

Package ‘rexposome’

February 25, 2025

Version 1.28.0

Title Exposome exploration and outcome data analysis

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

Description Package that allows to explore the exposome and to perform association analyses between exposures and health outcomes.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

VignetteBuilder knitr

Depends R (>= 3.5), Biobase

Imports methods, utils, stats, lsr, FactoMineR, stringr, circlize, corrplot, ggplot2, ggridges, reshape2, pryr, S4Vectors, imputeLCMD, scatterplot3d, glmnet, gridExtra, grid, Hmisc, gplots, gtools, scales, lme4, grDevices, graphics, ggrepel, mice

Suggests mclust, flexmix, testthat, BiocStyle, knitr, formatR, rmarkdown

biocViews Software, BiologicalQuestion, Infrastructure, DataImport, DataRepresentation, BiomedicalInformatics, ExperimentalDesign, MultipleComparison, Classification, Clustering

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

git_branch RELEASE_3_20

git_last_commit 6a4640d

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2025-02-24

Author Carles Hernandez-Ferrer [aut, cre],
Juan R. Gonzalez [aut],
Xavier Escribà-Montagut [aut]

Contents

| | |
|------------------------------|----|
| classification | 3 |
| clustering | 3 |
| correlation | 5 |
| expo | 6 |
| expos | 6 |
| ExposomeClust | 7 |
| ExposomeCorr | 8 |
| ExposomePCA | 9 |
| ExposomeSet | 11 |
| exposureNames | 16 |
| expo_c | 17 |
| extract | 17 |
| ExWAS | 18 |
| exwas | 20 |
| ex_imp | 21 |
| familyNames | 21 |
| get_robust_sd | 22 |
| highAndLow | 23 |
| ilod | 24 |
| imExposomeSet | 25 |
| imputation | 28 |
| imputeLOD | 28 |
| invExWAS | 29 |
| loadExposome | 30 |
| loadExposome_plain | 32 |
| loadImputed | 33 |
| me | 34 |
| mExWAS | 35 |
| mexwas | 36 |
| ndim | 37 |
| normalityTest | 37 |
| pca | 38 |
| phenotypeNames | 39 |
| plot3PCA | 40 |
| plotClassification | 41 |
| plotCorrelation | 41 |
| plotEffect | 42 |
| plotEXP | 43 |
| plotExwas | 44 |
| plotFamily | 45 |
| plotHistogram | 46 |
| plotLOD | 46 |
| plotMissings | 47 |
| plotPCA | 48 |
| plotPHE | 49 |
| plotVolcano | 50 |
| readExposome | 50 |
| rexposome | 52 |
| standardize | 53 |
| tableLOD | 54 |

| | |
|-------------------------|----|
| <i>classification</i> | 3 |
| tableMissings | 54 |
| tef | 55 |
| toES | 56 |
| trans | 56 |
| volcano_plot | 57 |

Index **59**

classification *Method to get the classification of the samples from an ExposomeClust.*

Description

Method to get the classification of the samples from an ExposomeClust.

Usage

```
classification(object)
```

Arguments

object An [ExposomeClust](#) to get the samples' classification.

Value

A labelled vector with the classification of each exposure.

See Also

[clustering](#) as a constructor for [ExposomeClust](#), [plotClassification](#) to plot the groups

Examples

```
data("eclust")
tt <- classification(expo_c)
table(tt)
```

clustering *Method to perform clustering on the samples of an ExposomeSet*

Description

This method allows to create an [ExposomeClust](#) object from an [ExposomeSet](#) object by clustering samples through the exposure levels. The method is flexible to accept any clustering method (method) that allows to obtain a classification (cmethod) of the samples. The function assigned to argument method must have an argument called data, waiting for the matrix of exposures (samples as rows, exposures as columns). If the result object of the method has no accessor \$classification, then a cmethod is required and will be applied on the result of method to obtain a labelled vector with the samples' classification.

Usage

```
clustering(object, method, cmethod, ..., warnings = TRUE)
```

Arguments

| | |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeSet containing the exposures used for the clustering process |
| method | Function applied to the exposures of object. This function must have an argument named as data that will receive the matrix of exposures. |
| cmethod | (optional) Function to obtain the classification from the object generated with method. |
| ... | Passed to content of method. |
| warnings | (default TRUE) If set to FALSE warnings will not be displayed. |

Value

ExposomeClust with the original exposures and the classification of each exposure.

Note

The function assigned to cmethod will be directly applied to the result of the method as: cmethod(model); being model the result of method.

See Also

[classification](#) to see how to obtain the classification of the samples from an [ExposomeClust](#), [plot-Classification](#) to plot the groups

Examples

```
data("exposome")

# EXAMPLE 1: Clustering with mclust
library(mclust)
c <- clustering(expo[12:32, ], method = Mclust, G = 2)
table(classification(c)) # This works since the result of Mclust has an accessor
                        # $classification

# EXAMPLE 2: Clustering with flexmix
library(flexmix)
# First we create a function to apply flexmix to the ExposomeSet
flexmix_clust <- function(data, ...) {
  data <- as.matrix(data)
  flexmix(formula = data~1, ...)
}

# Then if we apply the method to the ExposomeSet it will crash:
# c <- clustering(expo[12:32, ], method = flexmix_clust, k = 2, model = FLXMCmvnorm())
# Because the method does not know how to obtain the classification for the result
# since flexmix has not an accessor called $classification

# We create a function to get the classification
flexmix_clas <- function(model, ...) {
  return(clusters(model))
}
```

```
# We put it to the ExposomeClust
c <- clustering(expo[12:32, ], method = flexmix_clust, cmethod = flexmix_clas,
  k = 2, model = FLXMCmvnorm())
classification(c) # This works because the ExposomeClust has a way to get
  # the classification
```

correlation

Creation of an ExposomeCorr from an ExposomeSet.

Description

Method to calculate the correlation between the exposures of an [ExposomeSet](#). The correlation method takes into account the nature of each pair of exposures: continuous vs. continuous uses cor function from R base, categorical vs. categorical uses cramerV function from lsr R package and categorical vs. continuous exposures correlation is calculated as the square root of the adjusted r-square obtained from fitting a lineal model with the categorical exposures as dependent variable and the continuous exposure as independent variable. The function creates and returns an [ExposomeCorr](#) object.

Usage

```
correlation(object, ..., warnings = TRUE)
```

Arguments

| | |
|----------|-----------------------------------------------------------------------------------------------------|
| object | ExposomeSet which exposures will be used to calculate their correlation |
| ... | Other arguments passed to cor , cramersV or to lm . |
| warnings | (default TRUE) If set to FALSE warnings will not be displayed. |

Value

ExposomeCorr with the correlation between the selected exposures and their description

See Also

[plotCorrelation](#) to plot the correlations of an [ExposomeCorr](#), [clustering](#) to see how the exposures can cluster samples, [pca](#) to compute PCA on exposures

Examples

```
data("exposome")
expo.c <- correlation(expo)
expo.c
expo.c.table <- extract(expo.c)
```

| | |
|------|-----------------------------------------|
| expo | ExposomeSet <i>for testing purposes</i> |
|------|-----------------------------------------|

Description

ExposomeSet with full set of 104 exposures, 1200 simulated samples and 4 phenotypes (asthma status, BMI measure, sex, and age).

Usage

```
data("exposome")
```

Format

An object of class ExposomeSet of dimension 104 x 1200 x 4.

Value

An ExposomeSet object.

Examples

```
data("exposome")
dim(expo)
exposureNames(expo)
familyNames(expo)
sampleNames(expo)
phenotypeNames(expo)
```

| | |
|-------|--------------------------------------------------------|
| expos | <i>Returns the exposures matrix of an ExposomeSet.</i> |
|-------|--------------------------------------------------------|

Description

Given an [ExposomeSet](#) it returns the inner matrix of exposures, having the exposures as columns and the samples as rows.

Usage

```
expos(object)
```

Arguments

object An [ExposomeSet](#).

Value

A matrix of exposures

Examples

```
data("exposome")
expos(expo)[1:3, 1:3]
```

ExposomeClust *Class ExposomeClust*

Description

Class `ExposomeClust` obtained from [clustering](#) on an `ExposomeSet` object, represents the groups of samples created applying a clustering method on the `ExposomeSet`' exposures.

Usage

```
## S4 method for signature 'ExposomeClust,ANY'
plot(x, y, ...)

## S4 method for signature 'ExposomeClust'
classification(object)

## S4 method for signature 'ExposomeClust'
plotClassification(object, type = "heatmap", ...)

## S4 method for signature 'ExposomeClust'
sampleNames(object)
```

Arguments

| | |
|---------------------|-----------------------------------------------|
| <code>x</code> | Object of class <code>ExposomeClust</code> |
| <code>y</code> | NOT USED |
| <code>...</code> | Argument given to <code>heatmap.2</code> |
| <code>object</code> | An object of class <code>ExposomeClust</code> |
| <code>type</code> | (default "heatmap") Type of plot. |

Value

An object of class `ExposomeClust`

Methods (by generic)

- `plot(x = ExposomeClust, y = ANY)`: Wrapper for `plotClassification` method.
- `classification(ExposomeClust)`: Return classe assigned to each sample
- `plotClassification(ExposomeClust)`: Draws a heatmap for the samples' classification.
- `sampleNames(ExposomeClust)`: Method to obtain samples' names

Slots

`model` Result obtained on applying method on the exposures.

`method` Function used to perform the clustering of the exposures.

`call` Call used to create this object.

`samples` Name of the exposures after the clustering process.

See Also

[clustering](#) to apply a clustering on an [ExposomeSet](#) and create an [ExposomeClust](#).

ExposomeCorr *Class ExposomeCorr*

Description

Class `ExposomeCorr` contains a matrix of correlations between continuous exposures calculated using `cor`. It also contains the description of each exposures (fData of the original [ExposomeSet](#)) in order to maintain a coherence with the original source. It extends [eSet-class](#).

Usage

```
## S4 method for signature 'ExposomeCorr,ANY'
plot(x, y, ...)

## S4 method for signature 'ExposomeCorr'
dim(x)

## S4 method for signature 'ExposomeCorr'
extract(object, sort, ...)

## S4 method for signature 'ExposomeCorr'
plotCorrelation(object, type = c("circos", "matrix"), ...)
```

Arguments

| | |
|---------------------|-----------------------------------------------------------------------------|
| <code>x</code> | Object of class ExposomeCorr |
| <code>y</code> | NOT USED |
| <code>...</code> | Arguments passed to <code>corrplot</code> when <code>type="matrix"</code> . |
| <code>object</code> | An ExposomeCorr object. |
| <code>sort</code> | NOT USED |
| <code>type</code> | To choose between <code>"circos"</code> and <code>"matrix"</code> . |

Value

An object of class `ExposomeCorr`

Methods (by generic)

- `plot(x = ExposomeCorr, y = ANY)`: Wrapper for `plotClassification` method.
- `dim(ExposomeCorr)`: Return the dimension of the internat matrix of correlation.
- `extract(ExposomeCorr)`: Return the raw correlation matrix
- `plotCorrelation(ExposomeCorr)`: Draws both a matrix of circos plot of correlations

Slots

assayData Contains the correlation matrix (see [eSet](#), [AssayData](#)).

featureData Contains the description of the exposures including the family where they belong (see [eSet](#), [AnnotatedDataFrame](#)).

See Also

[pca](#) to study the behaviour between samples and exposures in an [ExposomeSet](#)

ExposomePCA

Class ExposomePCA

Description

Class ExposomePCA contains a matrix of exposures used to compute the PCA, also a table of phenotypes and a set containing the multiple results of computing the PCA.

Usage

```
## S4 method for signature 'ExposomePCA,ANY'  
plot(x, y, ...)
```

```
## S4 method for signature 'ExposomePCA'  
exposureNames(object)
```

```
## S4 method for signature 'ExposomePCA'  
extract(object, table = "exposures", ...)
```

```
## S4 method for signature 'ExposomePCA'  
ndim(object)
```

```
## S4 method for signature 'ExposomePCA'  
phenotypeNames(object)
```

```
## S4 method for signature 'ExposomePCA'  
plot3PCA(  
  object,  
  cmpX,  
  cmpY,  
  cmpZ,  
  phenotype,  
  main,  
  angle = 35,  
  pch = 16,  
  legend = TRUE,  
  plines = TRUE  
)
```

```
## S4 method for signature 'ExposomePCA'  
plotEXP(object, exposure)
```

```
## S4 method for signature 'ExposomePCA'
plotPCA(
  object,
  set,
  cmpX = 1,
  cmpY = 2,
  show.exposures = FALSE,
  show.samples = FALSE,
  phenotype
)

## S4 method for signature 'ExposomePCA'
plotPHE(object, phenotype, exp2fac = 5)
```

Arguments

| | |
|----------------|--------------------------------------------------------------------------------------------|
| x | Object of class ExposomePCA |
| y | NOT USED |
| ... | NOT USED |
| object | An ExposomePCA object |
| table | Can takes values "exposures" or "individuals". |
| cmpX | PC to place in X-axis |
| cmpY | PC to place in Y-axis |
| cmpZ | PC to place in Z-axis |
| phenotype | (when set="samples") A phenotype can be selected so the samples are coloured by its value. |
| main | Title |
| angle | Point of view |
| pch | Size of the dots |
| legend | Boolean to show or hide the legend |
| plines | Boolean to show of hide the dotted lines that helps to place the dots in the X/Y axes |
| exposure | Vector of exposures to be shown in the plot |
| set | Can take values "exposures", "samples" or "all" |
| show.exposures | When set to TRUE, labels for exposures are shown |
| show.samples | When set to TRUE, labels for samples are shown |
| exp2fac | Number of different values to considere an exposures continuous |

Value

An object of class ExposomePCA

Methods (by generic)

- `plot(x = ExposomePCA, y = ANY)`: Wrapper for `plotPCA` method.
- `exposureNames(ExposomePCA)`: Getter to obtain the exposures's names.
- `extract(ExposomePCA)`: Method to extract the raw results of the PCA.
- `ndim(ExposomePCA)`: Number of principal components in an `ExposomePCA`.
- `phenotypeNames(ExposomePCA)`: Getter to obtain the phenotype's names.
- `plot3PCA(ExposomePCA)`: Method to draw a 3D plot for PCA
- `plotEXP(ExposomePCA)`: Plot correlation between exposures and PCA
- `plotPCA(ExposomePCA)`: Method to draw a 2D plot for PCA
- `plotPHE(ExposomePCA)`: Plot association score between phenotypes and PCA

Slots

`pca` list containing all elements of the PCA

`phenoData` Contains the phenotypes or variables experimenter-supplied (see [eSet](#), [AnnotatedDataFrame](#)).

`featureData` Contains the description of the exposures including the family where they belong (see [eSet](#), [AnnotatedDataFrame](#)).

See Also

[correlation](#) to study the correlation between exposures in a [ExposomeSet](#)

ExposomeSet

Class ExposomeSet

Description

Class `ExposomeSet` contains the exposure levels, the exposure's description and the samples phenotype. It is the starting object for `rexposome` package and extends [eSet](#).

Usage

```
## S4 method for signature 'ExposomeSet,ANY'
plot(x, y, ...)

## S4 method for signature 'ExposomeSet'
clustering(object, method, cmethod, ..., warnings = TRUE)

## S4 method for signature 'ExposomeSet'
correlation(object, ..., warnings = TRUE)

## S4 method for signature 'ExposomeSet'
dim(x)

## S4 method for signature 'ExposomeSet'
expos(object)

## S4 method for signature 'ExposomeSet'
```

```
exposureNames(object)

## S4 method for signature 'ExposomeSet'
exwas(
  object,
  formula,
  filter,
  family,
  ...,
  baselevels,
  tef = TRUE,
  verbose = FALSE,
  warnings = TRUE,
  robust = FALSE
)

## S4 method for signature 'ExposomeSet'
familyNames(object, by.exposure = FALSE)

## S4 method for signature 'ExposomeSet'
highAndLow(
  object,
  ngroups = 3,
  intervals = c("standard", "extreme"),
  select,
  drop = FALSE,
  warnings = TRUE
)

## S4 method for signature 'ExposomeSet'
ilod(
  object,
  seed = 1234,
  lod.col = "LOD",
  pNA = 0.2,
  tLog = FALSE,
  method = "QRILC",
  warnings = TRUE,
  ...
)

## S4 method for signature 'ExposomeSet'
imputation(object, select, ..., messages = FALSE)

## S4 method for signature 'ExposomeSet'
invExWAS(object, formula, filter, tef = TRUE, verbose = FALSE, warnings = TRUE)

## S4 method for signature 'ExposomeSet'
mexwas(object, phenotype, family, warnings = TRUE)

## S4 method for signature 'ExposomeSet'
normalityTest(
```

```
    object,
    exposure,
    th = 0.05,
    min.val = 5,
    na.rm = TRUE,
    warnings = TRUE
)

## S4 method for signature 'ExposomeSet'
pca(object, npc = 10, pca = FALSE, ...)

## S4 method for signature 'ExposomeSet'
phenotypeNames(object)

## S4 method for signature 'ExposomeSet'
plotFamily(x, family, group, group2, scatter = TRUE, na.omit = TRUE)

## S4 method for signature 'ExposomeSet'
plotHistogram(x, select, density = TRUE, show.trans = FALSE)

## S4 method for signature 'ExposomeSet'
plotLOD(object, lod.col = "LOD", x.max = 100, sort = TRUE)

## S4 method for signature 'ExposomeSet'
plotMissings(
  object,
  set = c("exposures", "phenotypes"),
  x.max = 100,
  sort = TRUE
)

## S4 method for signature 'ExposomeSet'
standardize(object, select, method = "normal", na.rm = TRUE, warnings = TRUE)

## S4 method for signature 'ExposomeSet'
Summary(x, set = c("exposures", "phenotypes"), select, ..., na.rm = FALSE)

## S4 method for signature 'ExposomeSet'
tableLOD(object, output = "n", lod.col = "LOD", sort = TRUE)

## S4 method for signature 'ExposomeSet'
tableMissings(
  object,
  set = c("exposures", "phenotypes"),
  output = "n",
  sort = TRUE
)

## S4 method for signature 'ExposomeSet'
trans(object, fun, select, by.exposure = FALSE, ...)
```

Arguments

| | |
|-------------|-----------------------------------------------------------------------------------------|
| x | An ExposomeSet object. |
| y | NOT USED |
| ... | Arguments to be passed to imputeFAMD |
| object | An ExposomeSet object. |
| method | Method to be used. |
| cmethod | Function implementing a systems to retrieve classification from clustering output |
| warnings | If set to TRUE it prints the warning messages. |
| formula | Formula, not including exposures, to be tested. No need to provide response (left term) |
| filter | Expression to be used to filter ExposomeSet |
| family | Family describing the nature of the health outcome |
| baselevels | Labeled vector with the default base level for categorical exposures. |
| tef | If TRUE it computed the threshold for effective tests. |
| verbose | If set to TRUE is shows messages on progression. |
| by.exposure | If set to TRUE ir returns the family which each exposure belongs |
| ngroups | Number of intervals to be used |
| intervals | If "standard" all levels are kept, if "extreme" intermetiate levels are set to NA. |
| select | Vector selecting thee xposures to be used. |
| drop | If set to TRUE exposures are replaced |
| seed | Numeric seed |
| lod.col | Indicator of the column where the LOD is located |
| pNA | Maximum percentage allowed of values under LOD |
| tLog | If set to TRUE it transforms all the exposures to lod before the imputation. |
| messages | If set to TRUE messages from mice's function will be displayed. |
| phenotype | Health outcome to be used as dependent variable. |
| exposure | Vecror of exposures to be used. |
| th | Threshold of P-Value used to considere normalit |
| min.val | Minimum number of observations to perform test |
| na.rm | If set to TRUE removes NA values |
| npc | Number of PC to be kept |
| pca | Perform PCA (only numerical variables) or FAMD (numerical and categorical) |
| group | Phenotype to group exposures |
| group2 | Phenotype to group exposures |
| scatter | If set to true it shows the samples value in the plot |
| na.omit | If set to TRUE, NA values are discarded |
| density | If set to TRUE a desntiry plot is draw on the histogram |
| show.trans | If set to TRUE, three extra plots are drawn with usual transformations |
| x.max | Threshold for x axis (in %) |
| sort | If set to TRUE, results are ordered |
| set | Cantake values "exposures" or "phentotypes". |
| output | Can take values "n" (count) ot "p" (percentage) |
| fun | Function to bt used in the transformation process |

Value

An object of class `ExposomeSet`

Methods (by generic)

- `plot(x = ExposomeSet, y = ANY)`: Wrapper for `plotFamily` method.
- `clustering(ExposomeSet)`: Performs clustering on samples based on exposure levels.
- `correlation(ExposomeSet)`: Computes correlation on exposures.
- `dim(ExposomeSet)`: Returns the number of exsures, samples and phenotypes.
- `expos(ExposomeSet)`: Returns a `data.frame` with exposures.
- `exposureNames(ExposomeSet)`: Getter to obtain the exposures's names.
- `exwas(ExposomeSet)`: Performs an EXposome-Wide Association Study
- `familyNames(ExposomeSet)`: Getter to obtain the families's names of the family of each exposure.
- `highAndLow(ExposomeSet)`: Performs a discretization of continuous exposures.
- `ilod(ExposomeSet)`: Imputation of under-LOD values of exposures.
- `imputation(ExposomeSet)`: Imputation of missing values of exposures.
- `invExWAS(ExposomeSet)`: Performs an EXposome-Wide Association Study (modelling the exposures as response)
- `mexwas(ExposomeSet)`: Performs a Multiple-EXposure-Wide Association Study.
- `normalityTest(ExposomeSet)`: Test the normality of each exposure.
- `pca(ExposomeSet)`: Performs a PCA
- `phenotypeNames(ExposomeSet)`: Getter to obtain the phenotypes's names.
- `plotFamily(ExposomeSet)`: Draws a boxplot or accumulated-bar plot for each exposure in a given family.
- `plotHistogram(ExposomeSet)`: Draws an histogram of a given continuous exposure or a pie chart if a given categorycal exposure.
- `plotLOD(ExposomeSet)`: Draws a barchart with the amount of under-LOD values.
- `plotMissings(ExposomeSet)`: Draws a bar-plot with the amount of missing values.
- `standardize(ExposomeSet)`: Standardization of exposures.
- `Summary(ExposomeSet)`: Summary of both continuous and categorical exposures
- `tableLOD(ExposomeSet)`: Returns a vector with the number of under-LOD values per exposure.
- `tableMissings(ExposomeSet)`: Returns a vector with the number of missing values per exposure.
- `trans(ExposomeSet)`: Transformation of exposures.

Slots

`assayData` Contains the exposures matrix with column number equal to `nrow(phenoData)` (see [eSet](#), [AssayData](#)).

`featureData` Contains the description of the exposures including the family where they belong (see [eSet](#), [AnnotatedDataFrame](#)).

`phenoData` Contains the phenotypes or variables experimenter-supplied (see [eSet](#), [AnnotatedDataFrame](#)).

See Also

[readExposome](#) to create an ExposomeSet from files, [loadExposome](#) to create an ExposomeSet from `data.frames`

| | |
|---------------|--------------------------------------------------------------------------------|
| exposureNames | <i>Getter to obtain the exposures's names of an ExposomeSet or ExposomePCA</i> |
|---------------|--------------------------------------------------------------------------------|

Description

Getter to obtain the exposures's names of an ExposomeSet or ExposomePCA

Usage

```
exposureNames(object)
```

Arguments

object [ExposomeSet](#) that will be queried for the exposures's names.

Value

The name of the exposures as a character vector.

Warning

[exposureNames](#) collides with [featureNames](#) of [eSet](#). Although in `rexposome 1.0.0` both function can be used as synonyms, this usage is discouraged and it is not assured.

See Also

[phenotypeNames](#) to get the phenotypes, [familyNames](#) to get the families of exposures

Examples

```
data("exposome")
exposureNames(expo)
```

| | |
|--------|-------------------------------------------|
| expo_c | ExposomeClust <i>for testing purposes</i> |
|--------|-------------------------------------------|

Description

ExposomeClust created from an ExposomeSet with full set of 104 exposures, 1200 simulated samples and 4 phenotypes (asthma status, BMI measure, sex, and age). The clustering was done using hclust and cutree with k = 3.

Usage

```
data("eclust")
```

Format

An object of class ExposomeClust of dimension 99 x 1200 x 5.

Value

An ExposomeSet object.

Examples

```
data("eclust")
dim(expo_c)
table(classification(expo_c))
```

| | |
|---------|-------------------------------------------------------------|
| extract | <i>Raw data from ExWAS, ExposomeClust and ExposomeCorr.</i> |
|---------|-------------------------------------------------------------|

Description

Returns internal table of results of objects of class ExWAS, ExposomeClust and ExposomeCorr.

Usage

```
extract(object, ...)
```

Arguments

| | |
|--------|------------------------------------------------------|
| object | Object of class ExWAS, ExposomeClust or ExposomeCorr |
| ... | NO USED |

Value

A data.frame containing the raw result from PsyGeNET or a data.frame with the result Jaccard Index for each disease.

ExWAS

*Class ExWAS***Description**

Class ExWAS obtained from `exwas` method of an `ExposomeSet` object, contains the result of testing the association of exposures of an `ExposomeSet` to its phenotypes. "ExWAS" is the acronym of "Exposome-Wide Association Study". The function can be applied to one of to many phenotypes in the `ExposomeSet` object.

Usage

```
## S4 method for signature 'ExWAS,ANY'
plot(x, y, ...)

## S4 method for signature 'ExWAS'
extract(object, sort = TRUE, ...)

## S4 method for signature 'ExWAS'
get_robust_sd(object, sort = TRUE, ...)

## S4 method for signature 'ExWAS'
names(x)

## S4 method for signature 'ExWAS'
plotEffect(x, y, select, labels, xlab, ylab)

## S4 method for signature 'ExWAS'
plotExwas(
  object,
  ...,
  subtitles,
  color,
  exp.order,
  labels,
  show.effective = TRUE
)

## S4 method for signature 'ExWAS'
plotVolcano(x, p.value = -log10(0.001), labels, show.effect = FALSE)

## S4 method for signature 'ExWAS'
tef(object)
```

Arguments

| | |
|---------------------|---------------------------------------------------------------------------------------|
| <code>x</code> | An <code>ExWAS</code> object |
| <code>y</code> | An <code>ExWAS</code> object |
| <code>...</code> | NOT USED |
| <code>object</code> | An object of class <code>ExWAS</code> , <code>mExWAS</code> or <code>n1ExWAS</code> . |

| | |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sort</code> | If TRUE, the results are ordered by pvalue. |
| <code>select</code> | (optional) Vector with the selected exposures |
| <code>labels</code> | (optional) Character vector with the labels for each exposure. It must be labeled vector. |
| <code>xlab</code> | (optional) Label for x-axis |
| <code>ylab</code> | (optional) Label for y-axis |
| <code>subtitles</code> | Character vector with the "subtitle" for each plot when given more than one ExWAS. |
| <code>color</code> | (optional) A vector of colors. The vector must have length equal to the number of families. The vector must be names with the name of the families. |
| <code>exp.order</code> | (optional) Order of the exposures. |
| <code>show.effective</code> | (default TRUE) draws a brown line on the threshold given by the effective number of tests. |
| <code>p.value</code> | (default $-\log_{10}(0.001)$) Is the threshold from where the exposures can be taken as significant. |
| <code>show.effect</code> | (default false) Apply exp to obtained beta. |

Value

An object of class ExWAS

Methods (by generic)

- `plot(x = ExWAS, y = ANY)`: Wrapper for `plotExwas` method.
- `extract(ExWAS)`: Method to obtain the matrix of association scores
- `get_robust_sd(ExWAS)`: Method to obtain the matrix of association scores
- `names(ExWAS)`: Method to obtain the Threshold for effective tests (TEF)
- `plotEffect(ExWAS)`: Draws a plot with the confidence interval of each exposure. Allows to compare two ExWAS instances.
- `plotExwas(ExWAS)`: Method to plot a manhatan plot for association between exposures and phenotypes
- `plotVolcano(ExWAS)`: Method to plot a volcano plot for association between exposures and phenotypes
- `tef(ExWAS)`: Method to obtain the Threshold for effective tests (TEF)

Slots

`effective` Number containing the effective number of tests.

`formula` Tested formula.

`comparison` Result of performing the test to find association between levels of exposures and phenotype.

`description` Description of the exposures used in the ExWAS. (in description file).

See Also

[exwas](#) to perform an Exposome-Wide Association Study and to create an [ExWAS](#), [mexwas](#) to perform a Multivariate Exposome-Wide Association Study and to create a [mExWAS](#)

| | |
|-------|--------------------------------------------------------------------------------------|
| exwas | <i>Testing the association between an exposure and a phenotype of an ExposomeSet</i> |
|-------|--------------------------------------------------------------------------------------|

Description

The `exwas` method performs an "Exposome-Wide Association Study" (ExWAS) using the exposures in `ExposomeSet` and one of its phenotype.

Usage

```
exwas(
  object,
  formula,
  filter,
  family,
  ...,
  baselevels,
  tef = TRUE,
  verbose = FALSE,
  warnings = TRUE,
  robust = FALSE
)
```

Arguments

| | |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code> | ExposomeSet that will be used for the ExWAS. |
| <code>formula</code> | formula indicating the test to be done. If any exposure is included it will be used as covariate. <code>exwas</code> method will perform the test for each exposure. |
| <code>filter</code> | expression to be used to filter the individuals included into the test. |
| <code>family</code> | Family of the distribution followed by the health outcome to be tested (gaussian, binomial, ... check glm). |
| <code>...</code> | NOT USED |
| <code>baselevels</code> | (optional) If set, must be a labeled vector with the default base level for categorical exposures. |
| <code>tef</code> | (default TRUE) If TRUE it computed the effective number of tests and the threshold for the effective number of tests. Usually it needs imputed data. |
| <code>verbose</code> | (default FALSE) If set to true messages along the tests are shown. |
| <code>warnings</code> | (default TRUE) If set to FALSE warnings will not be displayed. |

Value

An `codeExWAS` object with the result of the association study

References

An Environment-Wide Association Study (ExWAS) on Type 2 Diabetes Mellitus. Chirag J. Patel, Jayanta Bhattacharya, Atul J. Butte. May 20, 2010 Plos One

Evaluating the effective numbers of independent tests and significant p-value thresholds in commercial genotyping arrays and public imputation reference datasets. Miao-Xin Li, Juilian M. Y. Yeung, Stacey S. Cherny and Pak C. Sham. May 2012 Hum Genet.

See Also

[extract](#) to obtain a table with the result of the ExWAS, [plotExwas](#) to plot the results of the ExWAS

Examples

```
data(exposome)
w1 <- exwas(expo[1:5, ], asthma~1, family = "binomial")
w2 <- exwas(expo[1:5, ], asthma~sex+age, family = "binomial")
w3 <- exwas(expo[1:5, ], asthma~age, family = "binomial", filter = sex=="boy")
```

| | |
|--------|-------------------------------------------|
| ex_imp | imExposomeSet <i>for testing purposes</i> |
|--------|-------------------------------------------|

Description

imExposomeSet created from the raw data included in the package. The creation process is illustrated in the vignette.

Usage

```
data("ex_imp")
```

Format

An object of class imExposomeSet of dimension 47 x 109 x 10.

Value

An imExposomeSet object.

Examples

```
data("ex_imp")
ex_imp
```

| | |
|-------------|------------------------------------------------------------------|
| familyNames | <i>Getter to obtain the exposures's names of an ExposomeSet.</i> |
|-------------|------------------------------------------------------------------|

Description

This method returns the name of the families in an [ExposomeSet](#), but it can return a vector, labeled with the exposures in the [ExposomeSet](#), containing the family belonging to each exposure.

Usage

```
familyNames(object, by.exposure = FALSE)
```

Arguments

- `object` `ExposomeSet` that will be queried for the exposures's family-names.
- `by.exposure` (default FALSE) If TRUE a vector labeled with each exposure name will be returned with the family of each exposures. If FALSE a vector with the (unique) name of the families of exposures will be returned.

Value

The families of the exposures into the [ExposomeSet](#), or the family of each exposure into the [ExposomeSet](#).

See Also

[exposureNames](#) to get the name of the exposures, [phenotypeNames](#) to get the phenotypes

Examples

```
data("exposome")
# Get families
familyNames(expo)
# Get the family of each exposure
familyNames(expo, by.exposure = TRUE)
```

| | |
|---------------|----------------------------|
| get_robust_sd | <i>Raw data from ExWAS</i> |
|---------------|----------------------------|

Description

Returns internal table of robust SD of objects of class `ExWAS`

Usage

```
get_robust_sd(object, ...)
```

Arguments

- `object` Object of class `ExWAS`
- `...` NO USED

Value

A `data.frame` containing the raw result from robust SD

highAndLow

*Function to convert continuous exposures to categorical exposures***Description**

This method allows to convert continuous exposures of an `ExposomeSet` to categorical exposures using the n-percentile groups, defined by `ngroups` argument. By default, all levels are kept but if `intervals` is set to "extrem", the levels between the extremes (aka. lowes and highest) are discarded and their values set to NA.

Usage

```
highAndLow(
  object,
  ngroups = 3,
  intervals = "standard",
  select,
  drop = FALSE,
  warnings = TRUE
)
```

Arguments

| | |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code> | An object of class <code>ExposomeSet</code> . |
| <code>ngroups</code> | (default 3) Number of intervals to be created. |
| <code>intervals</code> | (default "standard") If set to "sctandard" all levels are set. If set to "extreme" obly lowes and highest levels are kept (others are set to NA). |
| <code>select</code> | (optional) Subset of exposures where the discretization is applied. If missing, all exposures are used. |
| <code>drop</code> | (default FALSE) If set to FALSE original exposures are kept and discretized exposures are add to <code>ExposomeSet</code> . If set to TRUE, original exposures are replaced by categorical exposures. |
| <code>warnings</code> | (defaultr TRUE) If set to FALSE warnings are not shown. |

Value

A new `ExposomeSet` with categorical exposures.

See Also

[trans](#) to transform exposures, [standardize](#) to standardize exposures.

Examples

```
# No drop
data("exposome")
exp.hl <- highAndLow(expo, intervals = "standard", select = "ldde_lip")
dim(exp.hl)
# exposures  samples phenotyes
#      105      1200      4
dim(expo)
```

```

# exposures  samples phenotypes
#      104      1200          4
# exps.hl has 107 exposures: the original 104 plus the new
#           3 factored exposures

# Drop
exp.hl <- highAndLow(expo, intervals = "standard",
  select = "ldde_lip", drop = TRUE)
dim(exp.hl)
# exposures  samples phenotypes
#      104      1200          4
dim(expo)
# exposures  samples phenotypes
#      104      1200          4

```

ilod

Function to impute under-LOD values from an ExposomeSet

Description

This function is a wrapper of the functions `impute.MinProb` from the package `imputeLCMD`.

Usage

```

ilod(
  object,
  seed = 1234,
  lod.col = "LOD",
  pNA = 0.2,
  tLog = FALSE,
  method = "QRILC",
  warnings = TRUE,
  ...
)

```

Arguments

| | |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code> | ExposomeSet which exposures will be imputed. |
| <code>seed</code> | (default 1234) Seed to make the imputation reproducible. |
| <code>lod.col</code> | (default "LOD") Name of the column in <code>fData</code> with the threshold of the LOD. |
| <code>pNA</code> | (default 0.2) Maximum percentage allowed of values under LOD |
| <code>tLog</code> | (default FALSE) If set to TRUE it transforms all the exposures to lod before the imputation. |
| <code>method</code> | (default "QRILC") Method to be used to impute the under-LOD values. Two allowed: QRILC method (value "QRILC") and stochastic minimal value approach (value "MinProb"). |
| <code>warnings</code> | (default TRUE) If set to FALSE warnings will not be displayed. |
| <code>...</code> | Arguments passed to <code>impute.QRILC</code> or <code>impute.MinProb</code> from <code>imputeLCMD</code> . |

Value

A new ExposomeSet with the imputed exposures.

See Also

[plotMissings](#) to plot the missing data of an ExposomeSet, [tableMissings](#) to get a table with the missing data of an ExposomeSet

Examples

```
## Not run:
#Being x an ExposomeSet
x <- ilod(x)

## End(Not run)
```

| | |
|---------------|----------------------------|
| imExposomeSet | <i>Class imExposomeSet</i> |
|---------------|----------------------------|

Description

Class imExposomeSet was designed to store the exposures obtained after a multiple imputation process done using mice. The data.frame obtained from mice has the particularity to contain the columns .imp and .id joint to phenotypes and exposures. The slots assayData, featureData, and phenoData are coordinated.

Usage

```
## S4 method for signature 'imExposomeSet,ANY'
plot(x, y, ...)

## S4 method for signature 'imExposomeSet'
dim(x)

## S4 method for signature 'imExposomeSet'
expos(object)

## S4 method for signature 'imExposomeSet'
exposureNames(object)

## S4 method for signature 'imExposomeSet'
extract(object, rid = -1, ...)

## S4 method for signature 'imExposomeSet'
exwas(
  object,
  formula,
  filter,
  family,
  ...,
  baselevels,
```

```

    tef = TRUE,
    verbose = FALSE,
    warnings = TRUE,
    robust = FALSE
)

## S4 method for signature 'imExposomeSet'
familyNames(object, by.exposure = FALSE)

## S4 method for signature 'imExposomeSet'
phenotypeNames(object)

## S4 method for signature 'imExposomeSet'
plotFamily(x, family, group, group2, scatter = FALSE, na.omit = TRUE)

## S4 method for signature 'imExposomeSet'
sampleNames(object)

## S4 method for signature 'imExposomeSet'
x[[i]]

## S4 method for signature 'imExposomeSet,ANY,ANY,ANY'
x[i, j, k, ..., drop = FALSE]

## S4 method for signature 'imExposomeSet'
toES(object, rid = 1)

```

Arguments

| | |
|-------------|------------------------------------------------------------------|
| x | An imExposomeSet object. |
| y | NOT USED |
| ... | Other used arguments |
| object | An object of class imExposomeSet |
| rid | Number of the imputation to be extracted |
| formula | Formula, not including exposures, to be tested. |
| filter | Expression to be used to filter ExposomeSet |
| family | Family describing the nature of the health outcome |
| baselevels | Labeled vector with the base-level of the categorical exposures |
| tef | If set to TRUE the threshold for effective test is computed. |
| verbose | If set to TRUE it shows messages on progression. |
| warnings | If set to TRUE it shows warnings on progression. |
| by.exposure | If set to TRUE it returns the family which each exposure belongs |
| group | Phenotype to group exposures |
| group2 | NOT USED |
| scatter | If set to true it shows the samples value in the plot |
| na.omit | NOT USED |
| i | Character corresponding to selected exposures. |

| | |
|------|---------------------------------------------------|
| j | Character corresponding to selected sample names. |
| k | Character corresponding to selected phenotypes. |
| drop | NOT USED |

Value

An object of class imExposomeSet

Methods (by generic)

- `plot(x = imExposomeSet, y = ANY)`: Wrapper for `plotFamily` method.
- `dim(imExposomeSet)`: Returns the number of exsures, samples and phenotypes.
- `expos(imExposomeSet)`: Returns a `data.frame` with exposures.
- `exposureNames(imExposomeSet)`: Method to obtain samples' names
- `extract(imExposomeSet)`: Method to extract exposures for a single imputation
- `exwas(imExposomeSet)`: Performs an EXposome-Wide Association Study
- `familyNames(imExposomeSet)`: Getter to obtain the families's names of the family of each exposure.
- `phenotypeNames(imExposomeSet)`: Getter to obtain the families's names of the family of each exposure.
- `plotFamily(imExposomeSet)`: Draws a boxplot or accumulated-bar plot for each exposure in the all imputed sets.
- `sampleNames(imExposomeSet)`: Method to obtain samples' names
- `[[]`: Get an `ExposomeSet` with the selected imputation
- `x[i]`: Subset an `imExposomeSet`
- `toES(imExposomeSet)`: Returns an [ExposomeSet](#) with ethe given imputation.

Slots

`nimputation` Number of imputations done with mice.

`assayData` `data.frame` containing `.imp`, `.id` and the exposures.

`featureData` `data.frame` containing the description of the exposures.

`phenoData` `data.frame` containing `.imp`, `.id` and the phenotypes.

Note

Sample order is not guarantee

See Also

[loadImputed](#) to create an `imExposomeSet` from `data.frames`

| | |
|------------|--------------------------------------------------------------|
| imputation | <i>Function to impute missing values from an ExposomeSet</i> |
|------------|--------------------------------------------------------------|

Description

This function is a wrapper of the functions `mice` and `complete` from the package `mice`. Also to the `impute` from the package `Hmisc`. The function is designed to use those functions to impute missing values on exposures (not in phenotypes).

Usage

```
imputation(object, select, ..., messages = FALSE)
```

Arguments

| | |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code> | ExposomeSet which exposures will be imputed. |
| <code>select</code> | Exposures to be imputed. If missing, all exposes will be imputed. |
| <code>...</code> | Argument given to function <code>mice</code> of package <code>mice</code> (<code>printFlag</code> is set from <code>messages</code>). |
| <code>messages</code> | (default FALSE) If set to TRUE messages from <code>mice</code> 's function will be displayed. |

Value

A new ExposomeSet with the imputed exposures.

See Also

[plotMissings](#) to plot the missing data of an ExposomeSet, [tableMissings](#) to get a table with the missing data of an ExposomeSet

Examples

```
## Not run:
#Being x an ExposomeSet
x <- imputation(x)

## End(Not run)
```

| | |
|-----------|-----------------------------------------------------------|
| imputeLOD | <i>Function to impute values under limit of detection</i> |
|-----------|-----------------------------------------------------------|

Description

Homologous function to the `ilod` method of `ExposomeSet` for a given data.frame of exposures and a vector of threshold. The use of this function is related to HELIX Project.

Usage

```
imputeLOD(
  x,
  lod,
  description = NULL,
  pNA = 0.2,
  pLOD = 0,
  log = TRUE,
  seed = NULL
)
```

Arguments

| | |
|-------------|-------------------------------------------------------------------------|
| x | data.frame containing the exposures as columns and the samples as rows. |
| lod | vector containing the |
| description | 1 means values under LOD while 2 means quantifiable value, 3 |
| pNA | (default: 0.2) maximum percentage of allowed missing data |
| pLOD | (default: 0) minimum percentage of values under LOD |
| log | (default: NA) log transformation to normalize data |
| seed | (default: NULL) |

Value

A new data.frame with the imputed exposures.

Examples

```
## Not run:
inma.imp <- imputeLOD(x = raw, lod = lod, description = desc,
  pNA = pNA, pLOD = pLOD, log = log, seed = seed)

## End(Not run)
```

| | |
|----------|----------------------------------------------------------------------------------------------------------------------------|
| invExWAS | <i>Testing the association between an exposure and a phenotype of an ExposomeSet (modelling the exposures as response)</i> |
|----------|----------------------------------------------------------------------------------------------------------------------------|

Description

The invExWAS method performs an "Exposome-Wide Association Study" (ExWAS) using the exposures in [ExposomeSet](#) and one of its phenotype. (modelling the exposures as response)

Usage

```
invExWAS(object, formula, filter, tef = TRUE, verbose = FALSE, warnings = TRUE)
```

Arguments

| | |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeSet that will be used for the ExWAS. |
| formula | formula indicating the test to be done. If any exposure is included it will be used as covariate. exwas metho will perform the test for each exposure. |
| filter | expression to be used to filter the individuals included into the test. |
| tef | (default TRUE) If TRUE it computed the effective number of tests and the threshold for the effective number of tests. Usually it needs imputed data. |
| verbose | (default FALSE) If set o true messages along the tests are shown. |
| warnings | (default TRUE) If set to FALSE warnings will not be displayed. |

Value

An codeExWAS object with the result of the association study

See Also

[extract](#) to obtain a table with the result of the ExWAS, [plotExwas](#) to plot the results of the association

Examples

```
data(exposome)
w1 <- invExWAS(expo, ~BMI)
w2 <- invExWAS(expo, ~BMI + sex)
plotExwas(w1, w2)
```

loadExposome

Creation of an ExposomeSet from data.frames

Description

Given three data.frames that defines the exposome (measures of exposome, exposome description and individuals phentype) it loads them and creates an object of type [ExposomeSet](#).

Usage

```
loadExposome(
  exposures,
  description,
  phenotype,
  description.famCol = "family",
  exposures.asFactor = 5,
  warnings = TRUE
)
```

Arguments

| | |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| exposures | data.frame of exposures. |
| description | data.frame with the description of the exposures (relation between exposures and exposure-family). |
| phenotype | data.frame with the phenotypes of interest. |
| description.famCol | (default "family") Index where the family's name (per exposures) is found in file "description". It can be both numeric or character. |
| exposures.asFactor | (default 5) The exposures with more than this number of unique items will be considered as "continuous" while the exposures with less or equal number of items will be considered as "factor". |
| warnings | (default TRUE) If TRUE shows useful information/warnings from the process of loading the exposome. |

Details

The rows of the exposure's data.frames, that corresponds to samples' names, must be the same than the phenotype's data.frames. In the same way, the columns in exposure's data.frames must be the same in description data.frame.

Value

An object of class [ExposomeSet](#).

Note

[ExposomeSet](#)'s fData will contain some inner columns called .std, .trn, .fct and .type in order to trace the transformations an exposure suffers and to know, at any moment, if an exposure is categorical or continuous. The "description" file can contain a column called type with values "factor" and "numeric" to specify how an exposure needs to be understood. If given, this column will be renamed to .type. If not given, it will be created using exposures.asFactor value.

See Also

[ExposomeSet](#) for class description, [readExposome](#) for constructor from txt/csv files.

Examples

```
path <- file.path(path.package("rexposome"), "extdata")
description <- file.path(path, "description.csv")
phenotype <- file.path(path, "phenotypes.csv")
exposures <- file.path(path, "exposures.csv")
dd <- read.csv(description, header=TRUE)
ee <- read.csv(exposures, header=TRUE)
pp <- read.csv(phenotype, header=TRUE)
rownames(dd) <- dd[, 2]
dd <- dd[, -2]
rownames(ee) <- ee[, 1]
ee <- ee[, -1]
rownames(pp) <- pp[, 1]
pp <- pp[, -1]
exp <- loadExposome(
```

```

exposures = ee,
description = dd,
phenotype = pp,
description.famCol = "Family"
)

```

loadExposome_plain *Creation of an ExposomeSet from single data.frame*

Description

Creation of an ExposomeSet from single data.frame

Usage

```

loadExposome_plain(
  data,
  data_id,
  sep = ",",
  pheno_cols,
  na.strings = c("NA", "-", "?", " ", "" ),
  families = NULL,
  exposures.asFactor = 5,
  warnings = TRUE
)

```

Arguments

| | |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data | data.frame With the exposures and phenotypes (in no particular order!) or string with the path to a file (.csv, .tsv, .txt) with the table of exposures and phenotypes |
| data_id | character Name of the column on the selected table that contains the ID |
| sep | character (default ",") Separator used by read.table to load the files "exposures", "description" and "phenotype". Only applies when providing a path on the data argument |
| pheno_cols | character Character vector of the phenotype columns (all the other columns are considered exposures) |
| na.strings | character (default c("NA", "-", "?", " ", "")) Character defining the NA values in expsome's files. |
| families | list (default NULL) List to specify the families of the exposures, construct it as: list(Family1 = c("exposure_1_1", "exposure_1_2", "exposure_1_n"), Family2 = c("exposure_2_1", "exposure_2_2", "exposure_2_n"), FamilyM = c("exposure_M_1", "exposure_M_2", "exposure_M_n")). All the exposures on the data table have to be on this provided list with their respective families. The family classification is optional, input NULL to bypass the family classifier |
| exposures.asFactor | numeric (default 5) The exposures with more than this number of unique items will be considered as "continuous" while the exposures with less or equal number of items will be considered as "factor". |
| warnings | (default TRUE) If TRUE shows useful information/warnings from the process of loading the exposome. |

Value

An object of class `ExposomeSet`.

Examples

```
path <- file.path(path.package("rexposome"), "extdata")
phenotype <- file.path(path, "phenotypes.csv")
exposures <- file.path(path, "exposures.csv")
ee <- read.csv(exposures, header=TRUE)
pp <- read.csv(phenotype, header=TRUE)
# Create fake dataset with exposures and phenotypes combined
data <- cbind(ee, pp)

loadExposome_plain <- function(data, data_id = "idnum",
  pheno_cols = c("rhinitis", "wheezing", "sex", "age", "cbmi", "blood_pre", "whistling_chest", "flu"))
```

loadImputed

Creation of an imExposomeSet from data.frames

Description

Given a `data.frame` from code with the multiple imputations of both exposures and phenotypes, join with a `data.frame` with exposures' description, and object of class `imExposomeSet` is created.

Usage

```
loadImputed(
  data,
  description,
  description.famCol = "family",
  description.expCol = "exposure",
  exposures.asFactor = 5,
  warnings = TRUE
)
```

Arguments

| | |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>data</code> | The <code>data.frame</code> of both exposures and phenotypes obtained from mice. |
| <code>description</code> | <code>data.frame</code> with the description of the exposures (relation between exposures and exposure-family). |
| <code>description.famCol</code> | (default "family") Index where the family's name (per exposures) if found in file "description". It can be both numeric or character. |
| <code>description.expCol</code> | (default "exposures") Index where the exposure's name if found in file "description". It can be both numeric or character. |
| <code>exposures.asFactor</code> | (default 5) The exposures with more than this number of unique items will be considered as "continuous" while the exposures with less or equal number of items will be considered as "factor". |
| <code>warnings</code> | (default TRUE) If TRUE shows useful information/warnings from the process of loading the exposome. |

Details

The coordination of the information is based in the columns `.imp` and `.id` from the `data.frame` obtained from `mice`. The division of exposures and phenotypes is based in `description.data.frame`, that are the exposures. Hence, the elements in the main `data.frame` that are not in the `description`, are the phenotypes.

Value

An object of class `imExposomeSet`.

See Also

`imExposomeSet` for class description

Examples

```
data("me") # me is an imputed matrix of exposure and phenotypes
path <- file.path(path.package("rexposome"), "extdata")
description <- file.path(path, "description.csv")
dd <- read.csv(description, header=TRUE, stringsAsFactors=FALSE)
dd <- dd[dd$Exposure %in% colnames(me), ]
ex_imp <- loadImputed(data = me, description = dd,
description.famCol = "Family",
description.expCol = "Exposure")
```

me

data.frame for testing purposes

Description

`data.frame` created with `mice` used to test and illustrate the creation of `imExposomeSet`.

Usage

```
data("me")
```

Format

An object of class `data.frame` with 654 rows and 57 columns.

Value

A `data.frame`.

Examples

```
data("me")
dim(me)
colnames(me)
```

| | |
|--------|---------------------|
| mExWAS | <i>Class mExWAS</i> |
|--------|---------------------|

Description

Class mExWAS obtained from `mexwas` method of an `ExposomeSet` object, contains the result of testing the multiple models of exposures of an `ExposomeSet` to a set of given phenotypes. "mExWAS" is the #' acronym of "Multivariate Exposome-Wide Association Study".

Usage

```
## S4 method for signature 'mExWAS,ANY'
plot(x, y, ...)

## S4 method for signature 'mExWAS'
extract(object, type = "test", sort = TRUE)

## S4 method for signature 'mExWAS'
plotExwas(
  object,
  ...,
  subtitles,
  color,
  exp.order,
  labels,
  show.effective = TRUE
)
```

Arguments

| | |
|----------------|-------------------------------------------------------------------------------------|
| x | Object of class mExWAS |
| y | NOT USED |
| ... | Other used arguments. |
| object | object of class mExWAS. |
| type | (default "test"). Can take "test" or "raw" to obtain a data.frame of glmnet result. |
| sort | NOT USED |
| subtitles | NOT USED |
| color | NOT USED |
| exp.order | NOT USED |
| labels | NOT USED |
| show.effective | NOT USED |

Value

An object of class mExWAS

Methods (by generic)

- `plot(x = mExWAS, y = ANY)`: Wrapper for `plotExwas` method.
- `extract(mExWAS)`: Method to obtain the generated model
- `plotExwas(mExWAS)`: Method to plot a heap-map with the coefficient of each exposure

Slots

`result` klist with the fitted model and result.

`phenotype` Name of the phenotype used in the analysys.

`description` feature data from original `ExposomeSet`.

See Also

[mexwas](#) to perform a Multivariate Exposome-Wide Association Study and to create a `mExWAS`, [exwas](#) to perform an Exposome-Wide Association Study and to create an `ExWAS`

| | |
|--------|---------------------------------------------------------------------------------------------------------------------|
| mexwas | <i>Testing the association between an exposure and a phenotype of an ExposomeSet using a multivariate approach.</i> |
|--------|---------------------------------------------------------------------------------------------------------------------|

Description

The `mexwas` method performs an "Multi Exposome-Wide Association Study" (m-ExWAS) using the exposures in [ExposomeSet](#) and one of its phenotype. It uses the packages `glmnet` and `partDSA`.

Usage

```
mexwas(object, phenotype, family, warnings = TRUE)
```

Arguments

| | |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code> | ExposomeSet that will be used for the ExWAS. |
| <code>phenotype</code> | Target phenotype used for the study. If missing all the phenotypes in the ExposomeSet will be used. |
| <code>family</code> | It must decribe the nature of the outcome. Can take values "gaussian", "binomial", "poisson", "multinomial", "cox" or "mgaussian". |
| <code>warnings</code> | (default TRUE) If set to FALSE warnings will not be displayed. |

Value

Returns an object of class [mExWAS](#)

See Also

[extract](#) to obtain a table with the result of the ExWAS, [plotExwas](#) to plot the results of the ExWAS

Examples

```
data("exposome")
wt <- mexwas(expo[3:7, 1:100], phenotype = "asthma", family = "binomial")
```

| | |
|------|----------------------------------------------------------|
| ndim | <i>Number of principal components in an ExposomePCA.</i> |
|------|----------------------------------------------------------|

Description

Number of principal components in an ExposomePCA.

Usage

```
ndim(object)
```

Arguments

object ExposomePCA to obtain the number of components.

Value

The number of components in the ExposomePCA.

See Also

[plotPCA](#) to plot the PCA values of an [ExposomePCA](#), [clustering](#) to see how the exposures can cluster samples, [correlation](#) to compute the correlation between exposures

Examples

```
data("exposome")
epca <- pca(expo[3:7, 1:100])
ndim(epca)
```

| | |
|---------------|-------------------------------------------------------------------------------------------------------------|
| normalityTest | <i>It creates a data.frame of boolean indicating if the exposures follows a normal distribution or not.</i> |
|---------------|-------------------------------------------------------------------------------------------------------------|

Description

This functions uses [shapiro.test](#) to test the normality of the exposures and returns a data.frame with a boolean value and a p-value for each exposure.

Usage

```
normalityTest(  
  object,  
  exposure,  
  th = 0.05,  
  min.val = 5,  
  na.rm = TRUE,  
  warnings = TRUE  
)
```

Arguments

| | |
|----------|---------------------------------------------------------------------------------------|
| object | ExposomeSet with the exposome to be tested. |
| exposure | Name of the exposure to be tested, if missing all the exposures will be tested. |
| th | (by default 0.05) Threshold to considere an exposure to follow a normal distribution. |
| min.val | (by default 5) Minimum number of values not missings to test the exposures. |
| na.rm | (by default TRUE) Removes the NA values to test the normality on the exposure. |
| warnings | (by default: TRUE) Show warnings if required. |

Value

A data.frame with three columns: exposure, normality and p.value. "exposure" column contains the name of each exposure. "normalty" column contains a logical value indicating if the obtained p-value is under the given threshold. "p.value" column contains the obtained p-value from [shapiro.test](#).

See Also

[plotHistogram](#) to draw the shape of an exposure, [plotMissings](#) to draw a plot with the missing data an [ExposomeSet](#), [imputation](#) to impute missing data on the exposures of an [ExposomeSet](#)

Examples

```
data("exposome")
normalityTest(expo)
```

pca

Creation of an [ExposomePCA](#) from an [ExposomeSet](#).

Description

Method to calculate a PCA based on the exposures of an [ExposomeSet](#). Only numerical-exposures (non categorical) will be computed. The function returns an [ExposomePCA](#) object. This PCA is performed by using [FactoMineR](#) package.

Usage

```
pca(object, npc = 10, pca = FALSE, ...)
```

Arguments

| | |
|--------|----------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeSet which exposures will be used for the PCA |
| npc | (by default 10) number of dimensions kept in the results |
| pca | (default FALSE) Set to TRUE to Perform PCA (only numerical variables) or FALSE to perform FAMD (numerical and categorical) |
| ... | Arguments to be passed to imputeFAMD |

Value

An [ExposomePCA](#) with the values of the PCA.

See Also

[plotPCA](#) to plot the PCA values of an [ExposomePCA](#), [clustering](#) to see how the exposures can cluster samples, [correlation](#) to compute the correlation between exposures

Examples

```
data("exposome")
epca <- pca(expo[12:20, ])
```

| | |
|----------------|---------------------------------------------------------------------------------|
| phenotypeNames | <i>Getter to obtain the phenotype's names of an ExposomeSet or ExposomePCA.</i> |
|----------------|---------------------------------------------------------------------------------|

Description

Getter to obtain the phenotype's names of an ExposomeSet or ExposomePCA.

Usage

```
phenotypeNames(object)
```

Arguments

object ExposomeSet that will be queried for the phenotype's names.

Value

The name of the phenotypes as a character vector.

See Also

[exposureNames](#) to get the name of the exposures, [familyNames](#) to get the families of exposures

Examples

```
data("exposome")
phenotypeNames(expo)
```

`plot3PCA`*Plotting PCA in a 3D space*

Description

Method to draw a plot for samples using three PC contained in an [ExposomePCA](#)

Usage

```
plot3PCA(  
  object,  
  cmpX,  
  cmpY,  
  cmpZ,  
  phenotype,  
  main,  
  angle = 35,  
  pch = 16,  
  legend = TRUE,  
  plines = TRUE  
)
```

Arguments

| | |
|------------------------|----------------------------------------------------------------------------|
| <code>object</code> | An object of class ExposomePCA |
| <code>cmpX</code> | Component to be placed at X axis |
| <code>cmpY</code> | Component to be placed at Y axis |
| <code>cmpZ</code> | Component to be placed at Z axis |
| <code>phenotype</code> | Used to color samples by phenotype |
| <code>main</code> | Title for the plot |
| <code>angle</code> | (default 35) angle between x and y axis. |
| <code>pch</code> | (default 16) plotting "character", i.e. symbol to use. |
| <code>legend</code> | (default TRUE) If TRUE shows the legend. |
| <code>plines</code> | (default TRUE) If TRUE it draws the lines from each dot to the base plane. |

Value

A list with different graphics option from `scatterplot3d`.

See Also

[pca](#) to compute PCA on an [ExposomeSet](#), [plotPCA](#) to plot the PCA, [ExposomePCA](#) as main class
`data("exposome") epca <- pca(expo[3:7, 1:100]) plot3PCA(epca, cmpX = 1, cmpY = 2, cmpZ = 3,
phenotype = "sex")`

| | |
|--------------------|-------------------------------------------------------------------------------------------------|
| plotClassification | <i>Draw the profile of the levels exposures after a classification with a clustering method</i> |
|--------------------|-------------------------------------------------------------------------------------------------|

Description

Draw the profile of the levels exposures after a classification with a clustering method

Usage

```
plotClassification(object, type = "heatmap", ...)
```

Arguments

| | |
|--------|---------------------------------------------------|
| object | Object of class <code>ExposomeClust</code> |
| type | Two types are available: "heatmap" or "valuemap". |
| ... | NOT USED |

Value

A list with different graphics parameters.

See Also

[clustering](#) as a constructor for `ExposomeClust`, [classification](#) to see how to obtain the classification of the samples from an `ExposomeClust`

Examples

```
## Not run:
data("eclust")
plotClassification(expo_c)

## End(Not run)
```

| | |
|-----------------|---------------------------------------------------------------------------------------------|
| plotCorrelation | <i>It draws both circos or matrix plot for the correlation in <code>ExposomeCorr</code></i> |
|-----------------|---------------------------------------------------------------------------------------------|

Description

While the circos plot can be used to see the general behaviours intra and extra families of exposures, the matrix plot allows for a detailed view of the correlations within an `ExposomeCorr` object.

Usage

```
plotCorrelation(object, type = "circos", ...)
```

Arguments

| | |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeCorr which correlations will be plotted. |
| type | (default "circos") Can take both "circos" or "matrix". |
| ... | Arguments given to corrplot of package corrplot if a matrix is draw. Moreover extra arguments are can be passed to inner functions to draw both the matrix and the circos of correlations. |

Value

A list with different graphics parameters.

See Also

[correlation](#) as a constructor for [ExposomeCorr](#) objects, [pca](#) to compute PCA on exposures

Examples

```
data("exposome")
expo.c <- correlation(expo)
plotCorrelation(expo.c, type="circos")
plotCorrelation(expo.c, type="matrix")
```

plotEffect

Function to draw a plot of the pvalues stored in an ExWAS object

Description

This function draws a sort of manhattan plots using the p-value of the association of the exposures with phenotypes of an ExWAS object.

Usage

```
plotEffect(x, y, select, labels, xlab, ylab)
```

Arguments

| | |
|--------|-------------------------------------------------------------------------------------------|
| x | An ExWAS object which effect will be plotted. |
| y | (optional) Another ExWAS object. If provided its effects will be plotted in Y-axis. |
| select | (optional) Character with exposures to be shown. |
| labels | (optional) Character vector with the labels for each exposure. It must be labeled vector. |
| xlab | (optional) Label for X-axis. |
| ylab | (optional) Label for Y-axis. |

Value

An object of class ggplot.

See Also

[exwas](#) as a constructor for [ExWAS](#) objects, [extract](#) to obtain a table with the result of the ExWAS

Examples

```
data(exposome)
w1 <- exwas(expo[1:5, ], asthma~1, family = "binomial")
w2 <- exwas(expo[1:5, ], asthma~sex+age, family = "binomial")
plotEffect(w1, w2)
```

plotEXP

Plot correlation between exposures and PCA

Description

Function used to plot the correlation between the exposures in an [ExposomePCA](#) and the values for each component of the PCA in the same [ExposomePCA](#)

Usage

```
plotEXP(object, exposure)
```

Arguments

object An object of class [ExposomePCA](#)
exposure (optional) to select a set of exposures to be plotted. If not given all are used.

Value

An object of class ggplot.

See Also

[pca](#) to compute PCA on an [ExposomeSet](#), [plotPHE](#) to plot the P-Value of association between phenotypes and PCA, [ExposomePCA](#) as main class

Examples

```
data("exposome")
epca <- pca(expo[3:7, 1:100])
plotEXP(epca)
```

plotExwas

Function to draw a plot of the pvalues stored in an ExWAS object

Description

This function draws a sort of manhattan plots using the p-value of the association of the exposures with phenotypes of an ExWAS object.

Usage

```
plotExwas(
  object,
  ...,
  subtitles,
  color,
  exp.order,
  labels,
  show.effective = TRUE
)
```

Arguments

| | |
|----------------|--------------------------------------------------------------------------------------------------------|
| object | An ExWAS object which p-values will be plotted. |
| ... | Other objects of class ExWAS. |
| subtitles | (optional) Characters used as "subtitle" when more than one ExWAS is given. |
| color | (optional) Character vector of HTML colors, labeled with family's names. Used to colore the exposures. |
| exp.order | (optional) Character vector of exposures used to order and subset the plot. |
| labels | (optional) Character vector with the labels for each exposure. It must be labeled vector. |
| show.effective | (default TRUE) If set to FALSE, line showing effective test threshold is not shown. |

Value

An object of class ggplot.

See Also

[exwas](#) as a constructor for [ExWAS](#) objects, [extract](#) to obtain a table with the result of the ExWAS

Examples

```
data(exposome)
w1 <- exwas(expo[1:5, ], asthma~1, family = "binomial")
plotExwas(w1)
```

| | |
|------------|---------------------------------------------------------------|
| plotFamily | <i>It draws the profile of the exposome in an ExposomeSet</i> |
|------------|---------------------------------------------------------------|

Description

This function draw a profile of the full exposome into an [ExposomeSet](#) or the profile of the exposures in a specific family. For continuous families, box-plots are drawn; while for categorical families accumulative bar-charts.

Usage

```
plotFamily(x, family, group, group2, scatter = TRUE, na.omit = TRUE)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------|
| x | ExposomeSet which exposome will be plotted. |
| family | Name of the family that will be drawn. 'all' is allowed to draw a grid with all the families. |
| group | If set it displays the family grouped by the given phenotype. |
| group2 | If set it displays the family grouped by the given phenotype. |
| scatter | (default TRUE) If the family to be plotted is continuous, the samples will be shown. |
| na.omit | (default TRUE) Do not show NA values. |

Value

A ggplot object if a family was selected. `invisible()` if argument family was set to "all".

See Also

[plotHistogram](#) to draw the shape of an exposure, [plotMissings](#) to plot the missing data from an [ExposomeSet](#)

Examples

```
data("exposome")
plt <- plotFamily(expo, family = "Metals")
plt <- plt + ggplot2::ggtitle("Metals")
plt
plt <- plotFamily(expo, family = "Indoor air")
plt <- plt + ggplot2::ggtitle("Indoor air")
plt
```

| | |
|---------------|-----------------------------------------------------------------|
| plotHistogram | <i>It draws a histogram for each exposure in an ExposomeSet</i> |
|---------------|-----------------------------------------------------------------|

Description

It draws a grid with an histogram per exposure in an [ExposomeSet](#).

Usage

```
plotHistogram(x, select, density = TRUE, show.trans = FALSE)
```

Arguments

| | |
|------------|-----------------------------------------------------------------------------------------------------------------------------------|
| x | ExposomeSet which exposome will be plotted. |
| select | Name fo the exposure to be plotted. If missing, all exposures will be used. |
| density | (default TRUE) If TRUE a density plot is drawn overlapping the histogram. |
| show.trans | (default FALSE) If set to TRUE it will draw a panel of auxiliar plots with the continuous exposure transformed with log and sqrt. |

Value

A ggplot object.

See Also

[plotFamily](#) to draw the profile of a family of exposures, [plotMissings](#) to plot the missing data from an [ExposomeSet](#)

Examples

```
data("exposome")
plotHistogram(expo[1:3, ], select = "ldde_lip")
```

| | |
|---------|-----------------------------------------------------------------------------------|
| plotLOD | <i>It draws a chart with the percentage of under-LOD values in an ExposomeSet</i> |
|---------|-----------------------------------------------------------------------------------|

Description

This function can be used to draw the amount of under-LOD values in the exposures of an [ExposomeSet](#).

Usage

```
plotLOD(object, lod.col = "LOD", x.max = 100, sort = TRUE)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeSet which exposome will be plotted. |
| lod.col | (default "LOD") Name of the column in fData containing the LOD thresholds. |
| x.max | (default 100) Fix the maximum value of the X-axis. |
| sort | (default TRUE) If TRUE the chart will be ordered from the features with less missing data to the ones with more under-LOD values. |

Value

A ggplot object.

See Also

[plotFamily](#) to draw the profile of a family of exposures, [plotHistogram](#) to draw the shape of an exposure, [tableMissings](#) to get a table with the missing data of an [ExposomeSet](#), [imputation](#) to impute missing data on the exposures of an [ExposomeSet](#)

Examples

```
data("exposome")
# The included has no missing data
plotLOD(expo)
```

| | |
|--------------|--------------------------------------------------------------------------|
| plotMissings | <i>It draws a chart with the percentage of missing in an ExposomeSet</i> |
|--------------|--------------------------------------------------------------------------|

Description

This function can be used to draw the number of missing data in exposures and in phenotypes of an [ExposomeSet](#).

Usage

```
plotMissings(object, set, x.max = 100, sort = TRUE)
```

Arguments

| | |
|--------|-------------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeSet which exposome will be plotted. |
| set | Can be set to "exposures" or to "phenotypes". |
| x.max | (default 100) Fix the maximum value of the X-axis. |
| sort | (default TRUE) If TRUE the chart will be ordered from the features with less missing data to the ones with more missing data. |

Value

A ggplot object.

See Also

[plotFamily](#) to draw the profile of a family of exposures, [plotHistogram](#) to draw the shape of an exposure, [tableMissings](#) to get a table with the missing data of an [ExposomeSet](#), [imputation](#) to impute missing data on the exposures of an [ExposomeSet](#)

Examples

```
data("exposome")
# The included has no missing data
plotMissings(expo, set = "exposures")
plotMissings(expo, set = "phenotypes")
```

plotPCA

Ploting PCA

Description

Method to draw a plot for PCA contained in an [ExposomePCA](#)

Usage

```
plotPCA(
  object,
  set,
  cmpX = 1,
  cmpY = 2,
  show.exposures = FALSE,
  show.samples = FALSE,
  phenotype
)
```

Arguments

| | |
|----------------|-----------------------------------------------------------------------------|
| object | An object of class ExposomePCA |
| set | Group ("all", "samples" or "exposures") that will be plotted. |
| cmpX | (default: 1) component to be placed at X axis |
| cmpY | (default: 2) component to be placed at Y axis |
| show.exposures | (default: FALSE) If set to true, labels indicating the exposures are shown. |
| show.samples | (default: FALSE) If set to true, labels indicating the samples are shown. |
| phenotype | If set is set to "samples" can be used to color samples by phenotype |

Value

An object of class `ggplot` or an object of class `gtable` if argument `set` was set to "all".

See Also

[pca](#) to compute PCA on an [ExposomeSet](#), [plotPCA](#) to plot the PCA, [ExposomePCA](#) as main class

Examples

```

data("exposome")
epca <- pca(expo[3:7, 1:100])
# A grid with exposures space, samples space and explained variance
plotPCA(epca, set = "all")
# Only exposures space
plotPCA(epca, set = "exposures") + ggplot2::theme(legend.position = "bottom")
# Only samples space
plotPCA(epca, set = "samples")
# Only samples space but coloured by phenotype
plotPCA(epca, set = "samples", phenotype = "sex") +
ggplot2::theme(legend.position = "bottom")

```

plotPHE

*Plot association score between phenotypes and PCA***Description**

Function used to plot the association between the phenotypes in an [ExposomePCA](#) and the values for each component of the PCA in the same [ExposomePCA](#)

Usage

```
plotPHE(object, phenotype, exp2fac = 5)
```

Arguments

| | |
|-----------|----------------------------------------------------------------------------------------------------------|
| object | An object of class ExposomePCA |
| phenotype | (optional) to select a set of phenotypes to be plotted. If not given all are used. |
| exp2fac | (default, 5) Threshold to consider a phenotype categorical (less or equal to) or continuous (more than). |

Value

An object of class ggplot.

See Also

[pca](#) to compute PCA on an [ExposomeSet](#), [plotEXP](#) to plot the correlation between exposures and PCA, [ExposomePCA](#) as main class

Examples

```

data("exposome")
epca <- pca(expo[3:7, 1:100])
plotPHE(epca)

```

| | |
|-------------|-------------------------------------------------------------------------|
| plotVolcano | <i>Function to draw a plot of the pvalues stored in an ExWAS object</i> |
|-------------|-------------------------------------------------------------------------|

Description

This function draws a sort of manhattan plots using the p-value of the association of the exposures with phenotypes of an ExWAS object.

Usage

```
plotVolcano(x, p.value = -log10(0.001), labels, show.effect = FALSE)
```

Arguments

| | |
|-------------|-------------------------------------------------------------------------------------------|
| x | An ExWAS object which effect will be plotted. |
| p.value | (default "-log10(0.001)") Threshold for P-Value. |
| labels | (optional) Character vector with the labels for each exposure. It must be labeled vector. |
| show.effect | (default FALSE) Applies an exponential transformation on the effects of the exposures. |

Value

An object of class ggplot.

See Also

[exwas](#) as a constructor for ExWAS objects, [extract](#) to obtain a table with the result of the ExWAS, [plotEffect](#) to see or compare effects of one or two models.

| | |
|--------------|----------------------------------------------|
| readExposome | <i>Creation of an ExposomeSet from files</i> |
|--------------|----------------------------------------------|

Description

Given the files that defines the exposome (measures of exposome, exposome description and individuals phenotype) it loads them and creates an object of type [ExposomeSet](#).

Usage

```
readExposome(
  exposures,
  description,
  phenotype,
  sep = ",",
  na.strings = c("NA", "-", "?", " ", "" ),
  exposures.samCol = "sample",
  description.expCol = "exposure",
  description.famCol = "family",
```

```

  phenotype.samCol = "sample",
  exposures.asFactor = 5,
  warnings = TRUE
)

```

Arguments

| | |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| exposures | String with the path to the file with the matrix of exposures. |
| description | String with the path to the file with the description of the exposures (relation between exposures and exposure-family). |
| phenotype | String with the path to the file with the phenotypes of interest. |
| sep | (default ", ") Separator used by <code>read.table</code> to load the files "exposures", "description" and "phenotype". |
| na.strings | (default c("NA", "-", "?", " ", "")) Character defining the NA values in exposure's files. |
| exposures.samCol | (default "sample") Index where the samples' name are found in file "exposures". It can be both charatcer or numeric. |
| description.expCol | (default "exposure") Index where the exposures' name are found in file "description". It can be both numeric or character. |
| description.famCol | (default "family") Index where the family's name (per exposures) if found in file "description". It can be both numeric or character. |
| phenotype.samCol | (default "sample") Index where the sample's name are found in file "phenotype". It can be both numeric or character. |
| exposures.asFactor | (default 5) The exposures with more than this number of unique items will be considered as "continuous" while the exposures with less or equal number of items will be considered as "factor". |
| warnings | (default TRUE) If TRUE shows useful information/warnings from the process of loading the exposome. |

Details

The rows of the exposure's file, that corresponds to samples' names, must be the same than the phenotype's file. In the same way, the columns in exposure's file must be the same found as rows in description file.

Value

An object of class `ExposomeSet`.

Note

`ExposomeSet`'s `fData` will contain some inner columns called `.std`, `.trn`, `.fct` and `.type` in order to trace the transformations an exposure suffers and to know, at any moment, if an exposure is categorical or continuous. The "description" file can contains a column called `type` with values "factor" and "numeric" to specify how an exposure needs to be understood. If given, this column will be renamed to `.type`. If not given, it will be created using `exposures.asFactor` value.

See Also

[ExposomeSet](#) for class description, [loadExposome](#) for constructor from loaded data.frames

Examples

```
## Locate the data-files
path <- file.path(path.package("rexposome"), "extdata")
description <- file.path(path, "description.csv")
phenotype <- file.path(path, "phenotypes.csv")
exposures <- file.path(path, "exposures.csv")

## Create ExposomeSet from files
exp <- readExposome(
  exposures = exposures,
  description = description,
  phenotype = phenotype,
  exposures.samCol = 1,
  description.expCol = 2,
  description.famCol = 1,
  phenotype.samCol = 1
)
```

rexposome

rexposome: Package for exposome exploration and outcome data analysis

Description

#' @section exposures loading and exposures management: rexposome offers two methods to bring exposome data to R and Bioconductor. [readExposome](#) allows to read three txt-like files (.csv, .tsv, ...) while [loadExposome](#) is used with matrix and data.frames. The class obtained is an [ExposomeSet](#), a class based in eSet for exposome data management.

exposures processing

The packages offers a wide set of functions to preprocess exposome data. Method [trans](#) allow to transform the exposures, method [normalityTest](#) allows to check for normality in exposome, [standardize](#) allows to standardize the exposures, among others. Finally, [impute](#) and [ilod](#) allow to use mice, Hmisc and imputeLCMD for exposure missing data and exposure under-lod data imputation.

exposures analyses

the two methods [exwas](#) and [mexwas](#) allows to test the association between exposures and health outcomes (phenotype data).

exposures plotting

The methods [plotFamily](#) allows to see how the exposures behaves within families. [plotCorrelation](#) helps to understand how exposures are related between themselves. [plotClassification](#) allows to visually detect clusters of samples that share the same pattern of levels of exposures.

```

standardize      #' Summary of an ExposomeSet. #' #' Given an ExposomeSet is
                  shows a summary for its exposures or #' its phenotypes. #' #' @name
                  Summary #' @rdname Summary-methods #' @aliases Summary #'
                  @param object codeExposomeSet with 'set' will be summarized. #'
                  @param set Set to be summarized ("exposures" or "phenotypes"). #'
                  @param select Subsetting of exposures of phenotypes. #' @return A
                  basic description of the exposures in the ExposomeSet #' @examples
                  #' data("exposome") #' Summary(expo, set = "exposures") #' @export
                  Summary setGeneric("Summary", function(object, set=c("exposures",
                  "phenotypes"), select) standardGeneric("Summary") ) Standardize of
                  an ExposomeSet.

```

Description

Given an [ExposomeSet](#) it standardizes the exposures by using mean/sd if method is "normal" or by using median/mad if method is "robust".

Usage

```
standardize(object, select, method = "normal", na.rm = TRUE, warnings = TRUE)
```

Arguments

| | |
|----------|-----------------------------------------------------------------------------------------------|
| object | ExposomeSet with exposures to be standardized. |
| select | Subsetting of exposures of phenotypes. |
| method | (default "normal") Character selecting the method to be applied ("normal" "iqr" or "robust"). |
| na.rm | (by default TRUE) Removes NA data to perform standardization. |
| warnings | (default TRUE) If set to FALSE warnings are not shown. |

Value

An [ExposomeSet](#) with the selected exposures standardized and keeping the others exposures as the original input object.

See Also

[highAndLow](#) to transform the continuous exposures to levelled factors, [trans](#) to transform the exposures

Examples

```

data("exposome")
exp.sn <- standardize(expo, method = "normal", select = "lbde100_lip")
exp.rs <- standardize(expo, method = "iqr", select = "lbde100_lip")
exp.rs <- standardize(expo, method = "robust", select = "lbde100_lip")

```

| | |
|----------|-------------------------------------------------------------------------------------|
| tableLOD | <i>It creates a vector with the amount of under-LOD exposures in an ExposomeSet</i> |
|----------|-------------------------------------------------------------------------------------|

Description

This function can be used to obtain a table with the under-LOD data in exposures of an [ExposomeSet](#).

Usage

```
tableLOD(object, output = "n", lod.col = "LOD", sort = TRUE)
```

Arguments

| | |
|---------|-----------------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeSet which exposome will be plotted. |
| output | (default "n") Can be "n" for number of values, and "p" for percentage. |
| lod.col | (default "LOD") Name of the column in fData containing the LOD thresholds. |
| sort | (default TRUE) If TRUE the chart will be ordered from the features with less missing data to the ones with more under-LOD values. |

Value

A numeric vector with number of under-LOD values per exposure. The vector is labeled with the exposure's names.

See Also

[plotFamily](#) to draw the profile of a family of exposures, [plotHistogram](#) to draw the shape of an exposure, [plotMissings](#) to draw a plot with the missing data an [ExposomeSet](#), [imputation](#) to impute missing data on the exposures of an [ExposomeSet](#)

Examples

```
data("exposome")
# The included has no missing data
tableLOD(expo, output = "n")
```

| | |
|---------------|-------------------------------------------------------------------------|
| tableMissings | <i>It creates a vector with the amount of missing in an ExposomeSet</i> |
|---------------|-------------------------------------------------------------------------|

Description

This function can be used to obtain a table with the missing data in exposures and in phenotypes of an [ExposomeSet](#).

Usage

```
tableMissings(object, set, output = "n", sort = TRUE)
```

Arguments

| | |
|--------|-------------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeSet which exposome will be plotted. |
| set | Can be set to "exposures" or to "phenotypes". |
| output | (default "n") Can be "n" for number of values, and "p" for percentage. |
| sort | (default TRUE) If TRUE the chart will be ordered from the features with less missing data to the ones with more missing data. |

Value

A numeric vector with number of missing values per exposure. The vector is labeled with the exposure's names.

See Also

[plotFamily](#) to draw the profile of a family of exposures, [plotHistogram](#) to draw the shape of an exposure, [plotMissings](#) to draw a plot with the missing data an [ExposomeSet](#), [imputation](#) to impute missing data on the exposures of an [ExposomeSet](#)

Examples

```
data("exposome")
# The included has no missing data
tableMissings(expo, set = "exposures")
tableMissings(expo, set = "phenotypes")
```

 tef

Function to get the Threshold for effective tests (TEF)

Description

Function to get the Threshold for effective tests (TEF)

Usage

```
tef(object)
```

Arguments

| | |
|--------|-----------------|
| object | An ExWAS object |
|--------|-----------------|

Value

A number indicating the effective threshold.

References

Evaluating the effective numbers of independent tests and significant p-value thresholds in commercial genotyping arrays and public imputation reference datasets. Miao-Xin Li, Juilian M. Y. Yeung, Stacey S. Cherny and Pak C. Sham. May 2012 Hum Genet.

See Also

[exwas](#) as a constructor for [ExWAS](#) objects

Examples

```
data(exposome)
w1 <- exwas(expo[1:5, ], asthma~1, family = "binomial")
w2 <- exwas(expo[1:5, ], asthma~sex+age, family = "binomial")
tef(w1)
tef(w2)
```

toES

Method to convert an imExposomeSet to an ExposomeSet

Description

This methods allows to select an imputed-set and use it to create an [ExposomeSet](#) from an [imExposomeSet](#).

Usage

```
toES(object, rid = 1)
```

Arguments

object An object of class [imExposomeSet](#)
rid (default 1) Number of the imputation to be extracted

Value

An object of class [ExposomeSet](#)-

Examples

```
data("ex_imp")
toES(ex_imp, rid = 1)
```

trans

Function to apply a transformation to the exposures of an Exposome-Set

Description

The exposures in an [ExposomeSet](#) can be transformed using this function. transform applies a function fun to the selected exposures.

Usage

```
trans(object, fun, select, by.exposure = FALSE, ...)
```


Arguments

| | |
|-------------|------------------------------------------------------------------------------------------------------------------------------|
| object | ExposomeSet which exposures will be transformed. |
| fun | Function to be applied on the exposures. |
| select | If not set, receive the name of all exposures. It can takes a character vector to select specific exposures. |
| by.exposure | (default FALSE) If TRUE applies fun to each exposure (given as a vector). If FALSE the matrix of exposures is fiven to fun. |
| ... | Argument given to fun. |

Value

A new ExposomeSet with selected exposures transformed with fun.

See Also

[highAndLow](#) to transform the continuous exposures to levelled factors, [standardize](#) to standardize by normal or robust methods the exposures

Examples

```
data("exposome")
exp.t <- trans(expo, fun = log, select = "ldde_lip")
```

volcano_plot

Function to draw a Volcano Plot

Description

Function that takes two numeric vectors (P-Value and fold change) and draws a volcano plot using [ggplot2](#)

Usage

```
volcano_plot(
  pval,
  fc,
  names,
  size = 2,
  tFC = 2,
  tPV = -log10(0.001),
  show.effect = FALSE
)
```

Arguments

| | |
|-------|---------------------------------------------------------|
| pval | numeric vector of P.Values |
| fc | numeric vector of fold change |
| names | character vector with the feature's names. |
| size | (default 2) Sice of the labels in case they are placed. |

tFC (default 2) fold change threshold. It can be set to NULL to do not filter.
tPV (default $-\log_{10}(0.001)$) P-Value threshold. It can be set to NULL to not filter.
show.effect (default FALSE) If set to TRUE, the X-axis will should $2^{\log_{2}FC}$ instead to teh default $\log_{2}FC$.

Value

A ggplot object

Examples

```
data(exposome)
w1 <- extract(exwas(expo[1:20, ], asthma~1, family = "binomial"))
volcano_plot(w1$pvalue, w1$effect, rownames(w1))
```

Index

- * **datasets**
 - ex_imp, [21](#)
 - expo, [6](#)
 - expo_c, [17](#)
 - me, [34](#)
- [(imExposomeSet), [25](#)
- [, imExposomeSet, ANY, ANY, ANY-method (imExposomeSet), [25](#)
- [[, imExposomeSet-method (imExposomeSet), [25](#)

- AnnotatedDataFrame, [9](#), [11](#), [15](#)
- AssayData, [9](#), [15](#)

- classification, [3](#), [4](#), [41](#)
- classification, ExposomeClust-method (ExposomeClust), [7](#)
- clustering, [3](#), [3](#), [5](#), [7](#), [8](#), [37](#), [39](#), [41](#)
- clustering, ExposomeSet-method (ExposomeSet), [11](#)
- cor, [5](#), [8](#)
- correlation, [5](#), [11](#), [37](#), [39](#), [42](#)
- correlation, ExposomeSet-method (ExposomeSet), [11](#)
- corrplot, [42](#)
- cramersV, [5](#)

- dim, ExposomeCorr-method (ExposomeCorr), [8](#)
- dim, ExposomeSet-method (ExposomeSet), [11](#)
- dim, imExposomeSet-method (imExposomeSet), [25](#)

- eSet, [9](#), [11](#), [15](#), [16](#)
- eSet-class, [8](#)
- ex_imp, [21](#)
- expo, [6](#)
- expo_c, [17](#)
- expos, [6](#)
- expos, ExposomeSet-method (ExposomeSet), [11](#)
- expos, imExposomeSet-method (imExposomeSet), [25](#)
- ExposomeClust, [3](#), [4](#), [7](#), [8](#), [41](#)
- ExposomeClust-class (ExposomeClust), [7](#)
- ExposomeCorr, [5](#), [8](#), [8](#), [42](#)
- ExposomeCorr-class (ExposomeCorr), [8](#)
- ExposomePCA, [9](#), [10](#), [37–40](#), [43](#), [48](#), [49](#)
- ExposomePCA-class (ExposomePCA), [9](#)
- ExposomeSet, [3](#), [5–9](#), [11](#), [11](#), [14](#), [16](#), [18](#), [20–22](#), [26–31](#), [33](#), [35](#), [36](#), [38](#), [40](#), [43](#), [45–56](#)
- ExposomeSet-class (ExposomeSet), [11](#)
- exposureNames, [16](#), [16](#), [22](#), [39](#)
- exposureNames, ExposomePCA-method (ExposomePCA), [9](#)
- exposureNames, ExposomeSet-method (ExposomeSet), [11](#)
- exposureNames, imExposomeSet-method (imExposomeSet), [25](#)
- extract, [17](#), [21](#), [30](#), [36](#), [43](#), [44](#), [50](#)
- extract (get_robust_sd), [22](#)
- extract, ExposomeCorr-method (ExposomeCorr), [8](#)
- extract, ExposomePCA-method (ExposomePCA), [9](#)
- extract, ExWAS-method (ExWAS), [18](#)
- extract, imExposomeSet-method (imExposomeSet), [25](#)
- extract, mExWAS-method (mExWAS), [35](#)
- ExWAS, [18](#), [18](#), [19](#), [36](#), [43](#), [44](#), [50](#), [56](#)
- exwas, [18](#), [19](#), [20](#), [36](#), [43](#), [44](#), [50](#), [52](#), [56](#)
- exwas, ExposomeSet-method (ExposomeSet), [11](#)
- exwas, imExposomeSet-method (imExposomeSet), [25](#)
- ExWAS-class (ExWAS), [18](#)

- FactoMineR, [38](#)
- familyNames, [16](#), [21](#), [39](#)
- familyNames, ExposomeSet-method (ExposomeSet), [11](#)
- familyNames, imExposomeSet-method (imExposomeSet), [25](#)
- featureNames, [16](#)

- get_robust_sd, [22](#)
- get_robust_sd, ExWAS-method (ExWAS), [18](#)

- ggplot2, [57](#)
- glm, [20](#)
- highAndLow, [23](#), [53](#), [57](#)
- highAndLow, ExposomeSet-method (ExposomeSet), [11](#)
- ilod, [24](#), [28](#), [52](#)
- ilod, ExposomeSet-method (ExposomeSet), [11](#)
- imExposomeSet, [25](#), [26](#), [33](#), [34](#), [56](#)
- imExposomeSet-class (imExposomeSet), [25](#)
- imExposomeSet-methods (imExposomeSet), [25](#)
- imputation, [28](#), [38](#), [47](#), [48](#), [54](#), [55](#)
- imputation, ExposomeSet-method (ExposomeSet), [11](#)
- impute, [52](#)
- impute.MinProb, [24](#)
- imputeLOD, [28](#)
- invExWAS, [29](#)
- invExWAS, ExposomeSet-method (ExposomeSet), [11](#)
- lm, [5](#)
- loadExposome, [16](#), [30](#), [52](#)
- loadExposome_plain, [32](#)
- loadImputed, [27](#), [33](#)
- me, [34](#)
- mExWAS, [19](#), [35](#), [36](#)
- mexwas, [19](#), [35](#), [36](#), [52](#)
- mexwas, ExposomeSet-method (ExposomeSet), [11](#)
- mExWAS-class (mExWAS), [35](#)
- names, ExWAS-method (ExWAS), [18](#)
- ndim, [37](#)
- ndim, ExposomePCA-method (ExposomePCA), [9](#)
- normalityTest, [37](#), [52](#)
- normalityTest, ExposomeSet-method (ExposomeSet), [11](#)
- pca, [5](#), [9](#), [38](#), [40](#), [42](#), [43](#), [48](#), [49](#)
- pca, ExposomeSet-method (ExposomeSet), [11](#)
- phenotypeNames, [16](#), [22](#), [39](#)
- phenotypeNames, ExposomePCA-method (ExposomePCA), [9](#)
- phenotypeNames, ExposomeSet-method (ExposomeSet), [11](#)
- phenotypeNames, imExposomeSet-method (imExposomeSet), [25](#)
- plot, ExposomeClust, ANY-method (ExposomeClust), [7](#)
- plot, ExposomeCorr, ANY-method (ExposomeCorr), [8](#)
- plot, ExposomePCA, ANY-method (ExposomePCA), [9](#)
- plot, ExposomeSet, ANY-method (ExposomeSet), [11](#)
- plot, ExWAS, ANY-method (ExWAS), [18](#)
- plot, imExposomeSet, ANY-method (imExposomeSet), [25](#)
- plot, mExWAS, ANY-method (mExWAS), [35](#)
- plot3PCA, [40](#)
- plot3PCA, ExposomePCA-method (ExposomePCA), [9](#)
- plotClassification, [3](#), [4](#), [41](#), [52](#)
- plotClassification, ExposomeClust-method (ExposomeClust), [7](#)
- plotCorrelation, [5](#), [41](#), [52](#)
- plotCorrelation, ExposomeCorr-method (ExposomeCorr), [8](#)
- plotEffect, [42](#), [50](#)
- plotEffect, ExWAS-method (ExWAS), [18](#)
- plotEXP, [43](#), [49](#)
- plotEXP, ExposomePCA-method (ExposomePCA), [9](#)
- plotExwas, [21](#), [30](#), [36](#), [44](#)
- plotExwas, ExWAS-method (ExWAS), [18](#)
- plotExwas, mExWAS-method (mExWAS), [35](#)
- plotFamily, [45](#), [46–48](#), [52](#), [54](#), [55](#)
- plotFamily, ExposomeSet-method (ExposomeSet), [11](#)
- plotFamily, imExposomeSet-method (imExposomeSet), [25](#)
- plotHistogram, [38](#), [45](#), [46](#), [47](#), [48](#), [54](#), [55](#)
- plotHistogram, ExposomeSet-method (ExposomeSet), [11](#)
- plotLOD, [46](#)
- plotLOD, ExposomeSet-method (ExposomeSet), [11](#)
- plotMissings, [25](#), [28](#), [38](#), [45](#), [46](#), [47](#), [54](#), [55](#)
- plotMissings, ExposomeSet-method (ExposomeSet), [11](#)
- plotPCA, [37](#), [39](#), [40](#), [48](#), [48](#)
- plotPCA, ExposomePCA-method (ExposomePCA), [9](#)
- plotPHE, [43](#), [49](#)
- plotPHE, ExposomePCA-method (ExposomePCA), [9](#)
- plotVolcano, [50](#)
- plotVolcano, ExWAS-method (ExWAS), [18](#)
- read.table, [32](#), [51](#)
- readExposome, [16](#), [31](#), [50](#), [52](#)
- rexpome, [52](#)

sampleNames, ExposomeClust-method
(ExposomeClust), 7

sampleNames, imExposomeSet-method
(imExposomeSet), 25

shapiro.test, 37, 38

standardize, 23, 52, 53, 57

standardize, ExposomeSet-method
(ExposomeSet), 11

Summary, ExposomeSet-method
(ExposomeSet), 11

tableLOD, 54

tableLOD, ExposomeSet-method
(ExposomeSet), 11

tableMissings, 25, 28, 47, 48, 54

tableMissings, ExposomeSet-method
(ExposomeSet), 11

tef, 55

tef, ExWAS-method (ExWAS), 18

toES, 56

toES, imExposomeSet-method
(imExposomeSet), 25

trans, 23, 52, 53, 56

trans, ExposomeSet-method (ExposomeSet),
11

volcano_plot, 57