

Figure-5I.R

sokole

2024-07-28

```
# This Script Generates Figure 5I
# 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()
## i 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
```

```

# Specifying gene of interest
genes.of.interest <- c("MAP3K5")

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

# Loading Siddhi's file - Beta Cell Numbers
Siddhi.Beta.Numbers <- read.csv("./Siddhi_Supplementary_Table_3_Donors_Number_and_Percent_of_Beta_Cells.csv")
colnames(Siddhi.Beta.Numbers)[1] <- c("Donor")

# Loading Siddhi's file - Normalized Counts
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)

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

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

# 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(".*PD", "PD", 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 <- "B_Beta"
rownames(Siddhi.Beta.Gene) <- NULL
Beta.genes.df <- rbind(Beta.genes.df, Siddhi.Beta.Gene)
}

#####
# Adding Percent of Beta Cells Information
#####

# Changing "-" to "."
Siddhi.Beta.Numbers$Donor <- gsub("\\-", ".", Siddhi.Beta.Numbers$Donor)
Siddhi.Beta.Numbers$Donor <- gsub(" ", ".", Siddhi.Beta.Numbers$Donor)

# Adding information
Beta.genes.df <- merge(Beta.genes.df, Siddhi.Beta.Numbers, by="Donor")

#####
# Plotting the Scatter Plot
#####

# Making Expression Numeric
Beta.genes.df$Expression <- as.numeric(Beta.genes.df$Expression)

# Plotting
p <- ggscatter(Beta.genes.df, x = "Beta.Percent.Across.Endocrine.Cell.Types", y = "Expression", size=2,
               add = "reg.line")+theme_bw()+
  xlab("Beta / Endocrine Cells (%)") + stat_cor(method = "pearson") +
  ylab("MAP3K5 Normalized Expression") + ylim(0.8,4.1)

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

```

