Package 'iOmicsPASSplus'

February 18, 2022

Type Package

Title A R-package for iOmicsPASS

Version 1.0

Description This is a R-package for carrying out multiple-omics data integration using networks, in a systems biology approach. The tool integrates up to three types of -omics data using networks as input to create co-expression scores. Then using a supervised approach, it identifies a set of subnetwork signatures that best discriminate between phenotypic outcomes. If there is no known network, the tool includes a network inference module that estimates a partial correlation network. The package also includes additional functionalities such as a prediction module for classifying samples based on identified signatures.

License GPL (>=2)

URL https://github.com/CSSBlab/iOmicsPASSplus

NeedsCompilation yes

Encoding UTF-8

LazyData false

Imports corpcor,
gplots,
grid,
gridExtra,
gridGraphics,
huge,
nnet,
pheatmap,
pracma,
RColorBrewer

RoxygenNote 7.1.2

Suggests knitr,
devtools,
rmarkdown

VignetteBuilder knitr

Depends R (>= 2.10)

R topics documented:

`iOmicsPASSplus-package` *Integrative -Omics for Prediction Analysis of Subnetwork Signatures – version 2 extended*

Description

`iOmicsPASSplus` 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 to estimate a sparse inverse covariance matrix, creating a confounding-free partial correlation network among features from up to three -omics datasets. The estimated network can be used to create co-expression scores in `iOmicsPASS.R()` to identify predictive signatures that best separates phenotypic outcomes. Those signatures can also be used to assign new samples with the same -omics data available into the phenotypic outcomes using `Predict.iOmicsPASS()`.

Note

User needs to have a GNU compiler to compile the C++ program. Detailed steps can be found in the vignette.

Author(s)

Hiromi WL Koh, Hyungwon Choi

`bioPathways` *Biological pathways collated from multiple databases.*

Description

A data frame with information of the biological pathways that each gene symbol is associated with, collated from multiple databases including Gene Ontology (GO) Consortium, KEGG, Reactome Pathway database and ConsensusPathDB.

Usage

```
data(bioPathways)
```

Format

A data frame with 17,250 unique gene symbols and 14,598 unique biological pathways:

Genesym Gene symbol

Pathwayid Pathway Identifier

Function Description of the pathway

For further details, see https://pubmed.ncbi.nlm.nih.gov/31024340/

---

createInputParam *Create Input parameter file for iOmicsPASS*

Description

Create Input parameter file for iOmicsPASS

Usage

```
createInputParam(
  data.X,
  data.Y = NULL,
  data.Z = NULL,
  within.net = NULL,
  btw.net = NULL,
  dir = "iOmicsPASS/inputFiles/",
  phenotype = NULL,
  pathway = NULL,
  standardize.data = T,
  log.transform = F,
  normalizeBy = NULL,
  min.obs = 5,
  min.prop = 0.8,
  knn.impute = F,
  knn.k = 10,
  max.block.impute = 1000,
  Cross.Validate = TRUE,
  num.Kfold = 5,
  min.thres = NULL,
  Enrichment = TRUE,
  tag = NULL,
  usePrior = FALSE,
  priorfile = NULL,
  bg.prop = 0.2,
  min.bg.size = 3,
  min.sig.size = 1
)
```

Arguments

data.X	dataframe X
data.Y	dataframe Y
data.Z	dataframe Z
within.net	network file for creating edges within X
btw.net	network file for creating edges between X and Y
dir	directory pointing to the location of the files
phenotype	file with group sample information
pathway	pathway file for network enrichment
standardize.data	whether to standardize each data (default=TRUE)
log.transform	whether to log-transform each data (default=FALSE)
normalizeBy	Either "Y" or "Z". if "Y", coexpression scores in X are normalized by Y and if "Z", coexpression scores in X are normalized by Z.
min.obs	minimum number of non-missing observations requires across each feature in each phenotypic outcome.
min.prop	minimum proportion of non-missing observations requires across each feature in each phenotypic outcome.
knn.impute	whether to perform K-nearest neighbor imputation for missing entries
knn.k	number of folds for K-nearest neighbor imputation
max.block.impute	number of blocks of samples to consider in KNN imputation
Cross.Validate	Whether to run cross-validation (default=TRUE)
num.Kfold	number of folds in CV
min.thres	threshold to be used to derive the shrunken centroids in iOmicsPASS
Enrichment	Whether to run network enrichment (default=TRUE)
tag	a string that will be appended to the end of the output files like a tag.
usePrior	whether to use an input prior to classify samples in discriminant model. If false, equal prior will be used.(default=FALSE)
priorfile	file of the prior probabilities calculated from createPrior().
bg.prop	minimum proportion of features in each pathway that are also present in the background list.
min.bg.size	minimum number of features in each pathway that are also present in the background list.
min.sig.size	minimum number of features in each pathway that are both part of the signature and present in the background list.

Value

a parameter file for input into iOmicsPASS.R()

Examples

```
data(PhenotypeFile)
data(bioPathways)

## Running with estimated network from NetDeconvolute() ##
createInputParam(data.X="Combined_data.txt", within.net="Estimated_Network_glasso.txt",
phenotype =PhenotypeFile, Enrichment=FALSE)

## with Network-based pathway Enrichment ##
createInputParam(data.X="Combined_data.txt", within.net="Estimated_Network_glasso.txt",
phenotype =PhenotypeFile, pathway =bioPathways, Enrichment=TRUE)

data(Tulip_Protein)
data(Tulip_microRNA)
data(PPI_network)
data(TargetScan_network)

## Using known biological networks ##
createInputParam(data.X=Tulip_Protein,data.Y=Tulip_microRNA,phenotype =PhenotypeFile,
log.transform=TRUE,btw.net=TargetScan_network,within.net=PPI_network, Enrichment=FALSE)
```

`createPrior` *Create prior probabilities for discriminant model in iOmicsPASS*

Description

Create prior probabilities for discriminant model in iOmicsPASS

Usage

```
createPrior(
  FILE,
  y = NULL,
  var.cat = NULL,
  testFile = NULL,
  predict = FALSE,
  outputDir = "iOmicsPASS/inputFiles/",
  tag = NULL
)
```

Arguments

`FILE`	dataframe with sample ID in the row names and clinical characteristics (for adjustment) in the columns.
`y`	character string of the phenotypic or group variable.
`var.cat`	a string or a vector of string matching the column names of FILE that are categorical variables.
`testFile`	dataframe for test samples matching the colnames and variables in FILE
`predict`	Whether or not to predict probabilities on a test dataset
`outputDir`	directory to write the outputfile
`tag`	a string that will be appended to the end of the output files like a tag.

Value

a dataframe with class probabilities assigned to each sample, used as prior in iOmicsPASS

Examples

```
data(PhenotypeFile)
PhenotypeFile2 = PhenotypeFile
row.names(PhenotypeFile2) = PhenotypeFile$TulipID
PhenotypeFile2=PhenotypeFile2[,-1]
prior_out = createPrior(PhenotypeFile2, y = "Group",outputDir = "iOmicsPASS/inputFiles/")
prior_test = createPrior(PhenotypeFile2, y = "Group",predict=TRUE, testFile = TestData)
```

INSTALL.iOmicsPASS *Compile and build the C++ program for running iOmicsPASS*

Description

Compile and build the C++ program for running iOmicsPASS

Usage

```
INSTALL.iOmicsPASS(currDir = NULL)
```

Arguments

currDir	Current directory where iOmicsPASSv2plus is unzipped and where the makefile is.

Examples

```
# set your working directory to iOmicsPASSv2plus folder #
INSTALL.iOmicsPASS()
```

---

iOmicsPASS.R *Carry out Predictive Analysis of Subnetwork Signatures*

Description

Carry out Predictive Analysis of Subnetwork Signatures

Usage

```
iOmicsPASS.R(
  ff = "input.param",
  outputDir = "iOmicsPASS/Output/",
  Cross.Validate = TRUE,
  plotCV = TRUE
)
```

Arguments

ff	input parameter
outputDir	directory to write the output files (default output to "iOmicsPASS/Output/")
Cross.Validate	Whether or not cross-validation was performed in iOmicsPASS (default = TRUE)
plotCV	Whether or not to plot the performance of the cross-validation (default = TRUE)

Value

dataframe with iOmicsPASS parameters used for prediction, multiple text files and plots written to output directory.

- AttributesTable.txt - Attributes table for every node in the network to be used in Cytoscape.
- BGlist.txt - A list of edges that are present in both the network and the input data.
- CVerrors.txt - A table of grid of threshold and their corresponding mean cross-validation errors and selected edges.
- CVplot_Penalty.pdf - A plot of the mean cross-validation error against the grid of threshold used to shrink the centroid.
- EdgesSelected_minThres.txt - dataframe with selected predictive edges and the dik scores for each phenotype that can be used to visualize directly in Cytoscape.
- Features_Neighbors.txt - Attributes table with neighboring node information for each node in the network.
- PredictiveEdges_Parameters.txt - A dataframe with with iOmicsPASS parameters used for prediction.
- SampleClass_Probabilities.txt - A dataframe with class probabilities assigned to the samples input.
- Ztransform_XXX.txt - Standardized data X/Y/Z.
- XXX_Enrichment_up.txt - a dataframe with the results of the enrichment of the selected edges (over-expressed) out of all the edges in the network for each class.
- XXX_Enrichment_down.txt - a dataframe with the results of the enrichment of the selected edges (under-expressed) out of all the edges in the network for each class.

Examples

```
data(PhenotypeFile)
data(bioPathways)

## Running with estimated network from NetDeconvolute() ##
createInputParam(data.X="Combined_data.txt", within.net="Estimated_Network_glasso.txt",
phenotype =PhenotypeFile,Enrichment=FALSE)
iOmicsPASS.output<-iOmicsPASS.R(ff="input_param")

## pick optimal threshold as 2.4 and turn off CV ##
## rerun with network-based pathway enrichment ##
createInputParam(data.X="Combined_data.txt", within.net="Estimated_Network_glasso.txt",
phenotype=PhenotypeFile,pathway=bioPathways,Enrichment=TRUE,min.thres=2.4,Cross.Validate=FALSE)

iOmicsPASS.output<-iOmicsPASS.R(ff="input_param",Cross.Validate=FALSE, plotCV=FALSE)

data(Tulip_Protein)
data(Tulip_microRNA)
```

```
data(PPI_network)
data(TargetScan_network)

## Using known biological networks ##
createInputParam(data.X=Tulip_Protein,data.Y=Tulip_microRNA,phenotype=PhenotypeFile,
log.transform=TRUE,within.net=PPI_network,btw.net=TargetScan_network,Cross.Validate=TRUE,
Enrichment=FALSE, tag="KnownNetwork")
iOmicsPASS.output<-iOmicsPASS.R(ff="input_param_KnownNetwork")
```

`NetDeconvolute` *Network inference module*

Description

This function helps to create an inferred partial correlation network via two approaches:(1) Supervised and (2) Hybrid. The supervised approach uses glasso to estimate a sparse inversed covariance matrix completely from the data alone. The hybrid approach combines a prior network with an estimated network from the supervised method to create a fused network.

Usage

```
NetDeconvolute(
  inputDat,
  detectOutliers = TRUE,
  option,
  Calibration = TRUE,
  tag = NULL,
  log.transform = TRUE,
  cutoff_fusedmat = 0.5,
  standardization = TRUE,
  NetworkFile = NULL,
  toSkipPCA = FALSE,
  Plot.PCA = TRUE,
  Plot.modelSelect = TRUE,
  Plot.Covariance = TRUE,
  useCorrelation = FALSE,
  lambda.vec = NULL,
  numLambda = 30,
  numFolds = 5,
  criterion = c("AIC", "BIC", "eBIC", "CV"),
  optLambda = NULL,
  verbose = F
)
```

Arguments

`inputDat`	a list object containing up to three matrices (X, Y and Z) with features as the row.names and sample IDs on the columns.
`detectOutliers`	whether to carry out outlier filtering using PCA. Outliers will be removed if outside of 4 SDs from the median of the first two PCs.

option	option=1 for supervised method and option=2 for hybrid method.
Calibration	whether to carry out the model selection step by fitting glasso on a grid of lambda values. To be turned off after picking an optimal lambda.
tag	a string that will be appended to the end of the output files like a tag.
log.transform	whether to log-transform each matrix in inputDat (default=TRUE).
cutoff_fusedmat	cut-off value between 0 to 1 used to convert the fused matrix into an adjacency matrix.
standardization	whether to carry out standardization for each data in inputDat. Recommended if integrating multiple datasets before computing cross-covariance matrix.
NetworkFile	prior network file. File should indicate which data type the feature comes from (i.e. X or Y) using "NodeA_DT" and "NodeB_DT" in case of same gene name. Required if option=2 (hybrid).
toSkipPCA	whether to carry out principal component analysis, not recommended if there are too many missing entries in data (default = FALSE).
Plot.PCA	whether to output the plots of PCA of the input data (default=TRUE)
Plot.modelSelect	whether to output the plots to help pick a regularization parameter based on AIC, BIC, eBIC and CV (default=TRUE).
Plot.Covariance	whether to output the heatmap illustrating the cross-covariance matrix. Not recommended if the dimension of the matrix exceeds 2000.
useCorrelation	whether to compute correlation instead of covariance matrix (default = FALSE). If correlation is selected, standardization of variables is not required.
lambda.vec	a vector of regularization parameter to use in the soft-thresholding in glasso.
numLambda	number of values in the regularization parameter.
numFolds	number of folds in the CV model selection
criterion	c("AIC","BIC","eBIC","CV"). Model selection criteria to use to pick the parameter that yield the best penalized log-likelihood.
optLambda	value of the regularization parameter to use to estimate the sparse inverse covariance.
verbose	whether to output steps and update status of running the function.

Value

Multiple output files with inferred network and estimated partial correlation matrix.

- PCAplot.pdf - A panel of PCA plots of individual -omics data and combined data highlighting outliers.
- Boxplots_outliers.pdf - A panel of boxplots of each -omics data across the samples, highlighting outliers
- Heatmap_CrossCovarianceMatrix.pdf - A heatmap of the calculated cross-covariance matrix of the combined data.
- CalibrationPlots_glasso.pdf - A multi-panel plot of four model selection criteria against a grid of lambda values to help user pick the optimal value.
- Plots_glasso.png - a multipanel plot illustrating the derivation of the precision matrix from the cross-covariance matrix.

- Plots_Hybridmethod.png - a multipanel plot illustrating the derivation of the precision matrix from supervised approach and combining the prior information to form the fused matrix in the hybrid approach.
- glasso_estimated_icov.txt - a text file containing the estimated sparse inverse of the covariance matrix (also known as precision matrix).
- PartialCorrelation_icov.txt - a text file containing the corresponding partial correlation calculated from the estimated inverse of covariance.
- Combined_data.txt - A dataframe with the various -omics data concatenated together after standardization.
- Estimated_Network_glasso.txt - The corresponding network file from the estimated precision matrix from the supervised approach.
- Fusednetwork_hybridmethod.txt - The corresponding network file from the estimated fused matrix from the hybrid approach.

Examples

```
data(Tulip_Protein)
data(Tulip_microRNA)

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")

########################
# supervised approach #
########################

## using covariance matrix to estimate network ##
NetDeconvolute(inputDat, option=1,log.transform=TRUE, tag="supervised",criterion="eBIC",
Calibration=TRUE, verbose=T)

###using correlation matrix to estimate network ###
NetDeconvolute(inputDat, option=1,log.transform=TRUE, tag= "correlation",criterion="eBIC",
Calibration=TRUE, useCorrelation=T, standardization=F, verbose=T)

#' ### Note that after standardization, covariance and correlation is the same ###
### Correlation can be used if data cannot be standardized, for example when looking at changes (post-pre) ###

## continuing if using covariance since we are working with expression data,
## we refine the lambda vector to zoom into a specific window ##
lambda_new=exp(seq(log(0.5),log(0.01), length=30))

NetDeconvolute(inputDat, option=1,log.transform=TRUE, tag= "supervised2",criterion="eBIC",
Calibration=TRUE, lambda.vec=lambda_new, verbose=T)

# pick a threshold using the calibration plot with highest CVscore or lowest AIC/BIC/eBIC.
# Then turn off calibration to FALSE to re-running the cross-validation for glasso.
NetDeconvolute(inputDat, option=1,log.transform=TRUE,tag= "supervised",criterion="eBIC",
Calibration=FALSE, optLambda=0.38,verbose=TRUE)
```

```
###################
# hybrid approach #
###################
data(TargetScan_network)
data(PPI_network)
TargetScan_network$NodeA_DT = "X"
TargetScan_network$NodeB_DT = "Y"
PPI_network$NodeA_DT = "X"
PPI_network$NodeB_DT = "X"

colnames(TargetScan_network) = c("NodeA","NodeB","Dir","NodeA_DT","NodeB_DT")
colnames(PPI_network) = c("NodeA","NodeB","Dir","NodeA_DT","NodeB_DT")
PriorNet = rbind(TargetScan_network,PPI_network)

NetDeconvolute(inputDat, option=2, NetworkFile=PriorNet, tag="hybrid",criterion="eBIC",
log.transform=TRUE,Calibration=FALSE, optLambda=0.382, verbose=TRUE)
```

PhenotypeFile *Phenotype file for the Tulip study*

Description

A data frame with two columns containing the phenotype assigned to the 18 subjects where 8 were obese insulin resistant (OIR, HOMA-IR >2.5) and 9 were lean insulin-sensitive (LIS, HOMA-IR <1.0) normoglycemic males.

Usage

```
data(PhenotypeFile)
```

Format

A two-column data frame with 17 subjects (row) and their phenotype:

TulipID Subject identifier in the Tuplip study

Group Phenotypic group of each subject. OIR for obese insulin resistant and LIS for lean insulin sensitive.

Age Age of the subject

BMI Body mass index of each subject

For further details, see https://pubmed.ncbi.nlm.nih.gov/31024340/

Plot_CrossValidation *Plot Cross-validation performance*

Description

Plot Cross-validation performance

Usage

```
Plot_CrossValidation(outputdir)
```

Arguments

outputdir	directory pointing to CVerrors.txt.

Value

a PDF file with the performance plot

Examples

```
Plot_CrossValidation("iOmicsPASS/output/")
```

PPI_network *Protein-Protein interaction network file*

Description

A data frame with three columns where the first two columns contain the pairs of protein identifiers that chemically/physically interact with each other. The third column indicates the sign of the direction of that interaction, which are assumed to be all positive interactions here.

Usage

```
data(PPI_network)
```

Format

A data frame with 1,201 unique proteins containing 1,499 interactions:

geneA_genesym Gene symbol of Protein A

geneB_genesym Gene symbol of Protein B that is interacting with Protein A

sign Direction of protein-protein interaction, assumed to be all positive (indicated by 1)

For further details, see https://pubmed.ncbi.nlm.nih.gov/31024340/

`Predict.iOmicsPASS` *Classification on external dataset*

Description

Classification on external dataset

Usage

```
Predict.iOmicsPASS(
  file,
  newData,
  standardize = TRUE,
  usePrior = FALSE,
  prior = NULL,
  prop = 0.8
)
```

Arguments

`file`	output from running iOmicsPASS.R().
`newData`	Test data where the various -omics data are concatenated over the same samples. First column should be the list of features to be matched to the predictive signatures.
`standardize`	whether to perform standardization on the data.
`usePrior`	boolean.
`prior`	filename of prior.
`prop`	proportion of features in test data that are part of the predictive signature. (default=0.8)

Value

a dataframe with the class probabilities for each sample

Examples

```
data(Tulip_Protein)
data(Tulip_microRNA)
data(PhenotypeFile)
data(PPI_network)
data(TargetScan_network)

row.names(Tulip_Protein) = Tulip_Protein$Protein
row.names(Tulip_microRNA) = Tulip_microRNA$miRNA


## create a testData by randomly sampling from original data ##
set.seed(22)
sample_pick = sample(PhenotypeFile$TulipID, 6)   ## 4 LIS and 2 OIR

Tulip_Protein_test = Tulip_Protein[,c(1,match(sample_pick, colnames(Tulip_Protein)))]
```

```
Tulip_microRNA_test = Tulip_microRNA[,c(1,match(sample_pick, colnames(Tulip_microRNA)))]

PhenotypeFile_test = PhenotypeFile[which(PhenotypeFile$TulipID %in% sample_pick),]

row.names(PhenotypeFile_test) = PhenotypeFile_test$TulipID
row.names(PhenotypeFile) = PhenotypeFile$TulipID

prior_train = createPrior(PhenotypeFile[,-1], y = "Group",predict=FALSE)

createInputParam(data.X=Tulip_Protein, data.Y=Tulip_microRNA, within.net=PPI_network,
 btw.net=TargetScan_network,log.transform=TRUE,phenotype =PhenotypeFile,
 Enrichment=FALSE,usePrior = TRUE, priorfile = prior_train, tag="train")

iOmicsPASS.train <- iOmicsPASS.R(ff = "input_param_train")

createInputParam(data.X=Tulip_Protein, data.Y=Tulip_microRNA, within.net=PPI_network,
 btw.net=TargetScan_network,log.transform=TRUE,phenotype =PhenotypeFile,Cross.Validate=FALSE,
 min.thres=1.8,Enrichment=FALSE,usePrior = TRUE, priorfile = prior_train, tag="train")
 iOmicsPASS.train <- iOmicsPASS.R(ff = "input_param_train", plotCV=FALSE)

 ## predict test data using signatures from training data ##
testData=rbind(Tulip_Protein_test[,-1],Tulip_microRNA_test[,-1])
testData=log2(testData)
testData = data.frame(row.names(testData), testData, check.names=FALSE)
colnames(testData)[1] = "Feature"

 ## remove group information ##
PhenotypeFile_test=PhenotypeFile_test[,-2]
prior_test = createPrior(PhenotypeFile[,-1], y = "Group",predict=TRUE,
testFile = PhenotypeFile_test[,-1])
pred.out <- Predict.iOmicsPASS(iOmicsPASS.train , testData, standardize = TRUE,prop = 0.9,
 usePrior=T,prior=prior_test )
```

`TargetScan_network` *microRNA to target gene network file*

Description

A dataframe with three columns where the first two columns represent the miRNA probes and their genes targets. The third column indicates the direction of interaction, where microRNAs that are translation inhibitor of its target genes as negative and those that regulates gene translation as positive.

Usage

```
data(TargetScan_network)
```

Format

A data frame with 882 unique genes and 125 unique microRNAs with 8,533 microRNA-gene targets:

GeneSym Gene symbol

miRNA microRNA probe ID

sign Direction of miRNA-gene regulation where positive regulation of gene translation are indicated by 1 and inhibition of translation by -1.

For further details, see https://pubmed.ncbi.nlm.nih.gov/31024340/

`Tulip_microRNA` *microRNA data in Tulip study.*

Description

A dataset containing 263 normalized microRNA copy number using multiplex RT-qPCR platform, MiRXES. Original data contains 368 microRNA probes and only those that were different (p<0.1) between insulin resistant and insulin sensitive individuals using 2-sample t-test were included in this example data.

Usage

```
data(Tulip_microRNA)
```

Format

A data frame with 263 microRNA probes (rows) across 17 subjects (columns):

miRNA microRNA probes

TulipXX Sample ID in Tulip study

For further details, see https://pubmed.ncbi.nlm.nih.gov/31024340/

`Tulip_Protein` *Protein expression data in Tulip study. Original data contains 1,499 proteins quantified and only those that were different (p<0.1) between insulin resistant and insulin sensitive individuals using 2-sample t-test were included in this example data.*

Description

A dataset containing the 266 plasma protein abundance quantified by LC-MS across 17 males.

Usage

```
data(Tulip_Protein)
```

Format

A data frame with 1,499 proteins (rows) across 17 subjects (columns):

Protein Protein symbol

TulipXX Sample ID in Tulip study

For further details, see https://pubmed.ncbi.nlm.nih.gov/31024340/

Index