EV proteomic analysis for mitohondrial and surface proteins

Adrian Parrilla

Intoduction

This script will identify mitochondrial and surface proteins in large and small EVs using the Mitocarta, Cell Surface Protein atlas and Human CD markers databases.

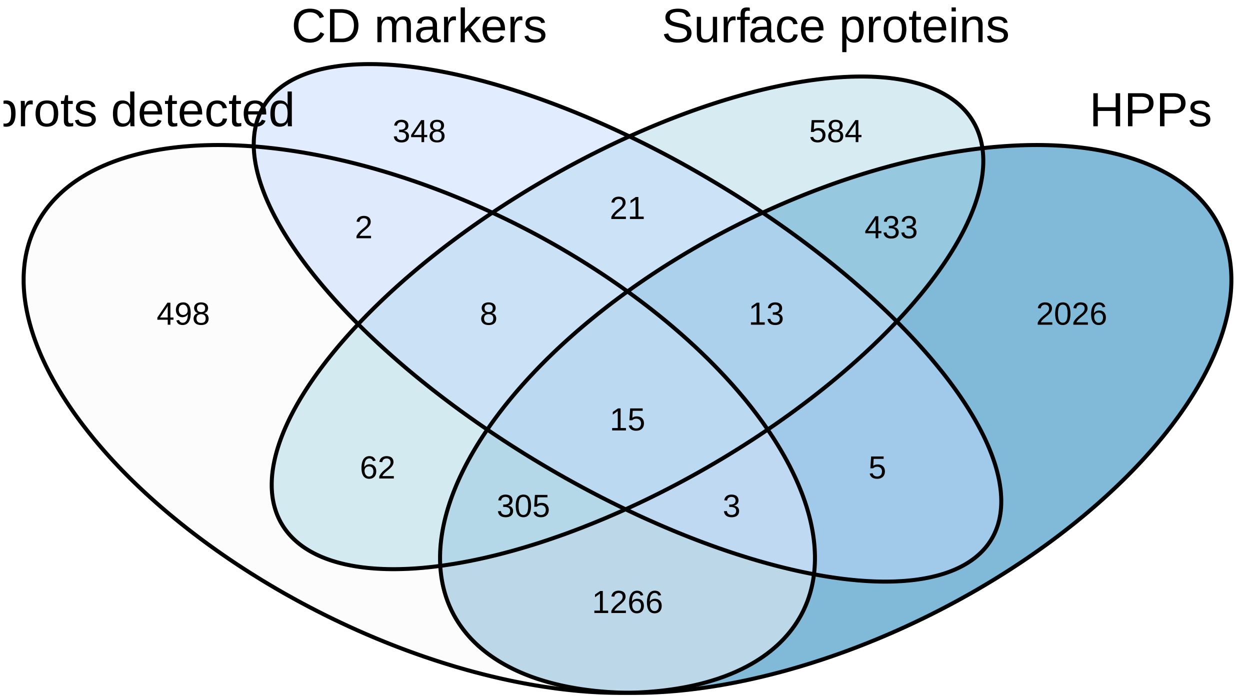
```
#Load needed libraries
library(tidyverse)
library(VennDiagram)
library(writexl)
library(readxl)
library(pheatmap)
library(RColorBrewer)
library(org.Hs.eg.db)
library(clusterProfiler)
library(ggrepel)
```

```
#Load data and reference datasets
setwd("../data")

mitocarta <- read_excel("Mitocarta.xlsx")
vesiclepedia <- read_excel("Vesiclepedia.xlsx")
HPPs <- read_excel("Human plasma proteins.xlsx")
CD_proteins <- read_excel("CD proteins.xlsx")
CSPA <- read_excel("Surface Proteins.xlsx")

full_ds <- read.table("Full_dataset_V2.csv", header = TRUE, sep = ",",
                      stringsAsFactors = FALSE)
```

```
# Definig Mitocarta evidence to Literature, manual or APEX_matrix | IMS  (OPTIONAL)

mitocarta_sub <- mitocarta[grep(c("literature|manual|APEX_matrix|APEX_IMS"),
                                mitocarta$Evidence),]
```

Overview of proteins detected

```
venn.plot.Vesic.HPPs.CD <- venn.diagram(
  x = list(full_ds$Gene, HPPs$Gene, CD_proteins$CD, CSPA$`ENTREZ gene symbol`),
  category.names = c("EV prots detected" , "HPPs", "CD markers", "Surface proteins"),
  filename = NULL,
  fill = c("#F9F9F9","#0072B4","#C3DBFD", "lightblue"),
  alpha = 0.5,
  label.col = "black",
  cex = 1,
  fontfamily = "sans",
  cat.cex = 1.5,
  cat.fontfamily = "sans",
  #cat.pos = c(-10, 5, 180),
  #cat.dist = c(0.04, 0.04),
  euler.d=FALSE,
  scaled=FALSE
)

grid.newpage()
grid.draw(venn.plot.Vesic.HPPs.CD)
```



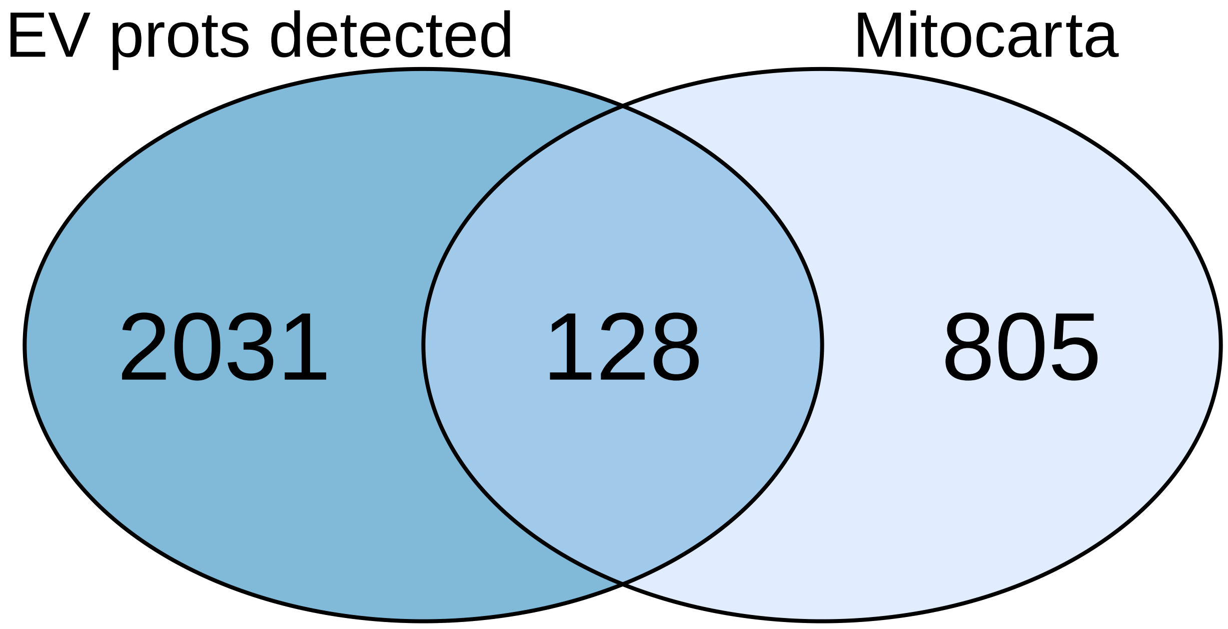
```
unlink(list.files()[grep(".log", list.files())])
```

Identification of Mitochondrial proteins

```
setwd("../outputs/")

# Count how many mitoprots have been detected
table(full_ds$Gene %in% mitocarta_sub$Symbol)
```

```
##
## FALSE  TRUE
##  2031   128
```

```
# Create the Venn diagram
venn.plot.mitoc <- venn.diagram(
  x = list(full_ds$Gene, mitocarta_sub$Symbol),
  category.names = c("EV prots detected" , "Mitocarta"),
  filename = NULL,
  fill = c("#0072B4","#C3DBFD"),
  alpha = 0.5,
  label.col = "black",
  cex = 3,
  fontfamily = "sans",
  cat.cex = 2,
  cat.fontfamily = "sans",
  cat.pos = c(-20, 20),
  cat.dist = c(0.04, 0.04),
  euler.d=FALSE,
  scaled=FALSE
)

unlink(list.files()[grep(".log", list.files())])  # remove the log files generated

grid.newpage()
grid.draw(venn.plot.mitoc)
```



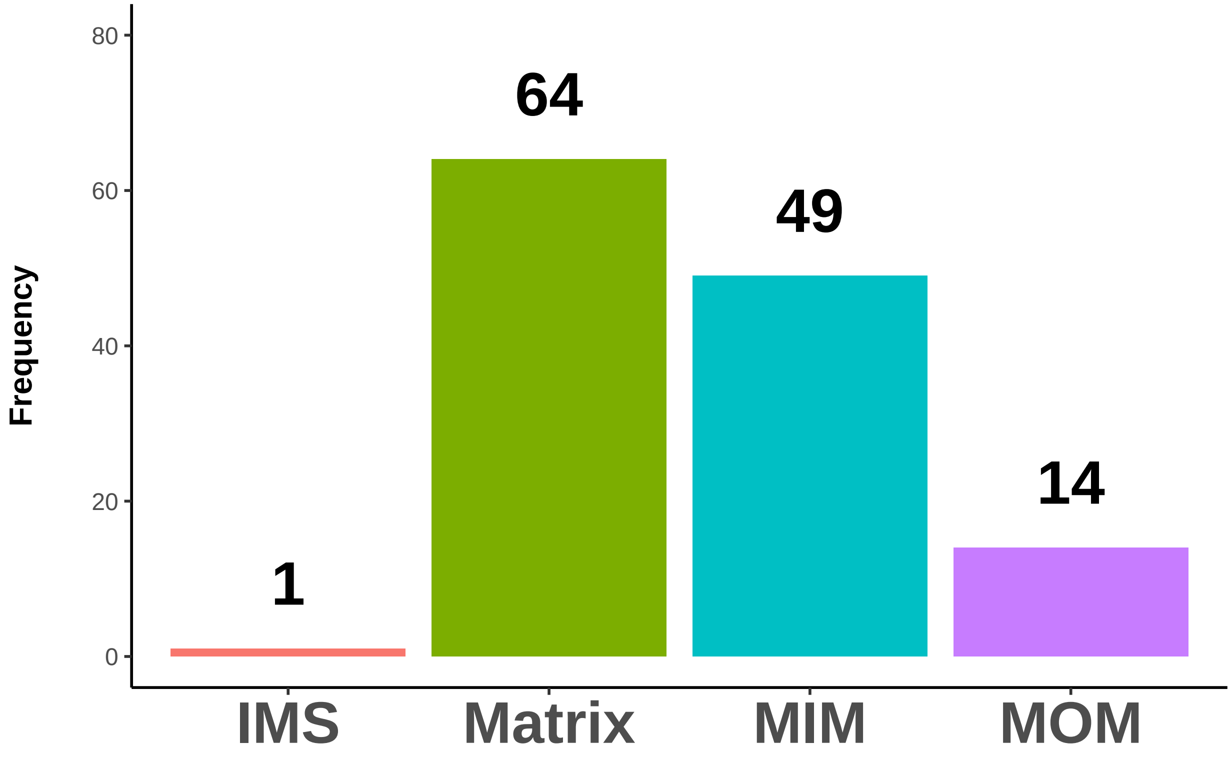
```
# Subset the results table to only include those rows that have mitoprots
#and select columns with normalised intensity
full_ds_mito <- full_ds %>% filter(full_ds$Gene %in% mitocarta_sub$Symbol) %>%
  select(Gene, Protein.Description, starts_with("norm_reporter_intensity"))
```

```
# Add information about the proteins from the Mitocarta dataset
full_ds_mito <- merge(full_ds_mito, mitocarta[,c("Symbol","Evidence",
                                                 "SubMitoLocalization", "MitoPathways")],
                      by.x = "Gene", by.y = "Symbol", all.x = TRUE)

#Save the resutls in an excel file
write_xlsx(full_ds_mito, "Mitochondrial proteins detected in EVs.xlsx")
```

```
# Assess the submitochondrial location of this proteins
table(full_ds_mito$SubMitoLocalization)
```

```
##
##    IMS Matrix    MIM    MOM
##      1     64     49     14
```

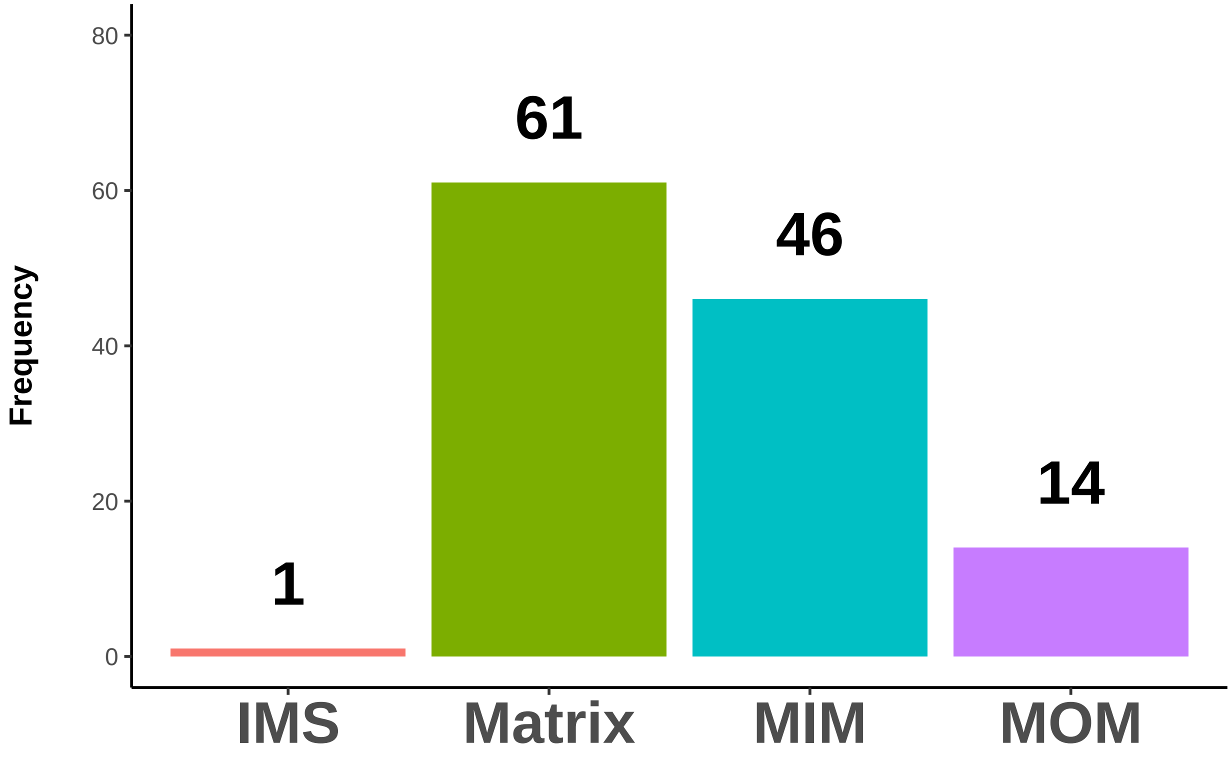
```
# Make a barplot with it
full_ds_mito %>% group_by(SubMitoLocalization) %>% summarise(Frequency = n()) %>%
  ggplot(aes(x = SubMitoLocalization, y = Frequency, fill=SubMitoLocalization)) +
  geom_bar(stat = "identity")+
  geom_text(aes(label=Frequency), vjust= -1, fontface = "bold", size = 8)+
  xlab("")+
  ylim(c(0,80))+
  theme_classic()+
  theme(legend.position="none",
        axis.title.y = element_text(face = "bold", margin = margin(0,20,0,0),
                                    size = rel(1.1), color = 'black'),
        axis.title.x = element_text(hjust = 0.5, face = "bold",
                                    margin = margin(20,0,0,0),
                                    size = rel(1.1), color = 'black'),
        axis.text.x = element_text(face = "bold", size = rel(2.5)),
        plot.title = element_text(hjust = 0.5))
```



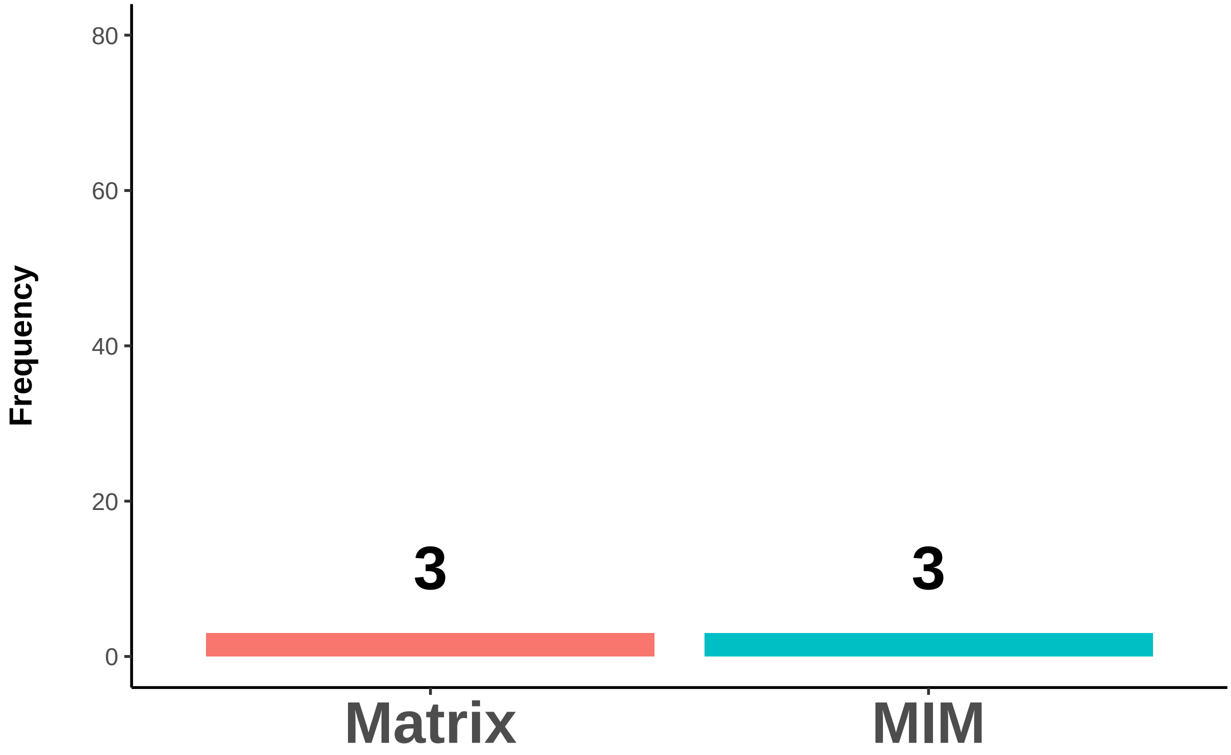
Same analysis for LEV and SEV separately

```r
# make a new data frame for LEV and SEV
full_ds_mito_LEV <- full_ds_mito[,c(1, grep("LEV_",colnames(full_ds_mito)), 20)]
full_ds_mito_LEV <- full_ds_mito_LEV %>% filter(rowSums(full_ds_mito_LEV[2:8])>= 0)

full_ds_mito_SEV <- full_ds_mito[,c(1, grep("SEV_",colnames(full_ds_mito)), 20)]
full_ds_mito_SEV <- full_ds_mito_SEV %>% filter(rowSums(full_ds_mito_SEV[2:8])>= 0)
```

```r
full_ds_mito_LEV %>%
  group_by(SubMitoLocalization) %>% summarise(Frequency = n()) %>%
  ggplot(aes(x = SubMitoLocalization, y = Frequency, fill=SubMitoLocalization)) +
  geom_bar(stat = "identity")+
  geom_text(aes(label=Frequency), vjust= -1, fontface = "bold", size = 8)+
  xlab("")+
  ylim(c(0,80))+
  theme_classic()+
  theme(legend.position="none",
        axis.title.y = element_text(face = "bold", margin = margin(0,20,0,0),
                                    size = rel(1.1), color = 'black'),
        axis.title.x = element_text(hjust = 0.5, face = "bold",
                                    margin = margin(20,0,0,0),
                                    size = rel(1.1), color = 'black'),
        axis.text.x = element_text(face = "bold", size = rel(2.5)),
        plot.title = element_text(hjust = 0.5))
```

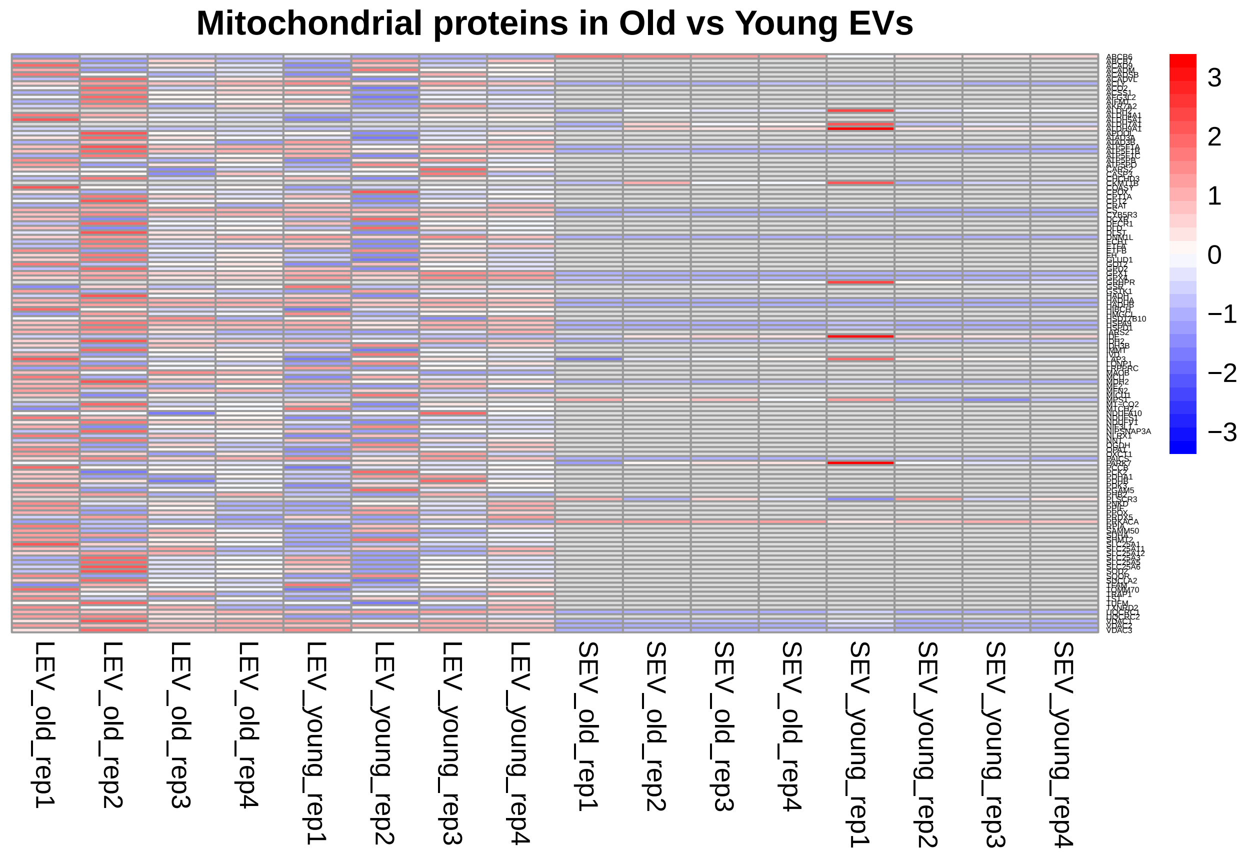


```
full_ds_mito_SEV %>%
  filter(Gene %in% c("ALDH2", "ALDH7A1", "CKMT1B", "GRHPR", "MPST", "PLSCR3")) %>%
  group_by(SubMitoLocalization) %>% summarise(Frequency = n()) %>%
  ggplot(aes(x = SubMitoLocalization, y = Frequency, fill=SubMitoLocalization)) +
  geom_bar(stat = "identity")+
  geom_text(aes(label=Frequency), vjust= -1, fontface = "bold", size = 8)+
  xlab("")+
  ylim(c(0,80))+
  theme_classic()+
  theme(legend.position="none",
        axis.title.y = element_text(face = "bold", margin = margin(0,20,0,0),
                                    size = rel(1.1), color = 'black'),
        axis.title.x = element_text(hjust = 0.5, face = "bold",
                                    margin = margin(20,0,0,0),
                                    size = rel(1.1), color = 'black'),
        axis.text.x = element_text(face = "bold", size = rel(2.5)),
        plot.title = element_text(hjust = 0.5))
```

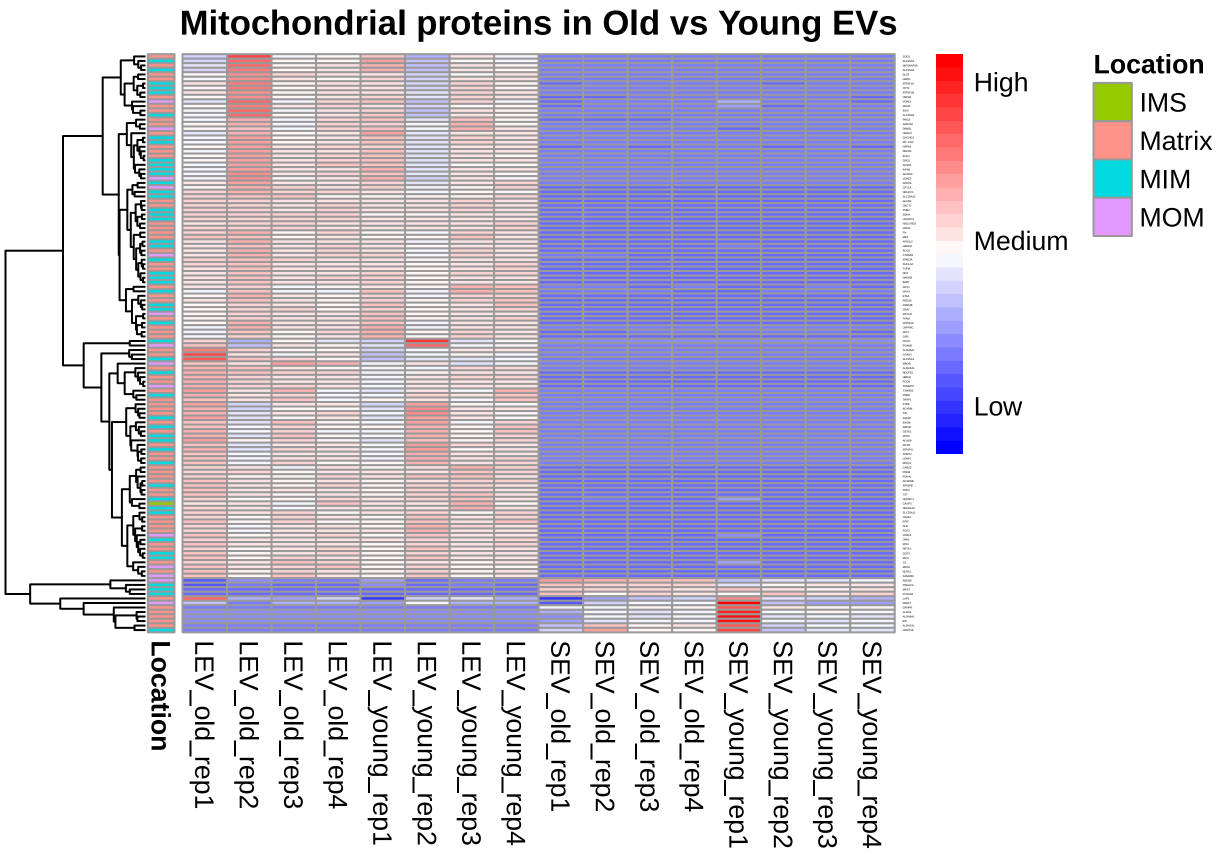


```
# Save proteins that are uniquely located in LEV or SEV
full_ds_mito_LEV_unique <- full_ds_mito_LEV[which(!full_ds_mito_LEV$Gene %in% full_ds_mito_SEV$Gene),]
#write_xlsx(full_ds_mito_LEV_unique, "Mitoprots LEV unique.xlsx")

full_ds_mito_SEV_unique <- full_ds_mito_SEV[which(!full_ds_mito_SEV$Gene %in% full_ds_mito_LEV$Gene),]
#write_xlsx(full_ds_mito_SEV_unique, "Mitoprots SEV unique.xlsx")
```

```
# Make a heatmap with mitoprots
heatmap_data <- full_ds_mito %>% select(Gene, starts_with("norm_reporter_intensity")) %>%
  column_to_rownames("Gene")
# Remove the prefix from the column names
colnames(heatmap_data)<- gsub("norm_reporter_intensity_", "", colnames(heatmap_data))

pheatmap(heatmap_data,
         show_rownames = T,
         cluster_rows = F, cluster_cols = F, scale = "row",
         colorRampPalette(colors = c("blue", "white", "red"))(30),
         main = "Mitochondrial proteins in Old vs Young EVs",
         fontsize_row = 3, fontsize_col = 10)
```

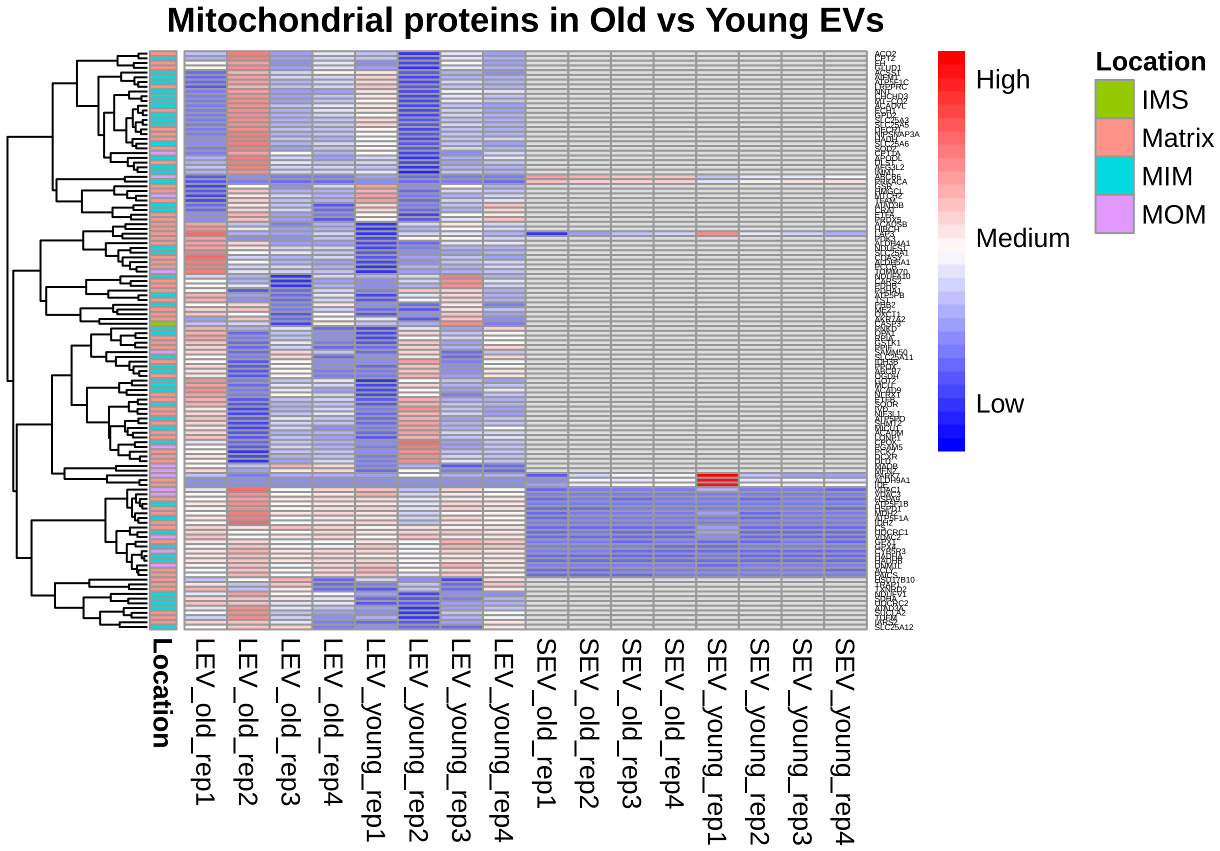


```r
#set NA to 0 so we can do row clustering
heatmap_data_0 <- heatmap_data
heatmap_data_0[is.na(heatmap_data_0)] <- 0

# Add Mitochondrial location as row annotations
row_annot <- data.frame(Location = full_ds_mito$SubMitoLocalization)
rownames(row_annot) <- rownames(heatmap_data_0)

pheatmap(heatmap_data_0,
         show_rownames = T,
         cluster_rows = T, cluster_cols = F, scale = "row",
         colorRampPalette(colors = c("blue", "white", "red"))(30),
         main = "Mitochondrial proteins in Old vs Young EVs",
         breaks = c(seq(-2.3,3.5,length.out=31)),
         legend_breaks = c(-1.6,0.8,3.1),
         annotation_row = row_annot,
         legend_labels = c("Low", "Medium", "High"),
         fontsize_row = 1, fontsize_col = 10)
```

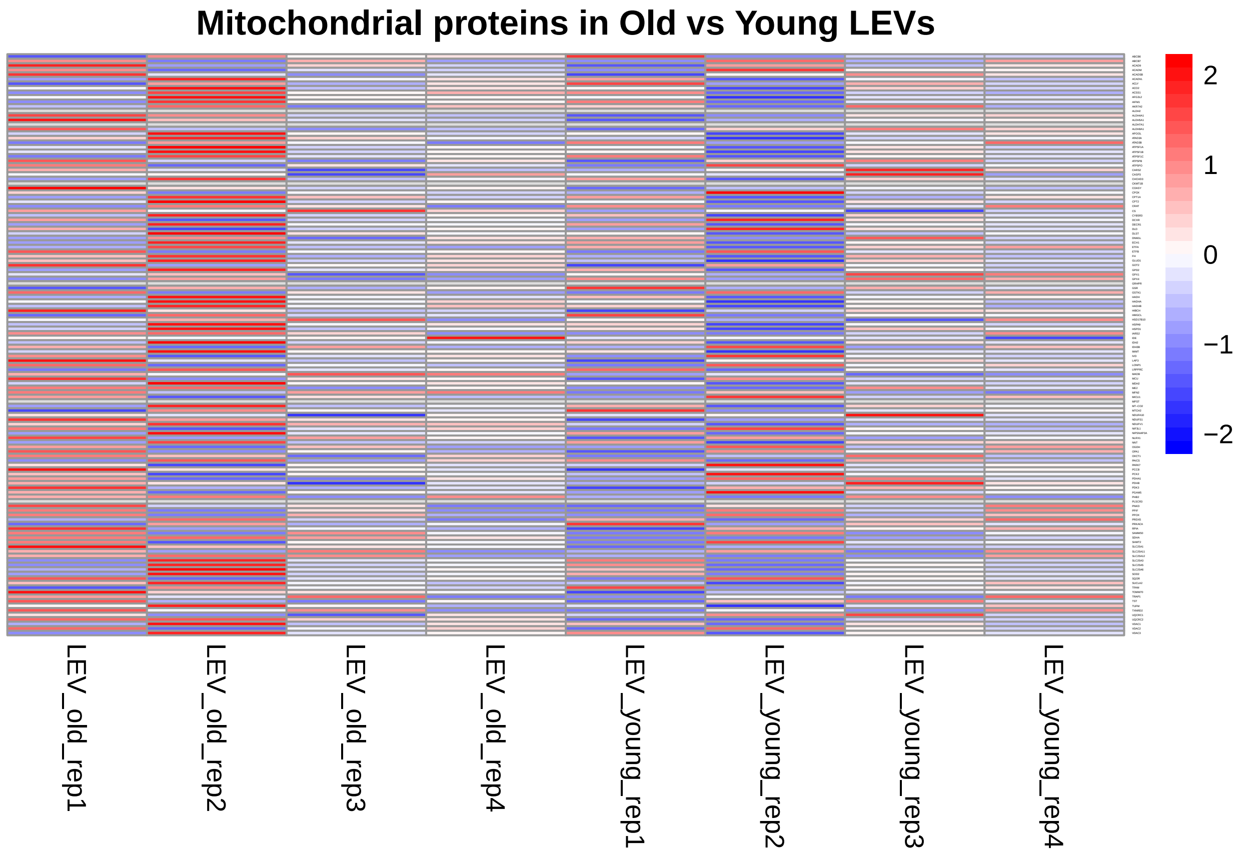


Clustering keeping NAs

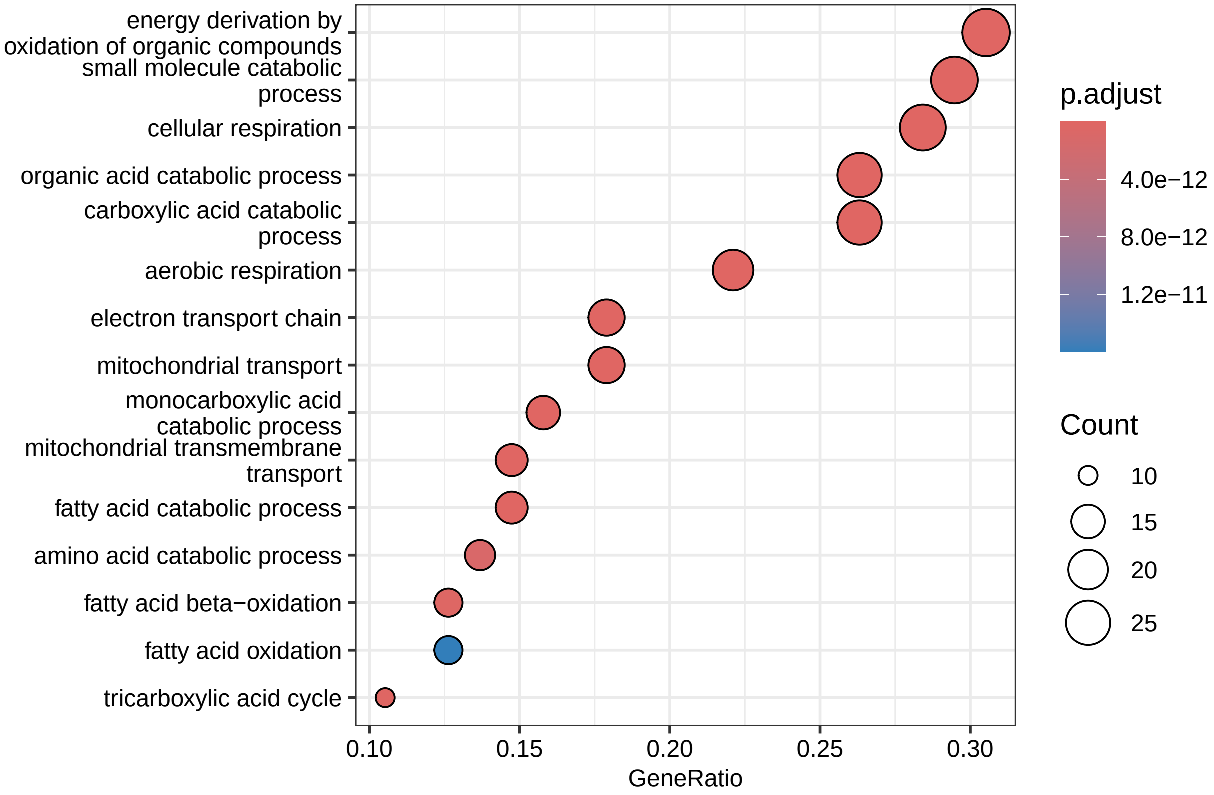
```
#Apply Z score normalization
heatmap_data_scaled <- as.data.frame(t(apply(heatmap_data[,c(1:16)], 1, function(x) {
  (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
})))

# remove those rows whose distance matrix can not be calculated in case clustering is not working
giveNAs = which(is.na(as.matrix(dist(heatmap_data_scaled))),arr.ind=TRUE)
tab = sort(table(c(giveNAs)),decreasing=TRUE)
checkNA = sapply(1:length(tab),function(i){
  sum(is.na(as.matrix(dist(heatmap_data_scaled[-as.numeric(names(tab[1:i])),]))))
})

rmv = names(tab)[1:min(which(checkNA==0))]
heatmap_data_scaled = heatmap_data_scaled[-as.numeric(rmv),]

row_annot_scl <- data.frame(
  Location = full_ds_mito$SubMitoLocalization[full_ds_mito$Gene %in% rownames(heatmap_data_scaled)])

rownames(row_annot_scl) <- rownames(heatmap_data_scaled)

pheatmap(heatmap_data_scaled,
         show_rownames = T,
         cluster_rows = T, cluster_cols = F, scale = "none",
         colorRampPalette(colors = c("blue", "white", "red"))(30),
         main = "Mitochondrial proteins in Old vs Young EVs",
         breaks = c(seq(-2.3,3.5,length.out=31)),
         legend_breaks = c(-1.6,0.8,3.1),
         annotation_row = row_annot_scl,
         legend_labels = c("Low", "Medium", "High"),
         fontsize_row = 3, fontsize_col = 10)
```



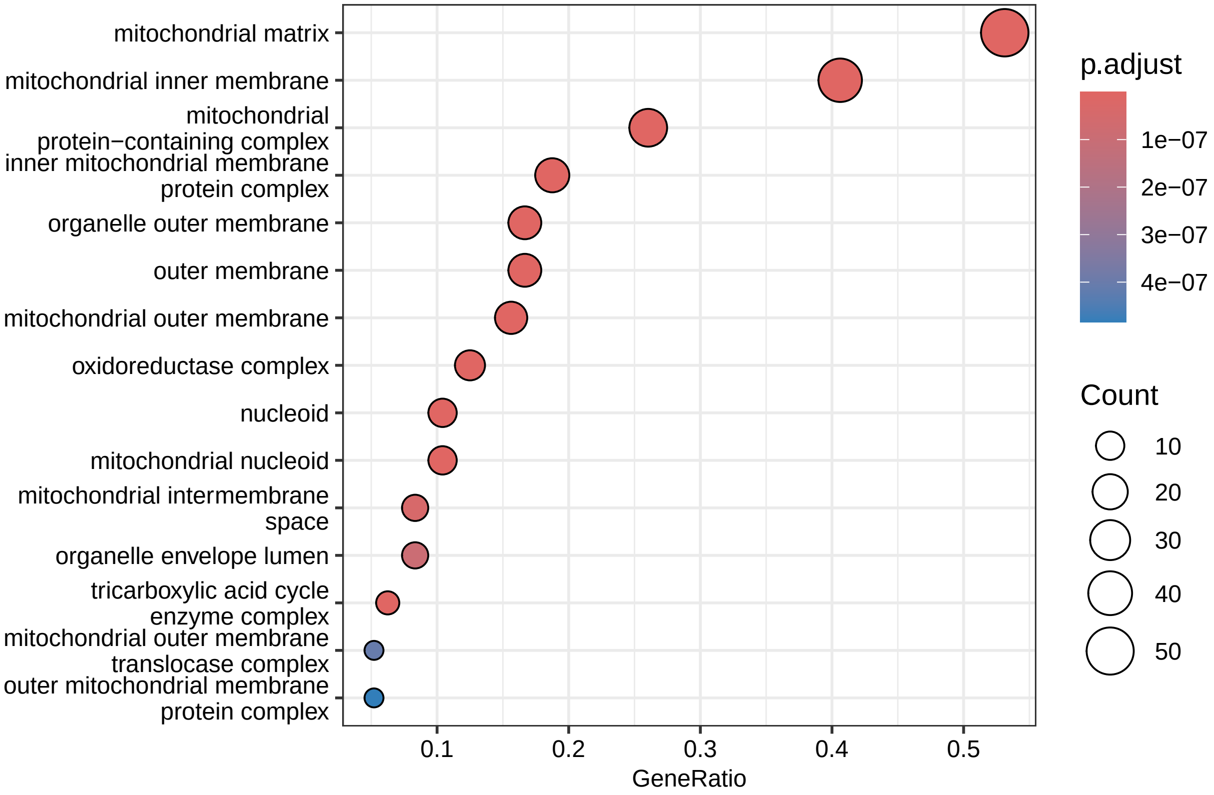
```
# Larve EVs mito prots only

heatmap_data_LEV <- heatmap_data_0 %>% select(starts_with("LEV"))
heatmap_data_LEV_0 <- heatmap_data_LEV %>% filter(rowSums(heatmap_data_LEV)>= 0)

pheatmap(heatmap_data_LEV_0,
         show_rownames = T,
         cluster_rows = F, cluster_cols = F,
         scale = "row",
         colorRampPalette(colors = c("blue", "white", "red"))(30),
         main = "Mitochondrial proteins in Old vs Young LEVs",
         #legend_breaks = c(seq(-1,3,length.out=4)),
         #legend_labels = c("NAs", "Low", "Medium", "High"),
         fontsize_row = 1, fontsize_col = 10)
```



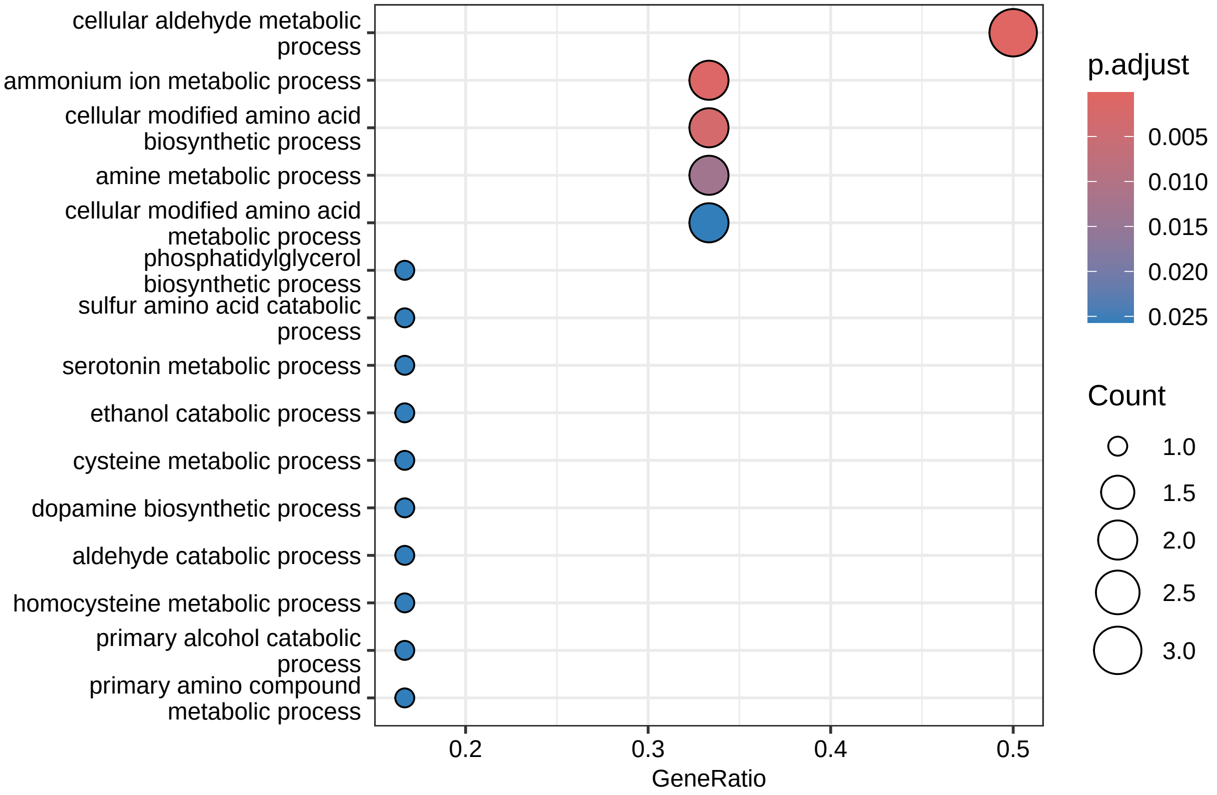
```r
#GO analysis

ego.mito.LEV.BP <- enrichGO(gene = full_ds_mito_LEV_unique$Gene,
                OrgDb         = 'org.Hs.eg.db',
                ont           = "BP", #it can be CC and MF
                keyType       = "SYMBOL",
                pAdjustMethod = "BH",
                qvalueCutoff = 0.05,
                readable      = TRUE)

dotplot(ego.mito.LEV.BP, showCategory=15, font.size= 9)
```



```
ego.mito.LEV.CC <- enrichGO(gene = full_ds_mito_LEV_unique$Gene,
                OrgDb         = 'org.Hs.eg.db',
                ont           = "CC",
                keyType       = "SYMBOL",
                pAdjustMethod = "BH",
                qvalueCutoff = 0.05,
                readable      = TRUE)

dotplot(ego.mito.LEV.CC, showCategory=15, font.size= 9)
```



```
ego.mito.SEV.BP <- enrichGO(gene = full_ds_mito_SEV_unique$Gene,
                            OrgDb         = 'org.Hs.eg.db',
                            ont           = "BP",
                            keyType       = "SYMBOL",
                            pAdjustMethod = "BH",
                            qvalueCutoff = 0.05,
                            readable      = TRUE)

dotplot(ego.mito.SEV.BP, showCategory=15, font.size= 9)
```

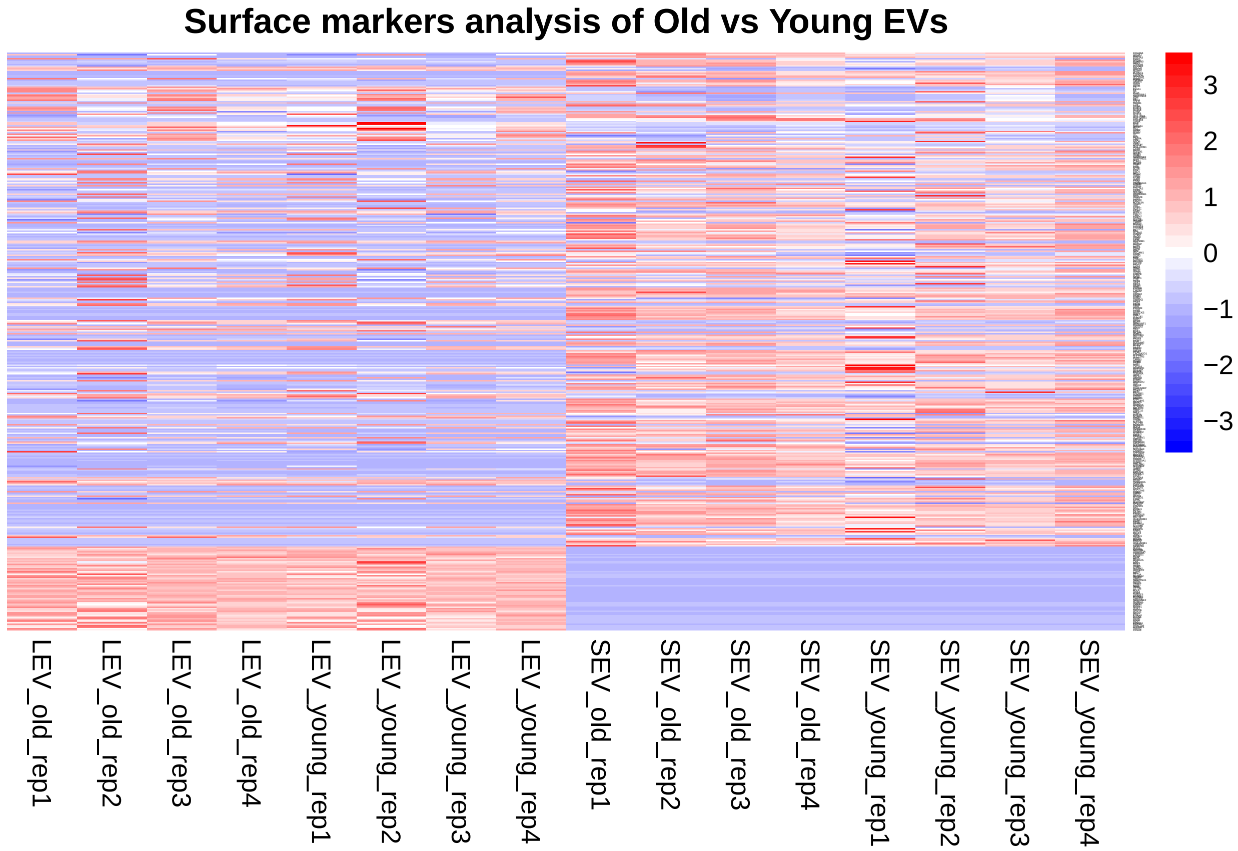


Surfaceome analisys: Identification of surface markers

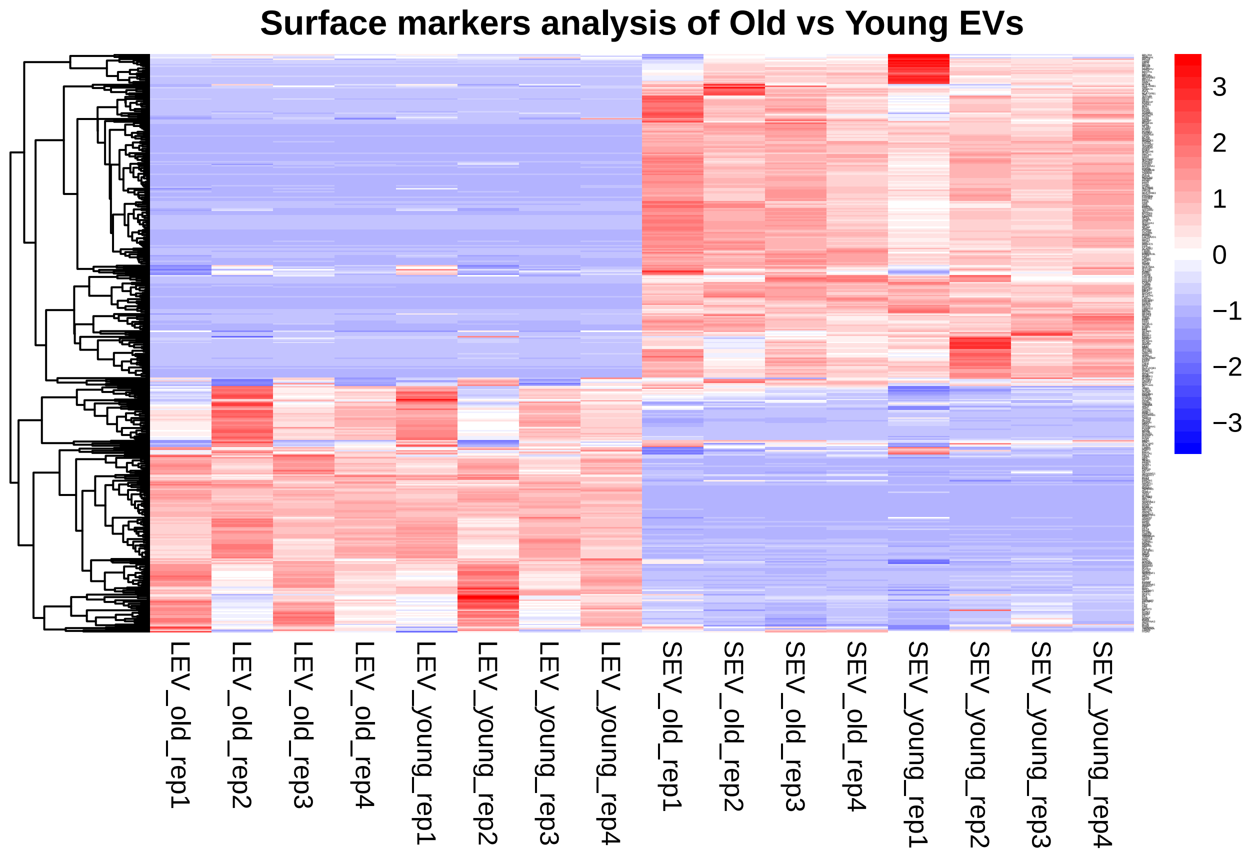
```
# make a vector for the alternative names of CD markers
CD_alternative <- unlist(strsplit(CD_proteins$`Alternative name`, ",\\s*"))
surf <- c()

# For each gene detected, look for coincidences in the 3 datasets
for (i in full_ds$Gene){
  if (i %in% CSPA$`ENTREZ gene symbol` | i %in%  CD_proteins$CD | i %in% CD_alternative){
    surf <- append(surf, i)
  }
}

full_ds_surf <- full_ds[full_ds$Gene %in% surf,]
full_ds_surf <- full_ds_surf%>% select(Gene, Protein.Description, starts_with("norm_reporter_intensity")
write_xlsx(full_ds_surf, "Surface proteins detected in EVs.xlsx")
```

```
# Heatmap
rownames(full_ds_surf)<- NULL

heatmap_surf <- full_ds_surf %>% select(Gene, starts_with("norm_reporter_intensity")) %>% column_to_rown
colnames(heatmap_surf)<- gsub("norm_reporter_intensity_", "", colnames(heatmap_surf))

heatmap_surf_0 <- heatmap_surf
heatmap_surf_0[is.na(heatmap_surf_0)] <- 0
```

```r
pheatmap(heatmap_surf_0,
         show_rownames = T,
         cluster_rows = F, cluster_cols = F,
         scale = "row",
         colorRampPalette(colors = c("blue", "white", "red"))(35),
         main = "Surface markers analysis of Old vs Young EVs",
         fontsize_row = 1, fontsize_col = 10)
```

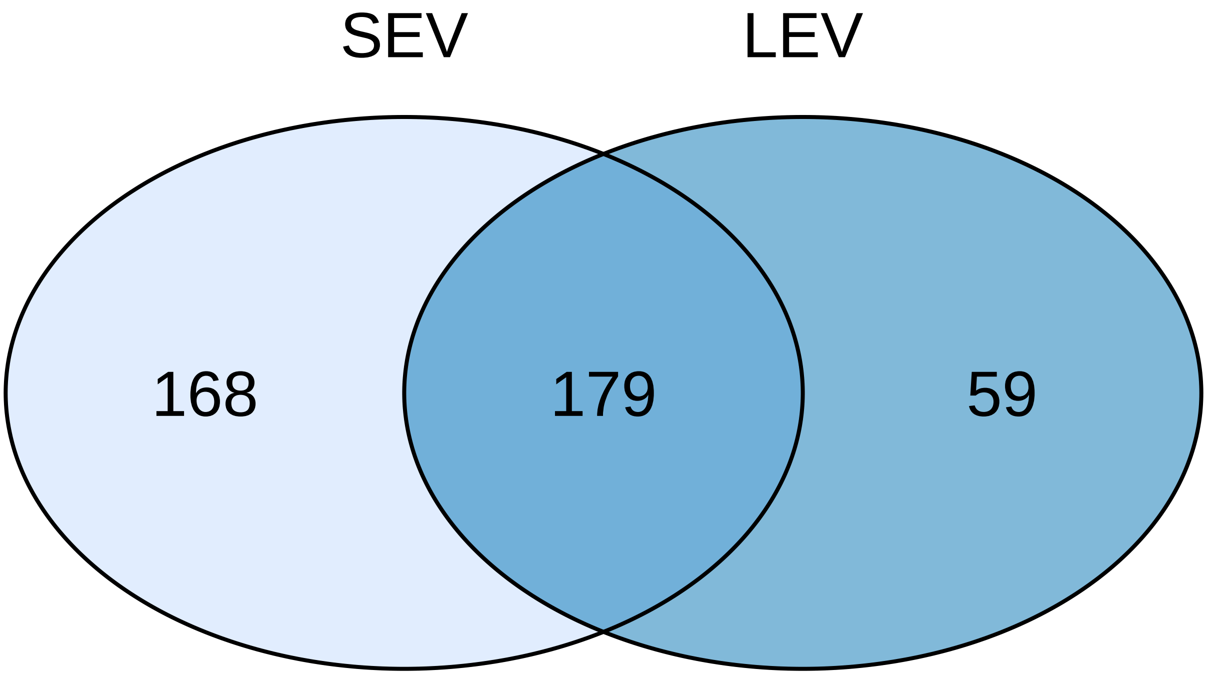


```r
pheatmap(heatmap_surf_0,
         show_rownames = T,
         cluster_rows = T, cluster_cols = F,
         scale = "row",
         colorRampPalette(colors = c("blue", "white", "red"))(35),
         main = "Surface markers analysis of Old vs Young EVs",
         fontsize_row = 1, fontsize_col = 10)
```



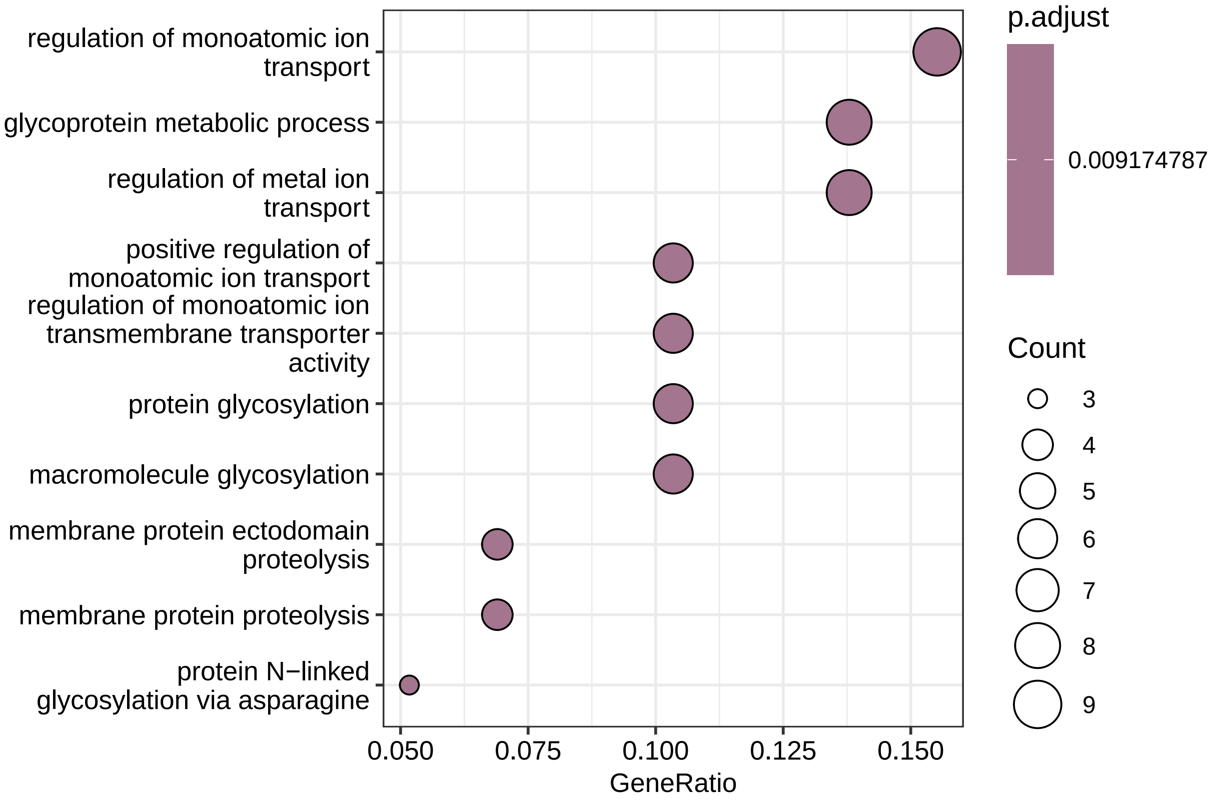
```r
# Save unique surface proteins for LEV and SEV
full_ds_surf_LEV <- full_ds_surf[,c(1, grep("LEV_",colnames(full_ds_surf)))]
full_ds_surf_LEV <- full_ds_surf_LEV[which(rowSums(full_ds_surf_LEV[,3:9])>=0),]

full_ds_surf_SEV <- full_ds_surf[,c(1, grep("SEV_",colnames(full_ds_surf)))]
full_ds_surf_SEV <- full_ds_surf_SEV[which(rowSums(full_ds_surf_SEV[,3:9])>=0),]

full_ds_surf_LEV_unique <- full_ds_surf_LEV[which(!full_ds_surf_LEV$Gene %in% full_ds_surf_SEV$Gene),]
full_ds_surf_LEV_unique <- merge(full_ds_surf_LEV_unique, full_ds_surf[,c(1,2)])
write_xlsx(full_ds_surf_LEV_unique, "Surface prots LEV unique.xlsx")

full_ds_surf_SEV_unique <- full_ds_surf_SEV[which(!full_ds_surf_SEV$Gene %in% full_ds_surf_LEV$Gene),]
full_ds_surf_SEV_unique <- merge(full_ds_surf_SEV_unique, full_ds_surf[,c(1,2)])
write_xlsx(full_ds_surf_SEV_unique, "Surface prots SEV unique.xlsx")
```

```r
# Venn diagram for LEV and SEV surface proteins
venn.plot.surface.markers.LEV.SEV <- venn.diagram(
  x = list(full_ds_surf_LEV$Gene, full_ds_surf_SEV$Gene),
  category.names = c("LEV", "SEV"),
  filename = NULL,
  fill = c("#0072B4","#C3DBFD"),
  alpha = 0.5,
  label.col = "black",
  cex = 2,
  fontfamily = "sans",
  cat.cex = 2,
  cat.fontfamily = "sans",
  cat.pos = c(0, 0),
  cat.dist = c(0.06, 0.06),
  euler.d=FALSE,
  scaled=FALSE
)

grid.newpage()
grid.draw(venn.plot.surface.markers.LEV.SEV)
```

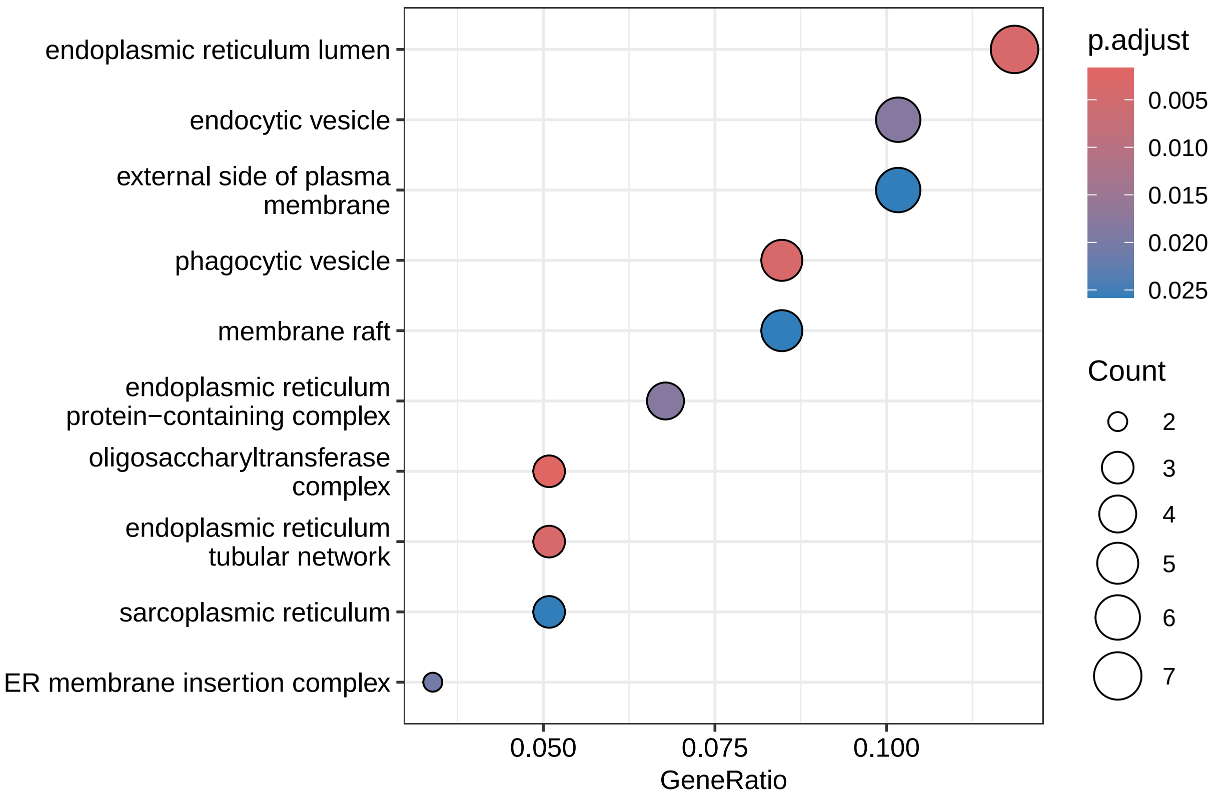


```
unlink(list.files()[grep(".log", list.files())])
```

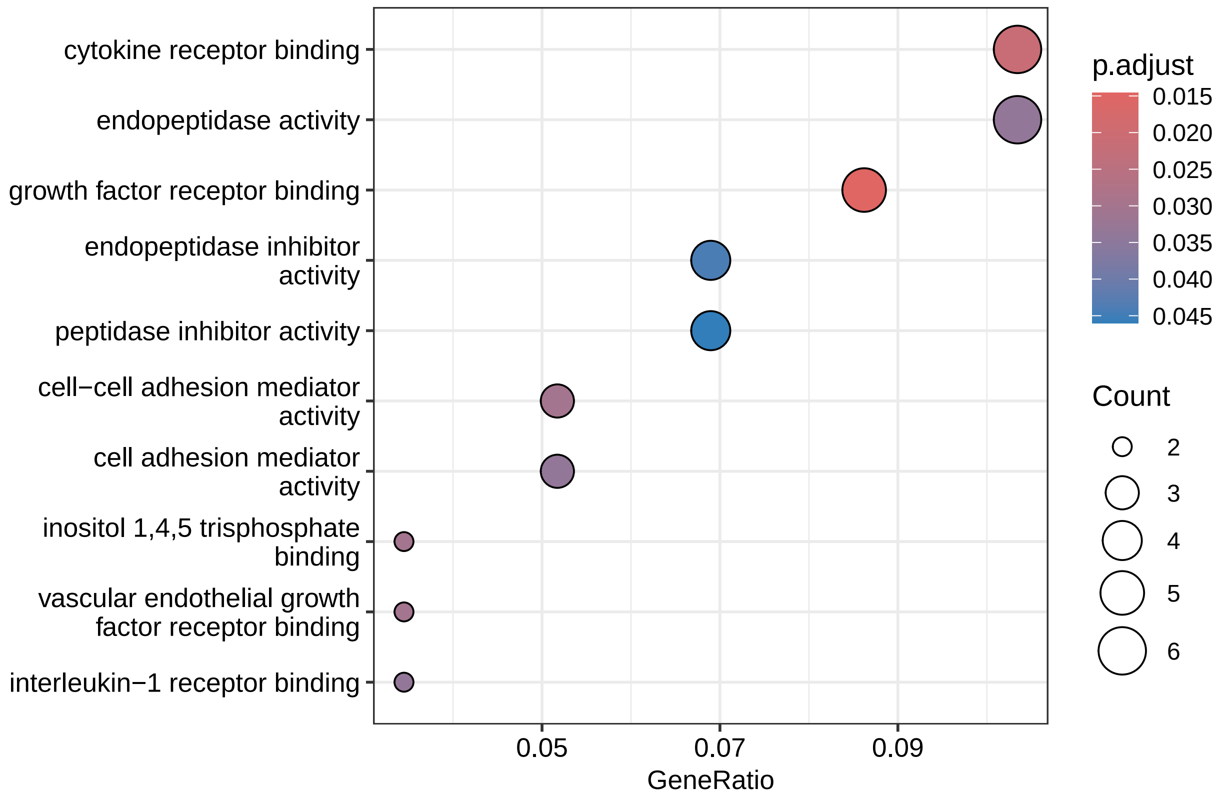
```
# GO analysis of the LEV surface proteins

LEV_surf_ego_BP <- enrichGO(gene = full_ds_surf_LEV_unique$Gene,
                            OrgDb         = 'org.Hs.eg.db',
                            ont           = "BP",
                            keyType       = "SYMBOL",
                            pAdjustMethod = "BH",
                            pvalueCutoff  = 0.05,
                            readable      = TRUE)


LEV_surf_ego_CC <- enrichGO(gene = full_ds_surf_LEV_unique$Gene,
                            OrgDb         = 'org.Hs.eg.db',
                            ont           = "CC",
                            keyType       = "SYMBOL",
                            pAdjustMethod = "BH",
                            pvalueCutoff  = 0.05,
                            readable      = TRUE)

LEV_surf_ego_MF <- enrichGO(gene = full_ds_surf_LEV_unique$Gene,
                            OrgDb         = 'org.Hs.eg.db',
                            ont           = "MF",
                            keyType       = "SYMBOL", #
                            pAdjustMethod = "BH",
                            pvalueCutoff  = 0.05,
                            readable      = TRUE)

dotplot(LEV_surf_ego_BP, showCategory=10, font.size = 10)
```



```
dotplot(LEV_surf_ego_CC, showCategory=10, font.size = 10)
```

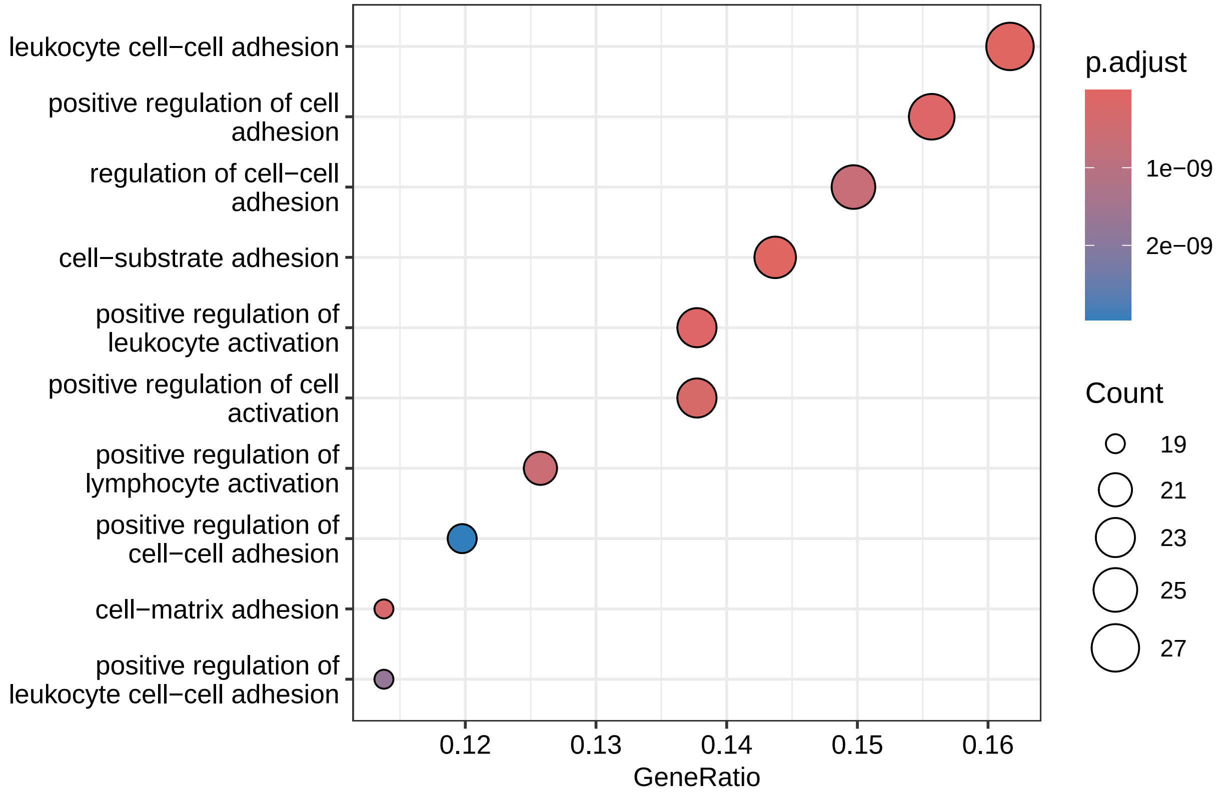


```
dotplot(LEV_surf_ego_MF, showCategory=10, font.size = 10)
```

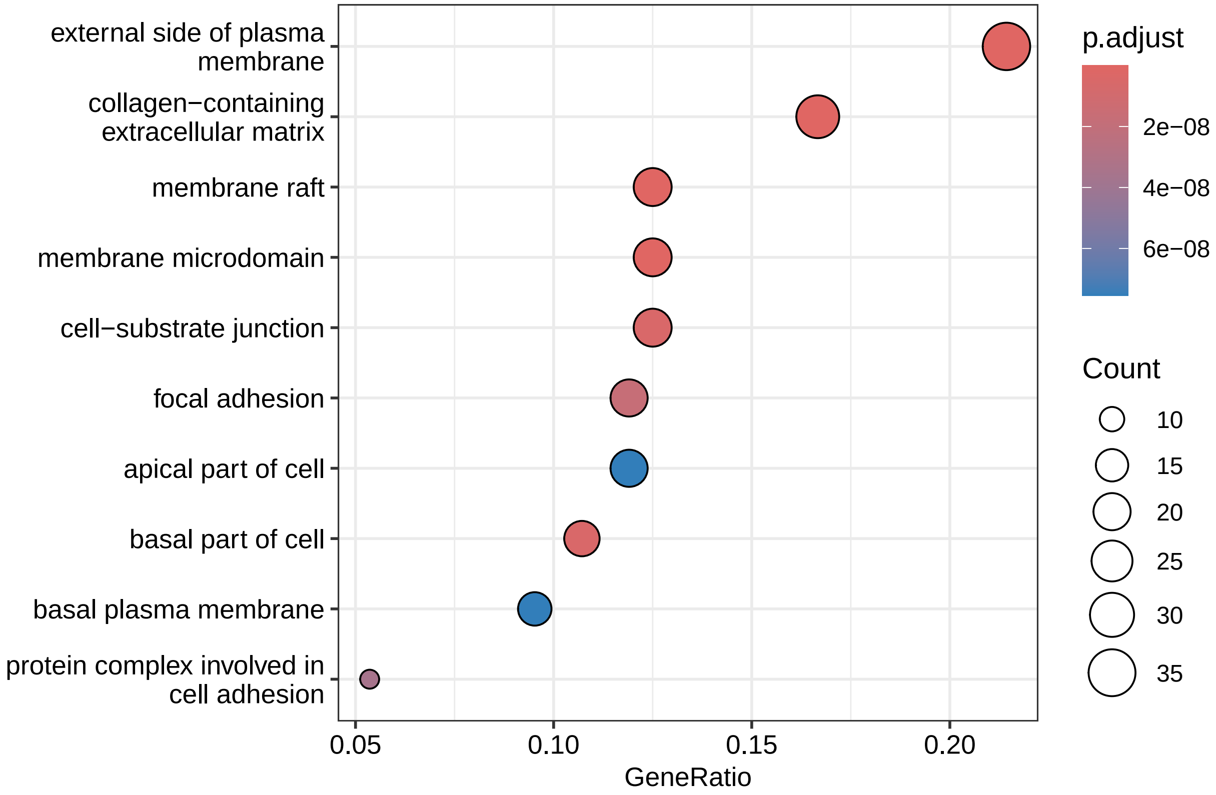


```
# GO analysis of the SEV surface proteins

SEV_surf_ego_BP <- enrichGO(gene = full_ds_surf_SEV_unique$Gene,
                            OrgDb         = 'org.Hs.eg.db',
                            ont           = "BP",
                            keyType       = "SYMBOL",
                            pAdjustMethod = "BH",
                            pvalueCutoff  = 0.05,
                            readable      = TRUE)

SEV_surf_ego_CC <- enrichGO(gene = full_ds_surf_SEV_unique$Gene,
                            OrgDb         = 'org.Hs.eg.db',
                            ont           = "CC",
                            keyType       = "SYMBOL",
                            pAdjustMethod = "BH",
                            pvalueCutoff  = 0.05,
                            readable      = TRUE)

SEV_surf_ego_MF <- enrichGO(gene = full_ds_surf_SEV_unique$Gene,
                            OrgDb         = 'org.Hs.eg.db',
                            ont           = "MF",
                            keyType       = "SYMBOL",
                            pAdjustMethod = "BH",
                            pvalueCutoff  = 0.05,
                            readable      = TRUE)
```

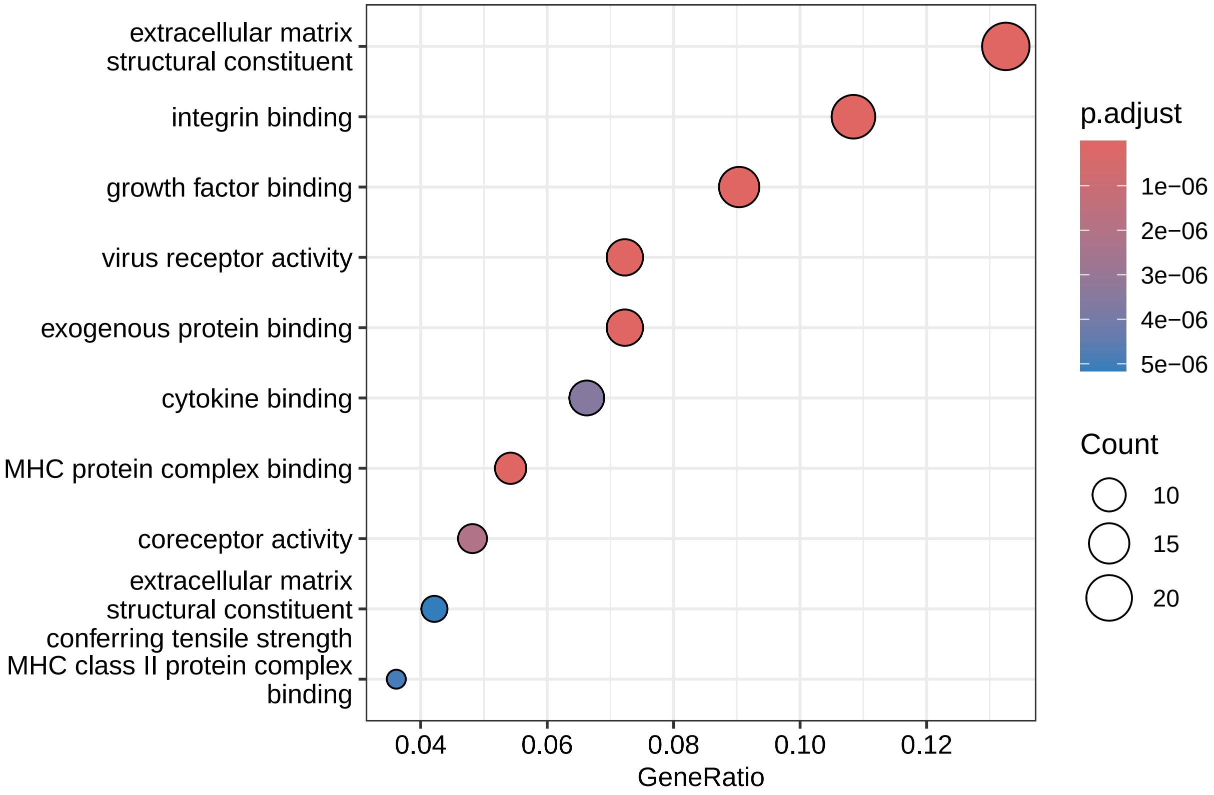
```
dotplot(SEV_surf_ego_BP, showCategory=10, font.size = 10)
```



```
dotplot(SEV_surf_ego_CC, showCategory=10, font.size = 10)
```



```
dotplot(SEV_surf_ego_MF, showCategory=10, font.size = 10)
```


extracellular matrix structural constituent
integrin binding
growth factor binding
virus receptor activity
exogenous protein binding
cytokine binding
MHC protein complex binding
coreceptor activity
extracellular matrix structural constituent conferring tensile strength
MHC class II protein complex binding
0.04
0.06
0.08
0.10
0.12
GeneRatio
p.adjust
1e−06
2e−06
3e−06
4e−06
5e−06
Count
10
15
20