Package 'RLowPC'

July 19, 2016

Type Package

Title Inference of co-expression gene network using relevance low order partial correlation from large scale expression data

Version 0.1

Date 2016-07-12

Author Wenbin Guo <wenbin.guo@hutton.ac.uk>

Maintainer Wenbin Guo <wenbin.guo@hutton.ac.uk>

Description This package is used to infer and evaluate first order, secnod order, low order and relevace low ordr partial correlation (RLowPC) from large scale gene expression data.
For each pair of genes, the general partial correlation is calculated by removing all the remained controls. However, in large network a number of remained controls may
not truely connect to the pair of genes, which are inrrelevant controls. To increase the precision of network predictions, we use RLowPC method to calculated partial
correlation by only removing the more relevant controls.

License GPL-3

LazyData TRUE

Depends R (>= 3.2.3)

RoxygenNote 5.0.1

R topics documented:

`adj2rankadj` *Convert network matrix to network rank matrix*

Description

The edge weights in the network matrix will be converted to their ranks. A high rank value indcates a high edge connection.

Usage

```
adj2rankadj(adjmatrix, directed = F)
```

Arguments

`adjmatrix`	A network matrix
`directed`	Logical. If TRUE, the network is considered as directed. If FALSE, the upper triangular part of the symmetric network matrix is used to calcuate the rank matrix.

Value

`adj2rankadj` returns a network matrix with ranks of the edge weightes.

`adjmatrix2edgelist` *Convert network matrix to edge list*

Description

Convert the network matrix to edge list, in which the first column contains the regulators, the second column presents the target genes and edge weights are in the third column.

Usage

```
adjmatrix2edgelist(adjmatrix, cutoff = 0, directed = F, order = T)
```

Arguments

`adjmatrix`	a network matrix
`cutoff`	threshod to cut the edge weights. Only the edges that pass the threshold will be shown in the output results.
`directed`	logical, if `FLASE` the adjmatrix is transformed to symmetric matrix and the upper triangular part of the matrix is used to generate the edge list.
`order`	logical, to order the edgelist according the edge weightes or not.

Value

adjmatrix2edgelist returns a data frame of edge list.

Examples

```
##load data
library(RLowPC)
data(gnwdata)
data.exp<-gnwdata$size100$ts1[,-c(1:3)]
genes<-colnames(data.exp)
##build correlation network
inf.cor<-abs(cor(data.exp))
diag(inf.cor)<-0
##convert matrix to edge list
adjmatrix2edgelist(inf.cor)
```

---

anova2de *Use ANOVA to filter low epxressed genes*

Description

Analysis of variance is used to filter the low expressed genes. More details can be seen in aov.

Usage

```
anova2de(data.exp, ncol.idx, pval.cut = 0.05, model = "expression~time")
```

Arguments

data.exp	gene expression data matrix with variables in columns and samples in rows. The first columns are the indeces for comparison, such as time points and conditions for comparision.
ncol.idx	the number of index columns.
pval.cut	a numeric value for significance cut-off. The default is pval.cut=0.05.
model	the model used for comparison. The default is model="expression~time".

Value

anova2de returns a list includes the ANOVA p-value results, the names of differentially expressed genes and the time-series data of differential expressed genes.

See Also

aov, RLowPC

average.consensus *Consensus network integrated from average edge ranks*

Description

Consensus network is built from taking the average ranks of the edges in multiple network predictions.

Usage

```
average.consensus(adjmatrix.list, directed = F)
```

Arguments

adjmatrix.list	a list of inferred network matrices with same row and column ordering.
directed	logical. If TRUE, the network is considered as directed. If FALSE, the upper triangular part of the symmetric matrix is used to calculate the rank matrix

Value

average.consensus returns a network with rank average edge ranks. The weights are rescaled to 0-1 and hihger values indicate higher ranks.

Examples

```
##create two random networks
library(RLowPC)
set.seed(4)
net1<-abs(matrix(rnorm(16),4,4))
net1<-pmax(net1,t(net1))
diag(net1)<-0
set.seed(5)
net2<-abs(matrix(rnorm(16),4,4))
net2<-pmax(net2,t(net2))
diag(net2)<-0
dimnames(net1)<-dimnames(net2)<-list(letters[1:4],letters[1:4])
net.list<-list(net1=net1,net2=net2)
inf.consensus<-average.consensus(adjmatrix.list = net.list,directed = F)
adj2rankadj(net1)
adj2rankadj(net2)
inf.consensus
```

confusion *Statistical derivations of confusinon table*

Description

Calculate statistical measures of the performance of binary classification [1] from the output of confusion matrix in table.evaluate.

Usage

```
confusion(input.table)
```

Arguments

input.table	the output confusion table from table.evaluate.

Details

true positive: tp; false positive: fp; true negative: tn; false negative: fn;
positives in reference network: p; negatives in reference network: n;
true positive rate: $tpr = recall = \frac{tp}{tp+fn}$; false positive rate: $fpr = \frac{fp}{fp+tn}$;
true negative rate: $tnr = \frac{tn}{tn+fp}$; false negative rate: $fnr = \frac{fn}{fn+tp}$;
precision: $precision = \frac{tp}{tp+fp}$; negative predictive value: $npv = \frac{tn}{tn+fn}$;
false discovery rate: $fdr = \frac{fp}{fp+tp}$; accuracy: $accuracy = \frac{tp+tn}{p+n}$;
f1 scaore: $f1 = \frac{2tp}{2tp+fp+fn}$;
Matthews correlation coefficient:

$$mcc = \frac{tp \times tn - fp \times fn}{\sqrt{(tp+fp) \times (tp+fn) \times (tn+fp) \times (tn+fn)}}$$

Value

confusion returns a data frame of measures of performance, see Details.

References

1. Powers DMW: Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. In. Adelaide, Australia; 2007.

---

cor2mi *Convert correlation matrix to MI matrix*

Description

The correlation ρ_{ij} between node i and j is converted into mutual information (MI) matrix by using equation [1]:

$$MI = -\frac{1}{2}(1 - \rho_{ij}^2)$$

if the observation data is normal distributed.

Usage

```
cor2mi(x)
```

Arguments

x	a correlation matrix

Value

cor2mi returns a mutual information matrix.

References

[1] Meyer PE, Kontos K, Lafitte F, Bontempi G: Information-theoretic inference of large transcriptional regulatory networks. EURASIP J Bioinform Syst Biol 2007:79879.

`cor2statistics` *Convert correlation to statistical significance measures.*

Description

`cor2statistics` convert a correlation vector to p-values, false discovery rates (FDRs) and probability of connections.

Usage

```
cor2statistics(cor.vector, ...)
```

Arguments

`cor.vector`	a correlation vector
`...`	parameters used in `fdrtool`.

Value

`cor2statistics` returns a list includes correlation matrix and p-value matrix.

See Also

`fdrtool`

`edgelist2adjmatrix` *Convert edge list to network matrix*

Description

The function is to convert edge list to network matrix.

Usage

```
edgelist2adjmatrix(edgelist, genes, cutoff = 0, directed = F)
```

Arguments

`edgelist`	a data frame of edge list of a network, in which columns are regulators, targets and edge weights.
`genes`	gene names to name the rows and columns of the output network matrix.
`cutoff`	the threshold to cut the edge list.
`directed`	logical, to create directed or undirected (symmetric) network matrix.

Value

`edgelist2adjmatrix` returns a network matrix.

Examples

```
##load data
library(RLowPC)
data(gnwdata)
data.exp<-gnwdata$size100$ts1[,-c(1:3)]
genes<-colnames(data.exp)
ref.edge<-gnwdata$size100$net1
ref.edge[,3]<-1
ref.adj<-edgelist2adjmatrix(ref.edge,genes,directed=F)
```

`firstPC` *First order partial correlation*

Description

First order PC is calcuated according to the edge connections in the input edgelist. The correlation of a pair of genes are calcuated as (a) correlation if they do not connect to the same neighbour genes and (b) the maximum of first order PC values by removing the controls one by one if they connect to shared neighbour genes. After that, the `cor2statistics` is used to calculate the p-value, FDR and connection probability from the output correlation.

Usage

```
firstPC(data.exp, edgelist, controlist = NULL, method = "pearson",
  progressbar = T)
```

Arguments

`data.exp`	gene expression data matrix with genes in columns and samples in rows.
`edgelist`	an edge list for fisrt order PC calculation. The PC values are calcuated over each pair of genes in the gene list.
`controlist`	list of vectors of neighbour genes for each pair of genes in `edgelist`. The vector order in the list must match to the order of corresponding paired genes in the input `edgelist`. For example the i element in `controlist` contains the shared neighbours of gene `edgelist[i,1]` and gene `edgelist[i,2]`.
`method`	estimators of correlation. Options are "pearson", "spearman" and "kendall".
`logical.`	If TRUE, a progressbar will show to indicate the code runing percentage.

Value

`firstPC` returns a data frame with columns of regulators, target genes, edge correlation weigthes, p-values, FRD and connection probability.

See Also

`cor2statistics`, `RLowPC`

`fit.structure` *Calculate structural fitness of networks*

Description

The function is to used to calcuate the scale-free or modularity fitness of networks. The scale-free fitness is calcuated as $-R^2 \times slope$ of the fit index of the model. Please refer to the package WGCNA [1,2] for details.The fitness of modularity is computed using igraph package [3].

Usage

```
fit.structure(edgelist, method = "scale-free")
```

Arguments

`edgelist`	a data frame of network edge list, with regulators in first column, target genes in second column and edge weights in third column.
`method`	a character string used to calculate the structural fitness. Options are "scale-free" and "modularity".

Value

`fit.structure` returns a numeric value measures the structural fitness.

References

[1] Langfelder P, Horvath S: WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics 2008, 9:559.

[2] Zhao W, Langfelder P, Fuller T, Dong J, Li A, Hovarth S: Weighted gene coexpression network analysis: state of the art. J Biopharm Stat 2010, 20(2):281-300.

[3] Csardi G, Nepusz T: The igraph software package for complex network research. InterJournal 2006, Complex Systems:1695.

Examples

```
library(RLowPC)
data(gnwdata)
edgelist<-gnwdata$size500$net1
edgelist[,3]<-1
fit.structure(edgelist,method='scale-free')
fit.structure(edgelist,method='modularity')
```

gnwdata *Synthetic data sources simulated using GeneNetWeaver*

Description

The sources include time-series composed of 1 perturbation x 5 experiments x 3 replicates for GNW100 and GNW500 networks, respectively. GNW100 includes 5 *In Silico* network structures with 100 genes used in DREAM4 challenge while GNW500 includes 5 network structures with 500 genes which are subsets of *E.coli* source network in GeneNetweaver. Each time course is sampled at 21 time points. In each experiment, one third of genes are randomly selcted and perturbed from steady states at initial time point and the perturbations are removed at half time duration. Biological and technical noises are added to the datasets according to the settings in DREAM4 challenge [1,2].

Usage

```
data(gnwdata)
```

Format

An object of data list

References

[1] Schaffter T, Marbach D, Floreano D: GeneNetWeaver: In silico benchmark generation and performance profiling of network inference methods. Bioinformatics 2011, 27(16):2263-2270.

[2] Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G: Revealing strengths and weaknesses of methods for gene network inference. Proceedings of the National Academy of Sciences 2010, 107(14):6286-6291.

Examples

```
data(gnwdata)
sapply(gnwdata,names)
```

LowPC *Low order partial correlation*

Description

Calculate the low order PC as shown in paper [1].

Usage

```
LowPC(data.exp, cutoff = 0.05, cutat = "pval", method = "pearson",
  progressbar = T)
```

Arguments

data.exp	gene expression data matrix with genes in columns and samples in rows.
cutoff	a cut-off of p-values or conenction proability.
cutat	a string indicates to cut the results at p-value "pval" or probability "prob" with a threshold cutoff.
method	a string character of method used to estimate correlation. Options are "pearson", "spearman" and "kendall".
progressbar	logical. If TRUE, a progressbar will show to indicate the code runing percentage.

Details

Zero order PC (correlation) is firstly computed. The significant edges in zero order PC network are used to calcuate first order PC. After that, the significant edges in first order PC are used to calculate second order PC. Instead of regressing all the remained genes one by one or pair by pair, the calculation of first and second PC only based on the subset of edges that significantly connect to the genes, which greatly reduces the computational cost compared to the paper [1]. For example, in a network with n genes, the first order PC between gene i and j is calculated by removing m shared genes, which significantly connect to i and j, one by one rather than all the $n-2$ remained genes ($m < n-2$). If a pair of genes do not connect to the same set of genes, the correlation between them is represeted as correlation, while if they connected only one shared gene, the edge connection is weighted as first order PC. The correlation significance is estimated using R package fdrtool.

Value

LowPC returns a list includes the results of zero order PC, first order PC and second order PC.

References

1. Zuo Y, Yu G, Tadesse MG, Ressom HW: Biological network inference using low order partial correlation. Methods (San Diego, Calif) 2014, 69(3):266-273.

See Also

zeroPC, firstPC, secondPC, RLowPC and cor2statistics.

---

node.degree *Calculate degree of nodes in gene networks*

Description

Calculate degree of nodes in gene networks

Usage

```
node.degree(net, directed = F)
```

Arguments

net	input network matrix or edge list with three columns, i.e. regulators, targets and weights.
directed	logical, if FALSE, the input network is treated as undirected.

Value

a vector of node degrees

---

plotAUC *Plot AUC curves*

Description

The function is used to plot AUROC, AUPR, pAUROC and pAUPR curves.

Usage

```
plotAUC(table.methods, plot.method = c("auroc", "aupr", "top.precision",
  "fpr.precision"), roc.lim = c(0, 1, 0, 1), pr.lim = c(0, 1, 0, 1),
  top.precision.lim = c(0, 1000, 0, 1), fpr.precision.lim = c(0, 1, 0, 1),
  fill = F, color = NA, plot.ncol = 2, ...)
```

Arguments

table.methods	a list of confusion tables of comparison for multiple networks to the same reference network. The tables are the outputs of table.evalution. The names of the list names(table.methods) will be used to plot the legend of the curves.
plot.method	a character string indicates to plot AUROC ("auroc"), AUPR ("aupr"), precision vs top weighted edges ("top.precision") or precision vs false positive rate ("fpr.precision") curves.
roc.lim	a vector to specify the plot limits, i.e. xmin, xmax, ymin and ymax of AUROC curves.
pr.lim	a vector to specify the plot limits, i.e. xmin, xmax, ymin and ymax of AUPR curves.
top.precision.lim	a vector to specify the plot limits, i.e. xmin, xmax, ymin and ymax of precision vs top weighted edges curves.
fpr.precision.lim	a vector to specify the plot limits, i.e. xmin, xmax, ymin and ymax of precision vs false positive rate curves.
fill	logical. If TRUE, the area under the curvers will be filled with colors.
color	a color string vectors to specify the colors for the curves.
plot.ncol	the number of columns of the layout of mutliple plots.
...	additional options for geom_line(...).

Value

plotAUC returns plots.

Examples

```
##load librarys
library(RLowPC)
library(minet)
library(gridExtra)
library(ggplot2)
##get dream4 datasets
data(gnwdata)
data.exp<-gnwdata$size100$ts1[,-c(1:3)]
genes<-colnames(data.exp)
ref.edge<-gnwdata$size100$net1
ref.edge[,3]<-1
ref.adj<-edgelist2adjmatrix(ref.edge,genes)
##infer gene networks
inf.cor<-abs(cor(data.exp))
diag(inf.cor)<-0
inf.mi<-build.mim(data.exp)
inf.clr<-clr(inf.mi)
inf.mrnet<-mrnet(inf.mi)
##generate confusion tables
table.cor<-table.evaluate(inf.adj = inf.cor,ref.adj = ref.adj)
table.mi<-table.evaluate(inf.adj=inf.mi,ref.adj = ref.adj)
table.clr<-table.evaluate(inf.adj=inf.clr,ref.adj = ref.adj)
table.mrnet<-table.evaluate(inf.adj=inf.mrnet,ref.adj = ref.adj)
##put confusion tables into list, and set names as the methods
table.methods<-list(cor=table.cor,mi=table.mi,clr=table.clr,mrnet=table.mrnet)
plotAUC(table.methods,fill=T,lwd=1)
```

`random.net` *Resort datasets and generate random networks*

Description

The function is used to generate random datasets or random networks

Usage

```
random.net(input, nexp = NA, type = "data", directed = F)
```

Arguments

`input`	a gene expression data frame or a string vector of gene names.
`nexp`	a numeric number to indicate how many experiments in the datasets. The gene expression datasets are resorted in both experiment-wise and gene-wise.
`type`	a character string to indicate the resort datasets ("data") or to generate random networks ("network").
`directed`	logical, if `FALSE`, the random network generated from gene names will be converted to symmetric matrix.

Details

(a) Input a gene expression data frame with genes in columns and samples in rows, the function resorts the sample points for each genes in each experiments. (b) Input a string vector of gene names, the function generates random network with random edge weightes.

Value

`random.net` returns a randomly resorted expression data frame or a random network matrix.

RLowPC *Relevance low order partial correlation*

Description

Relevance low order partial correlation (RLowPC) is an improved version of partial correlation (PC)[1, 2]. Instead of removing all remained controls for pair-wise PC calculation, RLowPC selects and regresses the the most relevant controls. See Details.

Usage

```
RLowPC(data.exp, edgelist, method = "pearson", pc.estimator = "shrink",
  progressbar = T)
```

Arguments

`data.exp`	gene expression matrix. Columns are variables and rows are samples.
`edgelist`	edge list. First column are the name of regulators, second coloumn are the target genes and the third column are the edge weights.
`method`	a character string to indicate which method is used to calculate correlation. Options are "pearson", "spearman" and "kendall". If `pc.estimator="shrink"`, the method is set to "pearson" since there are no "spearman" and "kendall" options for the function `pcor.shrink` to estimate shrink PC.
`pc.estimator`	a character string to indicate the estimator used to calculate the PC for each pair of nodes in the edge list. Options are "shrink" and "pc", correspoinding to the item (c) and (d) in Details: Step 2, respectively.
`progressbar`	logical. If TRUE, a progressbar will show to indicate the code runing percentage.

Details

In general cases, the PC for a pair of genes is calculated by removing all the remained genes (controls). However, there may be a number of inrrelevant controls involved in the removed genes that do not truely connect to the pair of genes. We developed a RLowPC method to calculate PC by only removing more revelant controls. The method is used to refine a pre-inferred network structure by reducing the indirect edges that have been predicted as direct.

Step 1: Input a proper size of pre-inferred network, which has room to be improved, for search space of indirect edges. For example the top weighted edges in a inferred PC network can be used as search space. Each pair of genes are assumed to connect to their most relevant neighbour genes since the connecting edges are all top ranked in the whole network.

Step 2: Calculate relevance low order partial correlation. For each pair of genes connected by an edge in the searching space, the edge weight is redefined as (a) zero order PC (correlation) if they do not connect to the same set of neighbour genes, (b) PC by removing all the shared neighbours simultaneously and (c) shrink PC if the covariance matrix used to estimate PC in (b) is not positive definite or invertible. If the searching space is very large, there might still be a number of irrelevant controls involved in shrink PC procedure in (c). An alternative is (d) deleting less connected neighbour genes until the covariance matrix in (b) is positive definite and invertible.

Value

RLowPC regurns a new edge list with an additional column of RLowPC edge weights.

References

1. Markowetz F, Spang R: Inferring cellular networks-a review. BMC Bioinformatics 2007, 8 Suppl 6:S5.

2. Sch\"afer J, Strimmer K: A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. Statistical Applications

Examples

```
##load library
library(RLowPC)
library(corpcor)
##load data
data(gnwdata)
data.exp<-gnwdata$size100$ts1[1:63,]
genes<-colnames(data.exp)[-c(1:3)]
##load reference network
ref.edge<-gnwdata$size100$net1
ref.edge[,3]<-1
ref.adj<-edgelist2adjmatrix(edgelist = ref.edge,genes = genes,directed = F)


##filter low expressed genes
data2anova<-data.frame(time=factor(paste0(data.exp$experiment,'_',data.exp$time)),data.exp[,-c(1:3)])
data.new<-anova2de(data.exp = data2anova,ncol.idx =1,model = 'expression~time',pval.cut = 0.01)
data.exp<-data.new$de.ts
genes<-data.new$de.gene
ref.adj<-ref.adj[genes,genes]

##infer correlation network
inf.cor<-abs(cor(data.exp))
diag(inf.cor)<-0
##infer PC network
inf.pcor<-abs(pcor.shrink(data.exp)[1:length(genes),1:length(genes)])
diag(inf.pcor)<-0

##infer LowPC
inf.LowPC.edge<-LowPC(data.exp,cutoff = 0.01,cutat='pval')
if(is.null(inf.LowPC.edge$secondPC)){
  inf.LowPC<-ref.adj*0 } else {
   inf.LowPC<-edgelist2adjmatrix(inf.LowPC.edge$secondPC[,1:3],genes = genes,directed = F)
  }

##inf RLowPC
```

```
reduction.sapce<-na.omit(adjmatrix2edgelist(adjmatrix = inf.pcor,directed = F,order = T)[1:1000,])
inf.RLowPC.edge<-RLowPC(data.exp = data.exp,edgelist = reduction.sapce,
                        method = 'pearson',pc.estimator = 'shrink')
inf.RLowPC.edge$cor.weight<-abs(inf.RLowPC.edge$cor.weight)
inf.RLowPC<-edgelist2adjmatrix(inf.RLowPC.edge[,c(1,2,4)],genes = genes,directed = T)
inf.RLowPC<-abs(inf.RLowPC)
inf.RLowPC<-pmax(inf.RLowPC,t(inf.RLowPC))

##infer first order PC based on reduction sapce
###first PC
inf.firstPC.edge<-firstPC(data.exp = data.exp,edgelist = reduction.sapce,
                          method = 'pearson',controlist = NULL)
inf.firstPC<-edgelist2adjmatrix(inf.firstPC.edge[,1:3],genes = genes,directed = F)
###second PC
inf.secondPC.edge<-secondPC(data.exp = data.exp,edgelist = reduction.sapce,
                            method = 'pearson',controlist = NULL)
inf.secondPC<-edgelist2adjmatrix(inf.secondPC.edge[,1:3],genes = genes,directed = F)
##Put the inferred networks into a list.
inf.list<-list(Cor=inf.cor,PC=inf.pcor,LowPC=inf.LowPC,firstPC=inf.firstPC,
               secondPC=inf.secondPC,RLowPC=inf.RLowPC)
sapply(inf.list,dim)
dim(ref.adj)
##calculate confusion table
inf.table<-lapply(inf.list,function(x) table.evaluate(x,ref.adj = ref.adj))
##cut the tables at top 1000 predictions
inf.table<-lapply(inf.table,function(x) table.cut(input.table = x)$table.output)

##plot the figure
x11()
plotAUC(table.methods = inf.table,
        roc.lim = c(0,1,0,1),
        pr.lim = c(0,1,0,0.4),
        top.precision.lim = c(0,700,0,0.4),
        fpr.precision.lim = c(0,1,0,0.4),
        color = c('blue','purple','green','lightblue','red','yellow'),
        lwd=1,fill = T)
```

---

secondPC *Second order partial correlation*

Description

Second order PC is calcuated according to the edge connections in the input edgelist. The correlation of a pair of genes are calcuated as (a) correlation if they do not connect to the same neighbour genes; (b) first order PC if they only connect to one shared genes and (c) second order PC by removing the controls pair by pair if they connect to more than two shared neighbours genes. After that, the fdrtool is used to calculate the p-value, FDR and connection probability from the correlation.

Usage

```
secondPC(data.exp, edgelist, controlist = NULL, method = "pearson",
  progressbar = T)
```

Arguments

data.exp	gene expression data matrix with genes in columns and samples in rows.
edgelist	an edge list for second order PC calculation. The PC values are calcuated over each pair of genes in the gene list.
controlist	list of vectors of neighbour genes for each pair of genes in edgelist. The vector order in the list must match to the order of corresponding paired genes in the input edgelist. For example the i element in controlist contains the shared neighbours of gene edgelist[i,1] and gene edgelist[i,2]. the i
method	estimators of correlation. Options are "pearson", "spearman" and "kendall".
logical.	If TRUE, a progressbar will show to indicate the code runing percentage.

Value

secondPC returns a data frame with columns of regulators, target genes, edge correlation weigthes, p-values, FRD and connection probability.

---

shared.neighbour *Extract shared neihbour genes*

Description

The function is used to extract the shared neighbour genes that both connect to a pair of candidate nodes.

Usage

```
shared.neighbour(node1, node2, edgelist, verbose = T)
```

Arguments

node1	a character string of the name of candidate node 1.
node2	a character string of the name of candidate node 2.
edgelist	the edge list of a network from which the neighbour genes are extracted for node1 and node2.

Value

shared.neighbour returns a sub edgelist.

table.cut *Cut the confusion table at threshods*

Description

The confusion table output from `table.evaluate` is cut at a threshold of top weighted edges, FPR or TPR. The `cutoff` in the function must match to the `cutat` metrices.

Usage

```
table.cut(input.table, cutoff = 1000, cutat = "top")
```

Arguments

`input.table`	the confusion table output from table.evalution.
`cutoff`	a numeric value of threshold for "top", "fpr" or "tpr".
`cutat`	the options are "top", "fpr" and "tpr", which indicate to cut the confusion table at top weighted edges, FPR or TPR.

Value

`table.cut` returns a list of results cantains the table after applying cutoff, the pAUROC value, pAUPR value, precesion for the remained edges, the corresponding top, fpr, tpr cutoffs and the statistical measures after applying function confusion.

table.evaluate *Evaluation of inferred gene networks*

Description

The inferred network is evaluated by comparing to the reference network. The output is a tables of TP, FP, TN and FP with different edge weight cut-offs [1].

Usage

```
table.evaluate(inf.adj, ref.adj, directed = F)
```

Arguments

`inf.adj`	the inferred network matrix. Column names and row names match to the reference network.
`ref.adj`	the reference network matrix with 1 inidating connected edge and 0 unconnected edge.
`directed`	logical, to compare as directed or undirected networks. In a undirected network, only the upper triangular of the network matrix is used for evaluation.

References

[1] Meyer PE, Lafitte F, Bontempi G: minet: A R/Bioconductor package for inferring large transcriptional networks using mutual information. BMC Bioinformatics 2008, 9:461.

Examples

```
##load library
library(RLowPC)
##load data
data(gnwdata)
data.exp<-gnwdata$size100$ts1[,-c(1:3)]
genes<-colnames(data.exp)
ref.edge<-gnwdata$size100$net1
ref.edge[,3]<-1
ref.adj<-edgelist2adjmatrix(ref.edge,genes)
inf.cor<-abs(cor(data.exp))
diag(inf.cor)<-0
table.cor<-table.evaluate(inf.adj = inf.cor,ref.adj = ref.adj)
head(table.cor)
```

`zeroPC` *Zero order partial correlation*

Description

The correlation and p-value of correlation.

Usage

```
zeroPC(data.exp, method = "pearson")
```

Arguments

`data.exp`	gene expression data.
`method`	"pearson", "spearman" or "kendall" method is used to calcuate the correlation values.

Value

`zeroPC` returns a data frame with columns of regulators, target genes, edge correlation weigthes, p-values, FRD and connection probability.

Examples

```
##load data
data(gnwdata)
##ts expression
data.exp<-gnwdata$size100$ts1[,-c(1:3)]
inf.zeroPC<-zeroPC(data.exp)
head(inf.zeroPC)
##refernece network
```

Index