

# Package ‘RLowPCor’

June 8, 2016

**Type** Package  
**Title** Gene regulatory network construction, refinement and evaluation on large scale gene expression data  
**Version** 0.1  
**Date** 2016-06-07  
**Author** Wenbin Guo <wenbin.guo@hutton.ac.uk>  
**Maintainer** Wenbin Guo <wenbin.guo@hutton.ac.uk>  
**Description** The package is used to construct, integrate, refine and evaluate gene regulatory networks on large scale gene expression data.  
**License** GPL-3  
**LazyData** TRUE  
**Depends** R (>= 3.2.3),  
ggplot2,  
plyr  
**RoxygenNote** 5.0.1

## R topics documented:

adj2rankadj . . . . .	1
adjmatrix2edgelist . . . . .	2
average.consensus . . . . .	3
confusion . . . . .	3
edgelist2adjmatrix . . . . .	5
RLowPCor . . . . .	5
table.evaluate . . . . .	7
<b>Index</b>	<b>8</b>

---

adj2rankadj	<i>Convert network matrix to network rank matrix</i>
-------------	--

---

## Description

Convert network matrix to network rank matrix

**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 matrix is used to calculate the rank matrix

**Value**

an network connection rank matrix

---

adjmatrix2edgelist	<i>Convert network matrix to edge list</i>
--------------------	--

---

**Description**

The inferred network matrix is converted to edge list

**Usage**

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

**Arguments**

adjmatrix	a network matrix
cutoff	threshold to cut the edge list
directed	logical, if FLASE the adjmatrix is transformed to symmetric matrix

**Value**

a edge list

**Examples**

```
##load data
library(networkBMA)
library(RLowPCor)
data(dream4)
data.exp<-dream4ts10[[1]][,-c(1:2)]
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)
```

---

average.consensus	<i>Consensus network from average rank</i>
-------------------	--

---

### Description

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

### Usage

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

### Arguments

`adjmatrix.list` A list of network prediction matrices with same rownames and colnames.  
`directed` Logical. If TRUE, the network is considered as directed. If FALSE, the upper triangular part of the matrix is used to calculate the rank matrix

### Value

a network with rank weighted edges. The weights are rescaled to 0-1 and higher values indicate higher ranks.

### Examples

```
##create two random networks
library(RLowPCor)
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	<i>Derivations of confusion</i>
-----------	---------------------------------

---

### Description

Calculate statistical measures of the performance of binary classification test from the output confusion matrix [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:  $fpr = \frac{fn}{fn+tp}$ ;  
 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

a table of of statistical measures of performance, see [@details](#)

## References

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

## Examples

```
library(networkBMA)
library(RLowPCor)
##load DREAM4 size100_1 datasets
data(dream4)
data.exp<-dream4ts10[[1]][,-c(1:2)]
genes<-colnames(data.exp)
ref.edge<-dream4gold10[[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)
```

---

edgelist2adjmatrix	<i>Convert edge list to network matrix</i>
--------------------	--

---

**Description**

The function is to convert edge list to network matrix.

**Usage**

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

**Arguments**

edgelist	a edge list of network
genes	gene names
cutoff	the threshold to cut the edge list
directed	logical, to create directed or undirected network matrix

**Value**

a network matrix

**Examples**

```
library(networkBMA)
library(RLowPCor)
##load DREAM4 size100_1 datasets
data(dream4)
data.exp<-dream4ts10[[1]][,-c(1:2)]
genes<-colnames(data.exp)
ref.edge<-dream4gold10[[1]]
ref.adj<-edgelist2adjmatrix(ref.edge,genes)
```

---

RLowPCor	<i>Relevance low order partial correlation</i>
----------	--

---

**Description**

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

**Usage**

```
RLowPCor(data.exp, edgelist, estimator = "pearson", pc.estimator = "shrink")
```

## Arguments

<code>data.exp</code>	gene expression matrix. Columns are variables and rows are samples.
<code>edgelist</code>	edge list. First column are the name of regulators, second column are the target genes and the third column are the edge weights.
<code>estimator</code>	a character string indicating which correlation coefficient (or covariance) is to be computed. Options are "pearson", "spearman" and "kendall". If shrinkage method is used to estimate PC, the estimator is set to "pearson".
<code>pc.estimator</code>	a character string indicating which method is used to estimate the PC of nodes connected to shared neighbours. Options are "shrink" and "pc", corresponding to the item (c) and (d) in Details: Step 2, respectively.

## Details

Step 1: Extract a sparse and scale-free topology from pre-inferred networks as an indirect edge search space for RLowPCor. Unlike a fully connected network, the nodes in the sparse network are assumed to connect to their more relevant neighbours. For example, correlation network can be cut with a range of thresholds until it most fit to scale-free topology. Step 2: Calculate relevance low order partial correlation. For each pair of nodes connected by an edge in the searching space, the edge weight is redefined as (a) Pearson correlation if they do not connect to the same set of neighbour nodes, (b) PC by removing shared neighbours 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 (c). An alternative is (d) deleting less connected neighbours of the nodes until the covariance matrix in (b) is positive definite and invertible.

## Value

a network matrix

## References

Schöberl J, Strimmer K: A Shrinkage Approach to Large-Scale Covariance Matrix Estimation and Implications for Functional Genomics. Statistical Applications in Genetics and Molecular Biology, The Berkeley Electronic Press 2005, 4(1).

## Examples

```
##load size 100_1 network DREAM4 datasets
library(networkBMA)
library(RLowPCor)
data(dream4)
data.exp<-dream4ts100[[1]]
#create edge list
edgelist<-dream4gold100[[1]]
edgelist<-edgelist[edgelist[,3]>0,]
##infer RLowPCor network
inf.net=RLowPCor(data.exp = data.exp[, -c(1:2)], edgelist = edgelist)
```

---

table.evaluate	<i>Evaluate inferred network to refence network</i>
----------------	---

---

### Description

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

### 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 indicating 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.

### Value

a confusion table of TP, FP, TN and FN

### References

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

```
library(networkBMA)
library(RLowPCor)
##load DREAM4 size100_1 datasets
data(dream4)
data.exp<-dream4ts10[[1]][, -c(1:2)]
genes<-colnames(data.exp)
ref.edge<-dream4gold10[[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)
```

# Index

`adj2rankadj`, [1](#)  
`adjmatrix2edgelist`, [2](#)  
`average.consensus`, [3](#)  
  
`confusion`, [3](#)  
  
`edgelist2adjmatrix`, [5](#)  
  
`RLowPCor`, [5](#)  
  
`table.evaluate`, [3](#), [4](#), [7](#)