metal stats per CELL

Hannah Reich

Aug 9, 2020

script updated 08/09/2020. per cell normalizations

```
#  per cell normalized metal data
# July 2019 for coauthors
# set the working directory
setwd("~/Desktop/PhD/chapters/chapter 4 - temp treatments/for_coauthors_v1/data_code/code/")
# load the packages
library(ggplot2)
library(cowplot)
```

```
##
## ********************************************************
```

```
## Note: As of version 1.0.0, cowplot does not change the
```

```
##   default ggplot2 theme anymore. To recover the previous
```

```
##   behavior, execute:
##   theme_set(theme_cowplot())
```

```
## ********************************************************
```

```
library(readxl)
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(reshape2)
library(ggpubr)
```

```
## Loading required package: magrittr
```

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

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

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:magrittr':
##
##     extract
```

```
## The following object is masked from 'package:reshape2':
##
##     smiths
```

```r
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(outliers)
library(tidyverse)
```

```
## ── Attaching packages ─────────
```

```
## ✓ tibble  2.1.3     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.4.0
## ✓ purrr   0.3.3
## ── Conflicts ──────────────────
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
## x Hmisc::src()       masks dplyr::src()
## x Hmisc::summarize() masks dplyr::summarize()
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
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 objects are masked from 'package:Hmisc':
##
##     is.discrete, summarize
```

```
## 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(RVAideMemoire)  # load for the se() function
```

```
## *** Package RVAideMemoire v 0.9-74 ***
```

```
##
## Attaching package: 'RVAideMemoire'
```

```
## The following object is masked from 'package:magrittr':
##
##     mod
```

```
# read in the metal data (concentration in nM but blank adjusted)
nm <- read_excel("~/Desktop/PhD/chapters/chapter 4 - temp treatments/for_coauthors_v1/data_code/metallome_um3.xls
x",
                 sheet = "nMconc_RAW")
nm <- as.data.frame(nm)
# read in the volume used from the celldensity excel data
vol <- read_excel("~/Desktop/PhD/chapters/chapter 4 - temp treatments/for_coauthors_v1/data_code/cell_density.xls
x",
                  sheet = "vol")
vol <- as.data.frame(vol)
# merge the vol used and TM data with one another
all <- merge(nm, vol, by = "sampleID")
# create a column for the total volume of cell material used (this figure's unit will be nM/um3)
all$tmvol <- (all$mLTM*all$CellDensity)
# create vectors for all of the nutrients that normalize the data by # of cells used (except V, Mo, and Cr)
all$Pcell <- ((all$Phosphorus/all$tmvol)*1000)
all$Manganesecell <- ((all$Manganese/all$tmvol)*1000)
all$Ironcell <- ((all$Iron/all$tmvol)*1000)
all$Cobaltcell <- ((all$Cobalt/all$tmvol)*1000)
all$Nickelcell <- ((all$Nickel/all$tmvol)*1000)
all$Coppercell <- ((all$Copper/all$tmvol)*1000)
all$Zinccell <- ((all$Zinc/all$tmvol)*1000)
# write out data table and do outlier detection in excel
#write.csv(all, "~/Desktop/PhD/chapters/chapter 4 - temp treatments/for_coauthors_v1/data_code/nutrientspercell_r
aw.csv", quote = FALSE, row.names = TRUE, col.names = TRUE)
# data checked via excel and the only values that were removed from the raw data were two negative zinc values in
psyg-100-30-2a
```

part 2: univariate figs and stats

```
# metal figures and analysis
# first the data were put through the outlier removal flow through (chunk 1) and were manually checked to ensure
 outliers removed were statistical outliers and not biological variability
# load more packages
library(MASS)
```

```
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(PMCMR)
```

```
## PMCMR is superseded by PMCMRplus and will be no longer maintained. You may wish to install PMCMRplus instead.
```

```
library(data.table) # for function `fread`
library(broom)      # for function `tidy`
```

```
##
## Attaching package: 'broom'
```

```
## The following object is masked from 'package:RVAideMemoire':
##
##     bootstrap
```

```
library(readxl)
# load the data
dat_all <- read_csv("~/Desktop/PhD/chapters/chapter 4 - temp treatments/for_coauthors_v1/data_code/nutrientsperce
ll_raw.csv")
```

```
## Parsed with column specification:
## cols(
##   sampleID = col_character(),
##   Species = col_character(),
##   Ironconc = col_double(),
##   Temp = col_character(),
##   treatmentID = col_character(),
##   Phosphorus = col_double(),
##   Manganese = col_double(),
##   Iron = col_double(),
##   Cobalt = col_double(),
##   Nickel = col_double(),
##   Copper = col_double(),
##   Zinc = col_double()
## )
```

```
dat_all <- as.data.frame(dat_all)
# check the structure, make sure all of the metals are numeric
str(dat_all)
```

```
## 'data.frame':    96 obs. of  12 variables:
##  $ sampleID   : chr  "min-10-26-1" "min-10-26-1" "min-10-26-2" "min-10-26-2" ...
##  $ Species    : chr  "min" "min" "min" "min" ...
##  $ Ironconc   : num  10 10 10 10 10 10 10 10 10 10 ...
##  $ Temp       : chr  "t26" "t26" "t26" "t26" ...
##  $ treatmentID: chr  "min-10-26" "min-10-26" "min-10-26" "min-10-26" ...
##  $ Phosphorus : num  1.48 1.544 1.156 1.325 0.814 ...
##  $ Manganese  : num  0.001837 0.001935 0.001177 0.001363 0.000985 ...
##  $ Iron       : num  0.00388 0.00385 0.00333 0.00352 0.00303 ...
##  $ Cobalt     : num  0.000235 0.000241 0.000129 0.000149 0.000116 ...
##  $ Nickel     : num  4.67e-05 4.27e-05 4.08e-05 5.04e-05 3.13e-05 3.29e-05 2.61e-05 3.11e-05 2.97e-05 4.18e-05
...
##  $ Copper     : num  6.47e-05 9.50e-05 6.22e-05 1.17e-04 4.58e-05 ...
##  $ Zinc       : num  0.000357 0.000398 0.000392 0.000792 0.000127 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   sampleID = col_character(),
##   ..   Species = col_character(),
##   ..   Ironconc = col_double(),
##   ..   Temp = col_character(),
##   ..   treatmentID = col_character(),
##   ..   Phosphorus = col_double(),
##   ..   Manganese = col_double(),
##   ..   Iron = col_double(),
##   ..   Cobalt = col_double(),
##   ..   Nickel = col_double(),
##   ..   Copper = col_double(),
##   ..   Zinc = col_double()
##   .. )
```

```
# make ironconc, temp, and treatmentID factors
dat_all$treatmentID <- as.factor(dat_all$treatmentID)
dat_all$Species <- as.factor(dat_all$Species)
#######################
# get average metal content for each metal by species
# creating a function to automate psyg/min comparisons for kw and average
# useful websites: https://www.guru99.com/r-apply-sapply-tapply.html; http://rstudio-pubs-static.s3.amazonaws.com/1204_67621a69f1dc465f81de9716ec063742.html

##################
######### do the stats by treatment
# make objects for min and psyg
m <- dat_all %>% filter(Species %in% c("min"))
p <- dat_all %>% filter(Species %in% c("psyg"))
# make groups for the posthoc test
groupm <- cbind(m$Ironconc, m$Temp)
##### Kruskal-wallis tests by element for each species, shows comparisons
# shapiro tests first (no ma will ever be normal though)
# wow, some of the ma is actually normal. but since some metals aren't non-parametric stats anyways to treat everybody the same
# min first
shapiro.test(m$Cobalt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m$Cobalt
## W = 0.81848, p-value = 3.496e-06
```

```
shapiro.test(m$Copper)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m$Copper
## W = 0.44198, p-value = 2.927e-12
```

```
shapiro.test(m$Iron)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m$Iron
## W = 0.93268, p-value = 0.008568
```

```
shapiro.test(m$Manganese)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m$Manganese
## W = 0.8332, p-value = 8.018e-06
```

```
shapiro.test(m$Nickel)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m$Nickel
## W = 0.77968, p-value = 4.648e-07
```

```
shapiro.test(m$Zinc)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m$Zinc
## W = 0.19312, p-value = 8.754e-15
```

```
shapiro.test(m$Phosphorus)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  m$Phosphorus
## W = 0.95942, p-value = 0.09562
```

```
# now psyg
shapiro.test(p$Cobalt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p$Cobalt
## W = 0.95793, p-value = 0.08329
```

```
shapiro.test(p$Copper)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p$Copper
## W = 0.98229, p-value = 0.6764
```

```
shapiro.test(p$Iron)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p$Iron
## W = 0.95861, p-value = 0.08873
```

```
shapiro.test(p$Manganese)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p$Manganese
## W = 0.97864, p-value = 0.5235
```

```
shapiro.test(p$Nickel)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p$Nickel
## W = 0.78047, p-value = 4.831e-07
```

```
shapiro.test(p$Zinc)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p$Zinc
## W = 0.54335, p-value = 7.109e-11
```

```
shapiro.test(p$Phosphorus)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  p$Phosphorus
## W = 0.97599, p-value = 0.4245
```

```
# min kruskal wallis tests
kruskal.test(m$Cobalt, m$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  m$Cobalt and m$treatmentID
## Kruskal-Wallis chi-squared = 19.207, df = 7, p-value = 0.007562
```

```
kruskal.test(m$Copper, m$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  m$Copper and m$treatmentID
## Kruskal-Wallis chi-squared = 14.9, df = 7, p-value = 0.0373
```

```
kruskal.test(m$Iron, m$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  m$Iron and m$treatmentID
## Kruskal-Wallis chi-squared = 36.787, df = 7, p-value = 5.146e-06
```

```
kruskal.test(m$Manganese, m$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  m$Manganese and m$treatmentID
## Kruskal-Wallis chi-squared = 25.058, df = 7, p-value = 0.0007411
```

```
kruskal.test(m$Nickel, m$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  m$Nickel and m$treatmentID
## Kruskal-Wallis chi-squared = 40.859, df = 7, p-value = 8.617e-07
```

```
kruskal.test(m$Zinc, m$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  m$Zinc and m$treatmentID
## Kruskal-Wallis chi-squared = 6.5544, df = 7, p-value = 0.4767
```

```
kruskal.test(m$Phosphorus, m$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  m$Phosphorus and m$treatmentID
## Kruskal-Wallis chi-squared = 24.769, df = 7, p-value = 0.0008339
```

```
# psyg kruskal wallis tests
kruskal.test(p$Cobalt, p$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  p$Cobalt and p$treatmentID
## Kruskal-Wallis chi-squared = 22.502, df = 7, p-value = 0.002081
```

```
kruskal.test(p$Copper, p$treatmentID)
```

```
## 
##  Kruskal-Wallis rank sum test
## 
## data:  p$Copper and p$treatmentID
## Kruskal-Wallis chi-squared = 30.769, df = 7, p-value = 6.859e-05
```

```
kruskal.test(p$Iron, p$treatmentID)
```

```
## 
##  Kruskal-Wallis rank sum test
## 
## data:  p$Iron and p$treatmentID
## Kruskal-Wallis chi-squared = 24.988, df = 7, p-value = 0.0007625
```

```
kruskal.test(p$Manganese, p$treatmentID)
```

```
## 
##  Kruskal-Wallis rank sum test
## 
## data:  p$Manganese and p$treatmentID
## Kruskal-Wallis chi-squared = 20.779, df = 7, p-value = 0.004112
```

```
kruskal.test(p$Nickel, p$treatmentID)
```

```
## 
##  Kruskal-Wallis rank sum test
## 
## data:  p$Nickel and p$treatmentID
## Kruskal-Wallis chi-squared = 25.31, df = 7, p-value = 0.0006686
```

```
kruskal.test(p$Zinc, p$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  p$Zinc and p$treatmentID
## Kruskal-Wallis chi-squared = 28.331, df = 7, p-value = 0.0001915
```

```
kruskal.test(p$Phosphorus, p$treatmentID)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  p$Phosphorus and p$treatmentID
## Kruskal-Wallis chi-squared = 27.182, df = 7, p-value = 0.0003091
```

```
## everything is significant, use the posthoc kruskal dunn test to detect signifigance between groups
# first min
posthoc.kruskal.dunn.test(Cobalt~treatmentID, data = m, p.adjust.method = "fdr")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                         comparisons of independent samples
##
## data:  Cobalt by treatmentID
##
##            min-10-26 min-10-28 min-100-26 min-100-28 min-100-30 min-50-26
## min-10-28  0.014     -         -          -          -          -
## min-100-26 0.089     0.471     -          -          -          -
## min-100-28 0.553     0.087     0.298      -          -          -
## min-100-30 0.254     0.227     0.567      0.567      -          -
## min-50-26  0.467     0.089     0.360      0.853      0.642      -
## min-50-28  0.122     0.374     0.801      0.375      0.712      0.467
## min-50-30  0.801     0.020     0.122      0.642      0.360      0.567
##            min-50-28
## min-10-28  -
## min-100-26 -
## min-100-28 -
## min-100-30 -
## min-50-26  -
## min-50-28  -
## min-50-30  0.212
##
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Copper~treatmentID, data = m, p.adjust.method = "fdr")
```

```
## Warning in posthoc.kruskal.dunn.test.default(c(6.47e-05, 9.5e-05, 6.22e-05, :
## Ties are present. z-quantiles were corrected for ties.
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                          comparisons of independent samples
##
## data:  Copper by treatmentID
##
##            min-10-26 min-10-28 min-100-26 min-100-28 min-100-30 min-50-26
## min-10-28  0.865     -         -          -          -          -
## min-100-26 0.810     0.928     -          -          -          -
## min-100-28 0.810     0.627     0.592      -          -          -
## min-100-30 0.074     0.036     0.032      0.217      -          -
## min-50-26  0.810     0.936     0.984      0.592      0.032      -
## min-50-28  0.810     0.627     0.592      0.984      0.217      0.592
## min-50-30  0.592     0.514     0.420      0.810      0.420      0.420
##            min-50-28
## min-10-28  -
## min-100-26 -
## min-100-28 -
## min-100-30 -
## min-50-26  -
## min-50-28  -
## min-50-30  0.810
##
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Iron~treatmentID, data = m, p.adjust.method = "fdr")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                          comparisons of independent samples
##
## data:  Iron by treatmentID
##
##            min-10-26 min-10-28 min-100-26 min-100-28 min-100-30 min-50-26
## min-10-28  0.6952    -         -          -          -          -
## min-100-26 0.4705    0.2919    -          -          -          -
## min-100-28 0.0227    0.0086    0.1241     -          -          -
## min-100-30 0.0003    9.8e-05   0.0040     0.2223     -          -
## min-50-26  0.1188    0.0533    0.3906     0.4915     0.0533     -
## min-50-28  0.2316    0.1241    0.6071     0.2919     0.0202     0.6589
## min-50-30  0.0040    0.0015    0.0370     0.5705     0.4705     0.2316
##            min-50-28
## min-10-28  -
## min-100-26 -
## min-100-28 -
## min-100-30 -
## min-50-26  -
## min-50-28  -
## min-50-30  0.1188
##
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Manganese~treatmentID, data = m, p.adjust.method = "fdr")
```

```
## 
##  Pairwise comparisons using Dunn's-test for multiple 
##                          comparisons of independent samples
## 
## data:  Manganese by treatmentID 
## 
##            min-10-26 min-10-28 min-100-26 min-100-28 min-100-30 min-50-26
## min-10-28  0.0171    -         -          -          -          -        
## min-100-26 0.0029    0.6616    -          -          -          -        
## min-100-28 0.0019    0.5053    0.8206     -          -          -        
## min-100-30 0.2186    0.2506    0.1305     0.0887     -          -        
## min-50-26  0.1616    0.3994    0.1624     0.1305     0.8206     -        
## min-50-28  0.0046    0.7584    0.8206     0.7556     0.1616     0.2186   
## min-50-30  0.1305    0.5100    0.2186     0.1616     0.7353     0.8206   
##            min-50-28
## min-10-28  -        
## min-100-26 -        
## min-100-28 -        
## min-100-30 -        
## min-50-26  -        
## min-50-28  -        
## min-50-30  0.2858   
## 
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Nickel~treatmentID, data = m, p.adjust.method = "fdr")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                         comparisons of independent samples
##
## data:  Nickel by treatmentID
##
##            min-10-26 min-10-28 min-100-26 min-100-28 min-100-30 min-50-26
## min-10-28  0.37816   -         -          -          -          -
## min-100-26 0.00853   0.00052   -          -          -          -
## min-100-28 0.81787   0.26996   0.01603    -          -          -
## min-100-30 0.03493   0.00236   0.61584    0.06077    -          -
## min-50-26  0.00557   0.00050   0.86898    0.01061    0.51976    -
## min-50-28  0.51289   0.81787   0.00088    0.37816    0.00557    0.00059
## min-50-30  0.21348   0.02115   0.22671    0.29636    0.47762    0.18486
##            min-50-28
## min-10-28  -
## min-100-26 -
## min-100-28 -
## min-100-30 -
## min-50-26  -
## min-50-28  -
## min-50-30  0.04037
##
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Zinc~treatmentID, data = m, p.adjust.method = "fdr")
```

```
## 
##  Pairwise comparisons using Dunn's-test for multiple 
##                          comparisons of independent samples 
## 
## data:  Zinc by treatmentID 
## 
##            min-10-26 min-10-28 min-100-26 min-100-28 min-100-30 min-50-26
## min-10-28  0.75      -         -          -          -          -        
## min-100-26 0.79      0.75      -          -          -          -        
## min-100-28 0.94      0.75      0.79       -          -          -        
## min-100-30 0.75      0.95      0.75       0.75       -          -        
## min-50-26  0.75      0.81      0.81       0.75       0.79       -        
## min-50-28  0.81      0.75      0.94       0.79       0.75       0.79     
## min-50-30  0.75      0.79      0.81       0.75       0.79       0.95     
##            min-50-28
## min-10-28  -        
## min-100-26 -        
## min-100-28 -        
## min-100-30 -        
## min-50-26  -        
## min-50-28  -        
## min-50-30  0.79     
## 
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Phosphorus~treatmentID, data = m, p.adjust.method = "fdr")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                          comparisons of independent samples
##
## data:  Phosphorus by treatmentID
##
##            min-10-26 min-10-28 min-100-26 min-100-28 min-100-30 min-50-26
## min-10-28  0.162     -         -          -          -          -
## min-100-26 0.011     0.265     -          -          -          -
## min-100-28 0.505     0.510     0.066      -          -          -
## min-100-30 0.668     0.066     0.003      0.267      -          -
## min-50-26  0.063     0.579     0.546      0.242      0.016      -
## min-50-28  0.304     0.668     0.142      0.705      0.162      0.373
## min-50-30  0.853     0.221     0.014      0.567      0.579      0.066
##            min-50-28
## min-10-28  -
## min-100-26 -
## min-100-28 -
## min-100-30 -
## min-50-26  -
## min-50-28  -
## min-50-30  0.373
##
## P value adjustment method: fdr
```

```
# now psyg
posthoc.kruskal.dunn.test(Cobalt~treatmentID, data = p, p.adjust.method = "fdr")
```

```
## 
##  Pairwise comparisons using Dunn's-test for multiple
##                          comparisons of independent samples
## 
## data:  Cobalt by treatmentID
## 
##             psyg-10-26 psyg-10-28 psyg-100-26 psyg-100-28 psyg-100-30
## psyg-10-28  0.156      -          -           -           -          
## psyg-100-26 0.610      0.374      -           -           -          
## psyg-100-28 0.027      0.471      0.098       -           -          
## psyg-100-30 0.446      0.471      0.749       0.162       -          
## psyg-50-26  0.008      0.204      0.026       0.557       0.052      
## psyg-50-28  0.026      0.441      0.074       0.885       0.152      
## psyg-50-30  0.162      0.885      0.420       0.441       0.551      
##             psyg-50-26 psyg-50-28
## psyg-10-28  -          -         
## psyg-100-26 -          -         
## psyg-100-28 -          -         
## psyg-100-30 -          -         
## psyg-50-26  -          -         
## psyg-50-28  0.631      -         
## psyg-50-30  0.162      0.403     
## 
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Copper~treatmentID, data = p, p.adjust.method = "fdr")
```

```
## 
##  Pairwise comparisons using Dunn's-test for multiple 
##                          comparisons of independent samples 
## 
## data:  Copper by treatmentID 
## 
##             psyg-10-26 psyg-10-28 psyg-100-26 psyg-100-28 psyg-100-30
## psyg-10-28  0.4949     -          -           -           -          
## psyg-100-26 0.0292     0.0039     -           -           -          
## psyg-100-28 0.5574     0.1771     0.1367      -           -          
## psyg-100-30 0.0260     0.0039     0.9343      0.1114      -          
## psyg-50-26  0.1740     0.0291     0.5004      0.4900      0.4858     
## psyg-50-28  0.5064     0.9343     0.0040      0.1966      0.0039     
## psyg-50-30  0.0949     0.0146     0.6790      0.2967      0.6098     
##             psyg-50-26 psyg-50-28
## psyg-10-28  -          -         
## psyg-100-26 -          -         
## psyg-100-28 -          -         
## psyg-100-30 -          -         
## psyg-50-26  -          -         
## psyg-50-28  0.0296     -         
## psyg-50-30  0.7652     0.0159    
## 
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Iron~treatmentID, data = p, p.adjust.method = "fdr")
```

```
## 
##  Pairwise comparisons using Dunn's-test for multiple
##                          comparisons of independent samples
## 
## data:  Iron by treatmentID
## 
##             psyg-10-26 psyg-10-28 psyg-100-26 psyg-100-28 psyg-100-30
## psyg-10-28  0.36045    -          -           -           -          
## psyg-100-26 0.04672    0.36045    -           -           -          
## psyg-100-28 0.00073    0.02090    0.23314     -           -          
## psyg-100-30 0.03869    0.28488    0.88369     0.28488     -          
## psyg-50-26  0.00339    0.04672    0.36045     0.75330     0.46548    
## psyg-50-28  0.02090    0.23314    0.75562     0.36045     0.88330    
## psyg-50-30  0.04645    0.33424    0.91788     0.26300     0.91788    
##             psyg-50-26 psyg-50-28
## psyg-10-28  -          -         
## psyg-100-26 -          -         
## psyg-100-28 -          -         
## psyg-100-30 -          -         
## psyg-50-26  -          -         
## psyg-50-28  0.64435    -         
## psyg-50-30  0.40447    0.81110   
## 
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Manganese~treatmentID, data = p, p.adjust.method = "fdr")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                         comparisons of independent samples
##
## data:  Manganese by treatmentID
##
##             psyg-10-26 psyg-10-28 psyg-100-26 psyg-100-28 psyg-100-30
## psyg-10-28  0.1415     -          -           -           -
## psyg-100-26 0.5617     0.5135     -           -           -
## psyg-100-28 0.0265     0.5617     0.1415      -           -
## psyg-100-30 0.0660     0.7787     0.2846      0.7787      -
## psyg-50-26  0.0053     0.2313     0.0265      0.5617      0.4331
## psyg-50-28  0.0265     0.5617     0.1473      0.9835      0.7787
## psyg-50-30  0.0265     0.5617     0.1415      0.9835      0.7787
##             psyg-50-26 psyg-50-28
## psyg-10-28  -          -
## psyg-100-26 -          -
## psyg-100-28 -          -
## psyg-100-30 -          -
## psyg-50-26  -          -
## psyg-50-28  0.5617     -
## psyg-50-30  0.5617     0.9835
##
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Nickel~treatmentID, data = p, p.adjust.method = "fdr")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                          comparisons of independent samples
##
## data:  Nickel by treatmentID
##
##             psyg-10-26 psyg-10-28 psyg-100-26 psyg-100-28 psyg-100-30
## psyg-10-28  0.2321     -          -           -           -
## psyg-100-26 0.0534     0.4919     -           -           -
## psyg-100-28 0.0412     0.3906     0.8676      -           -
## psyg-100-30 0.3906     0.6470     0.2550      0.2321      -
## psyg-50-26  0.0008     0.0464     0.2321      0.2469      0.0129
## psyg-50-28  0.0129     0.2469     0.6363      0.7162      0.1220
## psyg-50-30  0.3906     0.6470     0.2550      0.2321      1.0000
##             psyg-50-26 psyg-50-28
## psyg-10-28  -          -
## psyg-100-26 -          -
## psyg-100-28 -          -
## psyg-100-30 -          -
## psyg-50-26  -          -
## psyg-50-28  0.3906     -
## psyg-50-30  0.0129     0.1220
##
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Zinc~treatmentID, data = p, p.adjust.method = "fdr")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                         comparisons of independent samples
##
## data:  Zinc by treatmentID
##
##             psyg-10-26 psyg-10-28 psyg-100-26 psyg-100-28 psyg-100-30
## psyg-10-28  0.06397    -          -           -           -
## psyg-100-26 1.00000    0.06397    -           -           -
## psyg-100-28 0.71108    0.12285    0.71558     -           -
## psyg-100-30 0.12132    0.00039    0.12132     0.06397     -
## psyg-50-26  1.00000    0.06397    1.00000     0.71108     0.12132
## psyg-50-28  0.33316    0.37028    0.33911     0.64959     0.01479
## psyg-50-30  0.12285    0.00039    0.12285     0.06397     0.94533
##             psyg-50-26 psyg-50-28
## psyg-10-28  -          -
## psyg-100-26 -          -
## psyg-100-28 -          -
## psyg-100-30 -          -
## psyg-50-26  -          -
## psyg-50-28  0.33316    -
## psyg-50-30  0.12285    0.01479
##
## P value adjustment method: fdr
```

```
posthoc.kruskal.dunn.test(Phosphorus~treatmentID, data = p, p.adjust.method = "fdr")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                          comparisons of independent samples
##
## data:  Phosphorus by treatmentID
##
##             psyg-10-26 psyg-10-28 psyg-100-26 psyg-100-28 psyg-100-30
## psyg-10-28  0.067      -          -           -           -
## psyg-100-26 0.201      0.752      -           -           -
## psyg-100-28 0.053      0.935      0.752       -           -
## psyg-100-30 0.758      0.024      0.073       0.023       -
## psyg-50-26  0.023      0.752      0.409       0.758       0.015
## psyg-50-28  0.023      0.752      0.409       0.758       0.015
## psyg-50-30  0.866      0.042      0.127       0.033       0.866
##             psyg-50-26 psyg-50-28
## psyg-10-28  -          -
## psyg-100-26 -          -
## psyg-100-28 -          -
## psyg-100-30 -          -
## psyg-50-26  -          -
## psyg-50-28  0.984      -
## psyg-50-30  0.018      0.018
##
## P value adjustment method: fdr
```

```
#######################
# make the figure
# rename dataframe
um3 <- dat_all
# for the error bars
pd <- position_dodge(0.1)
# melt the data
# the order of the measured variables is important for facet wrapping later
melt2 <- melt(um3, id=c("Species", "Ironconc", "Temp", "treatmentID"), measure.vars=c("Phosphorus","Iron", "Manganese", "Zinc", "Cobalt", "Copper", "Nickel"))
```

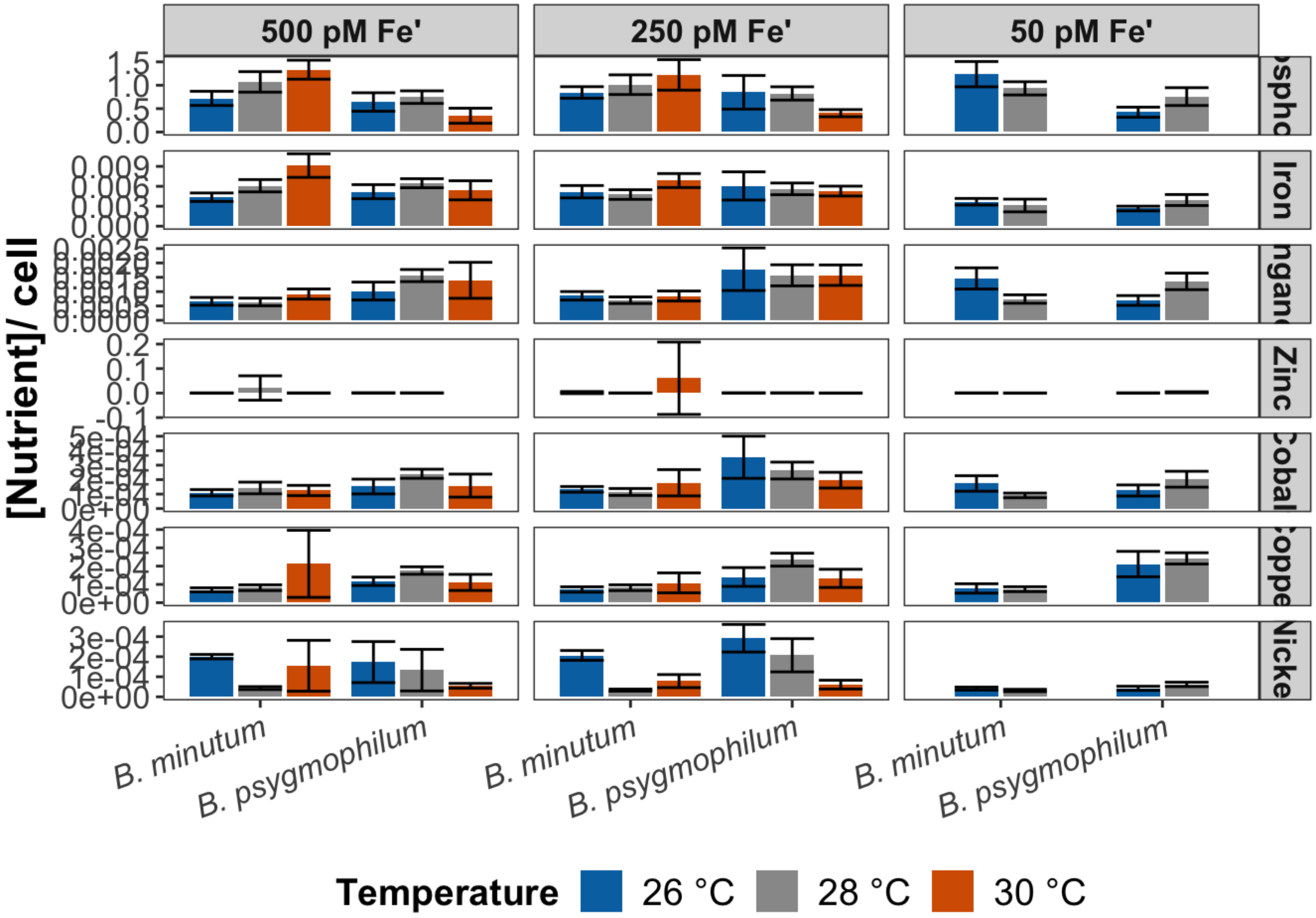
```
## Warning in melt(um3, id = c("Species", "Ironconc", "Temp", "treatmentID"), :
## The melt generic in data.table has been passed a data.frame and will attempt
## to redirect to the relevant reshape2 method; please note that reshape2 is
## deprecated, and this redirection is now deprecated as well. To continue using
## melt methods from reshape2 while both libraries are attached, e.g. melt.list,
## you can prepend the namespace like reshape2::melt(um3). In the next version,
## this warning will become an error.
```

```
melt2 <- as.data.frame(melt2)
# rename factors so they are publication ready
melt2$Species <- as.factor(melt2$Species)
melt2$Species <- factor(melt2$Species, levels = c("min", "psyg"))
levels(melt2$Species) <- c("B. minutum", "B. psygmophilum")
# breviolum iron conc
# this converts total dissolved iron concentrations (nM Fe) into bioavailable iron concentrations (pM Fe'). Brief
ly, Fe' values indicate the fraction of the total dissolved [Fe] that remain bioavailable after the addition of E
DTA
melt2$Ironconc <- as.factor(melt2$Ironconc)
melt2$Ironconc <- factor(melt2$Ironconc, levels =c("100", "50", "10"))
levels(melt2$Ironconc) <- c("500 pM Fe'", "250 pM Fe'", "50 pM Fe'")
melt2$treatmentID <- as.factor(melt2$treatmentID)
melt2$variable <- as.factor(melt2$variable)
# make sure the order of the nutrients reflects concentration in sym cell (from highest to lowest)
melt2$variable <- factor(melt2$variable, levels = c("Phosphorus","Iron", "Manganese", "Zinc", "Cobalt", "Copper",
"Nickel"))
# create a color blind palette
cols <- c( "t26" = "#0072B2", "t28" = "#999999", "t30" = "#D55E00")
# remove missing data
melt3 <- na.omit(melt2)
# get summary stats
stats <- melt2 %>% group_by(Ironconc, Species, Temp, variable, treatmentID) %>% summarise_all(list(avg = mean, st
dev = sd))
stats$Temp <- as.factor(stats$Temp)

# make the bar graph
bbar <- ggplot(data=stats, aes(x = Species, y=avg, fill=Temp)) +
  theme_bw() +
  geom_bar(stat = "identity", position = position_dodge2(preserve = "single"), aes(fill=Temp)) +
  geom_errorbar(aes(ymin=avg-stdev, ymax=avg+stdev), width=0.9, colour="black", position=position_dodge2(width =
0.9, preserve = "single")) +
  theme(axis.text.x = element_text(angle=20, vjust=1, hjust=1), axis.title.x = element_blank()) +
  facet_grid(variable~Ironconc, scales = "free") +
  theme(axis.text.y = element_text(size=12),
        panel.grid = element_blank()) +
  theme(axis.text.x = element_text(size=12, face = "italic")) +
  theme(axis.title = element_text(size=16, face = "bold")) +
  theme(legend.text = element_text(size=14)) +
  theme(legend.title = element_text(size=14, face = "bold")) +
  labs(y="[Nutrient]/ cell",
       x="Species") +
  theme(legend.position = "bottom")  +
  theme(strip.text = element_text(size=12, face = "bold",margin = margin(.15, .15, .15, .15, "cm"))) +
  scale_fill_manual(values = cols, name = "Temperature", breaks = c("t26", "t28", "t30"), labels=c("26 °C", "28 °
C", "30 °C"))
bbar
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```

```
## Warning: Removed 1 rows containing missing values (geom_errorbar).
```



multivariate figures and stats

```
# load for mosaic plots
library(ggmosaic)
#back to using the dat dataframe (includes everything)
dat <- um3
dat <- as.data.frame(dat)
# make treatmentID a factor
dat$Species <- as.factor(dat$Species)
dat$treatmentID <- as.factor(dat$treatmentID)
# make an object for the dependent variables (the trace metals)
dependent.vars <- dat[,6:12]
dependent.vars <- as.matrix(dependent.vars)
# make a na matrix of the data
na <- na.omit(dat)
#### LDA/DFA time
# SPECIES
rh <-lda(dat$Species ~ dependent.vars,tol = 1.0e-25, CV = F)
rh # this gives you info on priors ($prior), coefficients of linear discriminants ($scaling), group means ($means)
```

```
## Call:
## lda(dat$Species ~ dependent.vars, tol = 1e-25, CV = F)
## 
## Prior probabilities of groups:
##       min      psyg 
## 0.5052632 0.4947368 
## 
## Group means:
##      dependent.varsPhosphorus dependent.varsManganese dependent.varsIron
## min                 1.0466695             0.000852249        0.005395155
## psyg                0.6359264             0.001390462        0.005114709
##      dependent.varsCobalt dependent.varsNickel dependent.varsCopper
## min          0.0001333316         9.862733e-05         9.736088e-05
## psyg         0.0002154998         1.300990e-04         1.735203e-04
##      dependent.varsZinc
## min        0.0106241135
## psyg       0.0008076263
## 
## Coefficients of linear discriminants:
##                                   LD1           LD2
## dependent.varsPhosphorus    -5.554809    -0.4009159
## dependent.varsManganese    556.417812 -3879.4673416
## dependent.varsIron          64.448645  -158.0154103
## dependent.varsCobalt     13978.288857 16874.6915511
## dependent.varsNickel     -3085.293274 -1829.9620380
## dependent.varsCopper      9534.560365  5407.1254881
## dependent.varsZinc         -14.139780   -14.0802860
## 
## Proportion of trace:
## LD1 LD2 
##   1   0
```

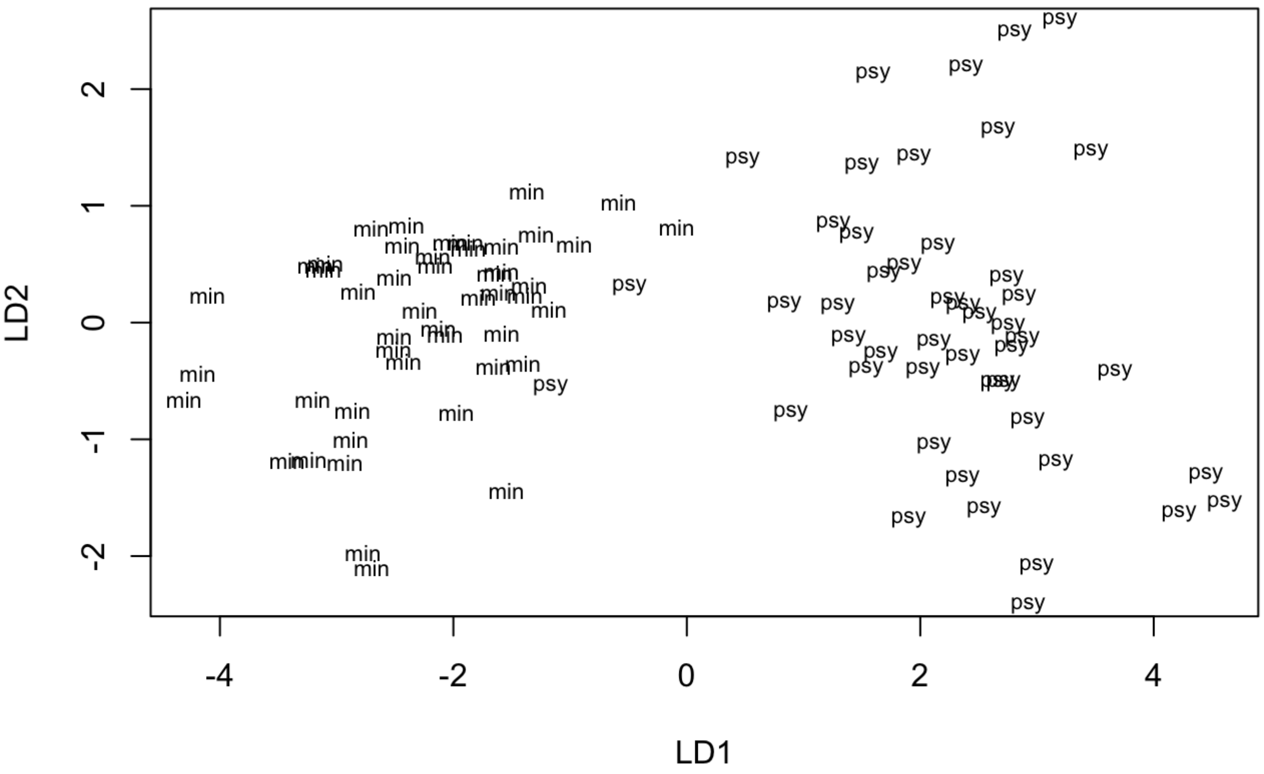
```
rh$counts # this tells you how many in each independent group
```

```
##  min psyg 
##   48   47
```

```
rh$svd # the singular values (svd) that gives the ratio of the between- and within-group standard deviations on t
he linear discriminant variables.
```

```
## [1] 2.182877e+01 3.313442e-15
```

```
x <- rh$svd
#pdf("brev_vol_dfa.pdf")
plot(rh, abbrev = 3, dimen = 2) # this plots the 2st to linear discriminants
```



```
dev.off()
```

```
## null device
##           1
```

```
prop = x^2/sum(x^2)
prop
```

```
## [1] 1.000000e+00 2.304094e-32
```

```
# create a table to see how well the dfa categorized things
# switch CV = T
rh_dfa <- lda(dat$Species ~ dependent.vars, CV=T, tol = 1.0e-25)
t <- table(na$Species, rh_dfa$class)
#pdf("brev_mos_vol.pdf")
mosaicplot(t, color = T,
           main = "Species",
           xlab = "Actual",
           ylab = "Predicted",
           las = 0.8)
dev.off()
```

```
## null device
##           1
```

```
p <- diag(prop.table(t,1))
total <- cbind(p,t)
total
```

```
##              p min psyg
## min  0.9791667  47    1
## psyg 0.9574468   2   45
```

```
# stacked histogram of predicted values
lda.val <- predict(rh)
ldahist(lda.val$x[,1], g = dat$Species, col = "grey")
#####################
# make objects for min and psyg
m <- dat %>% filter(Species %in% c("min"))
psy <- dat %>% filter(Species %in% c("psyg"))
##############
#####################
# only minutum
#####################
##############
# make a matrix of dependent variables
dependent.vars <- m[,6:12]
dependent.vars <- as.matrix(dependent.vars)
# make a na matrix of the ma
na <- na.omit(m)
#### LDA/DFA time
# IRON CONCENTRATION
rh <-lda(m$Ironconc ~ dependent.vars, CV = F, tol = 1.0e-25)
rh
```

```
## Call:
## lda(m$Ironconc ~ dependent.vars, CV = F, tol = 1e-25)
##
## Prior probabilities of groups:
##    10    50   100
## 0.250 0.375 0.375
##
## Group means:
##     dependent.varsPhosphorus dependent.varsManganese dependent.varsIron
## 10                  1.085214            0.0011006614        0.003393570
## 50                  1.026317            0.0008000421        0.005603518
## 100                 1.041325            0.0007388475        0.006521183
##     dependent.varsCobalt dependent.varsNickel dependent.varsCopper
## 10          0.0001321757         3.619167e-05         7.587933e-05
## 50          0.0001419353         1.060199e-04         8.750228e-05
## 100         0.0001254986         1.328585e-04         1.215405e-04
##     dependent.varsZinc
## 10        0.0003422088
## 50        0.0209743463
## 100       0.0071284839
##
## Coefficients of linear discriminants:
##                                   LD1           LD2           LD3
## dependent.varsPhosphorus    -2.399108     -1.000965     -5.016002
## dependent.varsManganese  -8242.267208   2439.449032  -2695.753241
## dependent.varsIron         791.606329    205.640506      4.932442
## dependent.varsCobalt     55152.404698 -26760.901529  53068.801686
## dependent.varsNickel      5410.955468  -2870.260925 -12588.750768
## dependent.varsCopper     -3868.943085   9983.786930  14552.759565
## dependent.varsZinc         -41.903036      6.158483    -44.820592
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.9826 0.0174 0.0000
```

```
rh$counts
```

```
##  10  50 100
##  12  18  18
```

```
rh$svd
```

```
## [1] 1.038634e+01 1.384093e+00 6.911013e-15
```

```
x <- rh$svd
#pdf("min_dfa_fe_vol.pdf")
plot(rh, abbrev = 3, dimen = 2)
dev.off()
```

```
## null device
##           1
```

```
prop = x^2/sum(x^2)
prop
```

```
## [1] 9.825514e-01 1.744860e-02 4.350242e-31
```

```
# create a table to see how well the dfa categorized things
# switch CV = T
rh_dfa <- lda(m$Ironconc ~ dependent.vars, CV=T, tol = 1.0e-25)
t <- table(na$Ironconc, rh_dfa$class)
#pdf("min_mos_fe_vol.pdf")
mosaicplot(t, color = T,
           main = "Iron concentration",
           xlab = "Actual",
           ylab = "Predicted",
           las = 0.8,
           off = 10)
dev.off()
```

```
## null device 
##           1
```

```
p <- diag(prop.table(t,1))
total <- cbind(p,t)
total
```

```
##             p 10 50 100
## 10  1.0000000 12  0   0
## 50  0.6111111  0 11   7
## 100 0.4444444  0 10   8
```

```
# stacked histogram of predicted values
lda.val <- predict(rh)
ldahist(lda.val$x[,1], g = m$Ironconc, col = "grey")
# TEMPERATURE TREATMENT
rh <-lda(m$Temp ~ dependent.vars, CV = F, tol = 1.0e-25)
rh
```

```
## Call:
## lda(m$Temp ~ dependent.vars, CV = F, tol = 1e-25)
## 
## Prior probabilities of groups:
##   t26   t28   t30 
## 0.375 0.375 0.250 
## 
## Group means:
##     dependent.varsPhosphorus dependent.varsManganese dependent.varsIron
## t26                0.9340503            0.0009914466        0.004407210
## t28                1.0057630            0.0006936478        0.004651112
## t30                1.2769582            0.0008813542        0.007993139
##     dependent.varsCobalt dependent.varsNickel dependent.varsCopper
## t26         0.0001384718         1.491467e-04         7.340422e-05
## t28         0.0001160221         3.612222e-05         7.923889e-05
## t30         0.0001515857         1.166059e-04         1.604788e-04
##     dependent.varsZinc
## t26       0.0009358537
## t28       0.0070832669
## t30       0.0304677733
## 
## Coefficients of linear discriminants:
##                                    LD1           LD2          LD3
## dependent.varsPhosphorus  3.954550e+00      1.966633    -3.995359
## dependent.varsManganese  -6.242046e+03   4229.210110 -3982.632419
## dependent.varsIron        2.791318e+02    539.238375   237.256170
## dependent.varsCobalt     -4.719156e+03 -34478.672784 58274.988514
## dependent.varsNickel     -2.347976e+04  10239.627178 -5158.074793
## dependent.varsCopper      1.651798e+04  -6021.752642  4467.279306
## dependent.varsZinc        7.512751e-01     29.704442   -41.996833
## 
## Proportion of trace:
##    LD1    LD2    LD3 
## 0.8268 0.1732 0.0000
```

```
rh$counts
```

```
## t26 t28 t30
##  18  18  12
```

```
rh$svd
```

```
## [1] 1.215891e+01 5.565646e+00 5.690638e-15
```

```
x <- rh$svd
#pdf("min_dfa_temp_vol.pdf")
plot(rh, abbrev = 3, dimen = 2,
     cex = 1.3,
     cex.axis = 1.5,
     cex.lab = 1.6)
dev.off()
```

```
## null device
##           1
```

```
prop = x^2/sum(x^2)
prop
```

```
## [1] 8.267689e-01 1.732311e-01 1.810992e-31
```

```
# create a table to see how well the dfa categorized things
# switch CV = T
rh_dfa <- lda(m$Temp ~ dependent.vars, CV=T, tol = 1.0e-25)
t <- table(na$Temp, rh_dfa$class)
#pdf("min_temp_mos_vol.pdf")
mosaicplot(t, color = T,
           main = "Temperature",
           xlab = "Actual",
           ylab = "Predicted",
           las = 0.8,
           off = 10)
dev.off()
```

```
## null device
##           1
```

```
p <- diag(prop.table(t,1))
total <- cbind(p,t)
total
```

```
##             p t26 t28 t30
## t26 1.0000000  18   0   0
## t28 0.9444444   0  17   1
## t30 0.5833333   2   3   7
```

```
# stacked histogram of predicted values
lda.val <- predict(rh)
ldahist(lda.val$x[,1], g = m$Temp, col = "grey")
# make a less ugly scatterplot of LDA, helpful website: https://rpubs.com/ifn1411/LDA
# useful for ellipses https://ggplot2.tidyverse.org/reference/stat_ellipse.html
# wrangle the data
newdata1 <- data.frame(Temp = na[,4], Ironconc = na[,3],lda= lda.val$x)
newdata1$Temp <- as.factor(newdata1$Temp)
newdata1$Ironconc <- as.factor(newdata1$Ironconc)
# create a color blind palette
cols <-c( "t26" = "#0072B2", "t28" = "#999999", "t30" = "#D55E00")
labs <- c("26°C", "28°C", "30°C")
##############
#####################
# only psygmophilum
#####################
##############
# make a matrix of dependent variables
dependent.vars <- psy[,6:12]
dependent.vars <- as.matrix(dependent.vars)
# make a na matrix of the data
na <- na.omit(psy)
###
# LDA/DFA time
# IRON CONCENTRATION
rh <-lda(psy$Ironconc ~ dependent.vars, CV = F, tol = 1.0e-25)
rh
```

```
## Call:
## lda(psy$Ironconc ~ dependent.vars, CV = F, tol = 1e-25)
## 
## Prior probabilities of groups:
##        10        50       100 
## 0.2553191 0.3829787 0.3617021 
## 
## Group means:
##     dependent.varsPhosphorus dependent.varsManganese dependent.varsIron
## 10                 0.5907927             0.001024233        0.003295174
## 50                 0.6926552             0.001640891        0.005649588
## 100                0.6077198             0.001383818        0.005832745
##     dependent.varsCobalt dependent.varsNickel dependent.varsCopper
## 10          0.0001639407         5.216667e-05         0.0002267604
## 50          0.0002713701         1.867085e-04         0.0001696013
## 100         0.0001927376         1.251706e-04         0.0001400884
##     dependent.varsZinc
## 10        0.0017053520
## 50        0.0005020392
## 100       0.0004975005
## 
## Coefficients of linear discriminants:
##                                    LD1           LD2           LD3
## dependent.varsPhosphorus     -1.785466      2.402896      9.622893
## dependent.varsManganese    -137.533269  -2229.578189   2429.683061
## dependent.varsIron         1854.140508   1032.770455  -1257.170667
## dependent.varsCobalt     -15036.863916 -13809.935145  -7961.895976
## dependent.varsNickel       7855.024718  -6129.472862  -5988.059128
## dependent.varsCopper     -10690.944818   3388.830558 -22853.761296
## dependent.varsZinc         -269.160590    238.354825   -694.823559
## 
## Proportion of trace:
##    LD1    LD2    LD3 
## 0.8833 0.1167 0.0000
```

```
rh$counts
```

```
##  10  50 100
##  12  18  17
```

```
rh$svd
```

```
## [1] 1.150419e+01 4.182512e+00 5.694745e-15
```

```
x <- rh$svd
#pdf("psyg_dfa_fe_vol.pdf")
plot(rh, abbrev = 3, dimen = 2)
dev.off()
```

```
## null device
##           1
```

```
prop = x^2/sum(x^2)
prop
```

```
## [1] 8.832526e-01 1.167474e-01 2.164319e-31
```

```
# create a table to see how well the dfa categorized things
# switch CV = T
rh_dfa <- lda(psy$Ironconc ~ dependent.vars, CV=T, tol = 1.0e-25)
t <- table(na$Ironconc, rh_dfa$class)
#pdf("psyd_mos_fe_vol.pdf")
mosaicplot(t, color = T,
           main = "Iron concentration",
           xlab = "Actual",
           ylab = "Predicted",
           las = 0.8,
           off = 10)
dev.off()
```

```
## null device 
##           1
```

```
p <- diag(prop.table(t,1))
total <- cbind(p,t)
total
```

```
##             p 10 50 100
## 10  1.0000000 12  0   0
## 50  0.8888889  0 16   2
## 100 0.8235294  0  3  14
```

```
# stacked histogram of predicted values
lda.val <- predict(rh)
ldahist(lda.val$x[,1], g = psy$Ironconc, col = "grey")
# TEMPERATURE TREATMENT
rh <-lda(psy$Temp ~ dependent.vars, CV = F, tol = 1.0e-25)
rh
```

```
## Call:
## lda(psy$Temp ~ dependent.vars, CV = F, tol = 1e-25)
##
## Prior probabilities of groups:
##       t26       t28       t30
## 0.3829787 0.3829787 0.2340426
##
## Group means:
##     dependent.varsPhosphorus dependent.varsManganese dependent.varsIron
## t26                0.6383349             0.001163960        0.004633167
## t28                0.7762304             0.001494967        0.005336849
## t30                0.4023969             0.001590096        0.005539187
##     dependent.varsCobalt dependent.varsNickel dependent.varsCopper
## t26         0.0002106983         0.0001691669         0.0001562465
## t28         0.0002354722         0.0001340527         0.0002178221
## t30         0.0001906746         0.0000597000         0.0001292925
##     dependent.varsZinc
## t26       0.0005989768
## t28       0.0013925918
## t30       0.0001918366
##
## Coefficients of linear discriminants:
##                                    LD1           LD2           LD3
## dependent.varsPhosphorus     -9.011006     -1.080019      8.380548
## dependent.varsManganese    5842.710064  -1276.546458   3144.482943
## dependent.varsIron          277.683412   -535.677692   -490.375303
## dependent.varsCobalt     -13786.622950  14838.499765 -27757.687900
## dependent.varsNickel      -1744.409546   1465.039199    149.497641
## dependent.varsCopper        802.292711 -17125.792510 -11255.970190
## dependent.varsZinc           33.403837   -254.617970   -586.440563
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.8658 0.1342 0.0000
```

```
rh$counts
```

```
## t26 t28 t30
##  18  18  11
```

```
rh$svd
```

```
## [1] 1.027875e+01 4.047211e+00 5.176292e-15
```

```
x <- rh$svd
#pdf("psyg_temp_dfa_vol.pdf")
plot(rh, abbrev = 3, dimen = 2,
     cex = 1.3,
     cex.axis = 1.5,
     cex.lab = 1.6)
dev.off()
```

```
## null device
##           1
```

```
prop = x^2/sum(x^2)
prop
```

```
## [1] 8.657742e-01 1.342258e-01 2.195643e-31
```

```
# create a table to see how well the dfa categorized things
# switch CV = T
rh_dfa <- lda(psy$Temp ~ dependent.vars, CV=T, tol = 1.0e-25)
t <- table(na$Temp, rh_dfa$class)
#pdf("psyg_temp_mos_vol.pdf")
mosaicplot(t, color = T,
           main = "Temperature",
           xlab = "Actual",
           ylab = "Predicted",
           las = 0.8,
           off = 10)
dev.off()
```

```
## null device
##           1
```

```
p <- diag(prop.table(t,1))
total <- cbind(p,t)
total
```

```
##             p t26 t28 t30
## t26 0.6666667  12   6   0
## t28 0.7777778   3  14   1
## t30 1.0000000   0   0  11
```

```
# stacked histogram of predicted values
lda.val <- predict(rh)
ldahist(lda.val$x[,1], g = psy$Temp, col = "grey")
# make a less ugly scatterplot of LDA, helpful website: https://rpubs.com/ifn1411/LDA
# useful for ellipses https://ggplot2.tidyverse.org/reference/stat_ellipse.html
# wrangle the data
newdata <- data.frame(Temp = na[,4], Ironconc = na[,3],lda= lda.val$x)
newdata$Temp <- as.factor(newdata$Temp)
newdata$Ironconc <- as.factor(newdata$Ironconc)
#### facet this plot (instead of making one scatter plot for bmin and one for bspyg). faceting makes life better
# add a species column for faceting. newdata1 is Bmin, newdata is Bpsyg. *headdesk*
newdata1$Species <- c("Breviolum minutum")
newdata$Species <- c("Breviolum psygmophilum")
# rbind them together
bothspp <- rbind(newdata, newdata1)
### make figure 3!
both <- ggplot(bothspp) +
  geom_point(aes(bothspp$lda.LD1, bothspp$lda.LD2,
                 color = bothspp$Temp,
                 shape = bothspp$Ironconc),
                 size = 3.5) +
  theme_bw() +
  facet_wrap(~Species) +
  labs(x = "LD1", y = "LD2") +
  scale_color_manual(values = cols, breaks = c("t26", "t28", "t30"), labels=c("26°C", "28°C", "30°C"), name = "Tem
perature") +
  scale_shape_manual(values = c(21,22,24),
                name = "Bioavailable Iron \nConcentration (pM Fe')") +
  theme(axis.text = element_text(size=12),
        axis.title = element_text(size=12, face = "bold"),
        legend.text = element_text(size=10),
        panel.grid = element_blank(),
        legend.title = element_text(size=10, face="bold"), legend.title.align = 0,
        strip.text = element_text(size = 12, face = "bold.italic")) +
  stat_ellipse(geom = "polygon", type = "t", alpha = 0.125, show.legend = FALSE,
               aes(bothspp$lda.LD1, bothspp$lda.LD2, fill = bothspp$Temp)) +
               scale_fill_manual(values = cols)
print(both)
#save_plot("~/Desktop/PhD/chapters/chapter 4 - temp treatments/for_coauthors_v1/figs/figs_v4_01212020/Fig3_um3_mu
ltivar.pdf",both, base_aspect_ratio = 1.9)
```