Package 'SmCCNet'

June 15, 2018

Type Package

Title Sparse multiple canonical correlation network analysis tool

Version 0.1.0

Date 2018-6-15

Author W. Jenny Shi [aut, cre],
Laura Saba [aut],
Katerina Kechris [aut]

Maintainer W. Jenny Shi <wjennyshi@gmail.com>

Description
A canoncial correlation based framework for integrating multiple omics data types and a quantitative phenotype of interest, and constructing phenotype-specific multi-omics networks.

URL https://github.com/KechrisLab/SmCCNet

BugReports https://github.com/KechrisLab/SmCCNet/issues

Depends R (>= 3.4.4)

Imports PMA, Matrix, pbapply, igraph

Suggests BiocStyle, knitr, rmarkdown

VignetteBuilder knitr

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

`getAbar` *Compute the similarity matrix based on one or more canonical correlation weight vectors.*

Description

Compute the similarity matrix based on the outer products of absolute canonical correlation weights.

Usage

```
getAbar(Ws, FeatureLabel = NULL)
```

Arguments

`Ws`	A canonical correlation weight vector or matrix. If `Ws` is a matrix, then each column corresponds to one weight vector.
`FeatureLabel`	A $1 \times p$ vector indicating feature names. If `FeatureLabel = NULL` (default), the feature names will be indices 1 through p, where p is the total number of omics features.

Value

A $p \times p$ symmetric non-negative matrix.

Examples

```
w <- matrix(rnorm(6), nrow = 3)
Ws <- apply(w, 2, function(x)return(x/sqrt(sum(x^2))))
abar <- getAbar(Ws)
```

`getMultiOmicsModules` *Extract multi-omics modules based on the similarity matrix.*

Description

Apply a hierarchical tree cutting to the similarity matrix and extract modules that contain both omics data types.

Usage

```
getMultiOmicsModules(Abar, P1, CutHeight = 1 - 0.1^10, PlotTree = TRUE)
```

Arguments

`Abar`	A similary matrix for all features (both omics data types).
`P1`	Total number of features for the first omics data type.
`CutHeight`	Height threshold for the hierarchical tree cutting. Default is $1 - 0.1^{10}$.
`PlotTree`	Logical. Whether to create a hierarchical tree plot.

Value

A list of multi-omics modules.

Examples

```
set.seed(123)
w <- rnorm(5)
w <- w/sqrt(sum(w^2))
abar <- getAbar(w)
modules <- getMultiOmicsModules(abar, P1 = 2, CutHeight = 0.5)
```

`getRobustPseudoWeights` *Calculate the canonical correlation weights based on sparse multiple canonical correlation analysis (SmCCA), sparse supervised canonical correlation analysis (SsCCA), or sparse canonical correlation analysis (SCCA).*

Description

Integrate two omics data type (and a quantitative phenotype), and calculate the absolute canonical correlation weights for the omics features using SmCCA SsCCA, or SCCA. SmCCA and SsCCA take into account a phenotype/trait. SmCCA maximizes the total (weighted or unweighted) pairwise canonical correlation weights between two omics data types and the trait. It requires the trait to be quantitative. SsCCA prioritizes omics features based on the trait, and assigns non-zero canonical weights to features that are more correlated to the trait. SCCA does not use any trait information for computing the canonical correlation weights. All of these three methods are included in this function, along with an omic feature subsampling scheme and an optional random feature partition scheme.

Usage

```
getRobustPseudoWeights(X1, X2, Trait, Lambda1, Lambda2, s1, s2, NoTrait,
  FilterByTrait, Bipartite, SubsamplingNum, PartitionNum, CCcoef = NULL,
  trace = FALSE)
```

Arguments

`X1`	An $n \times p_1$ data matrix (e.g. mRNA) with p_1 features and n subjects.
`X2`	An $n \times p_2$ data matrix (e.g. miRNA) with p_2 features and n subjects.
`Trait`	An n$\times$1 trait data matrix for the same n subjects.
`Lambda1`	LASSO pentalty parameter for `X1`. `Lambda1` needs to be between 0 and 1.
`Lambda2`	LASSO pentalty parameter for `X2`. `Lambda2` needs to be between 0 and 1.
`s1`	Proportion of mRNA features to be included. `s1` needs to be between 0 and 1.
`s2`	Proportion of miRNA features to be included. `s2` needs to be between 0 and 1.
`NoTrait`	Logical. Whether trait information is provided.
`FilterByTrait`	Logical. Whether only the top 80 correlation to the trait will be assigned nonzero weights.

Bipartite	Logical. Whether to include random partition.
SubsamplingNum	Number of feature subsamples. Larger number leads to more accurate results, but at a higher cost. We recommend to subsample at least 1000 times.
PartitionNum	Number of random partitions for each set of subsamples. This is only used if Bipartite = FALSE.
CCcoef	Optional coefficients for the SmCCA pairwise canonical correlations. If CCcoef = NULL (default), then the objective function is the total sum of all pairwise canonical correlations. It can also be a coefficient vector that follows the column order of combn(K, 2).
trace	Logical. Whether to display CCA algorithm trace.

Details

To choose SmCCA, set NoTrait = FALSE, FilterByTrait = FALSE. To choose SsCCA, set NoTrait = FALSE, FilterByTrait = TRUE. To choose SCCA, set Trait = NULL, NoTrait = TRUE.

Value

A canonical correlation weight matrix with $p_1 + p_2$ rows. Each column is the canonical correlation weights based on subsampled X1 and X2 features. The column number equals to SubsamplineNum.

Examples

```
## For illustration, we only subsample 5 times.
set.seed(123)

# SmCCA
W1 <- getRobustPseudoWeights(X1, X2, Trait = Y, Lambda1 = 0.05,
  Lambda2 = 0.05, s1 = 0.7, s2 = 0.9, NoTrait = FALSE, FilterByTrait = FALSE,
  Bipartite = FALSE, SubsamplingNum = 100, PartitionNum = NULL,
  CCcoef = NULL, trace = FALSE)

# SsCCA
W2 <- getRobustPseudoWeights(X1, X2, Trait = Y, Lambda1 = .05, Lambda2 = 0.5,
  s1 = 0.7, s2 = 0.9, NoTrait = FALSE, FilterByTrait = TRUE,
  Bipartite = FALSE, SubsamplingNum = 100, PartitionNum = NULL,
  CCcoef = NULL, trace = FALSE)

# SCCA
W3 <- getRobustPseudoWeights(X1, X2, Trait = Y, Lambda1 = 0.05,
  Lambda2 = 0.05, s1 = 0.7, s2 = 0.9, NoTrait = TRUE, Bipartite = FALSE,
  SubsamplingNum = 100, PartitionNum = 5, CCcoef = NULL, trace = FALSE)
```

---

plotMultiOmicsNetwork *Plot multi-omics module networks.*

Description

Plot multi-omics modules based on similarity matrix derived from pseudo canonical weights and pairwise feature correlations.

Usage

```
plotMultiOmicsNetwork(Abar, CorrMatrix, multiOmicsModule, ModuleIdx, P1,
  EdgeCut, FeatureLabel = NULL, AddCorrSign = TRUE, SaveFile = NULL,
  ShowType1Label = TRUE, ShowType2Label = TRUE, PlotTitle = "",
  NetLayout = "lgl", ShowNodes = TRUE, VertexLabelCex = 1,
  VertexSize = 1)
```

Arguments

`Abar`	A $p \times p$ similary matrix for both omics data types based on pseudo canonical correlation weights. p is the number of total features for the two omics data types. All entries are non-negative.
`CorrMatrix`	A $p \times p$ correlation matrix that provides sign information for the network.
`multiOmicsModule`	A list of multi-omics modules.
`ModuleIdx`	Index for the module to be plotted. It can not exceed the length of `multiOmicsModule`.
`P1`	Total number of features for the first omics data type.
`EdgeCut`	A numerical value between 0 and 1, indicating an edge threshold for the network. Any features (network nodes) without any edge strength that passes the threshold are excluded from the figure.
`FeatureLabel`	A $1 \times p$ vector indicating feature names. If `FeatureLabel = NULL` (default), the feature names will be indices 1 through p.
`AddCorrSign`	Logical. Whether to add a positive or negative sign to each network edge based on pairwise feature correlations.
`SaveFile`	A pdf file name for the figure output. If `SaveFile = NULL` (default), the figure will not be saved.
`ShowType1Label`	Logical. Whether to label the network nodes for the first omics data type.
`ShowType2Label`	Logical. Whether to label the network nodes for the second omics data type.
`PlotTitle`	A title for the figure. Default is without any title.
`NetLayout`	Graphical layout for the network. Possible options are `circle` for circle layout, `sphere` for 3D sphere, `fr` for Fruchterman-Reinhold, and `lgl` for the LGL algorithm. Refer to igraph manual for more details on the layout options.
`ShowNodes`	Logical. Whether to show network nodes.
`VertexLabelCex`	Scaling factor for the vertex labels.
`VertexSize`	Size of the vertices.

Value

A multi-omics network figure.

Examples

```
set.seed(123)
w <- rnorm(5)
w <- w/sqrt(sum(w^2))
abar <- getAbar(w)
modules <- getMultiOmicsModules(abar, P1 = 2, CutHeight = 0.5)
x <- cbind(X1[ ,1:2], X2[ , 1:3])
corr <- cor(x)
```

```
plotMultiOmicsNetwork(abar, corr, modules, ModuleIdx = 1, P1 = 2,
  EdgeCut = 0)
```

X1 *A simulated mRNA expression dataset.*

Description

A matrix containing simulated mRNA expression levels for 358 subjects (rows) and 500 features (columns).

Usage

```
X1
```

Format

An object of class `matrix` with 358 rows and 500 columns.

X2 *A simulated miRNA expression dataset.*

Description

A matrix containing simulated miRNA expression levels for 358 subjects (rows) and 100 features (columns).

Usage

```
X2
```

Format

An object of class `matrix` with 358 rows and 100 columns.

Y *A simulated phenotype dataset.*

Description

A matrix containing simulated quantitative phenotype measures for 358 subjects (rows).

Usage

```
Y
```

Format

An object of class `matrix` with 358 rows and 1 columns.

Index