p180 Metabolite Analysis

Audrey Brown

8/27/2019

Loading Packages and Initial Cleaning of Data

```
library(tidyverse)
library(vegan)
library(broom)

#for all raw metabolite adundances from MetIDQ version Nitrogen see file p180_RawData_AllMets.csv

#read in file with the 116 Metabolites detected in at least 50% of samples
p180Demo<-read_csv("p180_Aug2019.csv")
length(unique(p180Demo$Metabolite))

#find half lowest value for each metabolite
p180Demo <- p180Demo %>% mutate(posmin = invoke(pmin, na_if(., 0), na.rm = TRUE))
p180Demo$posmin<-as.numeric(p180Demo$posmin)
p180Demo<- p180Demo %>% mutate(halfposmin = (posmin*0.5))

#write file and replace zeros with half the lowest value for each metabolite
write.csv(p180Demo, file = "MetabolitesHalfMin_Aug19.csv")
```

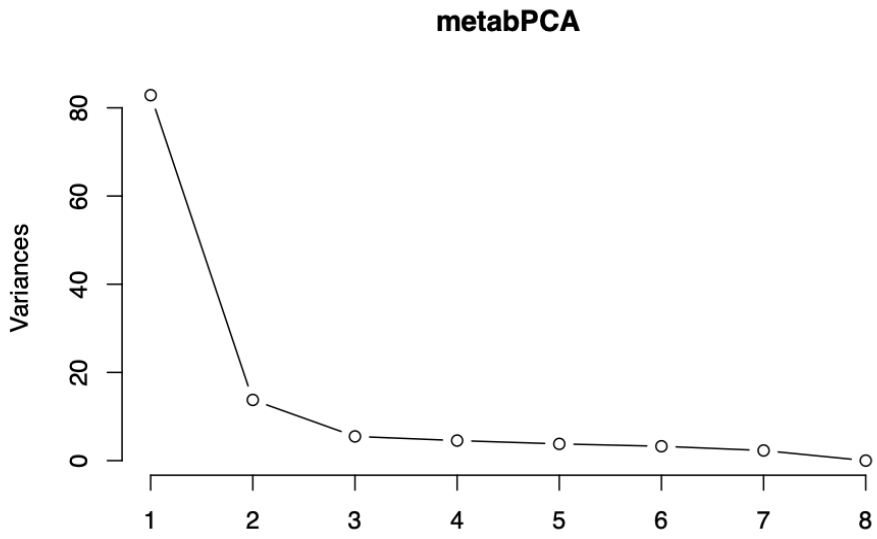
Transformation of Data and Principal Component Analysis

```
#read in the file that has zeros replaced
p180forPCA<-read.csv("MetabolitesforPCA_Aug19.csv")
#columns are metabolites, rows are samples, group is SLOPE or Control

#generate matrix
MetabMatrix<-data.matrix(p180forPCA)
MetabMatrix
rownames(MetabMatrix)<-c("C1", "C2", "C3", "C4", "S1", "S2", "S3", "S4")

#transform data
transformed <-log2(MetabMatrix[,1:116])
treatmentgroup<-MetabMatrix[,117]

#prinical component analysis
metabPCA<-prcomp(transformed,center=TRUE,scale = TRUE)
print(metabPCA)
plot(metabPCA,type="l")
```



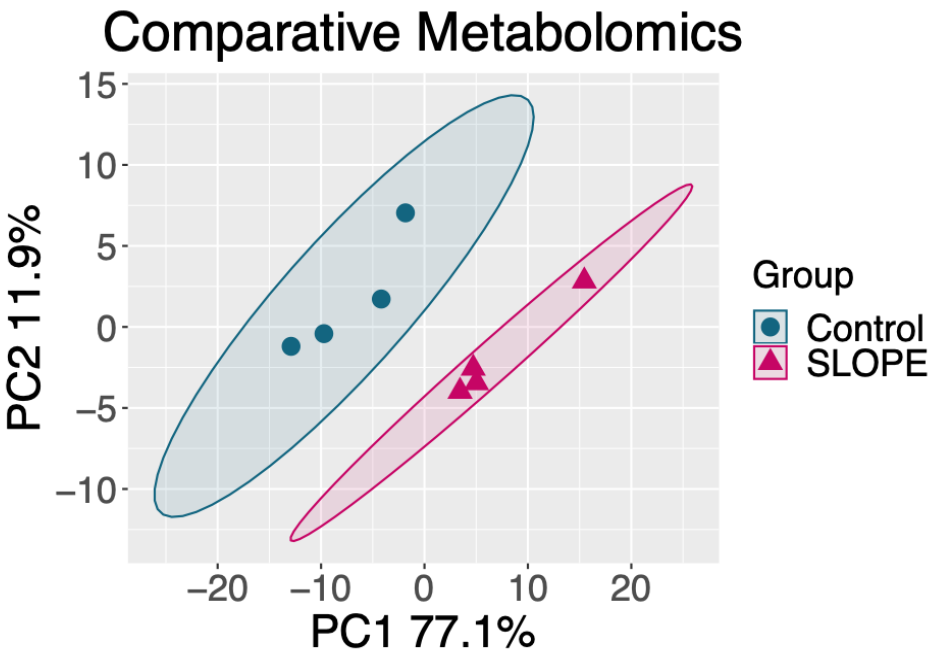

```
#getting PCA Loadings
PCALoading<-(metabPCA$rotation)
PCALoading<-as.data.frame(PCALoading)

#write loadings to file
write.csv(PCALoading, file = "PCA_Loadings_Aug19.csv")

#putting group back on
MetabPCAi<-data.frame(metabPCA$x,Group=p180forPCA$Group)

#Plotting PCA
MetabPCAplot <- ggplot(MetabPCAi,aes(x=PC1,y=PC2, col=Group)) +
  scale_color_manual(values=c("#136681", "#C80566"))+
  scale_fill_manual(values=c("#136681", "#C80566"))+
  geom_point(size=4, aes(shape= Group)) +
  ggtitle("Comparative Metabolomics") +
  theme(plot.title = element_text(hjust = 0.5)) +
  stat_ellipse(geom = "polygon", alpha = .1, aes(fill = Group)) +
  xlab("PC1 77.1%") +
  ylab("PC2 11.9%") +
  theme(axis.text=element_text(size=18), axis.title=element_text(size=22), plot.title=element_text(size

#Figure 5B
plot(MetabPCAplot)
```



Determine Significantly Different Metabolites between Groups

```
#changing group to categorical variable
MetabPCAi$Group<-as.factor(MetabPCAi$Group)

#Metabolites with statistical differences between groups
#Gather by metbaolites and group by treatment
preprocessed<- scale(transformed, center = TRUE, scale = TRUE)
preprocessed<- cbind(preprocessed, treatmentgroup)
preprocessedtbl<-as_tibble(preprocessed)

#run paired t-test
preprocessedtest<- preprocessedtbl %>%
  gather(key=Metabolite, value= value, 1:116)%>%
  group_by(Metabolite) %>%
  do(tidy(t.test(value ~ treatmentgroup, data = ., paired=TRUE)))

#perform multiple testing correction and add to tibble
p.adjusted<-p.adjust(preprocessedtest$p.value, method = "BH", n=116)

preprocessedtest<- as.data.frame(preprocessedtest) %>%
  mutate(p.adjusted = p.adjusted)
```

Determine Fold Change and Combine with Significance Data

```
#Metabolite fold changes----
p180forfc<- p180forPCA %>%
  gather(key=Metabolite, value= value, 1:116)

#Mean of each metabolite for each group
metmeans<- p180forfc %>%
  group_by(Metabolite, Group)%>%
  summarize(metmeanval = mean(value)) %>%
  arrange(desc(Group))

#Pull out values as a vector
SLOPEmeans<-metmeans$metmeanval[1:116]
Controlmeans<-metmeans$metmeanval[117:232]

#divide SLOPE by control and add met names back
fc<-SLOPEmeans/Controlmeans
fc<-as_tibble(fc)
fc<-fc %>% mutate(Metabolite = metmeans$Metabolite[1:116])
fc<-rename(fc, fold_change = value)

#merge fold change information with t-test tibble by metabolite name
finaldata<- merge(preprocessedtest, fc, by = "Metabolite")
View(finaldata)

#Write supplemental file 2
write.csv(finaldata, file = "p180_Metabolite_Data.csv")
```

PERMANOVA Analysis

```
preprocessed_d<-as.data.frame(preprocessed)
preprocessed_c <- data.frame(preprocessed_d[ ,1:116])

perMAN<-adonis(preprocessed_c ~ treatmentgroup, data = preprocessed_d, method = "eu") # PerMANOVA
print(perMAN) #p=0.037
```