DEG_analysis - Bulk RNA-seq

Hassan Saei

2024-12-09

Visualization of DEGs obtained form comparing kidney organoid cultured for prolonged period (early, mid and late time points)

We developed kidney organoids from male hiPSCs using Morizane et al. protocol () with some modifications. Most of the analysis on kidney organoids were performed short after the last day of differentiation (day 21). Less is known about the gene expression dynamics over prolonged culture. We harvested 3D kidney organoids at day 21 (early), mid (day 32), and late (day 42) time poinst and extracted total RNA and performed bulk RNA sequencing using Illumnia Novaseq 6000.

Install packages

Loading libraries

```
#rm(list=ls())
library(limma)
library(edgeR)
library(DESeq2)
library(ggplot2)
library(EnsDb.Hsapiens.v86)
library(dplyr)
library(radiant)
library(EnhancedVolcano)
library(tidyverse)
library(clusterProfiler)
library(RColorBrewer)
library(ggrepel)
library(ggplot2)
library("readxl")
library(data.table)
library(UpSetR)
library(ggrepel)
library(pheatmap)
library(colorRamp2)
```

Converting ENSG IDs to gene symbol

```
#str(EnsDb.Hsapiens.v86)
#columns(EnsDb.Hsapiens.v86)
#keys(EnsDb.Hsapiens.v86)
ens2sym <- AnnotationDbi::select(EnsDb.Hsapiens.v86, keys = keys(EnsDb.Hsapiens.v86),
                                 columns = c("SYMBOL"))
ens2sym_entrez <- AnnotationDbi::select(EnsDb.Hsapiens.v86, keys = keys(EnsDb.Hsapiens.v86), columns =
```

Reading differentially expressed genes from .txt files

- DEGs obtained from comparing day 32 versus day 22 organoids
- DEGs obtained from comparing day 42 versus day 22 organoids
- NABA_MATRISOME file contains genes encoding matrix and matrix associated proteins

```
DEG_mid <- read.table("Listes_Res_MultiTests/(d32)_vs_(d22)_f1.2_(7369).txt", sep = "\t", header = T)
DEG_late <- read.table("Listes_Res_MultiTests/(d42)_vs_(d22)_f1.2_(11925).txt", sep = "\t", header = T)
NABA <- read.table("Listes_Res_MultiTests/NABA_MATRISOME.v2023.2.txt", sep = "\t", header = T)
```

Anntating genes and keeping shared genes between three statistical methods (DESeq2, edgeR and limma-voom)

```
process_dataframe <- function(df, naba_df) {
  df <- df %>%
    mutate(Category = ifelse(Symbol %in% naba_df$Symbol, "Matrix", "No"))

  df <- with(df, df[!(pval_edgeR == "" | is.na(pval_edgeR)), ])
  df <- with(df, df[!(pval_Voom == "" | is.na(pval_Voom)), ])
  df <- with(df, df[!(pval_DEseq2 == "" | is.na(pval_DEseq2)), ])

  return(df)
}

DEG_mid <- process_dataframe(DEG_mid, NABA)
DEG_late <- process_dataframe(DEG_late, NABA)
```

Venndiagram for shared matrisom endcoding genes

```
library("ggVennDiagram")
```

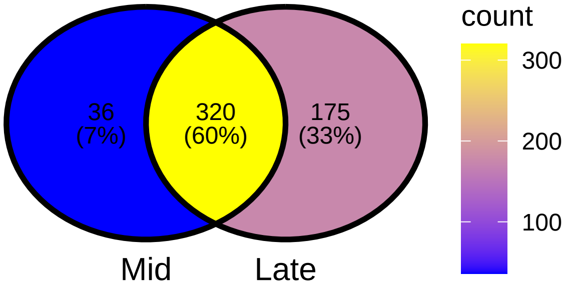
```
##
## Attaching package: 'ggVennDiagram'
```

```
## The following object is masked from 'package:tidyr':
##
##     unite
```

```
matrix_mid <- DEG_mid$Category == "Matrix"
matrix_late <- DEG_late$Category == "Matrix"

genes_mid <- DEG_mid %>%
  filter(matrix_mid) %>%
  pull(Symbol)

genes_late <- DEG_late %>%
  filter(matrix_late) %>%
  pull(Symbol)
```

```
list <- list(genes_mid, genes_late)
ggVennDiagram(list, label_alpha = 0, label_size = 3,
              category.names = c("Mid", "Late")) + ggplot2::scale_fill_gradient(low="blue",high = "yell
```

```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```



Save the plot in the .png format

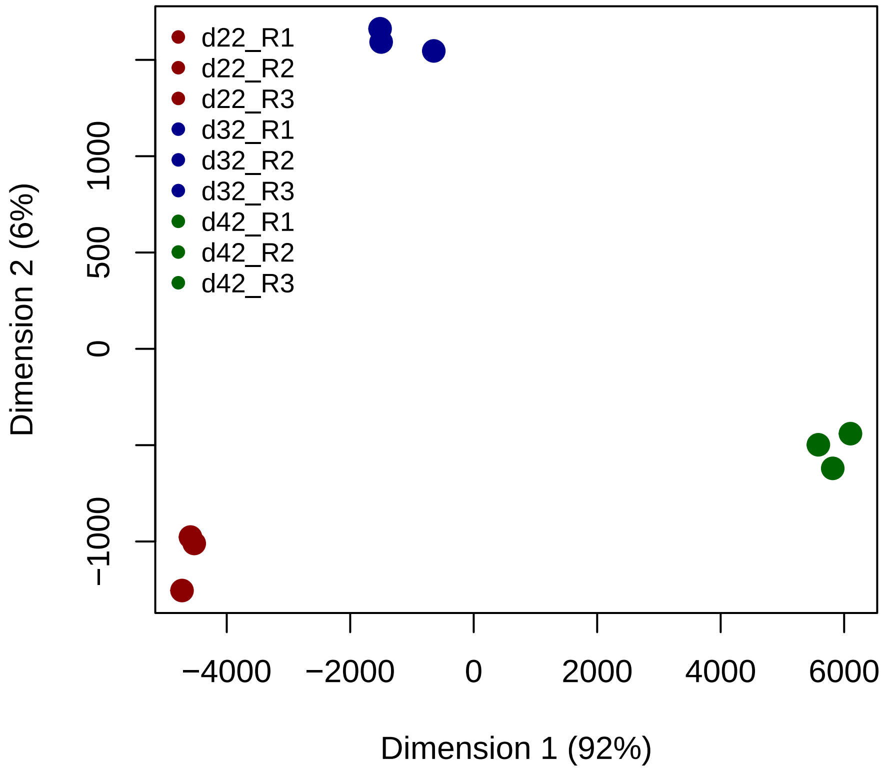
```
#png(filename = "Venn.png", width = 150, height = 100, res = 300, units = "mm")
#plot(p)
#dev.off()
```

Extract common DEGs between mid and early comparison

```
common_genes <- intersect(genes_mid, genes_late)

DEG_mid <- DEG_mid %>%
  mutate(Common = ifelse(Symbol %in% common_genes, "Common", "Unique"))

DEG_late <- DEG_late %>%
  mutate(Common = ifelse(Symbol %in% common_genes, "Common", "Unique"))

# save common genes

write.table(common_genes, file = "Common_genes_early_vs_late.txt", row.names = F, col.names = 'Common_DI
```

Prepare file to generate MDS plot

```
sampleTable <- read.table("Listes_Res_MultiTests/DataNormDESeq2.txt", header = T)
sampleTable <- sampleTable[,c(1:9)]

group <- factor(c(rep("d22", 3), rep("d32", 3), rep("d42", 3)))
colors <- c("darkred", "darkblue", "darkgreen")[group]

mds <- plotMDS(sampleTable, plot = FALSE)
plot(mds$x, mds$y, col = colors, pch = 19, cex = 1.5, xlab = "Dimension 1 (92%)", ylab = "Dimension 2 (
legend("topleft", legend = colnames(sampleTable), col = colors, pch = 19, cex = 0.8, bty = "n")
```



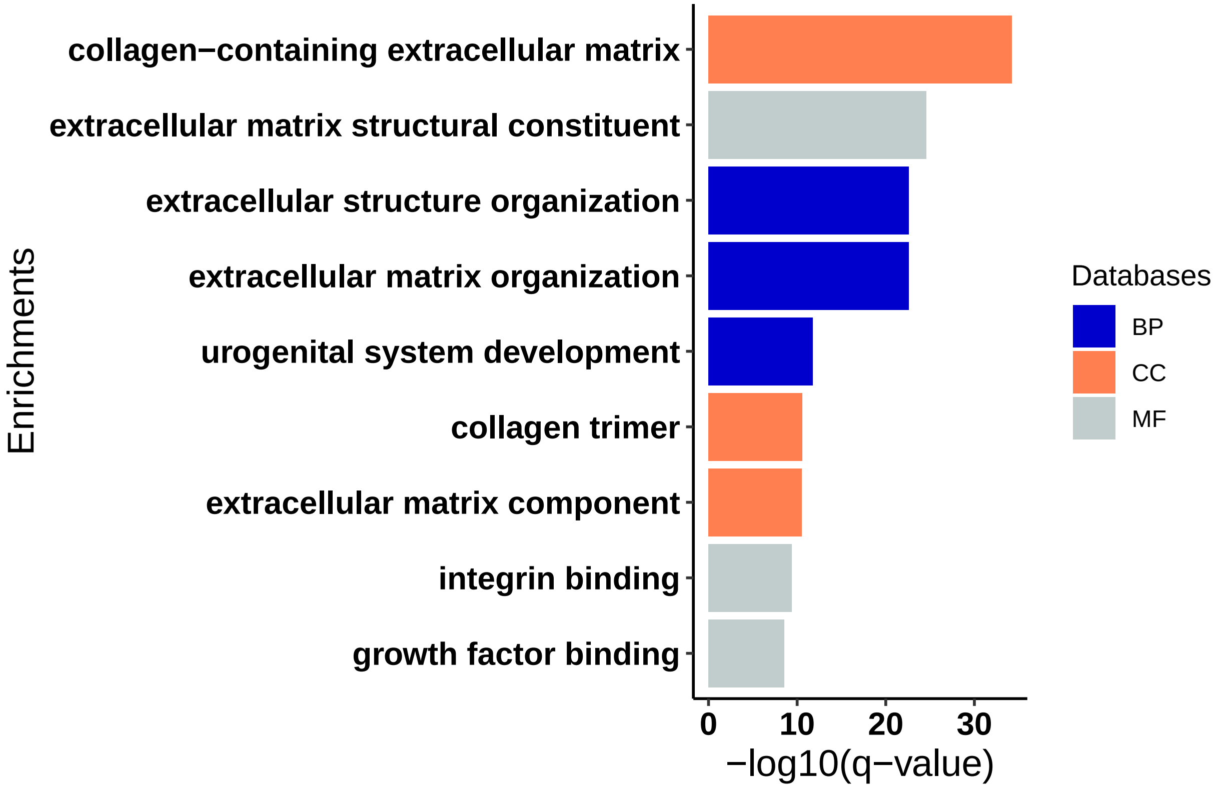
Save MDS plot

```
#png(filename = "PlotMDS.png", width = 140, height = 140, res = 300, units = "mm")
#plot(mds$x, mds$y, col = colors, pch = 19, cex = 1.5, xlab = "Dimension 1 (92%)", ylab = "Dimension 2
#dev.off()
```

Visualization of the enrichment results

```
# Enrichment
enrich <- fread("Listes_Res_MultiTests/GO_All_Venn_(d32)_vs_(d22)_f1.5_(3360).txt",
                     header = TRUE, sep = "\t")

enrich <- enrich[enrich$qvalue <= 0.05,]
df_BP <- subset(enrich, ONTOLOGY == "BP")
df_CC <- subset(enrich, ONTOLOGY == "CC")
df_MF <- subset(enrich, ONTOLOGY == "MF")

# Select the first three columns
df_BP <- df_BP[1:3,]
df_CC <- df_CC[1:3,]
df_MF <- df_MF[1:3,]

# Combine the dataframes back if needed
df_combined <- rbind(df_BP, df_CC, df_MF)
df_combined$log10_Adjusted <- -log10(df_combined$qvalue)
```

```
ggplot(df_combined, aes(x = log10_Adjusted, y = reorder(Description, log10_Adjusted), fill = ONTOLOGY))
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("BP" = "blue3", "CC" = "coral", "MF" = "azure3")) +
  labs(
    title = "",
    x = "-log10(q-value)",
    y = "Enrichments",
    fill = "Databases"
  ) +
  theme_classic() +
  theme(
    axis.text.y = element_text(size = 12, face = "bold", colour = "black"),
    axis.text.x = element_text(size = 12, face = "bold", colour = "black"),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    plot.title = element_text(size = 16, face = "bold"),
    )
```



#

Save Enrichemnt plot

```
#png(filename = "Enrichment_d32vsd22.png", width = 200, height = 150, res = 300, units = "mm")
#plot(p)
#dev.off()
```

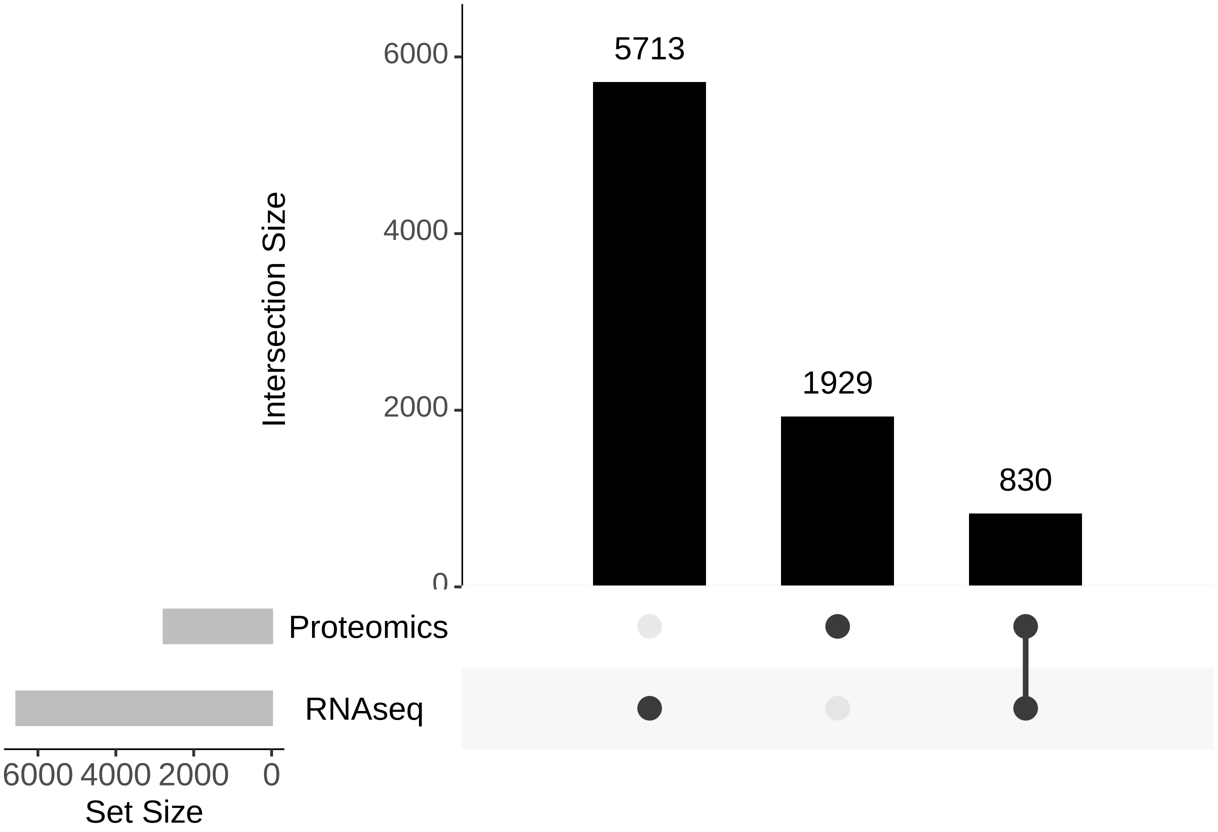
Visiulizing shared genes and proteins

```
DEG <- read.table("Listes_Res_MultiTests/(d42)_vs_(d32)_f1.2_(8892).txt", header = T, sep = "\t")
DEG <- DEG[,c(1:3,5)]
```

```r
DEG <- DEG[,-1]
DEG$log2FC <- log2(DEG$RatiosMoys_.d42._vs_.d32.)
DEG$pval_DEseq2 <- as.numeric(gsub(",", ".", DEG$pval_DEseq2))
DEG$pval_DEseq2 <- -log(DEG$pval_DEseq2)
DEG <- DEG[!is.na(DEG$pval_DEseq2), ]
DEG <- DEG[is.finite(DEG$pval_DEseq2), ]
DEG <- DEG[,-1]
colnames(DEG) <- c("SYMBOL", "Pvalue", "LogFC")

#DEG
```

Upset plot for comapring results from RNA-seq and proteomics

```r
process_gene_lists <- function(rna_file, proteomics_file) {
  # Read RNAseq data
  DEG <- rna_file

  # Read proteomics data
  pro <- fread(proteomics_file)

  # Filter significant proteomics results
  pro <- pro[pro$Significant == "+", ]

  # Select required columns
  pro <- pro[, c(2, 3, 7)]
  colnames(pro) <- c("Pvalue", "LogFC", "SYMBOL")

  # Reorder columns
  pro <- pro[, c("SYMBOL", "Pvalue", "LogFC")]

  # Convert Symbol columns to lists
  DEG_genes <- DEG$SYMBOL
  pro_genes <- pro$SYMBOL

  # Check for duplicates
  duplicated_DEG <- sum(duplicated(DEG_genes))
  duplicated_pro <- sum(duplicated(pro_genes))

  # Create a list for UpSet plot
  gene_list <- list(
    RNAseq = DEG_genes,
    Proteomics = pro_genes
  )

  # Find common genes
  common_genes <- intersect(gene_list$RNAseq, gene_list$Proteomics)

  # Return results
  return(list(
    common_genes = common_genes,
    length_common_genes = length(common_genes),
    duplicated_DEG = duplicated_DEG,
    duplicated_pro = duplicated_pro,
    gene_list = gene_list,
    pro = pro
  ))
}

result <- process_gene_lists(DEG, "Listes_Res_MultiTests/Proteomics.perseus.d38_vs_d22_DEPs.txt")
result2 <- process_gene_lists(DEG, "Listes_Res_MultiTests/Proteomics.perseus.d34_vs_d22_DEPs.txt")

# Create the UpSet plot
upset(fromList(result[['gene_list']]), order.by = "freq",
      main.bar.color = "black", sets.bar.color = "gray", point.size = 4, line.size = 1,
      text.scale = c(1.5, 1.5, 1.5), keep.order = F, set_size.show = F)
```



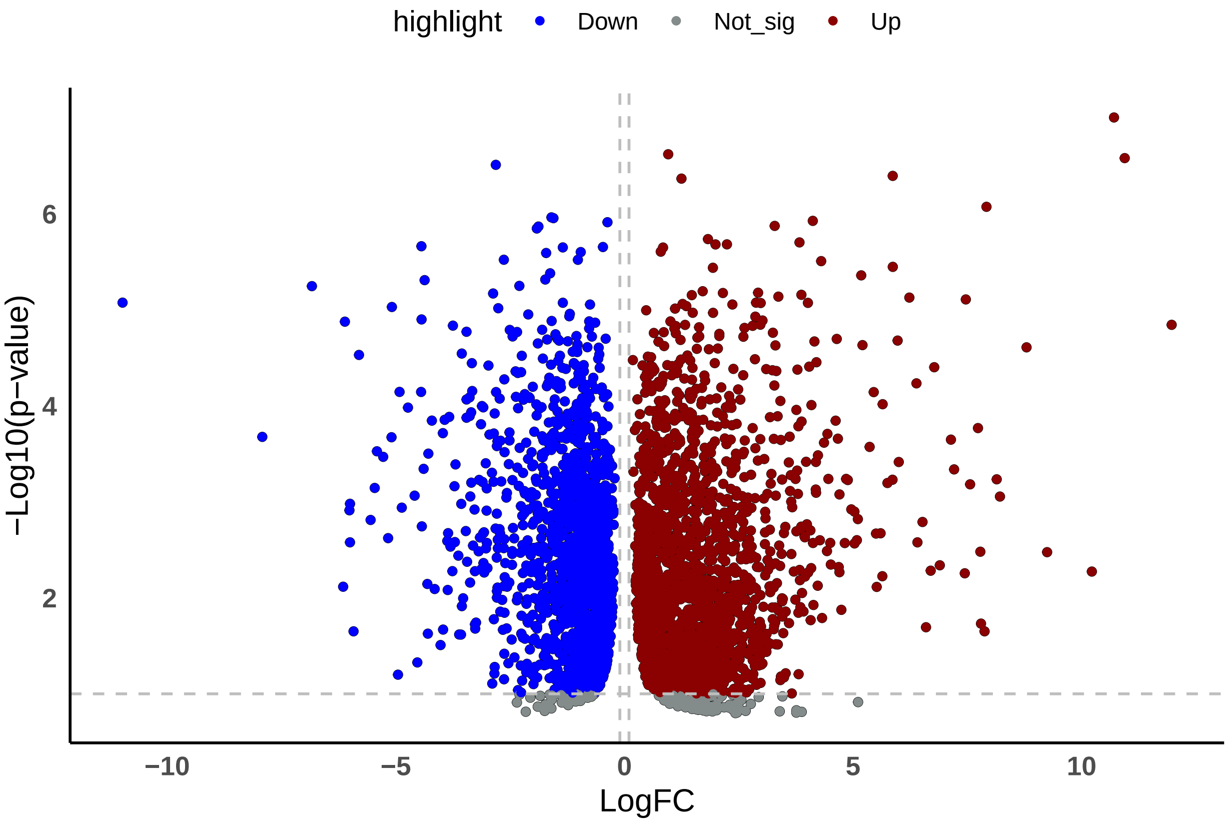
#

Save upset plot

```
#png(filename = "Upset.d34_vs_d22.perseus.DEP.png", width = 150, height = 130 , units = "mm", res = 300
#p
#dev.off()
```

Generate volcano plot for proteomics results

```
pro <- result[['pro']]
pro <- pro %>%
  mutate(
    highlight = case_when(
      Pvalue > 1 & LogFC > 0.1 ~ "Up",
      Pvalue > 1 & LogFC < -0.1 ~ "Down",
      TRUE ~ "Not_sig"
    )
  )

color_map <- c(
  "Up" = "darkred",
  "Not_sig" = "azure4",
  "Down" = "blue"
)

volcano <- ggplot(pro,
                  aes(x = LogFC, y = Pvalue, colour= highlight)) +
  geom_point(cex = 1.0, stroke = 0.6, color = "black") +
  geom_point(aes(fill = highlight), cex = 1.0, stroke = 0.5) +
  theme_minimal() +
  labs(x="LogFC",
       y="-Log10(p-value)") +
  scale_color_manual(values=color_map) +
  theme(legend.position="top", axis.line = element_line(color = "black"),
        panel.background = element_blank(),
        plot.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        axis.text = element_text(face = "bold", size = 10)) +
  geom_hline(yintercept=1, linetype="dashed", color = "gray") +
  geom_vline(xintercept=c(-0.1, 0.1), linetype="dashed", color = "gray")

plot(volcano)
```



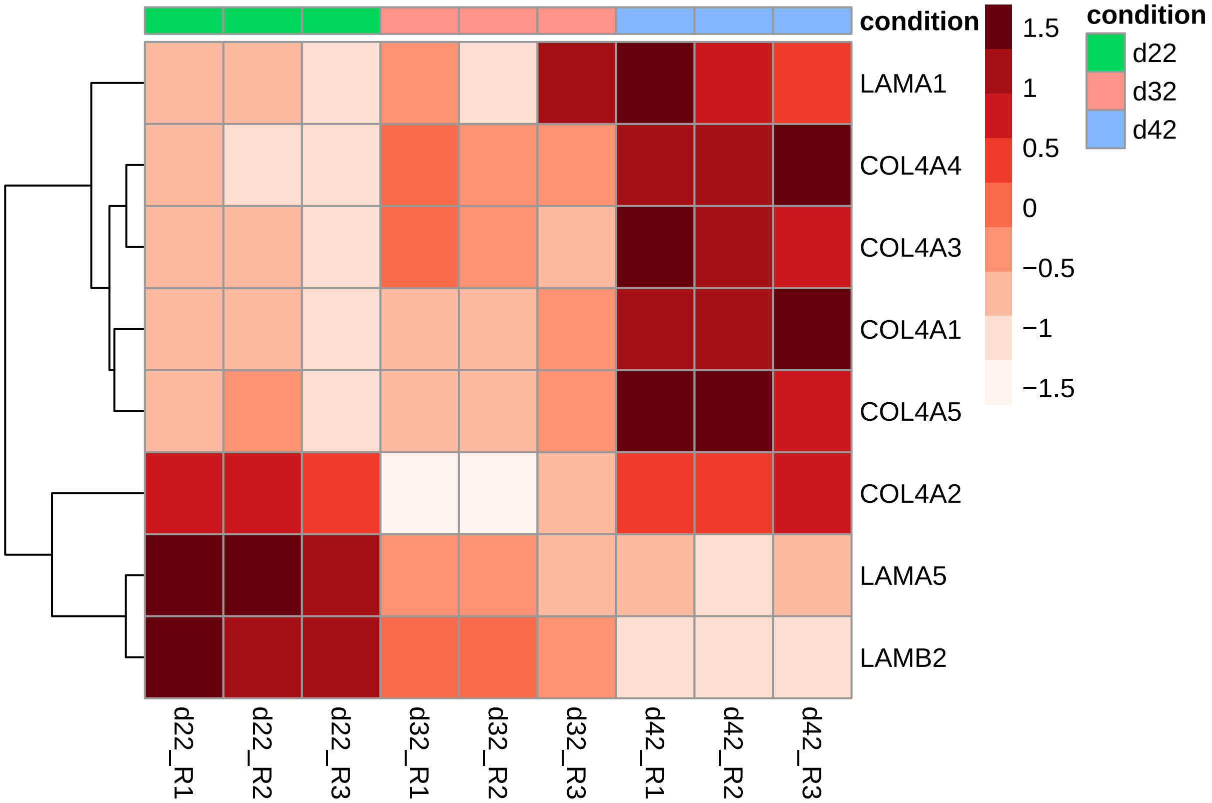
# Save volcano plot for d38 versus d22 comparison (proteomics)

```
#png(filename = "Volcano.d38_vs_d22.perseus.DEP.png", width = 100, height = 100, units = "mm", res = 30
#plot(volcano)
#dev.off()
```

Heatmap for GBM genes in bulk-RNA-seq dataset

```
sampleTable <- read.table("Listes_Res_MultiTests/DataNormDESeq2.txt", header = T)
sampleTable <- sampleTable[,c(1:9)]
sampleTable$GENEID <- rownames(sampleTable)
sampleTable <- merge(sampleTable, ens2sym_entrez, by="GENEID")
sampleTable <- sampleTable %>%
  distinct(SYMBOL, .keep_all = TRUE)
rownames(sampleTable) <- sampleTable$SYMBOL
sampleTable <- sampleTable[,-c(1)]
sampleTable <- sampleTable[,-10]
sampleTable$Gene.names <- rownames(sampleTable)
sampleTable <- sampleTable[,c("Gene.names", "d22_R1", "d22_R2","d22_R3", "d32_R1", "d32_R2", "d32_R3",

sampleTable <- sampleTable[sampleTable$Gene.names %in%
                             c("COL4A1", "COL4A2", "COL4A3", "COL4A4", "COL4A5", "LAMA1", "LAMA5", "LAM

sampleTable <- sampleTable[,-1]
sampleTable <- t(apply(sampleTable,1, scale))
colnames(sampleTable) <- c("d22_R1", "d22_R2", "d22_R3","d32_R1", "d32_R2", "d32_R3", "d42_R1", "d42_R2
```

```r
my_sample_col <- data.frame(condition = rep(c("d22", "d32", "d42"), c(3,3,3)))
row.names(my_sample_col) <- colnames(sampleTable)

pheatmap(sampleTable,
         clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean",
         clustering_method = "complete", cluster_cols=F,
         main = "", annotation_col = my_sample_col,
         fontsize = 10, color = brewer.pal(n = 9, name = "Reds"))
```


condition
LAMA1
COL4A4
COL4A3
COL4A1
COL4A5
COL4A2
LAMA5
LAMB2
d22_R1
d22_R2
d22_R3
d32_R1
d32_R2
d32_R3
d42_R1
d42_R2
d42_R3
condition
d22
d32
d42