



THE UNIVERSITY
OF BRITISH COLUMBIA

Department of
Anesthesiology, Pharmacology
& Therapeutics
Faculty of Medicine



Centre for
Heart Lung Innovation
UBC and St. Paul's Hospital



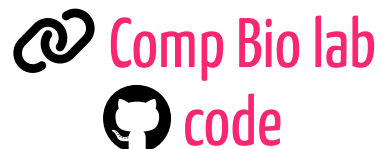
CompBio Lab @ HLI

cbl-hli.med.ubc.ca/

Getting started with mixOmics

Amrit Singh, PhD
Assistant Professor

April 13th, 2023 | 11:45-12:35 EST



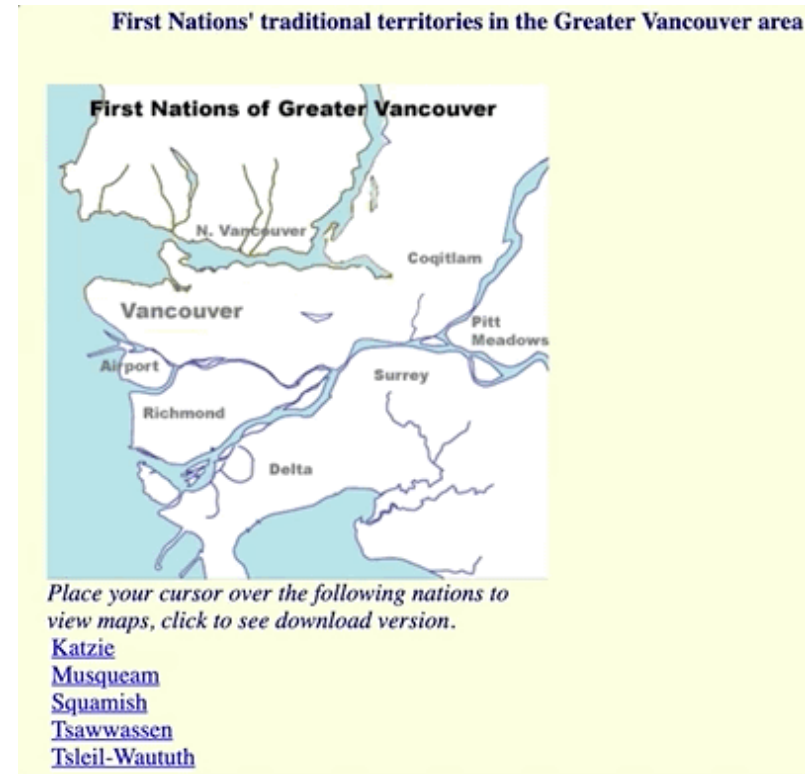
Land acknowledgement

I would like to acknowledge that I work on the traditional, ancestral, and unceded territory of the Coast Salish Peoples, including the territories of the xwməθkwə́yəm (Musqueam), Skwxwú7mesh (Squamish), Stó:lō and Səlílwətaʔ/Selilwitulh (Tsleil- Waututh) Nations.

Traditional: Traditionally used and/or occupied by Musqueam people

Ancestral: Recognizes land that is handed down from generation to generation

Unceded: Refers to land that was not turned over to the Crown (government) by a treaty or other agreement



Learning outcomes

By the end of this lecture you will be able to:

1. Describe what the *mixOmics* R-library can do.
2. Describe when to use which method and for what purpose (exploration, classification, integration).
3. Analyze data using mixOmics for various purposes (exploration, classification, integration)

High-dimensional data

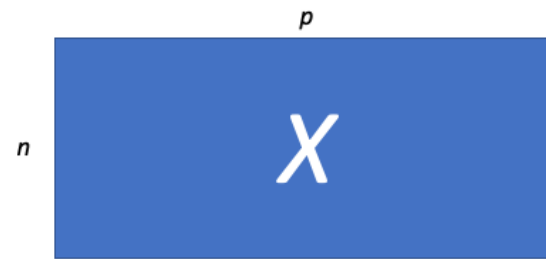
- $n \ll p$ (number of observations is much smaller than the number variables)
- data is highly correlated

univariate

```
##      p_1
## 1  0.249
## 2  0.247
## 3  0.281
## 4  0.846
## 5 -0.791
## 6 -0.337
## 7  0.179
## 8  0.022
## 9 -0.366
## 10 -0.487
```

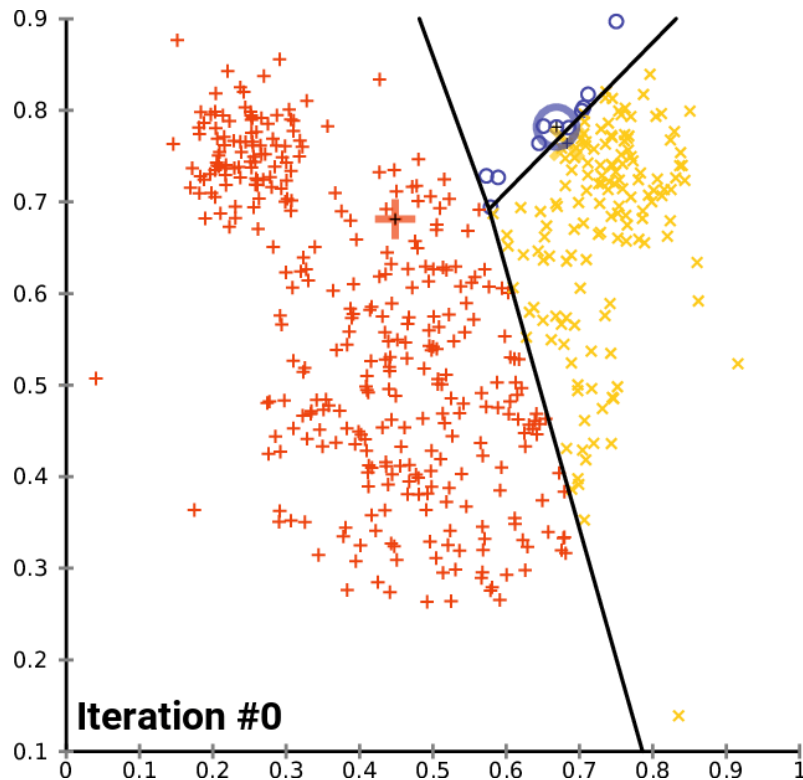
multivariate

```
##      p_1      p_2      p_3      p_4      p_5      p_6      p_7      p_8      p_9      p_10
## 1 -0.4720 -1.750  0.18000 -0.0114 -1.370  0.332 -0.433  0.157 -1.6800  0.6270
## 2 -2.1200  1.540 -1.58000 -0.1750 -1.140 -0.162 -0.215  1.460 -0.8570 -0.2320
## 3 -0.6970  0.655  1.76000  1.8900  0.659  0.717 -0.390  1.020 -0.4370 -0.0564
## 4  0.3740  0.984  0.00456 -2.0400 -1.080 -0.277  0.516 -0.768  0.2020 -0.2160
## 5  1.6400  0.605 -0.90400  0.2330 -0.931 -1.010  0.862  1.190  0.0228  0.6350
## 6 -0.7670  1.110 -1.68000 -1.0300  2.120  0.461 -0.995  0.359 -1.5200  0.7930
## 7 -1.6200  0.162  0.60500  0.9520  0.976 -0.706  1.710  0.182  0.7340 -1.0100
## 8 -0.0352 -0.895 -0.20600  0.0174 -2.090 -1.420  1.200  0.514 -0.5370  0.2280
## 9  0.5360  0.389  0.20900 -1.8200 -1.570  0.792 -1.170  0.891  0.8450  0.2370
## 10 -0.5940 -0.799  1.82000  0.2710 -0.921 -1.130 -0.870  0.414 -0.1110  1.1500
```



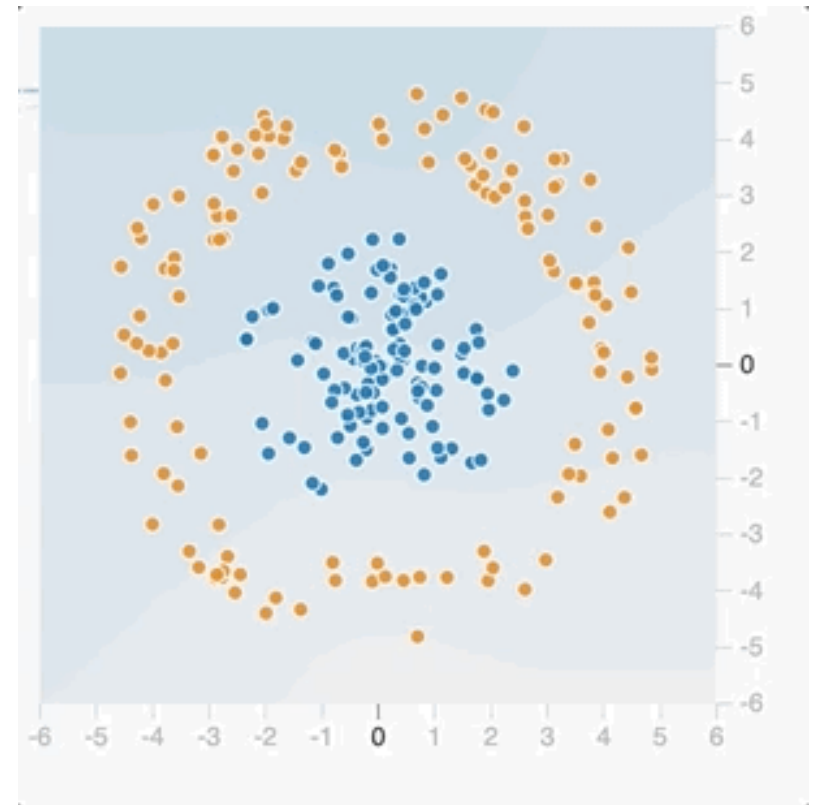
What can you do with high-dimensional data?

Unsupervised (clustering)



Chire 2017

Supervised (regression/classification)



Tensorflow playground

mixOmics

- initiative started and maintained by Prof Kim-Anh Lê Cao
- R-library with 19 methods for high-dimensional data (exploratory analyses, classification, regression, data integration, meta-analysis)

Lab head: [A/Prof Kim-Anh Lê Cao](#)

NHMRC Career Development Fellow

Melbourne Integrative Genomics (MIG) & School
of Mathematics and Statistics

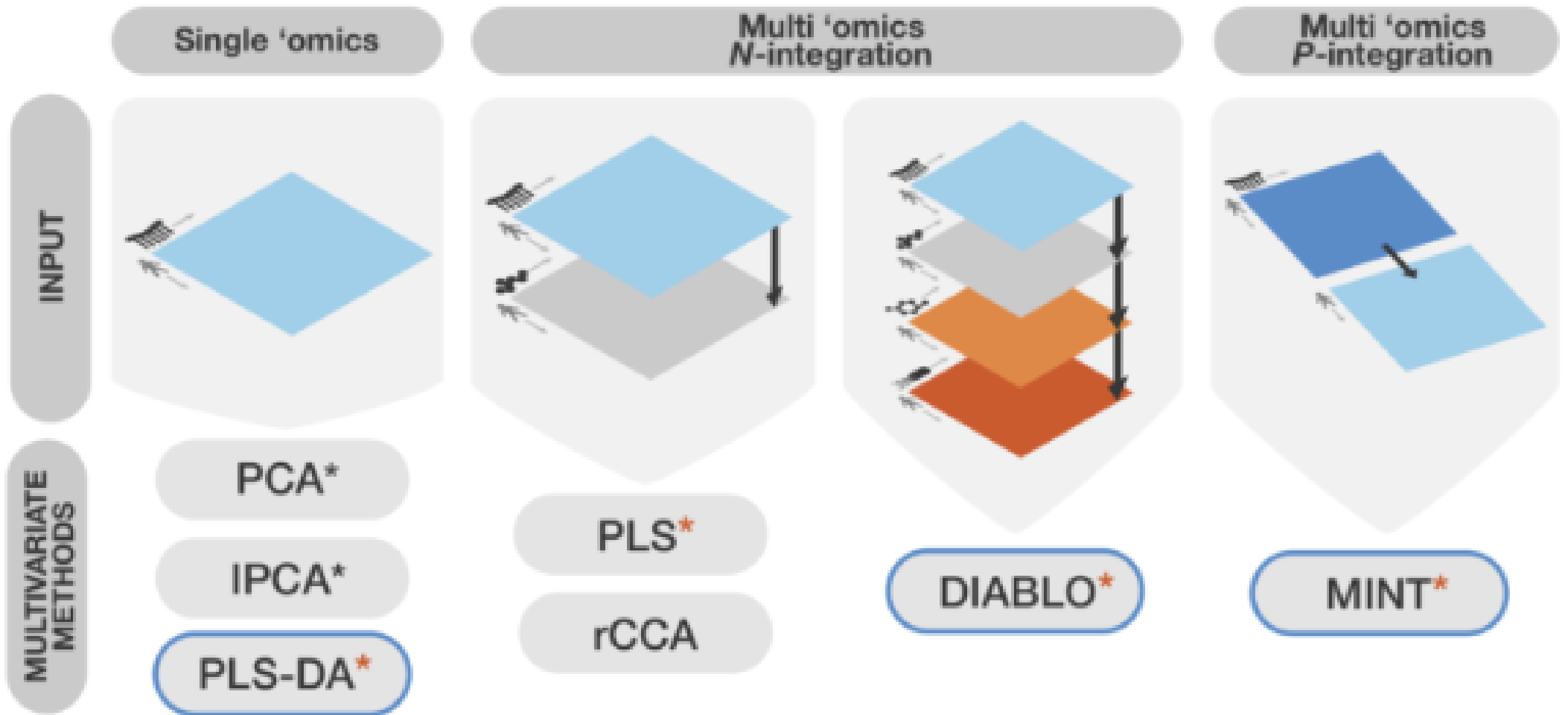
Building 184 ground floor | University of
Melbourne | Parkville VIC 3010

@: [kimanh.lecao\[at\]unimelb.edu.au](mailto:kimanh.lecao@unimelb.edu.au) | [twitter](#):

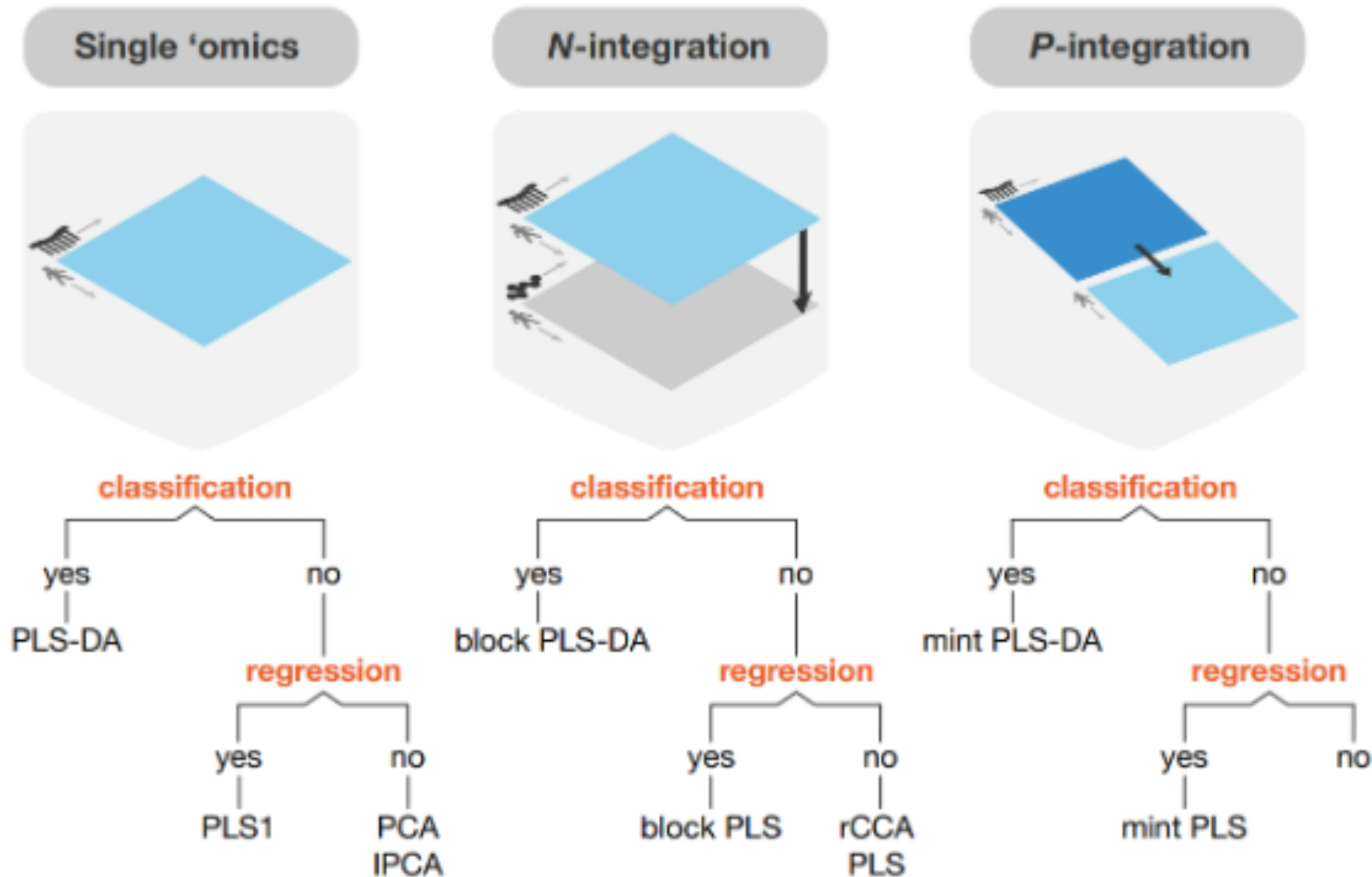
[@mixOmics_team](#) | [Ph](#): +61 3 8344 3971



What does mixOmics offer? methods...

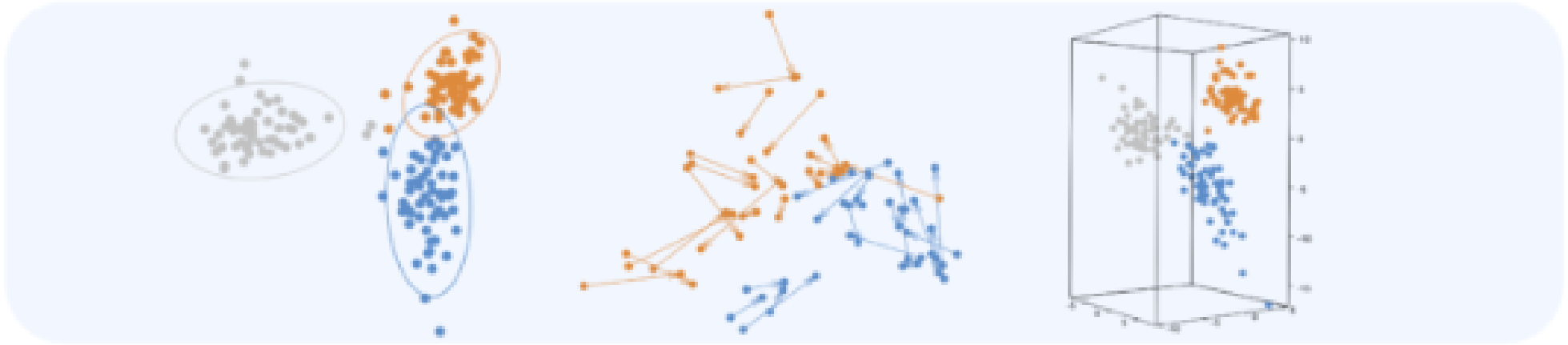


What does mixOmics offer? when to use these methods...

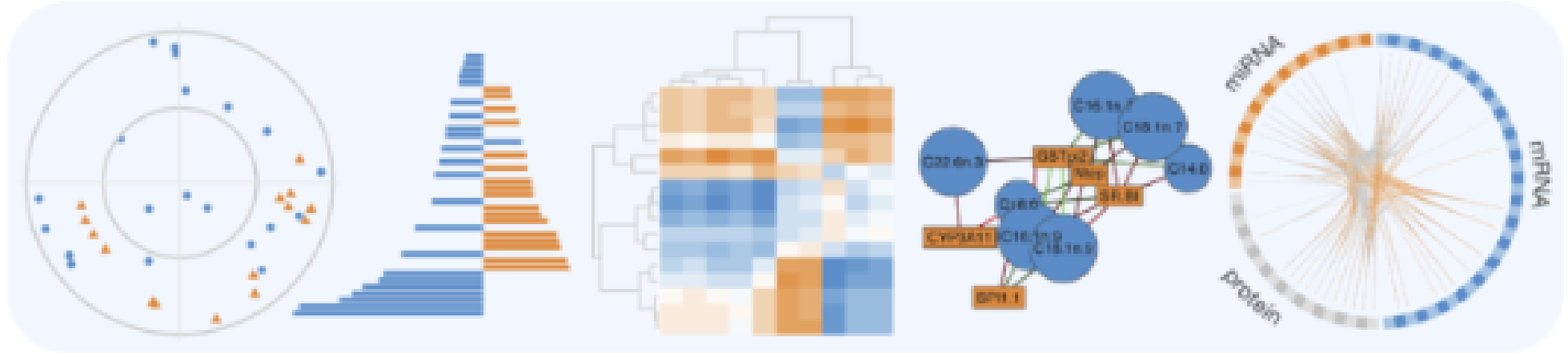


What does mixOmics offer? graphics...

SAMPLE PLOTS



VARIABLE PLOTS



Getting started with mixOmics

1. Download R
2. Download RStudio
3. install mixOmics

install mixOmics

```
if (!require("BiocManager", quietly = TRUE))  
  install.packages("BiocManager")  
  
BiocManager::install("mixOmics")
```

load vignette

```
openVignette("mixOmics")
```

Dataset used in this talk

Breast Cancer multi omics data from TCGA

This data set is a small subset of the full data set from The Cancer Genome Atlas that can be analysed with the DIABLO framework. It contains the expression or abundance of three matching omics data sets: mRNA, miRNA and proteomics for 150 breast cancer samples (Basal, Her2, Luminal A) in the training set, and 70 samples in the test set. The test set is missing the proteomics data set.

```
library(mixOmics)
data(breast.TCGA)
lapply(breast.TCGA$data.train, dim)
```

```
## $mirna
## [1] 150 184
##
## $mrna
## [1] 150 200
##
## $protein
## [1] 150 142
##
## $subtype
## NULL
```

breast cancer subtypes

```
addmargins(table(breast.TCGA$data.train$subtype))
```

```
##
## Basal  Her2  LumA  Sum
##    45    30    75  150
```

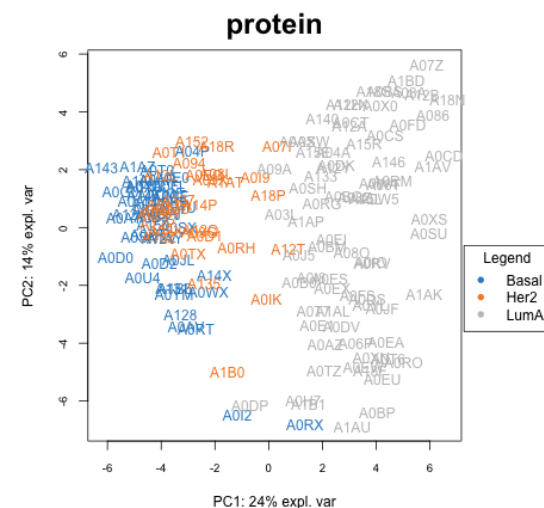
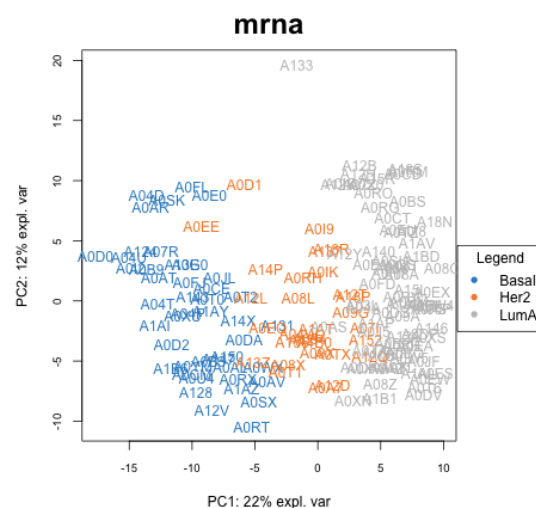
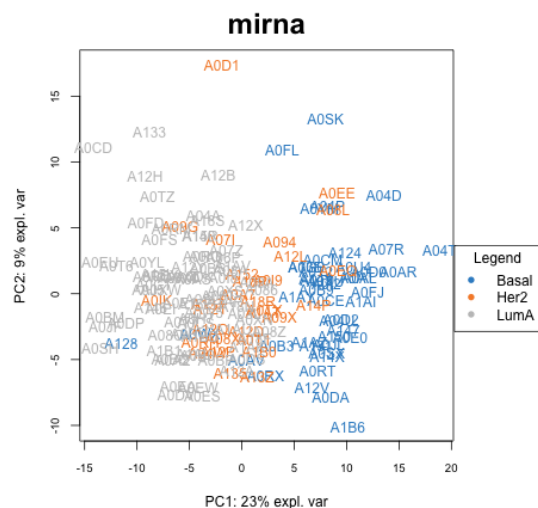
Analysis plan

Analysis	Methods	Functions
Exploratory data analysis	PCA	pca() plotIndiv()
Discriminant analysis	sPLSDA	splsda() tune(), perf() plotIndiv(), plotVar()
Data integration analysis	DIABLO	block.splsda() tune(), perf() plotDiablo(), circosPlot()

Exploratory data analysis using PCA

```
J <- length(breast.TCGA$data.train)-1
pcs <- lapply(breast.TCGA$data.train[1:J], pca)

mapply(function(pca, dataset){
  plotIndiv(pca,
    title=dataset,
    group=breast.TCGA$data.train$subtype,
    style="graphics",
    legend=TRUE)
}, pca=pcs, dataset=names(breast.TCGA$data.train)[1:J])
```



Discriminant analysis using sPLSDA

- based on the eda it seems **mrna** is better at separating classes than **mirna**, lets test this.
- this may or may not be true since we peeked at the data (need to test model with another dataset)

- mrna

```
mrna_model <- splsda(X = breast.TCGA$data.train$mrna,  
                    Y = breast.TCGA$data.train$subty,  
                    keepX = c(5, 5),  
                    ncomp = 2)  
mrna_perf <- perf(mrna_model, validation = "Mfold", fo  
mrna_perf$error.rate
```

```
## $overall  
##           max.dist centroids.dist mahalanobis.dist  
## comp1 0.2146667      0.1653333      0.1653333  
## comp2 0.0920000      0.1026667      0.1226667  
##  
## $BER  
##           max.dist centroids.dist mahalanobis.dist  
## comp1 0.3454815      0.1851852      0.1851852  
## comp2 0.1177778      0.1137778      0.1361481
```

- mirna

```
mirna_model <- splsda(X = breast.TCGA$data.train$mirna  
                    Y = breast.TCGA$data.train$subty,  
                    keepX = c(5, 5),  
                    ncomp = 2)  
mirna_perf <- perf(mirna_model, validation = "Mfold",  
mirna_perf$error.rate
```

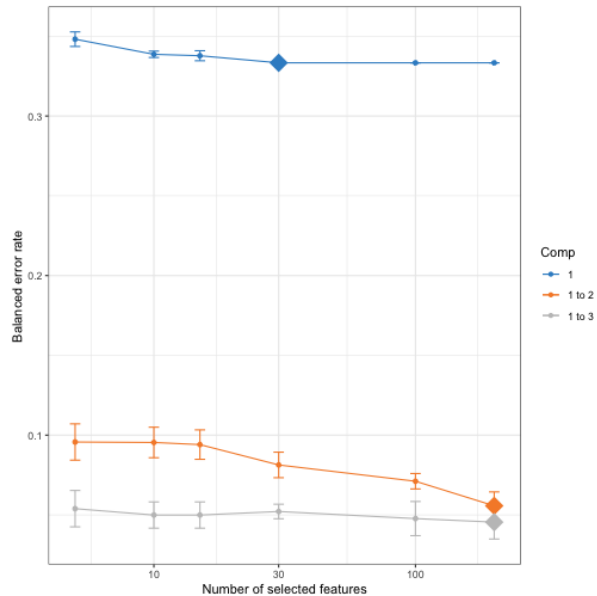
```
## $overall  
##           max.dist centroids.dist mahalanobis.dist  
## comp1 0.2613333      0.2946667      0.2946667  
## comp2 0.2226667      0.2426667      0.2133333  
##  
## $BER  
##           max.dist centroids.dist mahalanobis.dist  
## comp1 0.3884444      0.3197037      0.3197037  
## comp2 0.3042963      0.2680000      0.2551111
```

Find optimal models

- mrna

```
tune_mrna = tune(method = "splsda", X = breast.TCGA$da  
                  Y=breast.TCGA$data.train$subtype, nco  
                  test.keepX = c(5, 10, 15, 30, 100, 20  
                  progressBar = FALSE)
```

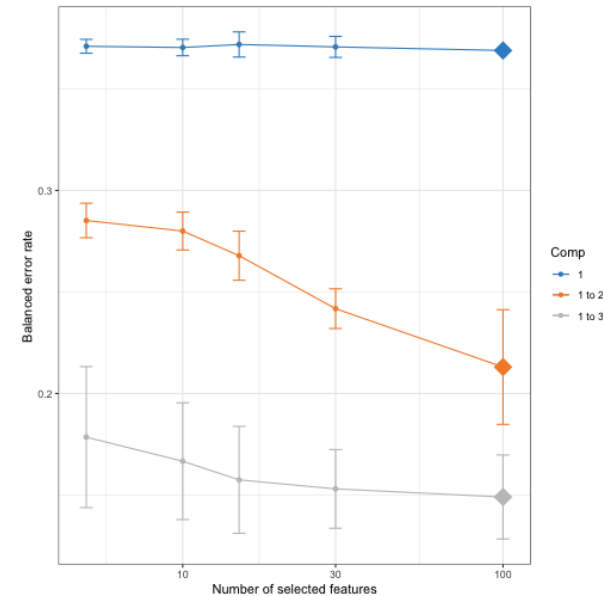
```
plot(tune_mrna)
```



- mirna

```
tune_mirna = tune(method = "splsda", X = breast.TCGA$d  
                  Y=breast.TCGA$data.train$subtype, nco  
                  test.keepX = c(5, 10, 15, 30, 100, 20  
                  progressBar = FALSE)
```

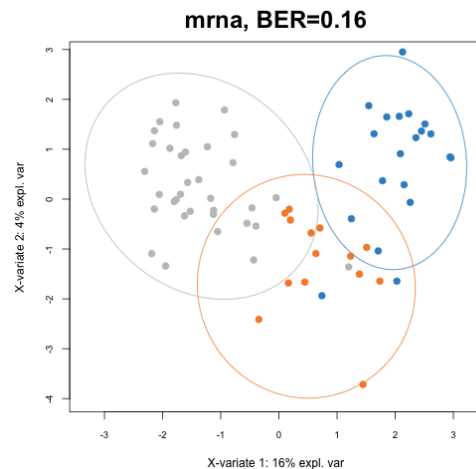
```
plot(tune_mirna)
```



Test sPLSDA models using data from other observations (patients)

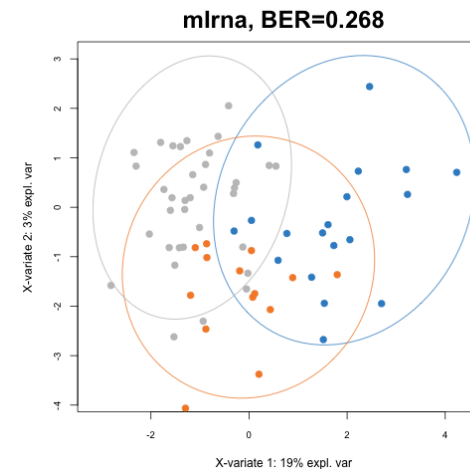
- mrna

```
mrna_model <- splsda(X = breast.TCGA$data.train$mrna,  
                    Y = breast.TCGA$data.train$subty  
                    keepX = rep(5, 3),  
                    ncomp = 3)  
mrna_pred <- predict(mrna_model, newdata = breast.TCGA  
plotIndiv(mrna_model, comp = 1:2, rep.space = "X-vari  
points(mrna_pred$variates[, 1], mrna_pred$variates[, 2
```



- mirna

```
mirna_model <- splsda(X = breast.TCGA$data.train$mirna  
                    Y = breast.TCGA$data.train$subty  
                    keepX = rep(5, 3),  
                    ncomp = 3)  
mirna_pred <- predict(mirna_model, newdata = breast.TC  
plotIndiv(mirna_model, comp = 1:2, rep.space = "X-vari  
points(mirna_pred$variates[, 1], mirna_pred$variates[,
```

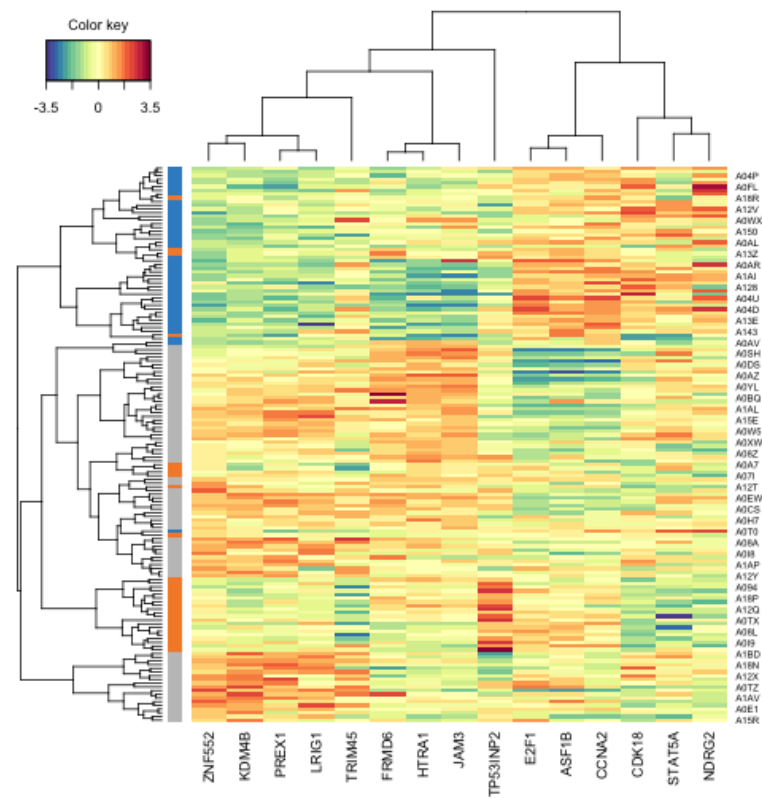


Variables in mrna model

```
rbind(selectVar(mrna_model, comp=1)$value,  
      selectVar(mrna_model, comp=2)$value,  
      selectVar(mrna_model, comp=3)$value)
```

##	value.var
## ZNF552	-0.75801237
## KDM4B	-0.58296361
## PREX1	-0.20979766
## LRIG1	-0.17185790
## CCNA2	0.10963799
## CDK18	0.69045321
## TP53INP2	-0.68042479
## NDRG2	0.22678162
## STAT5A	0.07254351
## TRIM45	0.06003335
## JAM3	0.72727214
## E2F1	-0.53439117
## FRMD6	0.33920140
## ASF1B	-0.23142418
## HTRA1	0.12994836

```
cim(mrna_model,  
     row.sideColors = mixOmics::color.mixo(as.numeric(b
```



Data integration using DIABLO

Design matters!

```
data = list(mrna = breast.TCGA$data.train$mrna, mirna
            protein = breast.TCGA$data.train$protein)
# set up a full design where every block is connected
# could also consider other weights, see our mixOmics
design = matrix(1, ncol = length(data), nrow = length(
            dimnames = list(names(data), names(dat
diag(design) = 0
design
```

```
##          mrna mirna protein
## mrna      0      1      1
## mirna     1      0      1
## protein   1      1      0
```

```
# set number of component per data set
ncomp = 3
test.keepX = list(mrna = c(10, 30), mirna = c(15, 25),

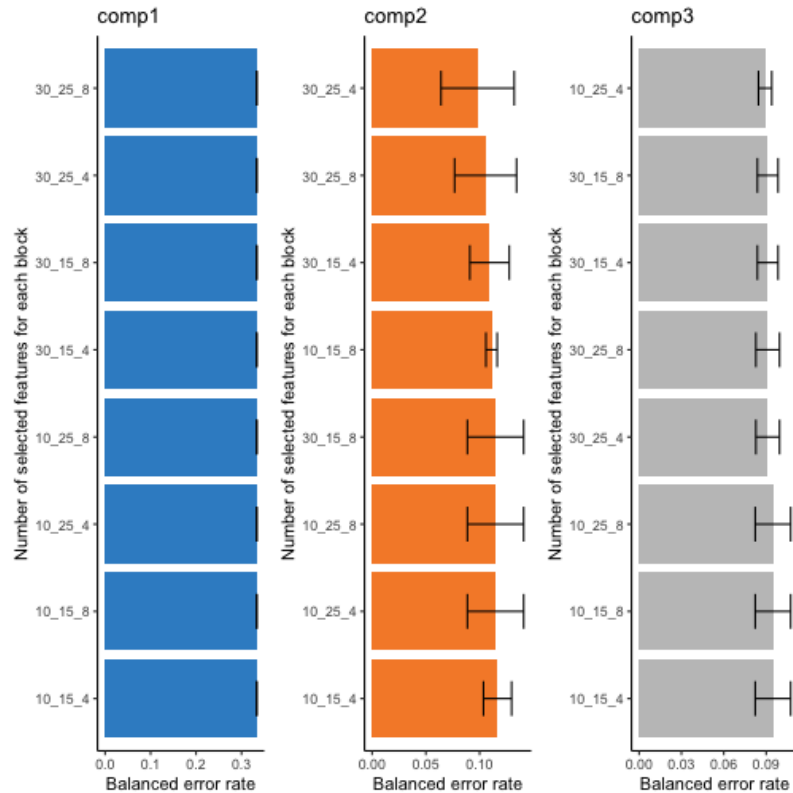
## setup cluster - use SnowParam() on Windows
BPPARAM <- BiocParallel::MulticoreParam(workers = para
tune <- tune.block.splsda(
  X = data,
  Y = breast.TCGA$data.train$subtype,
  ncomp = ncomp,
  test.keepX = test.keepX,
  design = design,
  nrepeat = 2,
  BPPARAM = BPPARAM
)
```

```
## Design matrix has changed to include Y; each block will be
## linked to Y.
```

```
##
## You have provided a sequence of keepX of length: 2 for block
## This results in 8 models being fitted for each component and
```

Finding the optimal DIABLO model

```
plot(tune)
```



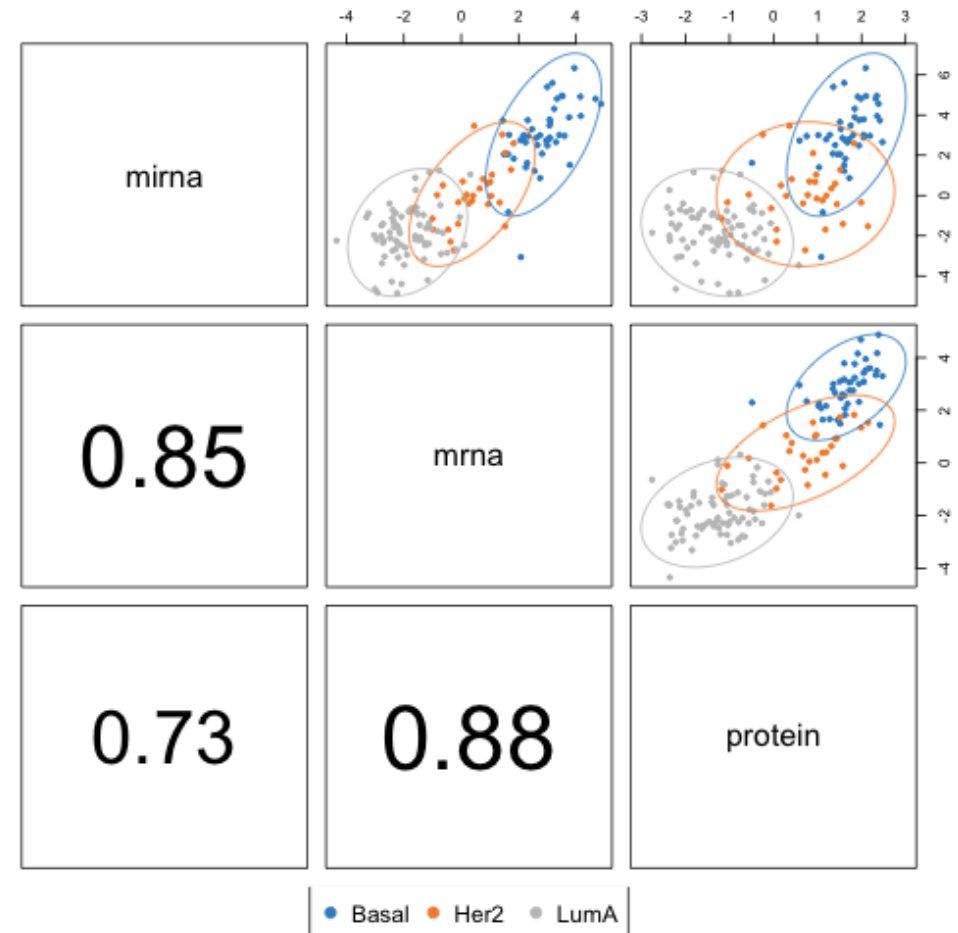
```
tune$choice.keepX
```

```
## $mrna  
## [1] 10 30 10  
##  
## $mirna  
## [1] 15 25 25  
##  
## $protein  
## [1] 4 4 4
```

DIABLO model

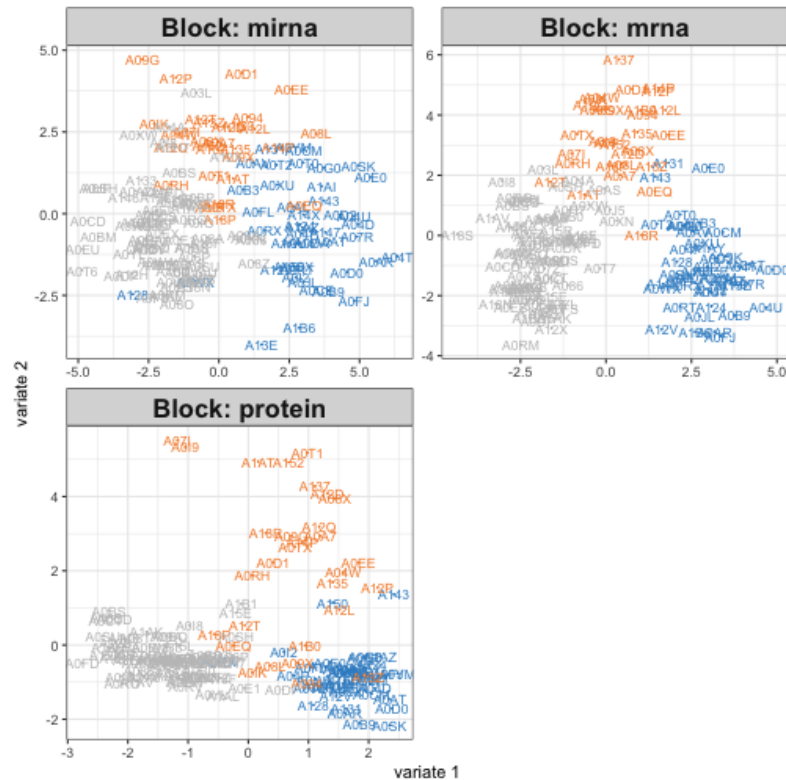
```
ncomp = length(tune$choice.keepX$mrna)

diablo <- block.splsda(X = breast.TCGA$data.train[1:3]
                      Y = breast.TCGA$data.train$subtype,
                      keepX=tune$choice.keepX,
                      ncomp = ncomp)
```

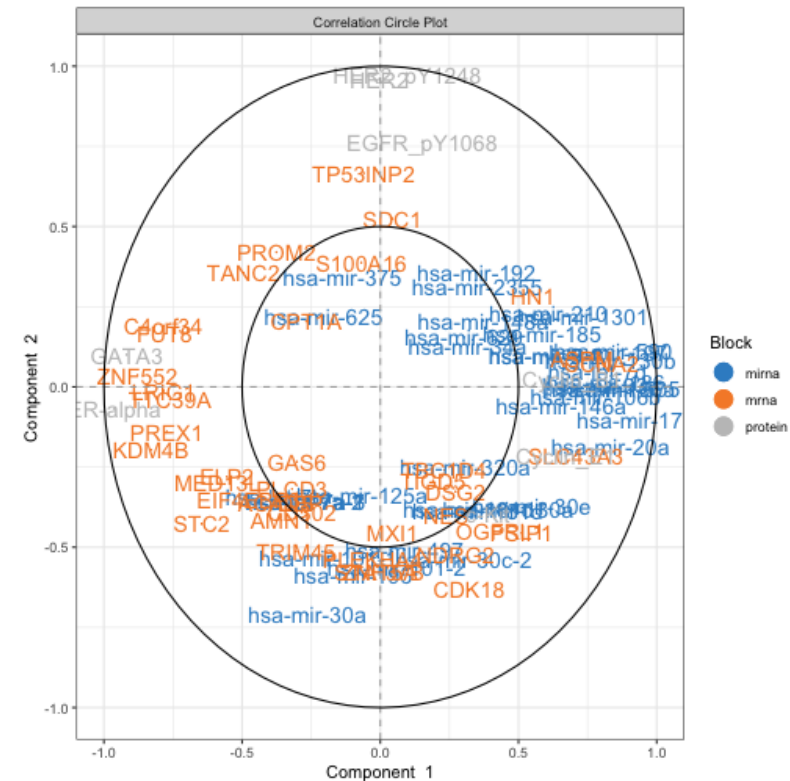


DIABLO: Sample and variable plots

```
plotIndiv(diablo)
```



```
plotVar(diablo, var.names = c(TRUE, TRUE, TRUE),  
        legend=TRUE, pch=c(16,16,1))
```

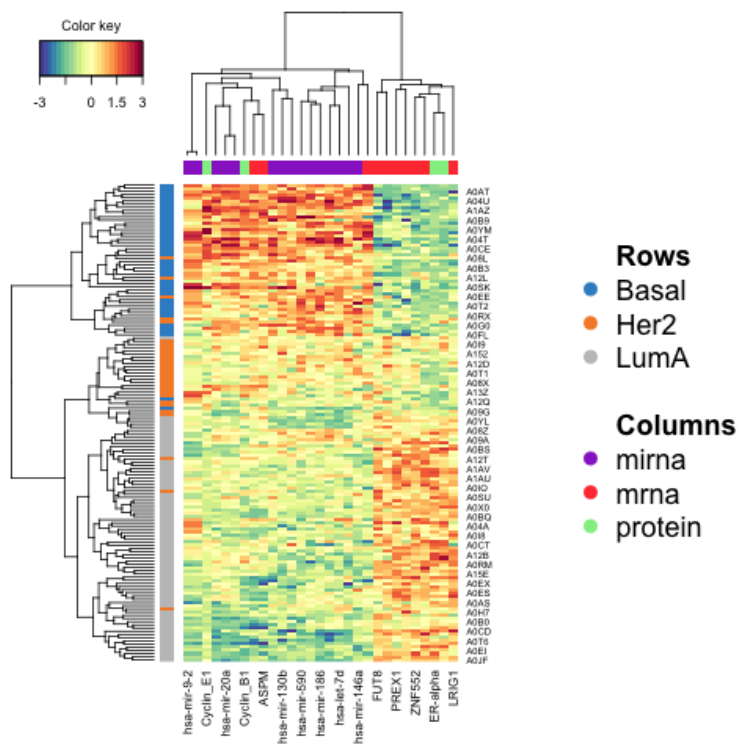


DIABLO

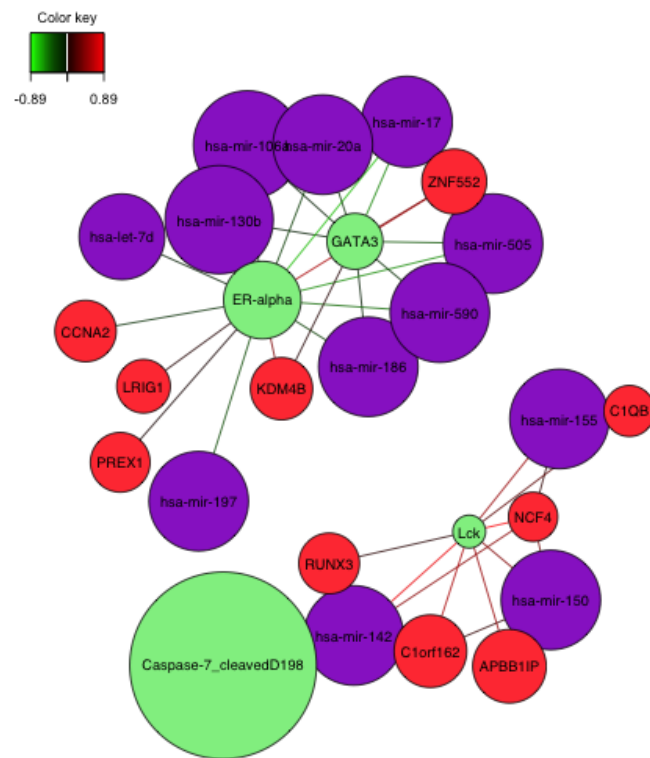
```
cimDiablo(diablo, color.blocks = c('darkorchid', 'brown', 'lightgreen'),
  comp = 1, margin=c(8,20), legend.position =
```

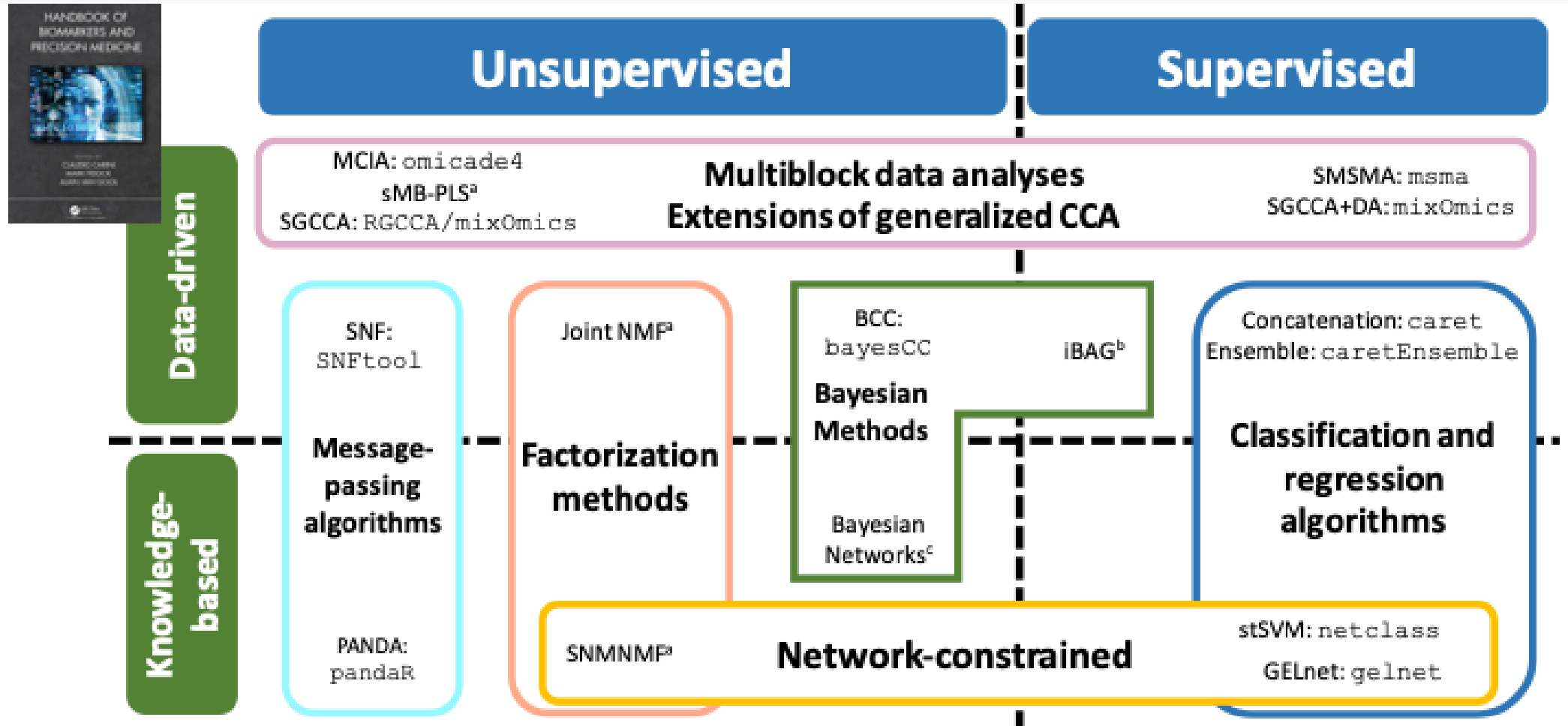
```
##
```

```
## trimming values to [-3, 3] range for cim visualisation. See
```



```
network(diablo, blocks = c(1,2,3),
  color.node = c('darkorchid', 'brown1', 'lightgreen'),
  cutoff = 0.75)
```







THE UNIVERSITY
OF BRITISH COLUMBIA

Department of
Anesthesiology, Pharmacology
& Therapeutics

Faculty of Medicine



Centre for
Heart Lung Innovation
UBC and St. Paul's Hospital




CompBio Lab @ HLI

cbl-hli.med.ubc.ca/

THANK YOU!

April 13th, 2023 | 11:45-12:35 EST

 **Comp Bio lab**
 **code**