Package 'NetSAM'

May 16, 2024

Type Package

Title Network Seriation And Modularization

Version 1.44.0

Date 2023-03-23

Author Jing Wang <jing.wang@bcm.edu>

Maintainer Zhiao Shi <zhiao.shi@gmail.com>

Description The NetSAM (Network Seriation and Modularization) package takes an edge-list representation of a weighted or unweighted network as an input, performs network seriation and modularization analysis, and generates as files that can be used as an input for the one-dimensional network visualization tool NetGestalt (http://www.netgestalt.org) or other network analysis. The NetSAM package can also generate correlation network (e.g. co-expression network) based on the input matrix data, perform seriation and modularization analysis for the correlation network and calculate the associations between the sample features and modules or identify the associated GO terms for the modules.

License LGPL

LazyLoad yes

Depends R (>= 3.0.0), seriation (>= 1.0-6), igraph (>= 2.0.0), tools (>= 3.0.0), WGCNA (>= 1.34.0), biomaRt (>= 2.18.0)

Imports methods, AnnotationDbi (>= 1.28.0), doParallel (>= 1.0.10), foreach (>= 1.4.0), survival (>= 2.37-7), GO.db (>= 2.10.0), R2HTML (>= 2.2.0), DBI (>= 0.5-1)

Suggests RUnit, BiocGenerics, org.Sc.sgd.db, org.Hs.eg.db, org.Mm.eg.db, org.Rn.eg.db, org.Dr.eg.db, org.Ce.eg.db, org.Cf.eg.db, org.Dm.eg.db, org.At.tair.db, rmarkdown, knitr, markdown

Collate zzz.R NetSAM.R NetAnalyzer.R MatNet.R MatSAM.R consensusNet.R mapToSymbol.R testFileFormat.R featureAssociation.R GOAssociation.R mergeDuplicate.R

NeedsCompilation no

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/NetSAM

git_branch RELEASE_3_19

git_last_commit fd6bcba

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-15

Contents

`NetSAM-package` *Network Seriation and Modularization*

Description

The NetSAM (Network Seriation and Modularization) package takes an edge-list representation of a weighted or unweighted network as an input, performs network seriation and modularization analysis, and generates as files that can be used as an input for the one-dimensional network visualization tool NetGestalt (http://www.netgestalt.org) or other network analysis. Meanwhile, the NetSAM package can also generate correlation network (e.g. co-expression network) based on the input matrix data and then perform seriation and modularization analysis for correlation network.

Details

Package:	NetSAM
Type:	Package
Version:	1.31.1
Date:	2021-05-15
License:	LGPL
LazyLoad:	yes

Author(s)

Jing Wang Maintainer: Zhiao Shi <zhiao.shi@gmail.com>

References

NetGestalt: integrating multidimensional omics data over biological networks. Nature Methods 10, 597-598 (2013).

See Also

NetSAM MatSAM

`consensusNet` *Construction of a consensus coexpression network*

Description

To increase robustness against errors in data, the consensusNet function uses a bootstrapping procedure to construct a coexpression network.

Usage

```
consensusNet(data, organism="hsapiens",bootstrapNum=100, naPer=0.5, meanPer=0.8,varPer=0.8,method="
```

Arguments

`data`	`data` should contain a file name with extension "cct" or "cbt" or a matrix or data.frame object in R. The first column and first row of the "cct" or "cbt" file should be the row and column names, respectively and other parts are the numeric values. The detail information of "cct" or "cbt" format can be found in the manual of NetGestalt (www.netgestalt.org). A matrix or data.frame object should have row and column names and only contain numeric or integer values.
`organism`	The organism of the input data. Currently, the package supports the following nine organisms: hsapiens, mmusculus, rnorvegicus, drerio, celegans, scerevisiae, cfamiliaris, dmelanogaster and athaliana. The default is "hsapiens".
`bootstrapNum`	Number of bootstrap data sets generated. Default is 100.
`naPer`	To remove ids with missing values in most of samples, the function calculates the percentage of missing values in all samples for each id and removes ids with over `naPer` missing values in all samples. The default `naPer` is 0.5.
`meanPer`	To remove ids with low values, the function calculates the mean of values for each id in all samples and remains top `meanPer` ids based on the mean. The default `meanPer` is 0.8.
`varPer`	Based on the remaining ids filtered by `meanPer`, the function can also remove less variable ids by calculating the standard deviation of values for each id in all samples and remaining top `varPer` ids based on the standard deviation. The default `varPer` is 0.8.

method	Method used for constructing correlation network with MatNet. Currently supports "rank", "value" and "rank_unsig". Default is "rank_unsig".
value	The corresponding value set for method. Default is 0.003.
pth	p-value threshold for including an edge. Default is 1.0e-6.
nMatNet	The number of concurrent running MatNet processes, default is 2.
nThreads	consensusNet function supports parallel computing based on multiple cores. The default is 4.

Author(s)

Jing Wang

Examples

```
 inputMatDir <- system.file("extdata","exampleExpressionData.cct",package="NetSAM")
 data <- read.table(inputMatDir, header=TRUE, row.names=1, stringsAsFactors=FALSE)
net <- consensusNet(data, organism="hsapiens",bootstrapNum=10, naPer=0.5, meanPer=0.8,varPer=0.8,method="rank_u
```

---

featureAssociation *Calculate the associations between modules and sample features*

Description

The featureAssociation function will calculate the associations between the sample features in the input annotation data and the modules identified by NetSAM or MatSAM function.

Usage

```
featureAssociation(inputMat, sampleAnn, NetSAMOutput, outputHtmlFile, CONMethod="spearman", CATMethod
```

Arguments

inputMat	inputMat should contain a file name with extension "cct" or "cbt" or a matrix or data.frame object in R. The first column and first row of the "cct" or "cbt" file should be the row and column names, respectively and other parts are the numeric values. The detail information of "cct" or "cbt" format can be found in the manual of NetGestalt (www.netgestalt.org). A matrix or data.frame object should have row and column names and only contain numeric or integer values.
sampleAnn	sampleAnn should be a directory containing a file name with "tsi" extension or a data.frame object in R. The detail information of "tsi" format can be found in the manual of NetGestalt (www.netgestalt.org). The first row of the sample annotation data is the feature names. The second row is the feature types. The function supports four types: BIN (binary feature, such as male and female), CAT (category feature, such as stage i, stage ii and stage iii), CON (continuous feature, such as age) and SUR (survival data, such as overall survival). The third row is the feature categories. If there is no category information for the features, the sample information will start from the third row . The first column is the sample names.

NetSAMOutput	The list object outputted from NetSAM or MatSAM.
outputHtmlFile	The output directory of the HTML file.
CONMethod	The method to calculate the associations between modules and continuous features. The function provides two methods: "spearman" and "pearson" and the default is "spearman".
CATMethod	The method to calculate the associations between modules and category features. The function provides two methods: "anova" and "kruskal" and the default is "kruskal".
BINMethod	The method to calculate the associations between modules and binary features. The function provides two methods: "test" and "rankest" and the default is "rankest".
fdrmethod	The FDR method for identifying the significantly associated GO terms. The default is "BH".
pth	The threshold of the p values to identify the significant associations.
collapse_mode	The method to collapse duplicate ids. "mean", "median", "maxSD", "maxIQR", "max" and "min" represent the mean, median, max standard deviation, max interquartile range, maximum and minimum of values for ids in each sample. The default is "maxSD".

Value

The function will output a data.frame object and a HTML file to show the significant associations.

Author(s)

Jing Wang

See Also

MatSAM NetSAM

Examples

```
inputMatDir <- system.file("extdata","exampleExpressionData.cct",package="NetSAM")
sampleAnnDir <- system.file("extdata","sampleAnnotation.tsi",package="NetSAM")
data(NetSAMOutput_Example)
outputHtmlFile <- paste(getwd(),"/featureAsso_HTML",sep="")
featureAsso <- featureAssociation(inputMat= inputMatDir, sampleAnn=sampleAnnDir, NetSAMOutput=netsam_output, out
```

`GOAssociation` *Identify the associated GO terms for each module*

Description

The GOAssociation function will identify the associated GO terms for each module from the NetSAM or MatSAM function.

Usage

```
GOAssociation(NetSAMOutput, outputHtmlFile, organism, outputType, fdrmethod, fdrth, topNum)
```

Arguments

`NetSAMOutput`	The list object outputted from the NetSAM or MatSAM function.
`outputHtmlFile`	The output directory of the HTML file.
`organism`	The organism of the input data matrix that has been used to identify the modules. Currently, the package supports the following nine organisms: hsapiens, mmusculus, rnorvegicus, drerio, celegans, scerevisiae, cfamiliaris, dmelanogaster and athaliana. The default is "hsapiens".
`outputType`	The function supports two types of output results. 1. "significant" represents all associated GO terms should be significant under a certain FDR threshold; 2. "top" represents the function first sorts all GO terms based on their hypergenometric test p values and then selects top GO terms as the associated terms. The default is "significant".
`fdrmethod`	The FDR method for identifying the significantly associated GO terms. The default is "BH".
`fdrth`	The FDR threshold.
`topNum`	The number of the selected top GO terms.

Value

The function will output a data.frame object and a HTML file to show the associated GO terms for each module.

Author(s)

Jing Wang

See Also

`MatSAM` `NetSAM`

Examples

```
data(NetSAMOutput_Example)
outputHtmlFile <- paste(getwd(),"/GOAsso_HTML",sep="")
GOAsso <- GOAssociation(NetSAMOutput=netsam_output, outputHtmlFile=outputHtmlFile, organism="hsapiens", fdrmetho
```

`mapToSymbol` *Map other ids to gene symbols*

Description

The mapToSymbol function can transform other ids from a gene list, network, matrix, sbt file or sct file to gene symbols.

Usage

```
mapToSymbol(inputData, organism="hsapiens", inputType="genelist", idType="auto", edgeType="unweighted
```

Arguments

`inputData`	mapToSymbol function supports five different types of data: "genelist", "network", "matrix", "sbt" and "sct". For "genelist" type, `inputData` should be a vector containing gene ids. For "network" type, `inputData` can be the directory of the input network file including the file name with "net" extension. If `edgeType` is "unweighted", each row represents an edge with two node names separated by a tab or space. If `edgeType` is "weighted", each row represents an edge with two node names and edge weight separated by a tab or space. `inputNetwork` can also be a data object in R (data object must be igraph, graphNEL, matrix or data.frame class). For "matrix" type, `inputData` should be a directory containing a file name with extension "cct" or "cbt" or a matrix or data.frame object in R. The first column and first row of the "cct" or "cbt" file should be the row and column names, respectively and other parts are the numeric values. The detail information of "cct" or "cbt" format can be found in the manual of NetGestalt (www.netgestalt.org). A matrix or data.frame object should have row and column names and only contain numeric or integer values. For "sbt" type, `inputData` should be a directory containing a file name with extension "sbt" . The first column of a "sbt" file is the track names, the second one is the descriptions and others are the ids contained in the track. A "sbt" file can contain multiple tracks. The detail information of "sbt" format can be found in the manual of NetGestalt (www.netgestalt.org). For "sct" type, `inputData` should be a directory containing a file name with extension "sct" . The first column of a "sct" file is id names, the first row is the column names and others are the numeric or integer values. The detail information of "sct" format can be found in the manual of NetGestalt (www.netgestalt.org).
`organism`	The organism of the input data. Currently, the package supports the following nine organisms: hsapiens, mmusculus, rnorvegicus, drerio, celegans, scerevisiae, cfamiliaris, dmelanogaster and athaliana. The default is "hsapiens".

inputType	The type of the input data. see detail information in inputData.
idType	The id type of the ids in the input data. MatSAM will use BiomaRt package to transform the input ids to gene symbols based on idType. The users can also set idType as "auto" that means MatSAM will automatically search the id type of the input data. However, this may take 10 minutes based on the users' internet speed. The default is "auto".
edgeType	The type of the input network: "weighted" or "unweighted".
collapse_mode	The method to collapse duplicate ids. "mean", "median", "maxSD", "maxIQR", "max" and "min" represent the mean, median, max standard deviation, max interquartile range, maximum and minimum of values for ids in each sample. The default is "maxSD". For SCT file, we suggest to use "max" or "min" to collapse duplicate ids in the statistic data.
is_outputFile	If is_outputFile is TRUE, the function will output the transformed data to a file. The default is FALSE.
outputFileName	The output file name.
verbose	Report the extra information on progress. The default is TRUE.

Author(s)

Jing Wang

Examples

```
###transform ids from a gene list to gene symbols###
geneListDir <- system.file("extdata","exampleGeneList.txt",package="NetSAM")
geneList <- read.table(geneListDir,header=FALSE,sep="\t",stringsAsFactors=FALSE)
geneList <- as.vector(as.matrix(geneList))
geneList_symbol <- mapToSymbol(inputData=geneList, organism="hsapiens", inputType="genelist",idType="affy_hg_u13

###transform ids in the input network to gene symbols###
inputNetwork <- system.file("extdata","exampleNetwork_nonsymbol.net",package="NetSAM")
network_symbol <- mapToSymbol(inputData=inputNetwork,organism="hsapiens",inputType="network",idType="entrezgene

###transform ids in the input matrix to gene symbols###
inputMatDir <- system.file("extdata","exampleExpressionData_nonsymbol.cct",package="NetSAM")
matrix_symbol <- mapToSymbol(inputData=inputMatDir,organism="hsapiens",inputType="matrix",idType="affy_hg_u133_

###transform ids in the sbt file to gene symbols###
inputSBTDir <- system.file("extdata","exampleSBT.sbt",package="NetSAM")
sbt_symbol <- mapToSymbol(inputData= inputSBTDir,organism="hsapiens",inputType="sbt",idType="affy_hg_u133_plus_

###transform ids in the sct file to gene symbols###
inputSCTDir <- system.file("extdata","exampleSCT.sct",package="NetSAM")
sct_symbol <- mapToSymbol(inputData= inputSCTDir,organism="hsapiens",inputType="sct",idType="affy_hg_u133_plus_
```

MatNet *Construction of correlation network from a matrix*

Description

The MatNet function can use one of three different methods to construct correlation network based on the input data matrix. The output correlation network can be used as an input of NetSAM function to identify hierarchical modules.

Usage

```
MatNet(inputMat, collapse_mode="maxSD", naPer=0.7, meanPer=0.8, varPer=0.8, corrType="spearman", matN
```

Arguments

inputMat	inputMat should contain a file name with extension "cct" or "cbt" or a matrix or data.frame object in R. The first column and first row of the "cct" or "cbt" file should be the row and column names, respectively and other parts are the numeric values. The detail information of "cct" or "cbt" format can be found in the manual of NetGestalt (www.netgestalt.org). A matrix or data.frame object should have row and column names and only contain numeric or integer values.
collapse_mode	If the input matrix data contains the duplicate ids, the function will collapse duplicate ids based on the collapse_mode. "mean", "median", "maxSD" and "maxIQR" represent the mean, median, max standard deviation or max interquartile range of id values in each sample. The default is "maxSD".
naPer	To remove ids with missing values in most of samples, the function calculates the percentage of missing values in all samples for each id and removes ids with over naPer missing values in all samples. The default naPer is 0.7.
meanPer	To remove ids with low values, the function calculates the mean of values for each id in all samples and remains top meanPer ids based on the mean. The default meanPer is 0.8.
varPer	Based on the remained ids filtered by meanPer, the function can also remove less variable ids by calculating the standard deviation of values for each id in all samples and remaining top varPer ids based on the standard deviation. The default varPer is 0.8.
corrType	The method to calculate correlation coefficient for each pair of ids. The function supports "spearman" (default) or "pearson" method.
matNetMethod	MatNet function supports three methods to construct correlation network: "value", "rank" and "directed". 1. "value" method: the correlation network only remains id pairs with correlations over cutoff threshold valueThr; 2. "rank" method: for each id A, the function first selects ids that significantly correlate with id A and then extracts a set of candidate neighbors (the number of ids is calculated based on rankBest) from the significant set that are most similar to id A. Then, for each id B in the candidate neighbors of id A, the function also extracts the same number of ids that are significant correlated and most similar to id B. If id

	A is also the candidate neighbors of id B, there will be an edge between id A and id B. Combining all edges can construct a correlation network; 3. "directed" method: the function will only remain the best significant id for each id as the edge.Combining all edges can construct a directed correlation network.
valueThr	Correlation cutoff threshold for "value" method. The default is 0.5.
rankBest	The percentage of ids that are most similar to one id for "rank" method. The default is 0.003 which means the "rank" method will select top 30 most similar ids for each id if the number of ids in the matrix is 10,000.
networkType	If networkType is "unsigned", the correlation of all pairs of ids will be changed to absolute values. The default is "signed".
netFDRMethod	p value adjustment methods for "rank" and "directed" methods. The default is "BH".
netFDRThr	fdr threshold for identifying significant pairs for "rank" and "directed" methods. The default is 0.05
idNumThr	If the matrix contains too many ids, it will take a long time and use a lot of memory to identify the modules. Thus, the function provides the option to set the threshold of number of ids for further analysis. After filtering by meanPer and varPer, if the number of ids is still larger than idNumThr, the function will select top idNumThr ids with the largest variance. The default is -1, which means there is no limitation for the matrix.
nThreads	MatNet function supports parallel computing based on multiple cores. The default is 3.

Note

For data with missing values, the function will take longer time to calculate correlation between each pair of ids than data without missing value.

Author(s)

Jing Wang

Examples

```
inputMatDir <- system.file("extdata","exampleExpressionData.cct",package="NetSAM")
matNetwork <- MatNet(inputMat=inputMatDir, collapse_mode="maxSD", naPer=0.7, meanPer=0.8, varPer=0.8, corrType="s
```

MatSAM *Correlation network construction, seriation and modularization from a matrix*

Description

The MatSAM function first uses MatNet function to identify the correlation network and then uses NetSAM function to identify the module and optimize the one-dimensional ordering of the nodes in each module.

Usage

```
MatSAM(inputMat, sampleAnn=NULL, outputFileName, outputFormat="msm", organism="hsapiens", map_to_symb
```

Arguments

inputMat	inputMat should contain a file name with extension "cct" or "cbt" or a matrix or data.frame object in R. The first column and first row of the "cct" or "cbt" file should be the row and column names, respectively and other parts are the numeric values. The detail information of "cct" or "cbt" format can be found in the manual of NetGestalt (www.netgestalt.org). A matrix or data.frame object should have row and column names and only contain numeric or integer values.
sampleAnn	sampleAnn should contain a file name with "tsi" extension (the detail information of "tsi" format can be found in the manual of NetGestalt (www.netgestalt.org)) or a data.frame object in R. If the data does not have sample annotation, this argument can be ignored. The first row of the data is the name of sample features. The second row is the type of each feature. The third row is the category of each feature. If there is no category information for the features, the sample information will start from the third row . The first column is the sample name.
outputFileName	Output file name. The file name extension is "msm" which can be uploaded to the NetGestalt directly.
outputFormat	The format of the output file. "msm" format can be used as an input in NetGestalt; "gmt" format can be used to do other network analysis (e.g. as an input in GSEA (Gene Set Enrichment Analysis) to do module enrichment analysis); "multiple" represents the MatSAM function will output five files: ruler file containing gene order information, hmi file containing module information, net file containing correlation network information, cct file containing the filtered data matrix, and tsi file containing the sample annotation with standardized format; and "none" represents the function will not output any file.
organism	The organism of the input data. Currently, the package supports the following nine organisms: hsapiens, mmusculus, rnorvegicus, drerio, celegans, scerevisiae, cfamiliaris, dmelanogaster and athaliana. The default is "hsapiens".
map_to_symbol	If map_to_symbol is TRUE, the function will first change the input ids to gene symbols and collapse multiple ids with the same gene symbol based on the collapse_mode method before identifying correlation network. The default is FALSE.
idType	The id type of the ids in the input matrix. MatSAM will use BiomaRt package to transform the input ids to gene symbols based on idType. The users can also set idType as "auto" that means MatSAM will automatically search the id type of the input data. However, this may take 10 minutes based on the users' internet speed. The default is "auto".
collapse_mode	The method to collapse duplicate ids. "mean", "median", "maxSD", "maxIQR", "max" and "min" represent the mean, median, max standard deviation, max interquartile range, maximum and minimum of values for ids in each sample. The default is "maxSD".
naPer	To remove ids with missing values in most of samples, the function calculates the percentage of missing values in all samples for each id and removes ids with over naPer missing values in all samples. The default naPer is 0.7.

`meanPer` To remove ids with low values, the function calculates the mean of values for a id in all samples and remains top `meanPer` ids based on the mean. The default `meanPer` is 0.8.

`varPer` Based on the remained ids filtered by `meanPer`, the function can also remove less variable ids by calculating the standard deviation of values for a id in all samples and remaining top `varPer` ids based on the standard deviation. The default `varPer` is 0.8.

`corrType` A character string indicating which correlation coefficient is to be computed for each pair of ids. The function supports "spearman" (default) or "pearson" method.

`matNetMethod` MatNet function supports three methods to construct correlation network: "value", "rank" and "directed". 1. "value" method: the correlation network only remains id pairs with correlations over cutoff threshold `valueThr`; 2. "rank" method: for each id A, the function first selects ids that significantly correlate with id A and then extracts a set of ids (the number of ids is calculated based on `rankBest`) that are most similar to id A from the significant set. Then, for each id B in the set, the function also extracts the same number of ids that are significant correlated and most similar to id B. If id A is in the set of id B, the edge between id A and id B will be remained. Combining all remained edges can construct a correlation network; 3. "directed" method: the function will only remain the best significant id for each id as the edge.Combining all edges can construct a directed correlation network.

`valueThr` Correlation cutoff threshold for "value" method. The default is 0.5.

`rankBest` The percentage of ids that are most similar to one id for "rank" method. The default is 0.003 which means the "rank" method will select top 30 most similar ids for each id if the number of ids in the matrix is 10,000.

`networkType` If `networkType` is "unsigned", the correlation of all pairs of ids will be changed to absolute values. The default is "signed".

`netFDRMethod` p value adjustment methods for "rank" and "directed" methods. The default is "BH".

`netFDRThr` fdr threshold for identifying significant pairs for "rank" and "directed" methods. The default is 0.05

`minModule` The minimum percentage of nodes in a module. The minimum size of a module is calculated by multiplying `minModule` by the number of nodes in the whole network. If the size of a module identified by the function is less than the minimum size, the module will not be further partitioned into sub-modules. The default is 0.003 which means the minimum module size is 30 if there are 10,000 nodes in the whole network. If the minimum module size is less than 5, the minimum module size will be set as 5. The `minModule` should be less than 0.2.

`stepIte` Because NetSAM uses random walk distance-based hierarchical clustering to reveal the hierarchical organization of an input network, it requires a specified length of the random walks. If `stepIte` is TRUE, the function will test a range of lengths ranging from 2 to `maxStep` to get the optimal length. Otherwise, the function will directly use `maxStep` as the length of the random walks. The default `maxStep` is 4. Because optimizing the length of the random walks will

take a long time, if the network is too big (e.g. the number of edges is over 200,000), we suggest to set `stepIte` as FALSE.

`maxStep` The length or max length of the random walks.

`moduleSigMethod`
To test whether a network under consideration has a non-random internal modular organization, the function provides three options: "cutoff", "zscore" and "permutation". "cutoff" means if the modularity score of the network is above a specified cutoff value, the network will be considered to have internal organization and will be further partitioned. For "zscore" and "permutation", the function will first generate a set of random modularity scores. Based on a unweighted network, the function uses the edge switching method to generate a given number of random networks with the same number of nodes and an identical degree sequence and calculates the modularity scores for these random networks. Based on a weighted network, the function shuffles the weights of all edges and calculate the modularity scores for network with random weights. Then, "zscore" method will transform the real modularity score to a z score based on the random modularity scores and then transform the z score to a p value assuming a standard normal distribution. The "permutation" method will compare the real modularity score with the random ones to calculate a p value. Finally, under a specified significance level, the function determines whether the network can be further partitioned. The default is "cutoff".

`modularityThr` Threshold of modularity score for the "cutoff" method. The default is 0.2

`ZRanNum` The number of random networks that will be generated for the "zscore" calculation. The default is 10.

`PerRanNum` The number of random networks that will be generated for the "permutation" p value calculation. The default is 100.

`ranSig` The significance level for determining whether a network has non-random internal modular organization for the "zscore" or "permutation" methods. The default is 0.05.

`idNumThr` If the matrix contains too many ids, it will take a long time and use a lot of memory to identify the modules. Thus, the function provides the option to set the threshold of number of ids for further analysis. After filtering by meanPer and varPer, if the number of ids is still larger than `idNumThr`, the function will select top `idNumThr` ids with the largest variance. The default is -1, which means there is no limitation for the matrix.

`nThreads` MatSAM function supports parallel computing based on multiple cores. The default is 3.

Value

Including a "msm" file, the function will output a list object containing module information, gene order information, correlation network and filtered matrix based on the ids in the network. The function will also output two HTML files that contain the significant associations between sample features and modules and associated GO terms for the modules.

Note

After identifying the modules, the MatSAM function will identify the associations between sample features and modules using the featureAssociation function or the associated GO terms for the modules using the GOAssociation function. For the featureAssociation function, MatSAM only uses the default parameters. For the GOAssociation function, MatSAM sets "outputType" as "top" and "topNum" as 1. The users can use the list object returned by MatSAM as the input of the function featureAssociation and GOAssociation to perform some further analysis based on the different parameters.

Author(s)

Jing Wang

See Also

MatNet NetSAM

Examples

```
inputMatDir <- system.file("extdata","exampleExpressionData.cct",package="NetSAM")
cat(inputMatDir)
sampleAnnDir <- system.file("extdata","sampleAnnotation.tsi",package="NetSAM")
cat(sampleAnnDir)
outputFileName <- paste(getwd(),"/MatSAM",sep="")
matModule <- MatSAM(inputMat=inputMatDir, sampleAnn=sampleAnnDir, outputFileName=outputFileName, outputFormat="m
```

---

mergeDuplicate *Merge the duplicate Ids in the matrix data*

Description

The mergeDuplicate function will merge the duplicate Ids in the matrix data and return the matrix with unique Ids. This function can also used to merge the duplicate mapped Ids when transforming the Ids of data matrix to other Ids.

Usage

```
mergeDuplicate(id, data, collapse_mode="maxSD")
```

Arguments

id	Duplicate Ids that should be a vector object in R.
data	the corresponding data matrix that has the same number of rows with id and should be a matrix or data.frame object in R.
collapse_mode	The method to collapse duplicate ids. "mean", "median", "maxSD", "maxIQR", "max" and "min" represent the mean, median, max standard deviation, max interquartile range, maximum and minimum of values for ids in each sample. The default is "maxSD".

Value

The function will return the data matrix with unique Ids.

Author(s)

Jing Wang

Examples

```
inputMatDir <- system.file("extdata","exampleExpressionData_nonsymbol.cct",package="NetSAM")
inputMat <- read.table(inputMatDir,header=TRUE,sep="\t",stringsAsFactors=FALSE,check.names=FALSE)
mergedData <- mergeDuplicate(id=inputMat[,1],data=inputMat[,2:ncol(inputMat)],collapse_mode="maxSD")
```

NetAnalyzer *Network analyzer*

Description

The NetAnalyzer function can calculate the degree, clustering coefficient, betweeness and closeness centrality for each node and the shortest path distance for each pair of nodes. The NetAnalyzer function can also plot the distributions for these measurements.

Usage

```
NetAnalyzer(inputNetwork, outputFileName,edgeType="unweighted")
```

Arguments

`inputNetwork`	The network under analysis. `inputNetwork` can be the directory of the input network file including the file name with "net" extension. If `edgeType` is "unweighted", each row represents an edge with two node names separated by a tab or space. If `edgeType` is "weighted", each row represents an edge with two node names and edge weight separated by a tab or space. `inputNetwork` can also be a data object in R (data object must be igraph, graphNEL, matrix or data.frame class).
`edgeType`	The type of the input network: "weighted" or "unweighted".
`outputFileName`	The name of the output file.

Value

The function will output two "txt" files and five "pdf" files. Two "txt" files contain degree, clustering coefficient, betweeness and closeness centrality for each node and the shortest path distance for each pair of nodes. Five "pdf" files are the distributions of these measurements.

Author(s)

Jing Wang

Examples

```
inputNetworkDir <- system.file("extdata","exampleNetwork.net",package="NetSAM")
outputFileName <- paste(getwd(),"/NetSAM",sep="")
NetAnalyzer(inputNetworkDir,outputFileName,"unweighted")
```

NetSAM *Network Seriation and Modularization*

Description

The NetSAM function uses random walk distance-based hierarchical clustering to identify the hierarchical modules of a weighted or unweighted network and then uses the optimal leaf ordering (OLO) method to optimize the one-dimensional ordering of the genes in each module by minimizing the sum of the pair-wise random walk distance of adjacent genes in the ordering.

Usage

```
NetSAM(inputNetwork, outputFileName, outputFormat="nsm", edgeType="unweighted", map_to_genesymbol=FAL
```

Arguments

inputNetwork	The network under analysis. inputNetwork can be the directory of the input network file including the file name with "net" extension. If edgeType is "unweighted", each row represents an edge with two node names separated by a tab or space. If edgeType is "weighted", each row represents an edge with two node names and edge weight separated by a tab or space. inputNetwork can also be a data object in R (data object must be igraph, graphNEL, matrix or data.frame class).
edgeType	The type of the input network: "weighted" or "unweighted".
outputFileName	The name of the output file.
outputFormat	The format of the output file. "nsm" format can be used as an input in NetGestalt; "gmt" format can be used to do other network analysis (e.g. as an input in GSEA (Gene Set Enrichment Analysis) to do module enrichment analysis); "multiple" represents the NetSAM function will output three files: ruler file containing gene order information, hmi file containing module information and net file containing network information; and "none" represents the function will not output any file.
map_to_genesymbol	Because pathway enrichment analysis in NetGestalt is based on gene symbol, setting map_to_genesymbol as TRUE can transform other ids in the network into gene symbols and thus allow users to do functional analysis based on the identified modules. If the input network is not a biology network or users do not plan to do enrichment analysis in the NetGestalt, users can set map_to_genesymbol as FALSE. The default is FALSE.

`organism` The organism of the input network. Currently, the package supports the following nine organisms: hsapiens, mmusculus, rnorvegicus, drerio, celegans, scerevisiae, cfamiliaris, dmelanogaster and athaliana. The default is "hsapiens".

`idType` The id type of the ids in the input network. MatSAM will use BiomaRt package to transform the input ids to gene symbols based on `idType`. The users can also set `idType` as "auto" that means MatSAM will automatically search the id type of the input data. However, this may take 10 minutes based on the users' internet speed. The default is "auto".

`minModule` The minimum percentage of nodes in a module. The minimum size of a module is calculated by multiplying `minModule` by the number of nodes in the whole network. If the size of a module identified by the function is less than the minimum size, the module will not be further partitioned into sub-modules. The default is 0.003 which means the minimum module size is 30 if there are 10,000 nodes in the whole network. If the minimum module size is less than 5, the minimum module size will be set as 5. The `minModule` should be less than 0.2.

`stepIte` Because NetSAM uses random walk distance-based hierarchical clustering to reveal the hierarchical organization of an input network, it requires a specified length of the random walks. If `stepIte` is TRUE, the function will test a range of lengths ranging from 2 to `maxStep` to get the optimal length. Otherwise, the function will directly use `maxStep` as the length of the random walks. The default `maxStep` is 4. Because optimizing the length of the random walks will take a long time, if the network is too big (e.g. the number of edges is over 200,000), we suggest to set `stepIte` as FALSE.

`maxStep` The length or max length of the random walks.

`moduleSigMethod`

To test whether a network under consideration has a non-random internal modular organization, the function provides three options: "cutoff", "zscore" and "permutation". "cutoff" means if the modularity score of the network is above a specified cutoff value, the network will be considered to have internal organization and will be further partitioned. For "zscore" and "permutation", the function will first generate a set of random modularity scores. Based on a unweighted network, the function uses the edge switching method to generate a given number of random networks with the same number of nodes and an identical degree sequence and calculates the modularity scores for these random networks. Based on a weighted network, the function shuffles the weights of all edges and calculate the modularity scores for network with random weights. Then, "zscore" method will transform the real modularity score to a z score based on the random modularity scores and then transform the z score to a p value assuming a standard normal distribution. The "permutation" method will compare the real modularity score with the random ones to calculate a p value. Finally, under a specified significance level, the function determines whether the network can be further partitioned. The default is "cutoff".

`modularityThr` Threshold of modularity score for the "cutoff" method. The default is 0.2

`ZRanNum` The number of random networks that will be generated for the "zscore" calculation. The default is 10.

`PerRanNum` The number of random networks that will be generated for the "permutation" p value calculation. The default is 100.

ranSig	The significance level for determining whether a network has non-random internal modular organization for the "zscore" or "permutation" methods.
edgeThr	If the network is too big, it will take a long time to identify the modules. Thus, the function provides the option to set the threshold of number of edges and nodes as edgeThr and nodeThr. If the size of network is over the threshold, the function will stop and the users should change the parameters and re-run the function. We suggest to set the threshold for node as 12,000 and the threshold for edge as 300,000. The default is -1, which means there is no limitation for the input network.
nodeThr	see edgeThr.
nThreads	NetSAM function supports parallel computing based on multiple cores. The default is 3.

Value

If output format is "nsm", the function will output not only a "nsm" file but also a list object containing module information, gene order information and network information. If output format is "gmt", the function will output the "gmt" file and a matrix object containing the module and annotation information.

Note

Because the seriation step requires pair-wise distance between all nodes, NetSAM is memory consuming. We recommend to use the 64 bit version of R to run the NetSAM. For networks with less than 10,000 nodes, we recommend to use a computer with 8GB memory. For networks with more than 10,000 nodes, a computer with at least 16GB memory is recommended.

Author(s)

Jing Wang

Examples

```
inputNetworkDir <- system.file("extdata","exampleNetwork.net",package="NetSAM")
outputFileName <- paste(getwd(),"/NetSAM",sep="")
result <- NetSAM(inputNetwork=inputNetworkDir, outputFileName=outputFileName, outputFormat="nsm", edgeType="unwe
```

---

netsam_output *An example of the list object returned by NetSAM function*

Description

The list object contains at least three parts: gene order information, module information and network information. This object can be used as an input of the function featureAssociation or GOAssociation.

Usage

```
data(NetSAMOutput_Example)
```

Format

list

`testFileFormat` *Test whether the data matrix and the annotation have a correct format*

Description

The testFileFormat function will test the format of the input data matrix and annotation data and return the standardized data matrix and sample annotation data.

Usage

```
testFileFormat(inputMat=NULL,sampleAnn=NULL,collapse_mode="maxSD")
```

Arguments

`inputMat`	`inputMat` should contain a file name with extension "cct" or "cbt" or a matrix or data.frame object in R. The first column and first row of the "cct" or "cbt" file should be the row and column names, respectively and other parts are the numeric values. The detail information of "cct" or "cbt" format can be found in the manual of NetGestalt (www.netgestalt.org). A matrix or data.frame object should have row and column names and only contain numeric or integer values.
`sampleAnn`	`sampleAnn` is a file name or a data.frame object in R.
`collapse_mode`	The method to collapse duplicate ids. "mean", "median", "maxSD", "maxIQR", "max" and "min" represent the mean, median, max standard deviation, max interquartile range, maximum and minimum of values for ids in each sample. The default is "maxSD".

Value

If there is no format error, the function will return the standardized data matrix and sample annotation data. Otherwise, the function will output the detailed position of the errors.

Note

If the users set inputMat as "", the testFileFormat function only test format of sample annotation data. If the users set sampleAnn as "", the testFileFormat function only test format of data matrix.

Author(s)

Jing Wang

Examples

```
inputMatDir <- system.file("extdata","exampleExpressionData.cct",package="NetSAM")
sampleAnnDir <- system.file("extdata","sampleAnnotation.tsi",package="NetSAM")

formatedData <- testFileFormat(inputMat=inputMatDir,sampleAnn=sampleAnnDir,collapse_mode="maxSD")
```

Index