BINF 7960 Project

Lucy Quirk

2024-03-20

Growth Rate and µ/µMAX

This script will utilize skills I learned in R, bash, and git during the BINF 7960 spring course. Skills learned in this course will help me visualize physiological data from my thesis, and create a clear script available on my github account. I will apply the use of creating and mutating data frames with dplyr to make a figure in ggplot.

Set up script and read in data

First, I will call in an R script which contains some useful functions I can use in this document. By writing `source(functions.R)`, all functions written in the file will be accessible to me. In the Carpentries course, we learned how to name and create different data structures (*lists*, *vectors*, matrices and *data frames*), and call these objects, calling the named function `checkAndLoadPackages` follows the same logic. First, I create a vector called `pkgs` and list each package I need for the script. Calling the vector `pkgs` within the function `checkAndLoadPackages()` will look at each package in the vector, first checking it is installed, install it if needed, then load all packages to my working directory.

Next, I will create two objects called "path_" and "path_out" with the location to reading data in and save output

```
source('./scripts/functions.R')
pkgs <- c('dplyr','stringr','ggplot2','rstatix','shadowtext','ggpubr')
checkAndLoadPackages(pkgs)
```

```
Bellow Packages Successfully Loaded:

     dplyr    stringr    ggplot2    rstatix shadowtext     ggpubr
      TRUE       TRUE       TRUE       TRUE       TRUE       TRUE
```

```
path_in <- c('./rawData/')
path_out <- c('./output')
physioData<-read.csv(paste(path_in, 'physio_exp.csv', sep = ''))
```

Clean up data

Before I can plot my data, I need to make some changes to the data frame. I use functions from the package **stringr** to replace the code for each organism with it's proper name. Next, I add a column of data for a missing treatment with the function `rbind`, we learned in class the rows of a data frame are lists and lists can be added to a data frame with the function `rbind`. Then, I add two columns with the shelf zone the organism was isolated and it's taxanomic group using a loop.

```
physioData$culture <- str_replace_all(physioData$culture, c('4' = 'C. closterium UGA4','8' = 'C. closte

#add an empty row for missing treatment in uga4
empty.04 <- list("C. closterium UGA4","High Fe", "04pfe19",0, 0, 0, 0, 0, 0, 0, 0, 0)
physioData<- rbind(physioData,empty.04)

for (i in 1:nrow(physioData)) {
  if (physioData$culture[i] == "G. oceanica"|physioData$culture[i] == "C. closterium UGA8"){
    physioData$Shelf[i] = "Inner shelf"}
  else{physioData$Shelf[i] = "Outer shelf"}
}
for (i in 1:nrow(physioData)) {
  if (physioData$culture[i] == "G. oceanica"|physioData$culture[i] == "G. huxleyi"){
    physioData$taxa[i] = "Coccolithophore"}
  else{physioData$taxa[i] = "Diatom"}
}
```

Pull out growth rates

For this project, I will create a figure of my growth rate data. The data file I read in includes multiple parameters so I will use **dplyr** pipes and `select` to extract the growth rate data.

```
colnames(physioData)
```

```
 [1] "culture"     "treatment"   "Sample"      "GrowthRate"  "chla.ug.L"
 [6] "CellSize"    "tQa"         "Fv.Fm"       "pH.starting" "pH.ending"
[11] "pH.change"   "cells.L"     "Shelf"       "taxa"
```

```
growth <- physioData %>%
  select(c(culture, treatment, GrowthRate, taxa, Shelf,Sample)) %>%
  drop_na(c(GrowthRate))

growth$treatment <- factor(growth$treatment, levels = c("High Fe","Low Fe"))
head(growth,3)
```

```
             culture treatment GrowthRate   taxa       Shelf     Sample
1 C. closterium UGA4    Low Fe     0.6601 Diatom Outer shelf 04pFe21.9a
2 C. closterium UGA4    Low Fe     0.5967 Diatom Outer shelf 04pFe21.9b
3 C. closterium UGA4    Low Fe     0.5830 Diatom Outer shelf 04pFe21.9c
```

Run statistical tests

When plotting data, it is helpful to show if changes observed were statistically different. To do this, I will create a function called `high.vs.low()` to see if the growth rate between high and low iron treatments was significant. This function will perform a few jobs for me:

1. Test for normality within iron treatments with a Wilks-shapiro test

2. Test for equal variance between iron treatments with an f-test
3. Run the appropriate test based on the results (wilcoxon rank test, t-test with unequal variance, or t-test with equal variance). The function will need the data frame, organism, variable (or parameter) which I am interested in testing, and shelf location of isolation. The output of the function is a data frame with the significance information.

```r
high.vs.low <- function(df, organism, var, location){
  df <- filter(df,culture == eval(quote(organism)))
  df <- data.frame(treatment = df$treatment, param = df[[var]], culture = df$culture)
  df$treatment <- factor(df$treatment)
  # separate df by organism and iron treatment
  # test the normality and variance WITHIN each treatment
  high.fe <- filter(df, grepl("High Fe",treatment) == TRUE)
  low.fe <- filter(df,grepl("Low Fe",treatment) == TRUE)
## 1.
  if (nrow(low.fe) >= 3){ #check there are enough rows for normality test
    normal.h <-shapiro_test(high.fe$param)
    normal.l <- shapiro_test(low.fe$param)
    if (normal.h$p.value < 0.05 || normal.l$p.value < 0.05){
      stat.test <- wilcox_test(df,param ~ treatment) %>%
        adjust_pvalue(method = 'fdr') %>%
        add_significance('p.adj')
      stat.test
## 2.
    }else{
      f.test <- var.test(param ~ treatment,df)
## 3.
      if(f.test$p.value > 0.05){
        stat.test <- t_test(df, param ~ treatment) %>%
          adjust_pvalue(method = 'fdr') %>%
          add_significance('p.adj')
        stat.test
      }else{
        stat.test <- t_test(df, param ~ treatment, var.equal = FALSE) %>%
          adjust_pvalue(method = 'fdr') %>%
          add_significance('p.adj')
        stat.test
      }
    }
  }
  stat.test <- stat.test %>% mutate('culture'=organism, 'shelf'=location,
                                    '.y.'='mu')
}
```

I run the function on each organism, then I create a data frame with the significance output from the function, printing it using the head() function we learned in class.

```r
mu.08 <- high.vs.low(growth,"C. closterium UGA8",'GrowthRate', 'Inner Shelf')
mu.06 <- high.vs.low(growth,"G. oceanica",'GrowthRate', 'Inner Shelf')
mu.13 <- high.vs.low(growth,"G. huxleyi",'GrowthRate', 'Outer Shelf')

mu.signif <- bind_rows(mu.13, mu.06, mu.08)
head(mu.signif,3)
```

```
# A tibble: 3 x 12
  .y.   group1  group2    n1    n2 statistic    df        p   p.adj p.adj.signif
  <chr> <chr>   <chr>  <int> <int>     <dbl> <dbl>    <dbl>   <dbl> <chr>
1 mu    High Fe Low Fe     3     6      8.59  5.78 0.000167 1.67e-4 ***
2 mu    High Fe Low Fe     3     6     13.2   2.88 0.00115  1.15e-3 **
3 mu    High Fe Low Fe     3     3      3.61  2.82 0.0404   4.04e-2 *
# i 2 more variables: culture <chr>, shelf <chr>
```

Plotting growth rate data

The figure I will make with show the mean and standard deviation of the growth rate under high and low iron treatments for each organism. We learned in class that rather than tediously calculating the mean and SD for each treatment and organism, `dplyr` can quickly do this for us.

First, I use the `group_by` function to group my growth rate data by organism and treatment. Then, I use the `summarise` function to calculate the mean and SD. I combine this data with the significance calculated above so I can include it in my figure.

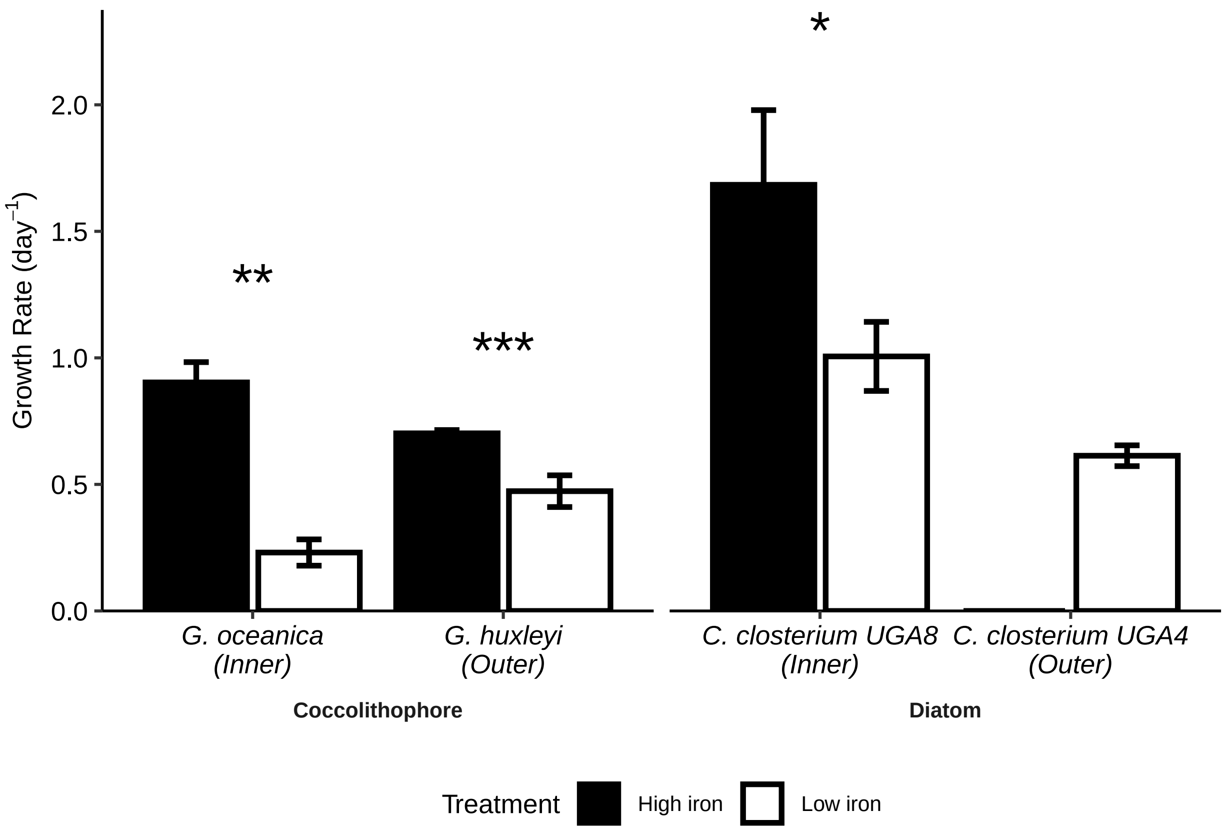
```
head(avg.mu,3)
```

```
# A tibble: 3 x 9
# Groups:   culture, treatment, taxa [3]
  culture          treatment taxa  Shelf    mu   mu.sd   p.adj p.adj.signif shelf
  <chr>            <chr>     <chr> <chr> <dbl>   <dbl>   <dbl> <chr>        <chr>
1 C. closterium ~ High Fe   Diat~ Oute~ 0     NA      NA      <NA>         <NA>
2 C. closterium ~ Low Fe    Diat~ Oute~ 0.613  0.0411 NA      <NA>         <NA>
3 C. closterium ~ High Fe   Diat~ Inne~ 1.68   0.295   0.0404 *            Inne~
```

Now the data is ready to plot. I will create a bar plot with error bars showing the standard deviation for the growth rate data using ggplot. In class, we learned how to make ggplots using aes() to select the x and y axis, and using geom_ to select the type of figure we wish to make.

```
mu.plot <-{
  avg.mu$culture <- factor(avg.mu$culture, levels=c('G. oceanica','G. huxleyi','C. closterium UGA8','C.
  brp <- ggplot(avg.mu, aes(culture, mu, fill=treatment))+
    geom_bar(linewidth=1, color='black',stat='identity', position =position_dodge2(0.8))+
    geom_errorbar(aes(ymin=mu-mu.sd, ymax=mu+mu.sd),position = position_dodge(width=0.9), width=0.2, li
    geom_text(aes(label=p.adj.signif, y=mu+mu.sd,vjust=-1.6),size=c(20/.pt),show.legend=FALSE) +
    facet_grid(~taxa, scales="free_x", space="free_x", switch="x")+
    labs(y=expression(paste('Growth Rate (',day^-1,')')),fill='Treatment')+
    theme_pubr()+
    theme(text = element_text(size=10),axis.title.x=element_blank(), axis.text.x = element_text(face="i
    scale_fill_manual(labels=c('High iron','Low iron'),values=c("black","white")) +
    scale_x_discrete(labels=c(
      "C. closterium UGA8" = "C. closterium UGA8\n(Inner)",
      "G. oceanica" = "G. oceanica\n(Inner)",
      "C. closterium UGA4" = "C. closterium UGA4\n(Outer)",
      "G. huxleyi" = "G. huxleyi\n(Outer)")) +
    scale_y_continuous(expand=expansion(mul=c(0,0.2)))
  print(brp)
}
```

```
Warning: Removed 5 rows containing missing values or values outside the scale range
(‘geom_text()‘).
```



Updating my github account

One of the last applications of this class to my research is the use of git. I have a github repository called Thesis in the current working directory. I will add this project to my github repository under the scripts folder using a bash coding chunk.

```bash
git add scripts/BINF7680.Rmd
git commit -m 'Adding bioinformatics class project'
git push origin main
```

```
## [main 4b95a38] Adding bioinformatics class project
##  1 file changed, 1 deletion(-)
## To github.com:Lucy-Quirk/Thesis.git
##    74f16ad..4b95a38  main -> main
```

The file uploaded to my github repository can be accessed with the link below:
https://github.com/Lucy-Quirk/Thesis/blob/main/scripts/BINF7680.Rmd