MIXOMICS - R WORKSHOP - MACQUARIE UNIVERSITY

Pat Buerger (patrick.buerger@mq.edu.au)

07-09-2023

Contents

1 INTRODUCTION

Abstract:

MixOmics example: Case Study of DIABLO with Breast TCGA Dataset, using PCA, PLS-DA (supervised), sparse PLS-DA (supervised + data reduction), and multiblock sparse PLS-DA (DIABLO).

Modified code from MixOmics R package

Source for R-code: https://mixomicsteam.github.io/mixOmics-Vignette/

1.1 Load Packages

```
#install.packages('markdown')
#install.packages('mixOmics')
#install.packages('tidyverse')
library(mixOmics)
library(tidyverse)
```

1.2 Load Data

```
# Load data
data(breast.TCGA)

# Extract training data and name each data frame
# Store as list
X <- list(mRNA = breast.TCGA$data.train$mrna,
          miRNA = breast.TCGA$data.train$mirna,
          protein = breast.TCGA$data.train$protein)

# Outcome
Y <- breast.TCGA$data.train$subtype
summary(Y)
```

```
## Basal  Her2  LumA
##    45    30    75
```

2 Principal Component Analysis (PCA)

```
# Check data first with PCA to detect outliers and general structure of data.
# 1 Run the method
MyResult.pca_miRNA <- pca(X$miRNA)

MyResult.pca_mRNA <- pca(X$mRNA)

MyResult.pca_protein <- pca(X$protein)
```

```
## 2.1 Plot the samples
## miRNA
# PLOT WITH ADDITIONAL SYMBOLS
plotIndiv(MyResult.pca_miRNA, ind.names = FALSE,
          group = Y,
          pch = as.factor(Y),
          legend = TRUE, title = 'miRNA: PCA comp 1 - 2',
          legend.title = 'miRNA', legend.title.pch = 'miRNA')
```

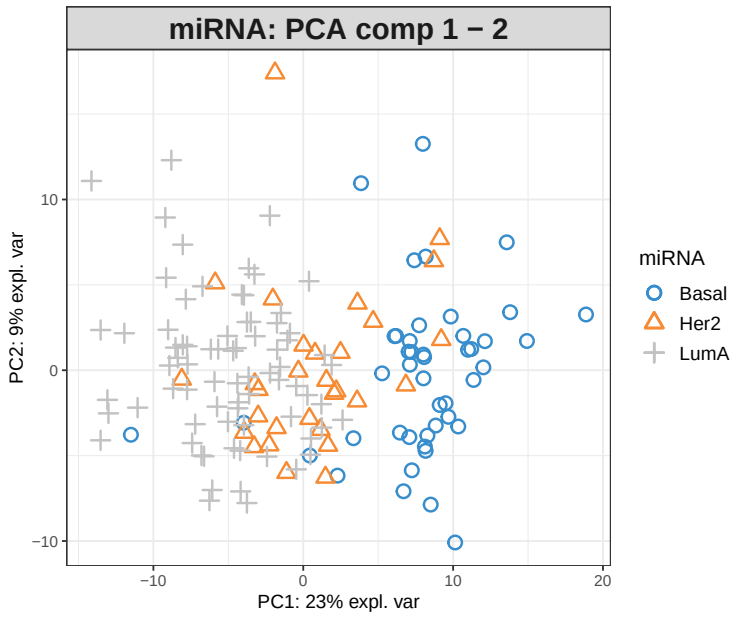

FIGURE: miRNA PCA: comp 1 - 2

```
## 2.2 Plot the samples
## mRNA
# PLOT WITH ADDITIONAL SYMBOLS
plotIndiv(MyResult.pca_mRNA, ind.names = FALSE,
          group = Y,
          pch = as.factor(Y),
          legend = TRUE, title = 'mRNA: PCA comp 1 - 2',
          legend.title = 'mRNA', legend.title.pch = 'mRNA')
```

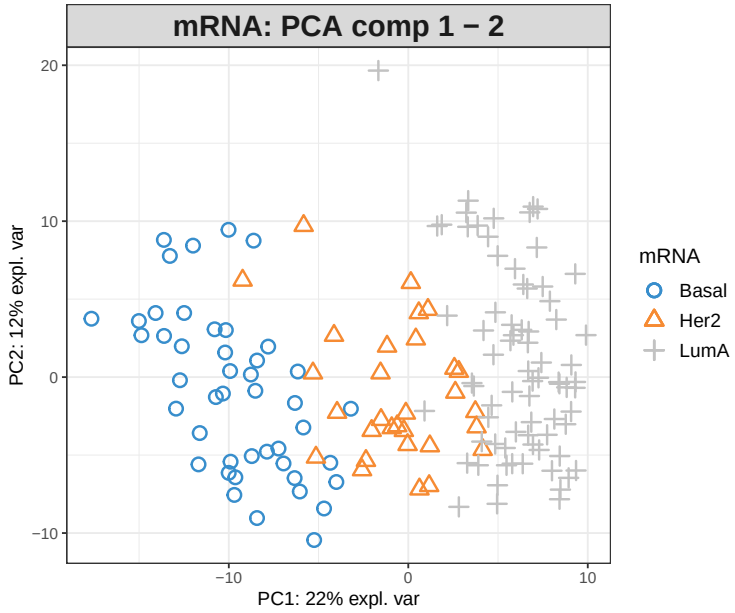

FIGURE: mRNA PCA: comp 1 - 2

```
## 2.3 Plot the samples
## protein
# PLOT WITH ADDITIONAL SYMBOLS
plotIndiv(MyResult.pca_protein, ind.names = FALSE,
          group = Y,
          pch = as.factor(Y),
          legend = TRUE, title = 'protein: PCA comp 1 - 2',
          legend.title = 'protein', legend.title.pch = 'protein')
```

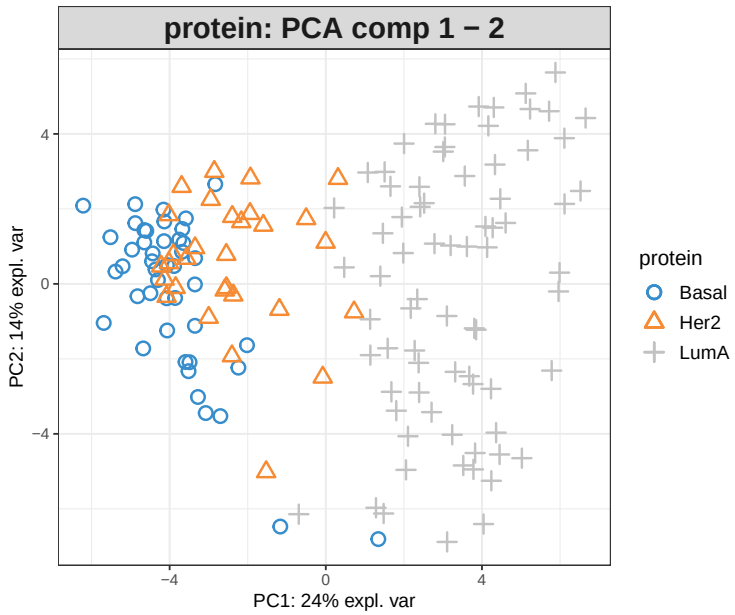

FIGURE: protein PCA: comp 1 - 2

2.0.0.1 PCA: Conclusion Data looks all right. No outliers detected. Seems to be some grouping of samples present. -> Problem: There is some source of variation in the data that we can not explain with our strain groupings. -> We need to have some sort of supervised analysis to resolve this.

3 Partial Least Squares – Discriminant Analysis (PLS-DA)

```
##### PLS-DA, sPLS-DA ####
# 1 Run the method
MyResult.plsda_miRNA <- plsda(X$miRNA, Y)

MyResult.plsda_mRNA <- plsda(X$mRNA, Y)

MyResult.plsda_protein <- plsda(X$protein, Y)
```

```
# sPLS-DA NEEDS TO BE TUNED AND NUMBERS OF COMPONENTS OPTIMISED. SKIPPED HERE DUE TO TIME CONSTRAINTS. S
#https://mixomicsteam.github.io/mixOmics-Vignette/id_05.html

# 2.1 Plot the samples
plotIndiv(MyResult.plsda_miRNA, ind.names = FALSE,
          group = Y,
          pch = as.factor(Y),
          legend = TRUE, title = 'miRNA: PLS-DA comp 1 - 2',
          legend.title = 'miRNA', legend.title.pch = 'miRNA')
```

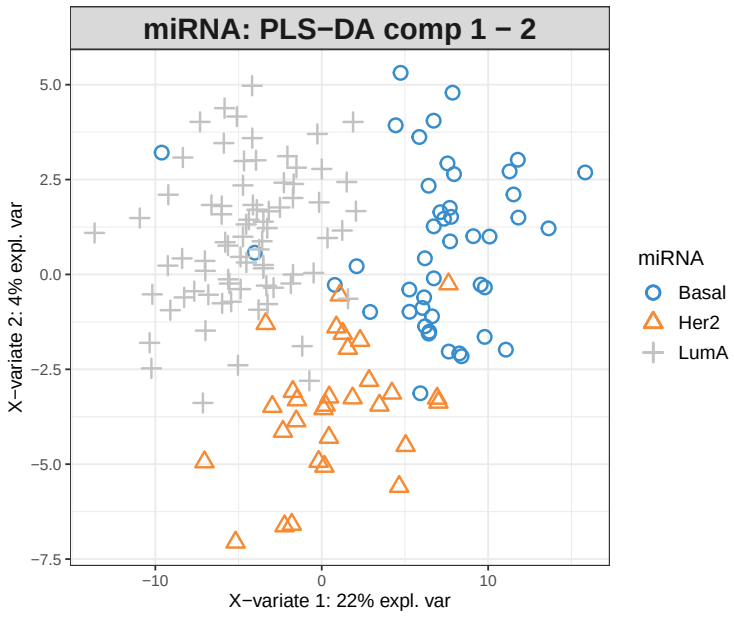

FIGURE: miRNA PLS-DA: comp 1 - 2, supervised analysis using the full data.

```
# sPLS-DA NEEDS TO BE TUNED AND NUMBERS OF COMPONENTS OPTIMISED. SKIPPED HERE DUE TO TIME CONSTRAINTS. S
#https://mixomicsteam.github.io/mixOmics-Vignette/id_05.html

# 2.2 Plot the samples
```

```
plotIndiv(MyResult.plsda_mRNA, ind.names = FALSE,
          group = Y,
          pch = as.factor(Y),
          legend = TRUE, title = 'mRNA: PLS-DA comp 1 - 2',
          legend.title = 'mRNA', legend.title.pch = 'mRNA')
```

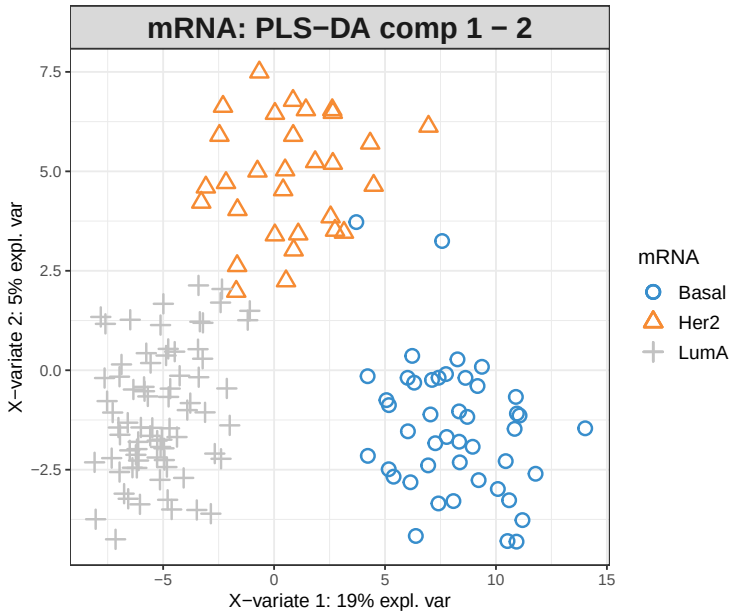

FIGURE: mRNA PLS-DA: comp 1 - 2, supervised analysis using the full data.

```
# sPLS-DA NEEDS TO BE TUNED AND NUMBERS OF COMPONENTS OPTIMISED. SKIPPED HERE DUE TO TIME CONSTRAINTS. S
#https://mixomicsteam.github.io/mixOmics-Vignette/id_05.html

# 2.3 Plot the samples
plotIndiv(MyResult.plsda_protein, ind.names = FALSE,
          group = Y,
          pch = as.factor(Y),
          legend = TRUE, title = 'protein: PLS-DA comp 1 - 2',
          legend.title = 'protein', legend.title.pch = 'protein')
```

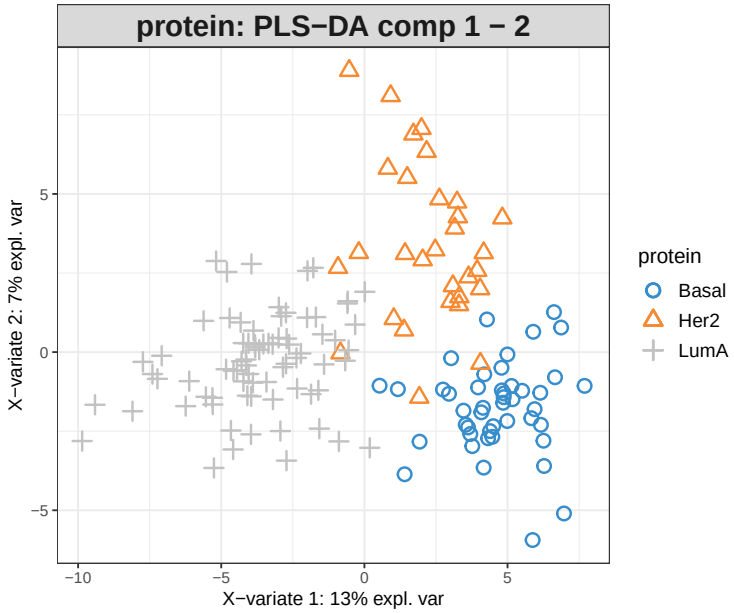

FIGURE: protein PLS-DA: comp 1 - 2, supervised analysis using the full data.

```
# 3.1 Plot the variables
plotVar(MyResult.plsda_miRNA)
```

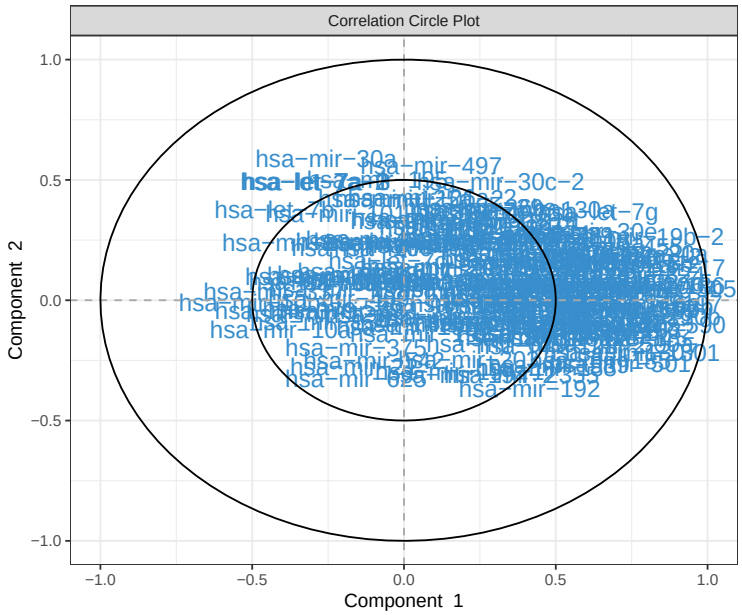

FIGURE: miRNA: PLS-DA: contributing variables to PLS-DA comp 1 - 2, supervised analysis using the full data.

```
# 3.2 Plot the variables
plotVar(MyResult.plsda_mRNA)
```

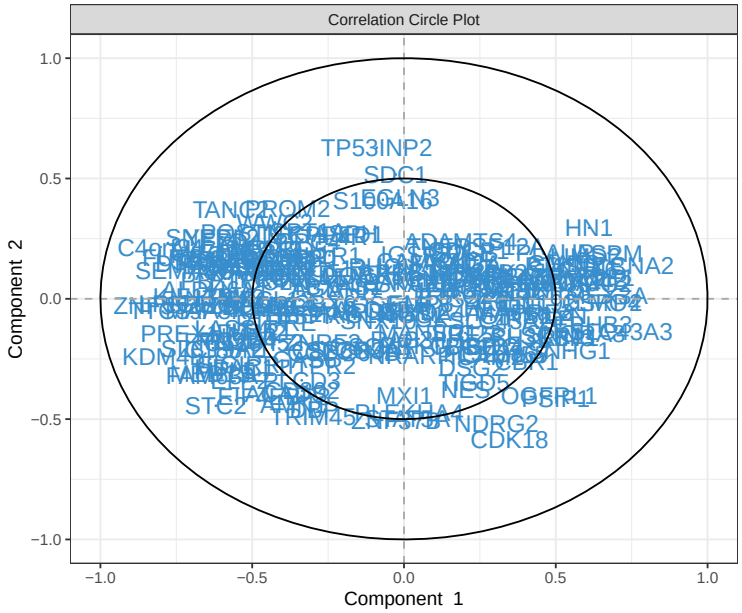

FIGURE: mRNA: PLS-DA: contributing variables to PLS-DA comp 1 - 2, supervised analysis using the full data.

```
# 3.3 Plot the variables
plotVar(MyResult.plsda_protein)
```

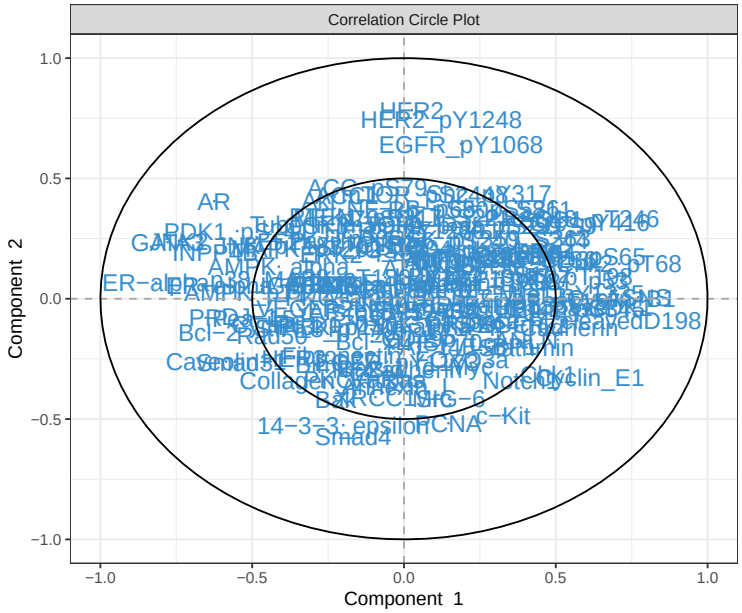

FIGURE: Protein: PLS-DA: contributing variables to PLS-DA comp 1 - 2, supervised analysis using the full data.

3.0.0.1 PLS-DA: Conclusion Lots of variable and some noise in the data. Data reduction might be beneficial using a sparse PLS-DA in order to separate groups and get a clear picture of the data and important variables.

4 sparse Partial Least Squares – Discriminant Analysis (sPLS-DA)

```
# DATA REDUCTION APPROACH.
#########
# miRNA
#########

# sPLS-DA NEEDS TO BE TUNED AND NUMBERS OF COMPONENTS OPTIMISED. SKIPPED HERE DUE TO TIME CONSTRAINTS. S
#https://mixomicsteam.github.io/mixOmics-Vignette/id_05.html

#### sPLS-DA, using reduced data, TOP 15 variables
MyResult.splsda.miRNA <- splsda(X$miRNA, Y, keepX = c(15,15)) # 1 Run the method

plotIndiv(MyResult.splsda.miRNA)                    # 2 Plot the samples
```

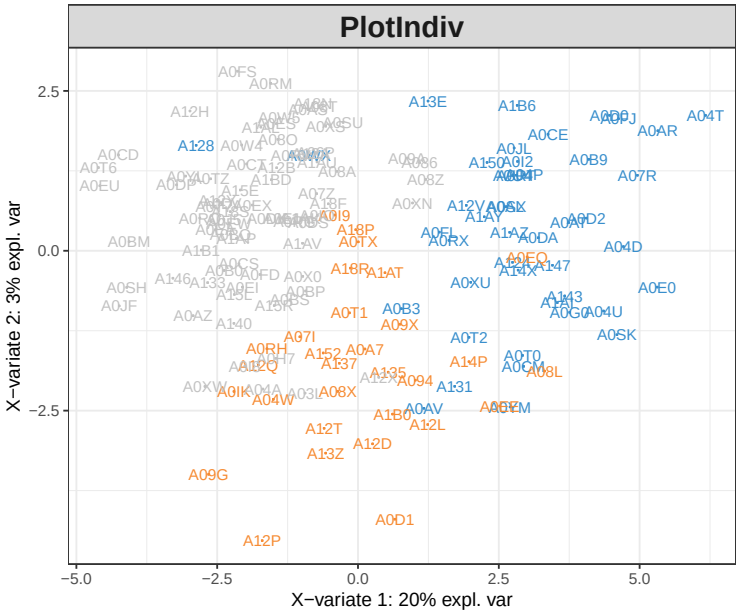

FIGURE: miRNA sparse PLS-DA: comp 1 - 2, supervised analysis using the only the most influential variables, here TOP 15.

```
plotVar(MyResult.splsda.miRNA)                    # 3 Plot the variables
```

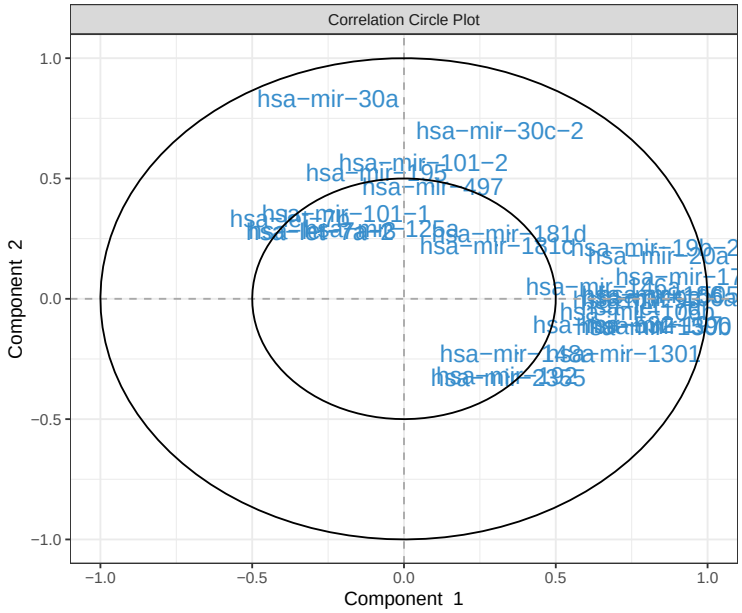

FIGURE: miRNA contributing variables to sparse PLS-DA: comp 1 - 2, supervised analysis using the only the most influential variables, here TOP 15.

```
selectVar(MyResult.splsda.miRNA, comp = 1)$value          # Selected variables on comp 1
```

```
##                value.var
## hsa-mir-17    0.53512288
## hsa-mir-505   0.41390204
## hsa-mir-590   0.40421610
## hsa-mir-130b  0.33995192
## hsa-mir-20a   0.30646439
## hsa-mir-106a  0.27838707
## hsa-mir-106b  0.25493987
## hsa-mir-186   0.10531789
## hsa-mir-197   0.08915897
## hsa-mir-1301  0.06329850
## hsa-let-7d    0.03813361
## hsa-mir-93    0.03323711
## hsa-mir-146a  0.02821574
## hsa-mir-19b-2 0.01874094
## hsa-mir-532   0.01422720
```

```
plotLoadings(MyResult.splsda.miRNA, comp = 1, method = 'mean', contrib = 'max')
```

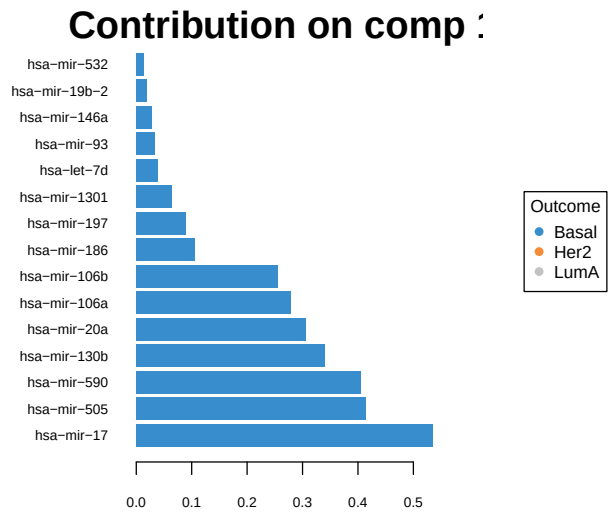

FIGURE: miRNA contributing variables to sparse PLS-DA: comp 1.

```
selectVar(MyResult.splsda.miRNA, comp = 2)$value          # Selected variables on comp 1
```

```
##                   value.var
## hsa-mir-30a     0.731118807
## hsa-mir-30c-2   0.394949283
## hsa-mir-101-2   0.330579061
## hsa-mir-101-1   0.213529839
## hsa-let-7b      0.194235330
## hsa-mir-497     0.187334933
## hsa-mir-125a    0.133615000
## hsa-mir-195     0.130327422
## hsa-mir-192    -0.123881223
## hsa-mir-2355   -0.111398761
## hsa-mir-148a   -0.103392290
## hsa-mir-181c    0.082626636
## hsa-mir-181d    0.037126166
## hsa-let-7a-3    0.016992438
## hsa-let-7a-2    0.001561175
```

```
plotLoadings(MyResult.splsda.miRNA, comp = 2, method = 'mean', contrib = 'max')
```

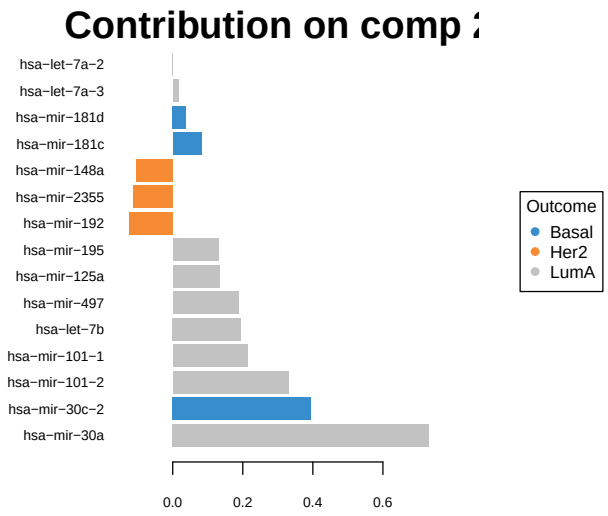

FIGURE: miRNA contributing variables to sparse PLS-DA: comp 2.

```
# DATA REDUCTION APPROACH.
#########
# mRNA
#########

# sPLS-DA NEEDS TO BE TUNED AND NUMBERS OF COMPONENTS OPTIMISED. SKIPPED HERE DUE TO TIME CONSTRAINTS. S
#https://mixomicsteam.github.io/mixOmics-Vignette/id_05.html


#### sPLS-DA, using reduced data, TOP 15 variables
MyResult.splsda.mRNA <- splsda(X$mRNA, Y, keepX = c(15,15)) # 1 Run the method

plotIndiv(MyResult.splsda.mRNA)                    # 2 Plot the samples
```

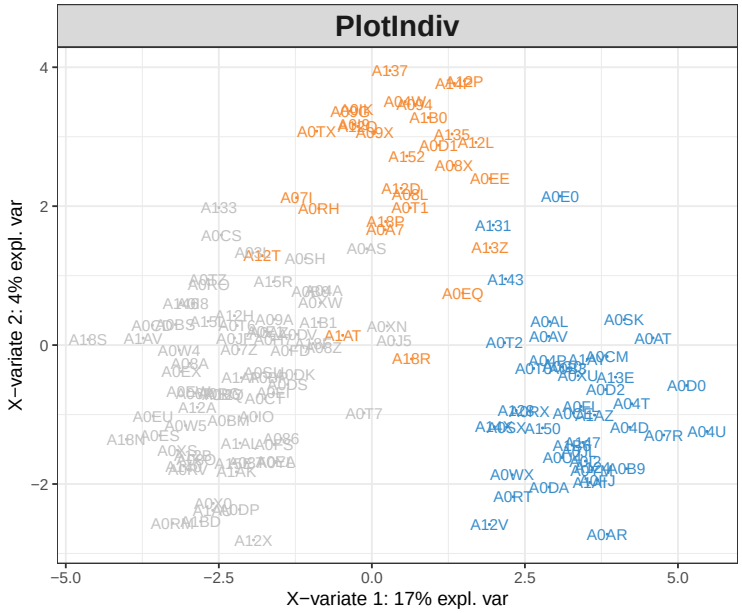

FIGURE: mRNA sparse PLS-DA: comp 1 - 2, supervised analysis using the only the most influential variables, here TOP 15.

```
plotVar(MyResult.splsda.mRNA)                    # 3 Plot the variables
```

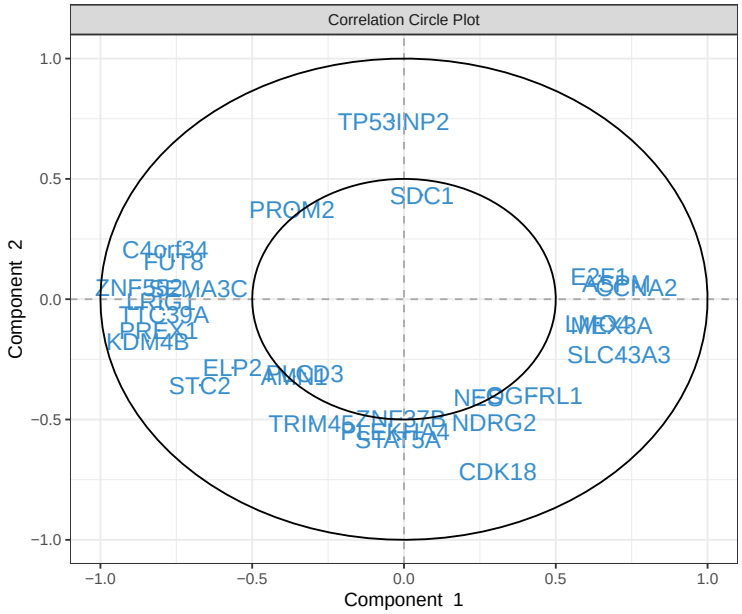

FIGURE: mRNA contributing variables to sparse PLS-DA: comp 1 - 2, supervised analysis using the only the most influential variables, here TOP 15.

```
selectVar(MyResult.splsda.mRNA, comp = 1)$value          # Selected variables on comp 1
```

```
##           value.var
```

```
## ZNF552  -0.50951316
## KDM4B   -0.42377034
## LRIG1   -0.32072098
## PREX1   -0.31919977
## CCNA2    0.28423322
## TTC39A  -0.26899129
## C4orf34 -0.26022547
## FUT8    -0.24803089
## ASPM     0.14511409
## MEX3A    0.13460957
## SLC43A3  0.13183948
## SEMA3C  -0.12219842
## STC2    -0.03245939
## LMO4     0.02781010
## E2F1     0.01908383
```

```
plotLoadings(MyResult.splsda.mRNA, comp = 1, method = 'mean', contrib = 'max')
```

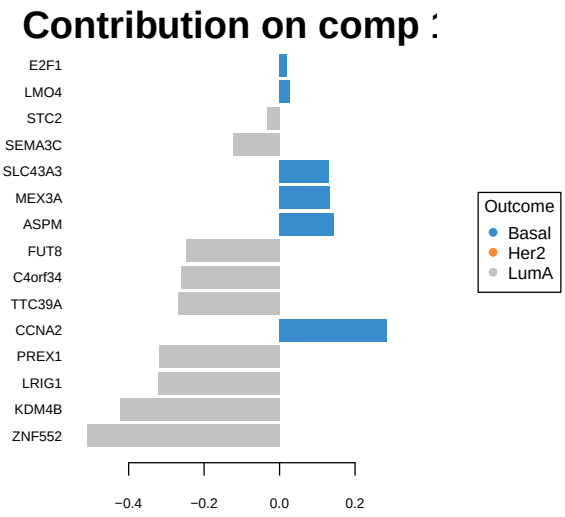

FIGURE: mRNA contributing variables to sparse PLS-DA: comp 1.

```
selectVar(MyResult.splsda.mRNA, comp = 2)$value         # Selected variables on comp 1
```

```
##           value.var
## TP53INP2  0.59228590
## CDK18    -0.54781654
## TRIM45   -0.30488946
## NDRG2    -0.27745506
## STAT5A   -0.25011165
## PLEKHA4  -0.22742692
## ZNF37B   -0.20035670
## PLCD3    -0.08312372
## OGFRL1   -0.06966381
```

```
## ELP2     -0.06901105
## STC2     -0.05971429
## NES      -0.04906850
## SDC1      0.03447395
## AMN1     -0.03046066
## PROM2     0.01090741
```

```
plotLoadings(MyResult.splsda.mRNA, comp = 2, method = 'mean', contrib = 'max')
```

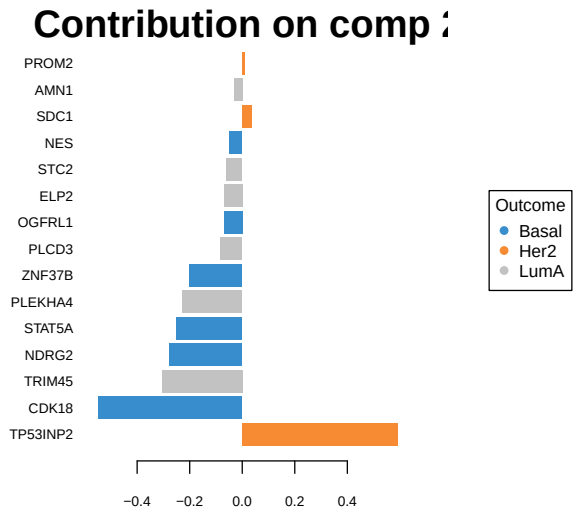

FIGURE: mRNA contributing variables to sparse PLS-DA: comp 2.

```
# DATA REDUCTION APPROACH.
#########
# Protein
#########

# sPLS-DA NEEDS TO BE TUNED AND NUMBERS OF COMPONENTS OPTIMISED. SKIPPED HERE DUE TO TIME CONSTRAINTS. S
#https://mixomicsteam.github.io/mixOmics-Vignette/id_05.html

#### sPLS-DA, using reduced data, TOP 15 variables
MyResult.splsda.protein <- splsda(X$protein, Y, keepX = c(15,15)) # 1 Run the method

plotIndiv(MyResult.splsda.protein)                  # 2 Plot the samples
```

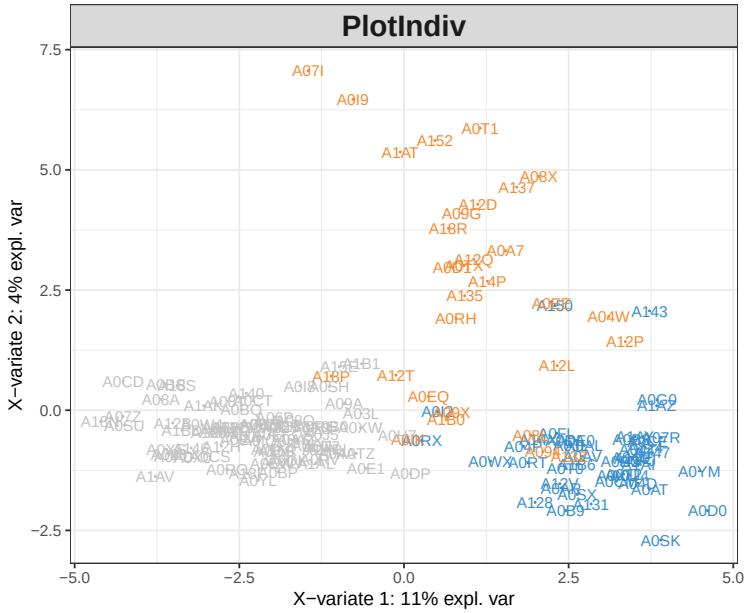

FIGURE: Protein sparse PLS-DA: comp 1 - 2, supervised analysis using the only the most influential variables, here TOP 15.

```
plotVar(MyResult.splsda.protein)                    # 3 Plot the variables
```

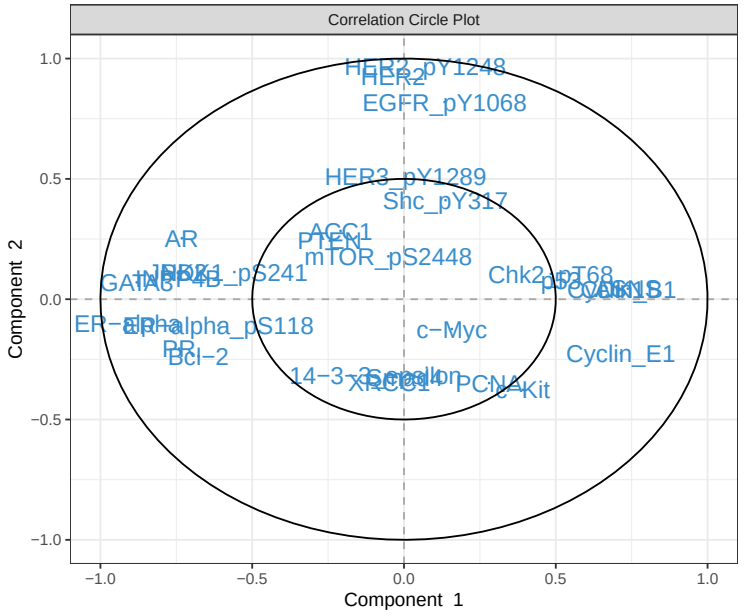

FIGURE: Protein contributing variables to sparse PLS-DA: comp 1 - 2, supervised analysis using the only the most influential variables, here TOP 15.

```
selectVar(MyResult.splsda.protein, comp = 1)$value          # Selected variables on comp 1
```

```
##                 value.var
```

```
## ER-alpha       -0.49448387
## GATA3          -0.44990575
## ASNS            0.31478915
## Cyclin_B1       0.28559119
## PR             -0.27669620
## JNK2           -0.25604971
## AR             -0.24232685
## INPP4B         -0.23251218
## Cyclin_E1       0.21296917
## CDK1            0.18216101
## Bcl-2          -0.17753398
## p53             0.06524030
## Chk2_pT68       0.03984212
## PDK1_pS241     -0.03067046
## ER-alpha_pS118 -0.02585243
```

```
plotLoadings(MyResult.splsda.protein, comp = 1, method = 'mean', contrib = 'max')
```

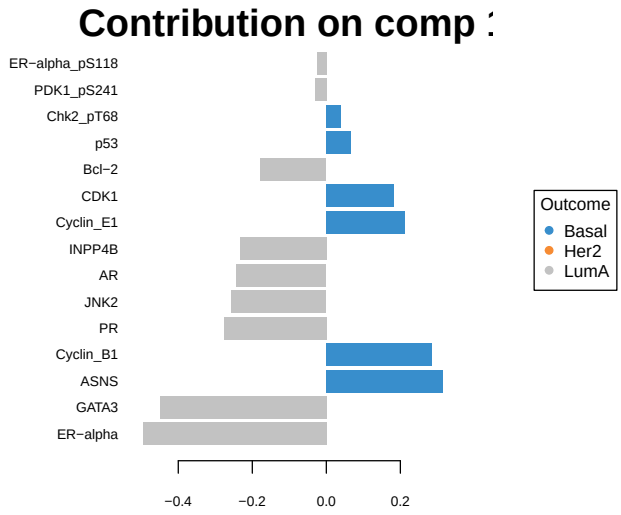

FIGURE: Protein contributing variables to sparse PLS-DA: comp 1.

```
selectVar(MyResult.splsda.protein, comp = 2)$value          # Selected variables on comp 1
```

```
##                 value.var
## HER2            0.638805635
## HER2_pY1248     0.602765329
## EGFR_pY1068     0.367647086
## c-Kit          -0.178052507
## HER3_pY1289     0.154661976
## AR              0.131304702
## XRCC1          -0.110377739
## Smad4          -0.057959733
## PCNA           -0.050270698
```

```
## Shc_pY317       0.034489077
## 14-3-3_epsilon -0.030816673
## c-Myc          -0.013079244
## mTOR_pS2448     0.011728989
## ACC1            0.005450146
## PTEN            0.005076423
```

```
plotLoadings(MyResult.splsda.protein, comp = 2, method = 'mean', contrib = 'max')
```

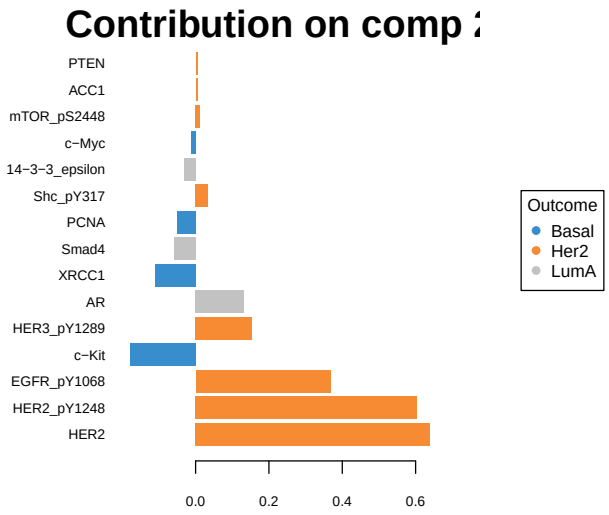

FIGURE: Protein contributing variables to sparse PLS-DA: comp 2.

4.0.0.1 sPLS-DA: Conclusion Data reduction seems beneficial using a sparse PLS-DA in order to separate groups and get a clear picture of the data and important variables. Lets try the integrated data analysis using all three data sets.

5 Multiblock PLS-DA (DIABLO)

5.1 Parameter choice

5.1.1 Design matrix

```
# HERE WE LOOK AT DESIGN MATRIX WITH 0.1.
# A full design with weights = 1 will favour the latter, but at the expense of classification accuracy,
# FOR A FULLY WEIGHTED DESIGN: use "matrix(1, ncol...", WHICH IS FOCUSSING ON EXTRACTING THE MOST CORRE
design <- matrix(0.1, ncol = length(X), nrow = length(X),
                 dimnames = list(names(X), names(X)))
diag(design) <- 0
design
```

```
##         mRNA miRNA protein
## mRNA     0.0   0.1     0.1
## miRNA    0.1   0.0     0.1
## protein  0.1   0.1     0.0
```

```
## ---- results='hold'--------------------------------
# How much correlation between data sets?
pls.res1 <- pls(X$mRNA, X$protein, ncomp = 1)
cor(pls.res1$variates$X, pls.res1$variates$Y)
```

```
##           comp1
## comp1 0.9031761
```

```
pls.res2 <- pls(X$mRNA, X$miRNA, ncomp = 1)
cor(pls.res2$variates$X, pls.res2$variates$Y)
```

```
##           comp1
## comp1 0.8456299
```

```
pls.res3 <- pls(X$protein, X$miRNA, ncomp = 1)
cor(pls.res3$variates$X, pls.res3$variates$Y)
```

```
##           comp1
## comp1 0.7982008
```

```
# Decent amount of correlation between data sets. The data sets taken in a pairwise manner are highly c
```

5.1.2 Number of components

```
## ----diablo-perf, message=FALSE, fig.cap='(ref:diablo-perf)'----
diablo.tcga <- block.plsda(X, Y, ncomp = 5, design = design)

set.seed(123) # For reproducibility, remove for your analyses
```

```
perf.diablo.tcga = perf(diablo.tcga, validation = 'Mfold', folds = 5, nrepeat = 5)
# The validation of the folds parameter was too high and had to be reduced.

# Plot of the error rates based on weighted vote
plot(perf.diablo.tcga)
```

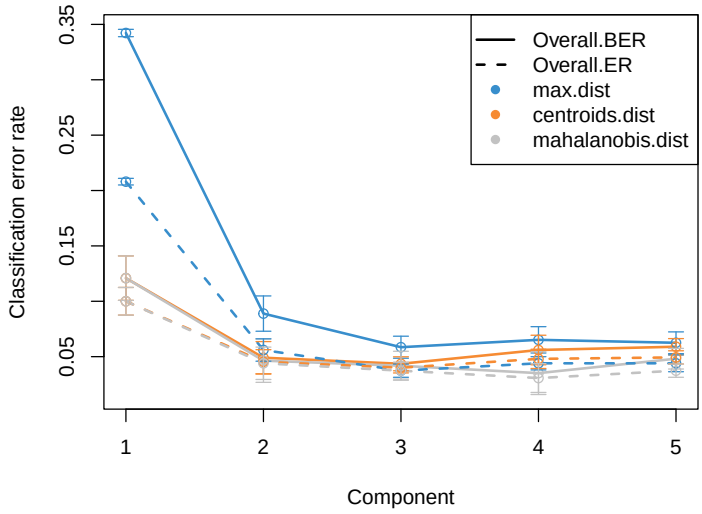

FIGURE: Multi block PLS-DA, error rates: based on weighted vote. Choosing the number of components. Error rate is minimum using how many dimensions?

```
## ------------------------------------------------------
perf.diablo.tcga$choice.ncomp$WeightedVote
```

```
##             max.dist centroids.dist mahalanobis.dist
## Overall.ER         3              2                2
## Overall.BER        3              2                2
```

```
## ------------------------------------------------------
ncomp <- perf.diablo.tcga$choice.ncomp$WeightedVote["Overall.BER", "centroids.dist"]
```

5.1.3 Number of variables to select

```
## ---- echo = FALSE, eval = TRUE, warning=FALSE-------
# chunk takes about 2 min to run
set.seed(123) # for reproducibility
test.keepX <- list(mRNA = c(5:9, seq(10, 25, 5)),
                   miRNA = c(5:9, seq(10, 20, 2)),
                   proteomics = c(seq(5, 25, 5)))

tune.diablo.tcga <- tune.block.splsda(X, Y, ncomp = 2,
```

```
                              test.keepX = test.keepX, design = design,
                              validation = 'Mfold', folds = 2, nrepeat = 1,
                              # use two CPUs for faster computation
                              BPPARAM = BiocParallel::SnowParam(workers = 2),
                              dist = "centroids.dist")


## ----------------------------------------------------
list.keepX <- tune.diablo.tcga$choice.keepX
list.keepX
```

```
## $mRNA
## [1] 10 15
##
## $miRNA
## [1] 14  7
##
## $protein
## [1]  5 20
```

5.2 Final model

```
## ---- message = TRUE----------------------------------
diablo.tcga <- block.splsda(X, Y, ncomp = ncomp,
                            keepX = list.keepX, design = design)
#diablo.tcga   # Lists the different functions of interest related to that object

## ----------------------------------------------------
diablo.tcga$design
```

```
##         mRNA miRNA protein Y
## mRNA     0.0   0.1     0.1 1
## miRNA    0.1   0.0     0.1 1
## protein  0.1   0.1     0.0 1
## Y        1.0   1.0     1.0 0
```

```
## ---- eval = FALSE-----------------------------------
## # mRNA variables selected on component 1
## selectVar(diablo.tcga, block = 'mRNA', comp = 1)
```

5.3 Sample plots

5.3.1 plotDiablo

```
## variables selected on component 1
selectVar(diablo.tcga, block = 'miRNA', comp = 1)$miRNA$name
```

```
##  [1] "hsa-mir-17"   "hsa-mir-590"  "hsa-mir-505"  "hsa-mir-130b" "hsa-mir-106b"
##  [6] "hsa-mir-20a"  "hsa-mir-106a" "hsa-mir-197"  "hsa-mir-1301" "hsa-mir-186"
## [11] "hsa-mir-93"   "hsa-let-7d"   "hsa-mir-532"  "hsa-mir-146a"
```

```
selectVar(diablo.tcga, block = 'mRNA', comp = 1)$mRNA$name
```

```
##  [1] "ZNF552"  "KDM4B"   "CCNA2"   "LRIG1"   "PREX1"   "FUT8"    "C4orf34"
##  [8] "TTC39A"  "ASPM"    "SLC43A3"
```

```
selectVar(diablo.tcga, block = 'protein', comp = 1)$protein$name
```

```
## [1] "ER-alpha"  "GATA3"     "ASNS"      "Cyclin_B1" "AR"
```

```
## ----plot-diablo, message=FALSE, fig.cap='(ref:plot-diablo)'----
plotDiablo(diablo.tcga, ncomp = 1)
```

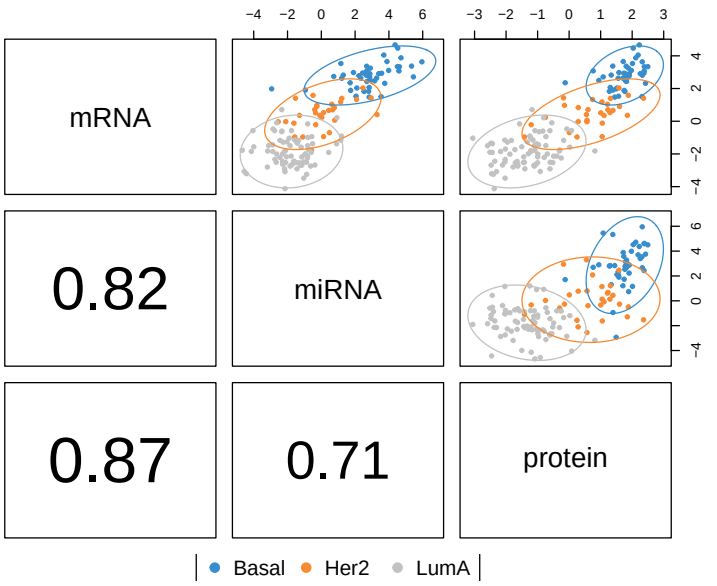

FIGURE: Multi block PLS-DA, diagnostic plot: component 1, 95% confidence intervals are plotted. Numbers indicate correlation coefficients between the first components from each data set.

```
## variables selected on component 1
selectVar(diablo.tcga, block = 'miRNA', comp = 2)$miRNA$name
```

```
## [1] "hsa-mir-30a"   "hsa-mir-30c-2" "hsa-mir-101-2" "hsa-mir-101-1"
## [5] "hsa-mir-192"   "hsa-mir-181c"  "hsa-mir-195"
```

```
selectVar(diablo.tcga, block = 'mRNA', comp = 2)$mRNA$name
```

```
##  [1] "TP53INP2" "CDK18"    "NDRG2"    "TRIM45"   "STAT5A"   "PLEKHA4"
##  [7] "ZNF37B"   "OGFRL1"   "PLCD3"    "SDC1"     "STC2"     "NES"
## [13] "ELP2"     "PROM2"    "PSIP1"
```

```
selectVar(diablo.tcga, block = 'protein', comp = 2)$protein$name
```

```
##  [1] "HER2"           "HER2_pY1248"    "EGFR_pY1068"    "c-Kit"
##  [5] "AR"             "HER3_pY1289"    "XRCC1"          "Smad4"
##  [9] "c-Myc"          "PCNA"           "ACC1"           "Shc_pY317"
## [13] "14-3-3_epsilon" "mTOR_pS2448"    "PTEN"           "Akt"
## [17] "4E-BP1_pS65"    "ACC_pS79"       "Cyclin_E1"      "MIG-6"
```

```
## ----plot-diablo, message=FALSE, fig.cap='(ref:plot-diablo)'----
plotDiablo(diablo.tcga, ncomp = 2)
```

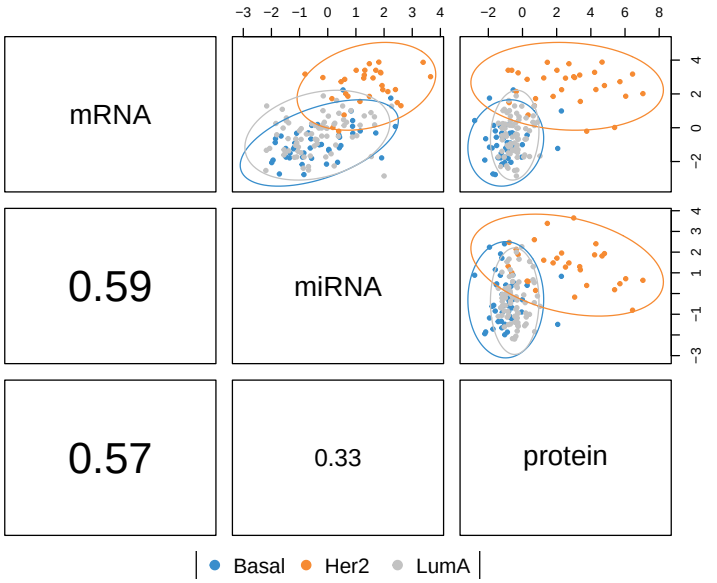

FIGURE: Multi block PLS-DA, diagnostic plot: component 2, 95% confidence intervals are plotted. Numbers indicate correlation coefficients between the first components from each data set.

```
## ----diablo-plotindiv, message=FALSE, fig.cap='(ref:diablo-plotindiv)'----
plotIndiv(diablo.tcga, ind.names = FALSE, legend = TRUE,
          title = 'TCGA, DIABLO comp 1 - 2')
```

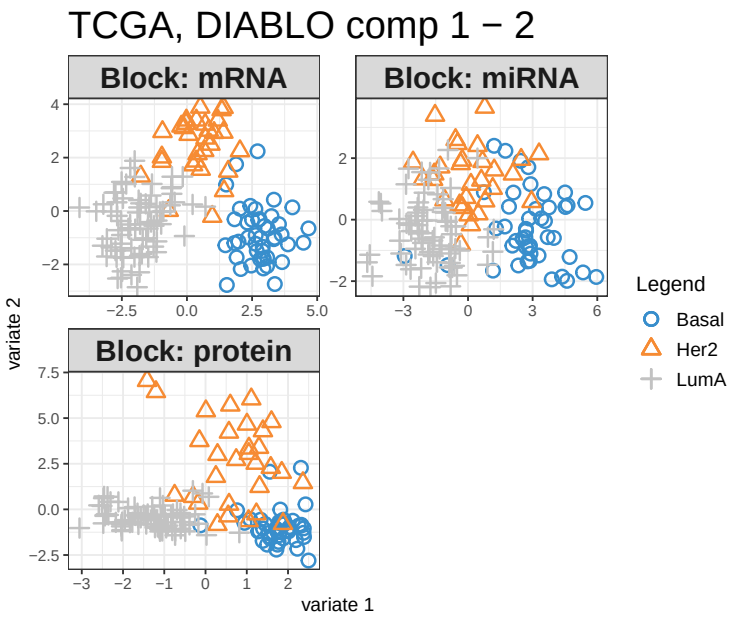

FIGURE: Multi block PLS-DA, individual omes: Check that extracted data sets can discriminate between samples: True.

```
plotIndiv(diablo.tcga, ind.names = FALSE, legend = TRUE, ellipse = TRUE, title = 'TCGA, DIABLO comp 1 -
```

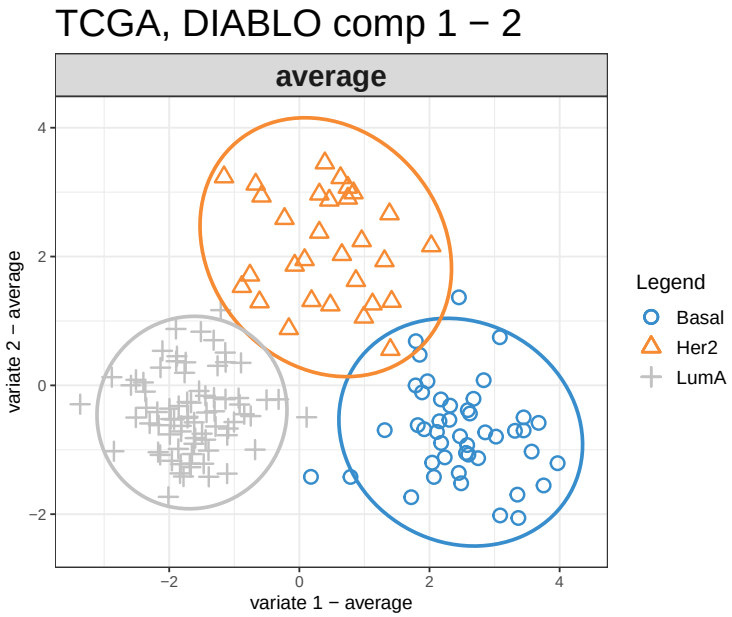

FIGURE: Multi block PLS-DA: Main plot that includes all three data sets - with symbols.

```
plotIndiv(diablo.tcga, ind.names = TRUE, legend = TRUE,
          title = 'TCGA, DIABLO comp 1 - 2', block = 'average')
```

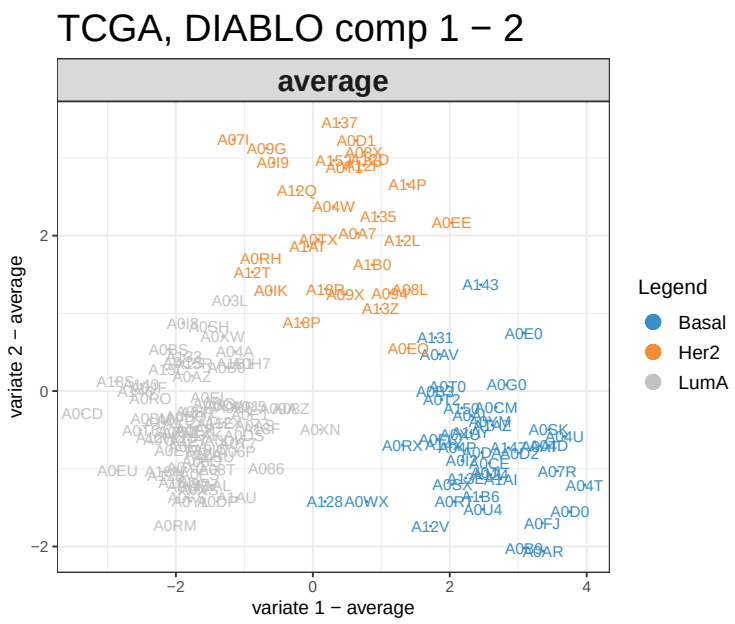

FIGURE: Multi block PLS-DA: Main plot that includes all three data sets - with individual sample names.

5.3.2 plotArrow

```
## ----diablo-plotarrow, message=FALSE, fig.cap='(ref:diablo-plotarrow)'----
plotArrow(diablo.tcga, ind.names = FALSE, legend = TRUE,
          title = 'TCGA, DIABLO comp 1 - 2')
```

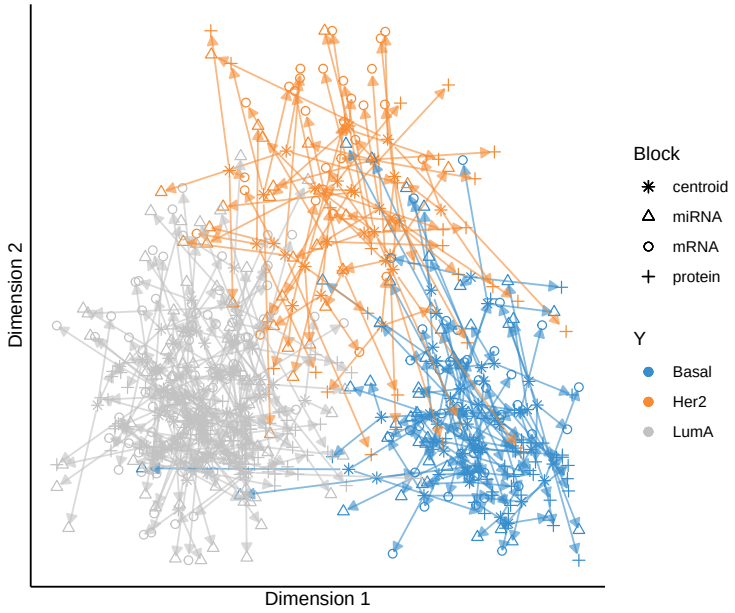

FIGURE: arrow plot for Multi block PLS-DA: Shows agreement between the three data sets. Some samples have a bit of variability between data sets.

5.3.3 plotVar

```
## ----diablo-plotvar, message=FALSE, fig.cap='(ref:diablo-plotvar)'----
plotVar(diablo.tcga, var.names = FALSE, style = 'graphics', legend = TRUE,
        pch = c(16, 17, 15), cex = c(2,2,2),
        col = c('darkorchid', 'brown1', 'lightgreen'),
        title = 'TCGA, DIABLO comp 1 - 2')
```

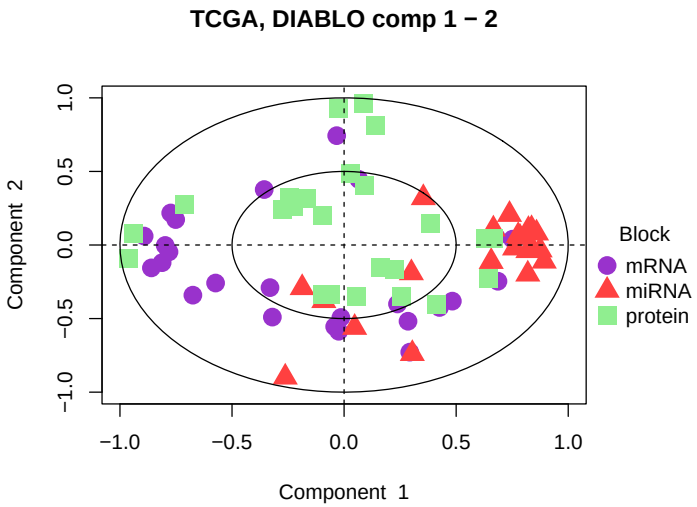

FIGURE: circle plot for Multi block PLS-DA: Shows correlation between variables.

```
## ----diablo-plotvar, message=FALSE, fig.cap='(ref:diablo-plotvar)'-----------
plotVar(diablo.tcga, var.names = TRUE, style = 'graphics', legend = TRUE,         pch = c(16, 17, 15), c
        col = c('darkorchid', 'brown1', 'lightgreen'),
        title = 'TCGA, DIABLO comp 1 - 2')
```

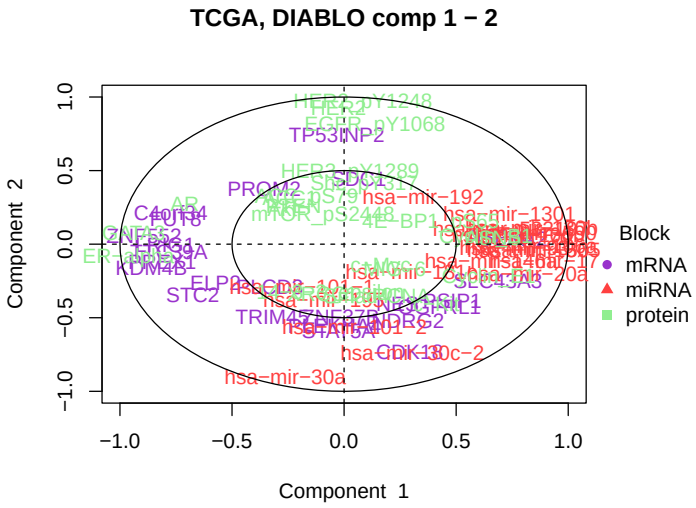

FIGURE: circle plot for Multi block PLS-DA: Shows correlation between variables - and names, which are overlapping unfortunately.

5.3.4 circosPlot

```
## ----diablo-circos, message=FALSE, fig.cap='(ref:diablo-circos)'----
circosPlot(diablo.tcga, cutoff = 0.7, line = TRUE,
           color.cor = c("chocolate3","grey20"), size.labels = 1.5)
```

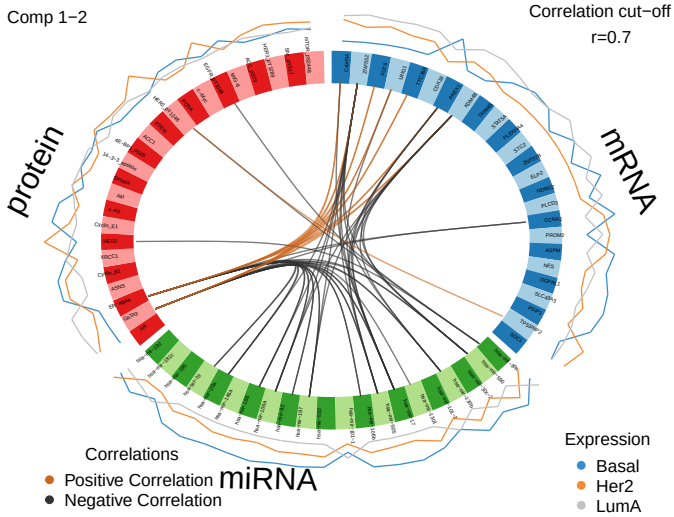

FIGURE: circos plot for Multi block PLS-DA: Plot represents the correlations between variables of different types, represented on the side quadrants..

5.3.5 cimDiablo

```
## ----diablo-cim, message=FALSE, fig.width=10, fig.height=8, fig.cap='(ref:diablo-cim)'----
# FIGURE CAN BE OUT OF BONDS, MARGINS NEED TO BE ADJUSTED..
cimDiablo(diablo.tcga, color.blocks = c('darkorchid', 'brown1', 'lightgreen'),
          comp = 1, margin=c(8,20), legend.position = "right")
```

```
# USE THIS CODE INSTEAD TO EXPORT HEATMAP
time<-format(Sys.time(),"%d-%m-%Y")
Title <- paste("Clustered Image Map for Multi block PLS-DA")
FileName <- paste(Title,time,".jpg")
jpeg(file=FileName, width = 2280, height = 1280, res = 210)
#
cimDiablo(diablo.tcga, color.blocks = c('darkorchid', 'brown1', 'lightgreen'), comp = 1, margin=c(8,20)
```

```
##
## trimming values to [-3, 3] range for cim visualisation. See 'trim' arg in ?cimDiablo
```

```
#
dev.off()
```

```
## pdf
##   2
```

FIGURE: Clustered Image Map (CIM) for Multi block PLS-DA: Shows correlation between variables - and names.

5.3.6 plotLoadings

```
## ----diablo-loading, message=FALSE, fig.cap='(ref:diablo-loading)'----
plotLoadings(diablo.tcga, comp = 1, contrib = 'max', method = 'median')
```

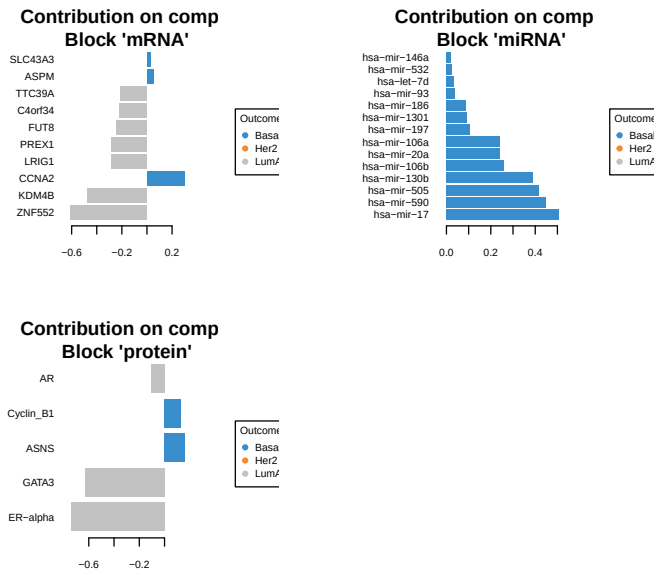

FIGURE: SYMBIONT contributing variables to sparse PLS-DA: comp 1.

```
## ----diablo-loading, message=FALSE, fig.cap='(ref:diablo-loading)'----
plotLoadings(diablo.tcga, comp = 2, contrib = 'max', method = 'median')
```

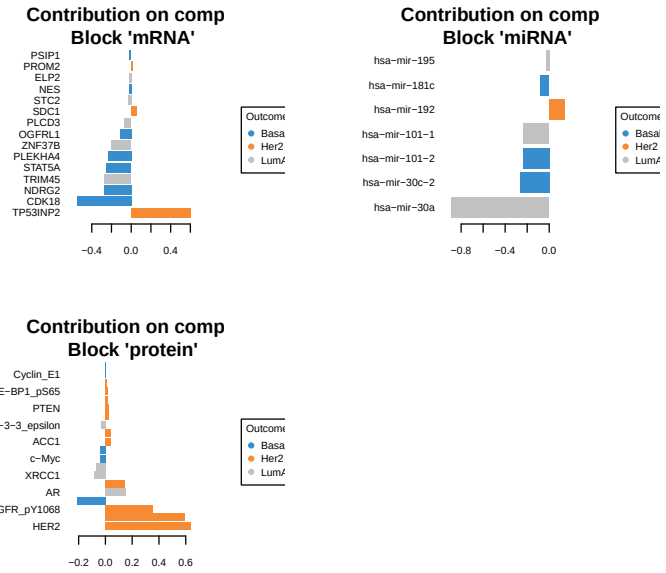

FIGURE: SYMBIONT contributing variables to sparse PLS-DA: comp 2.

```
## variables selected for block LIPIDS on component 1 & 2
selectVar(diablo.tcga, block = 'miRNA', comp = 1)$miRNA$value
```

```
##                value.var
## hsa-mir-17    0.50784008
## hsa-mir-590   0.44637708
## hsa-mir-505   0.41777112
## hsa-mir-130b  0.38936001
## hsa-mir-106b  0.26105215
## hsa-mir-20a   0.24302558
## hsa-mir-106a  0.24255557
## hsa-mir-197   0.10400667
## hsa-mir-1301  0.09207592
## hsa-mir-186   0.08917724
## hsa-mir-93    0.03781788
## hsa-let-7d    0.03212231
## hsa-mir-532   0.02281795
## hsa-mir-146a  0.02094122
```

```
selectVar(diablo.tcga, block = 'miRNA', comp = 2)$miRNA$value
```

```
##                 value.var
## hsa-mir-30a    -0.88888510
## hsa-mir-30c-2  -0.26428373
```

```
## hsa-mir-101-2 -0.23839272
## hsa-mir-101-1 -0.23505237
## hsa-mir-192    0.14465567
## hsa-mir-181c  -0.07819919
## hsa-mir-195   -0.03027055
```

```
## variables selected for block METABOLITES on component 1 & 2
selectVar(diablo.tcga, block = 'mRNA', comp = 1)$metabolites$value
```

```
## NULL
```

```
selectVar(diablo.tcga, block = 'mRNA', comp = 2)$metabolites$value
```

```
## NULL
```

```
## variables selected for block PROTEOME on component 1 & 2
selectVar(diablo.tcga, block = 'protein', comp = 1)$protein$value
```

```
##            value.var
## ER-alpha  -0.7416120
## GATA3     -0.6291728
## ASNS       0.1598018
## Cyclin_B1  0.1313683
## AR        -0.1065784
```

```
selectVar(diablo.tcga, block = 'protein', comp = 2)$protein$value
```

```
##                   value.var
## HER2            0.637172330
## HER2_pY1248     0.595322093
## EGFR_pY1068     0.355593269
## c-Kit          -0.212945250
## AR              0.156324373
## HER3_pY1289     0.144008226
## XRCC1          -0.087589029
## Smad4          -0.070788022
## c-Myc          -0.040872523
## PCNA           -0.040097552
## ACC1            0.039151934
## Shc_pY317       0.038155173
## 14-3-3_epsilon -0.035791135
## mTOR_pS2448     0.029555915
## PTEN            0.022830962
## Akt             0.019951184
## 4E-BP1_pS65     0.019412723
## ACC_pS79        0.013973232
## Cyclin_E1      -0.004477490
## MIG-6          -0.004385699
```

```
## ------------------------------
# Performance with Majority vote
perf.diablo.tcga$MajorityVote.error.rate
```

```
## $max.dist
##                   comp1      comp2      comp3      comp4       comp5
## Basal       0.02666667 0.03555556 0.02222222 0.06222222 0.062222222
## Her2        1.00000000 0.27333333 0.18666667 0.17333333 0.146666667
## LumA        0.00000000 0.00000000 0.00000000 0.00000000 0.005333333
## Overall.ER  0.20800000 0.06533333 0.04400000 0.05333333 0.050666667
## Overall.BER 0.34222222 0.10296296 0.06962963 0.07851852 0.071407407
##
## $centroids.dist
##                   comp1      comp2      comp3      comp4      comp5
## Basal       0.07111111 0.04000000 0.03555556 0.05333333 0.04888889
## Her2        0.24666667 0.11333333 0.10000000 0.12000000 0.13333333
## LumA        0.06933333 0.04533333 0.04000000 0.03466667 0.03466667
## Overall.ER  0.10533333 0.05733333 0.05066667 0.05733333 0.05866667
## Overall.BER 0.12903704 0.06622222 0.05851852 0.06933333 0.07229630
##
## $mahalanobis.dist
##                   comp1      comp2      comp3      comp4      comp5
## Basal       0.07111111 0.04000000 0.03111111 0.05333333 0.06222222
## Her2        0.24666667 0.09333333 0.08666667 0.05333333 0.08666667
## LumA        0.06933333 0.05333333 0.04266667 0.01866667 0.01333333
## Overall.ER  0.10533333 0.05733333 0.04800000 0.03600000 0.04266667
## Overall.BER 0.12903704 0.06222222 0.05348148 0.04177778 0.05407407
```

```
## ------------------------------
# Performance with Weighted vote
perf.diablo.tcga$WeightedVote.error.rate
```

```
## $max.dist
##                   comp1      comp2      comp3      comp4       comp5
## Basal       0.02666667 0.02666667 0.02222222 0.04888889 0.048888889
## Her2        1.00000000 0.24000000 0.15333333 0.14666667 0.133333333
## LumA        0.00000000 0.00000000 0.00000000 0.00000000 0.005333333
## Overall.ER  0.20800000 0.05600000 0.03733333 0.04400000 0.044000000
## Overall.BER 0.34222222 0.08888889 0.05851852 0.06518519 0.062518519
##
## $centroids.dist
##                   comp1      comp2      comp3      comp4      comp5
## Basal       0.06666667 0.03111111 0.02666667 0.04000000 0.03555556
## Her2        0.22666667 0.07333333 0.06666667 0.09333333 0.10666667
## LumA        0.06933333 0.04266667 0.03733333 0.03466667 0.03466667
## Overall.ER  0.10000000 0.04533333 0.04000000 0.04800000 0.04933333
## Overall.BER 0.12088889 0.04903704 0.04355556 0.05600000 0.05896296
##
## $mahalanobis.dist
##                   comp1      comp2      comp3      comp4      comp5
## Basal       0.06666667 0.01777778 0.02666667 0.04000000 0.04444444
## Her2        0.22666667 0.07333333 0.06666667 0.04666667 0.08666667
## LumA        0.06933333 0.04800000 0.03200000 0.01866667 0.01333333
```

```
## Overall.ER  0.10000000 0.04400000 0.03733333 0.03066667 0.03733333
## Overall.BER 0.12088889 0.04637037 0.04177778 0.03511111 0.04814815
```

5.4 Model performance and prediction

```
# NOT COVERED HERE
```

6 WORKING WITH OWN DATA

```
#getwd()
#setwd("PATH/TO/FOLDER")

# REQUIRED FORMAT:
# -> Your samples in rows, with unique IDs
# -> Your data in columns, with short and unique names

# Load from csv file
data_1 <- read.csv("PATH/TO/FOLDER/file1.csv", row.names = 1, header = TRUE, sep = '\t')

data_2 <- read.csv("PATH/TO/FOLDER/file1.csv", row.names = 1, header = TRUE, sep = '\t')

data_3 <- read.csv("PATH/TO/FOLDER/file1.csv", row.names = 1, header = TRUE, sep = '\t')

# Reformat data to matrix
data_1 <- as.matrix(data_1)

data_2 <- as.matrix(data_2)

data_3 <- as.matrix(data_3)

# Generate list of the data sets
X <- list(data_1 = data_1,
          data_2 = data_2,
          data_3 = data_3)
summary(X)

# Load Outcome file
data_meta <- read.csv("PATH/TO/FOLDER/MetaData.csv", header = TRUE, sep = '\t')

# Reformat Outcome file
Y_factor <- data_meta$YourFactor %>% as.factor()
summary(Y)

Y_scale <- data_meta$YourScale %>% as.numeric()
summary(Y_scale)
```

7 SESSION INFO

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.utf8         LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Australia/Sydney
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.2 forcats_1.0.0   stringr_1.5.0   dplyr_1.1.2
##  [5] purrr_1.0.1     readr_2.1.4     tidyr_1.3.0     tibble_3.2.1
##  [9] tidyverse_2.0.0 mixOmics_6.24.0 ggplot2_3.4.2   lattice_0.21-8
## [13] MASS_7.3-60     knitr_1.43
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.3          generics_0.1.3      stringi_1.7.12
##  [4] hms_1.1.3           digest_0.6.33       magrittr_2.0.3
##  [7] timechange_0.2.0    evaluate_0.21       grid_4.3.1
## [10] RColorBrewer_1.1-3  fastmap_1.1.1       plyr_1.8.8
## [13] Matrix_1.6-0        ggrepel_0.9.3       RSpectra_0.16-1
## [16] gridExtra_2.3       fansi_1.0.4         scales_1.2.1
## [19] codetools_0.2-19    cli_3.6.1           rlang_1.1.1
## [22] munsell_0.5.0       withr_2.5.0         yaml_2.3.7
## [25] ellipse_0.5.0       tools_4.3.1         parallel_4.3.1
## [28] tzdb_0.4.0          reshape2_1.4.4      BiocParallel_1.34.2
## [31] colorspace_2.1-0    corpcor_1.6.10      vctrs_0.6.3
## [34] R6_2.5.1            matrixStats_1.0.0   lifecycle_1.0.3
## [37] pkgconfig_2.0.3     pillar_1.9.0        gtable_0.3.3
## [40] glue_1.6.2          rARPACK_0.11-0      Rcpp_1.0.11
## [43] xfun_0.39           tidyselect_1.2.0    rstudioapi_0.15.0
## [46] farver_2.1.1        snow_0.4-4          htmltools_0.5.5
## [49] igraph_1.5.0.1      labeling_0.4.2      rmarkdown_2.23
## [52] compiler_4.3.1
```