Package 'coexnet'

August 24, 2022

Type Package

Title coexnet: An R package to build CO-EXpression NETworks from Microarray Data

Version 1.19.1

Author Juan David Henao [aut,cre], Liliana Lopez-Kleine [aut], Andres Pinzon-Velasco [aut]

Maintainer Juan David Henao <judhenaosa@unal.edu.co>

Description Extracts the gene expression matrix from GEO DataSets (.CEL files) as a AffyBatch object. Additionally, can make the normalization process using two different methods (vsn and rma). The summarization (pass from multi-probe to one gene) uses two different criteria (Maximum value and Median of the samples expression data) and the process of gene differentially expressed analisys using two methods (sam and acde). The construction of the co-expression network can be conduced using two different methods, Pearson Correlation Coefficient (PCC) or Mutual Information (MI) and choosing a threshold value using a graph theory approach.

License LGPL

LazyData TRUE

Depends R (>= 3.6)

Imports affy, siggenes, GEOquery, vsn, igraph, acde, Biobase, limma, graphics, stats, utils, STRINGdb, SummarizedExperiment, minet, rmarkdown

Suggests RUnit, BiocGenerics, knitr

VignetteBuilder knitr

biocViews GeneExpression, Microarray, DifferentialExpression, GraphAndNetwork, NetworkInference, SystemsBiology, Normalization, Network

RoxygenNote 7.0.2

PackageStatus Deprecated

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

git_branch master

git_last_commit 091421d

git_last_commit_date 2022-05-10

Date/Publication 2022-08-24

R topics documented:

affy . . . 2
CCP . . . 3
cofVar . . . 4
createNet . . . 5
difExprs . . . 6
exprMat . . . 7
findThreshold . . . 9
geneSymbol . . . 10
getAffy . . . 11
getInfo . . . 12
info . . . 13
net1 . . . 13
net2 . . . 13
ppiNet . . . 14
sharedComponents . . . 15

Index **16**

`affy` *Greene-5P01NS017771-220003: Microarray expression experiment related with Parkinson disease.*

Description

AffyBatch object, has a part of raw expression values obtained from microarray chip.

Author(s)

Greene JG, 2006.

Source

GEO Dataset from NCBI, GEO accession: GSE4773.

CCP *Finding Common Connection Pattern between different networks.*

Description

From the intersection of two or more networks, it obtains the connected components by deleting the solitary nodes in the intersection network. They must be igraph objects.

Usage

```
CCP(..., by = NULL)
```

Arguments

...	The networks (igraph objects) to obtain the Common Connection Patterns.
by	It can be, a degree value or a range of values (c(min, max)), to defined the Common Connection Pattern. If you pass one value all the nodes above this degree will be used. By default is NULL, it calculates the CCPs using all the network.

Details

The Common Connection Pattern (CCP), is a new methodological proposal to identify molecular components linked together and common in several biological networks. The principal assumption behind Common Connection Pattern is that the networks to be compared must have the same molecular information from, i.e., must compare one layer of molecular abstraction at the same time, for example, co-expression layer, protein-protein layer, the gene regulation layer, among others.

For this, the intersection between biological networks is calculated whose result are the sub-networks with diameter greater than zero, being each of them considered as a Common Connection Pattern.

Value

An igraph object with all the Common Connection Pattern in the same network.

Author(s)

Juan David Henao <judhenaosa@unal.edu.co>

Examples

```
# Loading data

data("net1")
data("net2")

# Obtaining Common Connection Patterns

ccp <- CCP(net1,net2)
ccp
```

cofVar *Calculating the coefficient of variation for expression matrix.*

Description

This function calculates the mean and the coefficient of variation to each row (genes or probesets) in an expression matrix in two ways: i) in the whole matrix ii) for the specific phenotype (case or control).

Usage

```
cofVar(expData, complete = TRUE, treatment = NULL, type = NULL)
```

Arguments

expData	The whole normalized expression matrix, rows: genes or probeset, columns: samples, it may be stored in a SummarizedExperiment object.
complete	Boolean to define if the function uses the whole expression matrix, by default TRUE.
treatment	A numeric vector with 0s and 1s for each sample in the expression matrix, the 0 expresses the control samples and 1 expresses the case samples, by default is NULL.
type	It can be "case" to calculate the mean and the coefficient of variation for the case samples or, otherwise, "control" to obtain these two values for the control samples.

Value

The expression matrix with two new columns, the first one with the averages and the other one with the coefficient of variation values.

Author(s)

Juan David Henao <judhenaosa@unal.edu.co>

Examples

```
## Creating the expression matrix

# The matrix have 200 genes and 20 samples

n <- 200
m <- 20

# The vector with treatment and control samples

treat <- c(rep(0,10),rep(1,10))
```

```
# Calculating the expression values normalized

mat <- as.matrix(rexp(n, rate = 1))
norm <- t(apply(mat, 1, function(nm) rnorm(m, mean=nm, sd=1)))

## Calculating the mean and the coefficient of variation

# For the whole expression matrix

complete <- cofVar(norm)
head(complete)

# For the case samples

case <- cofVar(expData = norm,complete = FALSE,treatment = treat,type = "case")
head(case)
```

`createNet` *Creating a co-expression network from expression matrix.*

Description

From an expression matrix, this function creates a co-expression network like a graph object using a threshold value and one similarity function.

Usage

```
createNet(expData, method, threshold)
```

Arguments

`expData`	A whole expression matrix or differentially expressed genes matrix, it may be stored in a SummarizedExperiment object.
`method`	A function to calculate the similarity matrix between genes. It can be "correlation" to use Pearson function or "mutual information" to use a based on entropy information function.
`threshold`	A value between 0 and 1 to filter the similarity matrix and create the co-expression network.

Value

An undirected co-expression network as igraph object.

Author(s)

Juan David Henao Sanchez <judhenaosa@unal.edu.co>

Liliana Lopez Kleine <llopezk@unal.edu.co>

See Also

findThreshold to obtain a threshold value based on biology network assumptions.

Examples

```
# Loading data

pathfile <- system.file("extdata","expression_example.txt",package = "coexnet")
data <- read.table(pathfile,stringsAsFactors = FALSE)

# Building the network

cor_pearson <- createNet(expData = data,threshold = 0.7,method = "correlation")
cor_pearson

mut_inf <- createNet(expData = data,threshold = 0.5,method = "mutual information")
mut_inf
```

---

difExprs *Differential expression analysis using two different methods.*

Description

Using the expression matrix calculate the differential expressed genes to two class analysis and fixing an expected FDR value. The methods are SAM and ACDE.

Usage

```
difExprs(expData, treatment, fdr, DifferentialMethod, plotting = FALSE)
```

Arguments

expData	A matrix with the expression matrix, it may be stored in a SummarizedExperiment object.
treatment	A vector with the ientifiers of the classes, 0 to control and 1 to case.
fdr	The expected FDR value.
DifferentialMethod	The method to calculate the differential expressed genes, can be "sam" or "acde"
plotting	The option to show the result in a plot. By default FALSE.

Value

A data.frame with the expression matrix to the expressed diferential genes only.

Author(s)

Juan David Henao <judhenaosa@unal.edu.co>

References

Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. Proceedings of the National Academy of Sciences, 98(9), 5116-5121.

Acosta J and Lopez-Kleine L (2015). acde: Artificial Components Detection of Differentially Expressed Genes. R package version 1.4.0.

See Also

exprMat to obtain the expression matrix.

Examples

```
## Loading the expression matrix

treat <- c(rep(0,10),rep(1,10))
norm <- read.table(system.file("extdata","expression_example.txt",package = "coexnet"))

## Running the function using both approaches

sam <- difExprs(expData = norm,treatment = treat,fdr = 0.2,DifferentialMethod = "sam")
acde <- difExprs(expData = norm,treatment = treat,fdr = 0.2,DifferentialMethod = "acde")
```

---

exprMat *Calculate the expression matrix from the raw expression data.*

Description

This function use a affyBatch object with the raw expression data to normalize and transform the matrix from probeset to gene considering the option to remove the batch effect in the long microarray data.

Usage

```
exprMat(affy, genes, NormalizeMethod, SummaryMethod, BatchCorrect = TRUE)
```

Arguments

affy	A AffyBatch object with the raw expression data.
genes	A table with two columns, in the firt one the name of each probe in the microarray with header "probe" and in the second one the corresponding gene or ID with header "ID".
NormalizeMethod	The method to normalize the raw data. Can be "vsn" to apply Variance Stabilizing Normalization function or "rma" to apply Robust Multi-Array Average function.

SummaryMethod	The method to pass from probeset to genes or ID. Can be "max" to selecto the probeset with the most average expression value or "median" to obtain the median of each sample of the set of probeset corresponding to particular gen or ID.
BatchCorrect	The option to apply batch effect correction, by default TRUE.

Value

A SummarizedExperiment object with the expression matrix.

Author(s)

Juan David Henao <judhenaosa@unal.edu.co>

References

Huber, W., Von Heydebreck, A., Sultmann, H., Poustka, A., & Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. Bioinformatics, 18(suppl 1), S96-S104.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer Barclay, Y. D., Antonellis, K. J., Scherf, U., & Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics, 4(2), 249-264.

See Also

getAffy to obtain the affyBatch object.

geneSymbol to obtain the data.frame with probeset and genes/ID from .SOFT file.

Examples

```
## Not run:

# Loading data

if (require(affydata)) {
 data(Dilution)
}

# Loading table with probeset and gene/ID information

data(info)

# Calculating the expression matrix

## RMA

rma <- exprMat(affy = Dilution,genes = info,NormalizeMethod = "rma",
SummaryMethod = "median",BatchCorrect = FALSE)
head(rma)
```

```
## VSN

vsn <- exprMat(affy = Dilution,genes = info,NormalizeMethod = "vsn",
SummaryMethod = "median",BatchCorrect = FALSE)
head(vsn)


## End(Not run)
```

`findThreshold` *Find the threshold value to create a co-expression network*

Description

Finds the threshold value to establish the cutoff in the process to define the edges in the co-expression network final from two steps. In the first one, obtains the subtraction from clustering coefficient values of the real and random networks created from the possible threshold values in the correlation matrix. In the second one, a Kolmogorov-Smirnov test is made to evaluate the degree distribution respect normality.

Usage

```
findThreshold(expData, method, plotting = FALSE)
```

Arguments

`expData`	A whole expression matrix or the expression matrix to differentially expressed genes, it may be stored in a SummarizedExperiment object.
`method`	The method name to create the correlation matrix, this can be "correlation" to obtain the Pearson Correlation Coefficient. On the other hand, can be "mutual information" to obtain the correlation values from an entropy-based method.
`plotting`	The option to show the result in a plot. By default FALSE.

Value

The best threshold value found using the two criteria and a plot showing the result.

Author(s)

Juan David Henao Sanchez <judhenaosa@unal.edu.co>

Liliana Lopez Kleine <llopezk@unal.edu.co>

References

Elo, L. L., Jarvenpaa, H., Oresic, M., Lahesmaa, R., & Aittokallio, T. (2007). Systematic construction of gene coexpression networks with applications to human T helper cell differentiation process. Bioinformatics, 23(16), 2096-2103.

Leal, L. G., Lopez, C., & Lopez-Kleine, L. (2014). Construction and comparison of gene co-expression networks shows complex plant immune responses. PeerJ, 2, e610.

See Also

difExprs to find the differentially expressed genes matrix.

Examples

```
# Loading data

pathfile <- system.file("extdata","expression_example.txt",package = "coexnet")
data <- read.table(pathfile,stringsAsFactors = FALSE)

# Finding threshold value

cor_pearson <- findThreshold(expData = data,method = "correlation")
cor_pearson
```

---

geneSymbol *Create a table relating probesets with genes*

Description

From the information in the .soft file, creates a data.frame with two columns. In the first one, the probeset names. In the second one, the name of the corresponding genes.

Usage

```
geneSymbol(GPL, directory = ".")
```

Arguments

GPL	The GPL ID.
directory	The path file where the .soft file is. By default is the current path file.

Value

A data.frame with two columns, in the first one, the probesets and in the second one, the corresponding gene to each probeset.

Author(s)

Juan David Henao <judhenaosa@unal.edu.co>

Examples

```
# Creating the table with probesets and genes/IDs

gene_table <- geneSymbol(GPL = "GPL2025",directory = system.file("extdata",package = "coexnet"))

# Cleaning the NA information

gene_na <- na.omit(gene_table)

# Cleaning gene/ID information empty

final_table <- gene_na[gene_na$ID != "",]

head(final_table)
```

---

getAffy *Charge and create an AffyBatch object*

Description

Charges the data from a file with the GSM samples compressed using the "filelist.txt" file to identify the GSM ID and create the AffyBatch object finally.

Usage

```
getAffy(GSE, directory = ".")
```

Arguments

GSE	The name of the file with the compressed samples data.
directory	The path file where the samples and the "filelist" file are to create the AffyBatch object, by default is the current directory.

Value

An AffyBatch Object.

Author(s)

Juan david Henao <judhenaosa@unal.edu.co>

See Also

getInfo to download expression data from GEO DataSets included the "filelist" file.

Examples

```
# Creating the AffyBatch object from raw data downloaded
# The data is from partial experiment results from Greene JG(2006), GEO accession: GSE4773

# Data without CDF environment information

affy <- getAffy(GSE = "GSE1234",directory = system.file("extdata",package = "coexnet"))
```

`getInfo` *Download raw expression data from GEO DataSet*

Description

Using a GSE ID list, from the GEO DataSets FTP, it searches and downloads each raw expression data in a separate file with the GSE ID as the name. After, the function uncompresses the downloaded file to be ready to use. Additionally, searches and downloaded the information of the chip using the GPL ID to obtain the .soft file.

Usage

```
getInfo(GSE, GPL, directory = ".")
```

Arguments

`GSE`	The list of GSE ID.
`GPL`	The GPL ID of the chip used in the microarray experiment.
`directory`	The path file to download and uncompress the raw expression data, by default is the current path file.

Value

A set of files with the uncompressed data and ready to use.

Author(s)

Juan David Henao <judhenaosa@unal.edu.co>

See Also

`getAffy` to charge the expression data.

Examples

```
## Not run:
# Extract data from GEO DataSets (Takes time)

getInfo(GSE = "GSE8216", GPL = "GPL2025",directory = tempdir())

## End(Not run)
```

info — *Table with the relation between the probesets and the corresponding gene.*

Description

Table where the probesets and genes are related.

Author(s)

Greene JG, 2006.

Source

GEO Dataset from NCBI, GPL accession: GPL8300.

net1 — *Simulated network #1*

Description

Simulated network in an igraph object.

Author(s)

Henao, 2017

net2 — *Simulated network #2*

Description

Simulated network in an igraph object.

Author(s)

Henao, 2017

ppiNet *Create a protein-protein interaction network*

Description

Creates a protein-protein interaction network using an edge list with the relations between proteins or a vector with the gene symbols or any other molecular identifier type, widely used in biological databases, to create a predictive PPI network using information of evidence in STRING database.

Usage

```
ppiNet(
  molecularIDs = NULL,
  file = NULL,
  speciesID = 9606,
 evidence = c("neighborhood", "neighborhood_transferred", "fusion", "cooccurence",
    "homology", "coexpression", "coexpression_transferred", "experiments",
   "experiments_transferred", "database", "database_transferred", "textmining",
    "textmining_transferred", "combined_score")
)
```

Arguments

molecularIDs	A vector of IDs recognized by STRING database to create a PPI network from them.
file	A file with an edge list to charge the PPI network.
speciesID	The numerical ID from STRING database to the species of interest, by defect, is "9006" corresponding to human species.
evidence	A vector with the evidence to support the interactions between the proteins, by default is all the evidence given in STRING database.

Value

An igraph object as the protein-protein interaction network where the nodes are the molecular identifiers given in the input.

Author(s)

Juan David Henao <judhenaosa@unal.edu.co>

Examples

```
## Not run:
# Creating a vector with identifiers

ID <- c("FN1","HAMP","ILK","MIF","NME1","PROCR","RAC1","RBBP7",
"TMEM176A","TUBG1","UBC","VKORC1")
```

```
# Creating the PPI network

ppi <- ppiNet(molecularIDs = ID,evidence = c("neighborhood","coexpression","experiments"))
ppi

## End(Not run)

# Creating a PPI network from external data

ppi <- ppiNet(file = system.file("extdata","ppi.txt",package = "coexnet"))
ppi
```

---

sharedComponents *Finding shared components between diferent networks.*

Description

From the intersection network, obtains the nodes without any degree value. These are the shared molecular components without any biological relation among them.

Usage

```
sharedComponents(...)
```

Arguments

...	The networks (igraph objects) to obtain the shared components.

Value

A vector with the names of the shared components.

Author(s)

Juan David Henao <judhenaosa@unal.edu.co>

Examples

```
# Loading data

data("net1")
data("net2")

# Obtaining shared components

share <- sharedComponents(net1,net2)
share
```

Index

affy, 2

CCP, 3
cofVar, 4
createNet, 5

difExprs, 6, *10*

exprMat, *7*, 7

findThreshold, *6*, 9

geneSymbol, *8*, 10
getAffy, *8*, 11, *12*
getInfo, *11*, 12

info, 13

net1, 13
net2, 13

ppiNet, 14

sharedComponents, 15