DEPI Day 3 Epistasis

Abigail Seeger

January 8, 2021

Contents

Load Necessary Packages

```
library(stringr)
library(stringi)
library(dplyr)
library(viridis)
library(ggplot2)
library(extrafont)
library(ggthemes)
library(lemon)
```

Load in Cleaned Data

```
depi_data <- read.table("C:/Users/Owner/Documents/Research/Shiu_Lab/Shiu_Lab_R/Data/Clean_DEPI_Data.csv
    sep = ",", header = TRUE)
```

Functions

```
add_number <- function(data_frame) {
    ### First, if the genotype is Col0 (only
    ### genotype with length 4), assign 0 as
    ### number Else, assign number as genotype
    ### with 'mpk' removed Example: mpk1 will
    ### be 1, mpk1-17 will be 1-17
    data_frame <- data_frame %>% mutate(number = ifelse(genotype !=
        "Col0", (stri_sub(genotype, 4, length(genotype))),
        0))
    ### Next, for all double mutants, replace
    ### '-' with '0' Example: 1-17 becomes 1017
    data_frame$number <- as.numeric(gsub("-",
        "0", data_frame$number))
    ### Almost there! There's a problem with
    ### two single digit double mutants We need
    ### a four digit number to sort correctly
    ### Example: mpk1-3 -> 1-3 -> 103, but we
    ### need it to be 1003 to sort correctly
    data_frame$number[data_frame$number ==
        "103"] <- "1003"
    data_frame$number[data_frame$number ==
        "506"] <- "5006"
    data_frame$number[data_frame$number ==
        "608"] <- "6008"
    data_frame$number[data_frame$number ==
        "609"] <- "6009"
    ### Convert number to a numberic in order
    ### to sort
    data_frame$number <- as.numeric(data_frame$number)
    data_frame <- data_frame %>% arrange(number)
    data_frame <- data_frame %>% mutate(number_2 = number)
    data_frame$number_2[nchar(data_frame$number_2) ==
        4] <- 0
    data_frame$number_2[nchar(data_frame$number_2) ==
        5] <- 0
    return(data_frame)
}
```

Note that this function uses the normalized value, instead of the measured value.

```
cell_371_data <- function(data_frame) {

    npq_phi2 <- data_frame %>% filter(measurement %in%
        c("npq", "phi2")) %>% group_by(time_point,
        measurement) %>% mutate_each(funs(./median(.[genotype ==
        "Col0"])), normalized_value) %>%
        group_by(time_point, measurement,
            genotype) %>% mutate(log2_fold = log2(median(normalized_value)))
```

```r
    start_end <- unique((data_frame %>% group_by(day) %>%
        filter(time_point %in% c(min(time_point),
            max(time_point))))$time_point)

    leaf_area <- data_frame %>% filter(measurement ==
        "leafarea") %>% filter(time_point %in%
        start_end) %>% group_by(time_point,
        measurement) %>% mutate_each(funs(./median(.[genotype ==
        "Col0"])), measured_value) %>% group_by(time_point,
        measurement, genotype) %>% mutate(log2_fold = log2(median(measured_value)))

    out <- rbind(npq_phi2, leaf_area) %>%
        group_by(genotype, time_point, measurement)

    return(as.data.frame(out))
}
```

Selection Coefficient Calculations

```r
selectionCoef <- data.frame(genotype = rep(NA,
    0), SelectionCoefficient = rep(NA, 0),
    Experiment = rep(NA, 0), Measurement = rep(NA,
        0))

for (e in c("Dec", "Jan", "Feb")) {
    for (m in c("phi2", "leafarea", "phi2")) {
        temp_data <- depi_data %>% filter(month ==
            e, measurement == m)
        temp_nrow <- 38 * length(unique(temp_data$time_point))
        for (i in unique(temp_data$time_point)) ### Create an empty data frame to fill with
        ### the information and calcuations:
        selectionCoefTmp <- data.frame(genotype = rep(NA,
            temp_nrow), SelectionCoefficient = rep(NA,
            temp_nrow), Experiment = rep(NA,
            temp_nrow), Measurement = rep(NA,
            temp_nrow), Time_Point = rep(NA,
            temp_nrow))
        count <- 1
        for (g in unique(temp_data$genotype)) {
            fm <- mean(filter(temp_data,
                genotype == g, month == e,
                measurement == m, time_point ==
                  i)$normalized_value)
            fwt <- mean(filter(temp_data,
                genotype == "Col0", month ==
                  e, measurement == m, time_point ==
                  i)$normalized_value)
            TempSelectionCoef <- (fm - fwt)/fwt
            selectionCoefTmp[count, 1] <- g
            selectionCoefTmp[count, 2] <- TempSelectionCoef
            selectionCoefTmp[count, 3] <- e
            selectionCoefTmp[count, 4] <- m
            selectionCoefTmp[count, 5] <- i
            count <- count + 1
        }
        selectionCoef <- rbind(selectionCoef,
            selectionCoefTmp)
    }
}

selectionCoef <- selectionCoef %>% arrange(Measurement,
    genotype)
```

Epistasis Calculations

```
all_double_mutants = list()
for (gen in unique(depi_data$genotype)) {
    if (str_detect(gen, "-") == T) {
        all_double_mutants = c(all_double_mutants,
            gen)
    }
}

### Epistasis Calculations: Initialize an
### empty data frame to populate with
### information:
geneticInteractions <- data.frame(genotype = rep(NA,
    0), MutantA = rep(NA, 0), MutantB = rep(NA,
    0), AdditiveEpistasis = rep(NA, 0), ProportionalEpistatis = rep(NA,
    0), Experiment = rep(NA, 0), Measurement = rep(NA,
    0), Time_Point = rep(NA, 0))

### Loop through each experiment and
### measurement:
for (e in c("Dec", "Jan", "Feb")) {
    for (m in c("phi2", "leafarea", "npq")) {
        temp_data <- depi_data %>% filter(month ==
            e, measurement == m)
        temp_nrow <- 25 * length(unique(temp_data$time_point))
        for (i in unique(temp_data$time_point)) {
            ### Filter to each specific experiment and
            ### measurement
            tempData <- filter(depi_data,
                month == e, measurement ==
                  m, time_point == i)
            ### Create an empty data frame to fill with
            ### the information and calcuations:
            geneticInteractionsTmp <- data.frame(genotype = rep(NA,
                temp_nrow), MutantA = rep(NA,
                temp_nrow), MutantB = rep(NA,
                temp_nrow), AdditiveEpistasis = rep(NA,
                temp_nrow), ProportionalEpistatis = rep(NA,
                temp_nrow), Experiment = rep(NA,
                temp_nrow), Measurement = rep(NA,
                temp_nrow), Time_Point = rep(NA,
                temp_nrow))
            ### Initialize a row count to use to
            ### populate the data frame
            rowCount <- 1
            ### For each of the double mutants:
            for (dm in unlist(all_double_mutants)) {
                ### Extract the single mutants from the
                ### double mutant
                ma <- unlist(strsplit(dm,
                  "-"))[1]
                mb <- paste("mpk", unlist(strsplit(dm,
                  "-"))[2], sep = "")
                ### Calculate the fitness of the dm, ma,
                ### mb, and wt
                fdm <- mean(filter(tempData,
                  genotype == dm)$normalized_value)
                fwt <- mean(filter(tempData,
                  genotype == "Col0")$normalized_value)
                fma <- mean(filter(tempData,
                  genotype == ma)$normalized_value)
                fmb <- mean(filter(tempData,
                  genotype == mb)$normalized_value)
                ### Calculate Additive and Proportional
                ### Epistasis
                AddEp <- fdm + fwt - (fma +
                  fmb)
                PropEp <- log((fdm * fwt)/(fma *
                  fmb))
                ### Populate the data frame with this
                ### information:
                geneticInteractionsTmp[rowCount,
                  1] <- dm
                geneticInteractionsTmp[rowCount,
                  2] <- ma
                geneticInteractionsTmp[rowCount,
                  3] <- mb
                geneticInteractionsTmp[rowCount,
                  4] <- AddEp
                geneticInteractionsTmp[rowCount,
                  5] <- PropEp
                geneticInteractionsTmp[rowCount,
                  6] <- e
                geneticInteractionsTmp[rowCount,
                  7] <- m
                geneticInteractionsTmp[rowCount,
                  8] <- i
                rowCount <- rowCount + 1
            }
            ### Add the rows of the temporary genetic
            ### interaction information to the main
```

```r
            ### data frame
            geneticInteractions <- rbind(geneticInteractions,
                geneticInteractionsTmp)
        }
    }
}
```

Selection Coefficient Visualizations

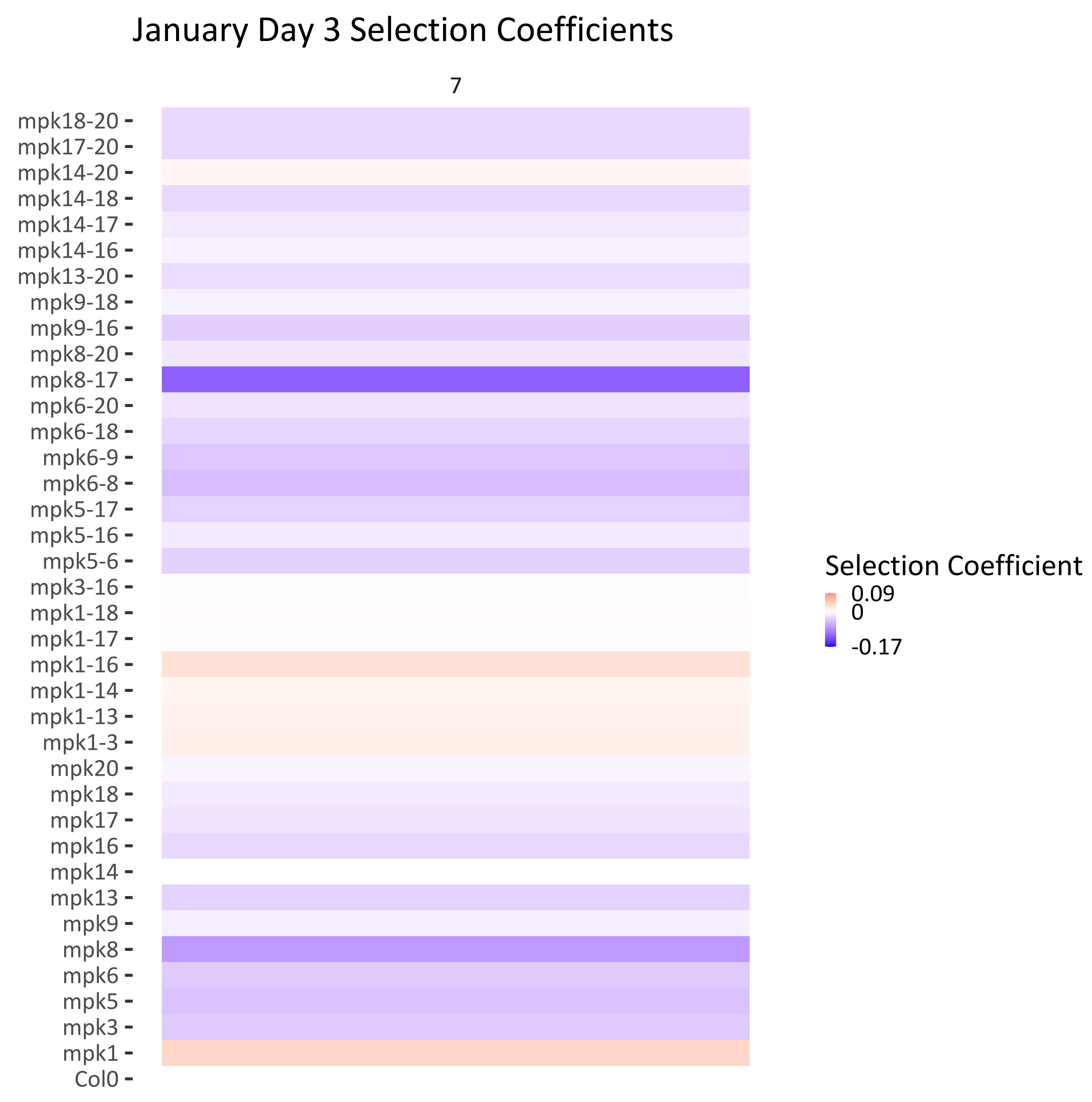
First, include a column of "number" and "number_2"

```r
selectionCoef <- add_number(selectionCoef)
selectionCoef$genotype <- reorder(selectionCoef$genotype,
    desc(selectionCoef$number))
selectionCoef <- add_day_col(selectionCoef)
```

```r
temp_plot <- selectionCoef %>% filter(Experiment ==
    "Jan", Measurement == "phi2")

temp_upper_bound <- round(max(temp_plot$SelectionCoefficient) +
    0.05, 2)
temp_lower_bound <- round(min(temp_plot$SelectionCoefficient) -
    0.05, 2)


ggplot(data = temp_plot, aes(x = Time_Point,
    y = genotype, fill = SelectionCoefficient)) +
    labs(fill = "Selection Coefficient",
        x = "Hours", y = NULL, title = "January Day 3 Selection Coefficients") +
    geom_tile(width = 10, height = 10) +
    facet_grid(genotype ~ day, scales = "free",
        switch = "y") + theme_tufte(base_family = "Calibri",
    base_size = 50) + theme(strip.background.y = element_blank(),
    strip.text.y = element_blank(), axis.title.x = element_blank(),
    axis.text.x = element_blank(), axis.ticks.x = element_blank(),
    panel.spacing = unit(0, "lines")) + scale_fill_gradient2(low = "blue",
    high = "red", mid = "white", midpoint = 0,
    limits = c(temp_lower_bound, temp_upper_bound),
    breaks = c(temp_lower_bound, 0, temp_upper_bound),
    labels = c(as.character(temp_lower_bound),
        "0", as.character(temp_upper_bound)))
```



Genetic Interactions Visualizations

```
geneticInteractions <- add_number(geneticInteractions)
geneticInteractions$genotype <- reorder(geneticInteractions$genotype,
    desc(geneticInteractions$number))
```

```
temp_plot <- geneticInteractions %>% filter(month ==
    "Jan", measurement == "phi2")
ggplot(data = cell_371_phi2_jan, aes(x = time_point,
    y = genotype, fill = log2_fold)) + labs(fill = "Log 2 Fold Change",
    x = "Hours", y = NULL, title = "January: Phi2 Log 2 Fold Change") +
```

```r
    geom_tile(width = 10, height = 10) +
    facet_grid(genotype ~ day, scales = "free",
        switch = "y") + theme_tufte(base_family = "Calibri",
    base_size = 50) + theme(strip.background.y = element_blank(),
    strip.text.y = element_blank(), axis.title.x = element_blank(),
    axis.text.x = element_blank(), axis.ticks.x = element_blank(),
    panel.spacing = unit(0, "lines")) + scale_fill_gradient2(low = "blue",
    high = "red", mid = "white", midpoint = 0,
    limits = c(, 0.8), breaks = c(-0.7, 0,
        0.8), labels = c("-0.7", "0", "0.8"))
```