

Figure-S2E.R

sokole

2024-07-19

```
# This Script Generates Figure S2E
# Script By: Eishani Kumar Sokolowski

# Empty the environment & suppress warnings
rm(list = ls())
options(warn=-1)

# Loading the required libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggrepel)
library(reshape2)
library(rstatix)
```

```
##
## Attaching package: 'rstatix'
```

```
## The following object is masked from 'package:stats':
##
##   filter
```

```
library(ggpubr)
```

```
#####  
# Loading files  
#####
```

```
# Loading DEGs  
alpha.only <- read.csv("./scRNA-seq_ERS_Alpha-specific_Upregulated_DEGs.csv")  
colnames(alpha.only) <- c("X","Genes")  
alpha.only.DEGs <- alpha.only$Genes # Making a list of DEGs
```

```
beta.only <- read.csv("./scRNA-seq_ERS_Beta-specific_Upregulated_DEGs.csv")  
colnames(beta.only) <- c("X","Genes")  
beta.only.DEGs <- beta.only$Genes # Making a list of DEGs
```

```
common <- read.csv("./scRNA-seq_ERS_Alpha_Beta_common_Upregulated_DEGs.csv")  
colnames(common) <- c("X","Genes")  
common.DEGs <- common$Genes # Making a list of DEGs
```

```
# Loading all genes files - min 10% expressed  
alpha.all.10.percent <- read.csv("./alpha.all.genes_FDR_ERS.csv")  
beta.all.10.percent <- read.csv("./beta.all.genes_FDR_ERS.csv")  
  
# Loading all genes files - min 0% expressed  
alpha.all.0.percent <- read.csv("./alpha.all.genes_FDR_ERS_NO_MIN_PERCENT_FOR_BULK_DEGs_HEATMAP.csv")  
beta.all.0.percent <- read.csv("./beta.all.genes_FDR_ERS_NO_MIN_PERCENT_FOR_BULK_DEGs_HEATMAP.csv")
```

```
#####  
# Getting LFC of DEGs in Alpha and Beta - ALPHA ONLY DEGs - UPREGULATED  
#####
```

```
# Extracting the DEGs  
alpha <- alpha.all.10.percent[alpha.all.10.percent$X %in% alpha.only.DEGs,] # Because alpha-specific DEGs  
beta <- beta.all.0.percent[beta.all.0.percent$X %in% alpha.only.DEGs,]
```

```
# Refining dataframe  
alpha <- alpha[-c(2,4:7)]  
colnames(alpha) <- c("Genes","Alpha")  
beta <- beta[-c(2,4:7)]  
colnames(beta) <- c("Genes","Beta")
```

```
# Combining  
df <- merge(alpha,beta,by="Genes")  
df <- melt(df, id = c("Genes"))  
colnames(df) <- c("Genes","Cell_Type","Avg_LFC")
```

```
# Finalizing  
df.alpha <- df  
df.alpha$Type <- "Alpha-specific DEGs"
```

```
#####
# Getting LFC of DEGs in Alpha and Beta – BETA ONLY DEGs – UPREGULATED
#####

# Extracting the DEGs
alpha <- alpha.all.0.percent[alpha.all.0.percent$X %in% beta.only.DEGs,]
beta <- beta.all.10.percent[beta.all.10.percent$X %in% beta.only.DEGs,] # Because beta-specific DEGs

# Refining dataframe
alpha <- alpha[-c(2,4:7)]
colnames(alpha) <- c("Genes","Alpha")
beta <- beta[-c(2,4:7)]
colnames(beta) <- c("Genes","Beta")

# Combining
df <- merge(alpha,beta,by="Genes")
df <- melt(df, id = c("Genes"))
colnames(df) <- c("Genes","Cell_Type","Avg_LFC")

# Finalizing
df.beta <- df
df.beta$Type <- "Beta-specific DEGs"

#####
# Getting LFC of DEGs in Alpha and Beta – COMMON DEGs – UPREGULATED
#####

# Extracting the DEGs
alpha <- alpha.all.10.percent[alpha.all.10.percent$X %in% common.DEGs,] # Because common DEGs
beta <- beta.all.10.percent[beta.all.10.percent$X %in% common.DEGs,] # Because common DEGs

# Refining dataframe
alpha <- alpha[-c(2,4:7)]
colnames(alpha) <- c("Genes","Alpha")
beta <- beta[-c(2,4:7)]
colnames(beta) <- c("Genes","Beta")

# Combining
df <- merge(alpha,beta,by="Genes")
df <- melt(df, id = c("Genes"))
colnames(df) <- c("Genes","Cell_Type","Avg_LFC")

# Finalizing
df.common <- df
df.common$Type <- "Common DEGs"
df <- rbind(df.alpha, df.beta, df.common)

#####
```

```

# Plotting Dot-and-Boxplot
#####

# Wilcoxon Test
stat.test <- df %>%
  group_by(Type) %>%
  wilcox_test(Avg_LFC ~ Cell_Type)
stat.test <- stat.test %>% add_xy_position(x="Type")
stat.test <- stat.test %>% mutate(asterisks =
  case_when(p < 0.001 ~ "***",
            p >= 0.001 & p < 0.01 ~ "**",
            p >= 0.01 & p < 0.05 ~ "*",
            p >= 0.05 ~ "ns"))

# Box plots with p-values
p <- ggboxplot(
  df, x = "Type", y = "Avg_LFC", color = "Cell_Type",
  palette = c("goldenrod", "mediumpurple2"), add = "jitter", size=0.5,
) + ylab("Average LFC upon ER Stress")

bxp <- p +
  stat_pvalue_manual(
    stat.test, label = "asterisks", tip.length = 0.01,
    hide.ns = FALSE
  ) + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

# View
ggarrange(bxp, nrow=1, ncol=2)

```

