Field Old and Young Leaves Lipid Analysis

Kenia E. Segura Aba

55/317/2020

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 lipid 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)
```

Lipids (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 Onli

L_neg <- read_tsv(paste(path, "HvL_L_neg/XCMS.annotated.diffreport..High_F_Lneg_Kenia_Thesis.vs.Low_F_Ln
    sep = ""))

# dependent variable: metabolite intensities
Y <- L_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)
```

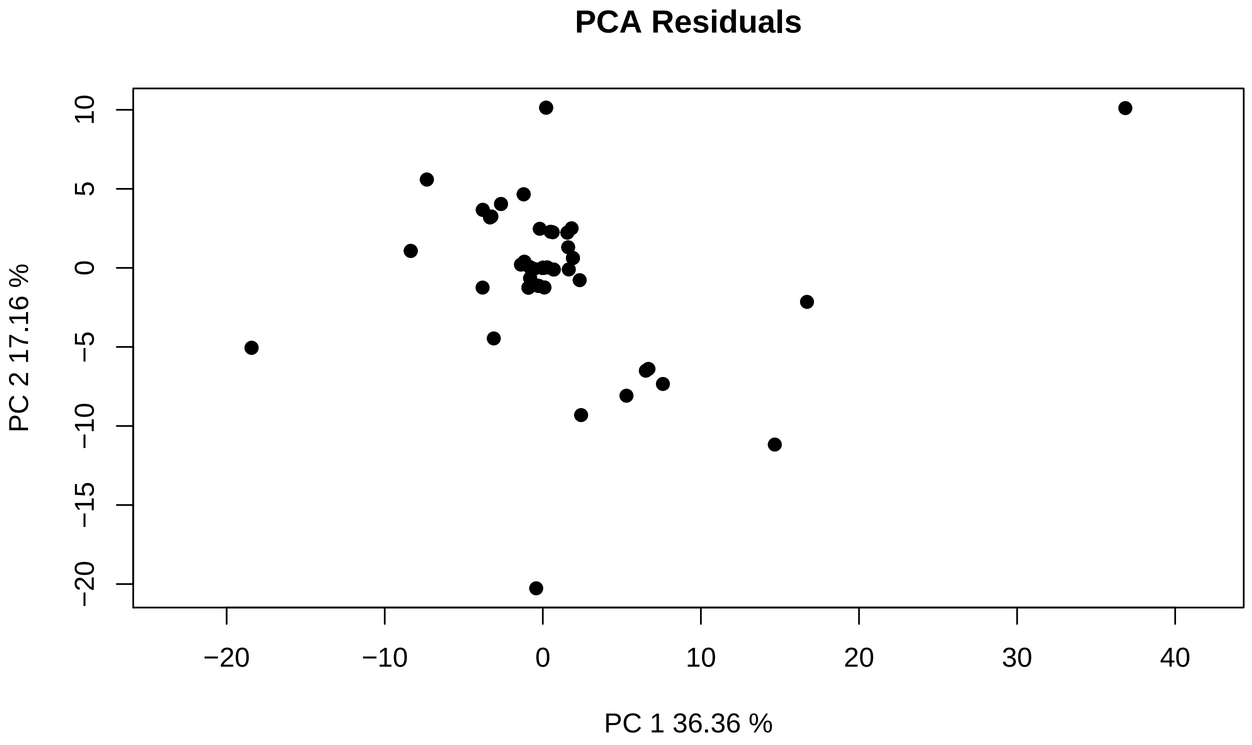
```
##
## Warning: Because variables in the linear model are redundant,
## the linear model design has been truncated (via QR decomposition).
## Original X columns: 72
## Final X columns (rank): 66
## Check coefficients or degrees of freedom in ANOVA to see changes.
```

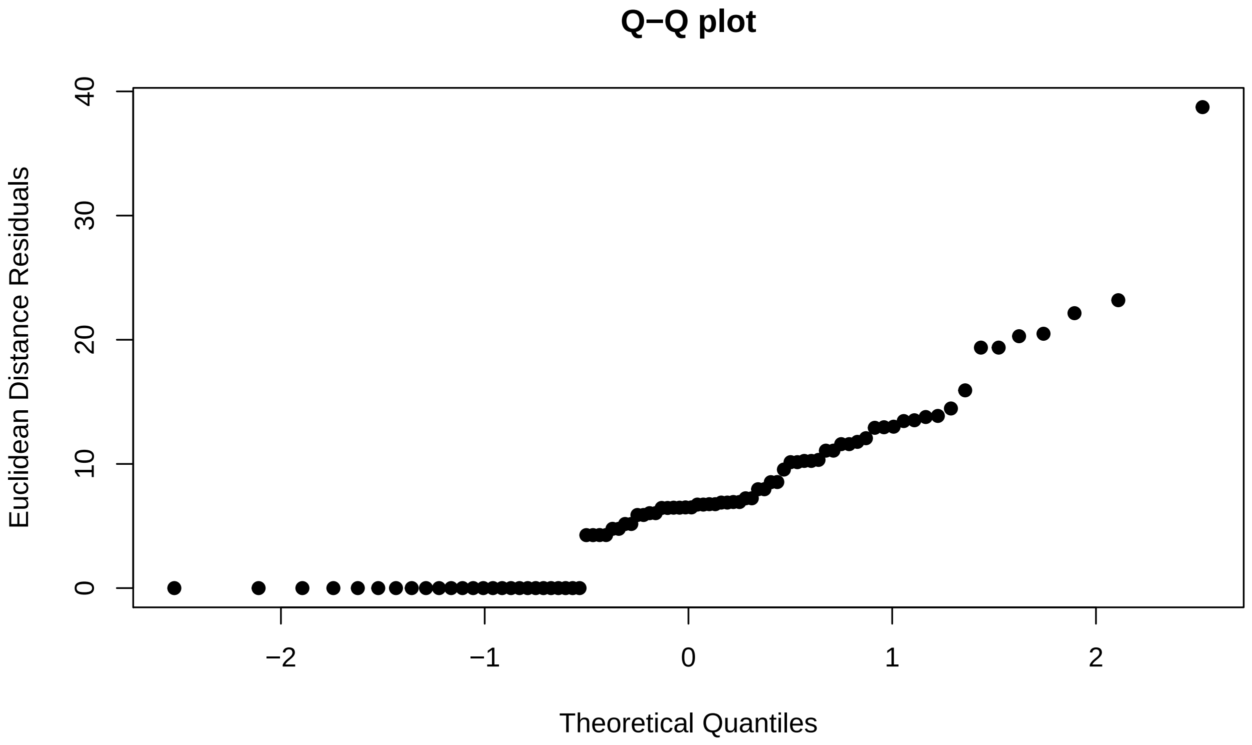
```r
summary(LMneg)
```

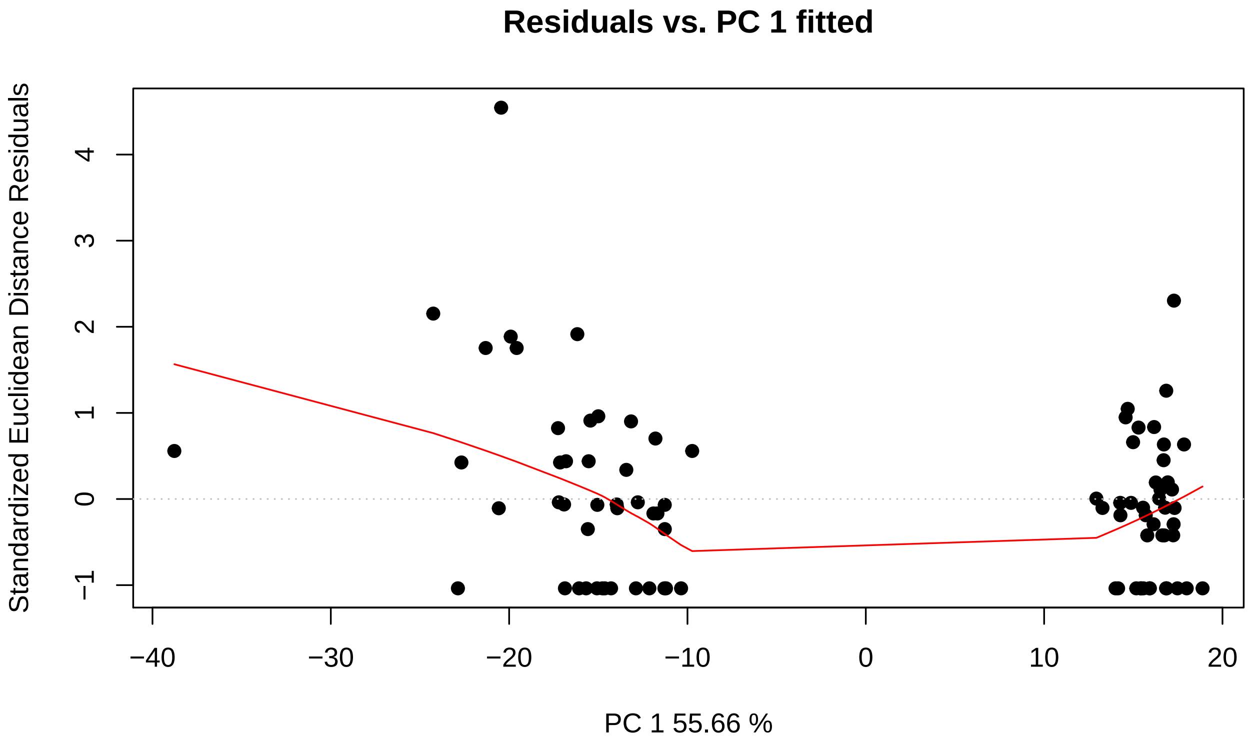
```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 86
## Number of dependent variables: 591
## Data space dimensions: 85
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Full Model Analysis of Variance
##
##                               Df Residual Df       SS Residual SS       Rsq
## Block * Water * Fungus * Age 65          20 41686.62     8548.38 0.8298322
##                                     F Z (from F)     Pr(>F)
## Block * Water * Fungus * Age 1.500477    2.19867 0.01553333
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##              Trace Proportion Rank
## Fitted    490.4308  0.8298321   65
## Residuals 100.5692  0.1701679   20
## Total     591.0000  1.0000000   85
##
## Eigenvalues
##
##                 PC1       PC2       PC3       PC4       PC5       PC6       PC7
## Fitted    272.96644  32.98964  32.09428  20.71673  13.40090  12.55719   9.24035
## Residuals  36.56861  17.26143  10.05691   9.19214   3.41508   3.24051   3.02027
## Total     282.62323  47.79102  39.35855  34.65218  21.78022  18.30859  14.85160
##                 PC8       PC9      PC10      PC11      PC12      PC13      PC14
## Fitted      6.73178   6.17656   5.09404   4.41788   4.28020   3.89574   3.63790
## Residuals   2.81642   2.35567   1.99117   1.60706   1.40543   1.34095   1.17502
## Total       8.80749   7.35894   6.54225   6.37401   5.85384   5.36106   4.91161
##                PC15      PC16      PC17      PC18      PC19      PC20      PC21
## Fitted      3.51825   3.08289   2.88967   2.78115   2.63025   2.40076   2.27338
## Residuals   0.97677   0.90909   0.87592   0.83967   0.77975   0.74129
## Total       4.23010   4.15957   3.63391   3.47032   3.31728   2.96695   2.78734
##                PC22      PC23      PC24      PC25      PC26      PC27      PC28
## Fitted      2.08592   2.04347   1.95052   1.81445   1.76152   1.69781   1.58094
## Residuals
## Total       2.59296   2.53096   2.37077   2.22498   2.14649   2.07248   2.01308
##                PC29      PC30      PC31      PC32      PC33      PC34      PC35
## Fitted      1.53268   1.46478   1.43931   1.41706   1.33730   1.30097   1.20824
## Residuals
## Total       1.82302   1.75515   1.72618   1.66669   1.60759   1.53718   1.49099
##                PC36      PC37      PC38      PC39      PC40      PC41      PC42
## Fitted      1.17131   1.11168   1.05680   1.01647   0.98765   0.96257   0.91497
## Residuals
## Total       1.46310   1.42742   1.40610   1.31282   1.25577   1.21513   1.17528
##                PC43      PC44      PC45      PC46      PC47      PC48      PC49
## Fitted      0.88240   0.83697   0.81446   0.76696   0.74862   0.72052   0.64946
## Residuals
## Total       1.10390   1.09029   1.04408   1.00797   0.99406   0.93520   0.91473
##                PC50      PC51      PC52      PC53      PC54      PC55      PC56
## Fitted      0.63414   0.62964   0.59512   0.58352   0.54423   0.51663   0.49232
## Residuals
## Total       0.88684   0.86982   0.81307   0.79798   0.74986   0.73195   0.72066
##                PC57      PC58      PC59      PC60      PC61      PC62      PC63
## Fitted      0.46503   0.43909   0.42943   0.42485   0.37047   0.34922   0.32522
## Residuals
## Total       0.70370   0.69824   0.65799   0.62653   0.61312   0.58532   0.56218
##                PC64      PC65      PC66      PC67      PC68      PC69      PC70
## Fitted      0.29472   0.28537
## Residuals
## Total       0.52254   0.51938   0.49426   0.48327   0.47219   0.44915   0.43946
##                PC71      PC72      PC73      PC74      PC75      PC76      PC77
## Fitted
## Residuals
## Total       0.43191   0.40660   0.39322   0.37963   0.35184   0.33592   0.31076
##                PC78      PC79      PC80      PC81      PC82      PC83      PC84
## Fitted
## Residuals
## Total       0.30790   0.29396   0.28222   0.26920   0.23012   0.20738   0.18308
##                PC85
## Fitted
## Residuals
## Total       0.17432
```

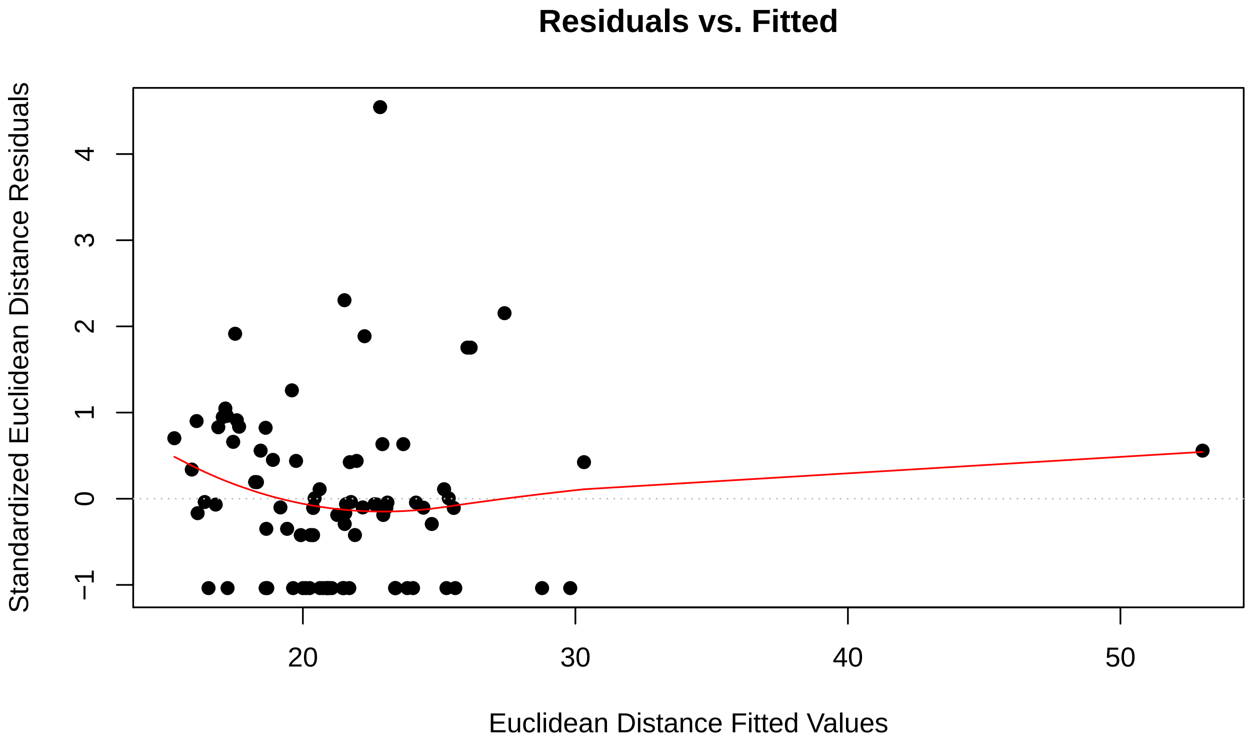
4. Examine RRPP plots to check for assumptions.

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

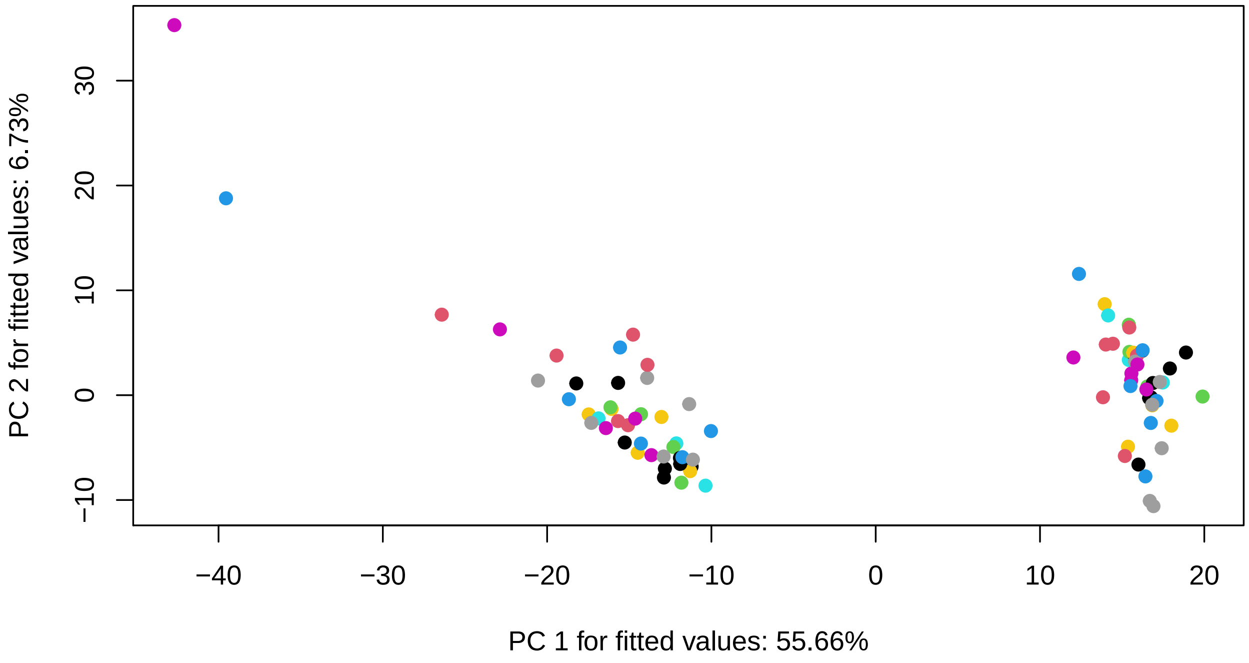

PCA Residuals
PC 2 17.16 %
PC 1 36.36 %
10
5
0
−5
−10
−15
−20
−20
−10
0
10
20
30
40


Q−Q plot
Euclidean Distance Residuals
Theoretical Quantiles
40
30
20
10
0
−2
−1
0
1
2





```
# 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)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 1000
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##                          Df   SS     MS      Rsq      F        Z Pr(>F)
## Block                     1  280 280.23 0.005578 0.6556            0.471
## Water                     1  392 391.81 0.007800 1.4547            0.219
## Fungus                    8 1473 184.14 0.029324 0.9635 -0.24130  0.604
## Age                       1  357 356.53 0.007097 2.0499  1.22991  0.085 .
## Block:Water               1  269 269.34 0.005362 0.6302            0.540
## Block:Fungus              8 1622 202.73 0.032285 1.0608 -0.02584  0.519
## Water:Fungus              8 1607 200.84 0.031983 1.0509 -0.11099  0.508
## Block:Age                 1  273 272.99 0.005434 1.5696            0.170
## Water:Age                 1  248 247.60 0.004929 1.4236  0.96100  0.165
## Fungus:Age                8 1519 189.86 0.030236 1.0917  0.05632  0.472
## Block:Water:Fungus        5  956 191.11 0.019022 0.4471 -1.60885  0.943
## Block:Water:Age           1  224 223.54 0.004450 1.2853  0.80988  0.214
## Block:Fungus:Age          8 1645 205.58 0.032739 1.1820  0.30980  0.375
## Water:Fungus:Age        8  1377 172.10 0.027407 0.9895 -0.29239  0.604
## Block:Water:Fungus:Age  5   870 173.92 0.017311 0.4069 -2.23316  0.993
## Residuals              20  8548 427.42 0.170168
## Total                  85 50235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = scaled_Y ~ Block * Water * Fungus * Age, SS.type = "III",
##     data = class, print.progress = F)
```

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)
```

```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 86
## Number of dependent variables: 591
## Data space dimensions: 85
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Statistics (distances) of coefficients with 95 percent confidence intervals,
## effect sizes, and probabilities of exceeding observed values based on
## 1000 random permutations using RRPP
##
##                                       d.obs UCL (95%)         Zd Pr(>d)
## (Intercept)                        24.19803  35.64365 -0.4729119  0.620
## Block                              11.83706  12.55310  1.5290872  0.067
## WaterLow                           42.76047  34.36113  2.5863806  0.022
## FungusCer                          23.38778  27.30614  1.0268871  0.137
## FungusCok                          31.69537  36.14552  1.4160457  0.091
## FungusCtrl                         26.79007  38.29414  0.5314773  0.226
## FungusNig                          32.26312  36.24134  1.1687741  0.105
## FungusPen                          23.01224  36.00832  0.1753623  0.353
## FungusPod                          36.68125  46.70474  0.8813344  0.153
## FungusPre                          24.53383  37.99423  0.3190067  0.306
## FungusXyl                          33.12923  49.16232  0.4968638  0.212
## AgeYoung                           40.78972  35.64402  2.1725801  0.028
## Block:WaterLow                     16.41173  16.61663  1.8511365  0.053
## Block:FungusCer                    13.39739  16.00505  1.0083769  0.132
## Block:FungusCok                    29.74544  27.01241  2.1426530  0.037
## Block:FungusCtrl                   14.16274  17.14137  0.9933859  0.130
## Block:FungusNig                    15.27122  18.14920  1.2472401  0.101
## Block:FungusPen                    12.31340  17.50949  0.6052180  0.228
## Block:FungusPod                    23.42378  26.16824  1.1699371  0.104
## Block:FungusPre                    13.14461  17.24867  0.7955078  0.177
## Block:FungusXyl                    22.11145  26.81095  0.9131044  0.148
## WaterLow:FungusCer                 36.21914  31.53127  2.5433040  0.026
## WaterLow:FungusCok                 29.48085  33.36796  1.4347404  0.074
## WaterLow:FungusCtrl                41.68765  50.92581  0.9916825  0.162
## WaterLow:FungusNig                 49.82589  52.54676  1.7045195  0.065
## WaterLow:FungusPen                 34.52427  49.92109  0.3206816  0.304
## WaterLow:FungusPod                 60.21651  57.80441  2.0562647  0.043
## WaterLow:FungusPre                 38.62012  51.59405  0.7116393  0.200
## WaterLow:FungusXyl                 28.34580  34.38400  1.2009618  0.102
## Block:AgeYoung                     16.52237  17.35928  1.6878180  0.060
## WaterLow:AgeYoung                  48.07208  51.04432  1.7625387  0.066
## FungusCer:AgeYoung                 33.70841  39.54882  1.1398509  0.124
## FungusCok:AgeYoung                 46.76176  47.42613  1.7611607  0.061
## FungusCtrl:AgeYoung                38.58544  50.39052  0.6893742  0.192
## FungusNig:AgeYoung                 39.55445  51.69111  0.8077519  0.164
## FungusPen:AgeYoung                 38.18884  50.00768  0.6894969  0.184
## FungusPod:AgeYoung                 59.47288  62.88714  1.5025910  0.071
## FungusPre:AgeYoung                 35.82642  53.59567  0.3462380  0.273
## FungusXyl:AgeYoung                 52.58623  64.41326  0.9675306  0.137
## Block:WaterLow:FungusCtrl          18.11765  22.32625  0.9326876  0.166
## Block:WaterLow:FungusNig           19.52787  22.29010  1.1925748  0.120
## Block:WaterLow:FungusPen           17.77583  22.56306  0.8116858  0.180
## Block:WaterLow:FungusPod           32.69674  29.51400  2.1172948  0.035
## Block:WaterLow:FungusPre           17.84318  22.72636  0.8600093  0.172
## Block:WaterLow:AgeYoung            21.14446  22.11875  1.5902806  0.073
## Block:FungusCer:AgeYoung           19.18678  22.93454  1.0053279  0.144
## Block:FungusCok:AgeYoung           38.73238  38.21217  1.9365927  0.048
## Block:FungusCtrl:AgeYoung          19.28688  23.30041  1.0302811  0.128
## Block:FungusNig:AgeYoung           21.24407  23.10024  1.4077812  0.080
## Block:FungusPen:AgeYoung           18.49880  23.19679  0.8965271  0.163
## Block:FungusPod:AgeYoung           36.49815  37.40606  1.7521275  0.059
## Block:FungusPre:AgeYoung           19.29507  23.27895  0.9590281  0.139
## Block:FungusXyl:AgeYoung           36.86060  38.18920  1.7916799  0.062
## WaterLow:FungusCer:AgeYoung        36.84415  41.98294  1.1594487  0.118
## WaterLow:FungusCok:AgeYoung        34.38588  45.56284  0.7260589  0.185
## WaterLow:FungusCtrl:AgeYoung       57.37355  68.34871  0.9808715  0.146
## WaterLow:FungusNig:AgeYoung        54.74143  69.12391  0.7781107  0.186
## WaterLow:FungusPen:AgeYoung        49.29366  68.17910  0.3865993  0.293
## WaterLow:FungusPod:AgeYoung        68.89462  78.50116  1.2566883  0.105
## WaterLow:FungusPre:AgeYoung        57.68937  68.09462  0.9808186  0.153
## WaterLow:FungusXyl:AgeYoung        32.20701  44.43867  0.5414562  0.233
## Block:WaterLow:FungusCtrl:AgeYoung 26.51413  30.61092  1.0966320  0.135
## Block:WaterLow:FungusNig:AgeYoung  25.38790  31.61976  0.9248373  0.152
## Block:WaterLow:FungusPen:AgeYoung  23.98300  30.92408  0.7008352  0.201
## Block:WaterLow:FungusPod:AgeYoung  41.19461  41.19735  1.8151912  0.051
## Block:WaterLow:FungusPre:AgeYoung  26.62948  30.97384  1.1626014  0.117
```

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)
```

```
##
## Pairwise comparisons
##
## Groups: Asp Cer Cok Ctrl Nig Pen Pod Pre Xyl
##
## RRPP: 1000 permutations
```

```
##
## LS means:
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##                  d UCL (95%)          Z Pr > d
## Asp:Cer   5.984787  9.260901 -1.8131690  0.999
## Asp:Cok   7.301032 11.111050 -1.5003886  0.988
## Asp:Ctrl  5.910522  8.629902 -1.8761111  0.991
## Asp:Nig   5.248888  8.733849 -1.5563848  0.988
## Asp:Pen   5.568655  8.710074 -1.5673828  0.985
## Asp:Pod   6.748146 10.173440 -1.0796062  0.878
## Asp:Pre   8.356723 12.082919 -0.6817305  0.743
## Asp:Xyl   8.039612 11.427463 -1.3884105  0.964
## Cer:Cok   7.335894 11.424340 -1.6652563  0.993
## Cer:Ctrl  6.568392  9.727435 -1.7774701  0.998
## Cer:Nig   5.752789  9.639389 -1.6513380  0.999
## Cer:Pen   6.324548  9.683988 -1.5802745  0.970
## Cer:Pod   6.400837 10.604180 -1.2505532  0.935
## Cer:Pre   8.573508 12.738086 -0.7676696  0.772
## Cer:Xyl   9.013375 12.788912 -1.2995811  0.937
## Cok:Ctrl  6.566911 10.708088 -1.6500633  0.996
## Cok:Nig   6.986185 11.387091 -1.3255510  0.990
## Cok:Pen   8.147220 11.859747 -1.2745158  0.945
## Cok:Pod   7.537427 11.616834 -1.4069033  0.962
## Cok:Pre  10.371201 15.227821 -0.6471046  0.738
## Cok:Xyl   9.095675 13.643680 -1.1584432  0.906
## Ctrl:Nig  5.634317  8.990223 -1.6018585  0.993
## Ctrl:Pen  6.605250  9.452790 -1.4033784  0.934
## Ctrl:Pod  7.586948 10.963004 -1.0249800  0.850
## Ctrl:Pre  8.160042 11.961480 -0.7114438  0.741
## Ctrl:Xyl  7.820879 11.457236 -1.3409580  0.955
## Nig:Pen   6.189182  9.536120 -1.3841169  0.949
## Nig:Pod   6.930353 10.921086 -0.9394695  0.846
## Nig:Pre   7.158988 11.264457 -0.9146748  0.829
## Nig:Xyl   8.345631 11.987543 -1.2924942  0.943
## Pen:Pod   7.030939 10.329655 -1.0903092  0.874
## Pen:Pre   7.878363 11.765339 -0.6573757  0.726
## Pen:Xyl   7.766885 11.304162 -1.4324216  0.962
## Pod:Pre  10.692690 14.908026 -0.4290057  0.649
## Pod:Xyl  10.507013 14.380316 -0.8095238  0.774
## Pre:Xyl   9.442164 13.752219 -0.9121541  0.863
```

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

```
##
## Pairwise comparisons
##
## Groups: High Low
##
## RRPP: 1000 permutations
##
## LS means:
```

```
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##                  d UCL (95%)         Z Pr > d
## High:Low 6.046111  7.472368 -0.661193   0.73
```

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

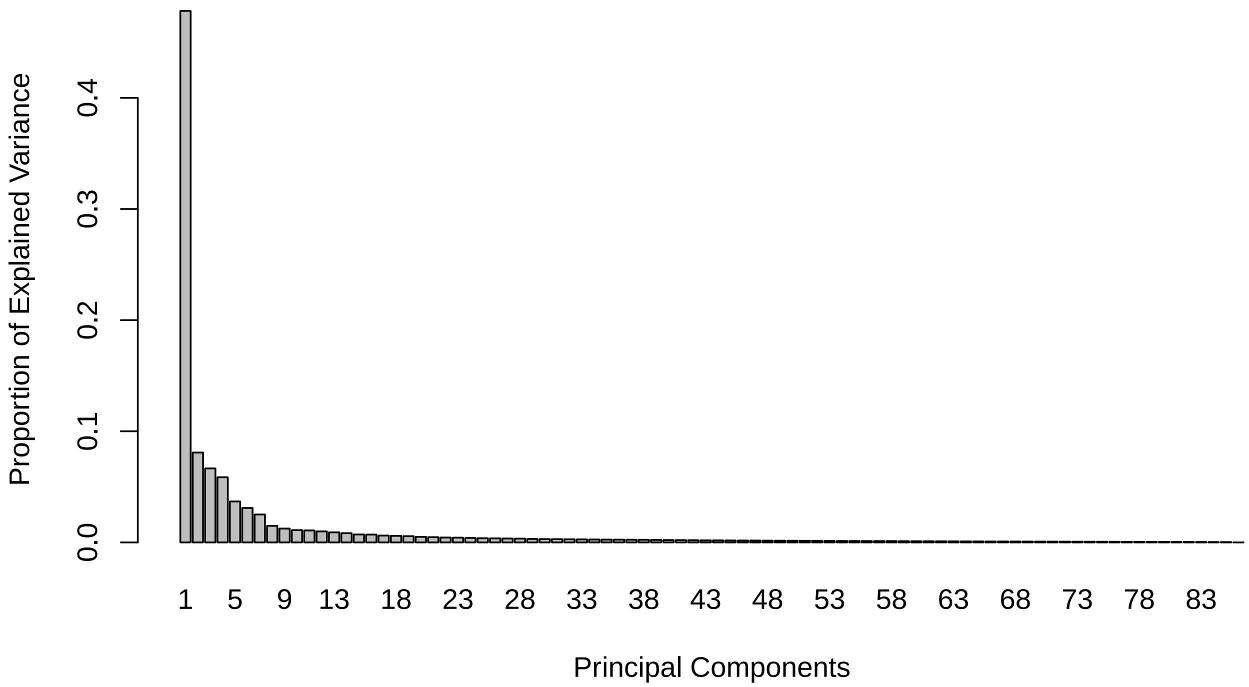
```
##
## Pairwise comparisons
##
## Groups: Old Young
##
## RRPP: 1000 permutations
##
## LS means:
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##                   d UCL (95%)          Z Pr > d
## Old:Young 32.11768  33.13689 -0.1068226   0.53
```

PCA

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

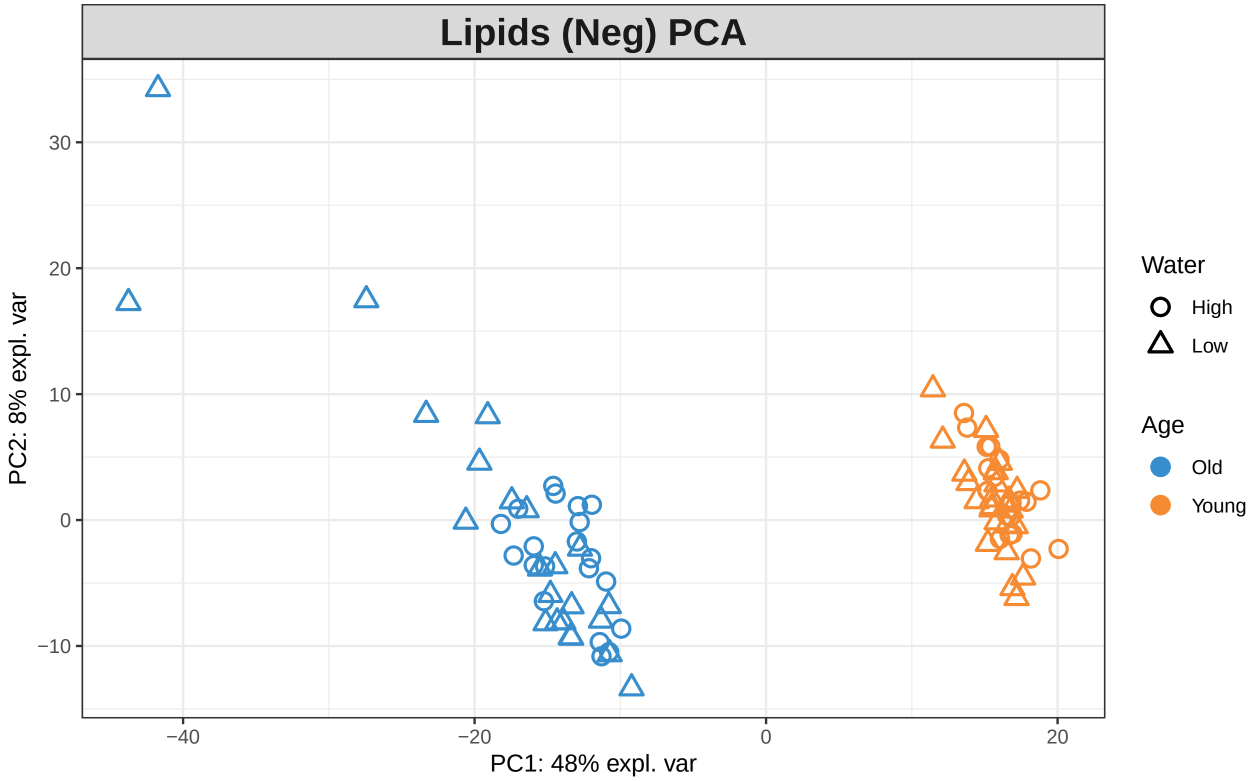
```
# tune how many components to use
tune.pca(scaled_Y)
```

```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 282.623227  47.791020  39.358547  34.652176  21.780215  18.308588  14.851605
##        PC8        PC9       PC10
##   8.807490   7.358935   6.542251
##
## Proportion of explained variance for the first 10 principal components, see object$explained_varianc
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 0.47821189 0.08086467 0.06659653 0.05863312 0.03685316 0.03097900 0.02512962
##        PC8        PC9       PC10
## 0.01490269 0.01245167 0.01106980
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 0.4782119 0.5590766 0.6256731 0.6843062 0.7211594 0.7521384 0.7772680 0.7921707
##       PC9      PC10
## 0.8046223 0.8156921
##
##  Other available components:
##  --------------------
##  loading vectors: see object$rotation
```

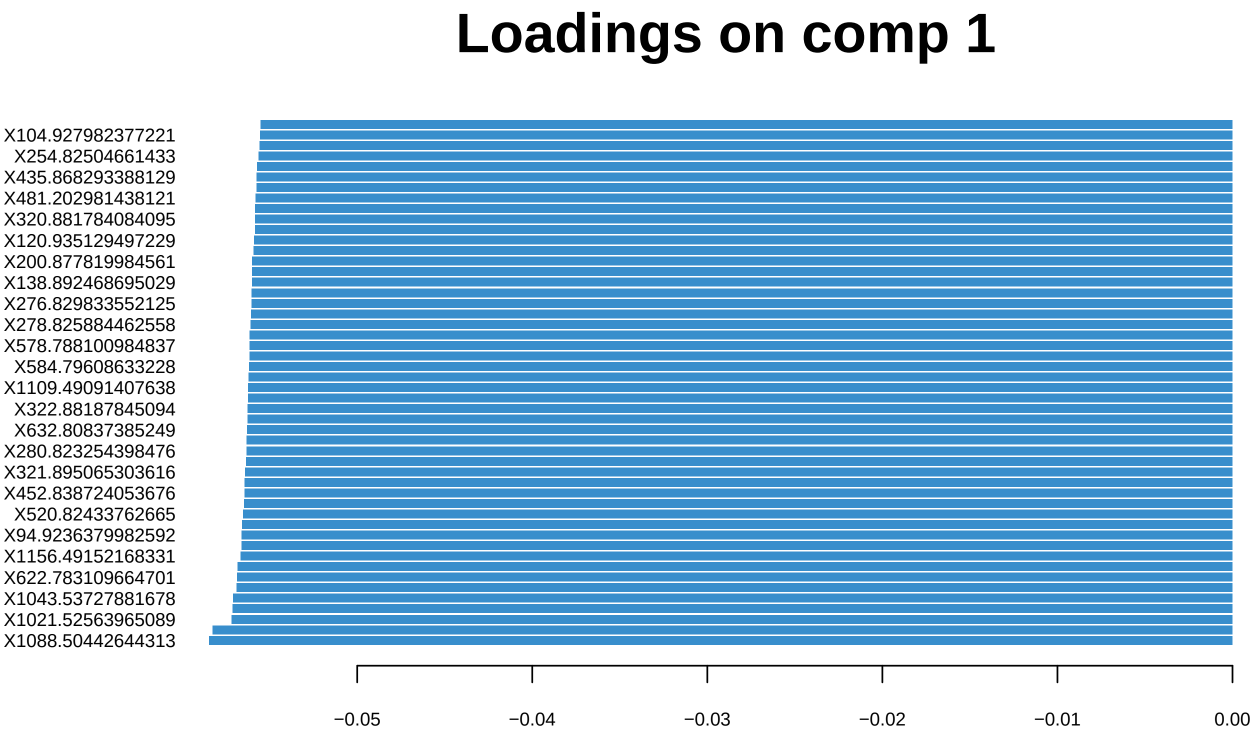


```r
pca.res <- mixOmics::pca(scaled_Y, ncomp = 4, scale = F)

# 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 = "Lipids (Neg) PCA")
```



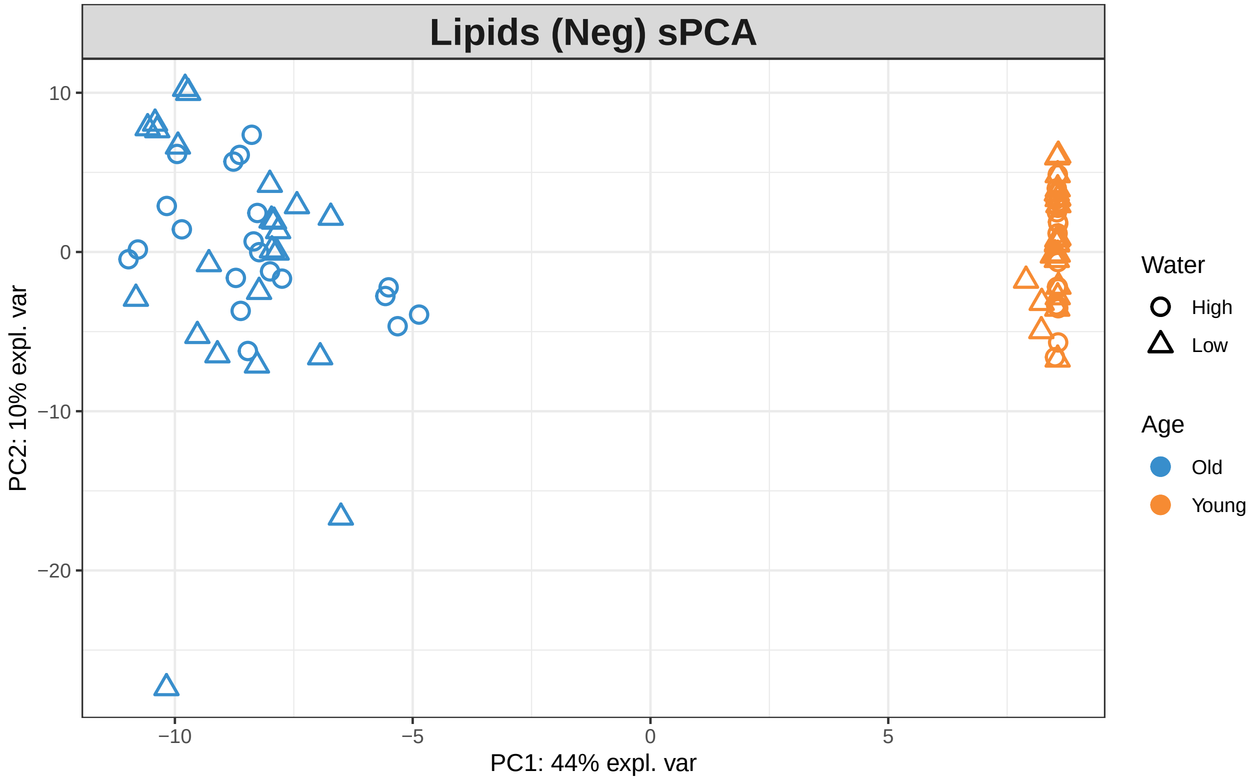
```r
# Look at variable coefficients in each component with the
# loading vectors The absolute value of loading vectors
# represent the importance of each variable to define each PC
plotLoadings(pca.res, ndisplay = 50)
```



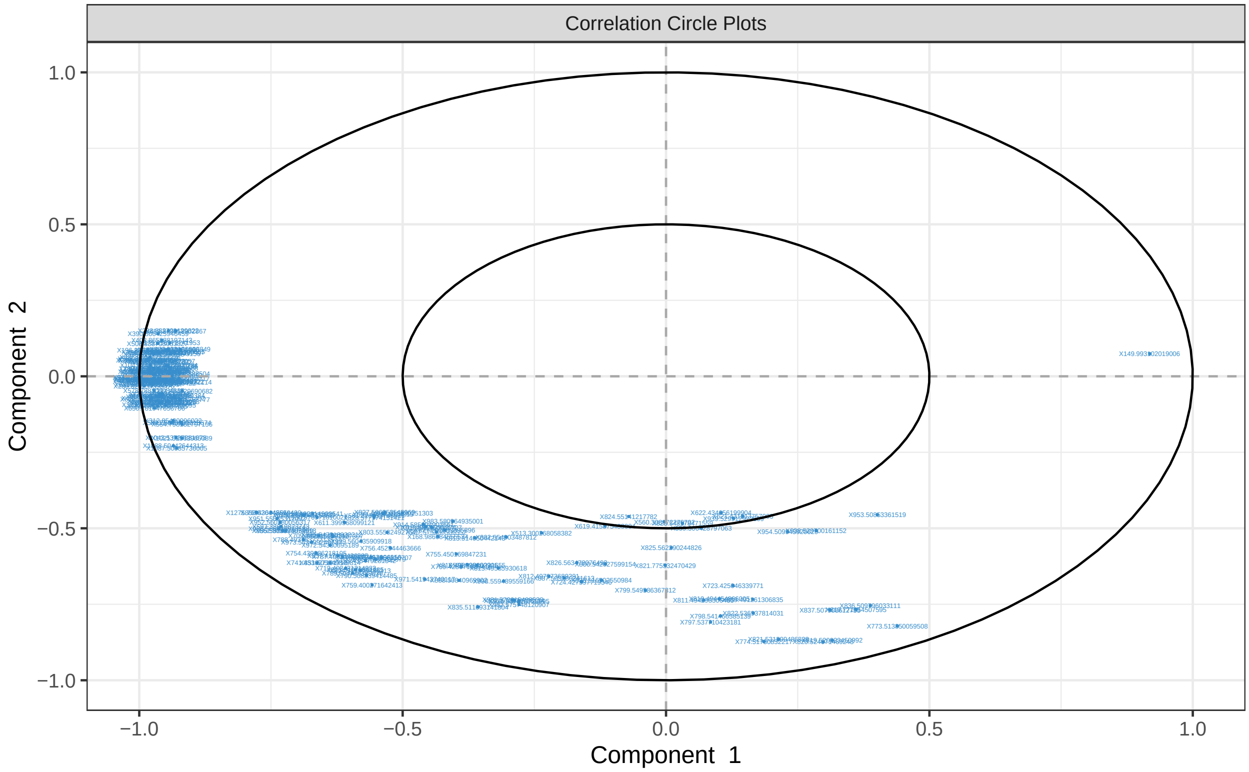
9. Identify the variables that contribute to the explained variance.

```r
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 = "Lipids (Neg) sPCA")
```



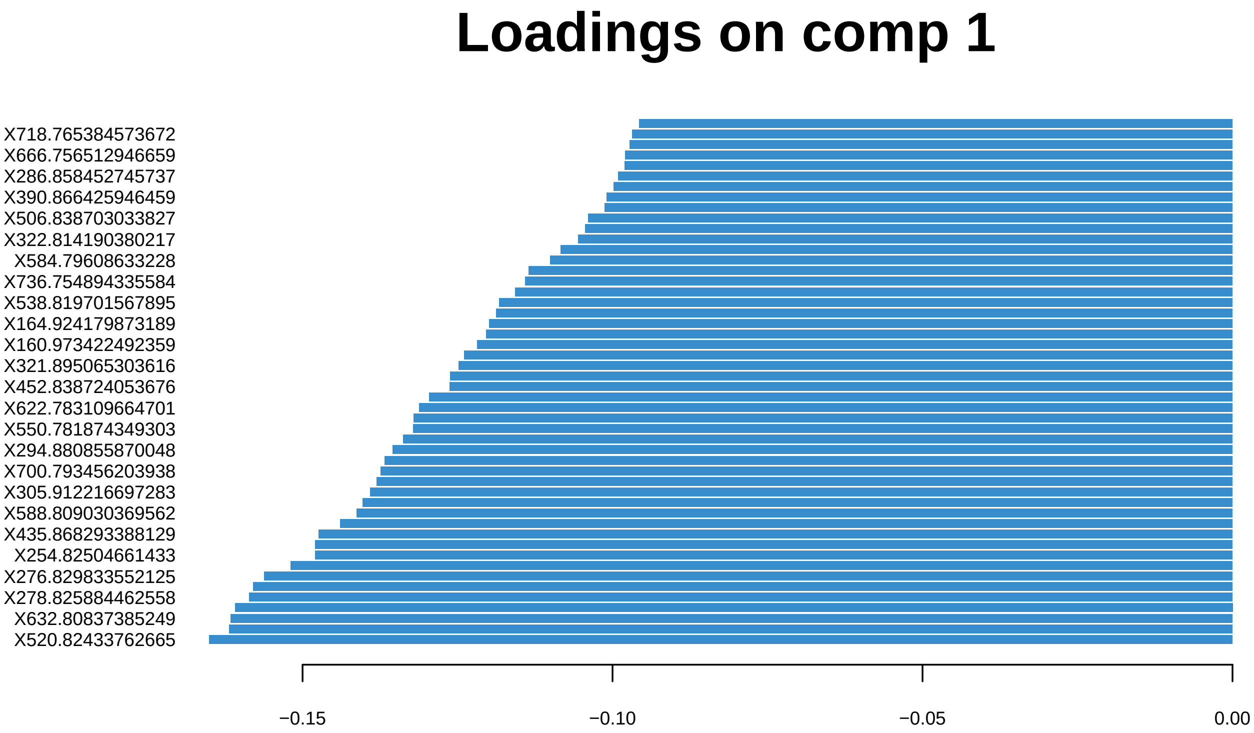
```r
# variables contributing to each component
plotVar(spca.res, cex = 1)
```



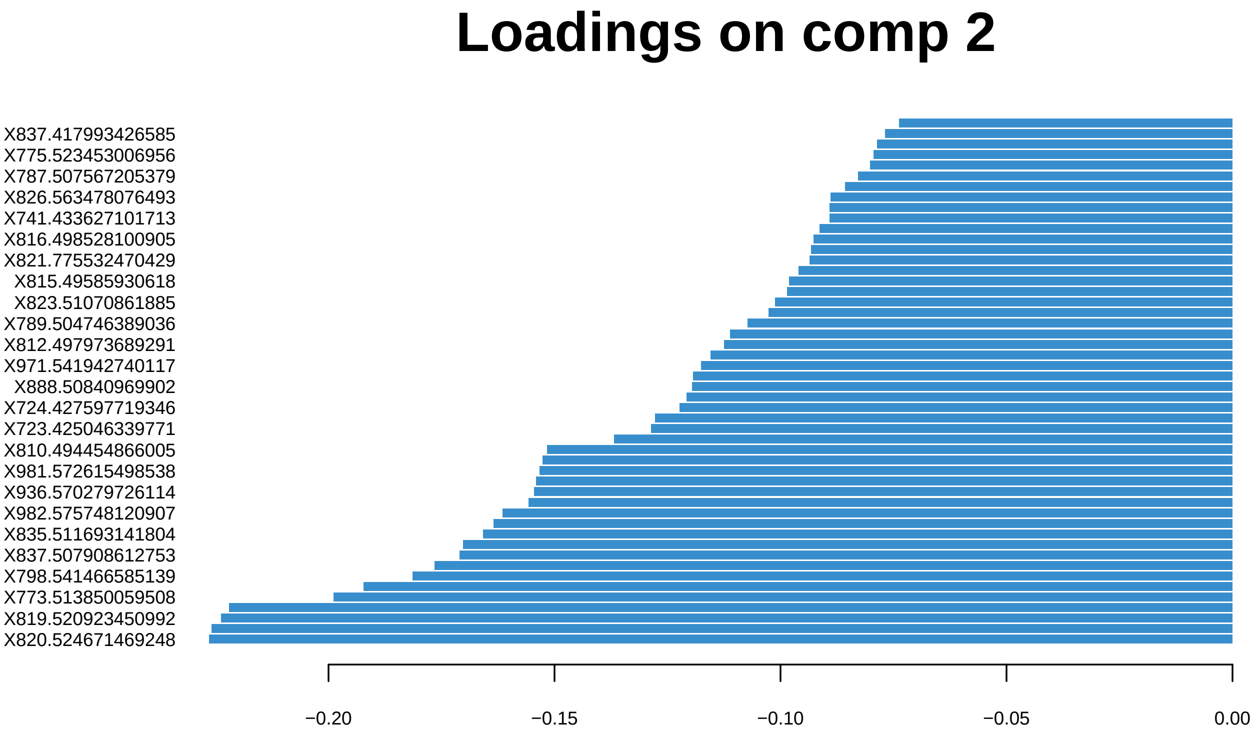
```r
selectVar(spca.res, comp = 1)$value  # view loading value of each metabolite
```

```
##                      value.var
## X520.82433762665  -0.165072912
## X92.9265961371452 -0.161815876
## X632.80837385249  -0.161565211
## X94.9236379982592 -0.160898854
## X278.825884462558 -0.158609383
## X280.823254398476 -0.157941755
## X276.829833552125 -0.156188794
## X322.88187845094  -0.151918822
## X254.82504661433  -0.147950985
## X200.877819984561 -0.147921122
## X435.868293388129 -0.147412069
## X196.866239697447 -0.143959977
## X588.809030369562 -0.141230412
## X438.854872874512 -0.140254691
## X305.912216697283 -0.139093115
## X198.86329083797  -0.138056397
## X700.793456203938 -0.137428901
## X288.855144051309 -0.136755380
## X294.880855870048 -0.135472609
## X436.866447143236 -0.133762222
## X550.781874349303 -0.132127356
## X104.927982377221 -0.132075982
## X622.783109664701 -0.131141190
## X386.857907446201 -0.129569564
## X452.838724053676 -0.126270277
## X264.852273731868 -0.126214828
## X321.895065303616 -0.124808065
## X668.770289802017 -0.123890315
## X160.973422492359 -0.121823580
## X502.852312200685 -0.120374640
## X164.924179873189 -0.119846234
## X578.788100984837 -0.118789398
## X538.819701567895 -0.118236419
## X650.781047656706 -0.115730300
## X736.754894335584 -0.114079964
## X503.853844449322 -0.113504499
## X584.79608633228  -0.110083739
## X305.845572423233 -0.108334696
## X322.814190380217 -0.105502889
## X556.795168987135 -0.104402700
## X506.838703033827 -0.103899222
## X648.781768082353 -0.101249287
## X390.866425946459 -0.100941529
## X482.792324162384 -0.099792260
## X286.858452745737 -0.099109402
## X120.935129497229 -0.098024416
## X666.756512946659 -0.097935871
## X624.779968468973 -0.097257133
## X718.765384573672 -0.096794090
## X314.851012961729 -0.095655460
## X314.781217400149 -0.095237298
## X312.78264367389  -0.095218435
## X606.806275180035 -0.093714110
## X138.892468695029 -0.092026718
## X318.870661609696 -0.091632828
## X426.836262524874 -0.087623092
## X694.782503158622 -0.086530398
## X308.901821857682 -0.086289582
## X408.865888197143 -0.086130831
## X232.890804278745 -0.084050456
## X230.896780922057 -0.074078606
## X166.92142137535  -0.073092994
## X156.891218586422 -0.071749526
## X330.891353810139 -0.071374374
## X320.881784084095 -0.070739092
## X122.932241661228 -0.068726030
## X618.770889841238 -0.068440020
## X319.884132029521 -0.065600941
## X296.887044555183 -0.065275867
## X558.813402806234 -0.063199360
## X564.821993700256 -0.062762803
## X298.883728129822 -0.059542999
## X178.946687551166 -0.059107093
## X220.867279165528 -0.058059507
## X434.867561319988 -0.056497240
## X114.93221806588  -0.054503571
## X175.956372741953 -0.054020025
## X192.925707043156 -0.052561819
## X582.79435709715  -0.046996569
## X626.799732769052 -0.044949087
## X312.85420996022  -0.040088837
## X1088.50442644313 -0.039149477
## X372.739602758722 -0.038440578
## X424.838388390384 -0.035532760
## X464.871364308467 -0.035492398
## X364.877336109855 -0.034317064
## X121.942087311452 -0.033259573
## X458.849058902867 -0.028982420
## X1043.53727881678 -0.028772040
## X1087.50085736005 -0.025897744
## X696.780418989217 -0.015433832
## X432.863692787504 -0.014226454
## X152.880846757777 -0.014110112
## X570.838120828849 -0.011847712
## X417.840961318674 -0.009022673
## X636.823967922114 -0.006785887
## X1021.52563965089 -0.004934769
## X554.798182797156 -0.004073056
## X287.884829690682 -0.003106421
## X149.993102019006  0.002717144
```

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



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

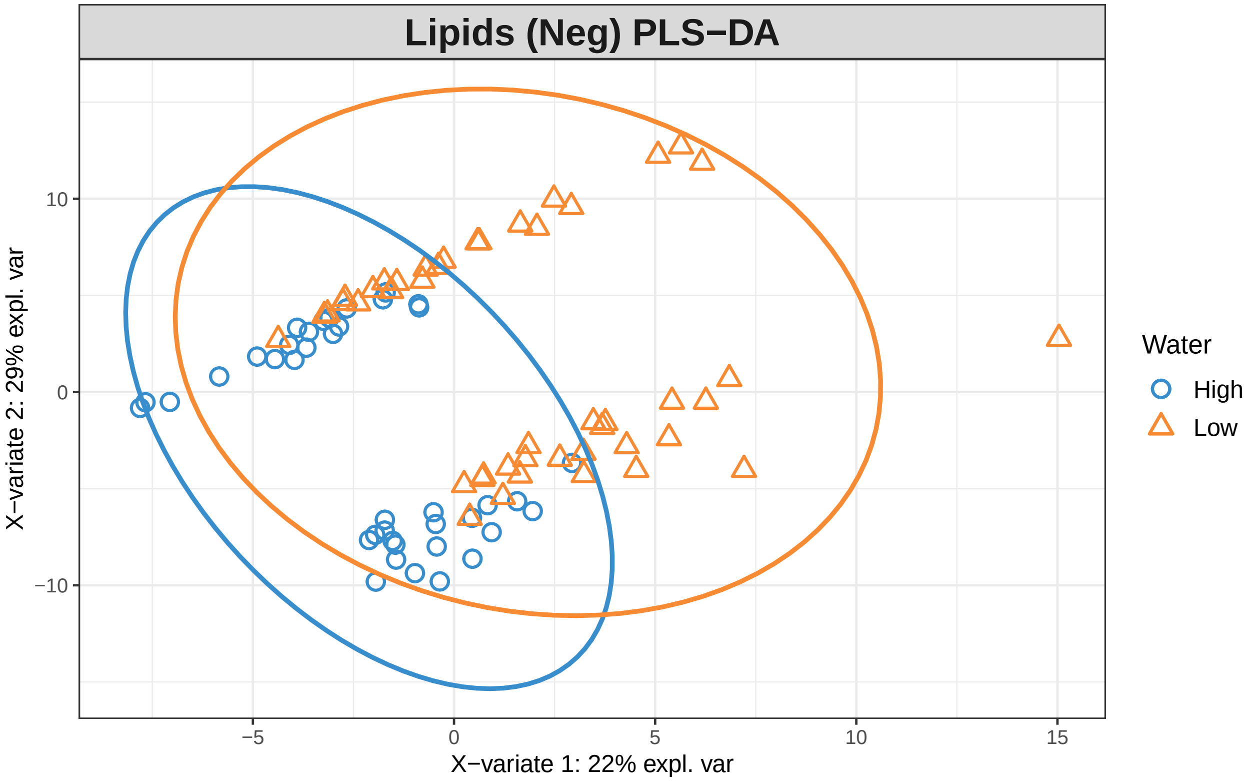


PLS-DA

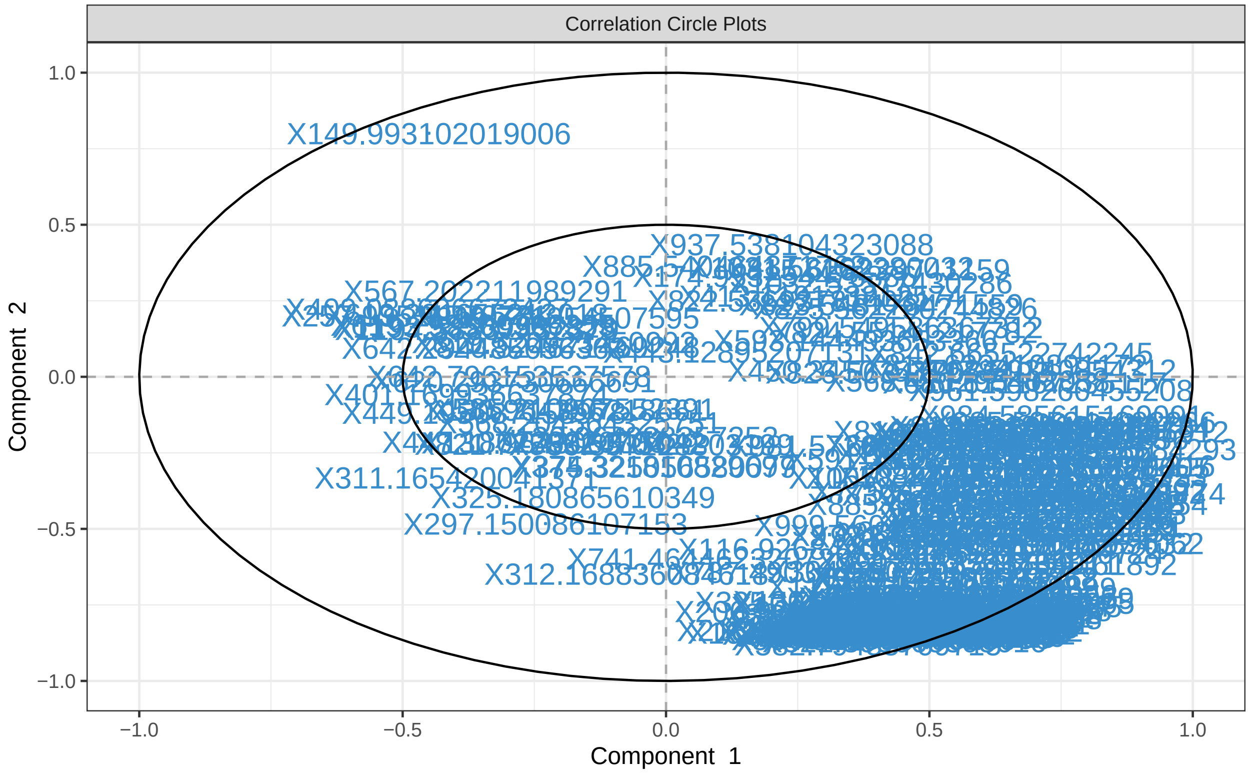
10. Classify samples into known groups and predict the class of new samples.

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

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



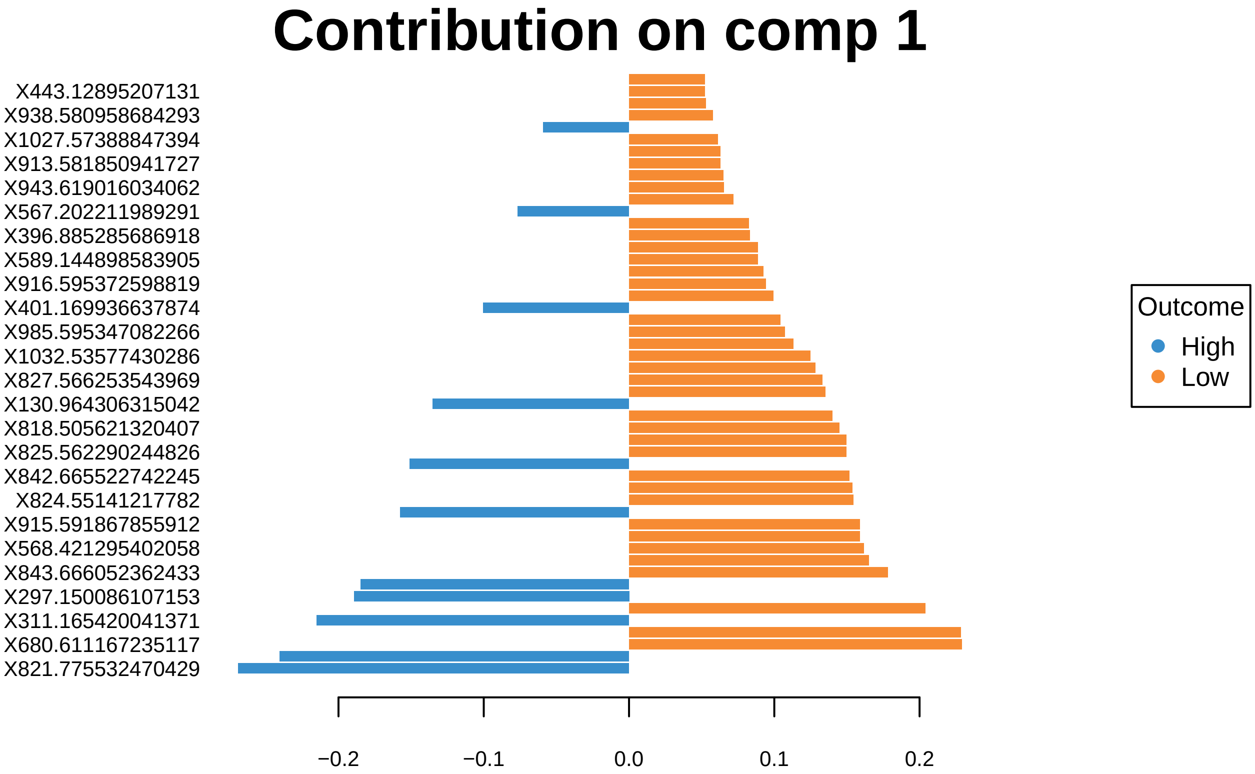
```
# plot and select the variables
plotVar(Lneg.splsda)
```



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

```
## $name
##   [1] "X821.775532470429" "X312.168836084618" "X680.611167235117"
##   [4] "X146.93743175893"  "X311.165420041371" "X815.623229741159"
##   [7] "X297.150086107153" "X447.180223820703" "X843.666052362433"
##  [10] "X961.598260455208" "X568.421295402058" "X1020.49336637357"
##  [13] "X915.591867855912" "X568.204564325751" "X824.55141217782"
##  [16] "X823.548178471559" "X842.665522742245" "X449.195087158978"
##  [19] "X825.562290244826" "X606.27669784705"  "X818.505621320407"
##  [22] "X799.549586367312" "X130.964306315042" "X242.939643504364"
##  [25] "X827.566253543969" "X174.953254476886" "X1032.53577430286"
##  [28] "X962.604422430784" "X985.595347082266" "X591.253174352751"
##  [31] "X401.169936637874" "X826.563478076493" "X916.595372598819"
##  [34] "X768.59340854312"  "X589.144898583905" "X914.585094903561"
##  [37] "X396.885285686918" "X1082.74498574183" "X567.202211989291"
##  [40] "X817.502267142856" "X943.619016034062" "X1031.53162380032"
##  [43] "X913.581850941727" "X984.585615169091" "X1027.57388847394"
##  [46] "X130.964356218661" "X938.580958684293" "X740.560102261774"
##  [49] "X443.12895207131"  "X814.623410891172" "X537.320770745072"
##  [52] "X593.144493653366" "X963.610536041216" "X986.600758488323"
##  [55] "X822.536937814031" "X885.540464371762" "X1058.74765481539"
##  [58] "X561.320035745021" "X420.244158424984" "X642.294485656363"
##  [61] "X325.180865610349" "X990.629963089632" "X937.538104323088"
##  [64] "X741.464462320728" "X959.587529993008" "X769.595475138005"
##  [67] "X374.239062154048" "X391.201599110739" "X989.626067860992"
##  [70] "X619.793500366355" "X642.796153537578" "X257.93511779731"
##  [73] "X323.212503942115" "X793.50817984514"  "X375.325056580079"
##  [76] "X374.321810829697" "X409.088606672427" "X619.291655213018"
##  [79] "X130.964338357252" "X620.293730666691" "X789.342675777525"
##  [82] "X960.590773904268" "X1057.74455579026" "X585.210907552391"
##  [85] "X780.597128303199" "X939.589454782854" "X819.520923450992"
##  [88] "X820.524671469248" "X351.245639523269" "X1028.57474860271"
##  [91] "X882.586559461892" "X1055.6011196724"  "X419.232757040941"
##  [94] "X675.351068102382" "X775.523453006956" "X450.345075311784"
##  [97] "X1193.58569987828" "X819.771054507595" "X413.119173812574"
## [100] "X1194.58837432379"
##
## $value
##                      value.var
## X821.775532470429 -0.268956746
## X312.168836084618 -0.240532689
## X680.611167235117  0.229027627
## X146.93743175893   0.228458906
## X311.165420041371 -0.214868003
## X815.623229741159  0.203851746
## X297.150086107153 -0.189310957
## X447.180223820703 -0.184727755
## X843.666052362433  0.178157617
## X961.598260455208  0.165044290
## X568.421295402058  0.161690744
## X1020.49336637357  0.158827849
## X915.591867855912  0.158703676
## X568.204564325751 -0.157464694
## X824.55141217782   0.154280163
## X823.548178471559  0.153650525
## X842.665522742245  0.151755802
## X449.195087158978 -0.150893628
## X825.562290244826  0.149506388
## X606.27669784705   0.149378446
## X818.505621320407  0.144871426
## X799.549586367312  0.139772055
## X130.964306315042 -0.135193948
## X242.939643504364  0.135003006
## X827.566253543969  0.133085408
## X174.953254476886  0.128150042
## X1032.53577430286  0.124708636
## X962.604422430784  0.113186593
## X985.595347082266  0.107167254
## X591.253174352751  0.104081573
## X401.169936637874 -0.100246852
## X826.563478076493  0.099397420
## X916.595372598819  0.094144910
## X768.59340854312   0.092374486
## X589.144898583905  0.088686280
## X914.585094903561  0.088676842
## X396.885285686918  0.083150377
## X1082.74498574183  0.082408344
## X567.202211989291 -0.076588465
## X817.502267142856  0.071604558
## X943.619016034062  0.065242224
## X1031.53162380032  0.064957018
## X913.581850941727  0.062745761
## X984.585615169091  0.062660064
## X1027.57388847394  0.061120474
## X130.964356218661 -0.059076557
## X938.580958684293  0.057613902
## X740.560102261774  0.052733498
## X443.12895207131   0.052270566
## X814.623410891172  0.052026333
## X537.320770745072  0.047670891
## X593.144493653366  0.047370621
## X963.610536041216  0.046732743
## X986.600758488323  0.045996202
## X822.536937814031  0.044755176
## X885.540464371762  0.043922575
## X1058.74765481539  0.042955724
## X561.320035745021  0.042843712
## X420.244158424984  0.042044321
## X642.294485656363 -0.038320797
## X325.180865610349 -0.034453629
## X990.629963089632  0.033753634
## X937.538104323088  0.033396335
## X741.464462320728 -0.032192793
## X959.587529993008  0.032122039
## X769.595475138005  0.031817789
## X374.239062154048  0.030472709
## X391.201599110739  0.030161647
## X989.626067860992  0.026273950
## X619.793500366355 -0.025748743
## X642.796153537578 -0.025481431
## X257.93511779731  -0.024837014
## X323.212503942115  0.023853394
## X793.50817984514   0.022943534
## X375.325056580079 -0.020436824
## X374.321810829697 -0.019191827
## X409.088606672427 -0.016407273
## X619.291655213018 -0.014280706
## X130.964338357252 -0.013523649
## X620.293730666691 -0.012951103
## X789.342675777525  0.012820788
## X960.590773904268  0.011942128
## X1057.74455579026  0.011633105
## X585.210907552391 -0.011118777
## X780.597128303199 -0.010717203
## X939.589454782854  0.009233104
## X819.520923450992 -0.009149708
## X820.524671469248 -0.007972684
## X351.245639523269  0.007625640
## X1028.57474860271  0.006948317
## X882.586559461892  0.006493979
## X1055.6011196724   0.005922544
## X419.232757040941  0.005372062
## X675.351068102382  0.004776881
## X775.523453006956  0.004511831
## X450.345075311784  0.004128185
## X1193.58569987828 -0.003714740
## X819.771054507595 -0.003148944
## X413.119173812574  0.001701076
## X1194.58837432379 -0.001533788
##
## $comp
## [1] 1
```

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



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$Age),
    FUN = "mean", simplify = T, data = class)
av.plsda <- mixOmics::plsda(av_Y[, 4:594], av_Y$Group.2)  # fungus

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

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                 =  39.7437437437437
##
## Sum of Indicator Values              =  481.84
##
## Sum of Significant Indicator Values  =  441.67
##
## Number of Significant Indicators     =  508
##
## Significant Indicator Distribution
##
##   1   2
## 468  40
```

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/Lipids Statis
```

Lipids (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 Onli

L_pos <- read_tsv(paste(path, "HvL_L_pos/XCMS.annotated.diffreport..High_F_Lpos_Kenia_Thesis.vs.Low_F_Lp
    sep = ""))

# dependent variable: metabolite intensities
Y <- L_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)
```

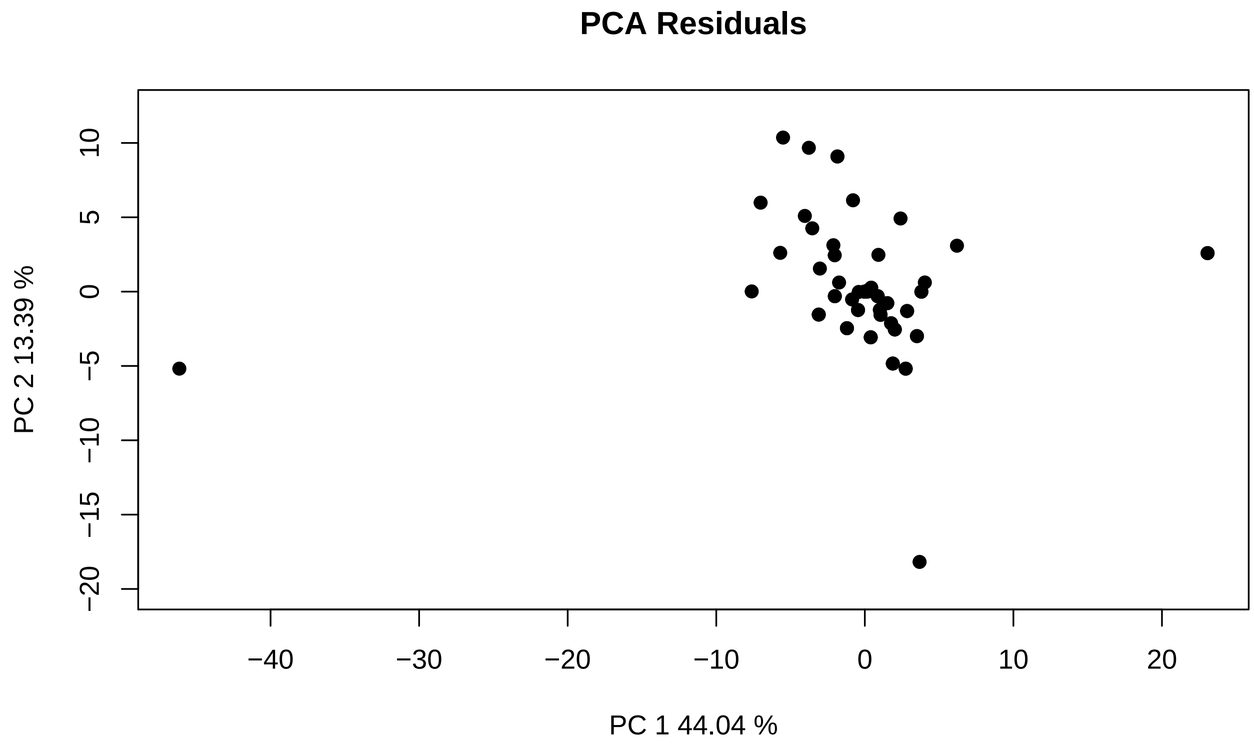
```
##
## Warning: Because variables in the linear model are redundant,
## the linear model design has been truncated (via QR decomposition).
## Original X columns: 72
## Final X columns (rank): 66
## Check coefficients or degrees of freedom in ANOVA to see changes.
```

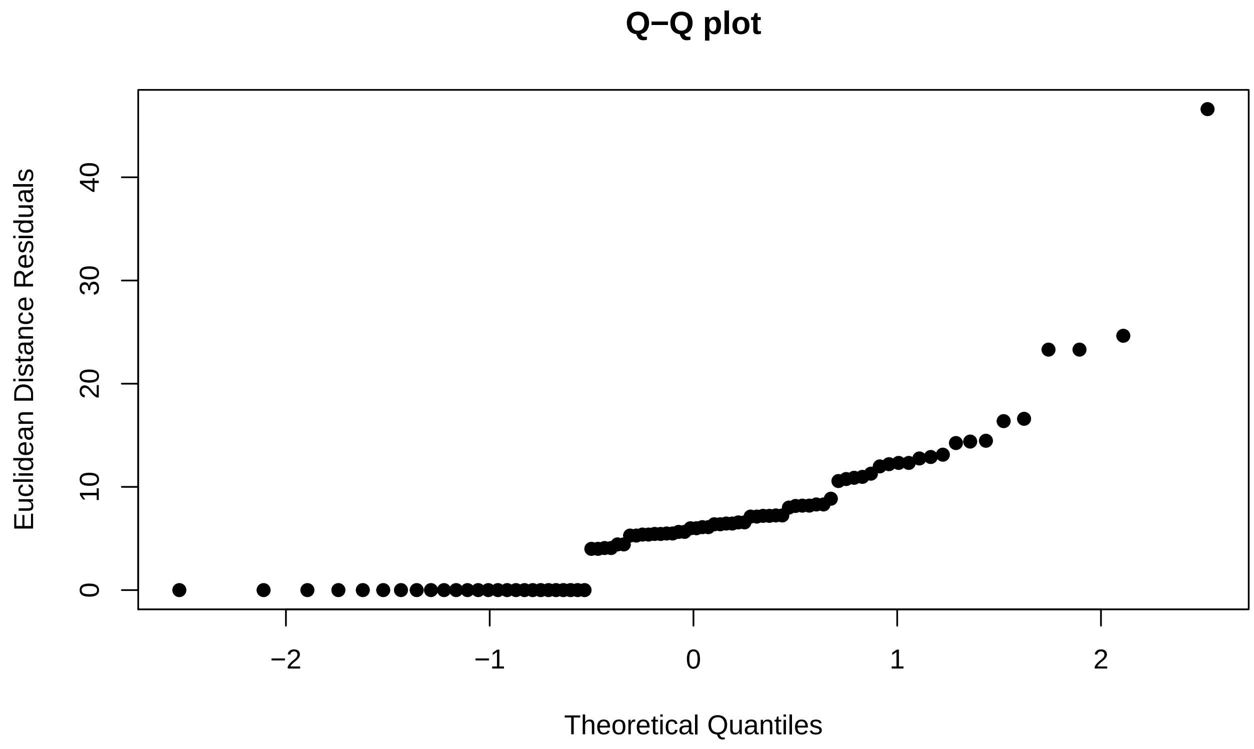
```r
summary(LMpos)
```

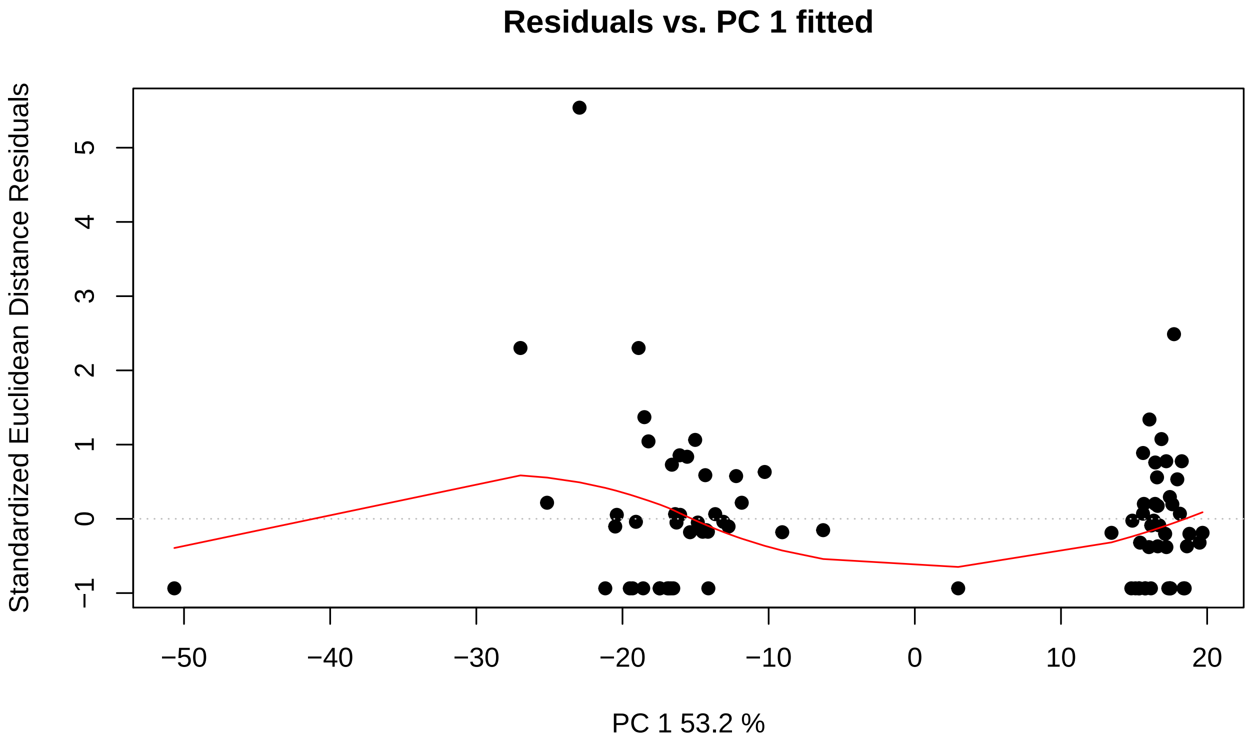
```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 86
## Number of dependent variables: 683
## Data space dimensions: 85
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Full Model Analysis of Variance
##
##                              Df Residual Df       SS Residual SS       Rsq
## Block * Water * Fungus * Age 65          20 49747.07    8307.933 0.8568955
##                                     F Z (from F)      Pr(>F)
## Block * Water * Fungus * Age 1.842431    2.58403 0.001533333
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##              Trace Proportion Rank
## Fitted    585.2596  0.8568955   65
## Residuals  97.7404  0.1431045   20
## Total     683.0000  1.0000000   85
##
## Eigenvalues
##
##                PC1       PC2       PC3       PC4       PC5       PC6       PC7
## Fitted    311.3806   67.3963   35.4350   20.5509   18.3187   13.2333   10.1072
## Residuals  43.0419   13.0908    9.7591    6.1830    3.1656    2.8320    2.7694
## Total     323.9242   81.4856   40.2016   31.0066   24.1634   16.4139   14.8787
##                PC8       PC9      PC10      PC11      PC12      PC13      PC14
## Fitted      9.1903    8.3884    7.4118    6.2372    5.4871    5.1245    4.2767
## Residuals   2.2790    2.0167    1.9683    1.6868    1.5468    1.5392    1.1654
## Total      14.0529   11.0515    9.9588    8.1565    6.8662    5.7272    5.3013
##               PC15      PC16      PC17      PC18      PC19      PC20      PC21
## Fitted      3.9830    3.3468    3.0941    2.7945    2.6539    2.5263    2.3272
## Residuals   1.0131    0.9826    0.8398    0.7293    0.5793    0.5523
## Total       5.0455    4.5578    4.2103    4.0028    3.6763    3.2359    2.9604
##               PC22      PC23      PC24      PC25      PC26      PC27      PC28
## Fitted      2.2112    2.0891    2.0518    1.9872    1.9052    1.7833    1.6175
## Residuals
## Total       2.7121    2.6379    2.5320    2.4061    2.1694    2.1174    2.0148
##               PC29      PC30      PC31      PC32      PC33      PC34      PC35
## Fitted      1.5478    1.4542    1.3892    1.3335    1.3000    1.2001    1.1867
## Residuals
## Total       1.9449    1.8270    1.6925    1.6654    1.6369    1.5569    1.4713
##               PC36      PC37      PC38      PC39      PC40      PC41      PC42
## Fitted      1.1283    1.0427    1.0269    0.9956    0.9134    0.8930    0.8515
## Residuals
## Total       1.4683   1.3572   1.2930   1.2518   1.2083   1.1966   1.1621
##                PC43     PC44     PC45     PC46     PC47     PC48     PC49
## Fitted      0.7939   0.7560   0.7484   0.7283   0.7200   0.6745   0.6565
## Residuals
## Total       1.1318   1.0728   1.0001   0.9838   0.9793   0.9330   0.8814
##                PC50     PC51     PC52     PC53     PC54     PC55     PC56
## Fitted      0.6386   0.5977   0.5673   0.5413   0.5356   0.5024   0.4850
## Residuals
## Total       0.8441   0.8065   0.7870   0.7750   0.7577   0.7354   0.7052
##                PC57     PC58     PC59     PC60     PC61     PC62     PC63
## Fitted      0.4689   0.4484   0.4178   0.3903   0.3368   0.3225   0.2984
## Residuals
## Total       0.6974   0.6670   0.6366   0.6172   0.5867   0.5777   0.5679
##                PC64     PC65     PC66     PC67     PC68     PC69     PC70
## Fitted      0.2570   0.2022
## Residuals
## Total       0.5630   0.5247   0.5181   0.4947   0.4828   0.4640   0.4442
##                PC71     PC72     PC73     PC74     PC75     PC76     PC77
## Fitted
## Residuals
## Total       0.4343   0.4046   0.3865   0.3833   0.3503   0.3398   0.3254
##                PC78     PC79     PC80     PC81     PC82     PC83     PC84
## Fitted
## Residuals
## Total       0.3111   0.2985   0.2695   0.2524   0.2354   0.2238   0.1813
##                PC85
## Fitted
## Residuals
## Total       0.1692
```

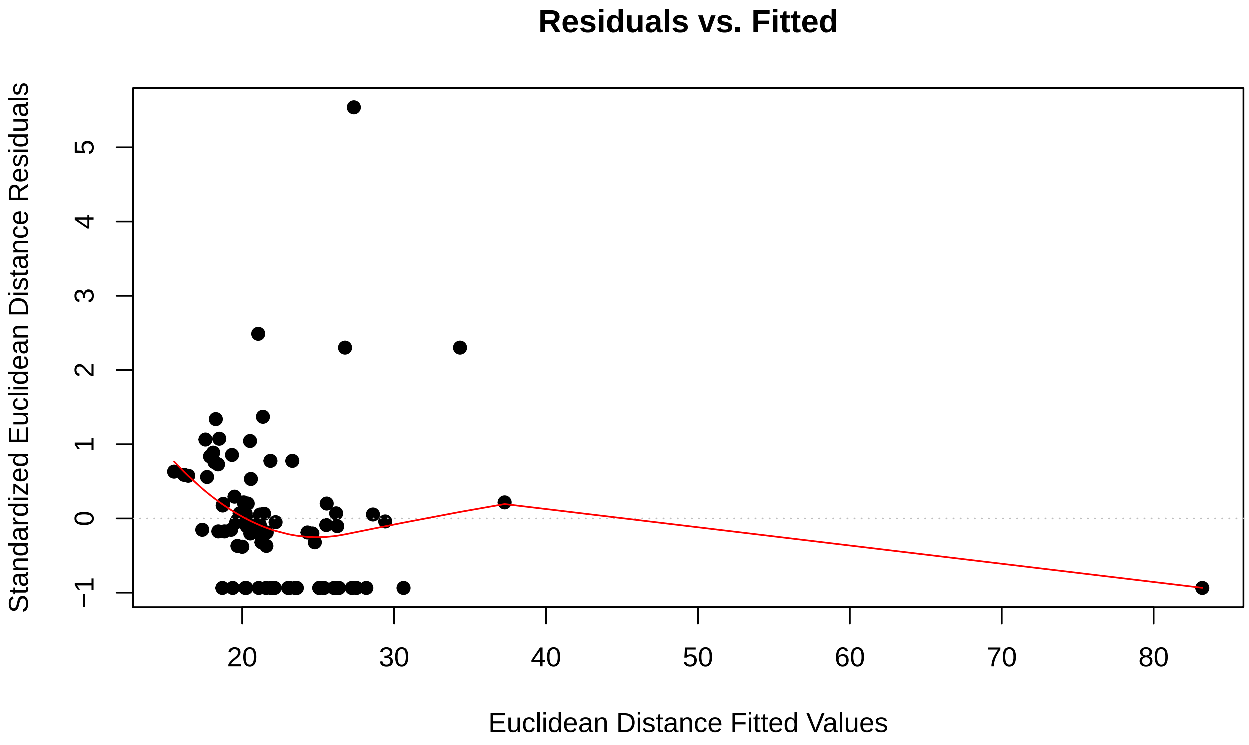
4. Examine RRPP plots to check for assumptions.

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

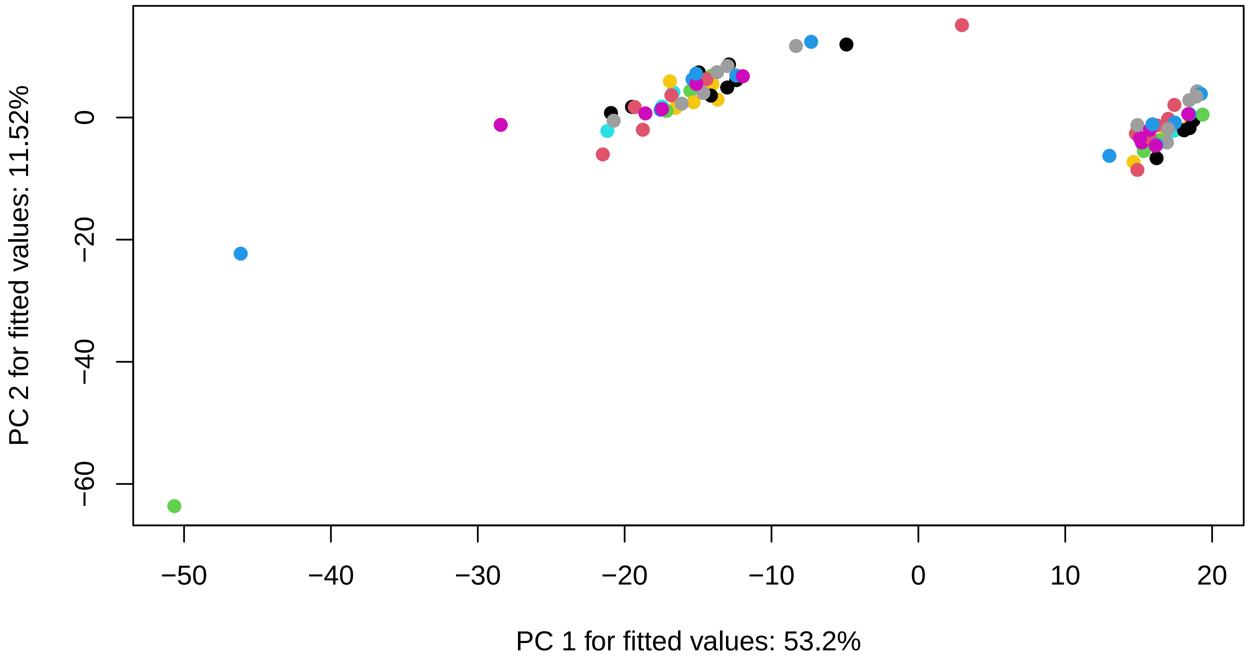

PCA Residuals
PC 2 13.39 %
10
5
0
−5
−10
−15
−20
−40
−30
−20
−10
0
10
20
PC 1 44.04 %


Q−Q plot
Euclidean Distance Residuals
40
30
20
10
0
−2
−1
0
1
2
Theoretical Quantiles





```
# 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)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 1000
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##                         Df    SS     MS      Rsq      F        Z Pr(>F)
## Block                   1   192 191.57 0.003300 0.4612           0.511
## Water                   1   483 483.20 0.008323 1.8253           0.082 .
## Fungus                  8  4998 624.77 0.086093 2.8420  1.37717  0.089 .
## Age                     1   262 261.78 0.004509 1.5367  0.97430  0.150
## Block:Water             1   265 264.73 0.004560 0.6373           0.473
## Block:Fungus            8  1973 246.62 0.033984 1.1219  0.04478  0.493
## Water:Fungus            8  5412 676.46 0.093216 3.0772  1.56181  0.069 .
## Block:Age               1   171 171.43 0.002953 1.0063  0.29459  0.282
## Water:Age               1   283 282.55 0.004867 1.6586  1.15732  0.124
## Fungus:Age              8  2905 363.15 0.050042 2.1317  1.44354  0.084 .
## Block:Water:Fungus      5  1099 219.83 0.018933 0.5292 -0.99440  0.830
## Block:Water:Age         1   244 244.17 0.004206 1.4333  0.96963  0.172
## Block:Fungus:Age        8  1537 192.10 0.026471 1.1276  0.14470  0.440
## Water:Fungus:Age        8  2992 374.00 0.051537 2.1954  1.68213  0.060 .
## Block:Water:Fungus:Age  5   852 170.36 0.014672 0.4101 -1.80737  0.967
## Residuals              20  8308 415.40 0.143105
## Total                  85 58055
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = scaled_Y ~ Block * Water * Fungus * Age, SS.type = "III",
##     data = class, print.progress = F)
```

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)
```

```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 86
## Number of dependent variables: 683
## Data space dimensions: 85
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Statistics (distances) of coefficients with 95 percent confidence intervals,
## effect sizes, and probabilities of exceeding observed values based on
## 1000 random permutations using RRPP
##
##                                         d.obs UCL (95%)         Zd Pr(>d)
## (Intercept)                         23.544152  41.51296 -0.5121739  0.724
## Block                                9.786914  12.32209  1.0567139  0.117
## WaterLow                            47.486069  34.26954  2.9724482  0.018
## FungusCer                           84.888912  34.80277  7.6812715  0.001
## FungusCok                           32.646383  45.67033  0.8632042  0.144
## FungusCtrl                          31.102453  45.45123  0.5713620  0.206
## FungusNig                           48.332971  46.76305  2.0404264  0.045
## FungusPen                           32.832861  47.28707  0.7381161  0.155
## FungusPod                           45.246604  52.84410  1.0466564  0.120
## FungusPre                           36.065412  44.67646  1.0129375  0.127
## FungusXyl                           47.452039  56.43460  1.0536559  0.107
## AgeYoung                            34.951950  36.52152  1.5051303  0.060
## Block:WaterLow                      16.270431  15.85712  1.8110925  0.044
## Block:FungusCer                     14.028072  15.79861  1.1535390  0.091
## Block:FungusCok                     25.376840  27.40444  1.3866883  0.076
## Block:FungusCtrl                    12.388277  18.14545  0.5191441  0.197
## Block:FungusNig                     18.908635  18.87739  1.9623782  0.050
## Block:FungusPen                     12.271008  16.55300  0.5871301  0.193
## Block:FungusPod                     25.571457  26.23888  1.3227452  0.061
## Block:FungusPre                     14.105004  17.31398  0.9721806  0.133
## Block:FungusXyl                     23.294892  27.74632  0.9970428  0.102
## WaterLow:FungusCer                  99.014931  41.39848  7.6380207  0.001
## WaterLow:FungusCok                  34.154397  44.56925  1.1760867  0.126
## WaterLow:FungusCtrl                 42.287924  66.10774  0.4521661  0.239
## WaterLow:FungusNig                  56.913034  68.46722  1.4593611  0.095
## WaterLow:FungusPen                  38.365090  65.68398  0.2064283  0.291
## WaterLow:FungusPod                 67.355512  73.29989  1.6504893  0.075
## WaterLow:FungusPre                 47.311513  67.27195  0.7842440  0.183
## WaterLow:FungusXyl                 36.459390  44.21099  1.3751149  0.106
## Block:AgeYoung                     13.093122  17.01666  0.9003245  0.128
## WaterLow:AgeYoung                  51.352937  54.07651  1.9126063  0.055
## FungusCer:AgeYoung                 83.041524  43.79695  6.4495792  0.001
## FungusCok:AgeYoung                 41.207030  53.07828  0.8644060  0.163
## FungusCtrl:AgeYoung                35.269524  56.06061  0.2047513  0.312
## FungusNig:AgeYoung                 54.192574  59.38761  1.7655244  0.062
## FungusPen:AgeYoung                 40.464539  54.71418  0.6943587  0.172
## FungusPod:AgeYoung                 55.508650  67.81827  0.9367630  0.139
## FungusPre:AgeYoung                 40.825824  60.44813  0.5865109  0.206
## FungusXyl:AgeYoung                 56.169478  69.64897  0.9582845  0.121
## Block:WaterLow:FungusCtrl          15.984669  24.53239  0.4830718  0.221
## Block:WaterLow:FungusNig           21.298759  24.38506  1.4956861  0.088
## Block:WaterLow:FungusPen           14.770721  23.89548  0.2280524  0.299
## Block:WaterLow:FungusPod           31.703834  30.77413  1.7622964  0.045
## Block:WaterLow:FungusPre           17.679391  24.27684  0.8037199  0.155
## Block:WaterLow:AgeYoung            22.098214  24.29755  1.6529781  0.078
## Block:FungusCer:AgeYoung           17.226895  25.02561  0.6229764  0.178
## Block:FungusCok:AgeYoung           33.305579  38.09000  1.1930870  0.088
## Block:FungusCtrl:AgeYoung          16.484303  25.51814  0.4343134  0.221
## Block:FungusNig:AgeYoung           22.845031  25.37889  1.6403888  0.084
## Block:FungusPen:AgeYoung           15.324704  25.28449  0.2725888  0.272
## Block:FungusPod:AgeYoung           33.670660  38.42250  1.2792464  0.076
## Block:FungusPre:AgeYoung           16.459573  25.13549  0.3927039  0.243
## Block:FungusXyl:AgeYoung           32.829489  39.07206  1.2348619  0.088
## WaterLow:FungusCer:AgeYoung        96.851669  48.52001  6.9266087  0.001
## WaterLow:FungusCok:AgeYoung        38.321637  52.04504  0.8179703  0.165
## WaterLow:FungusCtrl:AgeYoung       55.894646  76.29647  0.5713189  0.226
## WaterLow:FungusNig:AgeYoung        63.204696  76.63918  1.0650521  0.125
## WaterLow:FungusPen:AgeYoung        54.300446  75.81552  0.5139950  0.219
## WaterLow:FungusPod:AgeYoung        71.877291  87.69984  1.0882374  0.125
## WaterLow:FungusPre:AgeYoung        64.020273  77.24028  1.0331098  0.138
## WaterLow:FungusXyl:AgeYoung        40.163352  48.46593  1.0653317  0.124
## Block:WaterLow:FungusCtrl:AgeYoung 25.535467  32.42349  0.9401836  0.157
## Block:WaterLow:FungusNig:AgeYoung  25.124535  31.86652  0.8692389  0.163
## Block:WaterLow:FungusPen:AgeYoung  23.448861  32.12201  0.6276337  0.209
## Block:WaterLow:FungusPod:AgeYoung  38.663262  43.74092  1.3955388  0.074
## Block:WaterLow:FungusPre:AgeYoung  25.453156  32.46952  0.8830111  0.160
```

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.

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

```
##
## Pairwise comparisons
##
## Groups: Asp Cer Cok Ctrl Nig Pen Pod Pre Xyl
##
## RRPP: 1000 permutations
```

```
##
## LS means:
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##                   d UCL (95%)          Z Pr > d
## Asp:Cer  11.158167 15.229697 -0.5153920  0.688
## Asp:Cok   9.024054 13.438229 -0.9684936  0.921
## Asp:Ctrl  6.036957  8.894838 -1.6089663  0.997
## Asp:Nig   6.067511  9.206486 -1.4417287  0.991
## Asp:Pen   6.422362 10.090661 -0.8541315  0.822
## Asp:Pod   6.837826 10.333941 -1.0812562  0.901
## Asp:Pre   6.791038 10.090248 -1.0450779  0.908
## Asp:Xyl   8.375971 12.174533 -1.2108784  0.965
## Cer:Cok  11.157157 14.921579 -0.9835018  0.867
## Cer:Ctrl 11.093509 14.974012 -0.5900230  0.711
## Cer:Nig  12.427872 16.870747 -0.4284354  0.664
## Cer:Pen  10.300883 13.467506 -0.8800061  0.838
## Cer:Pod   9.946664 13.813806 -0.8025901  0.796
## Cer:Pre  13.628882 17.946004 -0.3504843  0.627
## Cer:Xyl  12.806248 16.976181 -0.5884314  0.696
## Cok:Ctrl  7.270125 11.665545 -1.2701963  0.984
## Cok:Nig   8.807049 13.759022 -0.8786852  0.888
## Cok:Pen   9.290374 12.764332 -1.2705225  0.936
## Cok:Pod   8.351191 12.146919 -1.3555557  0.978
## Cok:Pre  10.373256 15.687063 -0.6539647  0.749
## Cok:Xyl   9.690891 14.379249 -1.0371360  0.911
## Ctrl:Nig  5.990512  9.679057 -1.2370299  0.972
## Ctrl:Pen  6.799042 10.277522 -0.9566199  0.868
## Ctrl:Pod  6.660332  9.983192 -1.2368947  0.946
## Ctrl:Pre  7.240838 11.134378 -0.8693509  0.821
## Ctrl:Xyl  7.453307 11.526944 -1.2170584  0.957
## Nig:Pen   7.868371 11.825224 -0.6387479  0.723
## Nig:Pod   7.515830 11.375221 -0.9085770  0.861
## Nig:Pre   6.263519  9.724491 -1.4069097  0.982
## Nig:Xyl   7.929640 11.937092 -1.2633779  0.984
## Pen:Pod   6.942859 10.211510 -1.2630172  0.946
## Pen:Pre   8.384297 12.814658 -0.4723397  0.666
## Pen:Xyl   8.730691 12.506742 -1.0284870  0.896
## Pod:Pre   8.190942 12.574782 -0.6579508  0.718
## Pod:Xyl   8.309081 12.217757 -1.1506261  0.927
## Pre:Xyl   8.653157 13.019492 -1.0158471  0.934
```

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

```
##
## Pairwise comparisons
##
## Groups: High Low
##
## RRPP: 1000 permutations
##
## LS means:
```

```
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##                 d UCL (95%)           Z Pr > d
## High:Low 6.774547  7.928821 -0.7554347   0.77
```

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

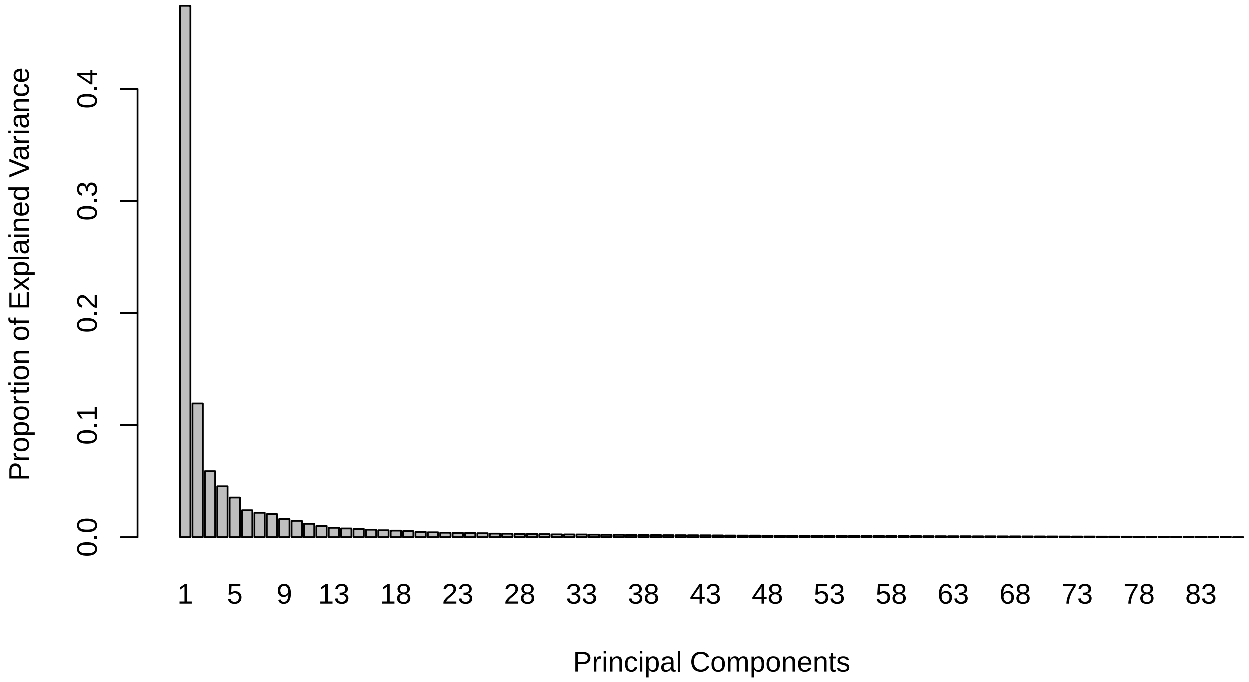
```
##
## Pairwise comparisons
##
## Groups: Old Young
##
## RRPP: 1000 permutations
##
## LS means:
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##                  d UCL (95%)           Z Pr > d
## Old:Young 33.85237  34.96782 -0.08753105  0.513
```

PCA

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

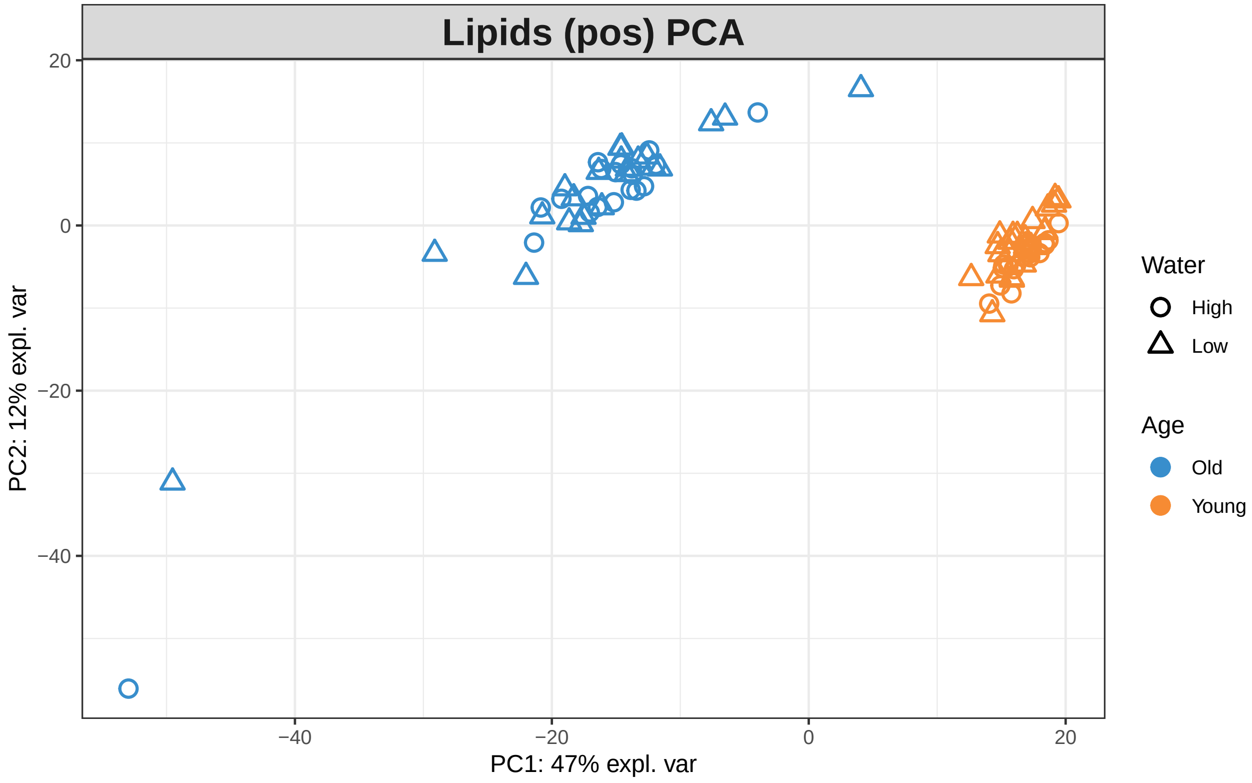
```
# tune how many components to use
tune.pca(scaled_Y)
```

```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 323.924219  81.485606  40.201552  31.006612  24.163363  16.413864  14.878736
##        PC8        PC9       PC10
##  14.052886  11.051477   9.958834
##
## Proportion of explained variance for the first 10 principal components, see object$explained_varianc
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 0.47426679 0.11930543 0.05886025 0.04539767 0.03537828 0.02403201 0.02178439
##        PC8        PC9       PC10
## 0.02057524 0.01618079 0.01458102
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 0.4742668 0.5935722 0.6524325 0.6978301 0.7332084 0.7572404 0.7790248 0.7996001
##       PC9      PC10
## 0.8157808 0.8303619
##
##  Other available components:
##  --------------------
##  loading vectors: see object$rotation
```

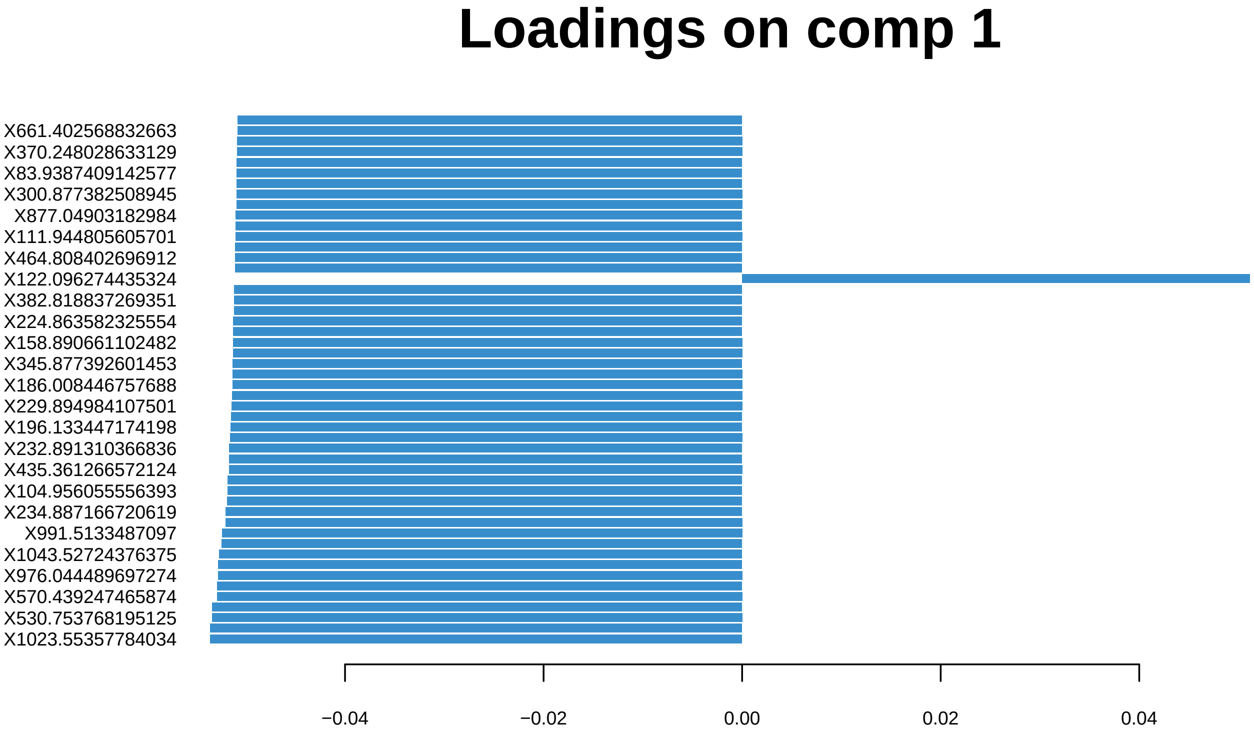


```
pca.res <- mixOmics::pca(scaled_Y, ncomp = 4, scale = F)

# 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 = "Lipids (pos) PCA")
```



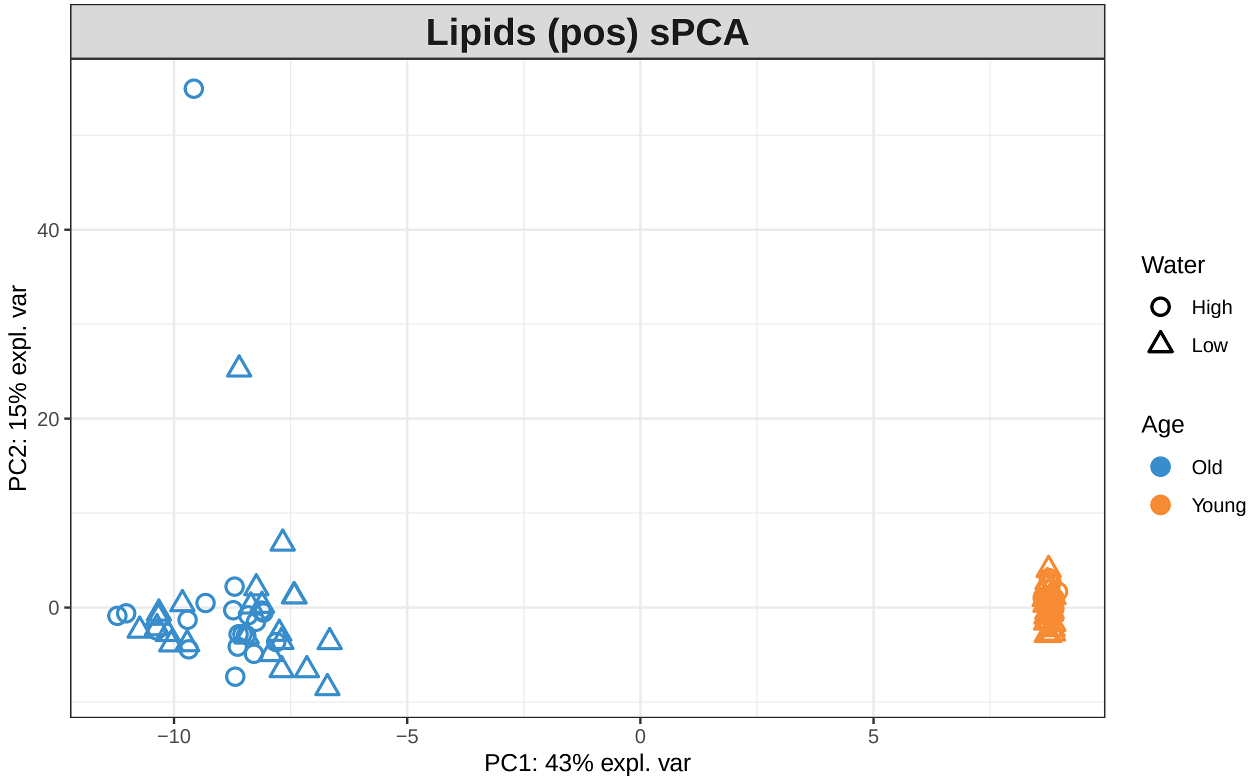
```r
# Look at variable coefficients in each component with the
# loading vectors The absolute value of loading vectors
# represent the importance of each variable to define each PC
plotLoadings(pca.res, ndisplay = 50)
```



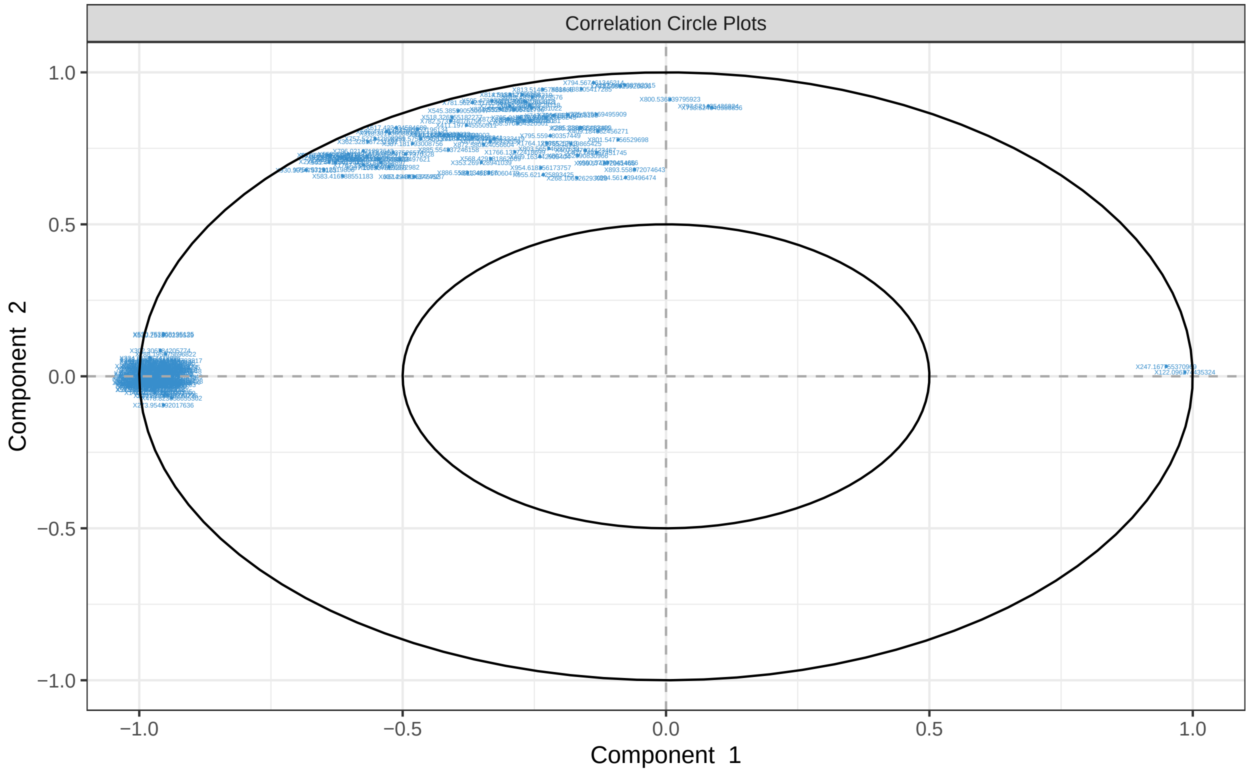
9. Identify the variables that contribute to the explained variance.

```r
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 = "Lipids (pos) sPCA")
```



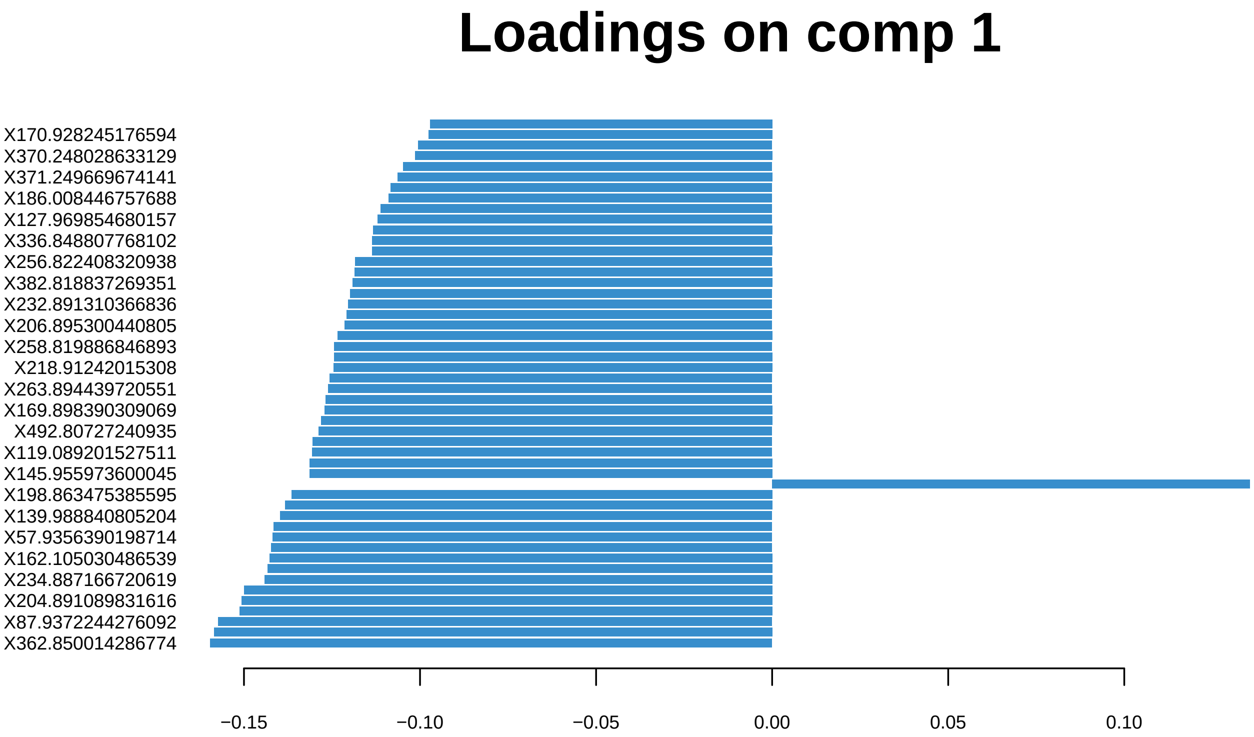
```r
# variables contributing to each component
plotVar(spca.res, cex = 1)
```



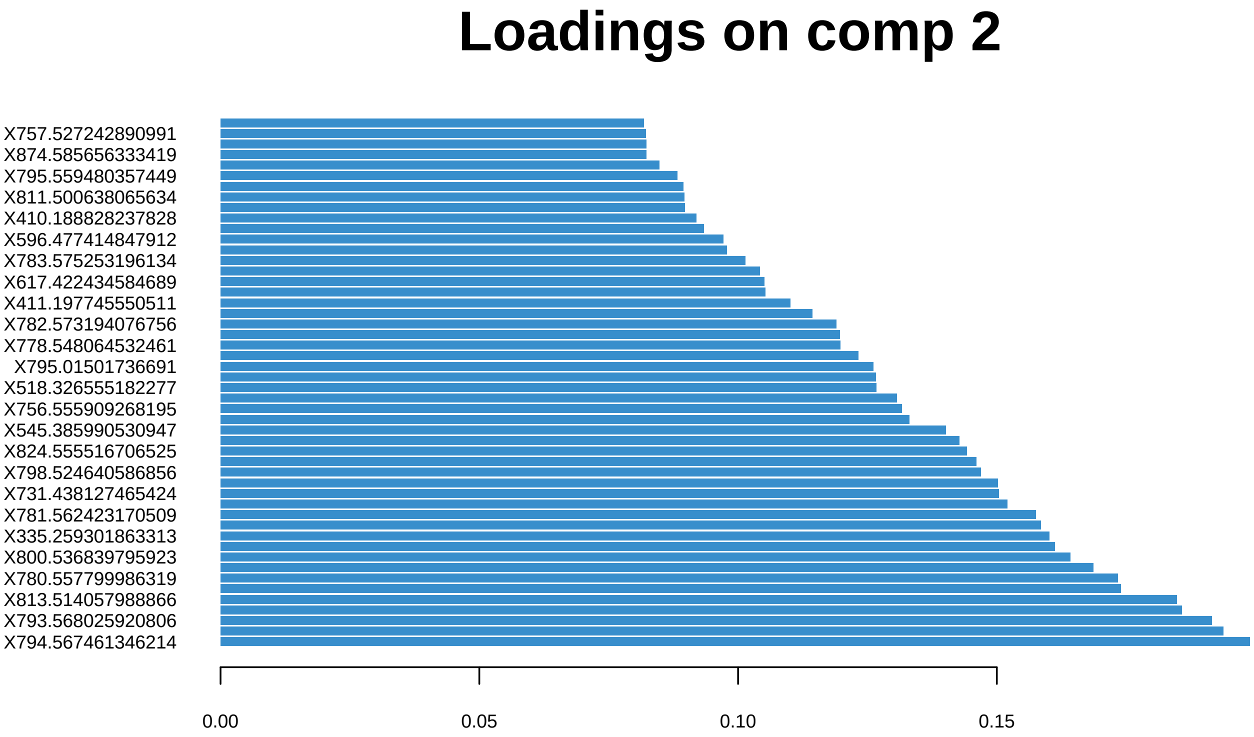
```
selectVar(spca.res, comp = 1)$value  # view loading value of each metabolite
```

```
##                       value.var
## X362.850014286774 -0.159573030
## X224.863582325554 -0.158537083
## X87.9372244276092 -0.157314059
## X83.9387409142577 -0.151246928
## X204.891089831616 -0.150671522
## X104.956055556393 -0.150021880
## X234.887166720619 -0.144167638
## X364.845496774838 -0.143319955
## X162.105030486539 -0.142753008
## X58.9434318608846 -0.142276355
## X57.9356390198714 -0.141822989
## X111.944805605701 -0.141513297
## X139.988840805204 -0.139681259
## X215.920069997245 -0.138354651
## X198.863475385595 -0.136531521
## X122.096274435324  0.135605641
## X145.955973600045 -0.131373201
## X233.892360347796 -0.131354692
## X119.089201527511 -0.130617949
## X265.89200225762  -0.130450927
## X492.80727240935  -0.128739481
## X196.866479266089 -0.128124070
## X169.898390309069 -0.127124323
## X59.9312918202218 -0.126757009
## X263.894439720551 -0.126056617
## X229.894984107501 -0.125621545
## X218.91242015308  -0.124611724
## X102.956529589348 -0.124403580
## X258.819886846893 -0.124375879
## X222.866646970389 -0.123411557
## X206.895300440805 -0.121369323
## X334.850567418996 -0.120785191
## X232.891310366836 -0.120391355
## X464.808402696912 -0.119839281
## X382.818837269351 -0.119173468
## X289.180837502032 -0.118599785
## X256.822408320938 -0.118373789
## X254.826170847043 -0.113601319
## X336.848807768102 -0.113564953
## X158.890661102482 -0.113387345
## X127.969854680157 -0.112018397
## X85.9412932684143 -0.111146257
## X186.008446757688 -0.108860583
## X123.893171681799 -0.108301109
## X371.249669674141 -0.106396937
## X156.890577468201 -0.104780693
## X370.248028633129 -0.101438611
## X86.9440336355191 -0.100487238
## X170.928245176594 -0.097613690
## X148.982997707158 -0.097202017
## X354.821653580704 -0.096413148
## X140.917722269277 -0.095744051
## X474.834862701709 -0.095682167
## X453.284246619693 -0.095127806
## X345.877392601453 -0.093330319
## X138.105500223899 -0.089226065
## X348.868132910006 -0.087124186
## X494.800996824241 -0.081949944
## X322.860984012415 -0.081887229
## X126.961546086716 -0.077483116
## X107.967388011407 -0.075506006
## X137.107830584556 -0.074423474
## X330.206201096997 -0.072905820
## X353.221795947634 -0.071444326
## X207.131144750374 -0.071267808
## X302.306784205774 -0.068643639
## X272.159521514782 -0.065818522
## X352.825807372581 -0.065374331
## X168.891353596568 -0.064573741
## X300.877382508945 -0.063914011
## X466.806284190184 -0.062405639
## X318.877519821346 -0.059986219
## X101.964130868905 -0.059983805
## X197.897681455234 -0.056076634
## X258.156149055828 -0.055752942
## X376.822405750392 -0.054987556
## X530.251890235539 -0.052339418
## X273.954392017636 -0.052032230
## X530.753768195125 -0.051727987
## X240.882199623023 -0.048861987
## X380.822059399454 -0.048579129
## X67.935667299925  -0.048022211
## X137.964613388296 -0.047945128
## X346.874781852769 -0.043863503
## X493.327689927145 -0.043750735
## X604.792120598703 -0.041949869
## X288.195075696822 -0.040206252
## X247.167755370969  0.039744714
## X448.834937696658 -0.037621213
## X80.9485263101571 -0.037184006
## X116.976270866716 -0.033789459
## X246.863710252474 -0.032831439
## X271.168846079276 -0.030851987
## X178.124738807255 -0.019670348
## X237.886222318036 -0.017236564
## X282.906299970548 -0.012362719
## X478.823058655302 -0.009662144
## X622.763654783817 -0.008761364
## X220.866377230149 -0.007083675
## X125.985452816428 -0.005493957
```

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



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

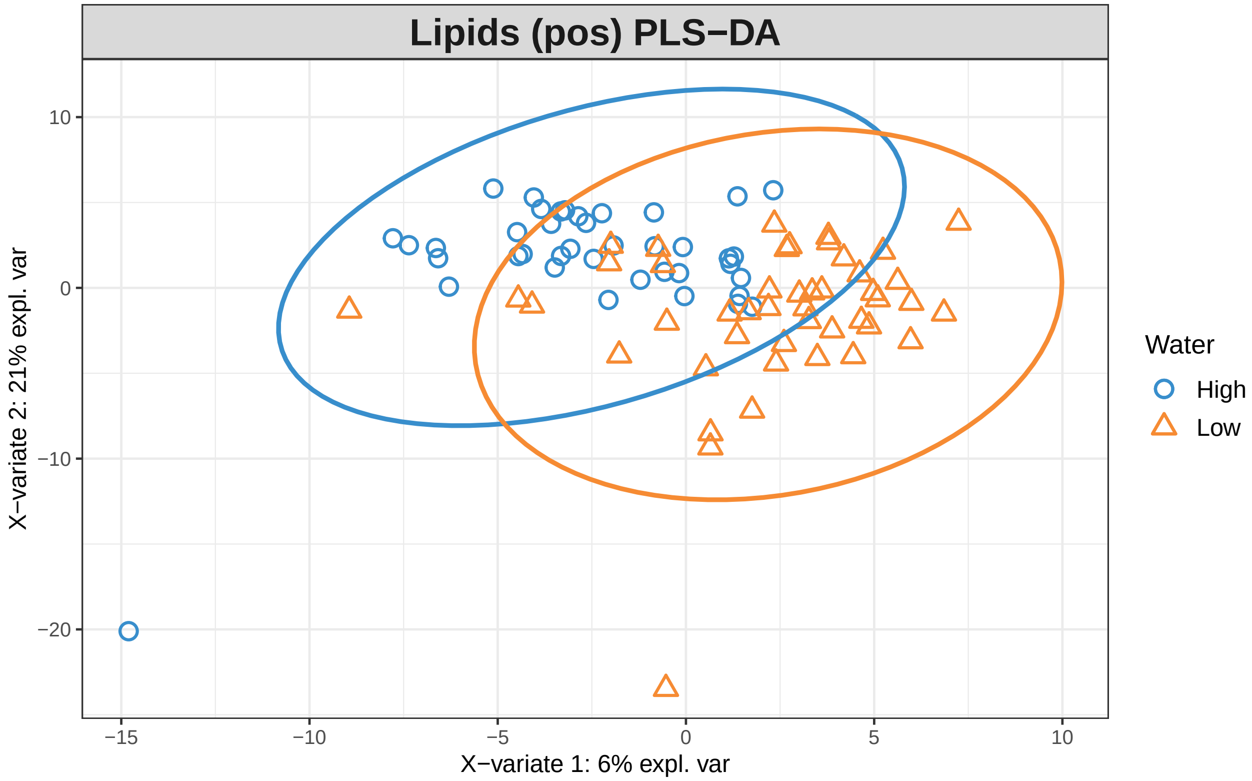


PLS-DA

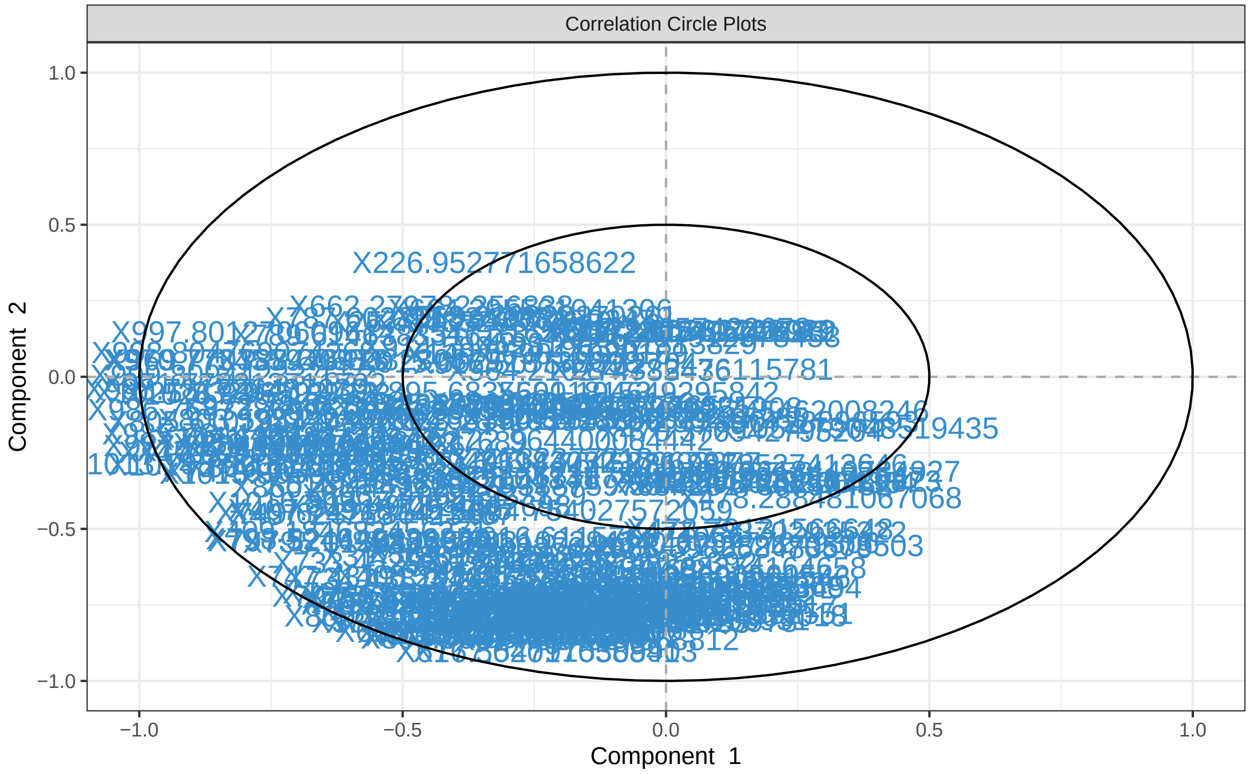
10. Classify samples into known groups and predict the class of new samples.

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

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



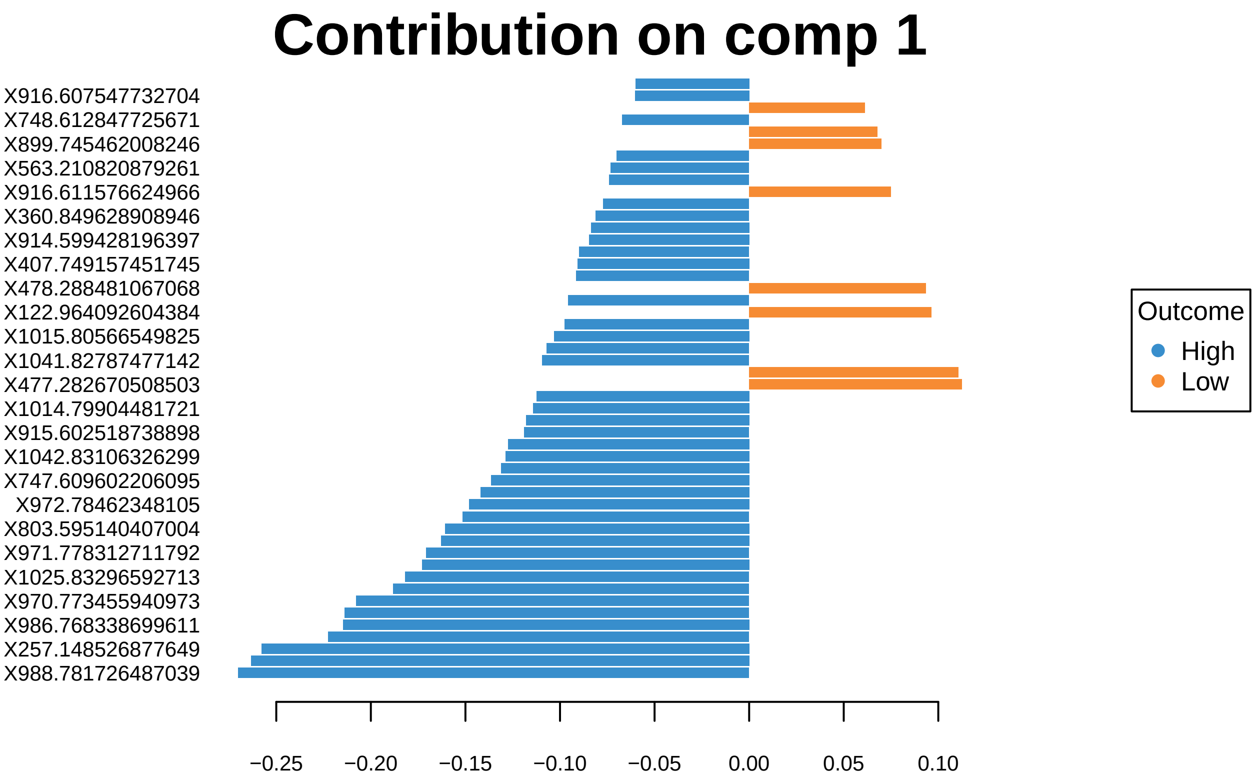
```
# plot and select the variables
plotVar(Lpos.splsda)
```



```r
selectVar(Lpos.splsda, comp = 1)
```

```
## $name
##   [1] "X988.781726487039" "X985.764446034175" "X257.148526877649"
##   [4] "X969.770337476416" "X986.768338699611" "X859.490920549613"
##   [7] "X970.773455940973" "X860.494441129175" "X1025.83296592713"
##  [10] "X999.809442337652" "X971.778312711792" "X1013.79578182787"
##  [13] "X803.595140407004" "X998.804769074268" "X972.78462348105"
##  [16] "X997.801270609614" "X747.609602206095" "X226.952771658622"
##  [19] "X1042.83106326299" "X662.279782256838" "X915.602518738898"
##  [22] "X813.493980957941" "X1014.79904481721" "X1026.83631383334"
##  [25] "X477.282670508503" "X917.616369656058" "X1041.82787477142"
##  [28] "X545.200371712615" "X1015.80566549825" "X748.61381927689"
##  [31] "X122.964092604384" "X140.069011020169" "X478.288481067068"
##  [34] "X747.610443830783" "X407.749157451745" "X407.247634423467"
##  [37] "X914.599428196397" "X814.497324612698" "X360.849628908946"
##  [40] "X787.602325834203" "X916.611576624966" "X987.776110169299"
##  [43] "X563.210820879261" "X805.610194434876" "X899.745462008246"
##  [46] "X932.633245807593" "X748.612847725671" "X593.278328326486"
##  [49] "X916.607547732704" "X804.600154179136" "X425.163750126439"
##  [52] "X940.605777249306" "X933.636354093017" "X594.281739759452"
##  [55] "X593.222551313829" "X795.01501736691"  "X214.918533541367"
##  [58] "X939.598916995568" "X230.890728896593" "X313.275324638018"
##  [61] "X547.216517172547" "X798.524640586856" "X934.642210652551"
##  [64] "X797.521425436924" "X477.783702935322" "X663.284203918693"
##  [67] "X81.9378527413646" "X788.606533230879" "X564.216274785436"
##  [70] "X230.890969151908" "X98.9618440734927" "X1065.51387458946"
##  [73] "X947.08374164658"  "X960.662949353494" "X908.539401143237"
##  [76] "X546.840523796761" "X317.180180129257" "X470.370942795204"
##  [79] "X122.964400084442" "X409.184682456271" "X680.291006850247"
##  [82] "X591.500065461815" "X478.79131566648"  "X381.1878733662"
##  [85] "X817.614863765478" "X592.503436573222" "X491.29800931546"
##  [88] "X875.710897377765" "X594.764527603877" "X1054.79048519435"
##  [91] "X963.598358483579" "X561.194649295842" "X806.613998928281"
##  [94] "X327.162058971025" "X1016.80997406356" "X664.295561041306"
##  [97] "X250.863856229823" "X937.589007902468" "X619.530788020978"
## [100] "X630.510279177547"
##
## $value
##                        value.var
## X988.781726487039 -0.2700789888
## X985.764446034175 -0.2633306164
## X257.148526877649 -0.2577584036
## X969.770337476416 -0.2225189574
## X986.768338699611 -0.2147416951
## X859.490920549613 -0.2137528732
## X970.773455940973 -0.2077093445
## X860.494441129175 -0.1881492238
## X1025.83296592713 -0.1818439624
## X999.809442337652 -0.1729421342
## X971.778312711792 -0.1706920266
## X1013.79578182787 -0.1629744407
## X803.595140407004 -0.1607893109
## X998.804769074268 -0.1514110946
## X972.78462348105  -0.1481852626
## X997.801270609614 -0.1420238880
## X747.609602206095 -0.1365073591
## X226.952771658622 -0.1311304624
## X1042.83106326299 -0.1288435620
## X662.279782256838 -0.1275267104
## X915.602518738898 -0.1189077259
## X813.493980957941 -0.1180217153
## X1014.79904481721 -0.1142540772
## X1026.83631383334 -0.1123947729
## X477.282670508503  0.1123932686
## X917.616369656058  0.1107058961
## X1041.82787477142 -0.1094341383
## X545.200371712615 -0.1069719242
## X1015.80566549825 -0.1031263669
## X748.61381927689  -0.0974973451
## X122.964092604384  0.0963835323
## X140.069011020169 -0.0956475439
## X478.288481067068  0.0934466423
## X747.610443830783 -0.0914464222
## X407.749157451745 -0.0907993565
## X407.247634423467 -0.0898480494
## X914.599428196397 -0.0847324506
## X814.497324612698 -0.0836365922
## X360.849628908946 -0.0811479911
## X787.602325834203 -0.0771246629
## X916.611576624966  0.0749049424
## X987.776110169299 -0.0739161581
## X563.210820879261 -0.0731678106
## X805.610194434876 -0.0700088434
## X899.745462008246  0.0700062587
## X932.633245807593  0.0678998427
## X748.612847725671 -0.0670702275
## X593.278328326486  0.0611621846
## X916.607547732704 -0.0603935533
## X804.600154179136 -0.0600331515
## X425.163750126439 -0.0589003045
## X940.605777249306  0.0535302573
## X933.636354093017  0.0531349839
## X594.281739759452  0.0527232900
## X593.222551313829 -0.0523446001
## X795.01501736691  -0.0498591346
## X214.918533541367 -0.0480719368
## X939.598916995568  0.0479538631
## X230.890728896593  0.0463028375
## X313.275324638018  0.0458973511
## X547.216517172547 -0.0443797294
## X798.524640586856 -0.0416831552
## X934.642210652551  0.0407034377
## X797.521425436924 -0.0397980004
## X477.783702935322  0.0376183251
## X663.284203918693 -0.0359224300
## X81.9378527413646  0.0347044519
## X788.606533230879 -0.0338421553
## X564.216274785436 -0.0320241638
## X230.890969151908 -0.0315072562
## X98.9618440734927  0.0294337716
## X1065.51387458946 -0.0265753866
## X947.08374164658   0.0253963842
## X960.662949353494  0.0230431739
## X908.539401143237 -0.0212266060
## X546.840523796761 -0.0211929299
## X317.180180129257 -0.0190508926
## X470.370942795204  0.0179440685
## X122.964400084442 -0.0176531316
## X409.184682456271 -0.0166744647
## X680.291006850247 -0.0139899087
## X591.500065461815  0.0139437805
## X478.79131566648   0.0139264852
## X381.1878733662   -0.0121734258
## X817.614863765478 -0.0099708973
## X592.503436573222  0.0073777539
## X491.29800931546   0.0065877943
## X875.710897377765 -0.0059947304
## X594.764527603877 -0.0055518658
## X1054.79048519435  0.0054330944
## X963.598358483579  0.0039941573
## X561.194649295842 -0.0036503611
## X806.613998928281 -0.0027902458
## X327.162058971025 -0.0025541355
## X1016.80997406356 -0.0022724579
## X664.295561041306 -0.0012799066
## X250.863856229823  0.0010761083
## X937.589007902468  0.0009047069
## X619.530788020978  0.0004483768
## X630.510279177547  0.0003168851
##
## $comp
## [1] 1
```

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



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$Age),
    FUN = "mean", simplify = T, data = class)
av.plsda <- mixOmics::plsda(av_Y[, 4:686], av_Y$Group.2)  # fungus

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

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                 =  46.6746746746747
##
## Sum of Indicator Values              =  548.85
##
## Sum of Significant Indicator Values  =  504.42
##
## Number of Significant Indicators     =  589
##
## Significant Indicator Distribution
##
##   1   2
## 541  48
```

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/Lipids Statis
```

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.