iOmicsPASS+

Integrative -Omics Predictive Analysis of Subnetwork Signatures (Version II)

Hiromi WL Koh & Hyungwon Choi

19 November, 2021

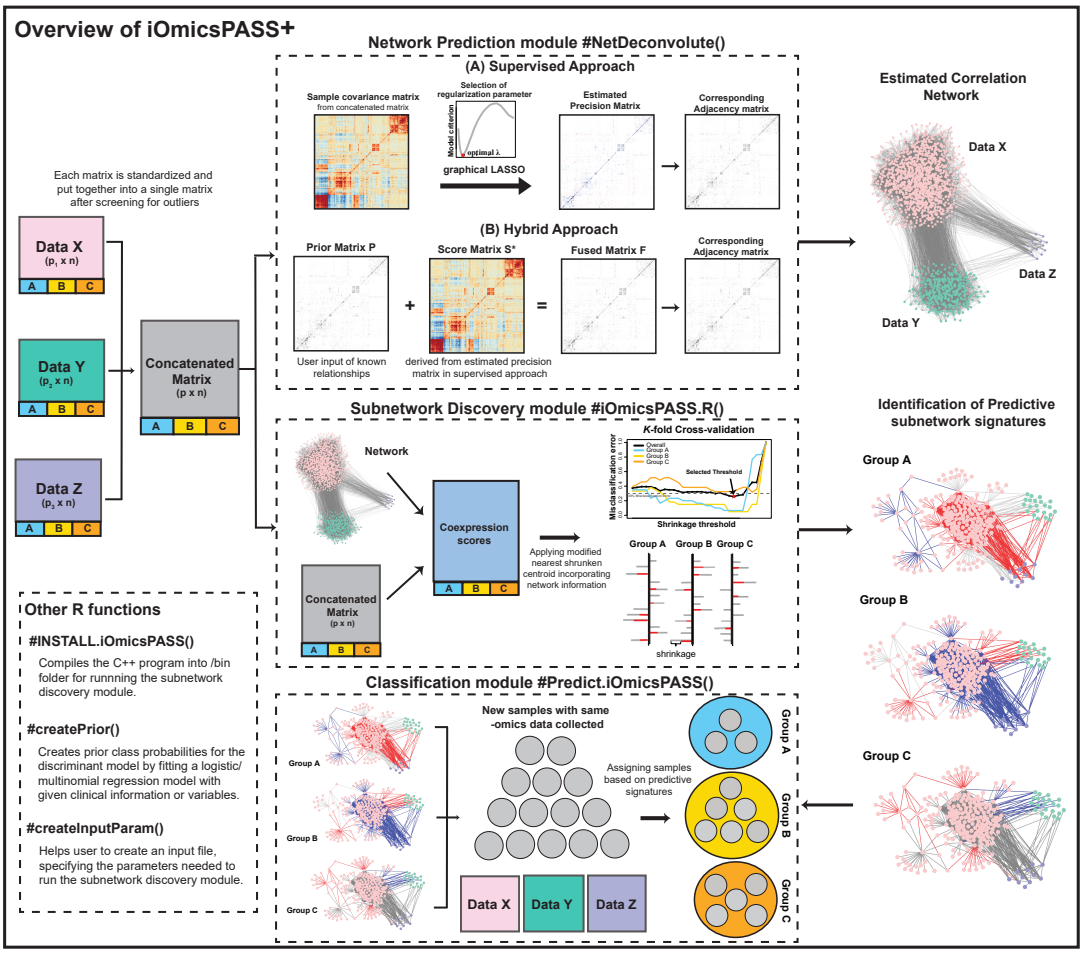
An Overview

iOmicsPASS+ is a R-package incorporating **iOmicsPASS** (Koh et al., 2019), extended to other types of -omics data allowing for flexibility and increasing usability. It includes several module including a network inference module `NetDeconvolute()` using graphical LASSO (glasso) to estimate a sparse inverse covariance matrix, creating a confounding-free partial correlation network among features from up to three -omics datasets.

iOmicsPASS has been improved to **iOmicsPASS+** allowing for higher flexibility and enabling applications to different types of omics data. Improvements include:

- **Specification of direction of association**
  Users may now specify the direction for every pair of interacting or co-varying molecule by adding an additional column in the network file. However, only molecules that show consistent sign of correlation in the empirical data as the user-specified direction will be considered.
- **Allows for a single network and input data**
  Previously, at least two data and two networks were required as input. Now, users can input only one single data and create co-expressions among the variables in the data with a single network file.
- **Addition of a Network estimation module** `NetDeconvolute()`
  Estimates a correlation network, linking the different features from up to three different data, using graphical LASSO (glasso) to estimate a sparse inverse covariance matrix, creating a confounding-free partial correlation network
- **New functions to help users compile and run iOmicsPASS using R**
  Functions included in the R package facilitate users to build `INSTALL.iOmicsPASS()`, create input parameter file `createInputParam()`, create prior probabilities `createPrior()` and run the software `iOmicsPASS.R()` in the R-console.
- **Addition of a Prediction module** `Predict.iOmicsPASS()`
  Uses the network signatures identified in the subentwork discovery module `iOmicsPASS.R()`to assign new samples to the phenotypic groups.
- **Adjustment for clinical information**
  Users can incorporate clinical information such as age, gender and BMI, to modify the prior class probabilities used for assigning samples to the different groups.

The figure below illustrates the overview of **iOmicsPASS+**



This vignette will cover the use of the various modularities in the R-package using a plasma protein and microRNA example datasets. For more information regarding the previous software (**iOmicsPASS**), refer to Koh et. al.(https://www.nature.com/articles/s41540-019-0099-y).

Setting up iOmicsPASS+

Users can either clone the github repository (https://github.com/cssblab/iOmicsPASSplus/) or download the zip-file directly to your local directory. Then extract the folder to install the R-package in R or R studio (download from https://cran.r-project.org).

To clone the repository to your local directory, in your command-line/Terminal window:

```
> git clone https://github.com/cssblab/iOmicsPASSplus.git
```

The software relies on `gcc` compiler to compile **iOmicsPASS** from within the R console. It also makes use of part of the boost library, distributed along with the software package under the Boost Software License (https://www.boost.org/LICENSE_1_0.txt).

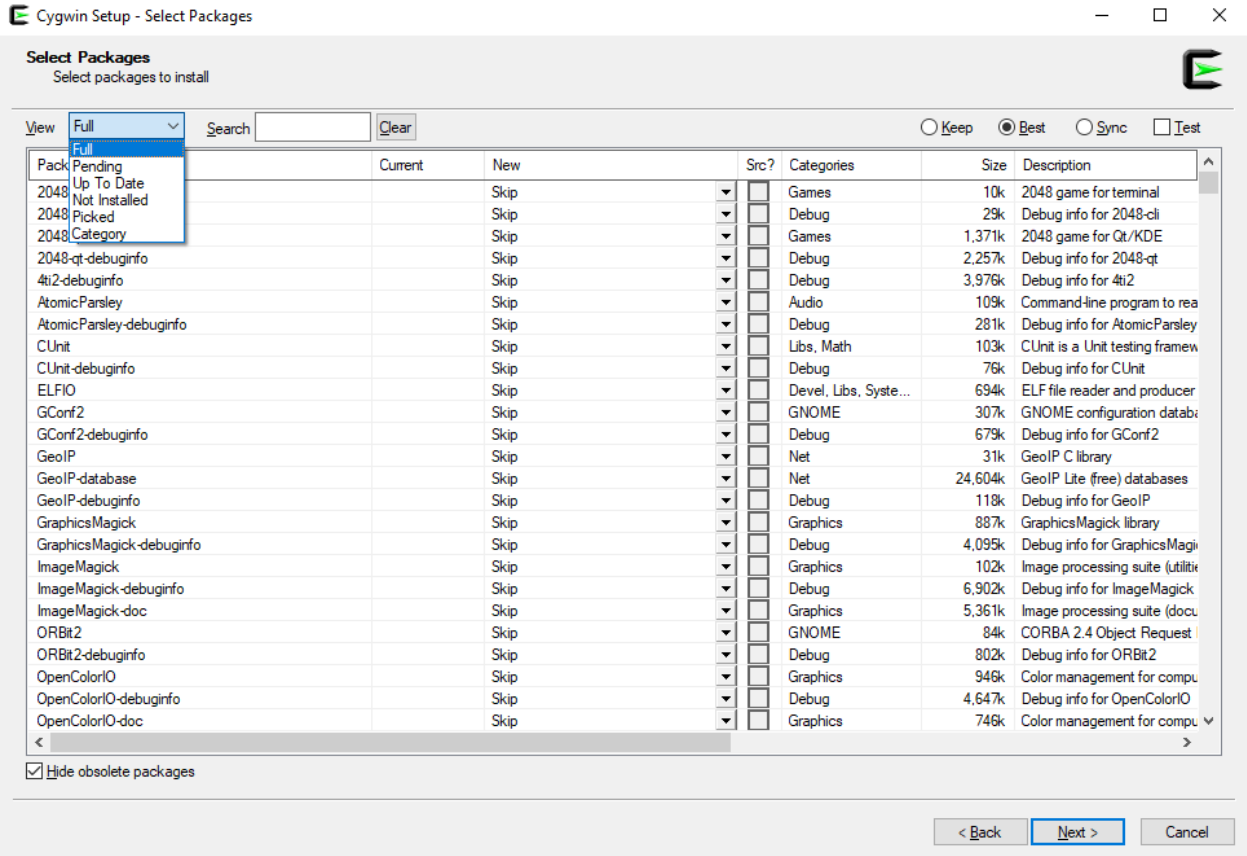
The R-packages has several external dependencies and its recommended that users install `Rtools` (https://cran.r-project.org/bin/windows/Rtools/) and the R-package `devtools` to allow for automatic installation of the dependencies:

```
if (!require(devtools)) install.packages("devtools")
```

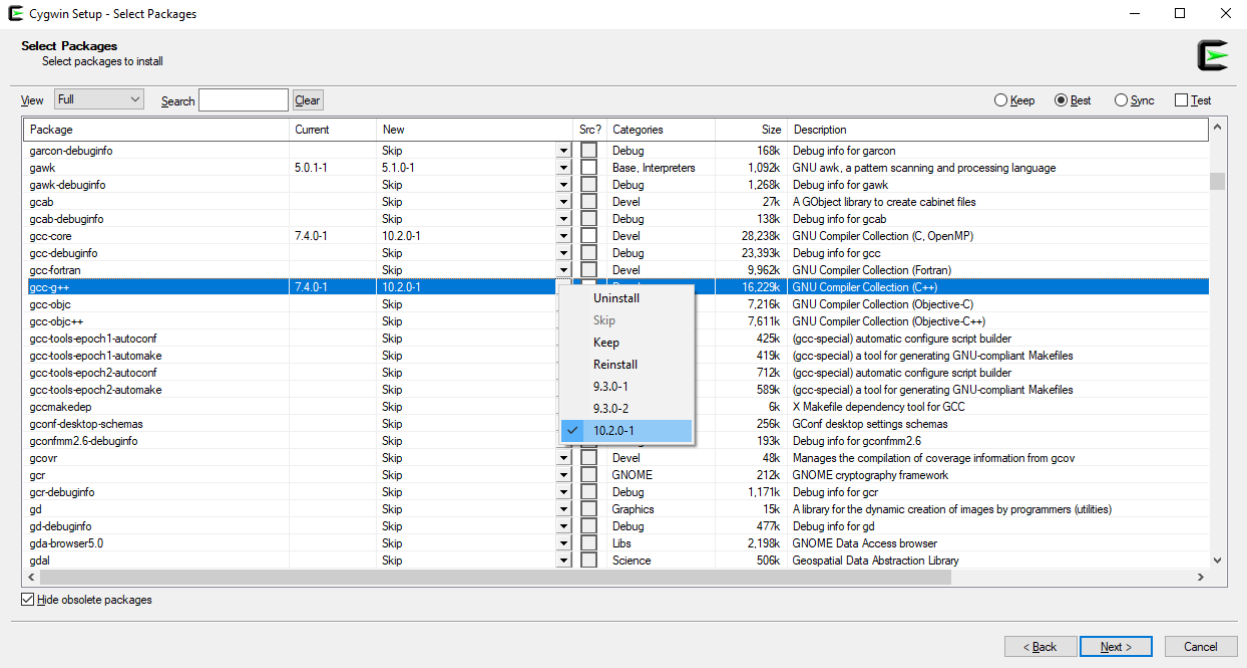
For Windows users

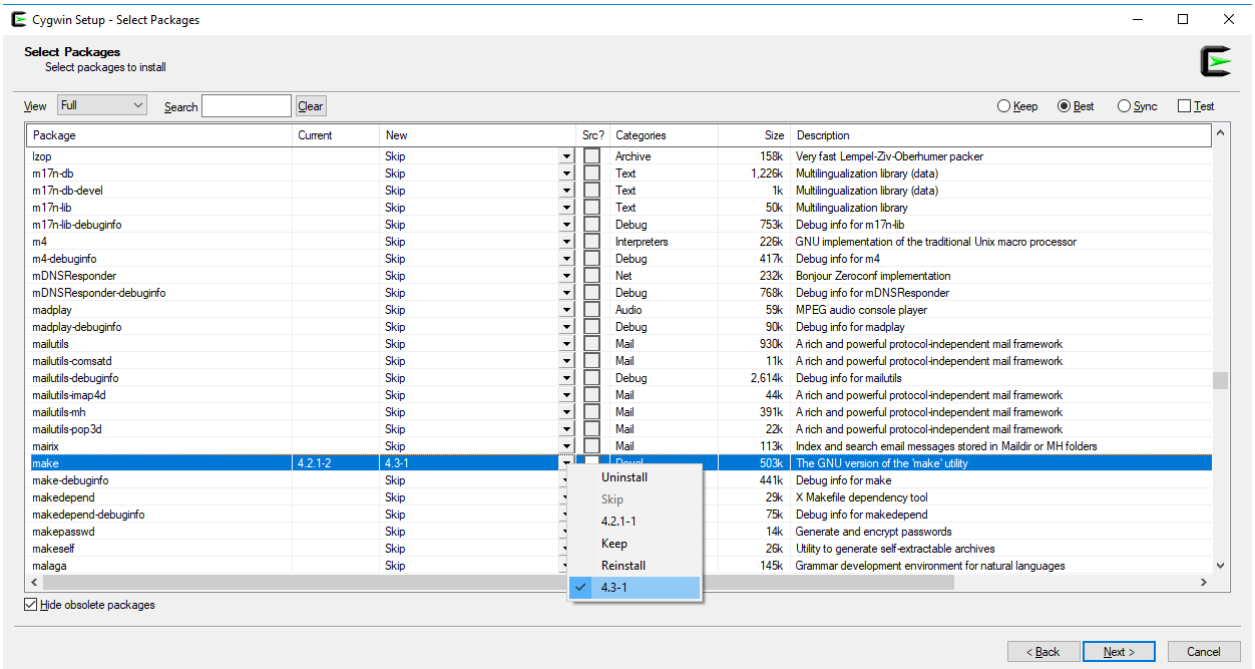
Executables for 64-bit Windows are included in the zip folder for direct use of **iOmicsPASS**. Else, installation of Cygwin is required (download available at https://www.cygwin.com/) to compile **iOmicsPASS**. Individual packages such as `bash`, `make` and `gcc` are released independently. Upon running setup, select `gcc` and `make` which is needed for compiling iOmicsPASS.

After picking the appropriate mirror in Cygwin installation, in the select packages menu, select "Full" in `View` drop-down menu:



Then, select search for `gcc` and `make` to right-click and install the latest packages. Click on Next to start the installation.





After installation, ensure that the directory is added to the system Path environment variable by navigating through the following:

`My Computer > Control Panel > Systems > Advanced system settings > Environment variable`

Then, click on edit to add `C:/cygwin64/bin` manually or use command prompt to create or set a variable permanently (as Administrator) by typing:

```
> setx PATH "C:/cygwin64/bin"
```

For Mac OS X/Linux users

You will require a GNU compiler to compile the C++ program which is available with the full installation of `Xcode` in Mac OS.

To make sure that "usr/local/bin" is already added in your Path variables, type the following in **Terminal**:

```
> PATH=$PATH:/usr/local/bin/
```

Note: You may need to restart your computer for the changes to take effect.

in R console

After cloning the repository **iOmicsPASSplus** from github or unzipping the zip folder, in R, set the working directory to the extracted folder and install the R-package as follows:

```
setwd("C:/PATH_TO_PROGRAM/iOmicsPASSplus/")
devtools::install_local("iOmicsPASSplus.tar.gz", dependencies =T)
## Alternatively ##
devtools::install_github("CSSBlab/iOmicsPASSplus", build_vignettes = TRUE)

# load the library
library(iOmicsPASSplus)
## only need to be run once to compile iOmicsPASS, creating a program in /bin folder.
INSTALL.iOmicsPASS()
```

Data: Plasma Protein and MicroRNA Biomarkers of Insulin Resistance

To illustrate the use of the various modules in the R-package, we will utilize the plasma protein and microRNA datasets from the Khoo et al., 2019 measured among 8 obese insulin-resistant (OIR, HOMA-IR>2.5) and 9 lean insulin-sensitive (LIS, HOMA-IR<1.0) normoglyceric males. The dataset `PhenotypeFile` describes the phenotype group of the 17 study participants as well as their age and BMI.

The example protein data `Tulip_Protein` contains 266 protein expression values across 17 samples. The original data contains 1,499 proteins and only those that were different between OIR and LIS (p-value <0.1) using 2-sample t-test were included in this example dataset.

The example microRNA data `Tulip_microRNA` contains 263 normalized microRNA copy number across 17 samples, quantified using multiplex RT-qPCR platform (MiRXES). The original data contains 368 microRNA probes and similarly, only those that were different between OIR and LIS (p-value <0.1) using 2-sample t-test were included in this example dataset.

```
## load the example data ##
data(Tulip_Protein)
data(Tulip_microRNA)
data(PhenotypeFile)

head(Tulip_Protein[,c(1:6)])
```

```
#>     Protein Tulip14 Tulip27 Tulip04 Tulip01 Tulip13
#> 2       A2M 223.319 310.373 268.934 125.636 111.264
#> 9      ACO1   0.569   0.947   0.945   1.428   0.900
#> 12     ACP5   1.186   0.737   0.975   0.704   0.970
#> 15    ACTN2   0.121   0.136   0.127   0.104   0.097
#> 19     ACY1   2.856   6.573   5.534   5.563   4.247
#> 27 ADAMTS13  15.991  12.204  10.936  12.271  14.831
```

```
head(Tulip_microRNA[,c(1:6)])
```

```
#>        miRNA    Tulip14    Tulip27    Tulip04    Tulip01    Tulip13
#> 1   miR-451a 15895.4945 11832.3047 24683.2339 17735.3449 19829.1541
#> 2   miR-1973 16608.7536  3727.1803  8016.7814 10483.9500  9000.1291
#> 3 miR-142-5p  3301.3771  2396.0467  3396.5876  3686.8594  2502.7142
#> 4  miR-16-5p   705.1141   472.3122   968.3487   823.8689   907.7191
#> 6 miR-486-5p   511.5152   287.9256   467.3003   460.4255   435.0923
#> 9 miR-15a-5p   348.4845   141.0079   388.1874   239.3890   271.4537
```

```
head(PhenotypeFile)
```

```
#>   TulipID Group Age      BMI
#> 1 Tulip14   OIR  27 27.77000
#> 2 Tulip27   OIR  32 30.77000
#> 3 Tulip04   OIR  27 30.42000
#> 4 Tulip01   OIR  23 27.52000
#> 5 Tulip13   OIR  30 34.28461
#> 6 Tulip20   OIR  30 31.31000
```

Example Network files and pathways for Enrichment

Distributed along with the R-package is two network files: (1) Protein-protein interaction (PPI) file and (2) microRNA-gene target file. The prior is a collection of protein interactions from iRefIndex and BioPlex 2.0, and the latter is experimentally validated microRNAs to gene targets from TargetScan. Also, we consolidated biological processes and pathways from ConsenusPathDB and Gene Ontology (GO) for the pathway enrichment module.

For the network files, the first two columns should be the names of the pair of interacting or associated features. The third column specifies the sign of the interaction/association. For PPI network, the sign of interaction will be "1" to indicate positive regulation and for microRNA-gene taget network, the signs will be "-1" to indicate negative inhibition.

```
data(PPI_network)
data(TargetScan_network)

head(PPI_network)
```

```
#>   geneA_genesym geneB_genesym sign
#> 1        ARPC1B         ARPC2    1
#> 2        ARPC1B         ARPC5    1
#> 3         ACTR2        ARPC1B    1
#> 4         ACTR3        ARPC1B    1
#> 5        ARPC1B         ARPC4    1
#> 6         MUCL1         PDIA5    1
```

```
head(TargetScan_network)
```

```
#>   GeneSym      miRNA sign
#> 1 ABHD14B miR-136-5p   -1
#> 2 ABHD14B miR-194-5p   -1
#> 3 ABHD14B   miR-320a   -1
#> 4 ABHD14B miR-338-3p   -1
#> 5  ABI3BP    miR-300   -1
#> 6  ABI3BP miR-381-3p   -1
```

For the biological pathway file, there should be three columns specifying the feature name, pathway identifier

and description of the pathway, respectively.

```
data(bioPathways)
head(bioPathways)
```

```
#>      Genesym  Pathwayid                                          Function
#> 1    ST8SIA5 KEGG:00604 Glycosphingolipid biosynthesis - ganglio series
#> 2 ST6GALNAC5 KEGG:00604 Glycosphingolipid biosynthesis - ganglio series
#> 3       GLB1 KEGG:00604 Glycosphingolipid biosynthesis - ganglio series
#> 4    ST3GAL5 KEGG:00604 Glycosphingolipid biosynthesis - ganglio series
#> 5    B3GALT4 KEGG:00604 Glycosphingolipid biosynthesis - ganglio series
#> 6    ST3GAL2 KEGG:00604 Glycosphingolipid biosynthesis - ganglio series
```

Network Inference Module #NetDeconvolute()

This module estimates a sparse partial correlation network between the different types of data, using an existing R-package `Huge` which carries out GLASSO. Users can specify up to three types of data for the network inference. Each data is first standardized to unit standard deviation and concatenated into a single matrix. Then, principal component analysis (PCA) is used to identify any potential outliers (i.e. more than 4 SDs from the median of PC1 and PC2).

There are two proposed ways to create a pseudo network: (1) **Supervised approach** that is completely driven by the data and (2) **Hybrid approach** that combines a known network as prior and network derived from the data to produce a resulting network.

The **supervised approach** is useful when studying a less well-annotated organism such as a new strain of viruses or a community of bacteria of which little understanding of how the biological system functions. Whereas a **hybrid approach** may be more useful when there is limited understanding of how the various biomolecules interact or co-vary in abundance. For instance, lipid species from the same class tend to show correlated variation across biological conditions, but there remains little understanding of how various lipid species from different class interact with one another.

Supervised approach

In the supervised approach, the sample covariance matrix S is computed and corrected to the nearest positive semi-definite (PSD) matrix if any of its eigenvalues are negative using the approach by Nicholas J. Higham, 1988. Upon ensuring that the covariance matrix satisfies the PSD property, graphical LASSO in `huge` R package is carried out next. A vector of lambda values are used to tune the L_1-penalty term in the model to yield a grid of corresponding penalized log-likelihoods.

In `NetDeconvolute()` function, model selection criteria, including AIC, BIC, e-BIC and cross-validation (CV), are incorporated to help users to select an optimal regularization parameter λ that minimizes AIC, BIC, e-BIC or maximizes the CV value. Users may choose to specify their own vector of lambda values, otherwise, the software will automatically generate a grid of 30 lambda values that is exponentially decreasing from 1 to 0.01:

$$\lambda_{grid} = \{\lambda_1, \lambda_2, ..., \lambda_{30}\} = exp\{log(1), log(0.853), ..., log(0.01)\}.$$

The selected regularization parameter λ is then used to refit the GLASSO model to yield the corresponding precision matrix and the partial correlation matrix. The non-zero entries in this matrix is then converted into an edge-level network file, indicating the direction of association, to be used for running the predictive analysis module `iOmicsPASS.R()`. The partial correlation between molecule i and j is calculated by Mohsen Pourahmadi, 2011:

$$\tilde{\rho}_{i,j} = \frac{-\hat{\omega}_{i,j}}{\sqrt{\hat{\omega}_{i,j}\hat{\omega}_{i,j}}}$$

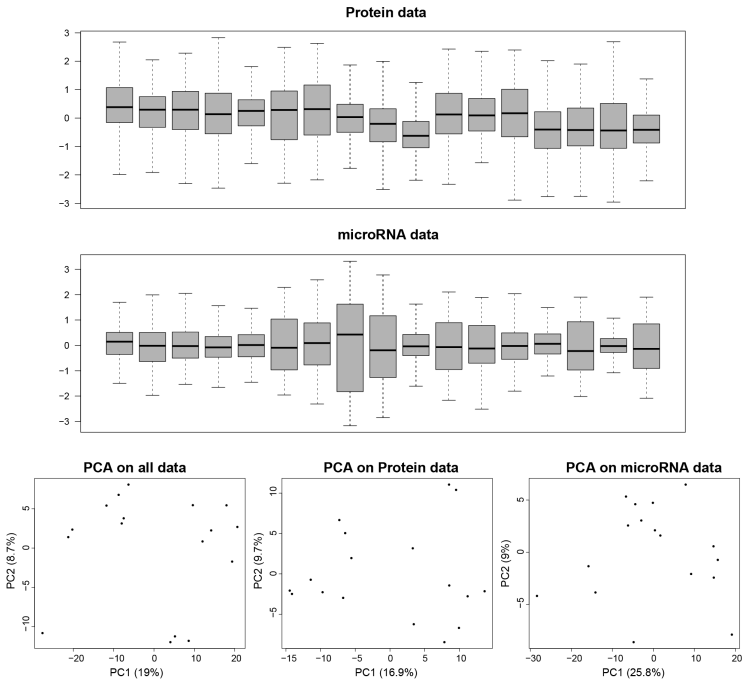
where $\hat{\omega}_{i,j}$ represents the $(i,j)^{th}$ entry in the estimated precision matrix, $\hat{\Omega}$.

Example Here, lets try to estimate a network connecting the plasma protein and microRNA dataset using the supervised approach. First, let us perform calibration to try to find the optimal λ value. It does not matter which criterion to choose for now as all four model selection criteria will be plotted if `Calibration=TRUE`. However, if $n << p$, it's recommended to use extended-BIC (eBIC) to assess the model fit.

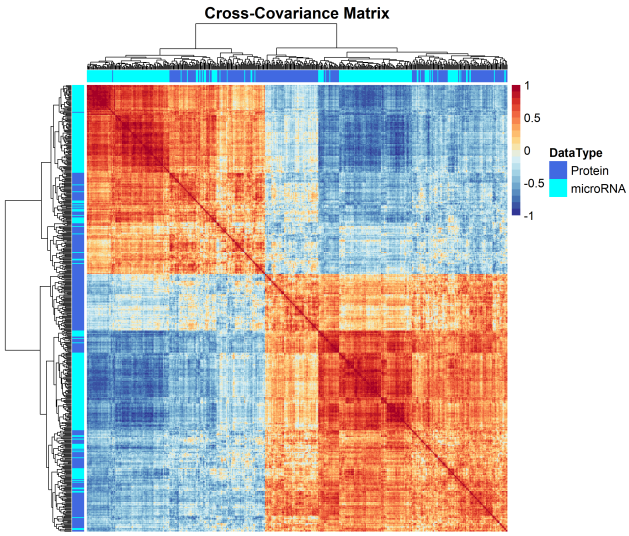
```
## creating an list object containing the two datasets and labeling them accordingly
row.names(Tulip_Protein) = Tulip_Protein$Protein
row.names(Tulip_microRNA) = Tulip_microRNA$miRNA
Tulip_Protein = Tulip_Protein[,-1]
Tulip_microRNA = Tulip_microRNA[,-1]
inputDat=list(Tulip_Protein, Tulip_microRNA)
names(inputDat) = c("Protein","microRNA")

NetDeconvolute(inputDat, option=1,log.transform=TRUE, tag="supervised",criterion="eBIC",Calibration=TRU
```

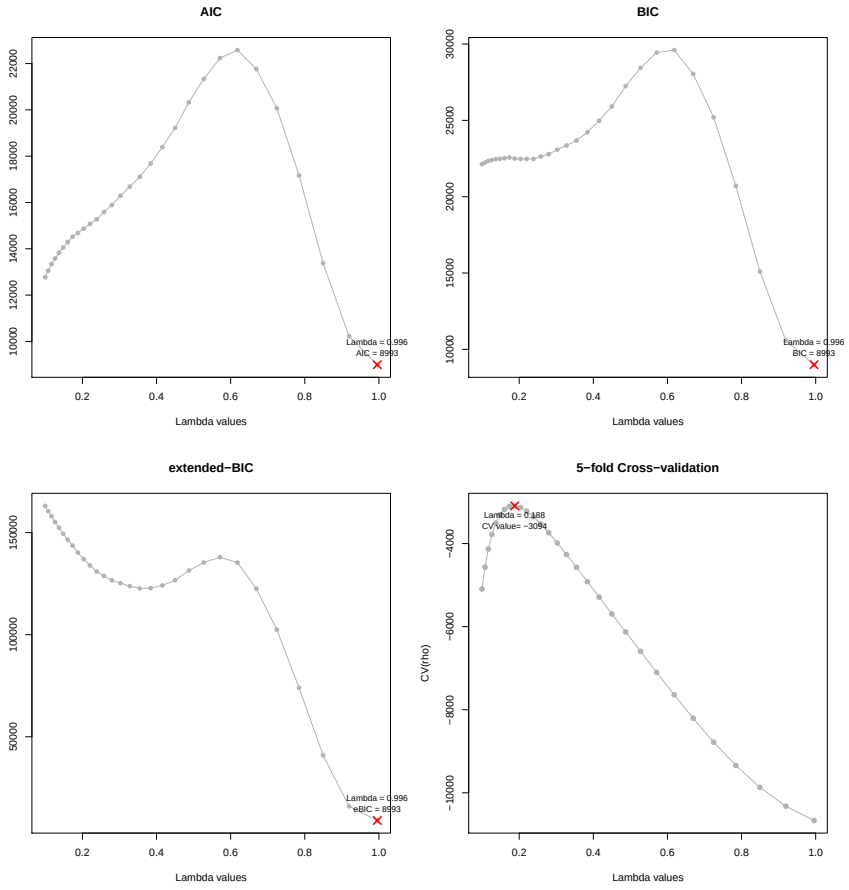
Graphical outputs such as boxplots and PCA plots are generated by default to help users to identify possible outlying samples that will be colored in red. Here, all the 17 samples passed the quality check (QC).



At the same time, the function will also produce the heatmap of the cross-covariance matrix, concatenating the datasets after standardization.



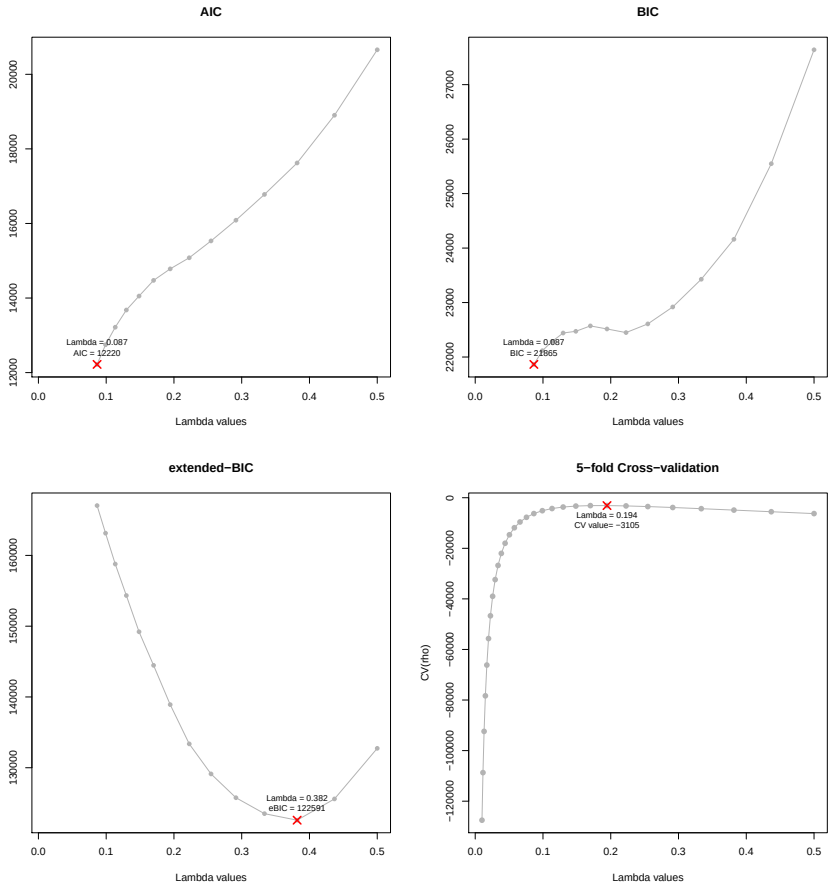
Next, inspecting the calibration plots, we noticed that the four model criterion starts declining steeply after 0.6 and there is a dip in between 0.2 to 0.5 for eBIC.



Thus, we shall re-define a narrower lambda vector for running the model fit again.

```
## Refining a narrower lambda vector ##
lambda_new=exp(seq(log(0.5),log(0.01), length=30))
NetDeconvolute(inputDat, option=1,log.transform=TRUE, tag="supervised2",criterion="eBIC", Calibration=TI
```

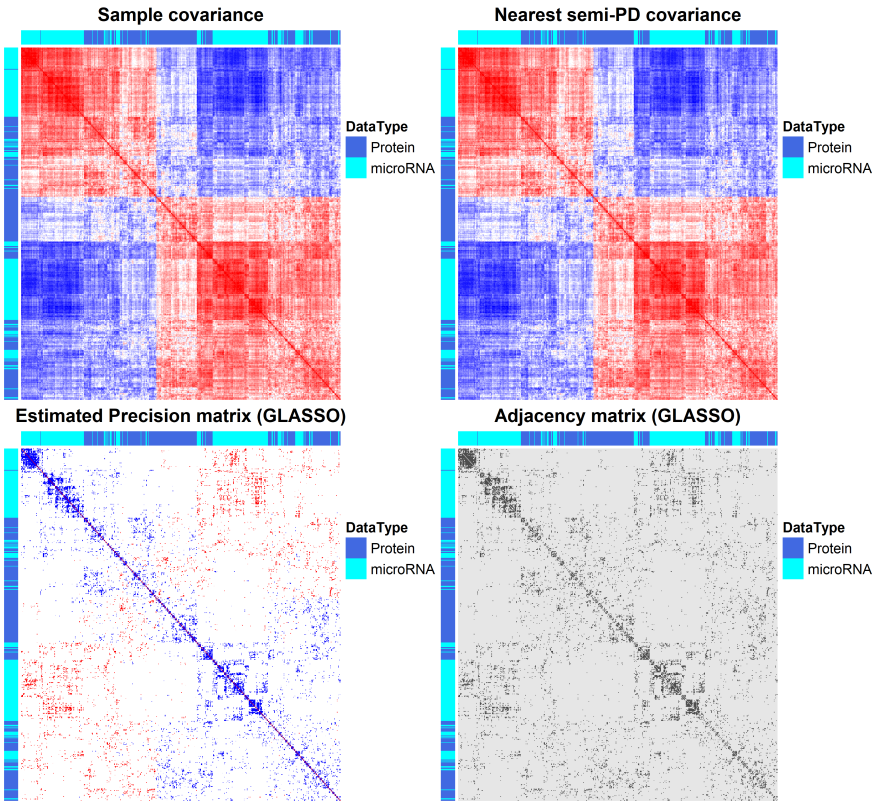
Let us now inspect the new calibration plots, we see that the model fit achieved the lowest eBIC when $\lambda = 0.382$ and highest cross-validation score at $\lambda = 0.194$.



We will select $\lambda = 0.382$ and set `Calibration=FALSE` to continue the estimation of the precision matrix.

```
NetDeconvolute(inputDat, option=1,log.transform=TRUE,tag="supervised",criterion="eBIC",
Calibration=FALSE, optLambda=0.382,verbose=TRUE)
```

Along with several `.txt` files as output, a graphical output consisting of four heatmaps showing the progression of the sample covariance matrix (top left), to the correction to the nearest SPD matrix (top right), to the estimated precision matrix (bottom left) and the corresponding adjacency matrix (bottom right) that forms the estimated network.



There are four `.txt` file output generated and they include:

- `Combined_data.txt`
  A data matrix with p rows (total features) across n samples derived by concatenating the different input data after standardizing and removing plausible outliers (data QC step).
- `glasso_estimated_icov.txt`
  A sparse $p \times p$ data matrix describing the estimated inverse covariance matrix or the precision matrix. The zero entries describe conditional-independence among the pair of features.
- `PartialCorrelation_icov.txt`
  A sparse $p \times p$ data matrix describing the partial correlation between each pair of feature converted from the estimated inverse covariance matrix or the precision matrix.
- `Estimated_Network_glasso.txt`
  A data file with seven columns (i.e. nodeA, nodeB, dir, partialcor, DatatypeA, DatatypeB, EdgeType) and each row describing a pair of feature that have an non-zero entry in the estimated precision matrix. The data is in the format required as a network file in running the predictive analysis module `iOmicsPASS.R`.

Note: Both `Combined_data.txt` and `Estimated_Network_glasso.txt` will be copied and placed in `/iOmicsPASS/inputFiles/` folder automatically for the predictive subnetwork discovery module.

Hybrid approach

In this approach, users can supplement a network file with known relationships between features and using it as a prior (matrix P) and GLASSO from the **supervised approach** will also be carried to estimate a network (matrix $\hat{\Omega}$). Then, both networks are combined to yield a fused network (matrix F).

Let us first define the following matrices: