Package 'LINC'

October 16, 2018

Type Package

Title co-expression of lincRNAs and protein-coding genes

Version 1.8.0

Date 2017-03-18

Author Manuel Goepferich, Carl Herrmann

Maintainer Manuel Goepferich <manuel.goepferich@gmx.de>

Depends R (>= 3.3.1), methods, stats

Description This package provides methods to compute co-expression networks of lincRNAs and protein-coding genes.
Biological terms associated with the sets of protein-coding genes predict the biological contexts of lincRNAs according to the 'Guilty by Association' approach.

biocViews Software, BiologicalQuestion, GeneRegulation, GeneExpression

License Artistic-2.0

LinkingTo Rcpp

Imports Rcpp (>= 0.11.0), DOSE, ggtree, gridExtra, ape, grid, png, Biobase, sva, reshape2, utils, grDevices, org.Hs.eg.db, clusterProfiler, ggplot2, ReactomePA

VignetteBuilder knitr

Suggests RUnit, BiocGenerics, knitr, biomaRt

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

git_branch RELEASE_3_7

git_last_commit fb436a1

git_last_commit_date 2018-04-30

Date/Publication 2018-10-15

R topics documented:

`Arith-methods` *Plus Operator '+' In* **LINC**

Description

`plotlinc` is the generic function that creates plots based on objects of the `LINC` class.

Methods

`signature(e1 = "LINCbio", e2 = "LINCfeature")` (see `feature()`)

`signature(e1 = "LINCcluster", e2 = "LINCfeature")` (see `feature()`)

`signature(input = "LINCmatrix", showCor = "LINCfeature")` (see `feature()`)

Examples

```
data(BRAIN_EXPR)

# add a custom name
crbl_cluster_feat <- crbl_cluster + feature(customID = "CEREBELLUM")
plotlinc(crbl_cluster_feat)
```

assignment-methods *Methods for the Getter Function* assignment *in Package* **LINC**

Description

Access the slot assignment of an object of class LINC

Arguments

x	a 'LINC' object, for instance LINCmatrix

Value

the respective substructure or information

Methods

signature(x = "LINCbio") assignment slot

signature(x = "LINCcluster") assignment slot

signature(x = "LINCmatrix") assignment slot

signature(x = "LINCsingle") assignment slot

Author(s)

Manuel Goepferich

Examples

```
data(BRAIN_EXPR)

assignment(crbl_cluster)
```

BRAIN_EXPR *mRNA Expression Of Normal BRAIN From GTEX And TCGA*

Description

These datasets represent sample mRNA expression matrices of brain tissue with normal tissue from the Genotype-Tissue Expression (GTEx) platform (ctx = Cortex, crbl = Cerebellum, phc = Hippocampus) and cancer tissue from The Cancer Genome Atlas (TCGA) platform (gbm = Glioblastoma).

Usage

```
data(BRAIN_EXPR)
```

Format

Gene expression matrices and objects of the LINC class.

Value

Rows represent genes and columns samples. Gene names are given as Entrez identifiers. For the normal tissue from GTEx, expression levels are in units of [FPKM], for cancer tissue from TCGA, the unit [RESM] is used to represent expression levels.

Source

http://www.gtexportal.org Genotype-Tissue Expression (GTEx)

https://tcga-data.nci.nih.gov/docs The Cancer Genome Atlas (TCGA)

References

Carithers et al. Biopreservation and Biobanking. October 2015, 13(5): 311-319. doi:10.1089/bio.2015.0032. PMID: 26484571.

Examples

```
data(BRAIN_EXPR)
```

---

changeOrgDb *Change the Gene Annotation / Model Organism*

Description

The standard gene annotation in **LINC** is `"org.Hs.eg.db"`. This function is only relevant in case the input gene expression matrix is from another organism than Homo Sapiens.

Usage

```
changeOrgDb(object, OrgDb)
```

Arguments

`object`	an instance of the `LINC` class
`OrgDb`	has to be one of: `c("org.Ag.eg.db", "org.At.tair.db",`

Value

object with changed gene annotation hook

Note

Please, also consider the documentation of **clusterProfiler** and **ReactomePA**.

Author(s)

Manuel Goepferich

Examples

```
data(BRAIN_EXPR)

# change the used gene annotation, here from "human" to "mouse"
murine_matrix <- changeOrgDb(crbl_matrix, OrgDb = 'org.Mm.eg.db')
```

---

clusterlinc-methods *Cluster Queried ncRNAs Based On Their Interaction Partners*

Description

The function `clusterlinc` will give an overview of ncRNAs in a dataset. An input `LINCmatrix` will be converted to a `LINCcluster`. The following steps are conducted (I) computation of a correlation test, (II) setup of a distance matrix, (III) calculation of a dendrogram and (IV) selection of co-expressed genes for each query. The result is a cluster of ncRNAs and their associated protein-coding genes.

Usage

```
clusterlinc(linc,
            distMethod    = "dicedist",
            clustMethod   = "average",
            pvalCutOff    = 0.05,
            padjust       = "none",
            coExprCut     = NULL,
            cddCutOff     = 0.05,
            verbose       = TRUE)
```

Arguments

`linc`	an object of the class `LINCmatrix`
`distMethod`	a method to compute the distance between ncRNAs; has to be one of `c("correlation", "pvalue",`
`clustMethod`	an algorithm to compute the dendrogram, has to be one of `c("ward.D", "ward.D2",`
`pvalCutOff`	a threshold for the selection of co-expressed genes. Only protein-coding genes showing a significance in the correlation test lower than `pvalCutOff` will assigned to queried ncRNAs as interaction partner.
`padjust`	one of `c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")`
`coExprCut`	a single `integer` indicating the number of co-expressed genes to select. If this argument is used for each ncRNA the `coExprCut = n` protein-coding genes with the lowest p-value in the correlation test will be assigned to queries, respectively.
`cddCutOff`	a threshold that is only relevant for `distMethod = "dicedist"`. In this method `cddCutOff` defines whether a ncRNA and a protein-coding gene can be considered as interaction partners. This influences the distance matrix and the clustering process.
`verbose`	whether to give messages about the progression of the function `TRUE` or not `FALSE`

Details

As a first step clusterlinc conducts the correlation test (stats::cor.test) using the correlation method and handeling of missing values inherited from the input LINCmatrix. Resulting p-values indicate the statistical robustness of correlations instead of absolute correlation values. Co-expression of ncRNAs to protein-coding genes is assumed if the p-value from the cor.test is lower than the given pvalCutOff. An alternative way to select co-expressed genes is provided by coExprCut. This argument has priority over pvalCutOff and can be used to pick the n genes with the lowest p-value for each ncRNA. In contrast to pvalCutOff, this will result in an equal number of assigned co-expressed genes. The argument padjust can be used for multiple testing correction. In most cases this is not compatible with distMethod = "dicedist".

For the computation of the distance matrix of ncRNA genes three methods can be applied. The first method "correlation" uses 1 - correlation as distance measure. In contrast, "pvalue" considers not the absolute correlation values, but p-values from the correlation test. A third method is termed "dicedist" and takes the Czekanovski dice distance [1] as distance measure. Here, the number of shared interaction partners between ncRNAs determines their relation to each other. The argument cddCutOff is an option to decide which p-values in the correlation matrix can be considered as interaction. A low threshold, for instance, will consider only interactions of ncRNAs and protein-coding genes supported by a p-value lower than the supplied threshold and therfore a robust correlation of these two genes. Based on the distance matrix a cluster of the ncRNAs will be computed by stats::hclust. Argument clustMethod defines which clustering method should be applied.

A LINCcluster can be recalculated with the command clusterlinc(LINCcluster, ...)) in order to change further arguments. plotlinc(LINCcluster, ...)) will plot a figure that shows the cluster of ncRNAs (dendrogram) and the number of co-expressed genes with respect to different thresholds. getbio(LINCcluster, ...)) will derive the biological terms associated with the co-expressed genes. Due to the correlation test longer calculation times can occur. A faster alternative to this function is singlelinc(). User-defined correlation test functions are supported for singlelinc() but not for clusterlinc().

Value

an object of the class 'LINCmatrix' (S4) with 6 Slots

results	a list containing an object of the class "phylo" with the additional entry neighbours, a list of queries and co-expressed genes
assignment	a character vector of protein-coding genes
correlation	a list of cormatrix, the correlation of non-coding to protein-coding genes, lnctolnc, the correlation of non-coding to non-coding genes and cortest, p-values of the correlation test of non-coding to protein-coding genes
expression	the original expression matrix
history	a storage environment of important methods, objects and parameters used to create the object
linCenvir	a storage environment ensuring the compatibility to other objects of the LINC class

Methods

signature(linc = "LINCcluster") (see details)

signature(linc = "LINCmatrix") (see details)

Compatibility

`plotlinc(LINCcluster, ...)`, `getbio(LINCcluster, ...)`, ...

Author(s)

Manuel Goepferich

References

[1] Christine Brun, Francois Chevenet, David Martin, Jerome Wojcik, Alain Guenoche and Bernard Jacq" Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network" (2003) Genome Biology, 5:R6.

See Also

`linc` ; `singlelinc`

Examples

```
data(BRAIN_EXPR)
class(crbl_matrix)

# call 'clusterlinc' with no further arguments
crbl_cluster <- clusterlinc(crbl_matrix)

# apply the distance method "correlation instead of "dicedist"
crbl_cluster_cor <- clusterlinc(crbl_matrix, distMethod = "correlation" )
# do the same as recursive call using the 'LINCcluster' object
# crbl_cluster_cor <- clusterlinc(crbl_cluster, distMethod = "correlation")

# select 25 genes with lowest p-values for each query
crbl_cluster_25 <- clusterlinc(crbl_matrix, coExprCut = 25)

# select onyl those with a p-value < 5e-5
crbl_cluster_5e5 <- clusterlinc(crbl_matrix, pvalCutOff = 5e-5)

# adjust for multiple testing
crbl_cluster_hochberg <- clusterlinc(crbl_matrix, distMethod = "correlation",
                                     padjust = "hochberg", pvalCutOff = 0.05)

# comparing two distance methods
 plotlinc(crbl_cluster)
 plotlinc(crbl_cluster_cor)
```

`correlation-methods` *Methods for the Getter Function* `correlation` *in Package* **LINC**

Description

Access the slot `correlation` of an object of class `LINC`

Arguments

`x`	a `'LINC'` object, for instance `LINCmatrix`

Value

the respective substructure or information

Methods

`signature(x = "LINCbio")` correlation slot
`signature(x = "LINCcluster")` correlation slot
`signature(x = "LINCmatrix")` correlation slot
`signature(x = "LINCsingle")` correlation slot

Author(s)

Manuel Goepferich

Examples

```
data(BRAIN_EXPR)

correlation(crbl_cluster)
```

`express-methods` *Methods for the Getter Function* `express` *in Package* **LINC**

Description

Access the slot `expression` of an object of class `LINC`

Arguments

`x`	a `'LINC'` object, for instance `LINCmatrix`

Value

the respective substructure or information

Methods

`signature(x = "LINCbio")` expression slot
`signature(x = "LINCcluster")` expression slot
`signature(x = "LINCmatrix")` expression slot
`signature(x = "LINCsingle")` expression slot

Author(s)

Manuel Goepferich

Examples

```
data(BRAIN_EXPR)

express(crbl_cluster)
```

feature *Manipulate Objects Of The 'LINC' class*

Description

feature provides useful options intended to be used with 'LINC' objects.

Usage

```
feature(setLevel   = NULL,
        customID   = NULL,
        customCol  = "black",
        showLevels = FALSE)
```

Arguments

setLevel	a character string of the class the object should be converted into
customID	a character string of a name for the object
customCol	a character string of a valid colour for the object
showLevels	whether to show the inherited classes of the object TRUE or not FALSE

Details

Custom ids and colours enable the identification of a particular object in plots created by plotlinc and querycluster. With setLevel it is possible to change the class of an object.

feature works in combination with the plus operator: 'object' + feature(customID = ..., customCol = ...)

Value

an object of the class 'LINCfeature' (S4) with 5 Slots (not shown)

Author(s)

Manuel Goepferich

See Also

querycluster ; strlinc

Examples

```
data(BRAIN_EXPR)

# add a custom name
crbl_cluster_feat <- crbl_cluster + feature(customID = "CEREBELLUM")
 plotlinc(crbl_cluster_feat)
```

getbio-methods *Find Enriched Biological Terms For A Cluster*

Description

The function provides an interface to the clusterProfiler package. For each query in a cluster it seeks the biological terms that can be associated with the co-expressed genes, respectively. The input for getbio is of the class 'LINCcluster'.

Usage

```
getbio(cluster,
       enrichFun   = 'enrichGO',
       ont = "BP",
       ...)
```

Arguments

cluster	a 'LINCcluster'. The number of co-expressed genes has to be sufficient.
enrichFun	a function given as character string which will derive significant biological terms based on the set of co-expressed genes from a gene annotation resource. Supported functions are: c("enrichGO", "enrichPathway", "enrichDO")
ont	a subontology, only used for enrichFun = 'enrichGO'. This has to be one of "MF", "BP", "CC".
...	further arguments, mainly for functions from clusterProfiler

Details

In contrast to the function singlelinc here, a group of queries, those present in the input cluster, will be analyzed for enriched biological terms. The annotation function can be one of c("enrichGO", "enrichPathway"
[1] The gene system of the input object has to be translated for the enrichment function in case genes are not given as Entrez ids. The function clusterProfiler:bitr [2] will be used in order to translate gene ids.

Value

an object of the class 'LINCmatrix' (S4) with 6 Slots

results	a list containing the identified enriched biological terms plus their respective p-values
assignment	a character vector of protein-coding genes
correlation	a list of cormatrix, the correlation of non-coding to protein-coding genes and lnctolnc, the correlation of non-coding to non-coding genes
expression	the original expression matrix
history	a storage environment of important methods, objects and parameters used to create the object
linCenvir	a storage environment ensuring the compatibility to other objects of the LINC class

Methods

signature(cluster = "LINCcluster") (see details)

Author(s)

Manuel Goepferich

References

[1] Yu G, Wang L, Han Y and He Q (2012). "clusterProfiler: an R package for comparing biological themes among gene clusters." OMICS: A Journal of Integrative Biology, 16(5), pp. 284-287. (https://www.bioconductor.org/packages/release/bioc/html/clusterProfiler.html)

See Also

clusterlinc ; singlelinc ;

Examples

```
data(BRAIN_EXPR)

## Find the enriched cellular components for each query in the cluster
crbl_cc <- getbio(crbl_cluster, ont = "CC")
 plotlinc(crbl_cc)
```

---

getcoexpr *Get IDs For Co-Expressed Genes from The 'LINC' Class*

Description

getcoexpr provides access to co-expressed genes of a query in 'LINC' objects.

Usage

```
getcoexpr(input,
          query   = NULL,
          keyType = NULL)
```

Arguments

input	a 'LINCcluster' or 'LINCsingle' object
query	for a 'LINCcluster' the name of the query gene
keyType	The 'keyType' (gene system) for the output.

Value

a vector containing the co-expressed genes for a query

Author(s)

Manuel Goepferich

Examples

```
data(BRAIN_EXPR)
# Get the co-expressed genes for the query gene '55384' alias MEG3
getcoexpr(crbl_cluster, query = "55384")

# The co-expressed genes can also be returned as gene symbols.
getcoexpr(crbl_cluster, query = "55384", keyType = 'SYMBOL')
```

---

getlinc-methods *Subsetting for* LINC *objects*

Description

getlinc is a function to derive substructures from LINC objects.

Usage

```
getlinc(input,
        subject = "queries")
```

Arguments

input	a 'LINC' object, for instance LINCmatrix
subject	has to be one of c("queries", "geneSystem", "results", "history", "customID")

Value

the respective substructure or information

Methods

signature(input = "ANY", subject = "character") (see details)

Author(s)

Manuel Goepferich

See Also

linc ;

Examples

```
data(BRAIN_EXPR)

# getlinc() is used to accesss information
getlinc(crbl_cluster, subject = "geneSystem")
getlinc(crbl_cluster, subject = "queries")
getlinc(crbl_cluster, subject = "customID")
```

history-methods *Methods for the Getter Function* history *in Package* **LINC**

Description

Access the slot history of an object of class LINC

Arguments

x	a 'LINC' object, for instance LINCmatrix

Value

the respective substructure or information

Methods

signature(x = "LINCbio") history slot

signature(x = "LINCcluster") history slot

signature(x = "LINCmatrix") history slot

signature(x = "LINCsingle") history slot

Author(s)

Manuel Goepferich

Examples

```
data(BRAIN_EXPR)

history(crbl_cluster)
```

justlinc-methods *Co-Expression Analysis Of ncRNA Genes In One Step*

Description

justlinc is a wrapper that uses different functions of the **LINC** package. It applies fixed thresholds for gene selection and requires only an (unprocessed) expression matrix as input. This enables a fast co-expression analysis with or without a list candidates.

Usage

```
justlinc(object,
    targetGenes = "lincRNA",
           rmPC = TRUE)
```

Arguments

object	a gene expresssion matrix with Ensembl gene ids (rows correspond to genes)
targetGenes	the gene biotype like "lincRNA" or a vector of querie(s) - one or multiple genes ids
rmPC	remove 30 percent of the variance in the data by PCA TRUE or not FALSE.

Details

This function was built for expression matrices which uses the Ensembl gene system with gene expression values for over 50.000 genes. The input will be matched with a static gene annotation. Genes will be selected for median expression and variance. The final correlation matrix of protein-coding genes versus the target genes (lincRNAs) considers 5000 protein-coding genes and 500 target genes. Enriched pathways will be computed for the co-expressed genes showing the highest Spearman's rank correlation. In case `targetGenes = "lincRNA"`, then the function will search for the 10 best lincRNAs. Supplying a vector of gene ids the method will determine the best co-expressed genes for the given queries. Importanly, this function is only a wrapper. Thresholds can be changed using `linc` and related functions.

Value

depending on the input in `targetGenes` one or two plots and the result of the co-expression analysis

Methods

`signature(object = "matrix")` (see details)

Author(s)

Manuel Goepferich

See Also

`linc`

Examples

```
# NOT RUN:

# large gene expression matrix not avaiable in this version

# data(LIVER_EXPR)
# a gene expression matrix with > 50,000 genes
# str(GTEX_LIVER_CRUDE)

# 'justlinc' will search for the 10 best candidates
try(justlinc(GTEX_LIVER_CRUDE), silent = TRUE)

# 'justlinc' called with queries
my_lincRNAs <- c("ENSG00000224153", "ENSG00000197813",
                 "ENSG00000179136", "ENSG00000259439",
                 "ENSG00000267462")

try(justlinc(GTEX_LIVER_CRUDE, targetGenes = my_lincRNAs), silent = TRUE)
```

linc-methods *Compute A Correlation Matrix of Co-expressed Coding And Non-Coding Genes*

Description

The function `linc` can be considered as the main function of this package. It converts a given input object into a `LINCmatrix`. This process includes (I) statistical analysis and (II) correction of the input, (III) separation of coding and non-coding genes and (IV) computation of a correlation matrix.The input could be for instance a gene expression matrix. Rows correspond to genes; columns represent samples.Besides a suitable `object` a vector identifying the protein-coding genes is required.

Usage

```
linc(object,
     codingGenes,
     corMethod    = "spearman",
     batchGroups,
     nsv          = 1,
     rmPC,
     outlier,
     userFun,
     verbose      = TRUE)
```

Arguments

`object`	a `matrix`, `data.frame` or `ExpressionSet` with genes corresponding to rows, preferentially the high-variance genes in a given set
`codingGenes`	a `logical` vector with the same length of the supplied genes in `object`. `TRUE` indicates that the gene is a protein-coding one. Alternatively, `codingGenes` can be a vector of gene biotypes.
`corMethod`	a method for the correlation function; has to be one of `c("pearson", "kendall", "spearman", "us`
`batchGroups`	a vector naming the batch conditions. The length of this vector has to match the number of samples supplied in `object`. There has to be at least two different batch conditions for the method to work.
`nsv`	a single `integer` indicating the number of hidden surrogate variables. This argument is only relevant in case `batchGroups` is used.
`rmPC`	a vector of principle components (PCs) which should be removed. PCs are counted staring from 1 up to the maximal count of samples.
`outlier`	a method for the genewise removal of single outliers; has to be one of `c("esd", "zscore")`
`userFun`	a function or its name that should be used to calculate the correlation between coding and non-coding genes. This argument has to be used in combination with `corMethod = "user"`
`verbose`	whether to give messages about the progression of the function `TRUE` or not `FALSE`

Details

object can be a matrix, a data.frame or an ExpressionSet with rows corresponding to genes and columns to samples, the assumed co-expression conditions. Genes with duplicated names, genes having 0 variance plus genes with to many missing or infinite values will be removed from the input. For inputs showing a high inter-sample variance (ANOVA) in combination with many single outliers a warning message will appear. By default Spearman's rank correlation will be computed between protein-coding to non-coding genes. For this method a time-efficient C++ implementation will be called. Longer computation times occur for genes > 5000 and samples > 100. Missing values are handled in a manner that only pairwise complete observations will be compared. A customized correlation function can be applied supplying the function in userFun and requires the formal arguments x and x. This has priority over corMethod.

A number of statistical methods are available in order to remove effects from a given input expression matrix which depend on the used platform or technology and may hide relevant biology. The argument batchGroups works as a rapper of the SVA package calling sva::svaseq. The number of hidden surrogate variables is set to nsv = 1 by default; it can be estimated utilizing the function sva::num.sv. For this model to work the description of at least two different batches are required in batchGroups. Principle Component Analysis (PCA) can be performed by rmPC = c(...) where ... represents a vector of principle components. The command rmPC = c(2:ncol(object)) will remove the first PC from the input. This method can be used to determine whether observations are due to the main variance in the dataset i.e. main groups or subtypes. Outliers are handled genewise. The extreme Studentized deviate (ESD) test by Rosner, Bernard (1983) will detect one up to four outliers in a gene and replace them by NA values. The alternative zscore will perform a robust zscore test suggested by Boris Iglewicz and David Hoaglin (1993) and detect a single outlier in a gene if |Z| > 3.5.

A LINCmatrix can be recalculated with the command linc(LINCmatrix, ...)) in order to change further arguments. plotlinc(LINCmatrix, ...)) will plot a figure depicting the statistical analysis and correlation values. As for most objects of the LINC class manipulation of the last slot linCenvir will likely result in unexpected errors.

Value

an object of the class 'LINCmatrix' (S4) with 6 Slots

results	a list containing the original input expression matrix or a transfomed matrix if rmPC, batchGroups or outlier was applied
assignment	a character vector of protein-coding genes
correlation	a list of $cormatrix, the correlation of non-coding to protein-coding genes and $lnctolnc, the correlation of non-coding to non-coding genes
expression	the original expression matrix
history	a storage environment of important methods, objects and parameters used to create the object
linCenvir	a storage environment ensuring the compatibility to other objects of the LINC class

Methods

signature(object = "data.frame", codingGenes = "ANY") (see details)

signature(object = "ExpressionSet", codingGenes = "ANY") (see details)

signature(object = "LINCmatrix", codingGenes = "missing") (see details)

signature(object = "matrix", codingGenes = "ANY") (see details)

Compatibility

plotlinc(LINCmatrix, ...), clusterlinc(LINCmatrix, ...), singlelinc(LINCmatrix, ...), ...

Author(s)

Manuel Goepferich

References

[1] https://www.bioconductor.org/packages/release/bioc/html/sva.html

[2] Rosner, Bernard (May 1983), Percentage Points for a Generalized ESD Many-Outlier Procedure,Technometrics, 25(2), pp. 165-172.

[3] Boris Iglewicz and David Hoaglin (1993), Volume 16:How to Detect and Handle Outliers", The ASQC Basic References in Quality Control: Statistical Techniques, Edward F. Mykytka, Ph.D., Editor.

See Also

justlinc ; clusterlinc ; singlelinc

Examples

```
data(BRAIN_EXPR)

# call 'linc' with no further arguments
crbl_matrix <- linc(cerebellum, codingGenes = pcgenes_crbl)

# remove first seven principle components
crbl_matrix_pc <- linc(cerebellum, codingGenes = pcgenes_crbl, rmPC = c(1:7))

# negative correlation by using 'userFun'
crbl_matrix_ncor <- linc(cerebellum, codingGenes = pcgenes_crbl,
                         userFun = function(x,y){ -cor(x,y) })

# remove outliers using the ESD method
crbl_matrix_esd <- linc(cerebellum, codingGenes = pcgenes_crbl, outlier = "esd")

# plot this object
plotlinc(crbl_matrix_esd)
```

LINCbio-class *Class* "LINCbio"

Description

"LINCbio"

Objects from the Class

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

Slots

`results`: Object of class `"list"` ~~

`assignment`: Object of class `"vector"` ~~

`correlation`: Object of class `"list"` ~~

`expression`: Object of class `"matrix"` ~~

`history`: Object of class `"environment"` ~~

`linCenvir`: Object of class `"environment"` ~~

Extends

Class `"LINCmatrix"`, directly.

Methods

`+ signature(e1 = "LINCbio", e2 = "LINCfeature")`: ...

linctable `signature(file_name = "character", input = "LINCbio")`: ...

overlaylinc `signature(input1 = "LINCbio", input2 = "LINCbio")`: ...

plotlinc `signature(input = "LINCbio", showCor = "character")`: ...

plotlinc `signature(input = "LINCbio", showCor = "missing")`: ...

Examples

```
showClass("LINCbio")
```

LINCcluster-class *Class* `"LINCcluster"`

Description

code"LINCcluster"

Objects from the Class

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

Slots

`results`: Object of class `"list"` ~~

`assignment`: Object of class `"vector"` ~~

`correlation`: Object of class `"list"` ~~

`expression`: Object of class `"matrix"` ~~

`history`: Object of class `"environment"` ~~

`linCenvir`: Object of class `"environment"` ~~

Extends

Class `"LINCmatrix"`, directly.

Methods

+ `signature(e1 = "LINCcluster", e2 = "LINCfeature")`: ...

clusterlinc `signature(linc = "LINCcluster")`: ...

getbio `signature(cluster = "LINCcluster")`: ...

linctable `signature(file_name = "character", input = "LINCcluster")`: ...

plotlinc `signature(input = "LINCcluster", showCor = "character")`: ...

plotlinc `signature(input = "LINCcluster", showCor = "missing")`: ...

Examples

```
showClass("LINCcluster")
```

---

linCenvir-methods *Methods for the Getter Function* linCenvir *in Package* **LINC**

Description

Access the slot `linCenvir` of an object of class `LINC`

Arguments

`x`	a `'LINC'` object, for instance `LINCmatrix`

Value

the respective substructure or information

Methods

`signature(x = "LINCbio")` linCenvir slot

`signature(x = "LINCcluster")` linCenvir slot

`signature(x = "LINCmatrix")` linCenvir slot

`signature(x = "LINCsingle")` linCenvir slot

Author(s)

Manuel Goepferich

Examples

```
data(BRAIN_EXPR)

linCenvir(crbl_cluster)
```

LINCfeature-class *Class* "LINCfeature"

Description

"LINCfeature"

Objects from the Class

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

Slots

customID: Object of class "character" ~~

customCol: Object of class "character" ~~

setLevel: Object of class "character" ~~

showLevels: Object of class "logical" ~~

Methods

+ signature(e1 = "LINCbio", e2 = "LINCfeature"): ...

+ signature(e1 = "LINCcluster", e2 = "LINCfeature"): ...

+ signature(e1 = "LINCmatrix", e2 = "LINCfeature"): ...

Examples

```
showClass("LINCfeature")
```

LINCmatrix-class *Class* "LINCmatrix"

Description

an object of the class 'LINCmatrix' (S4) with 6 Slots

Objects from the Class

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

Slots

results: Object of class "list" ~~

assignment: Object of class "vector" ~~

correlation: Object of class "list" ~~

expression: Object of class "matrix" ~~

history: Object of class "environment" ~~

linCenvir: Object of class "environment" ~~

Methods

+ `signature(e1 = "LINCmatrix", e2 = "LINCfeature")`: ...

clusterlinc `signature(linc = "LINCmatrix")`: ...

linc `signature(object = "LINCmatrix", codingGenes = "missing")`: ...

plotlinc `signature(input = "LINCmatrix", showCor = "character")`: ...

plotlinc `signature(input = "LINCmatrix", showCor = "missing")`: ...

singlelinc `signature(input = "LINCmatrix")`: ...

Examples

```
showClass("LINCmatrix")
```

LINCsingle-class *Class* "LINCsingle"

Description

"LINCsingle"

Objects from the Class

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

Slots

results: Object of class "list" ~~

assignment: Object of class "vector" ~~

correlation: Object of class "list" ~~

expression: Object of class "matrix" ~~

history: Object of class "environment" ~~

linCenvir: Object of class "environment" ~~

Extends

Class "LINCmatrix", directly.

Methods

plotlinc `signature(input = "LINCsingle", showCor = "character")`: ...

plotlinc `signature(input = "LINCsingle", showCor = "missing")`: ...

Examples

```
showClass("LINCsingle")
```

linctable-methods *Write To Table* **LINC**

Description

Write and save a table in **LINC**

Methods

`signature(file_name = "character", input = "LINCcluster")` Table of co-expressed genes.

Examples

```
# NOT RUN:
# write to a table
# linctable(file_name = "crbl_co_expr", input = crbl_cluster)
```

plotlinc-methods *Plot Objects Of The 'LINC' class*

Description

`plotlinc` is the generic function that creates plots based on objects of the **LINC** class.

Usage

```
plotlinc(input,
         showCor,
         returnDat = FALSE)
```

Arguments

`input`	usually a `LINCmatrix`, `LINCcluster`, `LINCbio` or `LINCsingle` object
`showCor`	a `vector` of gene names; length 2 up to 6 elements; the first entry is the subject (see example)
`returnDat`	whether to return the data used to create the plot `TRUE` or not `FALSE`

Details

A plot will be created based on the information found in the `slots`: `history` and `linCenvir` of a `LINCmatrix`, `LINCcluster`, `LINCbio` or `LINCsingle` object. Using the argument `showCor = ...` will output scatterplots for one query, the first entry in `...`, and up to five subjects (see example). This can be used to manually check the correlation of genes. Individual or modified plots can be generated by returning the plotting data with `returnDat` Unexpected extreme values may corrupt plots.

Value

an object of the class 'gtable' containing multiple 'grobs' or a 'list' if `returnDat = TRUE`

Methods

signature(input = "LINCbio", showCor = "character") (see details)

signature(input = "LINCbio", showCor = "missing") (see details)

signature(input = "LINCcluster", showCor = "character") (see details)

signature(input = "LINCcluster", showCor = "missing") (see details)

signature(input = "LINCmatrix", showCor = "character") (see details)

signature(input = "LINCmatrix", showCor = "missing") (see details)

signature(input = "LINCsingle", showCor = "character") (see details)

signature(input = "LINCsingle", showCor = "missing") (see details)

Compatibility

plotlinc(LINCmatrix, ...), plotlinc(LINCcluster, ...), plotlinc(LINCbio, ...),plotlinc(LINCsingle, ...

Author(s)

Manuel Goepferich

See Also

querycluster ; feature

Examples

```
data(BRAIN_EXPR)
plotlinc(crbl_matrix)
plotlinc(crbl_cluster)

# show correlations and expression; "647979" as query plus 4 subjects
plotlinc(crbl_cluster, showCor = c("647979", "6726", "3337", "3304" ,"3320"))
```

---

querycluster *Cluster One ncRNA Gene Based On Its Co-Expression in Multiple Datasets*

Description

querycluster takes a set of 'LINCcluster' objects, extracts the respective co-expressed protein-coding genes and plots a dendrogram with the distance matrix attached. This function is intended to be applied in a case where a particular ncRNA occurrs in datasets which represent different tissues, batches, statistical corrections, reduced gene sets, controls and so on. The output will show the clustering of the groups and therefore the information under which condition is the co-expression to the query most similar.

Usage

```
querycluster(query = NULL,
             queryTitle = NULL,
             traits = NULL,
             method = "spearman",
             returnDat = FALSE,
             mo_promise,
             ...)
```

Arguments

query	the query name, i.e. the gene id of a ncRNA present in the supplied input
queryTitle	a character string used as the title of the plot
traits	NULL or a single integer. For NULL all co-expressed genes will be used. A number instead will be considered as maximal number of traits.
method	a character string, has to be one of c("spearman", "dicedist")
returnDat	whether to return the data used to create the plot TRUE or not FALSE
mo_promise	mo_promise = 'list', a list of 'LINCcluster' objects (see example)
...	the 'LINCcluster' objects itself, but not a combination of both, mo_promise = 'list' and supplying the objects itself (see example)

Details

This function will search for co-expressed protein-coding genes which belong to a the defined query. Based on the co-expression in the input 'LINCcluster' objects a distance matrix is computed. The method "spearman" finds the union of all interaction partners for the query und calculates the correlation between the 'LINCcluster' objects. For this method the distance measure is (1 - correlation). Alternatively, method = "dicedist" takes the Czekanovski dice distance [1] as distance mesaure of the traits = n genes. This method, however, will not work with traits = NULL. Choosing a low number for n will limit the number of different values in the distance matrix.

Apart from queryTitle the command 'LINCcluster' + feature(customID = ..., customCol = ...) enables a customized plot as output. For this to work the supplied 'LINCcluster' objects in ... have to be modified by the function feature(...) in advance.

Value

an object of the class 'gg' or a 'list' if returnDat = TRUE

Author(s)

Manuel Goepferich

References

[1] Christine Brun, Francois Chevenet, David Martin, Jerome Wojcik, Alain Guenoche and Bernard Jacq" Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network" (2003) Genome Biology, 5:R6.

See Also

feature ; clusterlinc

Examples

```
data(BRAIN_EXPR)

# add custom names and colors
gbm_cluster <- gbm_cluster + feature(customID = "CANCER_GBM", customCol = "red")
ctx_cluster <- ctx_cluster + feature(customID = "HEALTHY_CTX", customCol = "blue")
hpc_cluster <- hpc_cluster + feature(customID = "HEALTHY_HPC", customCol = "blue")
crbl_cluster <- crbl_cluster + feature(customID = "HEALTHY_CRBL", customCol = "blue")

# plot the dendrogram
querycluster('647979', queryTitle = 'NORAD',
             gbm_cluster,  # Glioblastoma
             ctx_cluster,  # Cortex
             hpc_cluster,  # Hippocampus
             crbl_cluster) # Cerebellum

# objects can also be supplied as a list
query_list <- list(gbm_cluster,
                   ctx_cluster,
                   hpc_cluster,
                   crbl_cluster)

# mo_promise is the (informal) argument for multiple objects
querycluster(query = '647979', queryTitle = 'NORAD', mo_promise = query_list)

# used the Czekanovski dice distance based on the 25 best
# interaction partners in each tissue
querycluster(query = '647979', method = "dicedist", traits = 25, mo_promise = query_list)

# NOT RUN:
# querycluster(query = '647979', method = "dicedist", mo_promise = query_list)
```

`results-methods` *Methods for the Getter Function* `results` *in Package* **LINC**

Description

Access the slot `results` of an object of class `LINC`

Arguments

`x`	a `'LINC'` object, for instance `LINCmatrix`

Value

the respective substructure or information

Methods

`signature(x = "LINCbio")` results slot

`signature(x = "LINCcluster")` results slot

`signature(x = "LINCmatrix")` results slot

`signature(x = "LINCsingle")` results slot

Author(s)

Manuel Goepferich

Examples

```
data(BRAIN_EXPR)

results(crbl_cluster)
```

`singlelinc-methods` *Co-Expression Analysis Of A Single ncRNA Gene*

Description

The function `singlelinc` perfroms co-expression analysis for a single query. An input `LINCmatrix` will be converted to a `LINCsingle` object. As a first step (I) a set of co-expressed protein-coding genes of a query is determined. Secondly, (II) biological terms related to the these genes are derived. The result will show the co-expression for the query.

Usage

```
singlelinc(input,
           query = NULL,
         onlycor = FALSE,
         testFun = cor.test,
     alternative = "greater",
       threshold = 0.05,
         underth = FALSE,
       coExprCut = NULL,
      enrichFun  = 'enrichGO',
           ont   = "BP",
         verbose = TRUE,
         ...)
```

Arguments

`input`	an object of the class `LINCmatrix`
`query`	the name of the (ncRNA) gene to be evaluated; has to be present in `input`
`onlycor`	if `TRUE` co-expression will be decided based on absolute correlation values from the input `LINCmatrix` object. If `FALSE` co-expressed genes will be selected based on p-values from the correlation test.
`testFun`	a function to test the robustness of correlations. User-defined functions are allowed. The expected output is a p-value.
`alternative`	one of `c("two.sided", "less", "greater")`. This argument indicates the alternative in the correlation test. `"less"` instead of `"greater"` can be used for negative correlations.
`threshold`	a single number representing the threshold for selecting co-expressed genes

underth	if TRUE values lower than the threshold will be considered (intended for p-values). If FALSE values higher tahn the threshold will be considered (intended for absolute correlations)
coExprCut	a single integer indicating the maximal number of co-expressed genes to select. In case too many genes fulfill the supplied threshold criterion their number can be reduced by this argument.
enrichFun	a function given as character string which will derive significant biological terms based on the set of co-expressed genes from a gene annotation resource. Supported functions are: c("enrichGO", "enrichPathway", "enrichDO")
ont	a subontology, only used for enrichFun = 'enrichGO'. This has to be one of "MF", "BP", "CC".
verbose	whether to give messages about the progression of the function TRUE or not FALSE
...	further arguments, mainly for cor.test and functions from clusterProfiler

Details

In comparison to the function clusterlinc this function will provide more flexibility in terms of the selection of co-expressed genes. The option onlycor = TRUE in combination with a suitable threshold can be used to choose co-expressed protein-coding genes based on the correlation values inherited from the input LINCmatrix object. For this to work it is required to set underth = FALSE because then, values higher than the threshold will be picked. By default, co-expression depnds on the p-values from the correlation test (stats::cor.test) which demonstrate the robustness of a given correlation between two genes. A user-defined test function supplied in testFun requires the formal arguments x, y, method and use. Moreover, the p-values of the output should be accessible by $pvalue. The number of co-expressed genes can be restricted not only by threshold, but also by coExprCut. The value n for coExprCut = n will be ignored in case the number of genes which fulfill the threshold criterion is smaller than n. Options for enrichFun are for example: ReactomePA::enrichPathway() or clusterProfiler::enrichGO. Further arguments (...) are inteded to be passed to the called enrichFun function. enrichFun = 'enrichGO', ont = "CC" will call the subontology "Cellular Component" from GO. In case genes are not given as Entrez ids they will be translated. For more details see the documentation ofclusterProfiler.

Value

an object of the class 'LINCmatrix' (S4) with 6 Slots

results	a list of four entries: $query, the queried gene, $bio, a list of biological terms and their p-values, $cor, absolute correlations of co-expressed genes, $pval, p-values from the correlation test of co-expressed genes
assignment	a character vector of protein-coding genes
correlation	a list of $single, the correlation of the query to protein-coding genes
expression	the original expression matrix
history	a storage environment of important methods, objects and parameters used to create the object
linCenvir	a storage environment ensuring the compatibility to other objects of the LINC class

Methods

signature(input = "LINCmatrix") (see details)

Compatibility

plotlinc(LINCsingle, ...)), ...

Author(s)

Manuel Goepferich

See Also

linc ; singlelinc

Examples

```
data(BRAIN_EXPR)

# selection based on absolute correlation
meg3 <- singlelinc(crbl_matrix, query = "55384", onlycor = TRUE, underth = FALSE, threshold = 0.5)
plotlinc(meg3)

# using the 'cor.test' in combination with 'underth = TRUE'
meg3 <- singlelinc(crbl_matrix, query = "55384", underth = TRUE, threshold = 0.0005, ont = 'BP')
plotlinc(meg3)
```

Index