

Package ‘omicRexposome’

September 19, 2024

Title Exposome and omic data associatin and integration analysis

Version 1.27.0

Maintainer Xavier Escribà Montagut <xavier.escriba@isglobal.org>

Description omicRexposome systematizes the association evaluation between exposures and omic data, taking advantage of MultiDataSet for coordinated data management, rexposome for exposome data definition and limma for association testing. Also to perform data integration mixing exposome and omic data using multi co-inherent analysis (omicade4) and multi-canonical correlation analysis (PMA).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

VignetteBuilder knitr

Depends R (>= 3.4), Biobase

Imports stats, utils, grDevices, graphics, methods, rexposome, limma, sva, ggplot2, ggrepel, PMA, omicade4, gridExtra, MultiDataSet, SmartSVA, isva, parallel, SummarizedExperiment, stringr

Suggests BiocStyle, knitr, rmarkdown, snpStats, brgedata

biocViews ImmunoOncology, WorkflowStep, MultipleComparison, Visualization, GeneExpression, DifferentialExpression, DifferentialMethylation, GeneRegulation, Epigenetics, Proteomics, Transcriptomics, StatisticalMethod, Regression

git_url <https://git.bioconductor.org/packages/omicRexposome>

git_branch devel

git_last_commit 33b689f

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-09-18

Author Carles Hernandez-Ferrer [aut, cre],
Juan R. González [aut]

Contents

add_cls	2
add_exp	3
asr	4
association	4
crossomics	6
crs	7
getIntegration	7
mds	8
omicRexposome	8
plotAssociation	9
plotHits	10
plotIntegration	10
plotLambda	11
snpToContinuous	12
tableHits	12
tableLambda	13
Index	14

add_cls	<i>Method to add an ExposomeClust to a MultiDataSet</i>
---------	---

Description

This method allows to insert an object of class [ExposomeClust](#) as an independent dataset into an object of class [MultiDataSet](#).

Usage

```
add_cls(object, clsSet, ...)

## S4 method for signature 'MultiDataSet,ExposomeClust'
add_cls(object, clsSet, ...)
```

Arguments

object	An object of class MultiDataSet .
clsSet	An object of class ExposomeClust .
...	Arguments given to add_eset from MultiDataSet .

Value

A [MultiDataSet](#) with the [ExpressionSet](#) added as an independent dataset.

Examples

```
data("eclust", package = "rexposome")
library(MultiDataSet)
md <- new("MultiDataSet")
names(md)
md <- add_cls(md, expo_c)
names(md)
```

add_exp

Method to add an ExposomeSet to a MultiDataSet

Description

This method allows to insert an object of class [ExposomeSet](#) as an independent dataset into an object of class [MultiDataSet](#).

Usage

```
add_exp(object, expoSet, warnings = TRUE, ...)

## S4 method for signature 'MultiDataSet,ExposomeSet'
add_exp(object, expoSet, warnings = TRUE,
        ...)
```

Arguments

object	An object of class MultiDataSet .
expoSet	An object of class ExposomeSet .
warnings	(default TRUE) If set to FALSE warnings will not be displayed.
...	Arguments given to add_eset from MultiDataSet .

Value

A [MultiDataSet](#) with the [ExpressionSet](#) added as an independent dataset.

Examples

```
data("exposome", package = "rexposome")
library(MultiDataSet)
md <- new("MultiDataSet")
names(md)
md <- add_exp(md, expo)
names(md)
```

asr	<i>ResultSet for testing and illustration purposes</i>
-----	--

Description

ResultSet created using `association` method, testing proteome association to exposome ("mds"), adjusted by sex and age.

Usage

```
data("asr")
```

Format

An object of class `ResultSet` of length 15.

Value

A `ResultSet` object.

Examples

```
data("asr", package = "omicRexposome")
asr
```

association	<i># _____ Method to perform an association study between transcriptome and exposom</i>
-------------	---

Description

This function allows to perform an association study between gene expression from microarray and the exposome. An `ExpressionSet` is the object storing the gene expression and an `ExposomeSet` the one storing the exposome. Both of them needs to be encapsulated in a `MultiDataSet`. The association study is perform through standard `limma` pipeline. The function allows to perform multiple tests using the argument `exposures`.

Usage

```
association(object, formula, expset, omicset, set = "exposures",
  method = "ls", ..., baselevels, sva = "none", vfilter = NULL,
  verbose = FALSE, warnings = TRUE)

## S4 method for signature 'MultiDataSet'
association(object, formula, expset, omicset,
  set = "exposures", method = "ls", ..., baselevels, sva = "none",
  vfilter = NULL, verbose = FALSE, warnings = TRUE)
```

Arguments

object	A <code>MultiDataSet</code> object containing at least one omic data-sets like <code>ExpressionSet</code> , <code>MethylationSet...</code> and, at last, one <code>ExposomeSet</code> .
formula	formula to be evaluated by each exposure (or phenotype, see <code>set</code> argument). It should not contain any exposures (or phenotype), it will be added automatically when evaluated.
expset	Name of the <code>ExposomeSet</code> in object.
omicset	Name of the omic data-set in object.
set	(default "exposures") Can take value "exposures" to test the association of the exposures in the <code>ExposomeSet</code> vs. the features in the omic data-set. If takes "phenotypes" all phenotypes in <code>ExposomeSet</code> are tested.
method	(default "lm") Check <code>limma</code> help pages.
...	Arguments passed to <code>limma</code> 's <code>lmFit</code> .
baselevels	(optional) If set, must be a labeled vector with the default base level for categorical exposures.
sva	(default "none"). This argument can take value "none" to do not apply SVA. Value "fast" will run SVA using <code>isva</code> and <code>SmartSVA</code> . Value "slow" will run SVA using <code>sva</code> .
vfilter	(default NULL). Only used when <code>sva = "slow"</code> . Numeric number of probes used in <code>sva</code> . Recommended ~10% of real probes.
verbose	(default FALSE) If set to TRUE, a series of messages describing the process are shown.
warnings	(default TRUE) If set to TRUE, a series of warnings are shown when required user attention.

Value

An object of class `ResultSet`.

Examples

```
library(MultiDataSet)
data(brge_prot, package = "brgedata")
data(brge_expo, package = "brgedata")
mds <- createMultiDataSet()
mds <- add_eset(mds, brge_prot, dataset.type = "proteines")
mds <- add_eset(mds, brge_expo, dataset.type = "exposures", GRanges = NA)

asr <- association(mds, formula = Asthma ~ Sex + Age,
  expset = "exposures", omicset = "proteines")
asr
```

crossomics

*Function to perform a Transcriptome-Wide Association Study***Description**

This function allows to perform a Transcriptome-Wide Association Study by using an `ExposmeSet` and an `ExpressionSet`. It allows to perform an adjustment using Surrogate Variable Analysis (from R package `sva`).

Usage

```
crossomics(object, method = "mcca", ncomponents = 2, ..., na.rm = FALSE,
           permute = c(100, 3), verbose = FALSE, warnings = TRUE)
```

```
## S4 method for signature 'MultiDataSet'
crossomics(object, method = "mcca",
           ncomponents = 2, ..., na.rm = FALSE, permute = c(100, 3),
           verbose = FALSE, warnings = TRUE)
```

Arguments

<code>object</code>	A <code>MultiDataSet</code> object containing at last two data-sets like <code>ExposomeSet</code> , <code>ExpressionSet</code> , <code>MethylationSet</code> ...
<code>method</code>	(default "mcca") It can takes values "mcca" for Multiple Canonical Correlation Analysis or "mca" for Multiple Co-Inertia Analysis.
<code>ncomponents</code>	(default 2) Number of components to be estimated.
<code>...</code>	Other arguments given to <code>mca</code> (from <code>omicade4</code>) or to <code>MultiCCA</code> (from <code>PMA</code>).
<code>na.rm</code>	(default FALSE) If method was set to "mcca" and <code>na.rm</code> was set to TRUE, features containing missing values are removed.
<code>permute</code>	(default <code>c(100, 3)</code>). If <code>method="mcca"</code> and this argument is set to NULL no permutation test to tune-up the parameters for <code>MultiCCA</code> . When filled, <code>permute[1]</code> corresponds to the number permutations (default in <code>MultiCCA.permute</code> is 25) and <code>permute[2]</code> the number of iterations (default in <code>MultiCCA.permute</code> is 3).
<code>verbose</code>	(default FALSE) If set to TRUE, a series of messages describing the process are shown.
<code>warnings</code>	(default TRUE) If set to TRUE, a series of warnings are shown when required user attention.

Value

An object of class `ResultSet`.

Examples

```
library(MultiDataSet)
library(rexposome)
data(brge_prot, package = "brgedata")
data(brge_expo, package = "brgedata")
mds <- createMultiDataSet()
mds <- add_eset(mds, brge_prot, dataset.type = "proteines")
```

```

mds <- add_eset(mds, imputation(brge_expo),
  dataset.type = "exposures", GRanges = NA)

crs <- crossomics(mds, method = "mcia")
crs

```

crs

ResultSet for testing and illustration purposes

Description

ResultSet created using [crossomics](#) method, selecting "mcia" method. Result from the integration of proteome and exposome data ("mds").

Usage

```
data("crs")
```

Format

An object of class `ResultSet` of length 1.

Value

A `ResultSet` object.

Examples

```
data("crs", package = "omicRexposome")
crs
```

getIntegration

Method to extrat integration-feature result from a ResultSet

Description

Homologous methods from `MultiDataSet` (`getAssociation`) but for `ResultSet` created by [crossomics](#). It Returns a `data.frame` with the result from `mcia` (`omicade4`) or from `MultiCCA` (PMA).

Usage

```

getIntegration(object, ...)

## S4 method for signature 'ResultSet'
getIntegration(object, ...)

```

Arguments

object	An object of class ResultSet obtained from
...	NOT USED

Value

A data.frame

Examples

```
data("crs", package = "omicRexposome")
class(getIntegration(crs))
```

mds	<i>MultiDataSet for testing and illustration purposes</i>
-----	---

Description

MultiDataSet containing both proteome data-set and exposome data-set.

Usage

```
data("mds")
```

Format

An object of class MultiDataSet of length 2.

Value

A MultiDataSet object.

Examples

```
data("mds", package = "omicRexposome")
mds
```

omicRexposome	<i>omicRexposome: Package for exposome and omic data associatin and integration</i>
---------------	---

Description

omicRexposome: Package for exposome and omic data associatin and integration

exposome-omic data association study

The packages offers the function [association](#) that allows to perform an association study using transcriptome, methylome, etc. as dependent variable and exposome data as independent variable. The function relies on limma pipeline and generates an object of class ResultSet, that can be plotted using [plotAssociation](#).

exposome-omic data integration study

The packages offers the function [crossomics](#) that allows to perform two types of integration study: Multi Canonical Correlation Analysis and Multi Co-Inertia Analysis. The function allos to use any type and number of datasets (aka. exposome transcriptome, methylome, etc.). The function generates an object of class ResultSet, that can be plotted using [plotIntegration](#).

plotAssociation	<i>Function to draw de result of an association study</i>
-----------------	---

Description

This function draws two type of plots for the `ResultSet` from association functions

Usage

```
plotAssociation(object, rid = 1, coef = 2, contrast = 1,
  type = c("manhattan", "qq", "volcano"), tPV = NULL, tFC = NULL,
  show.effect = FALSE)

## S4 method for signature 'ResultSet'
plotAssociation(object, rid = 1, coef = 2,
  contrast = NULL, type = c("manhattan", "qq", "volcano"), tPV = NULL,
  tFC = NULL, show.effect = FALSE)
```

Arguments

<code>object</code>	An object of class <code>ResultSet</code> obtained from <code>assoc_*</code> functions.
<code>rid</code>	(default 1) Index or name of the test to be plotted.
<code>coef</code>	(default 2) Index of the coefficient to be extracted.
<code>contrast</code>	(default 1) When code corresponds to a mult categorical variable, <code>contrast</code> selects the comparison.
<code>type</code>	Can take "volcano", "qq", "manhattan" and "protein". "protein" lot is a type of Manhattan plot designed for protein association analysis.
<code>tPV</code>	(optional) Threshold for P.Value when <code>type="volcano"</code> .
<code>tFC</code>	(optional) Threshold for Fold Change or Effect when <code>type="volcano"</code> .
<code>show.effect</code>	(default FALSE) If set to TRUE, when <code>type="volcano"</code> the X-axis will show 2^{\logFC} instead of \logFC .

Value

A `ggplot2` object

See Also

[plotIntegration](#) for plotting integration results. [association](#) to create a `ResultSet` to be passed to this function.

Examples

```
data("asr", package = "omicRexposome")
plotAssociation(asr, type = "qq")
plotAssociation(asr, type = "volcano")
```

plotHits	<i>Plot number of hits per result in ResultSet</i>
----------	--

Description

This method draws a barplot with the number of hits in each result stored in the given [ResultSet](#).

Usage

```
plotHits(object, th = 0.05, width = 0.75)
```

```
## S4 method for signature 'ResultSet'
plotHits(object, th = 0.05, width = 0.75)
```

Arguments

object	An object of class ResultSet
th	(default 0.05) Threshold (p-value) to considere a result as a hit.
width	(default 0.70) width of the bar

Value

A ggplot2 object

See Also

[plotLambda](#) for a graphical representation of the lambda score per analysys, [tableLambda](#) for the lambda score per analysys, [tableHits](#) for the histts per analysys

Examples

```
data(asr, package = "omicRexposome")
plotHits(asr)
```

plotIntegration	<i>Function to draw de result of an integration study</i>
-----------------	---

Description

This function draws a plots for the ResultSet from integration function

Usage

```
plotIntegration(object, cmpX = 1, cmpY = 2, lb.th = 0.2,
  legend.show = TRUE, colors, ...)
```

```
## S4 method for signature 'ResultSet'
plotIntegration(object, cmpX = 1, cmpY = 2,
  lb.th = 0.2, legend.show = TRUE, colors, ...)
```

Arguments

object	An object of class ResultSet obtained from crossomics .
cmpX	(default 1) Value of the X-axis when plotting results from mcia .
cmpY	(default 2) Value of the Y-axis when plotting results from mcia .
lb.th	(default 0.20) Threshold to place labels on radar chart drawn when plotting results from MultiCCA .
legend.show	(default TRUE) If set to FALSE, right legend of radar plot is hidden when plotting results from MultiCCA .
colors	(optional) Names vector with the colors used to draw each dataset. Used when plotting results from MultiCCA . If missing, random colors are chosen.
...	Optional arguments are given to plot from omicade4 package (argument axes is filled with values from cmpX and cmpY).

Value

A ggplot2 object

See Also

[plotAssociation](#) for plotting association results. [crossomics](#) to create a [ResultSet](#) to be passed to this function.

Examples

```
data("crs", package = "omicRexposome")
plotIntegration(crs)
```

plotLambda

Plot lambda score for all results in a ResultSet

Description

This method draws a baplor with the lambda score of each result in the given [ResultSet](#).

Usage

```
plotLambda(object, width = 0.75)

## S4 method for signature 'ResultSet'
plotLambda(object, width = 0.75)
```

Arguments

object	An object of class ResultSet
width	(default 0.70) width of the bar

Value

A ggplot2 object

See Also

[plotHits](#) for a graphical representation of the hits per analysis, [tableLambda](#) for the lambda score per analysis, [tableHits](#) for the hits per analysis

Examples

```
data("asr", package = "omicRexposome")
plotLambda(asr)
```

snpToContinuous	<i>Transforms the discrete genotype from a snpSet to a matrix of a continuous variable.</i>
-----------------	---

Description

The function converts the categorical variable of SNPs to a continuous variable by normalizing each SNP as described in Abraham G. and Inouye M. 2014 (DOI: 10.1371/journal.pone.0093766).

Usage

```
snpToContinuous(snpSet, verbose = FALSE)
```

Arguments

snpSet	An object of class snpSet with set calls slot .
verbose	If set to TRUE, messages will be shown.

Value

An matrix of the calls of the SNPs converted to a continuous variable.

See Also

[crossomics](#) use this function

tableHits	<i>Counts the number of hits on the results stored in a ResultSet</i>
-----------	---

Description

Given a threshold it counts the number of hits in each result in the given [ResultSet](#).

Usage

```
tableHits(object, th = 0.05)

## S4 method for signature 'ResultSet'
tableHits(object, th = 0.05)
```

Arguments

object An object of class [ResultSet](#)
 th (default 0.05) Threshold (p-value) to considere a result as a hit.

Value

A labeled numeric vector with the exposures and the number of hits.

See Also

[tableLambda](#) for the lambda score per analysys, [plotLambda](#) for a graphical representation of the lambda score per analysys, [plotHits](#) for a graphical representation of the histis per analysys

Examples

```
data("asr", package = "omicRexposome")
tableHits(asr)
```

tableLambda	<i>Compute a lambda score on the results stored in a ResultSet</i>
-------------	--

Description

Compute lambda score on each result in the given [ResultSet](#) by using [lambdaClayton](#).

Usage

```
tableLambda(object, trim = 0.5)

## S4 method for signature 'ResultSet'
tableLambda(object, trim = 0.5)
```

Arguments

object An object of class [ResultSet](#)
 trim (default 0.5) percentage of right omitted values for [lambdaClayton](#).

Value

Returns a data.frame having the exposures and the computed lambda score.

A labeled numeric vector with the lambda score for each exposure.

See Also

[tableHits](#) for the number of hits per analysys, [plotHits](#) for a graphical representation of the histis per analysys, [plotLambda](#) for a graphical representation of the lambda score per analysys

Examples

```
data("asr", package = "omicRexposome")
tableLambda(asr)
```

Index

- * **datasets**
 - asr, [4](#)
 - crs, [7](#)
 - mds, [8](#)
- add_cls, [2](#)
- add_cls, MultiDataSet, ExposomeClust-method (add_cls), [2](#)
- add_eset, [2, 3](#)
- add_exp, [3](#)
- add_exp, MultiDataSet, ExposomeSet-method (add_exp), [3](#)
- asr, [4](#)
- association, [4, 4, 8, 9](#)
- association, MultiDataSet-method (association), [4](#)
- crossomics, [6, 7, 8, 11, 12](#)
- crossomics, MultiDataSet-method (crossomics), [6](#)
- crs, [7](#)
- ExposomeClust, [2](#)
- ExposomeSet, [3](#)
- ExpressionSet, [2, 3](#)
- getIntegration, [7](#)
- getIntegration, ResultSet-method (getIntegration), [7](#)
- lambdaClayton, [13](#)
- mcia, [11](#)
- mds, [8](#)
- MultiCCA, [11](#)
- MultiDataSet, [2, 3](#)
- omicade4, [11](#)
- omicRexposome, [8](#)
- omicRexposome-package (omicRexposome), [8](#)
- plotAssociation, [8, 9, 11](#)
- plotAssociation, ResultSet-method (plotAssociation), [9](#)
- plotHits, [10, 12, 13](#)
- plotHits, ResultSet-method (plotHits), [10](#)
- plotIntegration, [8, 9, 10](#)
- plotIntegration, ResultSet-method (plotIntegration), [10](#)
- plotLambda, [10, 11, 13](#)
- plotLambda, ResultSet-method (plotLambda), [11](#)
- ResultSet, [5-7, 9-13](#)
- snpToContinuous, [12](#)
- snpToContinuous, SnpSet-method (snpToContinuous), [12](#)
- sva, [5](#)
- tableHits, [10, 12, 12, 13](#)
- tableHits, ResultSet-method (tableHits), [12](#)
- tableLambda, [10, 12, 13, 13](#)
- tableLambda, ResultSet-method (tableLambda), [13](#)