# Package 'dna'

July 2, 2020

**Version** 2.1-2

**Date** 2020-06-30

**Depends** R (>= 4.0.0)

**Imports** methods

**Suggests** igraph, lars

**Description** Package for conducting differential network analysis from microarray data.

**Title** Differential Network Analysis

**LazyData** yes

**License** GPL-2 | GPL-3

**NeedsCompilation** yes

**Author** Ryan S. Gill [aut, cre],
Somnath Datta [aut],
Susmita Datta [aut]

**Maintainer** Ryan S. Gill <ryan.gill@louisville.edu>

**Repository** CRAN

**Date/Publication** 2020-07-02 16:50:08 UTC

## R topics documented:

cornet . . . . . 2
gennet . . . . . 3
get.common.networks-methods . . . . . 4
get.modules-methods . . . . . 5
get.network1-methods . . . . . 6
get.network2-methods . . . . . 6
get.results-methods . . . . . 7
HeavyMice . . . . . 8
LeanMice . . . . . 9
modules-class . . . . . 10
network.modules . . . . . 11

pairOfNetworks-class . . . . . 12
PCnet . . . . . 13
PLSnet . . . . . 14
resultsClassTest-class . . . . . 15
resultsIndTest-class . . . . . 16
resultsModTest-class . . . . . 17
RRnet . . . . . 19
summary-methods . . . . . 20
test.class.genes . . . . . 21
test.individual.genes . . . . . 23
test.modular.structure . . . . . 25

**Index** **28**

---

`cornet` *Correlation network*

---

## Description

Computes the connectivity scores for a network based on correlation.

## Usage

```
cornet(data,rescale.scores=FALSE)
```

## Arguments

| | |
|---|---|
| `data` | microarray dataset with genes in columns and samples in rows. |
| `rescale.scores` | indicates whether PLS scores should be rescaled so that the largest score for each gene should be 1 in magnitude. |

## Value

| | |
|---|---|
| `cornet` | a correlation matrix measuring the interactions between gene pairs. |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# small example using cornet without rescaled scores
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
s=cornet(X1)
print(round(s,4))

# small example using cornet with rescaled scores
s2=cornet(X1,rescale.scores=TRUE)
print(round(s2,4))
```

---

gennet *General Regression network*

---

## Description

Computes the connectivity scores for a network based on a specified regression method.

## Usage

```
gennet(data, f, recenter.data=FALSE, rescale.data=FALSE,
symmetrize.scores=FALSE, rescale.scores = FALSE, ...)
```

## Arguments

| | |
|---|---|
| data | microarray dataset with genes in columns and samples in rows. |
| f | regression method. |
| recenter.data | indicates whether data should be recentered. |
| rescale.data | indicates whether data should be rescaled. |
| symmetrize.scores | indicates whether PLS scores should be made to be symmetric. |
| rescale.scores | indicates whether PLS scores should be rescaled so that the largest score for each gene should be 1 in magnitude. |
| ... | Any additional arguments for f. |

## Value

| | |
|---|---|
| gennet | a matrix of interactions between gene pairs based on the regression method supplied by the user. |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

Hastie, T., Tibshirani, R., and Friedman, J. (2009) *The Elements of Statistical Learning*. Springer: New York.

## Examples

```
# small example using gennet with a user-defined ridge regression
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))

## Not run: ourRR=function(X,y,lambda=1){
## solve(t(X)%*%X+diag(ncol(X)))%*%t(X)%*%y}
## Not run: gennet(X1,f=ourRR,recenter.data=
## TRUE,rescale.data=TRUE,symmetrize.scores=
## TRUE,rescale.scores=FALSE)

# compare results with RRnet
RRnet(X1)
```

---

# get.common.networks-methods

*Method for Function* `get.common.networks`

---

## Description

get.common.networks-methods

## Methods

`signature(object = "pairOfNetworks")` returns the pair of networks from an object of class "pairOfNetworks". The network includes only the genes common to both networks, and the genes are listed in the same order for each network.

## Examples

```
# small example illustrating how a pair of networks is
# preprocessed to obtain and align a set of common genes
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")
```

```
X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")
networks=new("pairOfNetworks",network1=X1,network2=X2)
get.common.networks(networks)
```

---

`get.modules-methods` *Methods for Function* `get.modules`

---

## Description

get.modules-methods

## Methods

`signature(object = "modules")` returns the module number for each gene in a network. For the genes that are in none of the modules, the module number is listed as 0.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# artificial example to show how to obtain modules from a matrix of
# connectivity scores
set.seed(26)
s=matrix(runif(100,-1,1),10,10);diag(s)=1;s=round((s+t(s))/2,1)
the.modules=network.modules(s,m=3,epsilon=.7)
the.modules

# method for extracting the modules
get.modules(the.modules)
```

## get.network1-methods *Method for Function* get.network1

### Description

get.network1-methods

### Methods

`signature(object = "pairOfNetworks")` returns the first network from an object of class "pairOfNetworks".

### Examples

```
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")
X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")
networks=new("pairOfNetworks",network1=X1,network2=X2)
get.network1(networks)
```

## get.network2-methods *Method for Function* get.network2

### Description

get.network2-methods

### Methods

`signature(object = "pairOfNetworks")` returns the second network from an object of class "pairOfNetworks".

## Examples

```
# small example illustrating how a pair of networks is
# preprocessed to align a set of common genes and extract network 2
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")
X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")
networks=new("pairOfNetworks",network1=X1,network2=X2)
get.network2(networks)
```

---

# get.results-methods *Method for Function* get.results

---

## Description

get.results-methods

## Methods

`signature(object = "resultsIndTest")` returns the p-values and test statistics for tests for differential connectivity of individual genes.

`signature(object = "resultsClassTest")` returns the p-value, test statistic, and the class of genes for a test for differential connectivity of the class of genes.

`signature(object = "resultsModTest")` returns the p-value, test statistic, and the modules for each network for a test for overall modular structure.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# small example illustrating test procedures
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
```

```
colnames(X1)=paste("G",1:6,sep="")

X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")

# perform a test for differential connectivity of individual genes
# with PLS connectivity scores and squared distances
## Not run: tig=test.individual.genes(X1,X2)
## Not run: results.tig=get.results(tig)
## Not run: results.tig

# perform a test for differential connectivity of all genes
# with PLS connectivity scores and squared distances
## Not run: tcg=test.class.genes(X1,X2)
## Not run: results.tcg=get.results(tcg)
## Not run: results.tcg

# perform a test for modular structure using a minimum module size of 2
# and threshold of .5 with PLS connectivity scores
## Not run: test.modular.structure(X1,X2,min.module.size=2)
## Not run: results.tms=get.results(tms)
## Not run: results.tms
```

---

`HeavyMice` *Heavy Mice Dataset.*

---

## Description

This data set gives a subset of the microarray expression data from the liver tissue of heavy female mice. There were 3421 genes and 135 mice in the full data set; this data set was obtained by removing genes and mice with missing values. Then the 50 heaviest mice (weights greater than 40.5 g) were selected. Finally, univariate regressions of mouse weights on each individual gene were performed and 314 genes with z-scores greater than 5 were selected.

## Usage

```
data(HeavyMice)
```

## Format

A 50 by 314 matrix.

## References

Fuller, T., Ghazalpour, A., Aten, J., Drake, T., Lusis, A., and Horvath, S. (2007) Weighted gene coexpression network analysis strategies applied to mouse weight. *Mammalian Genome*, **28**, 463–472.

Ghazalpour, A., Doss, S., Zhang, B., Wang, S., Plaisier, C., Castellanos, R., Brozell, A., Schadt, E.E., Drake, T.A., Lusis, A.J., and Horvath, S. (2006) Integrated genetic and network analysis to characterize genes related to mouse weight. *PLoS Genetics*, **2**, 130.

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

---

`LeanMice` *Lean Mice Dataset.*

---

## Description

This data set gives a subset of the microarray expression data from the liver tissue of lean female mice. There were 3421 genes and 135 mice in the full data set; this data set was obtained by removing genes and mice with missing values. Then the 50 leanest mice (weights less than 36.9 g) were selected. Finally, univariate regressions of mouse weights on each individual gene were performed and 314 genes with z-scores greater than 5 were selected.

## Usage

```
data(LeanMice)
```

## Format

A 50 by 314 matrix.

## References

Fuller, T., Ghazalpour, A., Aten, J., Drake, T., Lusis, A., and Horvath, S. (2007) Weighted gene coexpression network analysis strategies applied to mouse weight. *Mammalian Genome*, **28**, 463–472.

Ghazalpour, A., Doss, S., Zhang, B., Wang, S., Plaisier, C., Castellanos, R., Brozell, A., Schadt, E.E., Drake, T.A., Lusis, A.J., and Horvath, S. (2006) Integrated genetic and network analysis to characterize genes related to mouse weight. *PLoS Genetics*, **2**, 130.

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

modules-class *Class* "modules"

## Description

The class of modules of a network.

## Objects from the Class

Objects can be created by calls of the form `new("modules",...)`.

## Slots

`module`**:** Object of class `"factor"` which represents the module number of the genes in a network. For genes that are in none of the modules, the module number is listed as 0.

## Methods

**show** `signature(object = "modules")`: lists the genes in each of the modules.

**summary** `signature(x = "modules")`: summarizes a network by listing the number of genes in each module.

**get.modules** `signature(object = "modules")`: returns the module number for each gene in a network. For the genes that are in none of the modules, the module number is listed as 0.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# artificial example to show how to obtain modules from a matrix of
# connectivity scores
set.seed(26)
s=matrix(runif(100,-1,1),10,10);diag(s)=1;s=round((s+t(s))/2,1)
the.modules=network.modules(s,m=3,epsilon=.7)
the.modules

# summary method useful for large networks
summary(the.modules)

# method for extracting the modules
get.modules(the.modules)

# plot a graph of the modules
## Not run: network.modules(s,m=3,epsilon=.7,plot=TRUE)
## Not run: tkplot.close('1')
```

`network.modules` *Determine modules for network*

## Description

Determine the modular structure for a network.

## Usage

```
network.modules(s,m,epsilon,plot=FALSE,interactive=FALSE,...)
```

## Arguments

| | |
|---|---|
| `s` | scores for a network. |
| `m` | minimum cluster size parameter. |
| `epsilon` | threshold parameter. |
| `plot` | indicates whether to create a graph for the network using the tkplot function in the igraph package. |
| `interactive` | indicates whether any plotted graphs should be interactive. |
| `...` | additional arguments passed to the tkplot function in the igraph package. |

## Value

| | |
|---|---|
| `modules` | an object of S4-class "modules" for the network |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# artificial example to show how to obtain modules from a matrix of
# connectivity scores
set.seed(26)
s=matrix(runif(100,-1,1),10,10);diag(s)=1;s=round((s+t(s))/2,1)
the.modules=network.modules(s,m=3,epsilon=.7)
the.modules
```

`pairOfNetworks-class` *Class* "pairOfNetworks"

## Description

The class of a pair of networks that can be used for differential network analysis.

## Objects from the Class

Objects can be created by calls of the form `new("pairOfNetworks",...)`.

## Slots

`network1`**:** Object of class "matrix"; the first network of expression values with genes in columns and subjects in rows.

`network2`**:** Object of class "matrix"; the second network of expression values with genes in columns and subjects in rows.

## Methods

**get.common.networks** `signature(object = "pairOfNetworks")`:

**get.network1** `signature(object = "pairOfNetworks")`: returns the first network from an object of class "pairOfNetworks".

**get.network2** `signature(object = "pairOfNetworks")`: returns the second network from an object of class "pairOfNetworks".

**show** `signature(object = "pairOfNetworks")`: lists the number of subjects and genes in each network as well as the number of genes common to both networks.

## Examples

```
# small example illustrating how a pair of networks is
# preprocessed to obtain and align a set of common genes
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")
X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")
networks=new("pairOfNetworks",network1=X1,network2=X2)
get.common.networks(networks)
```

```
# extract network 1
get.network1(networks)
# extract network 2
get.network2(networks)
```

---

PCnet *Principal Components network*

---

## Description

Computes the connectivity scores for a network based on principal components.

## Usage

```
PCnet(data,ncom=3,rescale.data=TRUE, symmetrize.scores=TRUE,
rescale.scores=FALSE)
```

## Arguments

| | |
|---|---|
| data | microarray dataset with genes in columns and samples in rows. |
| ncom | the number of PLS components (latent variables) in PLS models. |
| rescale.data | indicates whether data should be rescaled, |
| symmetrize.scores | indicates whether PLS scores should be made to be symmetric, |
| rescale.scores | indicates whether PLS scores should be rescaled so that the largest score for each gene should be 1 in magnitude, |

## Value

| | |
|---|---|
| PCnet | a matrix of interactions between gene pairs based on principal components regression. |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

Hastie, T., Tibshirani, R., and Friedman, J. (2009) *The Elements of Statistical Learning*. Springer: New York.

## Examples

```
# small example using PCnet with 3 principal components,
# data rescaled, and scores symmetrized but not rescaled
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
s=PCnet(X1)
print(round(s,4))

# small example using PCnet with 2 principal components,
# data rescaled, and scores symmetrized and rescaled
s2=PCnet(X1,ncom=2,rescale.data=TRUE,symmetrize.scores=TRUE,rescale.scores=TRUE)
print(round(s2,4))
```

---

PLSnet *Partial Least Squares network*

---

## Description

Computes the connectivity scores for a network based on partial least squares (PLS).

## Usage

```
PLSnet(data,ncom=3,rescale.data=TRUE,symmetrize.scores=TRUE,rescale.scores=FALSE)
```

## Arguments

| | |
|---|---|
| `data` | microarray dataset with genes in columns and samples in rows. |
| `ncom` | the number of PLS components (latent variables) in PLS models. |
| `rescale.data` | indicates whether data should be rescaled. |
| `symmetrize.scores` | indicates whether PLS scores should be made to be symmetric. |
| `rescale.scores` | indicates whether PLS scores should be rescaled so that the largest score for each gene should be 1 in magnitude. |

## Value

| | |
|---|---|
| `PLSnet` | a matrix of interactions between gene pairs based on partial least squares. |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

Pihur, V., Datta, S., and Datta, S. (2008) Reconstruction of genetic association networks from microarray data: a partial least squares approach. *Bioinformatics*, **24**(4), 561–568.

## Examples

```
# small example using PLSnet with 3 latent PLS components,
# data rescaled, and scores symmetrized but not rescaled
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
s=PLSnet(X1)
print(round(s,4))

# small example using PLSnet with 2 latent PLS components,
# data rescaled, and scores symmetrized and rescaled
s2=PLSnet(X1,ncom=2,rescale.data=TRUE,symmetrize.scores=TRUE,rescale.scores=TRUE)
print(round(s2,4))
```

---

# resultsClassTest-class

*Class* "resultsClassTest"

---

## Description

Tests whether the connectivity scores for a set of important genes differ between two networks.

## Objects from the Class

Objects can be created by calls of the form `new("resultsClassTest",...)`.

## Slots

`p.value`**:** Object of class "numeric"; the p-value for the significance test.

`delta`**:** Object of class "numeric"; the test statistic for the significance test.

`class.genes`**:** Object of class "character"; the list of important genes.

## Methods

**get.results** `signature(object = "resultsClassTest")`: returns the p-value, test statistic, and the class of genes for a test for differential connectivity of the class of genes.

**show** `signature(object = "resultsClassTest")`: summarizes the test by outputing the class of important genes, the value of the test statistic, and its p-value.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# small example illustrating test procedures
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")

X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")

# perform a test for differential connectivity of all genes
# with PLS connectivity scores and squared distances
## Not run: tcg=test.class.genes(X1,X2)
## Not run: results.tcg=get.results(tcg)
## Not run: results.tcg
```

---

# resultsIndTest-class *Class* "resultsIndTest"

---

## Description

Tests whether the connectivity scores for a single gene differ between two networks.

## Objects from the Class

Objects can be created by calls of the form `new("resultsIndTest",...)`.

## Slots

`p.values`**:** Object of class `"numeric"`; the p-values for the significance tests of all individual genes.

`d`**:** Object of class `"numeric"`; the test statistic for for all individual genes.

## Methods

**get.results** `signature(object = "resultsIndTest")`: returns the p-values and test statistics for tests for differential connectivity of individual genes.

**summary** `signature(x = "resultsIndTest")`: summarizes the tests for differential connectivity of individual genes by listing the number of genes which are significant at various significance levels.

**show** `signature(object = "resultsIndTest")`: summarizes the tests by outputing a data frame with the name, value of its test statistic, and p-value for up to the 20 most significant genes.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# small example illustrating test procedures
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")

X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")

# perform a test for differential connectivity of individual genes
# with PLS connectivity scores and squared distances
## Not run: tig=test.individual.genes(X1,X2)
## Not run: summary(tig)

# extract results for a test for differential connectivity of individual genes
## Not run: results.tig=get.results(tig)
## Not run: results.tig
```

---

# resultsModTest-class *Class* "resultsModTest"

---

## Description

Test whether the overall modular structure differs between the two networks.

## Objects from the Class

Objects can be created by calls of the form `new("resultsModTest",...)`.

## Slots

`p.value`**:** Object of class `"numeric"` ~~

`N`**:** Object of class `"numeric"` ~~

`modules1`**:** Object of class `"modules"` ~~

`modules2`**:** Object of class `"modules"` ~~

## Methods

**get.results** `signature(object = "resultsModTest")`: returns the p-value, test statistic, and the modules for each network for a test for overall modular structure.

**summary** `signature(x = "resultsModTest")`: summarizes the test for modular structure by summarizing the modules in each network and listing the test statistic and the p-value.

**show** `signature(object = "resultsModTest")`: summarizes the test for modular structure by summarizing the modules in each network and listing the test statistic and the p-value.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# small example illustrating test procedures
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")

X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")

# perform a test for modular structure using a minimum module size of 2
# and threshold of .5 with PLS connectivity scores
## Not run: test.modular.structure(X1,X2,min.module.size=2)
## Not run: summary(tms)

# extract results for a test of modular structure
## Not run: results.tms=get.results(tms)
## Not run: results.tms
```

RRnet *Ridge Regression network*

## Description

Computes the connectivity scores for a network based on ridge regression.

## Usage

```
RRnet(data,lambda=1,rescale.data=TRUE, symmetrize.scores=TRUE,
rescale.scores=FALSE)
```

## Arguments

| | |
|---|---|
| data | microarray dataset with genes in columns and samples in rows. |
| lambda | the ridge regression penalty parameter. |
| rescale.data | indicates whether data should be rescaled. |
| symmetrize.scores | indicates whether PLS scores should be made to be symmetric. |
| rescale.scores | indicates whether PLS scores should be rescaled so that the largest score for each gene should be 1 in magnitude. |

## Value

| | |
|---|---|
| RRnet | a matrix of interactions between gene pairs based on ridge regression. |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

Hastie, T., Tibshirani, R., and Friedman, J. (2009) *The Elements of Statistical Learning*. Springer: New York.

## Examples

```
# small example using RRnet with penalty parameter 1,
# data rescaled, and scores symmetrized but not rescaled
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
```

```
s=RRnet(X1)
print(round(s,4))

# small example using RRnet with penalty parameter 3,
# data rescaled, and scores symmetrized and rescaled
s2=RRnet(X1,lambda=3,rescale.data=TRUE,symmetrize.scores=TRUE,rescale.scores=TRUE)
print(round(s2,4))
```

---

`summary-methods` *Methods for Function* `summary`

---

## Description

summary-methods

## Methods

`signature(x = "modules")` summarizes a network by listing the number of genes in each module.

`signature(x = "resultsIndTest")` summarizes the tests for differential connectivity of individual genes by listing the number of genes which are significant at various significance levels.

`signature(x = "resultsModTest")` summarizes the test for modular structure by summarizing the modules in each network and listing the test statistic and the p-value.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# artificial example to show how to obtain a summary of the modules from a
# matrix of connectivity scores
set.seed(26)
s=matrix(runif(100,-1,1),10,10);diag(s)=1;s=round((s+t(s))/2,1)
the.modules=network.modules(s,m=3,epsilon=.7)
summary(the.modules)

# small example illustrating test procedures
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")

X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
```

```
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")

# perform a test for differential connectivity of individual genes
# with PLS connectivity scores and squared distances
## Not run: tig=test.individual.genes(X1,X2)
## Not run: summary(tig)

# perform a test for modular structure using a minimum module size of 2
# and threshold of .5 with PLS connectivity scores
## Not run: test.modular.structure(X1,X2,min.module.size=2)
## Not run: summary(tms)
```

---

`test.class.genes` *Test for differential connectivity of a class of genes*

---

## Description

Tests for differential connectivity of a class of genes between two networks using PLS scores.

## Usage

```
test.class.genes(X1,X2,genelist=NULL,scores="PLS",distance="abs",
num.permutations=1000,check.networks=TRUE,...)
```

## Arguments

| | |
|---|---|
| `X1` | network 1 with genes in columns and samples in rows. |
| `X2` | network 2 with genes in columns and samples in rows. |
| `genelist` | a list of the names of the subset of genes to be considered for testing. Alternately, a numerical vector of the indices for the genes can be supplied. |
| `scores` | type of connectivity score to be used. Either one of the built-in methods ("PLS", "PC", "RR", or "cor") can be used or a user-defined method can be supplied. |
| `distance` | distance function to be used. Either one of the built-in functions ("abs" or "sqr") can be used or a user-defined distance function can be supplied. |
| `num.permutations` | the number of random permutations. |
| `check.networks` | indicates whether get.common.networks is used to preprocess the networks before the test is performed. |
| `...` | additional arguments for scores or distance. |

## Value

| | |
|---|---|
| `results` | result of test (of class resultsClassTest). |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# small example illustrating test procedures
set.seed(12345)
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")

X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")

# perform a test for differential connectivity of all genes
# with PLS connectivity scores and squared distances
## Not run: tcg=test.class.genes(X1,X2)
## Not run: results.tcg=get.results(tcg)
## Not run: results.tcg

# perform a test for differential connectivity of a selected group of
# genes with PLS connectivity scores without rescaling the data,
# symmetrizing the scores, or rescaling the scores and with squared distances
# based on 10000 permutations
## Not run: our.genes=c(1,3)
## Not run: tcg2=test.class.genes(X1,X2,genelist=our.genes,scores="PLS",
## distance="sqr",num.permutations=10000,rescale.data=FALSE,
## symmetrize.scores=FALSE,rescale.scores=FALSE)
## Not run: tcg2
##
## or, equivalently
##
## Not run: our.genes=c("G2","G4")
## Not run: tcg2=test.class.genes(X1,X2,genelist=our.genes,scores="PLS",
## distance="abs",num.permutations=10000,rescale.data=FALSE,
## symmetrize.scores=FALSE,rescale.scores=FALSE)
## Not run: tcg2
```

```
# perform a test for differential connectivity of all genes
# with PLS connectivity scores and with custom distances
## Not run: our.dist=function(score1,score2){pmin(abs(score1-score2),1)}
## Not run: tcg3=test.class.genes(X1,X2,scores=PLSnet,distance=our.dist)
## Not run: tcg3

# perform a test for differential connectivity of all genes
# with correlation connectivity scores
## Not run: tcg4=test.class.genes(X1,X2,scores="cor")
## Not run: tcg4

# perform a test for differential connectivity of all genes
# with principal components regression connectivity scores
## Not run: tcg5=test.class.genes(X1,X2,scores="PC")
## Not run: tcg5

# perform a test for differential connectivity of individual genes
# with ridge regression connectivity scores with rescaled data,
# symmetrized and rescaled scores and a penalty parameter equal to 3
## Not run: tcg6=test.class.genes(X1,X2,scores="RR",rescale.scores=TRUE,
## lambda=3)
## Not run: tcg6

# perform a test for differential connectivity of individual genes
# with custom ridge regression connectivity scores with
# centered and rescaled data and symmetrized and rescaled scores
## Not run: ourRR=function(X,y,lambda=3){
## solve(t(X)%*%X+lambda*diag(ncol(X)))%*%t(X)%*%y}
## Not run: ourRRnet=function(X){gennet(X,f=ourRR,recenter.data=TRUE,
## rescale.data=TRUE,symmetrize.scores=TRUE,rescale.scores=TRUE)}
## Not run: tcg7=test.class.genes(X1,X2,scores=ourRRnet)
## Not run: tcg7
```

---

`test.individual.genes` *Test for differential connectivity of individual genes*

---

## Description

Tests for differential connectivity of individual genes between two networks using PLS scores.

## Usage

```
test.individual.genes(X1,X2,scores,distance="abs",
num.permutations=1000,check.networks=TRUE,...)
```

## Arguments

| | |
|---|---|
| X1 | network 1 with genes in columns and samples in rows. |
| X2 | network 2 with genes in columns and samples in rows. |

| | |
|---|---|
| scores | type of connectivity score to be used. Either one of the built-in methods ("PLS", "PC", "RR", or "cor") can be used or a user-defined method can be supplied. |
| distance | distance function to be used. Either one of the built-in functions ("abs" or "sqr") can be used or a user-defined distance function can be supplied. |
| num.permutations | the number of random permutations. |
| check.networks | indicates whether get.common.networks is used to preprocess the networks before the test is performed. |
| ... | additional arguments for scores or distance. |

## Value

| | |
|---|---|
| results | result of test (of class resultsIndTest). |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# small example illustrating test procedures
set.seed(12345)
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")

X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")

# perform a test for differential connectivity of individual genes
# with PLS connectivity scores and squared distances
## Not run: tig=test.individual.genes(X1,X2)
## Not run: summary(tig)

# extract results for a test for differential connectivity of individual genes
## Not run: results.tig=get.results(tig)
```

```
## Not run: results.tig

# perform a test for differential connectivity of individual genes
# with PLS connectivity scores without rescaling the data,
# symmetrizing the scores, or rescaling the scores and with squared distances
# based on 10000 permutations
## Not run: tig2=test.individual.genes(X1,X2,scores="PLS",distance="sqr",
## num.permutations=10000,rescale.data=FALSE,symmetrize.scores=FALSE,
## rescale.scores=FALSE)
## Not run: summary(tig2)

# perform a test for differential connectivity of individual genes
# with PLS connectivity scores and with custom distances
## Not run: our.dist=function(score1,score2){pmin(abs(score1-score2),1)}
## Not run: tig3=test.individual.genes(X1,X2,scores=PLSnet,distance=our.dist)
## Not run: summary(tig3)

# perform a test for differential connectivity of individual genes
# with correlation connectivity scores
## Not run: tig4=test.individual.genes(X1,X2,scores="cor")
## Not run: summary(tig4)

# perform a test for differential connectivity of individual genes
# with principal components regression connectivity scores
## Not run: tig5=test.individual.genes(X1,X2,scores="PC")
## Not run: summary(tig5)

# perform a test for differential connectivity of individual genes
# with ridge regression connectivity scores with rescaled data,
# symmetrized and rescaled scores and a penalty parameter equal to 3
## Not run: tig6=test.individual.genes(X1,X2,scores="RR",
## rescale.scores=TRUE,lambda=3)
## Not run: summary(tig6)

# perform a test for differential connectivity of individual genes
# with custom ridge regression connectivity scores with
# centered and rescaled data and symmetrized and rescaled scores
## Not run: ourRR=function(X,y,lambda=3){
## solve(t(X)%*%X+lambda*diag(ncol(X)))%*%t(X)%*%y}
## Not run: ourRRnet=function(X){gennet(X,f=ourRR,recenter.data=TRUE,
## rescale.data=TRUE,symmetrize.scores=TRUE,rescale.scores=TRUE)}
## Not run: tig7=test.individual.genes(X1,X2,scores=ourRRnet)
## Not run: summary(tig7)
```

---

`test.modular.structure` *Test for differential modular structures*

---

## Description

Tests for differential modular structures between two networks using PLS scores.

## Usage

```
test.modular.structure(X1,X2,scores="PLS",min.module.size=5,epsilon=.5,
num.permutations=1000,check.networks=TRUE,...)
```

## Arguments

| | |
|---|---|
| X1 | network 1 with genes in columns and samples in rows. |
| X2 | network 2 with genes as columns and samples in rows. |
| scores | type of connectivity score to be used. Either one of the built-in methods ("PLS", "PC", "RR", or "cor") can be used or a user-defined method can be supplied. |
| min.module.size | minimum module size parameter. |
| epsilon | threshold parameter. |
| num.permutations | the number of random permutations. |
| check.networks | indicates whether get.common.networks is used to preprocess the networks before the test is performed. |
| ... | additional arguments for scores. |

## Value

| | |
|---|---|
| results | result of test (of class resultsModTest). |

## Author(s)

The authors are Ryan Gill, Somnath Datta, and Susmita Datta. The software is maintained by Ryan Gill <rsgill01@louisville.edu>.

## References

Gill, R., Datta, S., and Datta, S. (2010) A statistical framework for differential network analysis from microarray data. *BMC Bioinformatics*, **11**, 95.

## Examples

```
# small example illustrating test procedures
set.seed(12345)
X1=rbind(
c(2.5,6.7,4.5,2.3,8.4,3.1),
c(1.2,0.7,4.0,9.1,6.6,7.1),
c(4.3,-1.2,7.5,3.8,1.0,9.3),
c(9.5,7.6,5.4,2.3,1.1,0.2))
colnames(X1)=paste("G",1:6,sep="")

X2=rbind(
c(4.5,2.4,6.8,5.6,4.5,1.2,4.5),
c(7.6,9.0,0.1,3.4,5.6,5.5,1.2),
c(8.3,4.5,7.0,1.2,4.3,3.7,6.8),
```

```
c(3.4,1.1,6.9,7.2,3.1,0.9,6.6),
c(3.4,2.2,1.3,5.5,9.8,6.7,0.6))
colnames(X2)=paste("G",8:2,sep="")

# perform a test for modular structure using a minimum module size of 2
# and threshold of .5 with PLS connectivity scores
## Not run: tms=test.modular.structure(X1,X2,min.module.size=2)
## Not run: summary(tms)

# extract results for a test of modular structure
## Not run: results.tms=get.results(tms)
## Not run: results.tms

# perform a test for modular structure using a minimum module size of 2
# and threshold of .5 with PLS connectivity scores without rescaling the data,
# symmetrizing the scores, or rescaling the scores based on 10000 permutations
## Not run: tms2=test.modular.structure(X1,X2,scores="PLS",min.module.size=2,
## num.permutations=10000,rescale.data=FALSE,symmetrize.scores=FALSE,
## rescale.scores=FALSE)
## Not run: summary(tms2)

# perform a test for modular structure using a minimum module size of 2
# and threshold of .5 with correlation connectivity scores
## Not run: tms3=test.modular.structure(X1,X2,scores="cor",min.module.size=2)
## Not run: summary(tms3)

# perform a test for modular structure using a minimum module size of 3
# and threshold of .7 with principal components regression connectivity scores
## Not run: tms4=test.modular.structure(X1,X2,scores="PC",min.module.size=3,
## epsilon=.7)
## Not run: summary(tms4)

# perform a test for modular structure using a minimum module size of 2
# and threshold of .9 with ridge regression connectivity scores with
# rescaled data, symmetrized and rescaled scores and a penalty parameter
# equal to 3
## Not run: tms5=test.modular.structure(X1,X2,scores="RR",min.module.size=2,
## epsilon=.5,rescale.scores=TRUE,lambda=3)
## Not run: summary(tms5)

# perform a test for modular structure using a minimum module size of 2 and
# threshold of .9 with custom ridge regression connectivity scores with
# centered and rescaled data and symmetrized and rescaled scores
## Not run: ourRR=function(X,y,lambda=3){
## solve(t(X)%*%X+lambda*diag(ncol(X)))%*%t(X)%*%y}
## Not run: ourRRnet=function(X){gennet(X,f=ourRR,recenter.data=TRUE,
## rescale.data=TRUE,symmetrize.scores=TRUE,rescale.scores=TRUE)}
## Not run: tms6=test.modular.structure(X1,X2,scores=ourRRnet,
## min.module.size=2,epsilon=.9)
## Not run: summary(tms6)
```

# Index


∗ **classes**
modules-class, 10
pairOfNetworks-class, 12
resultsClassTest-class, 15
resultsIndTest-class, 16
resultsModTest-class, 17
∗ **datasets**
HeavyMice, 8
LeanMice, 9
∗ **methods**
get.common.networks-methods, 4
get.modules-methods, 5
get.network1-methods, 6
get.network2-methods, 6
get.results-methods, 7
summary-methods, 20
∗ **modules**
get.common.networks-methods, 4
get.modules-methods, 5
get.network1-methods, 6
get.network2-methods, 6
∗ **tests**
get.results-methods, 7
resultsIndTest-class, 16
resultsModTest-class, 17
summary-methods, 20

cornet, 2

gennet, 3
get.common.networks (get.common.networks-methods), 4
get.common.networks,pairOfNetworks (get.common.networks-methods), 4
get.common.networks,pairOfNetworks-method (pairOfNetworks-class), 12
get.common.networks-methods, 4
get.common.networks-modules (get.common.networks-methods), 4
get.modules (get.modules-methods), 5
get.modules,modules (get.modules-methods), 5
get.modules,modules-method (modules-class), 10
get.modules-methods, 5
get.network1 (get.network1-methods), 6
get.network1,pairOfNetworks (get.network1-methods), 6
get.network1,pairOfNetworks-method (pairOfNetworks-class), 12
get.network1-methods, 6
get.network2 (get.network2-methods), 6
get.network2,pairOfNetworks (get.network2-methods), 6
get.network2,pairOfNetworks-method (pairOfNetworks-class), 12
get.network2-methods, 6
get.results (get.results-methods), 7
get.results,resultsClassTest (get.results-methods), 7
get.results,resultsClassTest-method (resultsClassTest-class), 15
get.results,resultsIndTest (get.results-methods), 7
get.results,resultsIndTest-method (resultsIndTest-class), 16
get.results,resultsModTest (get.results-methods), 7
get.results,resultsModTest-method (resultsModTest-class), 17
get.results-methods, 7

HeavyMice, 8

LeanMice, 9

modules-class, 10

network.modules, 11

pairOfNetworks-class, 12
PCnet, 13
PLSnet, 14

resultsClassTest-class, 15
resultsIndTest-class, 16
resultsModTest-class, 17
RRnet, 19

show,modules-method *(*modules-class*)*, 10
show,pairOfNetworks-method
*(*pairOfNetworks-class*)*, 12
show,resultsClassTest-method
*(*resultsClassTest-class*)*, 15
show,resultsIndTest-method
*(*resultsIndTest-class*)*, 16
show,resultsModTest-method
*(*resultsModTest-class*)*, 17
summary *(*summary-methods*)*, 20
summary,modules *(*summary-methods*)*, 20
summary,modules-method *(*modules-class*)*,
10
summary,resultsIndTest
*(*summary-methods*)*, 20
summary,resultsIndTest-method
*(*resultsIndTest-class*)*, 16
summary,resultsModTest
*(*summary-methods*)*, 20
summary,resultsModTest-method
*(*resultsModTest-class*)*, 17
summary-methods, 20

test.class.genes, 21
test.individual.genes, 23
test.modular.structure, 25