

Field Old and Young Leaves Secondary Metabolite Analysis

Kenia E. Segura Aba

5/17/2020

This file was run in R version 4.0.0. The packages used are `tidyverse` version 1.3.0, `readr` version 1.3.1, `RRPP` version 0.5.2, `mixOmics` version 6.12.0, and `labdsv` version 2.0-1. This file must be in the same directory as the Box sync folder in order to run. The following analysis of secondary metabolites was conducted using a split-split-plot analysis of variance (ANOVA) of *P. virgatum* leaves (pre-processed in XCMS Online with a `pairwise` job) using residual randomization permutation procedure (RRPP). Patterns in metabolite classification were visualized using `mixOmics` for principle component analysis (PCA) and partial least squares discriminant analysis (PLS-DA). Dufrene-Legendre indicator analysis was performed to identify specific metabolites indicative of plant response to water treatment and fungal treatment (`labdsv`).

1. Load necessary packages

```
library(tidyverse)
library(readr)
library(RRPP)
library(mixOmics)
library(labdsv)
```

Secondary Metabolites (neg)

RRPP

2. Define dependent variable matrix and class matrix.

```
path <- "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/XCMS Onl

SM_neg <- read_tsv(paste(path, "HvL_SM_neg/XCMS.annotated.diffreport..High_F_SMneg_Kenia_Thesis.vs.Low_L
  sep = ""))

# dependent variable: metabolite intensities
Y <- SM_neg[, c(9, 23:108)] %>% data.frame(row.names = 1) %>%
  t %>% data.frame()
scaled_Y <- scale(Y)

# class: sample factors
class <- read_csv(paste(path, "HvL_class.csv", sep = ""), header = T,
  row.names = 1)
```

3. Define and run multivariate regression models, then print out the results.

```
# LMneg <- lm.rrpp(scaled_Y ~ Block * Water * Fungus * Age,
# data = class, SS.type = 'III', print.progress = F);
# summary(LMneg)

# Error: vector memory exhausted (limit reached?) Error
```

```
# during wrapup: vector memory exhausted (limit reached?)
# Error: no more error handlers available (recursive
# errors?); invoking 'abort' restart
```

4. Examine RRPP plots to check for assumptions.

```
# residuals vs fitted values (homoscedasticity check)
# diagnostics <- plot(LMneg, type = 'diagnostics')

# pca plot pcplot <- plot(LMneg, type = 'PC', pch = 19, col =
# interaction(class$Water, class$Fungus))
```

5. Perform an RRPP ANOVA and print results.

```
## Old Leaves ANOVAneg <- anova(LMneg, effect.type = 'F',
## error = c('Residuals', 'Block:Water', 'Block:Water:Fungus',
## 'Block:Water:Fungus:Age', 'Residuals',
## 'Block:Water:Fungus', 'Block:Water:Fungus',
## 'Block:Water:Fungus:Age', 'Block:Water:Fungus:Age',
## 'Block:Water:Fungus:Age', 'Residuals',
## 'Block:Water:Fungus:Age', 'Block:Water:Fungus:Age',
## 'Block:Water:Fungus:Age', 'Residuals')) ; summary(ANOVAneg,
## formula = T)
```

6. Test lm.rrpp model coefficients. “d” is the amount of change in a variable for the coefficient indicated.

```
# negcoef <- coef(LMneg, test = T) ; summary(negcoef)
```

7. Test pairwise differences between least squares means. Similar to tukeyHSD function in the r stats package. The pairwise function will generate tables with confidence intervals and p-values for the pairwise statistic, Euclidean distance between least-squares means.

```
# fungus negpw <- pairwise(LMneg, groups = class$Fungus);
# summary(negpw, confidence = 0.95, stat.table = T)

# water negpw2 <- pairwise(LMneg, groups = class$Water);
# summary(negpw2, confidence = 0.95, stat.table = T)

# age negpw3 <- pairwise(LMneg, groups = class$Age);
# summary(negpw3, confidence = 0.95, stat.table = T)
```

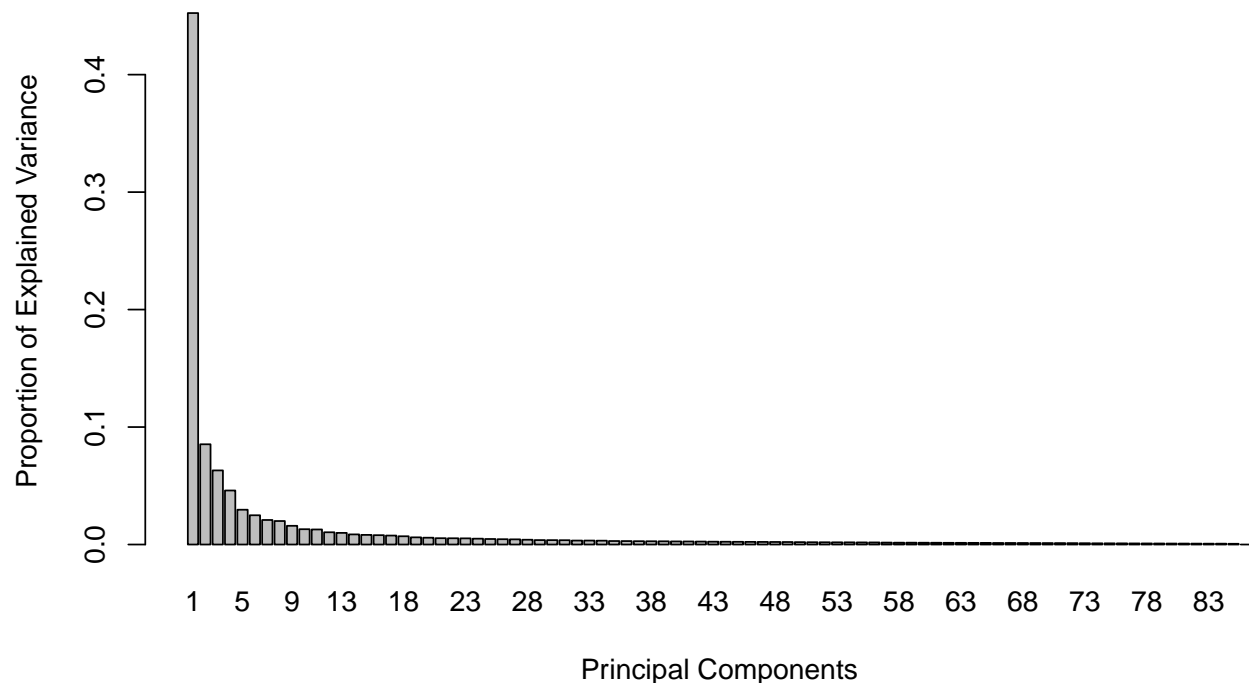
PCA

8. Identify the major source of variation in data and determine if the variation is sourced from experimental bias or biological conditions.

```
# tune how many components to use
tune.pca(scaled_Y)
```

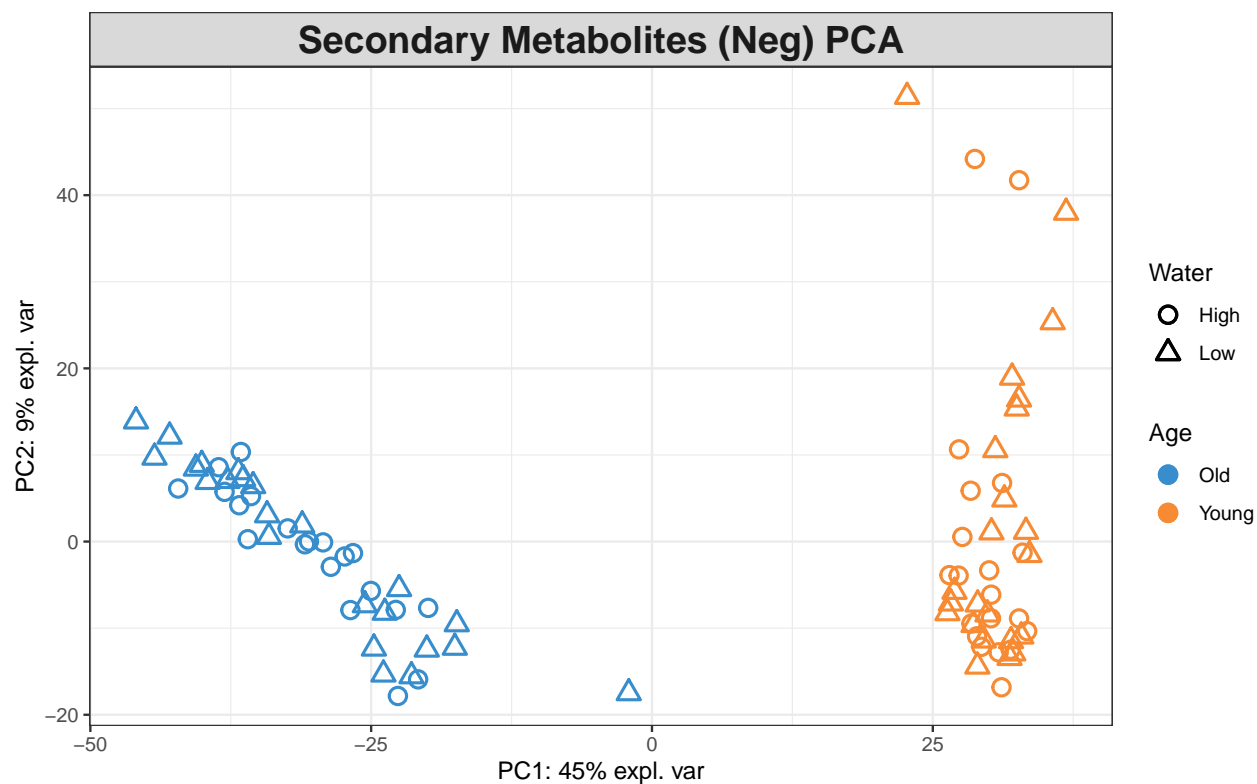
```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 976.77245 184.24131 136.20933 99.29918 63.94842 53.74830 45.11541 43.16815
##      PC9      PC10
## 34.39959 28.15873
##
## Proportion of explained variance for the first 10 principal components, see object$explained_variance
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
```

```
## 0.45241892 0.08533641 0.06308908 0.04599314 0.02961946 0.02489500 0.02089644
##          PC8          PC9          PC10
## 0.01999451 0.01593311 0.01304249
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##          PC1          PC2          PC3          PC4          PC5          PC6          PC7          PC8
## 0.4524189 0.5377553 0.6008444 0.6468376 0.6764570 0.7013520 0.7222484 0.7422430
##          PC9          PC10
## 0.7581761 0.7712186
##
## Other available components:
## -----
## loading vectors: see object$rotation
```



```
pca.res <- mixOmics::pca(scaled_Y, ncomp = 4, scale = F)

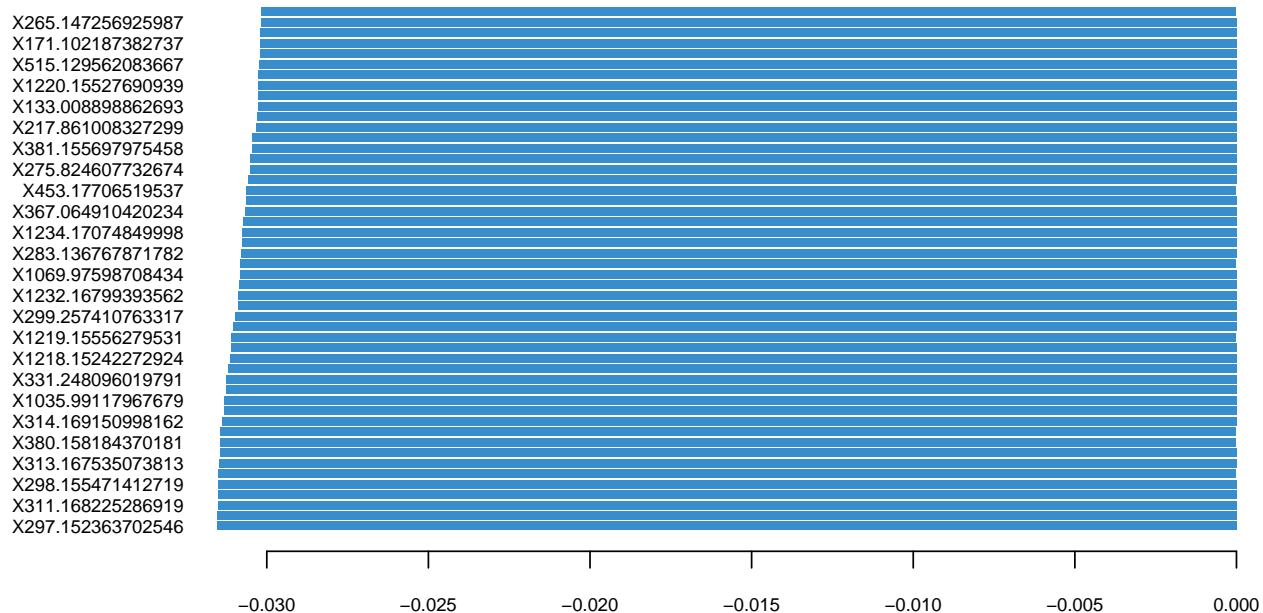
# plot pca
plotIndiv(pca.res, group = class$Age, ind.names = F, pch = as.factor(class$Water),
  legend = T, legend.title = "Age", legend.title.pch = "Water",
  title = "Secondary Metabolites (Neg) PCA")
```



*# Look at variable coefficients in each component with the
loading vectors The absolute value of loading vectors
represent the importance of each variable to define each PC*

```
plotLoadings(pca.res, ndisplay = 50)
```

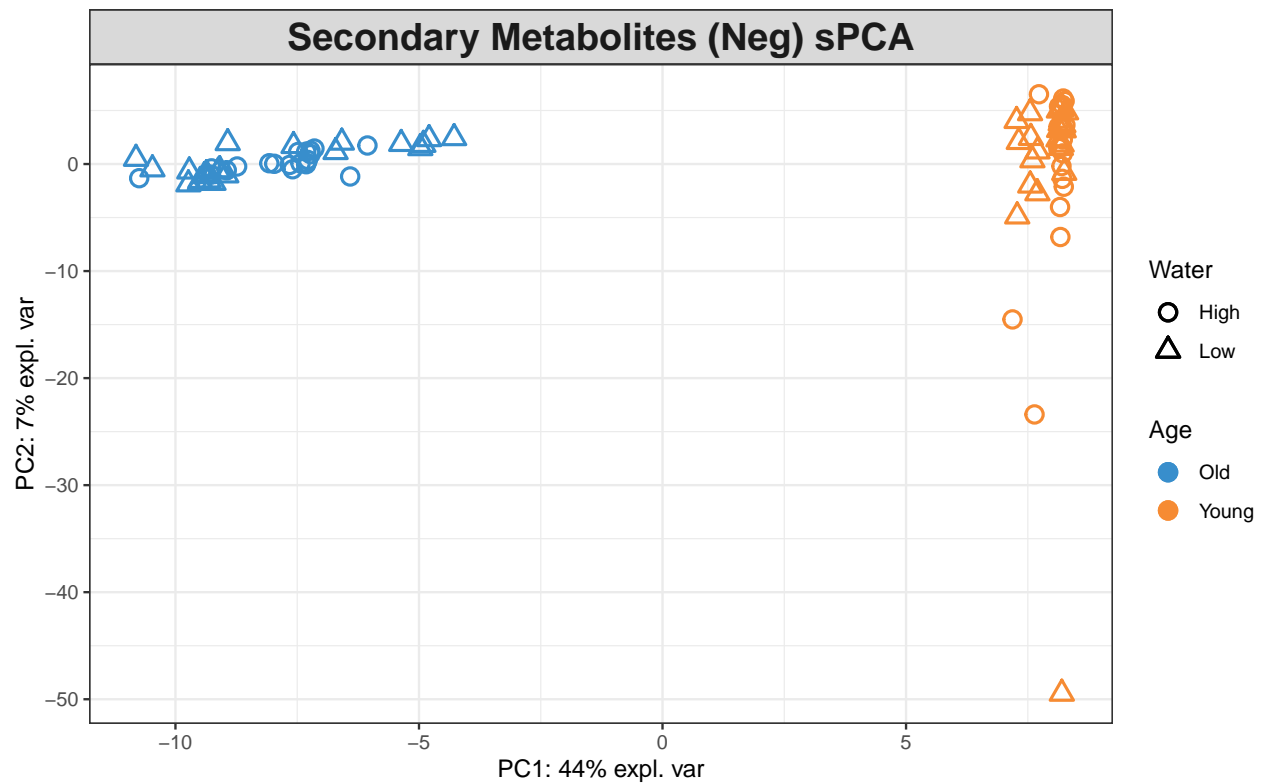
Loadings on comp 1



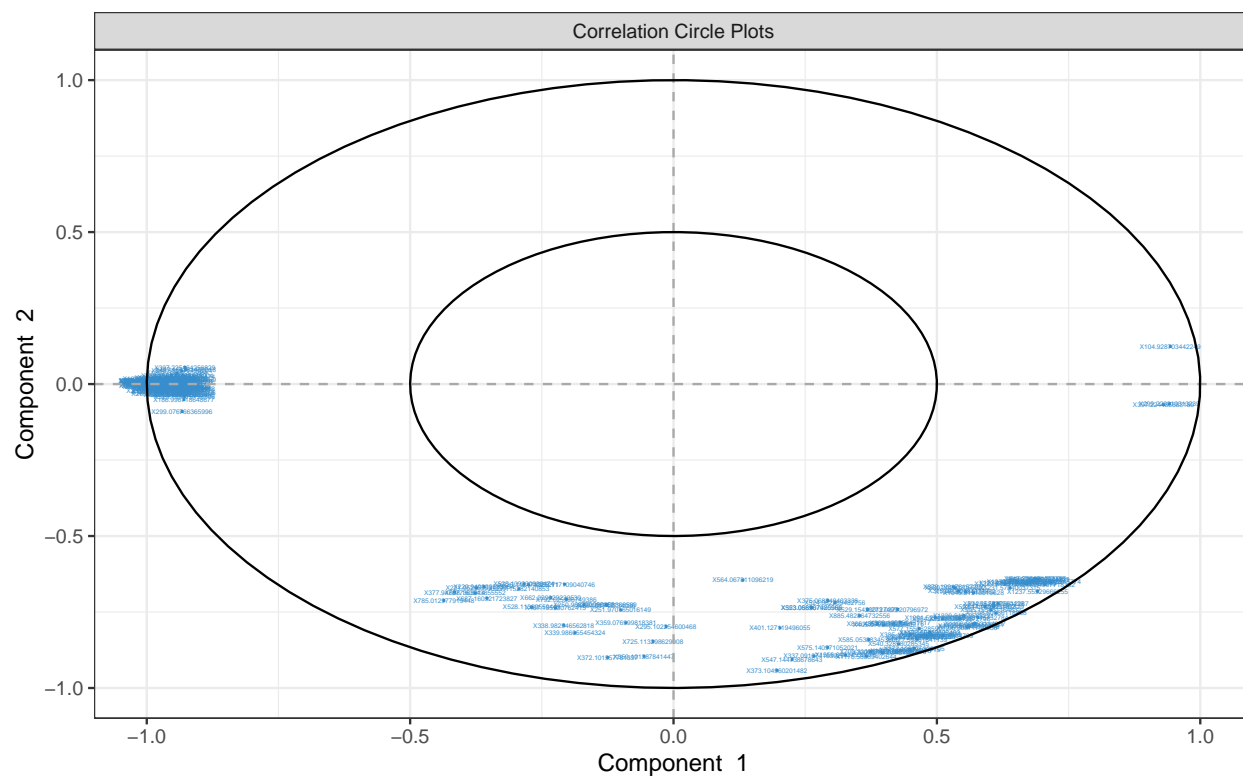
9. Identify the variables that contribute to the explained variance.

```
spca.res <- mixOmics::spca(scaled_Y, ncomp = 4, keepX = c(100,  
  100, 10, 10))
```

```
# plot spca  
plotIndiv(spca.res, group = class$Age, ind.names = F, pch = as.factor(class$Water),  
  legend = T, legend.title = "Age", legend.title.pch = "Water",  
  title = "Secondary Metabolites (Neg) sPCA")
```



```
# variables contributing to each component  
plotVar(spca.res, cex = 1)
```



```
selectVar(spca.res, comp = 1)$value # view loading value of each metabolite
```

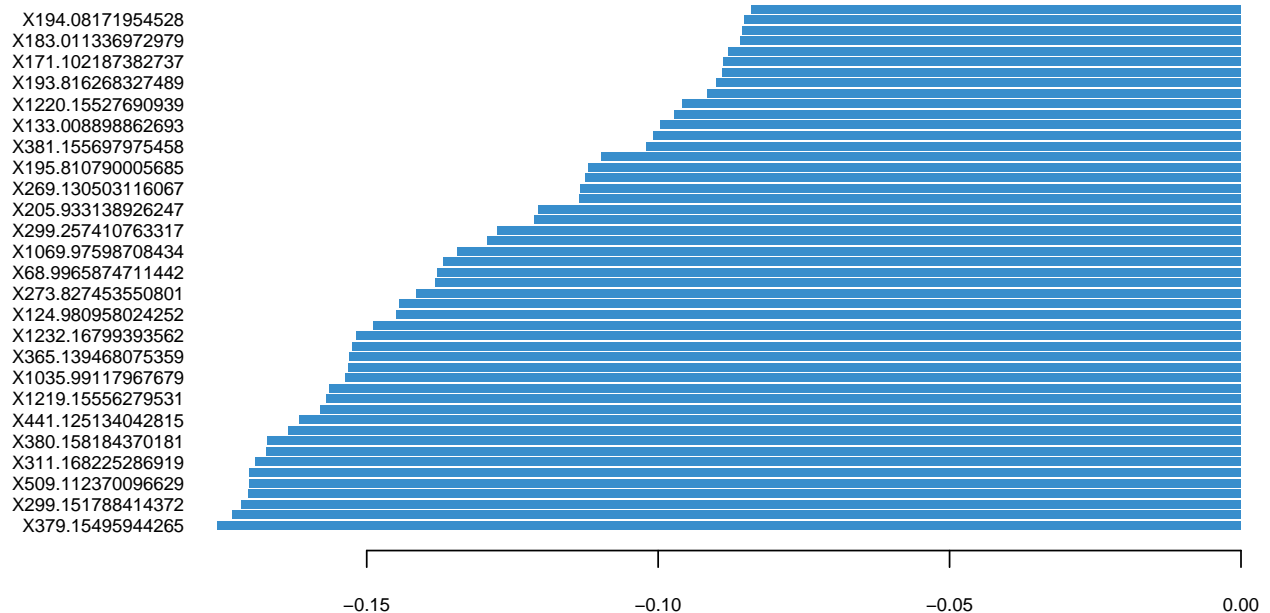
```
##          value.var
## X379.15495944265 -0.175584121
## X297.152363702546 -0.172954372
## X299.151788414372 -0.171422887
## X312.171051916693 -0.170335331
## X509.112370096629 -0.170143358
## X298.155471412719 -0.170141641
## X311.168225286919 -0.169096325
## X313.167535073813 -0.167140730
## X380.158184370181 -0.167045349
## X1218.15242272924 -0.163351126
## X441.125134042815 -0.161515239
## X447.142038786161 -0.157976013
## X1219.15556279531 -0.156828162
## X314.169150998162 -0.156336872
## X1035.99117967679 -0.153624853
## X331.248096019791 -0.153184660
## X365.139468075359 -0.152937389
## X1233.17083138513 -0.152366985
## X1232.16799393562 -0.151681361
## X577.10009366011 -0.148903956
## X124.980958024252 -0.144830654
## X1234.17074849998 -0.144411028
## X273.827453550801 -0.141459428
## X275.824607732674 -0.138156908
## X68.9965874711442 -0.137841643
## X283.136767871782 -0.136792978
```

X1069.97598708434 -0.134435299
 ## X236.095706999979 -0.129319158
 ## X299.257410763317 -0.127492571
 ## X181.837637931687 -0.121149101
 ## X205.933138926247 -0.120524226
 ## X199.804780742128 -0.113417893
 ## X269.130503116067 -0.113327192
 ## X197.807745662731 -0.112455201
 ## X195.810790005685 -0.111942699
 ## X366.142765108093 -0.109684271
 ## X381.155697975458 -0.102072082
 ## X309.172926332558 -0.100811612
 ## X133.008898862693 -0.099651698
 ## X183.835303476307 -0.097200469
 ## X1220.15527690939 -0.095766970
 ## X515.129562083667 -0.091557298
 ## X193.816268327489 -0.089965637
 ## X442.128728731669 -0.088960486
 ## X171.102187382737 -0.088783447
 ## X266.150532978834 -0.087963720
 ## X183.011336972979 -0.085846705
 ## X179.840983561524 -0.085565674
 ## X194.08171954528 -0.085258941
 ## X1235.17126272375 -0.083956128
 ## X265.147256925987 -0.083518993
 ## X453.17706519537 -0.080482900
 ## X353.199867493115 -0.075730520
 ## X220.026230094362 -0.074750830
 ## X521.163968352493 -0.074171322
 ## X966.265109809181 -0.067880255
 ## X243.1597702527 -0.067409469
 ## X217.861008327299 -0.067097569
 ## X279.163265503225 -0.067083794
 ## X433.12696860558 -0.063015216
 ## X333.134475258791 -0.060593092
 ## X267.149024018543 -0.060085738
 ## X437.815459144749 -0.058850180
 ## X427.110052721532 -0.048475900
 ## X332.251344160452 -0.047435383
 ## X1341.99375354789 -0.047178421
 ## X367.064910420234 -0.044579203
 ## X104.928703442249 0.043511953
 ## X310.176119528475 -0.043333691
 ## X1351.02587377764 -0.042135292
 ## X307.856405051197 -0.039662001
 ## X299.220310313239 0.039556484
 ## X293.123557536578 -0.038154914
 ## X209.079235584094 -0.037230434
 ## X291.883421740192 -0.036653132
 ## X238.973208443778 -0.032939939
 ## X1247.18600555383 -0.031276302
 ## X239.903661942424 -0.030684517
 ## X393.170435486951 -0.027809620
 ## X421.226481884033 -0.026270025

```
## X326.186529965011 -0.023339040
## X252.903088588244 -0.022607582
## X325.183670312083 -0.020591070
## X248.581571439204 -0.019700103
## X177.844946689888 -0.018897962
## X299.076766365996 -0.018880989
## X173.871421845004 -0.014424130
## X265.294664439591 -0.013382265
## X231.880543001532 -0.011711287
## X397.224485683189 0.011620209
## X186.996718648677 -0.009950133
## X215.128416576056 -0.008304907
## X338.966547006778 -0.007714851
## X327.183299114118 -0.007185985
## X323.133873135505 -0.006734269
## X1283.03860476474 -0.006575537
## X397.225164256829 -0.005020389
## X187.097170219206 -0.003173179
## X497.166361456018 -0.002142507
## X385.189523042069 -0.002069941
```

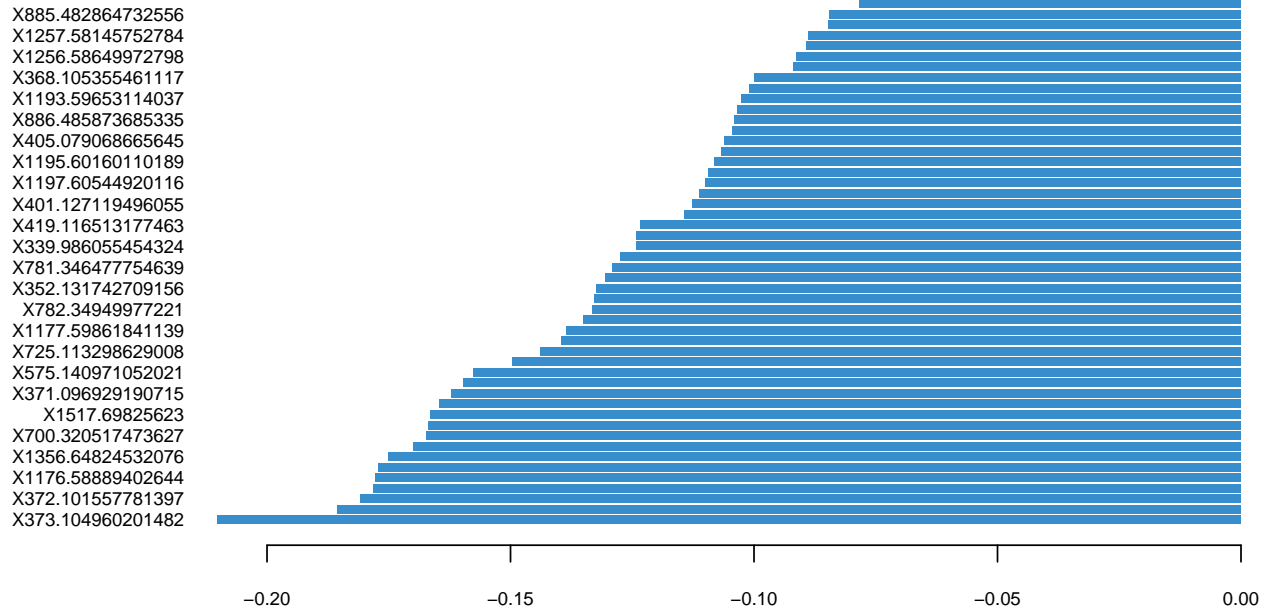
```
# plot loadings for comp 1
plotLoadings(spca.res, ndisplay = 50)
```

Loadings on comp 1



```
# plot loadings for comp 2
plotLoadings(spca.res, comp = 2, ndisplay = 50)
```


Loadings on comp 2



PLS-DA

10. Classify samples into known groups and predict the class of new samples.

```
neg.splsda <- mixOmics::splsda(scaled_Y, class$Water, keepX = c(100,
100))

# plot pls-da
plotIndiv(neg.splsda, ind.names = F, legend = T, title = "Secondary Metabolites (Neg) PLS-DA",
legend.title = "Water", ellipse = T)
```



```
selectVar(neg.splsda, comp = 1)
```

```
## $name
## [1] "X162.981984153263" "X515.153117414136" "X192.93799445282"
## [4] "X517.162152579604" "X193.066153993778" "X348.931231948595"
## [7] "X191.055213504967" "X535.176831218371" "X192.058516250057"
## [10] "X252.964590488188" "X534.174481872424" "X533.171528094349"
## [13] "X516.158708456826" "X536.179324673908" "X213.000641590061"
## [16] "X303.939715678049" "X464.890695937725" "X195.050697262523"
## [19] "X149.045299850651" "X392.105970805046" "X586.816085640511"
## [22] "X196.054408679814" "X273.960274921232" "X251.970165016149"
## [25] "X326.878160915464" "X491.159086547707" "X391.104121524513"
## [28] "X403.193108225713" "X668.039489283947" "X405.095791979711"
## [31] "X406.101086047129" "X250.968720098059" "X694.962550783936"
## [34] "X72.9969704954269" "X275.95896151189" "X113.991785192212"
## [37] "X126.881254250263" "X180.913218095842" "X165.04070789774"
## [40] "X448.915938372971" "X132.979090052561" "X112.987422804397"
## [43] "X226.904692611641" "X415.039120851625" "X505.171720581533"
## [46] "X272.959182648565" "X355.122532680517" "X286.860382544445"
## [49] "X249.012095790337" "X345.203279135189" "X601.006935338381"
## [52] "X96.961546643796" "X509.919155451559" "X156.908656713119"
## [55] "X465.89235737661" "X206.978989975056" "X154.997800813469"
## [58] "X180.06006506257" "X113.08454504931" "X181.070868536285"
## [61] "X302.94058504438" "X670.868854394574" "X166.933450883666"
## [64] "X176.965877570787" "X158.978329481195" "X115.990345226536"
## [67] "X159.979745780356" "X390.11079638633" "X264.987546353933"
## [70] "X164.936539559186" "X325.928047847518" "X137.899704774239"
## [73] "X274.957350769755" "X277.214068216649" "X117.018789896319"
## [76] "X234.982818862633" "X70.9998383237057" "X180.901485362489"
## [79] "X602.133917943731" "X118.94219501274" "X262.934923482667"
## [82] "X263.93454767186" "X154.973770121775" "X130.87917045935"
## [85] "X116.987131577406" "X163.059226885946" "X651.890947260194"
## [88] "X272.962737638412" "X321.090939374288" "X192.991127492295"
## [91] "X324.925259020932" "X648.882600063028" "X128.878230043894"
## [94] "X671.872797039619" "X429.123734786214" "X191.020390233799"
## [97] "X310.988352169764" "X173.009165502811" "X492.161356368014"
## [100] "X192.023938938643"
##
## $value
## value.var
## X162.981984153263 0.282598503
## X515.153117414136 -0.281984019
## X192.93799445282 0.275963290
## X517.162152579604 -0.210804106
## X193.066153993778 -0.199848487
## X348.931231948595 0.191612239
## X191.055213504967 -0.183435047
## X535.176831218371 -0.183169822
## X192.058516250057 -0.182299343
## X252.964590488188 0.177588103
## X534.174481872424 -0.175679237
## X533.171528094349 -0.175159785
## X516.158708456826 -0.174846898
## X536.179324673908 -0.167337490
```

```

## X213.000641590061 0.163850460
## X303.939715678049 0.156181336
## X464.890695937725 -0.151716910
## X195.050697262523 -0.140004679
## X149.045299850651 -0.134438914
## X392.105970805046 -0.123992449
## X586.816085640511 -0.116718321
## X196.054408679814 -0.113243138
## X273.960274921232 0.111349113
## X251.970165016149 0.110251267
## X326.878160915464 -0.109664462
## X491.159086547707 -0.109097963
## X391.104121524513 -0.105063448
## X403.193108225713 0.103198546
## X668.039489283947 0.099427452
## X405.095791979711 -0.098980854
## X406.101086047129 -0.096129459
## X250.968720098059 0.095580736
## X694.962550783936 0.086301928
## X72.9969704954269 0.082182523
## X275.95896151189 0.076402594
## X113.991785192212 -0.075986710
## X126.881254250263 -0.075011233
## X180.913218095842 0.066898325
## X165.04070789774 -0.066573198
## X448.915938372971 -0.066111226
## X132.979090052561 0.064169833
## X112.987422804397 -0.063551321
## X226.904692611641 0.062117857
## X415.039120851625 -0.061450645
## X505.171720581533 -0.060950506
## X272.959182648565 0.060514404
## X355.122532680517 -0.059852694
## X286.860382544445 0.059105871
## X249.012095790337 -0.057921105
## X345.203279135189 0.057600004
## X601.006935338381 -0.055973360
## X96.961546643796 0.054235174
## X509.919155451559 0.050184148
## X156.908656713119 0.049846815
## X465.89235737661 -0.047547829
## X206.978989975056 0.046824326
## X154.997800813469 0.046245125
## X180.06006506257 -0.042373884
## X113.08454504931 -0.040960923
## X181.070868536285 -0.039611546
## X302.94058504438 0.039500375
## X670.868854394574 -0.039011361
## X166.933450883666 0.037449332
## X176.965877570787 -0.036827352
## X158.978329481195 0.035950272
## X115.990345226536 0.035720002
## X159.979745780356 0.035397135
## X390.11079638633 -0.034827381

```

```

## X264.987546353933 -0.033095374
## X164.936539559186 0.033043394
## X325.928047847518 0.032671276
## X137.899704774239 -0.032619581
## X274.957350769755 0.029274343
## X277.214068216649 0.028773974
## X117.018789896319 0.028204548
## X234.982818862633 0.027639726
## X70.9998383237057 0.025430640
## X180.901485362489 -0.024715950
## X602.133917943731 0.024641400
## X118.94219501274 0.023396172
## X262.934923482667 0.022934236
## X263.93454767186 0.021425087
## X154.973770121775 0.018777724
## X130.87917045935 -0.018699235
## X116.987131577406 0.018066177
## X163.059226885946 -0.018036409
## X651.890947260194 -0.016205331
## X272.962737638412 -0.015895037
## X321.090939374288 0.013756236
## X192.991127492295 -0.013014148
## X324.925259020932 0.012547578
## X648.882600063028 -0.011818827
## X128.878230043894 -0.011691570
## X671.872797039619 -0.011544897
## X429.123734786214 -0.010740552
## X191.020390233799 0.010349771
## X310.988352169764 0.006407721
## X173.009165502811 0.005861959
## X492.161356368014 -0.005044446
## X192.023938938643 0.001344442
##
## $comp
## [1] 1

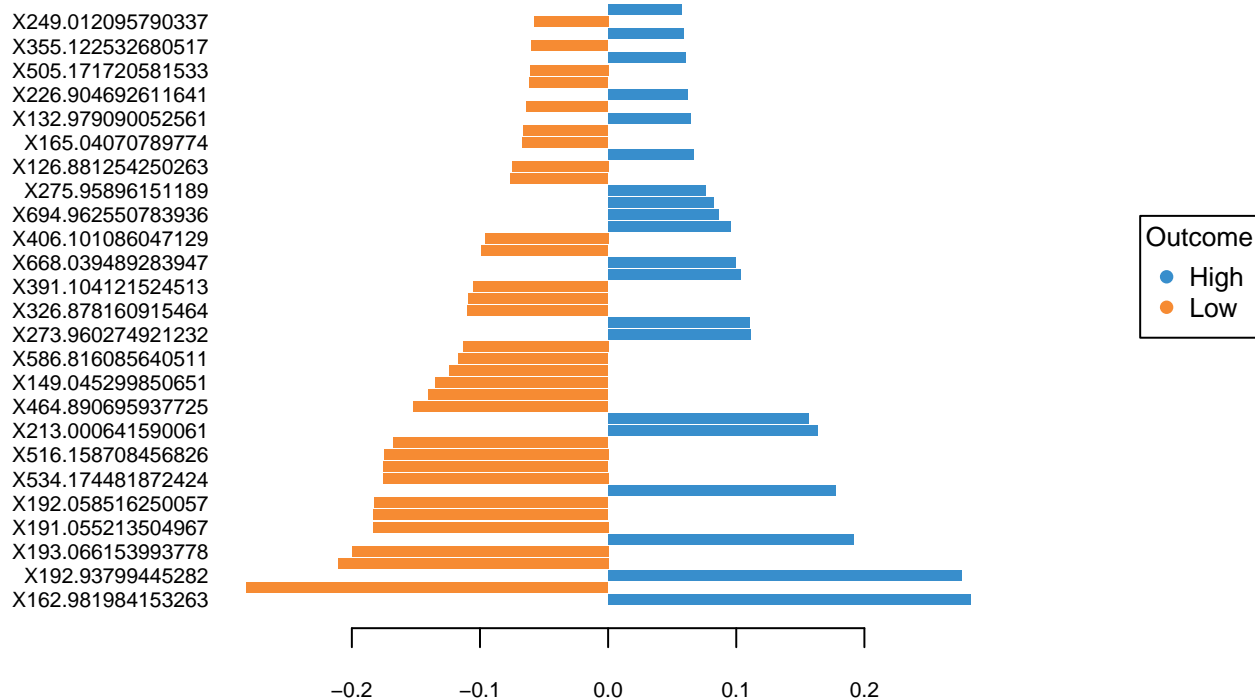
```

```

plotLoadings(neg.splsda, contrib = "max", method = "mean", ndisplay = 50)

```

Contribution on comp 1



Heatmaps of Averaged Data

11. Create averaged metabolite matrices and rerun PLS-DA to create a heatmap.

```
av_Y <- aggregate(Y, by = list(class$Water, class$Fungus, class$Age),
  FUN = "mean", simplify = T, data = class)
av.plsda <- mixOmics::plsda(av_Y[, 4:2159], av_Y$Group.2) # fungus

# heatmap
negcim <- cim(av.plsda, title = "Average Secondary Metabolites (neg)",
  col.names = F, xlab = "Secondary Metabolites", save = "png",
  name.save = "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/
```

Indicator Analysis

12. Identify indicator metabolites characteristic of each treatment using Dufrene-Legendre Indicator Analysis.

```
indicator_Age <- indval(Y, clustering = class$Age, numitr = 999,
  type = "long")
summary(indicator_Age)
```

```
##
## Sum of probabilities          = 76.6826826826827
##
## Sum of Indicator Values      = 1768.37
##
## Sum of Significant Indicator Values = 1674.91
##
## Number of Significant Indicators = 1982
```

```
##
## Significant Indicator Distribution
##
##      1      2
## 1345  637
```

13. Disect indval object.

```
relfrq <- indicator_Age$relfrq # relative frequency of species in classes
relabu <- indicator_Age$relabu # relative abundance of species in classes
indval <- indicator_Age$indval  # the indicator value for each species
maxcls <- data.frame(indicator_Age$maxcls) # the class each species has max indicator value for
indcls <- data.frame(indicator_Age$indcls) # the indicator value for each species to its max class
pval <- data.frame(indicator_Age$pval)  # the probability of obtaining as high an indicator value as ob
```

14. Export results to a csv file.

```
write.csv(cbind(relfrq, relabu, indval, maxcls, indcls, pval),
          "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Secondary Me
```

Secondary Metabolites (Pos)

RRPP

2. Define dependent variable matrix and class matrix.

```
path <- "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/XCMS Onl

SM_pos <- read_tsv(paste(path, "HvL_SM_pos/XCMS.annotated.diffreport..Low_F_SMpos_Kenia_Thesis.vs.High_L
  sep = ""))

# dependent variable: metabolite intensities
Y <- SM_pos[, c(9, 23:108)] %>% data.frame(row.names = 1) %>%
  t %>% data.frame()
scaled_Y <- scale(Y)

# class: sample factors
class <- read.csv(paste(path, "HvL_class.csv", sep = ""), header = T,
  row.names = 1)
```

3. Define and run multivariate regression models, then print out the results.

```
# LMpos <- lm.rrpp(scaled_Y ~ Block * Water * Fungus * Age,
# data = class, SS.type = 'III', print.progress = F);
# summary(LMpos)

# Error: vector memory exhausted (limit reached?) Error
# during wrapup: vector memory exhausted (limit reached?)
# Error: no more error handlers available (recursive
# errors?); invoking 'abort' restart
```

4. Examine RRPP plots to check for assumptions.

```
# residuals vs fitted values (homoscedasticity check)
# diagnostics <- plot(LMpos, type = 'diagnostics')
```

```
# pca plot pcplot <- plot(LMpos, type = 'PC', pch = 19, col =
# interaction(class$Water, class$Fungus))
```

5. Perform an RRPP ANOVA and print results.

```
## Old Leaves ANOVApos <- anova(LMpos, effect.type = 'F',
## error = c('Residuals', 'Block:Water', 'Block:Water:Fungus',
## 'Block:Water:Fungus:Age', 'Residuals',
## 'Block:Water:Fungus', 'Block:Water:Fungus',
## 'Block:Water:Fungus:Age', 'Block:Water:Fungus:Age',
## 'Block:Water:Fungus:Age', 'Residuals',
## 'Block:Water:Fungus:Age', 'Block:Water:Fungus:Age',
## 'Block:Water:Fungus:Age', 'Residuals')) ; summary(ANOVApos,
## formula = T)
```

6. Test lm.rrpp model coefficients. “d” is the amount of change in a variable for the coefficient indicated.

```
# poscoef <- coef(LMpos, test = T) ; summary(poscoef)
```

7. Test pairwise differences between least squares means. Similar to tukeyHSD function in the r stats package. The pairwise function will generate tables with confidence intervals and p-values for the pairwise statistic, Euclidean distance between least-squares means.

```
# fungus pospw <- pairwise(LMpos, groups = class$Fungus);
# summary(pospw, confidence = 0.95, stat.table = T)

# water pospw2 <- pairwise(LMpos, groups = class$Water);
# summary(pospw2, confidence = 0.95, stat.table = T)

# age pospw3 <- pairwise(LMpos, groups = class$Age);
# summary(pospw3, confidence = 0.95, stat.table = T)
```

PCA

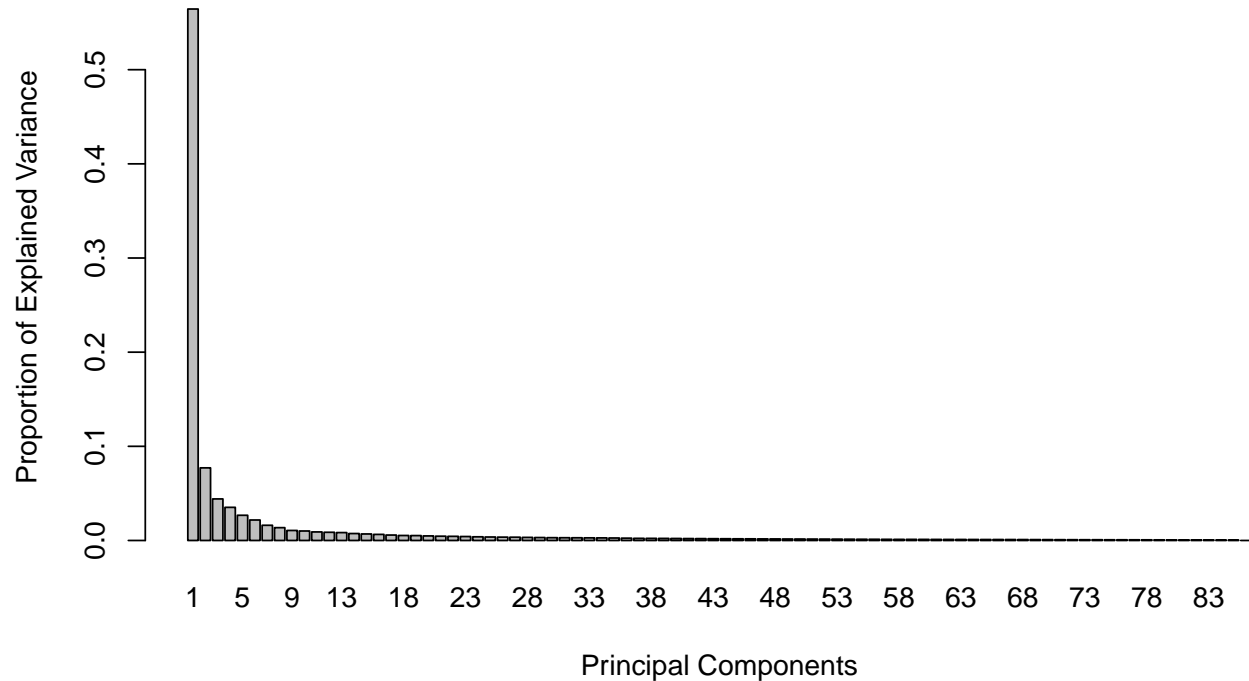
8. Identify the major source of variation in data and determine if the variation is sourced from experimental bias or biological conditions.

```
# tune how many components to use
tune.pca(scaled_Y)
```

```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 1917.43117 261.99579 149.90545 119.50430  90.92230  73.92355  54.81558
##      PC8      PC9     PC10
##  46.18308  36.09735  34.22262
##
## Proportion of explained variance for the first 10 principal components, see object$explained_variance:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 0.56444839 0.07712564 0.04412878 0.03517936 0.02676547 0.02176142 0.01613647
##      PC8      PC9     PC10
## 0.01359525 0.01062624 0.01007437
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 0.5644484 0.6415740 0.6857028 0.7208822 0.7476476 0.7694091 0.7855455 0.7991408
##      PC9     PC10
##
```

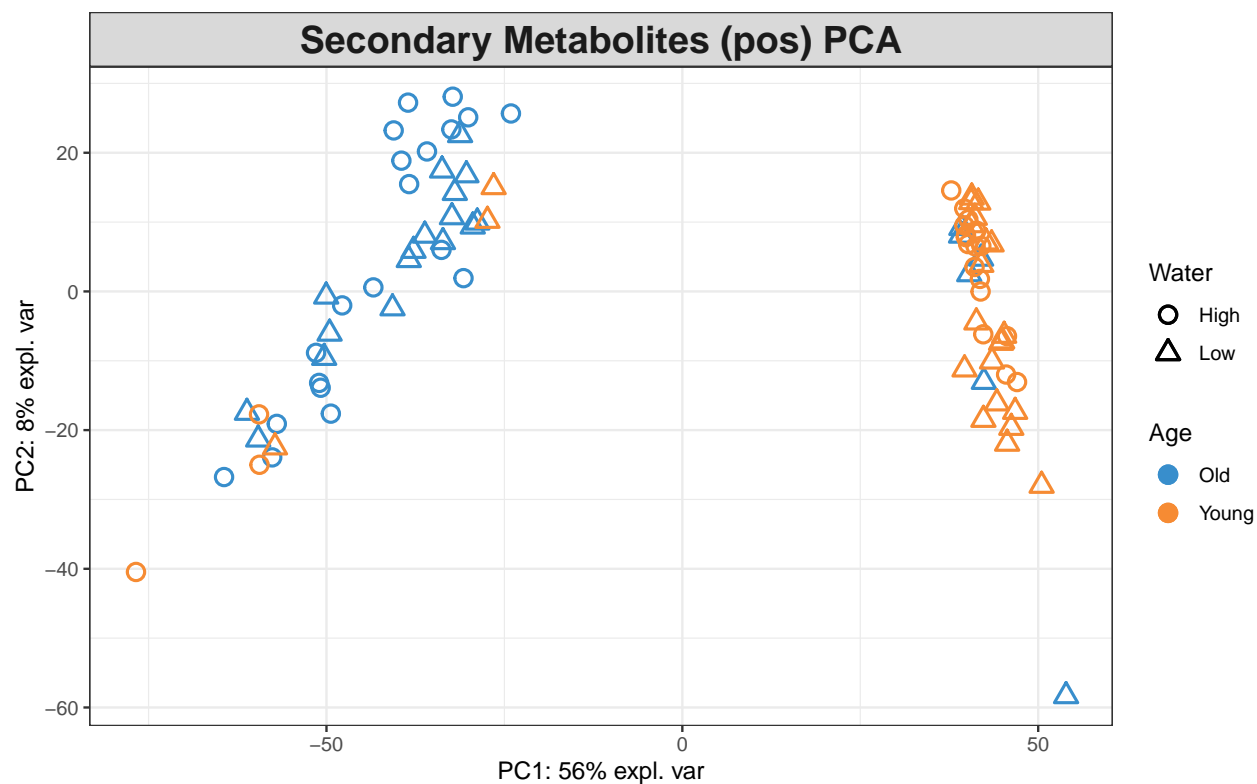


```
## 0.8097670 0.8198414
##
## Other available components:
## -----
## loading vectors: see object$rotation
```



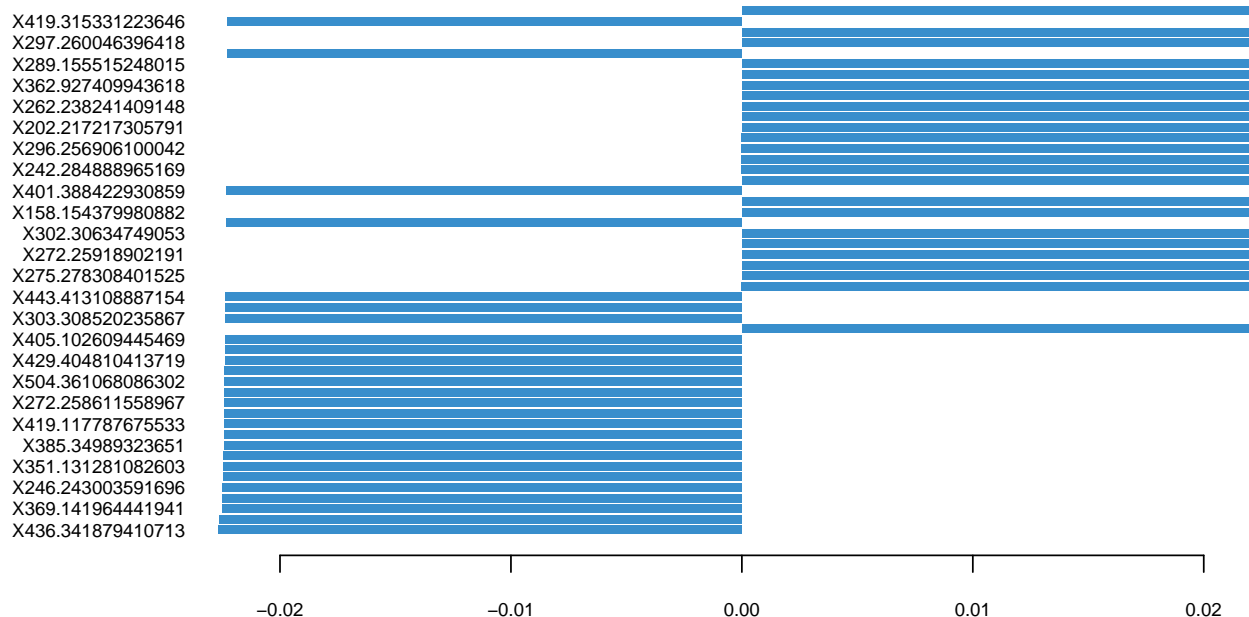
```
pca.res <- mixOmics::pca(scaled_Y, ncomp = 4, scale = F)

# plot pca
plotIndiv(pca.res, group = class$Age, ind.names = F, pch = as.factor(class$Water),
  legend = T, legend.title = "Age", legend.title.pch = "Water",
  title = "Secondary Metabolites (pos) PCA")
```



Look at variable coefficients in each component with the
 # loading vectors The absolute value of loading vectors
 # represent the importance of each variable to define each PC
 plotLoadings(pca.res, ndisplay = 50)

Loadings on comp 1

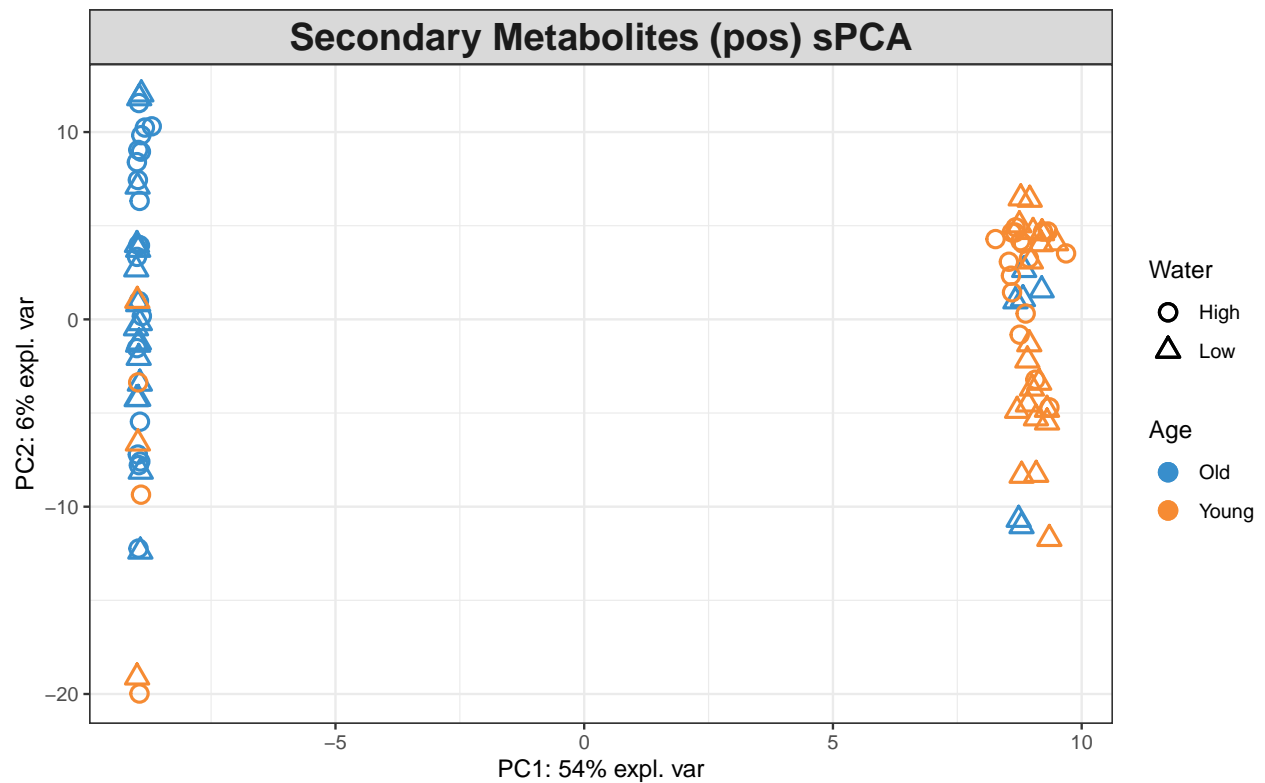


9. Identify the variables that contribute to the explained variance.

```
spca.res <- mixOmics::spca(scaled_Y, ncomp = 4, keepX = c(100,  
  100, 10, 10))
```

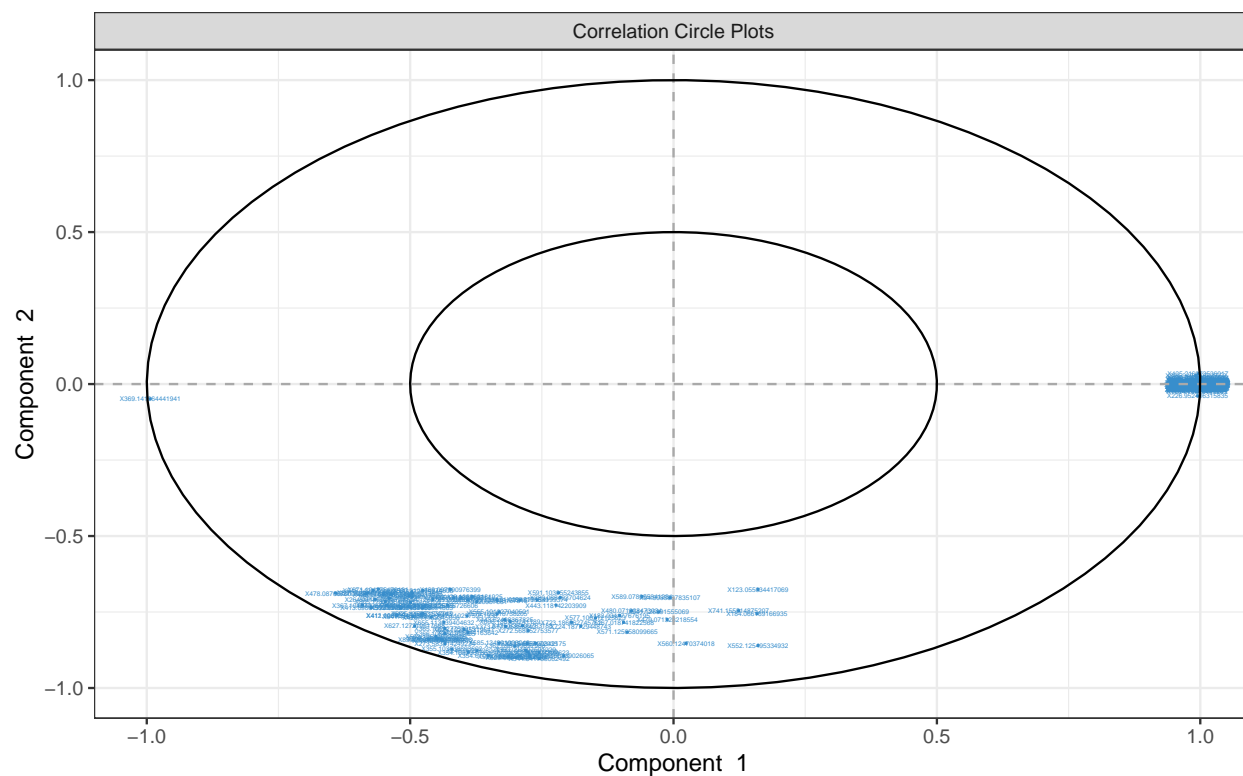
```
# plot spca
```

```
plotIndiv(spca.res, group = class$Age, ind.names = F, pch = as.factor(class$Water),  
  legend = T, legend.title = "Age", legend.title.pch = "Water",  
  title = "Secondary Metabolites (pos) sPCA")
```



```
# variables contributing to each component
```

```
plotVar(spca.res, cex = 1)
```



```
selectVar(spca.res, comp = 1)$value # view loading value of each metabolite
```

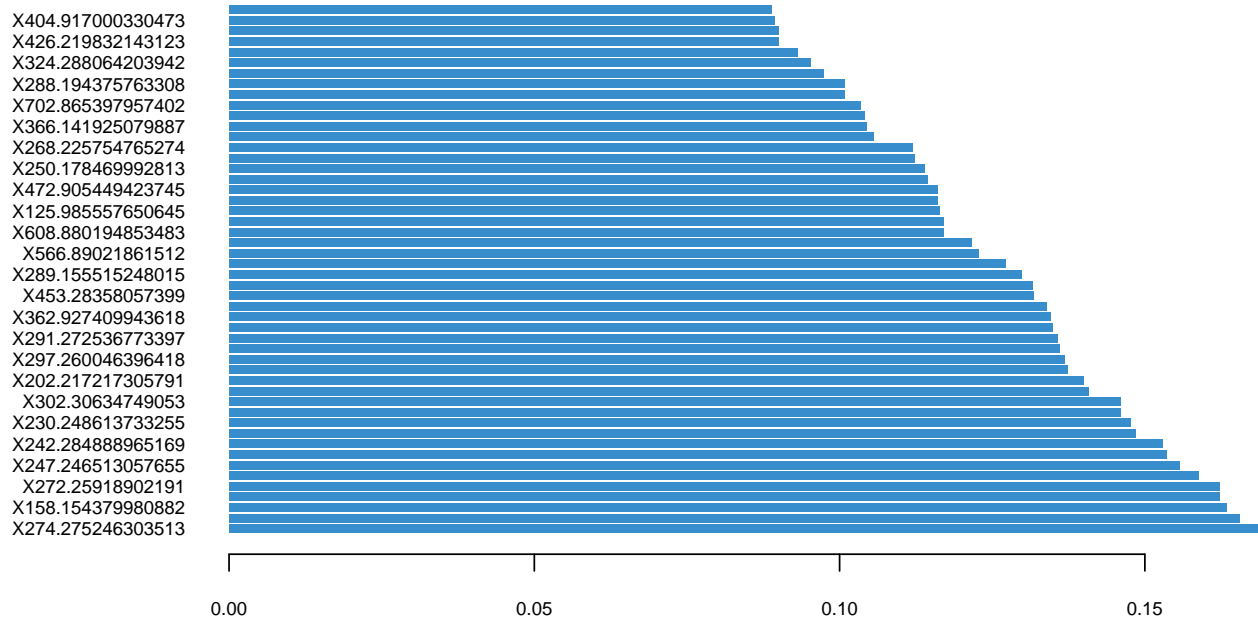
```
##          value.var
## X274.275246303513 0.168553055
## X275.278308401525 0.165484341
## X158.154379980882 0.163319429
## X290.269734404511 0.162236459
## X272.25918902191 0.162165775
## X246.243601996273 0.158825122
## X247.246513057655 0.155737391
## X296.256906100042 0.153480957
## X242.284888965169 0.152813869
## X218.212162552597 0.148436844
## X230.248613733255 0.147647402
## X276.280987233832 0.145976626
## X302.30634749053 0.145944261
## X427.247055494237 0.140808875
## X202.217217305791 0.139928404
## X498.902625620148 0.137281250
## X297.260046396418 0.136921145
## X430.914988645422 0.136088093
## X291.272536773397 0.135665688
## X159.157359102284 0.134804702
## X362.927409943618 0.134474260
## X744.857174471612 0.133948678
## X453.28358057399 0.131720274
## X303.309746146421 0.131542937
## X289.155515248015 0.129805027
## X676.868800816749 0.127184713
```

##	X566.89021861512	0.122751241
##	X282.279863362864	0.121643675
##	X608.880194853483	0.117104619
##	X337.106042793454	0.117043976
##	X125.985557650645	0.116318378
##	X634.877855286265	0.116121715
##	X472.905449423745	0.116020606
##	X365.137355849006	0.114358255
##	X250.178469992813	0.113958914
##	X262.238241409148	0.112229416
##	X268.225754765274	0.111982791
##	X338.110468810699	0.105576833
##	X366.141925079887	0.104366733
##	X540.892675003118	0.104147571
##	X702.865397957402	0.103370728
##	X425.216073536017	0.100867131
##	X288.194375763308	0.100789906
##	X383.205453620305	0.097454260
##	X324.288064203942	0.095320957
##	X948.821357709972	0.093087661
##	X426.219832143123	0.089989287
##	X812.845359317314	0.089977232
##	X404.917000330473	0.089327585
##	X770.85342413268	0.088800482
##	X226.952406315835	0.088562636
##	X330.337255335114	0.087745868
##	X326.379184544176	0.087510919
##	X180.136065127908	0.085739825
##	X412.258319346706	0.081955603
##	X244.227783214246	0.081358521
##	X880.833931496238	0.081152395
##	X1016.81008754217	0.079408177
##	X906.82871352652	0.077103134
##	X1152.78655587268	0.076694112
##	X1110.79200421023	0.076238375
##	X377.219706468154	0.074685624
##	X838.840910421205	0.074007696
##	X396.33280177295	0.070349163
##	X336.928556564172	0.069609274
##	X1042.80445534077	0.068674002
##	X231.251872185007	0.067861138
##	X1084.79829499205	0.067787099
##	X528.411402693818	0.067066966
##	X357.262059964962	0.065666501
##	X974.816805996354	0.063488154
##	X341.267839538756	0.062640567
##	X635.88205262261	0.062518123
##	X228.19658658527	0.062078898
##	X369.141964441941	-0.061732535
##	X312.193886813145	0.060809291
##	X1220.77503401223	0.060488396
##	X493.326755221831	0.055990120
##	X297.240923995837	0.052902766
##	X771.858731905389	0.051387133

```
## X703.870248267492 0.047789396
## X310.310808077241 0.047471016
## X1178.77977943495 0.047270834
## X498.400747376419 0.045783594
## X401.288302317161 0.041528629
## X454.373712455989 0.040730654
## X1314.75693731751 0.034938117
## X1356.75343152557 0.033876536
## X403.234673958751 0.029386789
## X243.28816065153 0.029210560
## X371.249139931553 0.028424443
## X439.351459615779 0.028072428
## X1288.76407846258 0.025336552
## X1246.76817339089 0.024252348
## X394.244340027895 0.023973629
## X567.894680313594 0.021553626
## X484.385198584392 0.014395295
## X329.218567906439 0.012375199
## X499.9070043412 0.009874591
## X375.204922913092 0.001601477
```

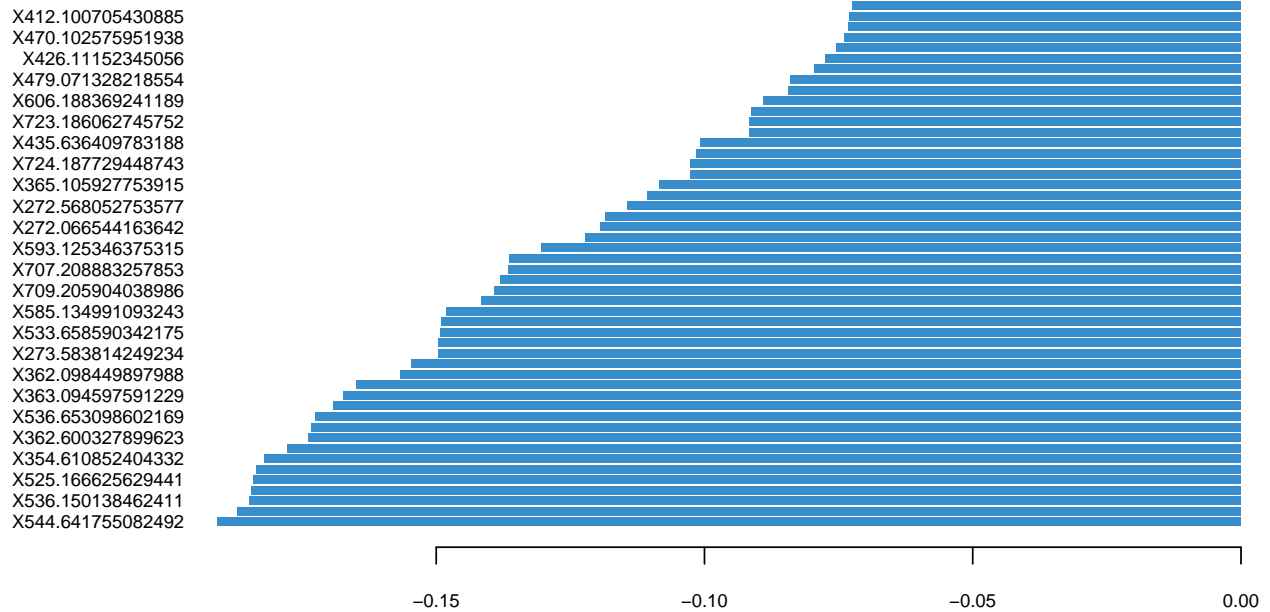
```
# plot loadings for comp 1
plotLoadings(spca.res, ndisplay = 50)
```

Loadings on comp 1



```
# plot loadings for comp 2
plotLoadings(spca.res, comp = 2, ndisplay = 50)
```

Loadings on comp 2



PLS-DA

10. Classify samples into known groups and predict the class of new samples.

```
pos.splsda <- mixOmics::splsda(scaled_Y, class$Water, keepX = c(100,
100))

# plot pls-da
plotIndiv(pos.splsda, ind.names = F, legend = T, title = "Secondary Metabolites (pos) PLS-DA",
  legend.title = "Water", ellipse = T)
```



```
selectVar(pos.splsda, comp = 1)
```

```
## $name
## [1] "X280.998615179239" "X94.9985887835107" "X309.037244931154"
## [4] "X277.019215188405" "X155.019395281618" "X661.469130354404"
## [7] "X167.018741822568" "X102.94429140562" "X137.008985820613"
## [10] "X453.228353499448" "X210.167136359035" "X465.037994623396"
## [13] "X368.88656616144" "X82.9984028305764" "X693.566495239479"
## [16] "X433.024646005709" "X653.463970760235" "X690.576030217547"
## [19] "X689.567980181359" "X113.132328590734" "X482.077154199374"
## [22] "X474.941923543397" "X504.240870169439" "X492.240818815669"
## [25] "X479.071328218554" "X127.123241740884" "X302.020552776489"
## [28] "X238.206866954128" "X686.540318414344" "X481.072591486914"
## [31] "X594.278897880979" "X228.898964277071" "X403.021112339946"
## [34] "X140.068498985117" "X717.554952131779" "X685.536139841416"
## [37] "X112.006606945286" "X531.260488652143" "X593.275090974113"
## [40] "X542.931752917604" "X739.547085504185" "X152.951256120806"
## [43] "X223.132302781562" "X196.159372726458" "X480.071838473034"
## [46] "X239.20853440653" "X293.061576006401" "X599.501124495797"
## [49] "X740.550634744712" "X147.119320310598" "X662.551874509279"
## [52] "X517.246265308622" "X743.578463224299" "X224.190711740745"
## [55] "X715.547662297667" "X635.541706480271" "X738.621195292574"
## [58] "X290.944131171082" "X364.83874458705" "X146.117609400659"
## [61] "X113.107701641596" "X646.474070941442" "X406.954559313587"
## [64] "X478.22613994532" "X661.540092854784" "X490.225522748138"
## [67] "X732.525717107642" "X731.522446726715" "X716.55084089848"
## [70] "X450.196501084222" "X882.871119366888" "X503.230110738367"
## [73] "X115.125082413555" "X755.521701526632" "X439.037270289756"
## [76] "X185.962621934939" "X129.141772434808" "X639.495637448484"
## [79] "X135.003444028557" "X439.215644570264" "X169.169232950259"
## [82] "X411.185776446045" "X184.961414116865" "X294.062793766388"
## [85] "X664.585958751482" "X238.929370638952" "X183.149912377648"
## [88] "X197.167494118394" "X568.562203068192" "X674.527992589059"
## [91] "X896.861270241689" "X270.979302898182" "X634.538579640345"
## [94] "X593.510771862241" "X183.964276243644" "X558.126801549664"
## [97] "X656.473054823286" "X814.883282789404" "X640.498767036228"
## [100] "X230.890247251421"
##
## $value
## value.var
## X280.998615179239 -0.321945788
## X94.9985887835107 0.296703108
## X309.037244931154 -0.289357538
## X277.019215188405 -0.266338686
## X155.019395281618 0.219588145
## X661.469130354404 -0.218973105
## X167.018741822568 0.202086719
## X102.94429140562 0.191843569
## X137.008985820613 0.189892205
## X453.228353499448 0.185955994
## X210.167136359035 0.174514548
## X465.037994623396 -0.173353328
## X368.88656616144 -0.151474590
## X82.9984028305764 0.150696961
```

X693.566495239479 -0.147264622
X433.024646005709 -0.146163666
X653.463970760235 -0.116676082
X690.576030217547 -0.116602539
X689.567980181359 -0.116047649
X113.132328590734 0.115624919
X482.077154199374 0.115309710
X474.941923543397 0.113274784
X504.240870169439 0.090909688
X492.240818815669 0.087031092
X479.071328218554 0.086627270
X127.123241740884 0.078373929
X302.020552776489 0.078276456
X238.206866954128 0.077457364
X686.540318414344 -0.076773763
X481.072591486914 0.075262718
X594.278897880979 -0.074903908
X228.898964277071 -0.072524731
X403.021112339946 -0.071721760
X140.068498985117 0.071080470
X717.554952131779 -0.070985952
X685.536139841416 -0.070531628
X112.006606945286 0.070026696
X531.260488652143 0.068180207
X593.275090974113 -0.067999048
X542.931752917604 0.067995258
X739.547085504185 -0.067458012
X152.951256120806 -0.065586135
X223.132302781562 0.062159174
X196.159372726458 0.060409486
X480.071838473034 0.057062951
X239.20853440653 0.055478237
X293.061576006401 -0.055175401
X599.501124495797 -0.054379873
X740.550634744712 -0.052672373
X147.119320310598 0.052423479
X662.551874509279 -0.052418976
X517.246265308622 0.049347217
X743.578463224299 -0.049143106
X224.190711740745 0.047688036
X715.547662297667 -0.045694086
X635.541706480271 -0.045366405
X738.621195292574 -0.043483499
X290.944131171082 -0.042805697
X364.83874458705 -0.042163140
X146.117609400659 0.041985650
X113.107701641596 0.040894653
X646.474070941442 -0.040408549
X406.954559313587 0.037619903
X478.22613994532 0.036566719
X661.540092854784 -0.034537320
X490.225522748138 0.034513652
X732.525717107642 -0.033327521
X731.522446726715 -0.032820416

```

## X716.55084089848 -0.032692096
## X450.196501084222 0.031154907
## X882.871119366888 0.028056996
## X503.230110738367 0.026582725
## X115.125082413555 0.025972119
## X755.521701526632 -0.025552649
## X439.037270289756 -0.025054948
## X185.962621934939 0.024679303
## X129.141772434808 0.024453016
## X639.495637448484 -0.023462969
## X135.003444028557 0.021559515
## X439.215644570264 0.019652716
## X169.169232950259 0.019379109
## X411.185776446045 0.018423257
## X184.961414116865 0.018041475
## X294.062793766388 -0.016483596
## X664.585958751482 -0.015638290
## X238.929370638952 -0.015480910
## X183.149912377648 0.011787922
## X197.167494118394 0.011241872
## X568.562203068192 -0.011116846
## X674.527992589059 -0.008730321
## X896.861270241689 0.006881297
## X270.979302898182 0.006478125
## X634.538579640345 -0.006453193
## X593.510771862241 -0.006141649
## X183.964276243644 0.005894103
## X558.126801549664 0.005345960
## X656.473054823286 -0.004332920
## X814.883282789404 0.003524209
## X640.498767036228 -0.003103064
## X230.890247251421 -0.001472436
##
## $comp
## [1] 1

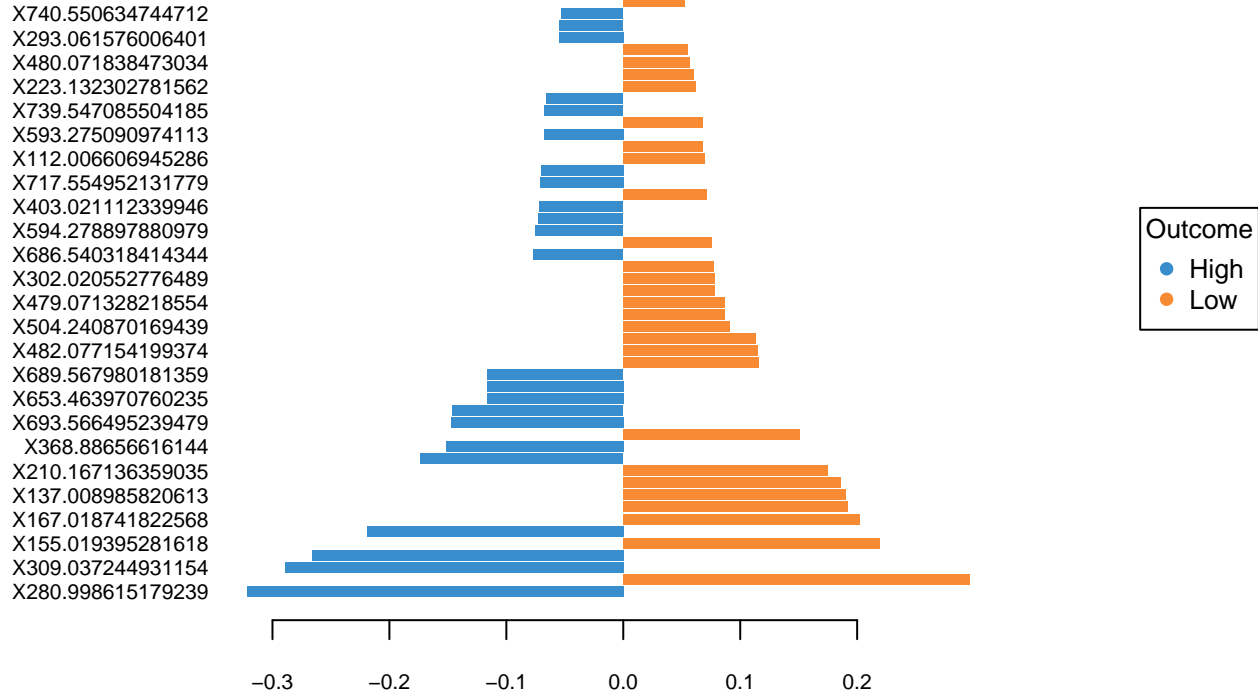
```

```

plotLoadings(pos.splsda, contrib = "max", method = "mean", ndisplay = 50)

```

Contribution on comp 1



Heatmaps of Averaged Data

11. Create averaged metabolite matrices and rerun PLS-DA to create a heatmap.

```
av_Y <- aggregate(Y, by = list(class$Water, class$Fungus, class$Age),
  FUN = "mean", simplify = T, data = class)
av.plsda <- mixOmics::plsda(av_Y[, 4:3397], av_Y$Group.2) # fungus

# heatmap
poscim <- cim(av.plsda, title = "Average Secondary Metabolites (pos)",
  col.names = F, xlab = "Secondary Metabolites", save = "png",
  name.save = "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/
```

Indicator Analysis

12. Identify indicator metabolites characteristic of each treatment using Dufrene-Legendre Indicator Analysis.

```
indicator_Age <- indval(Y, clustering = class$Age, numitr = 999,
  type = "long")
summary(indicator_Age)
```

```
##
## Sum of probabilities          = 148.271271271271
##
## Sum of Indicator Values      = 2455.37
##
## Sum of Significant Indicator Values = 2273.69
##
## Number of Significant Indicators = 3058
```

```
##
## Significant Indicator Distribution
##
##      1      2
## 1802 1256
```

13. Disect indval object.

```
relfrq <- indicator_Age$relfrq # relative frequency of species in classes
relabu <- indicator_Age$relabu # relative abundance of species in classes
indval <- indicator_Age$indval # the indicator value for each species
maxcls <- data.frame(indicator_Age$maxcls) # the class each species has max indicator value for
indcls <- data.frame(indicator_Age$indcls) # the indicator value for each species to its max class
pval <- data.frame(indicator_Age$pval) # the probability of obtaining as high an indicator value as ob
```

14. Export results to a csv file.

```
write.csv(cbind(relfrq, relabu, indval, maxcls, indcls, pval),
  "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Secondary Me
```

References

- Collyer, M.L., Adams, D.C. 2018. RRPP: An r package for fitting linear models to high-dimensional data using residual randomization. *Methods in Ecology and Evolution*. 9(7):1772-1779.
- Dufrene, M. and Legendre, P. 1997. Species assemblages and indicator species: the need for a flexible asymmetrical approach. *Ecol. Monogr.* 67(3):345-366.
- Rohart, F., Gautier, B., Singh, A., & Lê Cao, K. A. 2017. mixOmics: An R package for 'omics feature selection and multiple data integration. *PLoS computational biology*, 13(11):e1005752.