Package ‘petal’

February 16, 2016

Type Package

Title Construction of small-world scale-free networks and extraction of vicinity networks for vertcies of interest

Version 0.1.5

Author Juli Petereit

Maintainer Juli Petereit <julipetereit@gmail.com>

Description This package is primlary written for the life scientist, no prior network theory is requried to construct small-world sclae-free network models and extract groups of interest from it. It is primarly for expression data, but any data can be loaded.

License GPL-3

LazyData TRUE

RoxygenNote 5.0.1

R topics documented:

`count` *counting function*

Description

This function counts the appearance of a particular variable within each row of a data matrix

Usage

```
count(variable, M)
```

Arguments

`variable`	character or numeric, variable that should be counted
`M`	data matrix in which to find the variable in

Value

numeric vector indicating the number of appearances of the variable of interest for each row of the provided data matrix

`createSWSFnetFromFile` *Builds a small-world scale-free network*

Description

This function creates a number of network models to determine the best threshold producing a small-world scale-free network. All network models are accessible as .RData files. The file winningThresh.RData includes the possible thresholds to construct a sw-sf network with the first entry representing the 'best' threshold for the model construction.

Usage

```
createSWSFnetFromFile(expMFile = NULL, metric = c("SP", "PE", "KE", "EU",
  "MA", "CA"), thresholds = NULL, MMoExists = FALSE)
```

Arguments

`expMFile` data file name (.txt) in tab-separated format, first column must be vertex ID (genes), first row are condition identifiers

`metric` charcter defining association between each vertex ID, correlation options: "SP" - Spearman Correlation, "PE" - Pearson Correlation, and "KE" - Kendall as define in corstats; distances: "EU" - Euclidean, "MA" - Manhattan, and "CA" - canaberra as defined in diststats

`thresholds` numeric vector representing a series of thresholds that the user can choose; this variable can only be assigned with the metric is one of the three correlation metrics

`MMoExists` wheter the measure matrix exists (from a previous run), set to FALSE, if set to TRUE it will load the measure matrix (file must be in the same directory as the run)

Value

None, output files are generated

`create_MMo` *Calculating all pairwise assocations based on indicated metric*

Description

This function takes a data matrix and calculates all pairwise associations of the rows within the matrix. Association measures are sorted from best to worst, the ordered measure matrix is returned and written to a .RData file

Usage

```
create_MMo(expMatrix, metricChoice)
```

Arguments

`expMatrix` numeric data matrix

`metricChoice` choice of metric, options include: Pearson Correlation "PE", Spearman Correlation "SP", Kendall Correlation "KE", Euclidean Distance "EU", Manhattan Distance "MA", and Canberra Distance "CA"

Value

ordered measure matrix, sorted from best to worst association measure

`create_smMMs` *Internal function of makeAdjMatrices*

Description

Splits the large ordered measure matrix into smaller matrices according to the indicated thresholds

Usage

```
create_smMMs(orderedMM, thresholds, metric)
```

Arguments

`orderedMM`	measure matrix including all pairwise comparisions between vertex IDs based on a particular metric defining assosiction; association measured are sorted from best to worst
`thresholds`	vector of numeric values indicating the thresholds on which to build the network models
`metric`	charater (i.e. "SP", "PE", "EU", etc) indicating the metric used to define association between vertex IDs

Value

None

`dataAssessToFile` *Creates an output file providing general information about the data file and its analysis*

Description

This function creates an output file including multiple infromation about the data at hand as well as steps taken in the analysis. This is an internal function of createSWSFnet

Usage

```
dataAssessToFile(expMFile, metricUsed, CalculateThres, thresholds,
  winningThresh)
```

Arguments

`expMFile`	filename containing data matrix
`metricUsed`	metric being used to define association
`CalculateThres`	whether user specified thresholds or not, boolean: TRUE or FALSE
`thresholds`	list of thresholds considered for the network model generation
`winningThresh`	'best' threshold that generates a scale-free, small-world network

Value

None; output file is generated

`dataToVNs` *Builds a small-world scale-free network and extracts VNs based on genes of interest*

Description

This function calls createSWSFnetFromFile() and downstreamAnalysis(). Its a one strep function to create a scale-free small-world network, determines the 'best' network model, from which then vicinity networks (VNs) are extracted based on vertex IDs the user can upload. For more specifics refer to the two functions createSWSFnetFromFile() and downstreamAnalysis()

Usage

```
dataToVNs(dataFile, GoIFile = NULL, annoFile = NULL, metric = NULL,
  thresholds = NULL)
```

Arguments

`dataFile`	data file name (.txt) in tab-separated format, first column must be vertex ID (genes), first row are condition identifiers
`GoIFile`	a .txt file including vertex IDs of interest, IDs must be tab-separated
`annoFile`	wheter there is an annoation file for the vertex IDs
`metric`	charcter defining association between each vertex ID, correlation options: "SP" - Spearman Correlation, "PE" - Pearson Correlation, and "KE" - Kendall as define in corstats; distances: "EU" - Euclidean, "MA" - Manhattan, and "CA" - canaberra as defined in diststats
`thresholds`	numeric vector representing a series of thresholds that the user can choose; this variable can only be assigned with the metric is one of the three correlation metrics

Value

None, output files are generated

`densityM` *Internal function of downstreamAnalysis, calculates the density of a network*

Description

The density of the passed network is calculated (number of edges in the graph versus number of all possible edges).

Usage

```
densityM(FriendsM, FileOut = NULL, outReturn = TRUE)
```

Arguments

`FriendsM`	subnetwork in matrix format; diagonal must be set to 1
`FileOut`	name of file to write to if desired
`outReturn`	whether to return the calculated density, boolean variable TRUE(default) or FALSE

Value

decimal between 0 and 1; zero indicating there are no edges in the network, and a network with density 1 is a clique (completely connected network).

`densityMatrix` *Calculates the density of a provided network*

Description

The density of the passed network is calculated (number of edges in the graph versus number of all possible edges).

Usage

```
densityMatrix(FriendsM, moreInfo = FALSE)
```

Arguments

`FriendsM`	a network matrix with 0 and 1 indicating none- and existing links, respectively; diagonal must be set to 1
`moreInfo`	whether to return more information, boolean variable TRUE or FALSE(default)

Value

decimal between 0 and 1; zero indicating there are no edges in the network, and a network with density 1 is a clique (completely connected network).

densityNames *Calculates the density of a subnetwork based on vertex IDs*

Description

Calculates the density (number of edges in the graph versus number of all possible edges) of a subnetwork based on indicated rowname identifiers.

Usage

```
densityNames(Friends, adjMatrix, moreInfo = FALSE)
```

Arguments

`Friends`	a list of names, which correspond to the associated network matrix's rownames
`adjMatrix`	a network matrix with 0 and 1 indicating none- and existing links, respectively; diagonal must be set to 1
`moreInfo`	whether to return more information, boolean variable TRUE or FALSE(default)

Value

decimal between 0 and 1; zero indicating there are no edges in the network, and a network with density 1 is a clique (completely connected network).

downstreamAnalysis *Automated subnetwork extraction*

Description

This function finds all maximal cliques of a subgroup of vertex ids, then the vicinity network (1-neighbor network) is identified for each maximal clique.

Usage

```
downstreamAnalysis(winnerT, metric, GoIFile, outFile, expMatrixFile,
  annoTFile = NULL, adjMatrix = NULL)
```

Arguments

`winnerT`	threshold used to construct the network model
`metric`	metrix used to construct the network model
`GoIFile`	gene of interest (GoI) file, a txt file inlucuding vertex ids which are of particular interest
`outFile`	indicate a filename to which to write the information to

`expMatrixFile` filename of original datafile (expression matrix)

`annoTFile` annotation file if there is one, first column must correspond to vertex IDs, first row must be column headers

`adjMatrix` custom adjacency matrix can be loaded if petal network construction was not used

Value

None, but multiple files are created

`findLargestCliques` *Finds all largest cliques*

Description

This function calls the largest.cliques function from igraph version 0.7

Usage

```
findLargestCliques(adjMatrix)
```

Arguments

`adjMatrix` a network matrix with 0 and 1 indicating none- and existing links, respectively

Value

a numeric matrix; each column is a largest clique containing the vertex ids

---

`findMaximalCliques` *Finds all maximal cliques This function calls the maximal.cliques function from igraph version 0.7*

Description

Finds all maximal cliques

This function calls the maximal.cliques function from igraph version 0.7

Usage

```
findMaximalCliques(adjMatrix)
```

Arguments

`adjMatrix` a network matrix with 0 and 1 indicating none- and existing links, respectively

Value

a list of containing numeric vetors of vertex ids; each list element is a clique

---

findWinningThresh *Functions calculates best and alternative thresholds for a scale-free small-world network*

Description

This function takes the network statistics file created by makeThresholdTable() and weights the parameters against each other to determine the best network that is scale-free and small-world and follows other important properties such as the consideration of the biggest component of a network model.

Usage

```
findWinningThresh(threshFile, minR2 = 0.8, maxSlope = -1, minUsed = 40,
  minBigComp = 95)
```

Arguments

threshFile	filename of file containing network statistics
minR2	maximum acceptable slope of the linear regression of the log transformed degree distribution; default set to 0.8
maxSlope	maximum acceptable slope of the linear regression of the log transformed degree distribution; default set to -1; slope should be -3, -1
minUsed	minimum acceptable percentage of the original dataset used to build the network model; default is 40
minBigComp	minimum acceptable size of the biggest component in percentage; default set to 95

Value

possibleThresh numeric vector presenting possible thresholds which construct scale-free small-world network models, the vector is sorted from best to worse

generateThresholds *Generates a series of thresholds based on the distribution of the numeric vector*

Description

This function generates a series of thresholds based on the passed numeric vector. These thresholds are used to generate a number of visible network models.

Usage

```
generateThresholds(orderedV, adjMNum, decreasingOrder = TRUE)
```

Arguments

`orderedV`	sorted numeric vector, for correlation and mutual infromation the vector should be in decending order, for distance the vector must be in increasing order (i.e. first entry should represent best association value)
`adjMNum`	number indicating how many thresholds (i.e. network models) should be calculated
`decreasingOrder`	wheter the orderedV is in decending order (TRUE) or increasing order (FALSE)

Value

numeric vector

getVertexStats *Calculating all vertices' cluster coefficient and degree*

Description

Based on a network matrix, for each vertex of the network/row of the matrix, its cluster coeffient and connectivity is calculated and return in table format

Usage

```
getVertexStats(adjMatrix, totxtFile = FALSE)
```

Arguments

`adjMatrix`	binary adjacency matrix, diagonal is assumed to be 1
`totxtFile`	whether to right the table to a separate text file, boolean variable TRUE or FALSE(default)

Value

matrix with colnames "Index", "Gene","LocalCC","k"

graphHistQQ *Creates a tiff image of data histogram and the corresponding Q-Q plot*

Description

Creates a tiff files including the histogram of the data as well as the corresponding q-qplot see qqnormstats for more information based on a numeric data matrix

Usage

```
graphHistQQ(dataMatrix, fileName = "Data_Hist-QQ")
```

Arguments

`dataMatrix`	numeric data matrix
`fileName`	prefix of the desired output file, defaul is "Data_Hist-QQ"

Value

None; output file is generated

graphHistQQFromFile *Creates a tiff image of data histogram and the corresponding Q-Q plot from a file*

Description

Creates a tiff files including the histogram of the data as well as the corresponding q-qplot see qqnormstats for more information. Input file is read in and the data matrix is stored as expM.RData.

Usage

```
graphHistQQFromFile(inputFile, fileName = "Data_Hist-QQ")
```

Arguments

`inputFile`	file name of a data matrix; file must be tab delimited, first line must be table header, first column must be identifiers, inner part of the matrix must be numeric
`fileName`	prefix of the desired output file, defaul is "Data_Hist-QQ"

Value

None; output file is generated

makeAdjMatrices *Function creating adjacency matrices for each threshold*

Description

For each threshold an binary adjacency matrix is constructed.

Usage

```
makeAdjMatrices(orderedMMo, thresholds, metric)
```

Arguments

- `orderedMMo` measure matrix including all pairwise comparisions between vertex IDs based on a particular metric defining assosiction; association measured are sorted from best to worst
- `thresholds` thresholds defining similarity, list of numeric values
- `metric` charater (i.e. "SP", "PE", "EU", etc) indicating the metric used to define association between vertex IDs

Value

None, output files are generated

makeCurrentSymAdjMatrix *Internal function of makeAdjMatrices*

Description

Adjacency matricies are produced and checked for correctness

Usage

```
makeCurrentSymAdjMatrix(shm, currentThreshIndex, thresholds, edgeNum)
```

Arguments

- `shm` metric prefix
- `currentThreshIndex` threshold index that is currently considered
- `thresholds` vector of numeric values
- `edgeNum` numer of edges for the considered threshold

Value

edgeNum, the number of edges for the network model under considered threshold

makeCytoFile *Creates an network file for Cytoscape*

Description

Creates a txt file which can be loaded as is into Cytoscape to plot the network based on the indicated threshold and metric.

Usage

```
makeCytoFile(threshold, metric, orderedMM)
```

Arguments

`threshold` numeric value indicating the threshold building a network model

`metric` charater (i.e. "SP", "PE", "EU", etc) indicating the metric used to define association between vertex IDs

`orderedMM` measure matrix including all pairwise comparisions between vertex IDs based on a particular metric defining assosiction; association measured are sorted from best to worst

Value

None; output file is generated

makeThresholdTable *Calculatiing a varity of network statistics for a number of network models*

Description

This function calls all adjancency matrices which were build by makeAdjMatrices(), removed unconnected vertices from each network model and calculates network statistics for each model. Network statistics include: "R^2" and "slope/power" of the log-transformed connectrivity distribution to determine scale-freeness, "meanCC" and "meanPath" standing for the mean cluster coefficient and mean pathlength within the model to classify the network as small-world, "

Usage

```
makeThresholdTable(metricChoice, thresholds, numGenes,
  outputFile = "NetworkStats.txt")
```

Arguments

`metricChoice`	chracter indicating metric choice, must be one of the following: "PE", "SP", "KE", "EU", "MA", "CA"
`thresholds`	threholds considered to create the different network models
`numGenes`	number of rows in original data matrix
`outputFile`	file name set to "NetworkStats.txt", but can be adjusted

Value

none, output file is generated

`makeVNM` *Creating a vicinity network martrix*

Description

Creats a vicinity network based on one or multiple node. If created on multiple nodes the input nodes must be a clique in the network

Usage

```
makeVNM(currentSubgroup, adjMatrix)
```

Arguments

`currentSubgroup`	a single node identifier or a string of node identifiers (rownames of adjMartrix)
`adjMatrix`	boolean network matrix

Value

vicinity network of input node(s)

plotExpProfiles *Graphing multiple vectors in one plot*

Description

Plots multiple numeric vectors in rainbow color in not differently specified and saves the graph as a high resolution tiff

Usage

```
plotExpProfiles(currentSubgroup, expMatrix, fileName, main = NULL,
  cl = NULL)
```

Arguments

`currentSubgroup`	names of identifiers to be plotted
`expMatrix`	data matrix which includes identifiers in its rownames
`fileName`	file name to which the plot to save to, no file extension needed
`main`	overall title for the plot
`cl`	specification for the default plotting color, default set to rainbow (each vector has its own color, note differentiation between colors might be quite small)

Value

None

plotProfiles *Internal function of plotExpProfiles*

Description

Creates canvas and scales the plot for plotExpProfiles().

Usage

```
plotProfiles(expressionM, cond, header, farbe)
```

Arguments

`expressionM`	datamatrix which to plot all indicies
`cond`	colnames of the data matrix
`header`	overall title for the plot
`farbe`	specification for the default plotting color

Value

None

---

readinExpM *Custom file read function for a numeric data matrix*

Description

Reads in the infromation from a file, file must be tab delimited, first line must be table header, first column must be identifiers, inner part of the matrix must be numeric

Usage

```
readinExpM(fileName)
```

Arguments

`fileName`	file name which to read in, character sting with .txt

Value

readin numeric data matrix, rownames correspond to first column, colnames correspond to first row

---

readinTable *Custom file read function for a table format*

Description

Reads in the information from a file in table format, file must be tab delimited, first line must be table header

Usage

```
readinTable(fileName)
```

Arguments

`fileName`	file name which to read in, character sting with .txt

Value

matrix with no rownames

`writeToFile` *Custom writing to file function*

Description

This is a custom function to write information to file without row or column names

Usage

```
writeToFile(information, fileName, appending = TRUE)
```

Arguments

`information`	material to be written to file, matrix or vector
`fileName`	file name which to write to, charcter sting with .txt
`appending`	whether to append to the specified file or overwrite, boolean variable TRUE(default) or FALSE

Value

None

Index