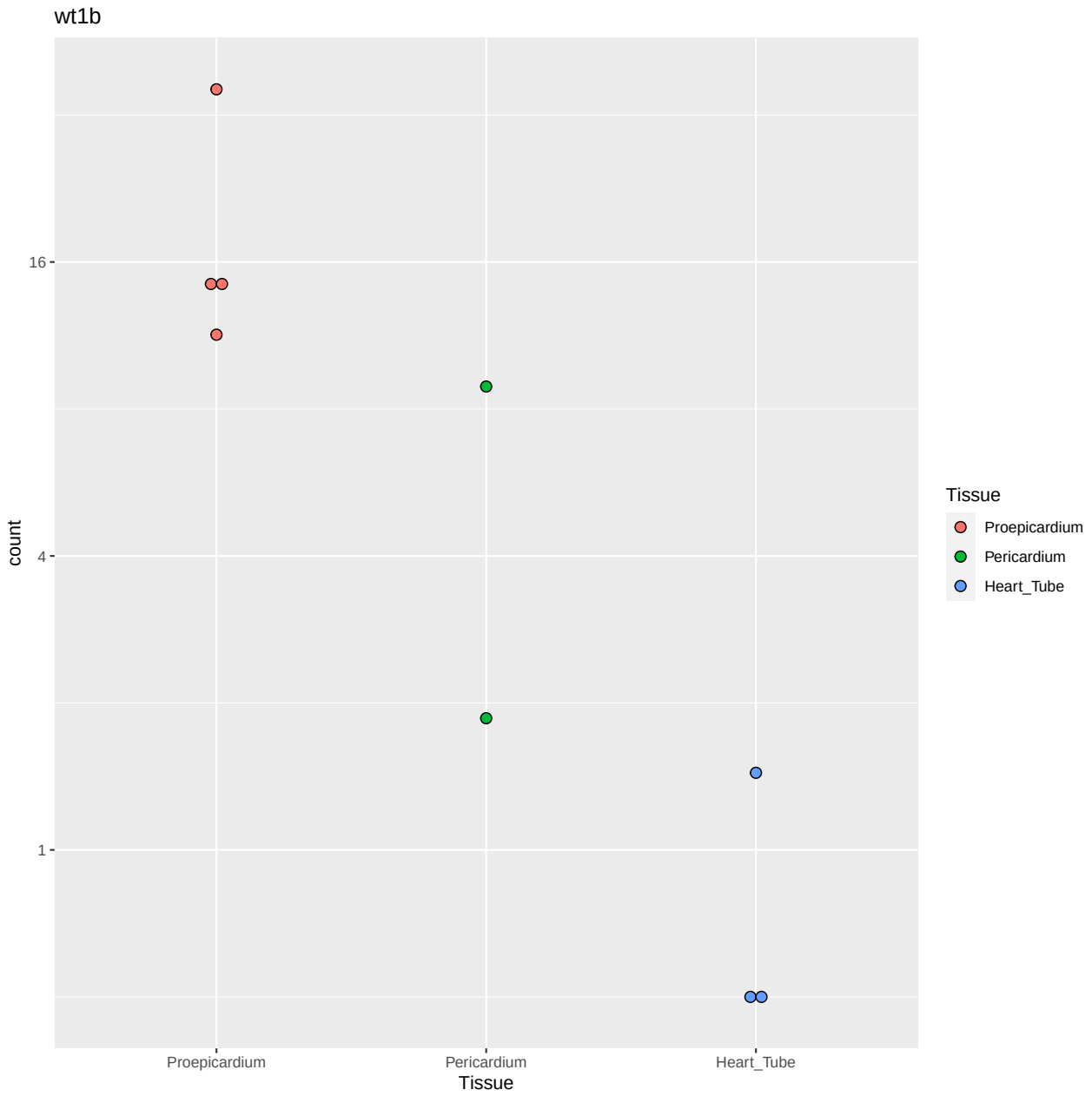
```
## svg
##   3
```

```
dev.off ()
```

```
## pdf
##   2
```

```
df <- plotCounts(dds_filtered, gene=c("ENSDARG00000007990"), intgroup = "Tissue",returnData = T)
df$Tissue <- factor(df$Tissue, levels=c("Proepicardium","Pericardium","Heart_Tube"))

ggplot(df, aes(x= Tissue, y= count, fill= Tissue, shape= Tissue))+
  geom_dotplot(binaxis = "y", binwidth = 0.075, stackdir = "center", size= 1)+

    scale_y_continuous(trans='log2')+ labs(title = "wt1b")+

  NULL
```

```
## Warning: Ignoring unknown parameters: size
```


wt1b
16
4
1
count
Proepicardium
Pericardium
Heart_Tube
Tissue
Tissue
Proepicardium
Pericardium
Heart_Tube

```
dev.copy(
svg,
file = paste0("wt1b_outlier_removed_outlier_removed.svg"),
width = 10,
height = 8
)
```

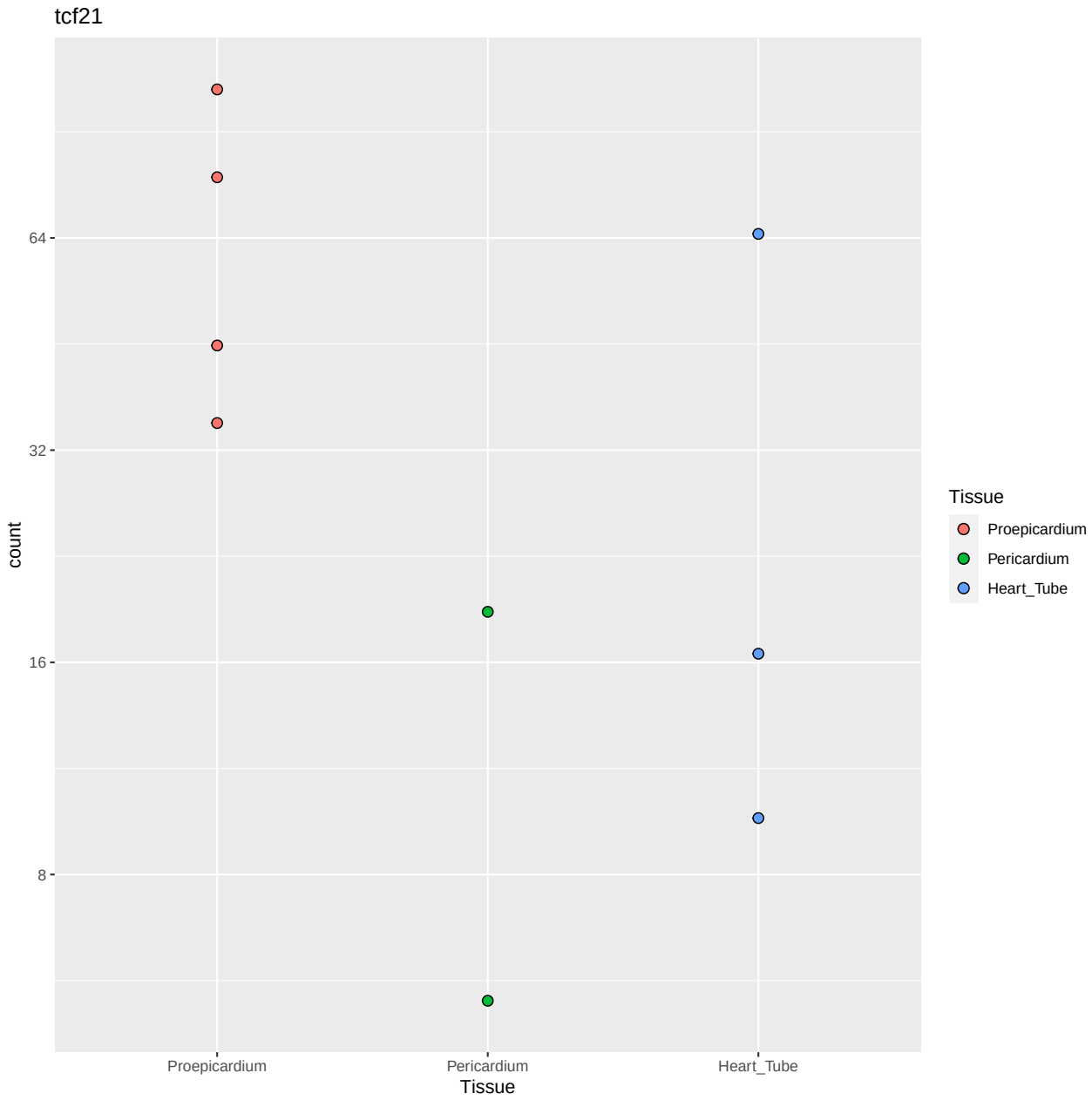
```
## svg
##   3
```

```
dev.off ()
```

```
## pdf
##   2
```

```
# plotCounts(dds_filtered, gene="", intgroup = "Tissue", main = "")
df <- plotCounts(dds_filtered, gene=c("ENSDARG00000036869"),
                 intgroup = "Tissue",returnData = T)
df$Tissue <- factor(df$Tissue, levels=c("Proepicardium","Pericardium","Heart_Tube"))

ggplot(df, aes(x= Tissue, y= count, fill= Tissue, shape= Tissue))+
  geom_dotplot(binaxis = "y", binwidth = 0.05, stackdir = "center", size= 1)+

    scale_y_continuous(trans='log2')+ labs(title = "tcf21")+

  NULL
```

```
## Warning: Ignoring unknown parameters: size
```


tcf21
count
64
32
16
8
Proepicardium
Pericardium
Heart_Tube
Tissue
Tissue
Proepicardium
Pericardium
Heart_Tube

```
dev.copy(
svg,
file = paste0("tcf21_outlier_removed_outlier_removed.svg"),
width = 10,
height = 8
)
```

```
## svg
##   3
```

```
dev.off ()
```

```
## pdf
##   2
```

Run DEGs for each comparison

```
results_PC_HT <- lfcShrink(dds_filtered, contrast = c("Tissue","Pericardium","Heart_Tube"),
                           alpha=0.05, parallel=TRUE, 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
```

```
print("lfcshrinakge done...")
```

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

```
results_PC_HT_df <- as.data.frame(results_PC_HT)
head(results_PC_HT_df)
```

```
##                     baseMean log2FoldChange     lfcSE    pvalue      padj
## ENSDARG00000102141 4.575090     -0.1337083 0.8074058 0.7684672 0.8736181
```

```
## ENSDARG00000102123 1.772040     -1.4434930 2.1349382 0.1134416        NA
## ENSDARG00000114503 2.035033      0.3024214 1.1884707 0.6123065        NA
## ENSDARG00000098311 1.861369      0.6597230 1.1861607 0.2757776        NA
## ENSDARG00000104839 1.251609     -0.6506800 1.5017966 0.3529728        NA
## ENSDARG00000100143 1.667953      0.3260985 1.3043468 0.6063156        NA
```

```
results_PC_HT_df <- results_PC_HT_df %>% dplyr::arrange(padj)
head(results_PC_HT_df)
```

```
##                      baseMean log2FoldChange     lfcSE       pvalue
## ENSDARG00000005842   2809.094      -8.143145 0.4106901 3.857513e-89
## ENSDARG00000052958  45495.854     -10.948584 0.6322573 2.241918e-69
## ENSDARG00000052960  47831.752     -10.002481 0.5999525 1.997830e-64
## ENSDARG00000019096 108434.509     -10.524819 0.6351641 8.750278e-64
## ENSDARG00000029439   2544.258      -6.281478 0.3886497 7.533023e-60
## ENSDARG00000032976  53515.590     -10.330769 0.6491395 4.167480e-59
##                            padj
## ENSDARG00000005842 7.459658e-85
## ENSDARG00000052958 2.167710e-65
## ENSDARG00000052960 1.287802e-60
## ENSDARG00000019096 4.230322e-60
## ENSDARG00000029439 2.913472e-56
## ENSDARG00000032976 1.343179e-55
```

```
write.csv(results_PC_HT_df, "results_PC_vs_HT_outlier_removed.csv")
```

```
head(Mouse_Genes)
```

```
##   EnsmblID_Zebrafish ZFIN.symbol MGI.symbol     EnsmblID_Mouse
## 1 ENSDARG00000035697       spinb    Gm20816 ENSMUSG00000095634
## 2 ENSDARG00000035697       spinb    Gm33815 ENSMUSG00000102388
## 3 ENSDARG00000035697       spinb    Gm28079 ENSMUSG00000099550
## 4 ENSDARG00000035697       spinb    Gm29644 ENSMUSG00000099530
## 5 ENSDARG00000035697       spinb    Gm21118 ENSMUSG00000094052
## 6 ENSDARG00000063908      mt-co2     mt-Co2 ENSMUSG00000064354
##                                                                Gene.description
## 1                         predicted gene, 20816 [Source:MGI Symbol;Acc:MGI:5434172]
```

```
## 2                          predicted gene, 33815 [Source:MGI Symbol;Acc:MGI:5592974]
## 3                           predicted gene 28079 [Source:MGI Symbol;Acc:MGI:5578785]
## 4                           predicted gene 29644 [Source:MGI Symbol;Acc:MGI:5580350]
## 5                          predicted gene, 21118 [Source:MGI Symbol;Acc:MGI:5434473]
## 6 mitochondrially encoded cytochrome c oxidase II [Source:MGI Symbol;Acc:MGI:102503]
```

Pericardium_vs_HeartTube

```
results_PC_HT_df_annotated <- results_PC_HT_df
results_PC_HT_df_annotated$ensmbl_id <- rownames(results_PC_HT_df_annotated)
head(results_PC_HT_df_annotated)
```

```
##                     baseMean log2FoldChange     lfcSE       pvalue
## ENSDARG00000005842   2809.094      -8.143145 0.4106901 3.857513e-89
## ENSDARG00000052958  45495.854     -10.948584 0.6322573 2.241918e-69
## ENSDARG00000052960  47831.752     -10.002481 0.5999525 1.997830e-64
## ENSDARG00000019096 108434.509     -10.524819 0.6351641 8.750278e-64
## ENSDARG00000029439   2544.258      -6.281478 0.3886497 7.533023e-60
## ENSDARG00000032976  53515.590     -10.330769 0.6491395 4.167480e-59
##                            padj          ensmbl_id
## ENSDARG00000005842 7.459658e-85 ENSDARG00000005842
## ENSDARG00000052958 2.167710e-65 ENSDARG00000052958
## ENSDARG00000052960 1.287802e-60 ENSDARG00000052960
## ENSDARG00000019096 4.230322e-60 ENSDARG00000019096
## ENSDARG00000029439 2.913472e-56 ENSDARG00000029439
## ENSDARG00000032976 1.343179e-55 ENSDARG00000032976
```

```
results_PC_HT_df_annotated <- merge(results_PC_HT_df_annotated, Mouse_Genes,
                                    by.x= "ensmbl_id", by.y= "EnsmblID_Zebrafish", all.x=T)
head(results_PC_HT_df_annotated)
```

```
##            ensmbl_id   baseMean log2FoldChange     lfcSE       pvalue
## 1              dsRed 198.380076    -1.35140010 0.9897979 2.594589e-02
## 2   Egfp_Kozak_3_UTR   1.171878     0.00000000 1.6066765 1.000000e+00
## 3 ENSDARG00000000001  24.881659     0.02597369 0.5621364 9.399221e-01
## 4 ENSDARG00000000002 212.853165    -2.51743429 0.6873169 2.354339e-05
```

```
## 5 ENSDARG00000000018 345.709491    -0.11745602 0.3195158 5.604041e-01
## 6 ENSDARG00000000019 427.955571    -0.33319024 0.4232174 2.026673e-01
##           padj ZFIN.symbol MGI.symbol     EnsmblID_Mouse
## 1 0.0798238671        <NA>       <NA>               <NA>
## 2           NA        <NA>       <NA>               <NA>
## 3 0.9705213015     slc35a5    Slc35a5 ENSMUSG00000022664
## 4 0.0001977767        <NA>       <NA>               <NA>
## 5 0.7308534453        nrf1       Nrf1 ENSMUSG00000058440
## 6 0.3792142834       ube2h      Ube2h ENSMUSG00000039159
##                                                        Gene.description
## 1                                                                  <NA>
## 2                                                                  <NA>
## 3 solute carrier family 35, member A5 [Source:MGI Symbol;Acc:MGI:1921352]
## 4                                                                  <NA>
## 5        nuclear respiratory factor 1 [Source:MGI Symbol;Acc:MGI:1332235]
## 6     ubiquitin-conjugating enzyme E2H [Source:MGI Symbol;Acc:MGI:104632]
```

```
write.csv(results_PC_HT_df_annotated, "results_PC_vs_HT_annotated_outlier_removed.csv")
```

```
head(results_PC_HT_df)
```

```
##                     baseMean log2FoldChange     lfcSE       pvalue
## ENSDARG00000005842   2809.094      -8.143145 0.4106901 3.857513e-89
## ENSDARG00000052958  45495.854     -10.948584 0.6322573 2.241918e-69
## ENSDARG00000052960  47831.752     -10.002481 0.5999525 1.997830e-64
## ENSDARG00000019096 108434.509     -10.524819 0.6351641 8.750278e-64
## ENSDARG00000029439   2544.258      -6.281478 0.3886497 7.533023e-60
## ENSDARG00000032976  53515.590     -10.330769 0.6491395 4.167480e-59
##                             padj
## ENSDARG00000005842 7.459658e-85
## ENSDARG00000052958 2.167710e-65
## ENSDARG00000052960 1.287802e-60
## ENSDARG00000019096 4.230322e-60
## ENSDARG00000029439 2.913472e-56
## ENSDARG00000032976 1.343179e-55
```

```
genes_of_interest <- c("wt1a", "wt1b", "tcf21")
```

Volcano plot for Pericardium_vs_HeartTube

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

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.2     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x plyr::arrange()           masks clusterProfiler::arrange(), dplyr::arrange()
## x readr::col_factor()       masks scales::col_factor()
## x IRanges::collapse()       masks dplyr::collapse()
## x Biobase::combine()        masks BiocGenerics::combine(), dplyr::combine()
## x purrr::compact()          masks plyr::compact()
## x plyr::count()             masks matrixStats::count(), dplyr::count()
## x plyr::desc()              masks IRanges::desc(), dplyr::desc()
## x purrr::discard()          masks scales::discard()
## x tidyr::expand()           masks S4Vectors::expand()
## x plyr::failwith()          masks dplyr::failwith()
## x clusterProfiler::filter() masks dplyr::filter(), stats::filter()
## x S4Vectors::first()        masks dplyr::first()
## x plyr::id()                masks dplyr::id()
## x dplyr::lag()              masks stats::lag()
## x plyr::mutate()            masks clusterProfiler::mutate(), dplyr::mutate()
## x readr::parse_date()       masks curl::parse_date()
## x BiocGenerics::Position()  masks ggplot2::Position(), base::Position()
```

```
## x purrr::reduce()           masks GenomicRanges::reduce(), IRanges::reduce()
## x plyr::rename()            masks clusterProfiler::rename(), S4Vectors::rename(), dplyr::rename()
## x biomaRt::select()         masks clusterProfiler::select(), dplyr::select()
## x purrr::simplify()         masks clusterProfiler::simplify()
## x clusterProfiler::slice()  masks IRanges::slice(), dplyr::slice()
## x plyr::summarise()         masks clusterProfiler::summarise(), dplyr::summarise()
## x plyr::summarize()         masks dplyr::summarize()
```

```
draw_volcano<- function(fileinput, title, FCcutoff) {
  # read input file
  # drawing plots
  ggplot(data =fileinput , aes(x = log2FoldChange, y = -log10(padj))) +
    # draw lines


    # draw points
    geom_point(x = fileinput$log2FoldChange, y = -log10(fileinput$padj),alpha = 0.5, size = 1,color="lightskyblue1") +
    # draw coloured points
    geom_point(data = fileinput[which(fileinput$padj < 0.05 & fileinput$log2FoldChange < -FCcutoff),],

               aes(x=log2FoldChange, y = -log10(padj)),
               shape = 21, color = "orchid", fill = "orchid",
               alpha = 0.3, size = 1) +
    geom_point(data = fileinput[which(fileinput$padj < 0.05 & fileinput$log2FoldChange >= FCcutoff),],

               aes(x=log2FoldChange, y = -log10(padj)),
               shape = 21, color = "limegreen", fill = "limegreen",
               alpha = 0.3, size = 1) +
    # x axis scale
    scale_x_continuous(breaks = seq(-round(max(abs(fileinput$log2FoldChange))),
                                    round(max(abs(fileinput$log2FoldChange))),by = 1),
      limits = c(-round(max(abs(fileinput$log2FoldChange))),
                 round(max(abs(fileinput$log2FoldChange))))) +
    xlab("log2FoldChange") +
    scale_y_continuous( limits = c(0,round(max(-log10(abs(fileinput$padj+1)))))) +
    ylab("-Log10(pAdjusted)") +

    # set title
    ggtitle(title)+
```

```
    # x and y axis limits
    # black and white theme
    theme_bw() +
     geom_hline(size=1.2,yintercept = -log10(0.05), linetype = "dashed") +
    # geom_hline(yintercept = -log10(0.05), linetype = "dotted", col = "darkgoldenrod") +
    geom_vline(xintercept = FCcutoff, linetype = "dashed")+
    geom_vline(xintercept = -FCcutoff, linetype = "dashed")+
    # 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))
}
```

```
b = draw_volcano(results_PC_HT_df_annotated,"Pericardium_vs_HeartTube", 0.58)
 # print(b)
#Set genes for marking
wt_set<-c("wt1a","wt1b", "tcf21", "myl7")
wt_set
```

```
## [1] "wt1a"  "wt1b"  "tcf21" "myl7"
```

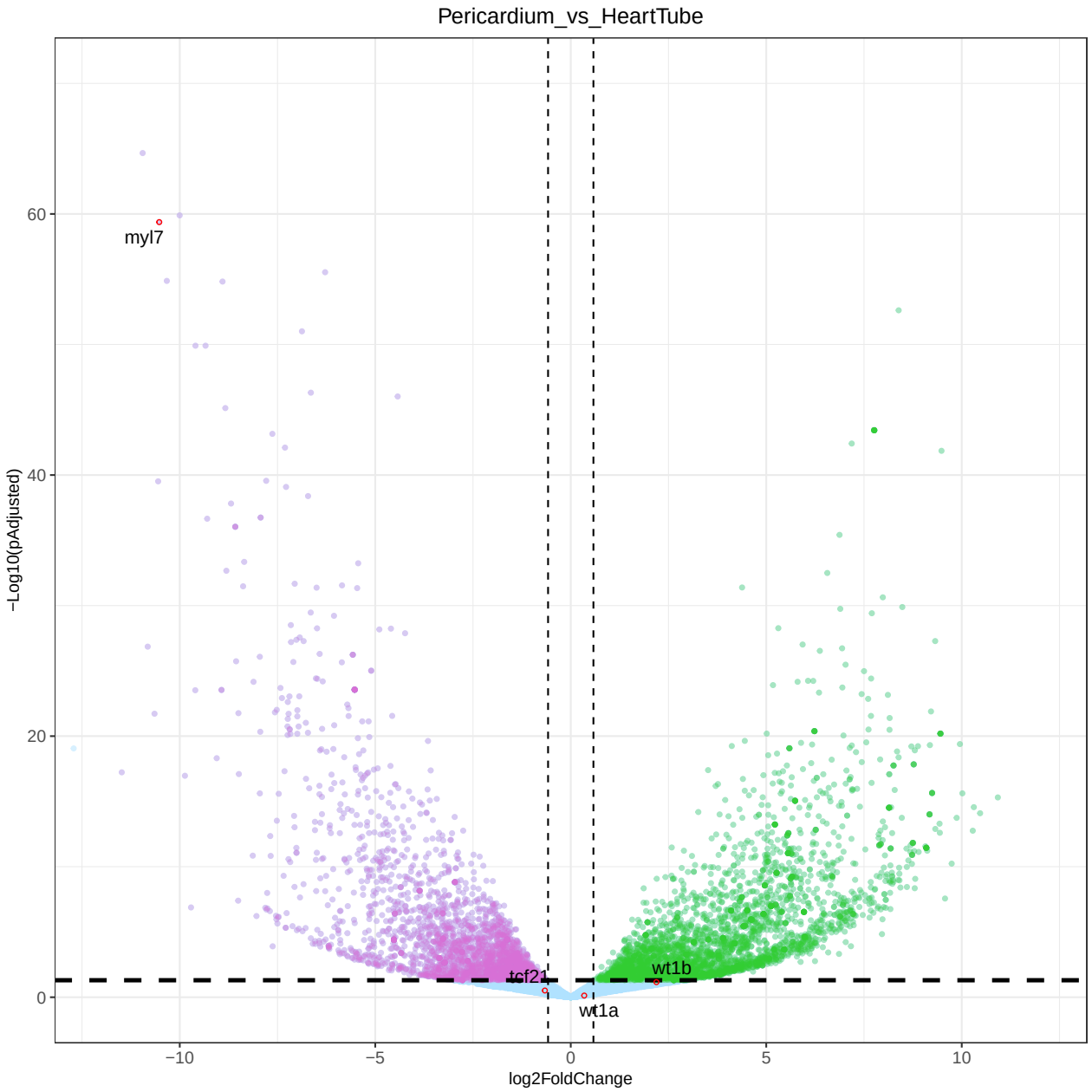
```
wt_geneset<-as.data.frame(results_PC_HT_df_annotated[results_PC_HT_df_annotated$ZFIN.symbol%in%wt_set,])

require(ggrepel)
#Paint the genes in the plot
c= b + geom_point(data=results_PC_HT_df_annotated[results_PC_HT_df_annotated$ZFIN.symbol%in%wt_set,],
                  color="red",size=1, shape= 21) +
  geom_text_repel(data = results_PC_HT_df_annotated[results_PC_HT_df_annotated$ZFIN.symbol%in%wt_set,],
      aes(label=results_PC_HT_df_annotated[results_PC_HT_df_annotated$ZFIN.symbol%in%wt_set,]$ZFIN.symbol),
  )+
   scale_x_continuous(limits = c(-12,12))+scale_y_continuous(limits = c(0,70))+#nudge_x = 0,
#nudge_y = 2,segment.size = 0.1)
NULL
```

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

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

```
  c
```

```
## Warning: Removed 2672 rows containing missing values (geom_point).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```


Pericardium_vs_HeartTube
myl7
tcf21
wt1a
wt1b
−Log10(pAdjusted)
log2FoldChange
60
40
20
0
−10
−5
0
5
10

```
dev.copy(
svg,
file = paste0("Volcano_Pericardium_vs_HeartTube_new_myl7_smalldots_outlier_removed.svg"),
width = 10,
height = 8
)
```

```
## svg
##   3
```

```
dev.off ()
```

```
## pdf
##   2
```

Proepicardium_vs_HeartTube

```
results_PE_HT <- lfcShrink(dds_filtered, contrast = c("Tissue","Proepicardium","Heart_Tube"),
                           alpha=0.05, parallel=TRUE, 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
```

```
print("lfcshrinakge done...")
```

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

```
results_PE_HT_df <- as.data.frame(results_PE_HT)
results_PE_HT_df <- results_PE_HT_df %>% dplyr::arrange(padj)
write.csv(results_PE_HT_df, "results_PE_vs_HT_outlier_removed.csv")
```

```
results_PE_HT_df_annotated <- results_PE_HT_df
results_PE_HT_df_annotated$ensmbl_id <- rownames(results_PE_HT_df_annotated)
head(results_PE_HT_df_annotated)
```

```
##                     baseMean log2FoldChange     lfcSE        pvalue
## ENSDARG00000005842  2809.094      -7.385775 0.3241751 3.066767e-116
## ENSDARG00000052958 45495.854      -9.831413 0.5200904  5.309820e-82
## ENSDARG00000052960 47831.752      -9.037869 0.4965066  5.454674e-76
## ENSDARG00000101251  4191.640      -4.606451 0.2505140  4.728505e-76
## ENSDARG00000029439  2544.258      -5.780291 0.3196757  6.671166e-74
## ENSDARG00000055046  2766.766       8.075420 0.4636389  1.654207e-69
##                            padj          ensmbl_id
## ENSDARG00000005842 6.058092e-112 ENSDARG00000005842
## ENSDARG00000052958  5.244509e-78 ENSDARG00000052958
## ENSDARG00000052960  2.693791e-72 ENSDARG00000052960
## ENSDARG00000101251  2.693791e-72 ENSDARG00000101251
## ENSDARG00000029439  2.635644e-70 ENSDARG00000029439
## ENSDARG00000055046  5.446202e-66 ENSDARG00000055046
```

```
results_PE_HT_df_annotated <- merge(results_PE_HT_df_annotated, Mouse_Genes,
                                    by.x= "ensmbl_id", by.y= "EnsmblID_Zebrafish", all.x=T)
head(results_PE_HT_df_annotated)
```

```
##            ensmbl_id   baseMean log2FoldChange     lfcSE       pvalue
## 1              dsRed 198.380076      0.2454555 0.5496709 4.954773e-01
## 2   Egfp_Kozak_3_UTR   1.171878      1.0163416 1.8105027 1.912357e-01
## 3 ENSDARG00000000001  24.881659     -1.0111884 0.7823146 4.087054e-02
## 4 ENSDARG00000000002 212.853165     -2.6691149 0.5621895 2.195179e-07
## 5 ENSDARG00000000018 345.709491      0.2837750 0.3243788 2.532403e-01
## 6 ENSDARG00000000019 427.955571     -0.6879171 0.3999820 2.359057e-02
##           padj ZFIN.symbol MGI.symbol     EnsmblID_Mouse
## 1 6.537745e-01        <NA>       <NA>               <NA>
## 2           NA        <NA>       <NA>               <NA>
## 3 1.082103e-01     slc35a5    Slc35a5 ENSMUSG00000022664
## 4 2.758496e-06        <NA>       <NA>               <NA>
## 5 4.150370e-01        nrf1       Nrf1 ENSMUSG00000058440
## 6 7.044719e-02       ube2h      Ube2h ENSMUSG00000039159
##                                                 Gene.description
```

```
## 1                                                                    <NA>
## 2                                                                    <NA>
## 3 solute carrier family 35, member A5 [Source:MGI Symbol;Acc:MGI:1921352]
## 4                                                                    <NA>
## 5        nuclear respiratory factor 1 [Source:MGI Symbol;Acc:MGI:1332235]
## 6     ubiquitin-conjugating enzyme E2H [Source:MGI Symbol;Acc:MGI:104632]
```

```
write.csv(results_PE_HT_df_annotated, "results_PE_vs_HT_annotated_outlier_removed.csv")
```

Volcano plot for Proepicardium_vs_HeartTube

```
b = draw_volcano(results_PE_HT_df_annotated,"Proepicardium_vs_HeartTube", 0.58)
# print(b)
#Set genes for marking
wt_set<-c("wt1a","wt1b", "tcf21", "myl7")
wt_set
```

```
## [1] "wt1a"  "wt1b"  "tcf21" "myl7"
```

```
wt_geneset<-as.data.frame(results_PE_HT_df_annotated[results_PE_HT_df_annotated$ZFIN.symbol%in%wt_set,])

require(ggrepel)
#Paint the genes in the plot
c= b + geom_point(data=results_PE_HT_df_annotated[results_PE_HT_df_annotated$ZFIN.symbol%in%wt_set,],
                  color="red",size=1, shape= 21) +
  geom_text_repel(data = results_PE_HT_df_annotated[results_PE_HT_df_annotated$ZFIN.symbol%in%wt_set,],
  aes(label=results_PE_HT_df_annotated[results_PE_HT_df_annotated$ZFIN.symbol%in%wt_set,]$ZFIN.symbol),
  )+
   scale_x_continuous(limits = c(-12,12)) +scale_y_continuous(limits = c(0,70))+
#nudge_x = 0,
#nudge_y = 2,segment.size = 0.1)
  NULL
```

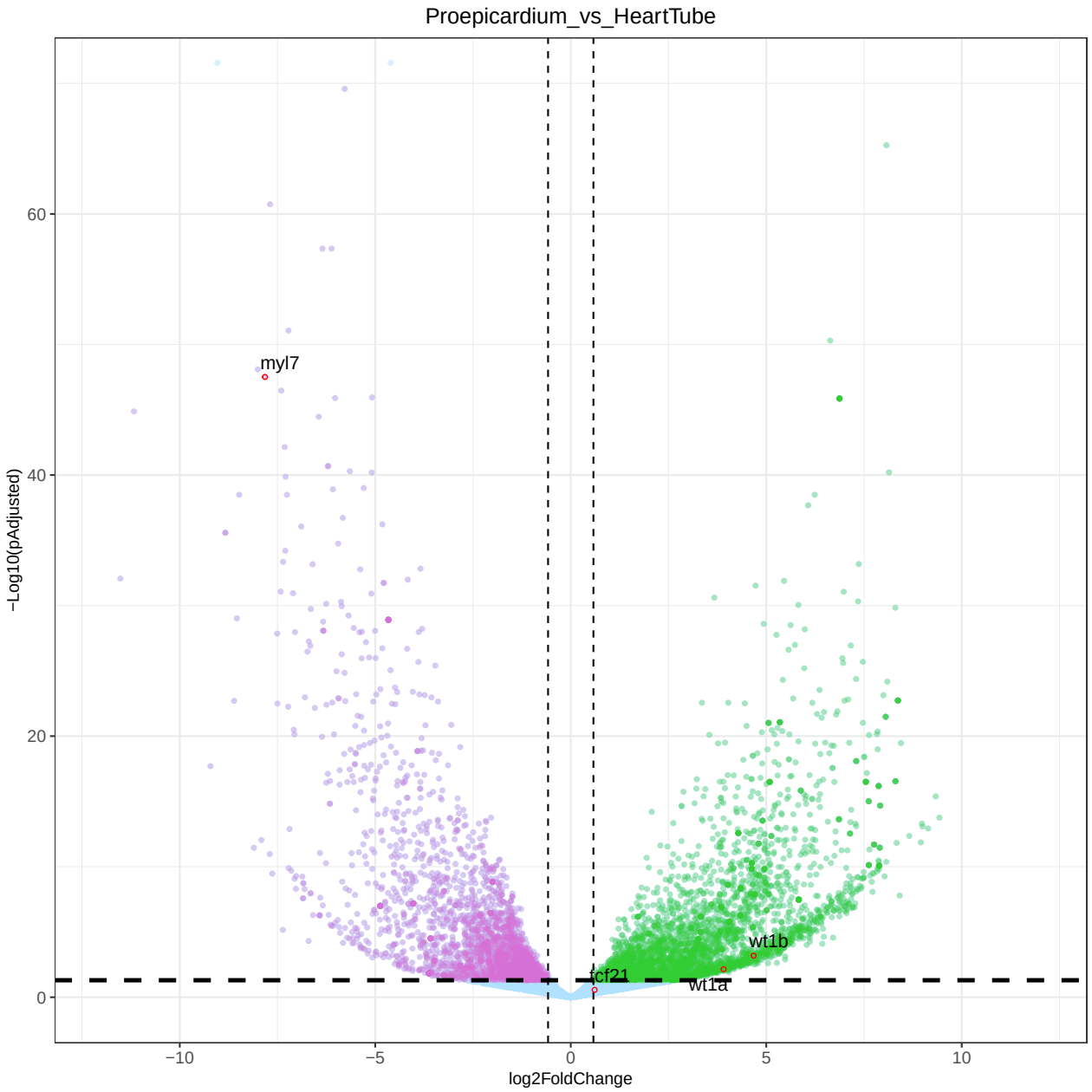
```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

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

```
c
```

```
## Warning: Removed 2132 rows containing missing values (geom_point).
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```


Proepicardium_vs_HeartTube
−Log10(pAdjusted)
log2FoldChange
myl7
wt1b
tcf21
wt1a
60
40
20
0
−10
−5
0
5
10

```
dev.copy(
svg,
file = paste0("Volcano_Proepicardium_vs_HeartTube_new_myl7_smalldots_outlier_removed.svg"),
width = 10,
height = 8
)
```

```
## svg
##   3
```

```
dev.off ()
```

```
## pdf
##   2
```

Proepicardium_vs_Pericardium

```
results_PE_PC <- lfcShrink(dds_filtered, contrast = c("Tissue","Proepicardium","Pericardium"),
                           alpha=0.05, parallel=TRUE, 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
```

```
print("lfcshrinakge done...")
```

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

```
results_PE_PC_df <- as.data.frame(results_PE_PC)
results_PE_PC_df <- results_PE_PC_df %>% dplyr::arrange(padj)
write.csv(results_PE_PC_df, "results_PE_vs_PC_outlier_removed.csv")
```

```
results_PE_PC_df_annotated <- results_PE_PC_df
results_PE_PC_df_annotated$ensmbl_id <- rownames(results_PE_PC_df_annotated)
head(results_PE_PC_df_annotated)
```

```
##                     baseMean log2FoldChange     lfcSE       pvalue         padj
## ENSDARG00000015866 1207.4219       3.132253 0.4626819 2.128551e-12 3.232843e-08
## ENSDARG00000022767 1085.5071       3.303083 0.5121528 1.492248e-11 1.133213e-07
## ENSDARG00000095863  335.4103       2.961818 0.4651978 3.837366e-11 1.942730e-07
## ENSDARG00000042780  233.2584       3.595179 0.5946590 1.304152e-10 4.951867e-07
## ENSDARG00000040298  489.7249       3.250768 0.5606739 9.239451e-10 2.806576e-06
## ENSDARG00000099200  467.4695       3.829021 0.7211024 5.982484e-09 1.298028e-05
##                              ensmbl_id
## ENSDARG00000015866 ENSDARG00000015866
## ENSDARG00000022767 ENSDARG00000022767
## ENSDARG00000095863 ENSDARG00000095863
## ENSDARG00000042780 ENSDARG00000042780
## ENSDARG00000040298 ENSDARG00000040298
## ENSDARG00000099200 ENSDARG00000099200
```

```
results_PE_PC_df_annotated <- merge(results_PE_PC_df_annotated, Mouse_Genes,
                                    by.x= "ensmbl_id", by.y= "EnsmblID_Zebrafish", all.x=T)
tail(results_PE_PC_df_annotated)
```

```
##                ensmbl_id    baseMean log2FoldChange     lfcSE    pvalue
## 25113 ENSDARG00000117813    9.467942   0.0003660026 0.1041216 0.9717054
## 25114 ENSDARG00000117818    3.819031  -0.0004519962 0.1707805 0.9779043
## 25115 ENSDARG00000117822   10.842692   0.0074782608 0.1217569 0.5093018
## 25116 ENSDARG00000117824    1.635296  -0.0017339435 0.1874146 0.9202117
## 25117 ENSDARG00000117826    3.920223   0.0005028112 0.1421965 0.9714847
## 25118            mCherry 1156.955367   0.0020868039 0.1654722 0.8958525
##            padj ZFIN.symbol MGI.symbol EnsmblID_Mouse Gene.description
## 25113        NA        <NA>       <NA>           <NA>             <NA>
## 25114        NA        <NA>       <NA>           <NA>             <NA>
## 25115 0.9788688        <NA>       <NA>           <NA>             <NA>
## 25116        NA        <NA>       <NA>           <NA>             <NA>
## 25117        NA        <NA>       <NA>           <NA>             <NA>
## 25118 0.9912566        <NA>       <NA>           <NA>             <NA>
```

```
write.csv(results_PE_PC_df_annotated, "results_PE_vs_PC_annotated_outlier_removed.csv")
```

Volcano plot Proepicardium_vs_Pericardium

```
results_PE_PC_df_annotated_1 <- results_PE_PC_df_annotated
results_PE_PC_df_annotated_1$padj[results_PE_PC_df_annotated_1$ZFIN.symbol=="wt1a"] <- 1
results_PE_PC_df_annotated_1$padj[results_PE_PC_df_annotated_1$ZFIN.symbol=="wt1b"] <- 1
```

```
results_PE_PC_df_annotated_volcano <- results_PE_PC_df_annotated

b = draw_volcano(results_PE_PC_df_annotated_volcano,"Proepicardium_vs_Pericardium", 0.58)
# print(b)
#Set genes for marking
wt_set<-c("wt1a","wt1b", "tcf21", "myl7")
wt_set
```

```
## [1] "wt1a"  "wt1b"  "tcf21" "myl7"
```

```
results_PE_PC_df_annotated_volcano$padj[
  results_PE_PC_df_annotated_volcano$ZFIN.symbol %in% wt_set &
    is.na(results_PE_PC_df_annotated_volcano$padj)] <- 1

wt_geneset <- as.data.frame(results_PE_PC_df_annotated_volcano[
  results_PE_PC_df_annotated_volcano$ZFIN.symbol %in%  wt_set,])

require(ggrepel)
#Paint the genes in the plot
c= b +
  geom_point(data=results_PE_PC_df_annotated_volcano[results_PE_PC_df_annotated_volcano$ZFIN.symbol%in%
                                                       wt_set,],
             color="red",size=2, shape= 21) +
  geom_text_repel(data = results_PE_PC_df_annotated_volcano[
    results_PE_PC_df_annotated_volcano$ZFIN.symbol %in% wt_set,],
aes(label=results_PE_PC_df_annotated_volcano[
  results_PE_PC_df_annotated_volcano$ZFIN.symbol %in% wt_set,]$ZFIN.symbol))+
```

```
  # ) +
   scale_x_continuous(limits = c(-12,12))+scale_y_continuous(limits = c(0,5))
```

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

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

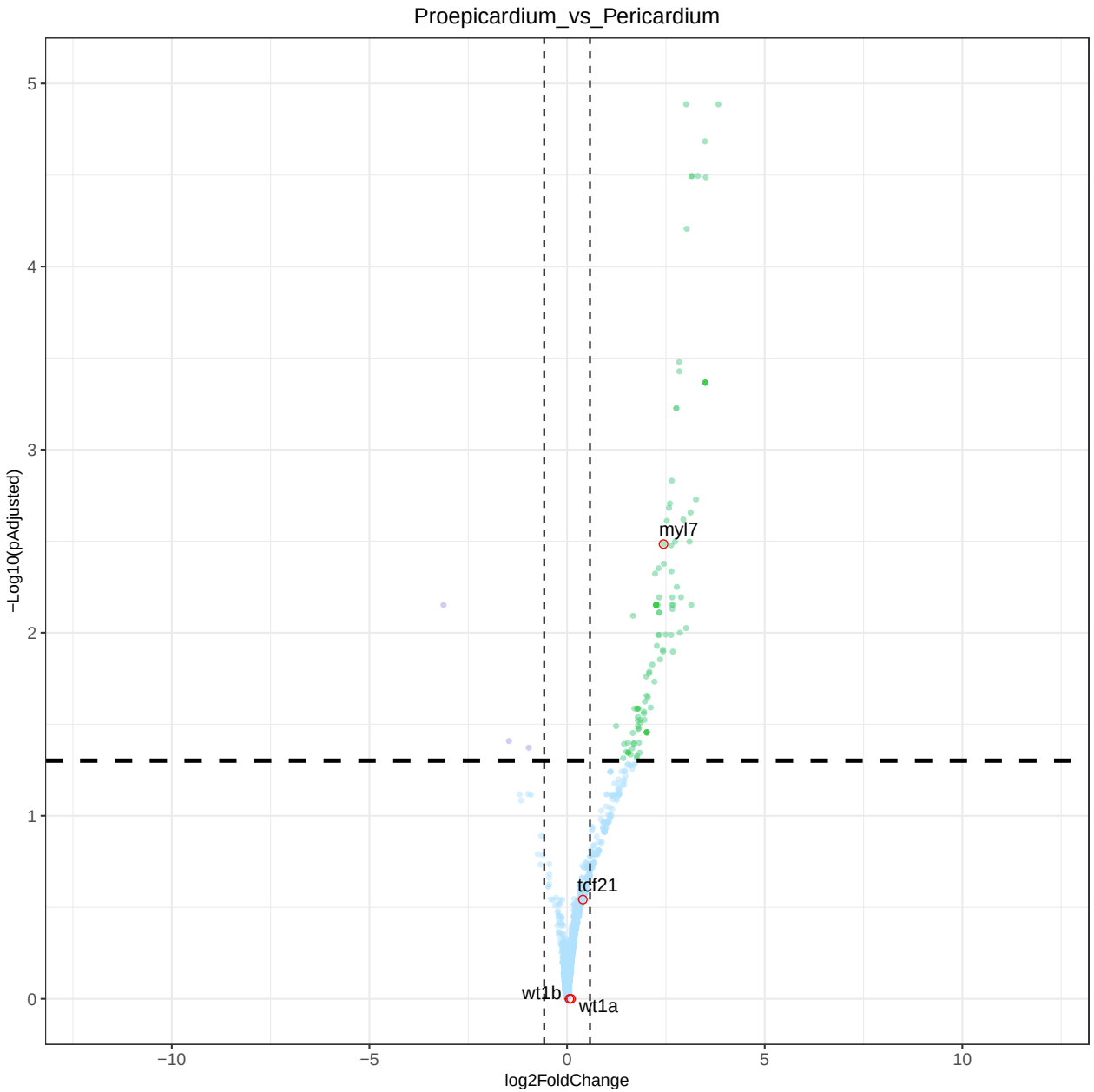
```
  #nudge_x = 0,
#nudge_y = 2,segment.size = 0.1)
  NULL
```

```
## NULL
```

```
c
```

```
## Warning: Removed 7591 rows containing missing values (geom_point).
```

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


Proepicardium_vs_Pericardium
−Log10(pAdjusted)
log2FoldChange
myl7
tcf21
wt1b
wt1a
5
4
3
2
1
0
−10
−5
0
5
10

```
dev.copy(
svg,
file = paste0("Volcano_Proepicardium_vs_Pericardium_new_myl7_smalldots_outlier_removed.svg"),
width = 10,
height = 8
)
```

```
## svg
##   3
```

```
dev.off ()
```

```
## pdf
##   2
```

```
results_PE_PC_df_annotated_volcano <- results_PE_PC_df_annotated

b = draw_volcano(results_PE_PC_df_annotated_volcano,"Proepicardium_vs_Pericardium", 0.58)
# print(b)
#Set genes for marking
wt_set<-c("wt1a","wt1b", "tcf21", "myl7")
wt_set
```

```
## [1] "wt1a"  "wt1b"  "tcf21" "myl7"
```

```
results_PE_PC_df_annotated_volcano$padj[
  results_PE_PC_df_annotated_volcano$ZFIN.symbol %in% wt_set &
    is.na(results_PE_PC_df_annotated_volcano$padj)] <- 1
wt_geneset<-as.data.frame(results_PE_PC_df_annotated_volcano[
  results_PE_PC_df_annotated_volcano$ZFIN.symbol %in% wt_set,])

require(ggrepel)
#Paint the genes in the plot
c= b + geom_point(data=results_PE_PC_df_annotated_volcano[
  results_PE_PC_df_annotated_volcano$ZFIN.symbol %in% wt_set,],
  color="red",size=2, shape= 21) +
```

```
geom_text_repel(data = results_PE_PC_df_annotated_volcano[
  results_PE_PC_df_annotated_volcano$ZFIN.symbol %in% wt_set,],
aes(label=results_PE_PC_df_annotated_volcano[
  results_PE_PC_df_annotated_volcano$ZFIN.symbol %in% wt_set,]$ZFIN.symbol))+
  # ) +
   scale_x_continuous(limits = c(-12,12))+scale_y_continuous(limits = c(0,70))
```

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

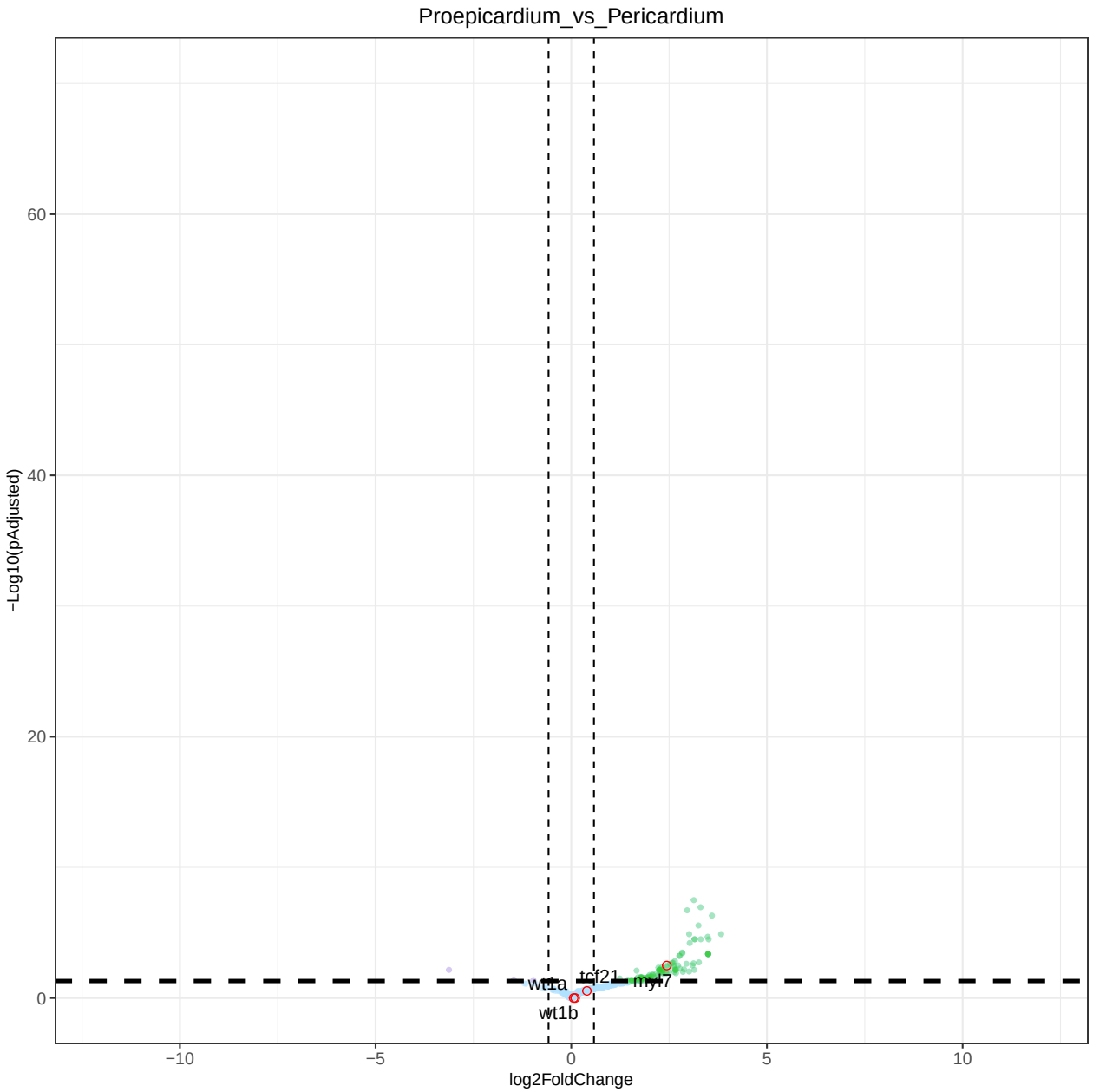
```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

```
  #nudge_x = 0,
#nudge_y = 2,segment.size = 0.1)
  NULL
```

```
## NULL
```

```
c
```

```
## Warning: Removed 7591 rows containing missing values (geom_point).
```


Proepicardium_vs_Pericardium
−Log10(pAdjusted)
60
40
20
0
−10
−5
0
5
10
log2FoldChange
wt1a
tcf21
myl7
wt1b

```
dev.copy(
svg,
file = paste0("Volcano_Proepicardium_vs_Pericardium_new_same_yscale_myl7_smalldots_outlier_removed.svg"),
width = 10,
height = 8
)
```

```
## svg
##   3
```

```
dev.off ()
```

```
## pdf
##   2
```

```
length(unique(rownames(counts_mat)))
```

```
## [1] 32536
```

Write gct files

Make dataframe with row annotation

```
rdesc_df <- data.frame(ensembl_gene_id = rownames(counts_mat))
head(rdesc_df)
```

```
##      ensembl_gene_id
## 1 ENSDARG00000102141
## 2 ENSDARG00000102123
## 3 ENSDARG00000114503
## 4 ENSDARG00000115971
## 5 ENSDARG00000098311
## 6 ENSDARG00000104839
```

```
dim(rdesc_df)
```

```
## [1] 32536     1
```

```
rdesc_df <- dplyr::left_join(x = rdesc_df,y= gene_symbols_ensmbl_df, by= "ensembl_gene_id")
head(rdesc_df)
```

```
##      ensembl_gene_id  zfin_id_symbol
## 1 ENSDARG00000102141          ptpn12
## 2 ENSDARG00000102123           phtf2
## 3 ENSDARG00000114503           phtf2
## 4 ENSDARG00000115971
## 5 ENSDARG00000098311 si:zfos-932h1.3
## 6 ENSDARG00000104839          mansc1
```

```
rdesc_df$zfin_id_symbol[rdesc_df$zfin_id_symbol==""] <- NA
rdesc_df[duplicated(rdesc_df$ensembl_gene_id),]
```

```
##          ensembl_gene_id     zfin_id_symbol
## 884   ENSDARG00000100479         zgc:173720
## 1036  ENSDARG00000116992            znf1051
## 1413  ENSDARG00000090160         zgc:173709
## 1897  ENSDARG00000117113         zgc:173607
## 2206  ENSDARG00000091869         zgc:171727
## 2366  ENSDARG00000068869          hspa12a.1
## 2486  ENSDARG00000089958   si:dkey-88n24.10
## 2585  ENSDARG00000105352         zgc:165555
## 2595  ENSDARG00000101249         zgc:165555
## 2600  ENSDARG00000099920         zgc:165555
## 2601  ENSDARG00000099920         zgc:163040
## 2611  ENSDARG00000105353         zgc:165555
## 2616  ENSDARG00000105382         zgc:165555
## 2620  ENSDARG00000105318         zgc:165555
## 2632  ENSDARG00000105376         zgc:165555
## 2639  ENSDARG00000105482         zgc:165555
## 2646  ENSDARG00000105355         zgc:165555
## 2647  ENSDARG00000105355         zgc:154164
```

```
## 2650  ENSDARG00000105409         zgc:165555
## 2858  ENSDARG00000077998             nitr3a
## 2860  ENSDARG00000057620             nitr3b
## 3772  ENSDARG00000068290            cyp2x12
## 3849  ENSDARG00000051890                 hp
## 4123  ENSDARG00000111501         zgc:165555
## 4134  ENSDARG00000111619         zgc:165555
## 4336  ENSDARG00000116986             ugt2b3
## 4337  ENSDARG00000116986         zgc:172315
## 4756  ENSDARG00000061547         zgc:165555
## 4757  ENSDARG00000061547         zgc:153409
## 4815  ENSDARG00000068122  si:ch211-215c18.3
## 5023  ENSDARG00000033009              h3f3c
## 5091  ENSDARG00000092222         wu:fb64b08
## 5178  ENSDARG00000093779           cxcl11.7
## 5268  ENSDARG00000060102             kank1a
## 5771  ENSDARG00000039501             ugt2a5
## 5776  ENSDARG00000051940             ugt2a1
## 6034  ENSDARG00000100803         zgc:173517
## 6036  ENSDARG00000102401         zgc:173517
## 6712  ENSDARG00000113977             fthl29
## 7098  ENSDARG00000116052             il4r.1
## 7117  ENSDARG00000020504              h3f3a
## 7279  ENSDARG00000089475            hbae1.3
## 7281  ENSDARG00000115405            hbbe1.3
## 7283  ENSDARG00000088330            hbae1.3
## 7449  ENSDARG00000116815         zgc:173575
## 8074  ENSDARG00000103066                smx5
## 8963  ENSDARG00000095549   si:dkeyp-106c3.2
## 9541  ENSDARG00000005941              clul1
## 10250 ENSDARG00000070661    si:ch73-42k18.1
## 10596 ENSDARG00000099276             ugt5b2
## 10598 ENSDARG00000104995             ugt5b1
## 10886 ENSDARG00000054799               rfc1
## 11051 ENSDARG00000069951            eef1a1b
## 11732 ENSDARG00000094346  si:ch211-114l13.3
## 11776 ENSDARG00000105176         zgc:174890
## 12044 ENSDARG00000095741              ddx41
## 12077 ENSDARG00000006220             ugt1a7
```

```
## 12078 ENSDARG00000006220            ugt1a6
## 12079 ENSDARG00000006220            ugt1a2
## 12080 ENSDARG00000006220            ugt1a1
## 12083 ENSDARG00000010563            spopla
## 12141 ENSDARG00000006301            raph1a
## 12250 ENSDARG00000069922             pla1a
## 12444 ENSDARG00000056917  si:rp71-45g20.10
## 12766 ENSDARG00000000540            asnsd1
## 13081 ENSDARG00000088571            soga3a
## 13969 ENSDARG00000076539           sc:d156
## 14049 ENSDARG00000104344          ccdc127a
## 14443 ENSDARG00000093477         cyp46a1.1
## 14531 ENSDARG00000077840            meis2b
## 14542 ENSDARG00000070757 si:ch211-182e10.4
## 14570 ENSDARG00000043475            tagapb
## 14635 ENSDARG00000034187            calm3a
## 15457 ENSDARG00000029150          hsp90ab1
## 16447 ENSDARG00000070206 si:busm1-160c18.6
## 16652 ENSDARG00000053135          tas1r2.1
## 16916 ENSDARG00000090768        zgc:173556
## 16918 ENSDARG00000114958               zp3
## 17050 ENSDARG00000058673             nkl.1
## 17457 ENSDARG00000021811            calm3a
## 18050 ENSDARG00000098788          pcdh2g16
## 18051 ENSDARG00000098788          pcdh2g13
## 18052 ENSDARG00000098788           pcdh2g5
## 18053 ENSDARG00000098788          pcdh2g29
## 18054 ENSDARG00000098788          pcdh2g28
## 18060 ENSDARG00000101318         pcdh2ab11
## 18071 ENSDARG00000089874          pcdh2ab8
## 19179 ENSDARG00000092199  si:ch211-250c4.4
## 19196 ENSDARG00000031427            calm3a
## 19246 ENSDARG00000111633        zgc:174193
## 20302 ENSDARG00000042396             odf3b
## 20394 ENSDARG00000068763             plcg2
## 21094 ENSDARG00000039500    zmp:0000001073
## 21480 ENSDARG00000089750    si:dkey-26g8.5
## 21668 ENSDARG00000015050            calm3a
## 21871 ENSDARG00000087657              fasn
```

```
## 22393 ENSDARG00000004866             mindy1
## 22697 ENSDARG00000103862             hoxa3a
## 23189 ENSDARG00000020850           eef1a1l1
## 24304 ENSDARG00000097307  si:dkey-238d18.10
## 24425 ENSDARG00000102483              numa1
## 24513 ENSDARG00000068436 si:ch1073-429i10.3
## 24616 ENSDARG00000058871         zgc:171857
## 24825 ENSDARG00000071029    si:dkey-147f3.4
## 25104 ENSDARG00000101869            magi3a
## 25300 ENSDARG00000094469   si:ch211-148l7.4
## 25695 ENSDARG00000102950   si:ch211-241b2.1
## 25714 ENSDARG00000105690         zgc:194215
## 25953 ENSDARG00000062986            pnpla7a
## 26326 ENSDARG00000074057             calm3a
## 26605 ENSDARG00000053817            or115-2
## 26758 ENSDARG00000110357            txndc15
## 27817 ENSDARG00000060734              appl1
## 27885 ENSDARG00000103158              ldb1a
## 28380 ENSDARG00000104598            pcdh1a6
## 28381 ENSDARG00000104598            pcdh1a3
## 28718 ENSDARG00000055113         wu:fb30f12
## 29636 ENSDARG00000061481         wu:fj05g07
## 29715 ENSDARG00000088276   si:ch211-190p8.2
## 29793 ENSDARG00000045248              h3f3d
## 29889 ENSDARG00000114031             smyhc2
## 30070 ENSDARG00000098613         zgc:174222
## 30133 ENSDARG00000071714    si:dkey-20i20.8
## 30540 ENSDARG00000023656              he1.1
## 30857 ENSDARG00000093249   si:ch211-226h8.4
## 30866 ENSDARG00000086247   si:ch73-189n23.1
## 30869 ENSDARG00000093841         zgc:152936
## 30878 ENSDARG00000090929    si:dkeyp-98a7.3
## 30881 ENSDARG00000089525    si:dkeyp-98a7.5
## 31597 ENSDARG00000051959            mmp15a
## 31998 ENSDARG00000111753         zgc:165555
## 32010 ENSDARG00000112954         zgc:165555
## 32018 ENSDARG00000115281         zgc:165555
## 32023 ENSDARG00000114534         zgc:165555
## 32028 ENSDARG00000110812         zgc:165555
```

```
## 32035 ENSDARG00000111247         zgc:165555
## 32040 ENSDARG00000110035         zgc:165555
## 32044 ENSDARG00000111165         zgc:165555
## 32046 ENSDARG00000115644         zgc:173552
## 32087 ENSDARG00000116781         zgc:165555
## 32091 ENSDARG00000111494         zgc:165555
## 32094 ENSDARG00000112528         zgc:165555
## 32096 ENSDARG00000077456         zgc:110216
## 32100 ENSDARG00000110152         zgc:165555
## 32104 ENSDARG00000115054         zgc:165555
## 32109 ENSDARG00000112491         zgc:165555
## 32115 ENSDARG00000115799         zgc:165555
```

```
rdesc_df <- rdesc_df[!duplicated(rdesc_df$ensembl_gene_id),]
rownames(rdesc_df) <- (rownames(counts_mat))
head(rdesc_df)
```

```
##                       ensembl_gene_id  zfin_id_symbol
## ENSDARG00000102141 ENSDARG00000102141          ptpn12
## ENSDARG00000102123 ENSDARG00000102123           phtf2
## ENSDARG00000114503 ENSDARG00000114503           phtf2
## ENSDARG00000115971 ENSDARG00000115971            <NA>
## ENSDARG00000098311 ENSDARG00000098311 si:zfos-932h1.3
## ENSDARG00000104839 ENSDARG00000104839          mansc1
```

```
length(rownames(counts_mat))
```

```
## [1] 32536
```

```
dim(gene_symbols_ensmbl_df)
```

```
## [1] 32667     2
```

```
dim(rdesc_df)
```

```
## [1] 32536     2
```

gct with raw counts

```
gct_obj <- new("GCT", mat=counts_mat)
gct_obj@cdesc <- metadata_smart
gct_obj@rdesc <- rdesc_df
write_gct(gct_obj, "smart_laura_counts_outlier_removed")
```

```
## Saving file to  ./smart_laura_counts_outlier_removed_n9x32536.gct
## Dimensions of matrix: [32536x9]
## Setting precision to 4
## Saved.
```

```
gct_obj
```

```
## Formal class 'GCT' [package "cmapR"] with 7 slots
##   ..@ mat    : int [1:32536, 1:9] 2 0 0 0 1 3 0 58 59 28 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:32536] "ENSDARG00000102141" "ENSDARG00000102123" "ENSDARG00000114503" "ENSDARG00000115971" ...
##   .. .. ..$ : chr [1:9] "S1_1" "S1_2" "S1_4" "S1_5" ...
##   ..@ rid    : chr [1:32536] "ENSDARG00000102141" "ENSDARG00000102123" "ENSDARG00000114503" "ENSDARG00000115971" ...
##   ..@ cid    : chr [1:9] "S1_1" "S1_2" "S1_4" "S1_5" ...
##   ..@ rdesc  :'data.frame':  32536 obs. of  2 variables:
##   .. ..$ ensembl_gene_id: chr [1:32536] "ENSDARG00000102141" "ENSDARG00000102123" "ENSDARG00000114503" "ENSDARG00000115971" ...
##   .. ..$ zfin_id_symbol : chr [1:32536] "ptpn12" "phtf2" "phtf2" NA ...
##   ..@ cdesc  :'data.frame':  9 obs. of  2 variables:
##   .. ..$ Sample_Name: chr [1:9] "S1_1" "S1_2" "S1_4" "S1_5" ...
##   .. ..$ Tissue     : chr [1:9] "Proepicardium" "Proepicardium" "Proepicardium" "Proepicardium" ...
##   ..@ version: chr(0)
##   ..@ src    : chr(0)
```

```
write.csv(counts_mat, "raw_counts.csv")
```

GCT with normalized data

```
rdesc_df_norm <- rdesc_df[rownames(rdesc_df)%in% rownames(assay(dds_filtered_rlog)),]
dim(rdesc_df_norm)
```

```
## [1] 21486     2
```

```
gct_obj_norm <- new("GCT", mat=assay(dds_filtered_rlog))
gct_obj_norm@cdesc <- metadata_smart
gct_obj_norm@rdesc <- rdesc_df_norm
write_gct(gct_obj_norm, "smart_laura_counts_outlier_removed")
```

```
## Saving file to  ./smart_laura_counts_outlier_removed_n9x21486.gct
## Dimensions of matrix: [21486x9]
## Setting precision to 4
## Saved.
```

```
gct_obj_norm
```

```
## Formal class 'GCT' [package "cmapR"] with 7 slots
##   ..@ mat    : num [1:21486, 1:9] 1.4272 -0.8331 0.0728 0.4087 0.4423 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:21486] "ENSDARG00000102141" "ENSDARG00000102123" "ENSDARG00000114503" "ENSDARG00000098311" ...
##   .. .. ..$ : chr [1:9] "S1_1" "S1_2" "S1_4" "S1_5" ...
##   .. ..- attr(*, "betaPriorVar")= num 3.45
##   .. ..- attr(*, "intercept")= num [1:21486, 1] 1.911 -0.413 0.661 0.645 0.133 ...
##   ..@ rid    : chr [1:21486] "ENSDARG00000102141" "ENSDARG00000102123" "ENSDARG00000114503" "ENSDARG00000098311" ...
##   ..@ cid    : chr [1:9] "S1_1" "S1_2" "S1_4" "S1_5" ...
##   ..@ rdesc  :'data.frame':  21486 obs. of  2 variables:
##   .. ..$ ensembl_gene_id: chr [1:21486] "ENSDARG00000102141" "ENSDARG00000102123" "ENSDARG00000114503" "ENSDARG00000098311" ...
##   .. ..$ zfin_id_symbol : chr [1:21486] "ptpn12" "phtf2" "phtf2" "si:zfos-932h1.3" ...
##   ..@ cdesc  :'data.frame':  9 obs. of  2 variables:
##   .. ..$ Sample_Name: chr [1:9] "S1_1" "S1_2" "S1_4" "S1_5" ...
##   .. ..$ Tissue     : chr [1:9] "Proepicardium" "Proepicardium" "Proepicardium" "Proepicardium" ...
##   ..@ version: chr(0)
##   ..@ src    : chr(0)
```

```
write.csv(assay(dds_filtered_rlog), "normalized_counts.csv")
```

Save RData

```
save.image("SMARTseq_analysis.Rdata")
```

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.2 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## 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
##
## attached base packages:
##  [1] grid      parallel  stats4    stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.5.1               stringr_1.4.0
##  [3] purrr_0.3.4                 readr_1.4.0
##  [5] tidyr_1.1.3                 tibble_3.1.2
##  [7] tidyverse_1.3.1             ggbiplot_0.55
##  [9] scales_1.1.1                plyr_1.8.6
## [11] ggrepel_0.9.1               curl_4.3.2
```

```
## [13] biomaRt_2.48.0              cmapR_1.4.0
## [15] openxlsx_4.2.4              enrichplot_1.12.2
## [17] clusterProfiler_4.0.0       DESeq2_1.32.0
## [19] SummarizedExperiment_1.22.0 Biobase_2.52.0
## [21] MatrixGenerics_1.4.0        matrixStats_0.59.0
## [23] GenomicRanges_1.44.0        GenomeInfoDb_1.28.0
## [25] IRanges_2.26.0              S4Vectors_0.30.0
## [27] BiocGenerics_0.38.0         ggplot2_3.3.5
## [29] dplyr_1.0.6
##
## loaded via a namespace (and not attached):
##   [1] readxl_1.3.1           backports_1.2.1        shadowtext_0.0.8
##   [4] fastmatch_1.1-0        BiocFileCache_2.0.0    igraph_1.2.6
##   [7] lazyeval_0.2.2         splines_4.1.0          flowCore_2.4.0
##  [10] BiocParallel_1.26.0    digest_0.6.27          invgamma_1.1
##  [13] htmltools_0.5.1.1      GOSemSim_2.18.0        viridis_0.6.1
##  [16] GO.db_3.13.0           SQUAREM_2021.1         fansi_0.5.0
##  [19] magrittr_2.0.1         memoise_2.0.0          Biostrings_2.60.0
##  [22] annotate_1.70.0        graphlayouts_0.7.1     modelr_0.1.8
##  [25] RcppParallel_5.1.4     cytolib_2.4.0          prettyunits_1.1.1
##  [28] colorspace_2.0-2       rvest_1.0.0            blob_1.2.1
##  [31] rappdirs_0.3.3         haven_2.4.1            xfun_0.24
##  [34] crayon_1.4.1           RCurl_1.98-1.3         jsonlite_1.7.2
##  [37] scatterpie_0.1.6       genefilter_1.74.0      survival_3.2-11
##  [40] ape_5.5                glue_1.4.2             polyclip_1.10-0
##  [43] gtable_0.3.0           zlibbioc_1.38.0        XVector_0.32.0
##  [46] DelayedArray_0.18.0    DOSE_3.18.1            DBI_1.1.1
##  [49] Rcpp_1.0.6             viridisLite_0.4.0      xtable_1.8-4
##  [52] progress_1.2.2         tidytree_0.3.4         bit_4.0.4
##  [55] truncnorm_1.0-8        httr_1.4.2             fgsea_1.18.0
##  [58] RColorBrewer_1.1-2     ellipsis_0.3.2         pkgconfig_2.0.3
##  [61] XML_3.99-0.6           farver_2.1.0           dbplyr_2.1.1
##  [64] locfit_1.5-9.4         utf8_1.2.1             labeling_0.4.2
##  [67] tidyselect_1.1.1       rlang_0.4.11           reshape2_1.4.4
##  [70] AnnotationDbi_1.54.0   cellranger_1.1.0       munsell_0.5.0
##  [73] tools_4.1.0            cachem_1.0.5           cli_2.5.0
##  [76] downloader_0.4         generics_0.1.0         RSQLite_2.2.7
##  [79] broom_0.7.8            evaluate_0.14          fastmap_1.1.0
##  [82] yaml_2.2.1             ggtree_3.0.2           fs_1.5.0
```

```
##  [85] knitr_1.33             bit64_4.0.5            tidygraph_1.2.0
##  [88] zip_2.2.0              KEGGREST_1.32.0        ggraph_2.0.5
##  [91] nlme_3.1-152           aplot_0.0.6            xml2_1.3.2
##  [94] DO.db_2.9              rstudioapi_0.13        compiler_4.1.0
##  [97] filelock_1.0.2         png_0.1-7              reprex_2.0.0
## [100] treeio_1.16.1          tweenr_1.0.2           geneplotter_1.70.0
## [103] stringi_1.6.2          highr_0.9              lattice_0.20-44
## [106] Matrix_1.3-4           vctrs_0.3.8            pillar_1.6.1
## [109] lifecycle_1.0.0        BiocManager_1.30.15    irlba_2.3.3
## [112] data.table_1.14.0      cowplot_1.1.1          bitops_1.0-7
## [115] patchwork_1.1.1        qvalue_2.24.0          R6_2.5.0
## [118] gridExtra_2.3          RProtoBufLib_2.4.0     MASS_7.3-54
## [121] assertthat_0.2.1       withr_2.4.2            GenomeInfoDbData_1.2.6
## [124] hms_1.1.0              rmarkdown_2.9          rvcheck_0.1.8
## [127] ashr_2.2-47            mixsqp_0.3-43          ggforce_0.3.3
## [130] lubridate_1.7.10
```