

Figure-4H.R

sokole

2024-07-27

```
# This Script Generates Figure 4H
# Script By: Eishani Kumar Sokolowski
```

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

```
# Loading libraries
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2     3.5.1      ✓ tibble     3.2.1
## ✓ lubridate  1.9.3      ✓ tidyr      1.3.1
## ✓ purrr      1.0.2
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(pheatmap)
library(ggpubr)
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##   hour, isoweek, mday, minute, month, quarter, second, wday, week,
##   yday, year
##
## The following objects are masked from 'package:dplyr':
##
##   between, first, last
##
## The following object is masked from 'package:purrr':
##
##   transpose
```

```
library(dplyr)
library(ggpubr)
library(rstatix)
```

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

```

genes.of.interest <- c("AOPEP")

#####
# Loading file
#####

# Loading Siddhi's file
Siddhi.Counts <- read.csv("./Siddhi_all_cell-types_pseudobulk_all_counts_normalized.csv")
Siddhi.Counts <- as.data.frame(t(Siddhi.Counts))

# Making dataframe matrix boxplot friendly
colnames(Siddhi.Counts) <- Siddhi.Counts[1,]
Siddhi.Counts <- Siddhi.Counts[-1, ]

# Checking what cell-types are in Siddhi's dataframe
Cell.Types <- as.data.frame(rownames(Siddhi.Counts))
colnames(Cell.Types) <- "Cell_Types"
Cell.Types$Cell_Types <- gsub("^[^_]*_", "\\1", Cell.Types$Cell_Types)
Cell.Types$Cell_Types <- gsub("*ND", "\\1", Cell.Types$Cell_Types)
Cell.Types$Cell_Types <- gsub("*PD", "\\1", Cell.Types$Cell_Types)
Cell.Types$Cell_Types <- gsub("*T2D", "\\1", Cell.Types$Cell_Types)
Cell.Types <- unique(Cell.Types)

#####
# Alpha
#####

# Extracting Alpha cell-type: ND and T2D donors
Siddhi.Alpha <- Siddhi.Counts[rownames(Siddhi.Counts) %like% "Alpha", ] # Alpha
Siddhi.Alpha <- Siddhi.Alpha[!rownames(Siddhi.Alpha) %like% "Proliferating", ] # Removing Proliferating Alpha
Siddhi.Alpha <- Siddhi.Alpha[!rownames(Siddhi.Alpha) %like% "PD", ] # Removing Pre-Diabetics

# Creating an empty dataframe to store the output of for loop
Alpha.genes.df <- data.frame(matrix(ncol = 5, nrow = 0))
colnames(Alpha.genes.df) <- c("Expression", "Gene", "Donor", "Disease_Type", "Cell_Type")

# For loop to create boxplot-friendly dataframe
for (gene in genes.of.interest)
{
  Siddhi.Alpha.Gene <- Siddhi.Alpha %>% select(gene)
  Siddhi.Alpha.Gene$Gene <- colnames(Siddhi.Alpha.Gene)
  Siddhi.Alpha.Gene$Donor <- rownames(Siddhi.Alpha.Gene)
  Siddhi.Alpha.Gene$Donor <- gsub("_.*", "", Siddhi.Alpha.Gene$Donor)
  Siddhi.Alpha.Gene$Disease_Type <- rownames(Siddhi.Alpha.Gene)
  Siddhi.Alpha.Gene$Disease_Type <- gsub(".*ND", "ND", Siddhi.Alpha.Gene$Disease_Type)
  Siddhi.Alpha.Gene$Disease_Type <- gsub(".*T2D", "T2D", Siddhi.Alpha.Gene$Disease_Type)
  colnames(Siddhi.Alpha.Gene)[1] <- "Expression"
  Siddhi.Alpha.Gene$Cell_Type <- "Alpha"
  rownames(Siddhi.Alpha.Gene) <- NULL
}

```

```

Alpha.genes.df <- rbind(Alpha.genes.df,Siddhi.Alpha.Gene)
}

#####
# Beta
#####

# Extracting Beta cell-type: ND and T2D donors
Siddhi.Beta <- Siddhi.Counts[rownames(Siddhi.Counts) %like% "Beta", ] # Beta
Siddhi.Beta <- Siddhi.Beta[!rownames(Siddhi.Beta) %like% "PD", ] # Removing Pre-Diabetic
s

# Creating an empty dataframe to store the output of for loop
Beta.genes.df <- data.frame(matrix(ncol = 5, nrow = 0))
colnames(Beta.genes.df) <- c("Expression","Gene","Donor","Disease_Type","Cell_Type")

# For loop to create boxplot-friendly dataframe
for (gene in genes.of.interest)
{
  Siddhi.Beta.Gene <- Siddhi.Beta %>% select(gene)
  Siddhi.Beta.Gene$Gene <- colnames(Siddhi.Beta.Gene)
  Siddhi.Beta.Gene$Donor <- rownames(Siddhi.Beta.Gene)
  Siddhi.Beta.Gene$Donor <- gsub("_.*","",Siddhi.Beta.Gene$Donor)
  Siddhi.Beta.Gene$Disease_Type <- rownames(Siddhi.Beta.Gene)
  Siddhi.Beta.Gene$Disease_Type <- gsub(".*ND","ND",Siddhi.Beta.Gene$Disease_Type)
  Siddhi.Beta.Gene$Disease_Type <- gsub(".*T2D","T2D",Siddhi.Beta.Gene$Disease_Type)
  colnames(Siddhi.Beta.Gene)[1] <- "Expression"
  Siddhi.Beta.Gene$Cell_Type <- "Beta"
  rownames(Siddhi.Beta.Gene) <- NULL
  Beta.genes.df <- rbind(Beta.genes.df,Siddhi.Beta.Gene)
}

#####
# Combining into a dataframe and making dot-and-boxplot
#####

# Making a dataframe
final.df <- rbind(Alpha.genes.df,Beta.genes.df)
final.df$Expression <- as.numeric(final.df$Expression)

# Wilcox test
stat.test <- final.df %>%
  group_by(Gene, Cell_Type) %>%
  wilcox_test(Expression ~ Disease_Type, alternative = "two.sided")
stat.test <- stat.test %>% add_xy_position(x = "Cell_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"))
stat.test$y.position <- (0.05*stat.test$y.position)+stat.test$y.position

```

```
# Box plots with p-values
p <- ggboxplot(
  final.df, x = "Cell_Type", y = "Expression", color = "Disease_Type",
  palette = c("gold4","navy"), facet.by = "Gene", add = "jitter", size=0.5) +
  ylab("Normalized Expression") + ylim(4.8,8.6) + stat_pvalue_manual(stat.test, label =
"asterisks", tip.length = 0.01, hide.ns = FALSE)

# View
ggarrange(p, nrow=1, ncol=2)
```

