

Figure-S3E.R

sokole

2024-07-22

```
# This Script Generates Figure S3E
# Script By: Eishani Kumar Sokolowski

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

# Loading 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(cowplot)
library(ggpubr)
```

```
##
## Attaching package: 'ggpubr'
```

```
## The following object is masked from 'package:cowplot':
##
##   get_legend
```

```
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ forcats   1.0.0   ✓ stringr   1.5.1
## ✓ lubridate 1.9.3   ✓ tibble    3.2.1
## ✓ purrr     1.0.2   ✓ tidyr     1.3.1
## ✓ readr     2.1.5
```

```
## — Conflicts ————— tidyverse_conflicts() —
## * dplyr::filter()    masks stats::filter()
## * dplyr::lag()       masks stats::lag()
## * lubridate::stamp() masks cowplot::stamp()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```
library(plyr)
```

```
## -----
## You have loaded plyr after dplyr – this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## -----
##
## Attaching package: 'plyr'
##
## The following object is masked from 'package:purrr':
##
##   compact
##
## The following object is masked from 'package:ggpubr':
##
##   mutate
##
## The following objects are masked from 'package:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##   discard
##
## The following object is masked from 'package:readr':
##
##   col_factor
```

```
#####
# Getting all HOMER results
#####

# Retrieving all the files at once
filelist <- list.files(pattern = ".txt")

# Assuming tab separated values without a header
datalist <- sapply(filelist, function(x) read.delim2(x), simplify = FALSE)

ERS <- datalist[["ERS.specific_closing_distal_DAPs_knownResults.txt"]]
ERS$p.adjust.value <- p.adjust(ERS$P.value, method = "BH")
ERS <- ERS[ERS$p.adjust.value < 0.10,]

INF <- datalist[["INF.specific_closing_distal_DAPs_knownResults.txt"]]
INF$p.adjust.value <- p.adjust(INF$P.value, method = "BH")
INF <- INF[INF$p.adjust.value < 0.10,]

Shared <- datalist[["ERS_INF_SHARED_Closing_Distal_CREs_FULL_COORDINATES_hg38_knownResults.txt"]]
Shared$p.adjust.value <- p.adjust(Shared$P.value, method = "BH")
Shared <- Shared[Shared$p.adjust.value < 0.10,]

ERS <- ERS[-c(2:ncol(ERS))]
INF <- INF[-c(2:ncol(INF))]
Shared <- Shared[-c(2:ncol(Shared))]

r <- rbind(ERS, INF, Shared)
r <- unique(r)

#####
# Compiling results
#####

for (i in 1:(length(datalist))) {

  # Calculating FDR values
  datalist[[i]]$p.adjust.value <- p.adjust(datalist[[i]]$P.value, method = "BH")
  datalist[[i]]$NegLog10P.Adj <- -log10(datalist[[i]]$p.adjust.value)

  # Rename columns
  colnames(datalist[[i]]) <- c("HOMER Motif Name", "Consensus", "P-value", "Log.P-value", "q-value (Benjamini)", "# Target Sequences with Motif", "% of Targets Sequences with Motif", "# Background Sequences with Motif", "% of Background Sequences with Motif", "FDR.p.adj", "Neg.Log.10.P.Adj")

  # Remove the percentage signs
  datalist[[i]]$"% of Targets Sequences with Motif" <- str_replace(datalist[[i]]$"% of Targets Sequences with Motif", "%", "")
  datalist[[i]]$"% of Background Sequences with Motif" <- str_replace(datalist[[i]]$"% of Background Sequences with Motif", "%", "")
}
```

```
# Find Fold Change
datalist[[i]]$Fold_Change <- as.numeric(datalist[[i]]$`% of Targets Sequences with Motif`)/as.numeric(datalist[[i]]$`% of Background Sequences with Motif`)

# Extracting Motif Name - Just gets the substring until brackets for motif name
datalist[[i]] <- datalist[[i]] %>%
  rowwise() %>%
  mutate(Motif = str_extract(`HOMER Motif Name`, "^(.+)")) %>%
  as.data.frame()

# Converting to gene name
datalist[[i]]$TF_Gene_Name <- toupper(datalist[[i]]$Motif)
datalist[[i]]$TF_Gene_Name <- gsub("\\.", "-", datalist[[i]]$TF_Gene_Name)

# Saving as one master dataframe
datalist[[i]]$DAP_Type <- names(datalist[i])
datalist[[i]]$DAP_Type <- gsub("_knownResults.txt", "", datalist[[i]]$DAP_Type)
final.df <- bind_rows(datalist)
}

# Finalizing dataframe
final.df$Motif <- toupper(final.df$Motif)
final.df$Category <- gsub("_closing_distal_DAPs", "", final.df$DAP_Type)
final.df$Category <- gsub("\\.", "-", final.df$Category)
final.df$Category <- gsub(".*SHARED.*", "Shared", final.df$Category)
final.df <- final.df[-c(1:9, 11, 14, 15)]

# Scaling the fold change
final.df <- dplyr::mutate(final.df, Scaled_FC = rescale(Fold_Change))

# Making asterisks column
final.df <- final.df %>% mutate(p.adj.asterisks =
  case_when(FDR.p.adj < 0.01 ~ "***",
            FDR.p.adj < 0.05 ~ "**",
            FDR.p.adj < 0.10 ~ "*",
            FDR.p.adj > 0.10 ~ "ns"))

#####
# Making heatmaps of motifs of interest
#####

# Making a vector
motifs.of.interest <- c("MAFA",
  "PDX1")

# Extracting
final.df <- final.df[final.df$Motif %in% motifs.of.interest,]

# Ordering by motif
final.df$Motif <- factor(final.df$Motif, levels=c("MAFA",
  "PDX1"))
```

```

# Ordering by treatment
final.df$Category <- factor(final.df$Category, levels=c("INF-specific",
                                                         "ERS-specific",
                                                         "Shared"))

# With asterisks (black color text)
p <- ggplot(final.df, aes(Motif, Category, fill=Scaled_FC)) +
  geom_tile(color = "black", lwd = 0.5, linetype = 1) + coord_flip() + theme_classic() +
  scale_fill_gradient2(low = "purple", mid = "black", high = "yellow", midpoint = 0.50,
breaks = c(0,1)) +
  geom_text(aes(label = p.adj.asterisks), color = "black", size = 4, angle=0) + theme(ax
is.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

# With asterisks (white color text)
q <- ggplot(final.df, aes(Motif, Category, fill=Scaled_FC)) +
  geom_tile(color = "black", lwd = 0.5, linetype = 1) + coord_flip() + theme_classic() +
  scale_fill_gradient2(low = "purple", mid = "black", high = "yellow", midpoint = 0.50,
breaks = c(0,1)) +
  geom_text(aes(label = p.adj.asterisks), color = "white", size = 4, angle=0) + theme(ax
is.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

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

```

