Package 'phenomis'

December 19, 2024

Type Package

Title Postprocessing and univariate analysis of omics data

Version 1.9.0

Date 2024-09-02

Description The 'phenomis' package provides methods to perform post-processing (i.e. quality control and normalization) as well as univariate statistical analysis of single and multi-omics data sets. These methods include quality control metrics, signal drift and batch effect correction, intensity transformation, univariate hypothesis testing, but also clustering (as well as annotation of metabolomics data). The data are handled in the standard Bioconductor formats (i.e. SummarizedExperiment and MultiAssayExperiment for single and multi-omics datasets, respectively; the alternative ExpressionSet and MultiDataSet formats are also supported for convenience). As a result, all methods can be readily chained as workflows. The pipeline can be further enriched by multivariate analysis and feature selection, by using the 'ropls' and 'biosigner' packages, which support the same formats. Data can be conveniently imported from and exported to text files. Although the methods were initially targeted to metabolomics data, most of the methods can be applied to other types of omics data (e.g., transcriptomics, proteomics).

biocViews BatchEffect, Clustering, Coverage, KEGG, MassSpectrometry, Metabolomics, Normalization, Proteomics, QualityControl, Sequencing, StatisticalMethod, Transcriptomics

Depends SummarizedExperiment

Imports Biobase, biodb, biodbChebi, data.table, futile.logger, ggplot2, ggrepel, graphics, grDevices, grid, htmlwidgets, igraph, limma, methods, MultiAssayExperiment, MultiDataSet, PMCMRplus, plotly, ranger, RColorBrewer, ropls, stats, tibble, tidyr, utils, VennDiagram

Suggests BiocGenerics, BiocStyle, biosigner, CLL, knitr, omicade4, rmarkdown, testthat

VignetteBuilder knitr

License CeCILL

Encoding UTF-8

LazyLoad yes

URL https://doi.org/10.1038/s41597-021-01095-3

RoxygenNote 7.2.3

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

git_branch devel

git_last_commit b15f3f7

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-12-18

Author Etienne A. Thevenot [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1019-4577>),
Natacha Lenuzza [ctb],
Marie Tremblay-Franco [ctb],
Alyssa Imbert [ctb],
Pierrick Roger [ctb],
Eric Venot [ctb],
Sylvain Dechaumet [ctb]

Maintainer Etienne A. Thevenot <etienne.thevenot@cea.fr>

Contents

phenomis-package *Postprocessing and univariate analysis of omics data*

Description

The 'phenomis' package provides methods to perform post-processing (i.e. quality control and normalization) as well as univariate statistical analysis of single and multi-omics data sets. These methods include quality control metrics, signal drift and batch effect correction, intensity transformation, univariate hypothesis testing, but also clustering (as well as annotation of metabolomics data). The data are handled in the standard Bioconductor formats (i.e. SummarizedExperiment and MultiAssayExperiment for single and multi-omics datasets, respectively; the alternative ExpressionSet and MultiDataSet formats are also supported for convenience). As a result, all methods can be readily chained as workflows. The pipeline can be further enriched by multivariate analysis and feature selection, by using the 'ropls' and biosigner' packages, which support the same formats. Data can be conveniently imported from and exported to text files. Although the methods were initially targeted to metabolomics data, most of the methods can be applied to other types of omics data (e.g., transcriptomics, proteomics).

Author(s)

E. A. Thévenot (CEA)

Maintainer: Etienne Thevenot <etienne.thevenot@cea.fr>

Examples

```
# See the package vignette
```

annotating *MS annotation*

Description

Annotation with chemical and biological databases by using the 'biodb' package suite. The present implementation currently enables to query the ChEBI database or a local database.

The parameters and their default values are printed for the selected database

Usage

```
annotating(
  x,
  database.c = c("chebi", "local.ms")[1],
 param.ls = list(query.type = c("mz", "chebi.id")[1], query.col = "mz", ms.mode = "pos",
    mz.tol = 10, mz.tol.unit = "ppm", fields = c("chebi.id", "name", "formula",
  "molecular.mass", "monoisotopic.mass"), fieldsLimit = 1, max.results = 3, local.ms.db
```

```
    = data.frame(), prefix = paste0(database.c, "."), sep = "|"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
annotating(
  x,
  database.c = c("chebi", "local.ms")[1],
 param.ls = list(query.type = c("mz", "chebi.id")[1], query.col = "mz", ms.mode = "pos",
    mz.tol = 10, mz.tol.unit = "ppm", fields = c("chebi.id", "name", "formula",
   "molecular.mass", "monoisotopic.mass"), fieldsLimit = 1, max.results = 3, local.ms.db
    = data.frame(), prefix = paste0(database.c, "."), sep = "|"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
annotating(
  x,
  database.c = c("chebi", "local.ms")[1],
 param.ls = list(query.type = c("mz", "chebi.id")[1], query.col = "mz", ms.mode = "pos",
    mz.tol = 10, mz.tol.unit = "ppm", fields = c("chebi.id", "name", "formula",
   "molecular.mass", "monoisotopic.mass"), fieldsLimit = 1, max.results = 3, local.ms.db
    = data.frame(), prefix = paste0(database.c, "."), sep = "|"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
annotating(
  x,
  database.c = c("chebi", "local.ms")[1],
 param.ls = list(query.type = c("mz", "chebi.id")[1], query.col = "mz", ms.mode = "pos",
    mz.tol = 10, mz.tol.unit = "ppm", fields = c("chebi.id", "name", "formula",
   "molecular.mass", "monoisotopic.mass"), fieldsLimit = 1, max.results = 3, local.ms.db
    = data.frame(), prefix = paste0(database.c, "."), sep = "|"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
annotating(
  x,
  database.c = c("chebi", "local.ms")[1],
 param.ls = list(query.type = c("mz", "chebi.id")[1], query.col = "mz", ms.mode = "pos",
    mz.tol = 10, mz.tol.unit = "ppm", fields = c("chebi.id", "name", "formula",
   "molecular.mass", "monoisotopic.mass"), fieldsLimit = 1, max.results = 3, local.ms.db
    = data.frame(), prefix = paste0(database.c, "."), sep = "|"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

```
annotating_parameters(database.c = c("chebi", "local.ms")[1])
```

Arguments

x	An S4 object of class SummarizedExperiment or MultiAssayExperiment (ExpressionSet and MultiDataSet are still supported)
database.c	character(1): database to be used for annotation; either the ChEBI distant database ('chebi'), or a local database ('local.ms')
param.ls	list: parameters for database query; the database can be queried by either the mass to charge ratio (mz) or the chebi ID; other query parameters include the ionization mode (ms.mode), the mz tolerance (mz.tol; e.g. 5 ppm for Orbitrap Mass Spectrometers), the fields to retrieve (fields), the maximum number of items to retrieve when a field contains more than one value (fieldsLimit), the maximum number of results to provide for each query (max.results), prefix of the new columns providing the queried information in the feature metadata (prefix), separator in case of multiple retrieved values (sep), local data base to be queried (local.ms.db); additional information is provided by the vignettes from the biodb and biodbChebi packages on Bioconductor
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

SummarizedExperiment or MultiAssayExperiment (or ExpressionSet and MultiDataSet) including the appended rowData data frame(s)

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
# see the (default) parameters (e.g. for ChEBI query)
annotating_parameters("chebi")
# mz annotation with ChEBI

sacurine.se <- annotating(sacurine.se, database.c = "chebi",
param.ls = list(query.type = "mz", query.col = "mass_to_charge",
ms.mode = "neg", prefix = "chebiMZ."))

# mz annotation with local database
msdbDF <- read.table(system.file("extdata/local_ms_db.tsv",
package = "phenomis"),
header = TRUE, sep = "\t", stringsAsFactors = FALSE)
sacurine.se <- annotating(sacurine.se, database.c = "local.ms",
param.ls = list(query.type = "mz", query.col = "mass_to_charge",
ms.mode = "neg",
mz.tol = 5, mz.tol.unit = "ppm", local.ms.db = msdbDF, prefix = "localMS."))
rowData(sacurine.se)[!is.na(rowData(sacurine.se)[, "localMS.accession"]), ]
# annotation from ChEBI identifiers

sacurine.se <- annotating(sacurine.se, database.c = "chebi",
```

```
param.ls = list(query.type = "chebi.id", query.col = "database_identifier",
prefix = "chebiID."))
head(rowData(sacurine.se))

annotating_parameters()
annotating_parameters("chebi")
```

---

clustering *clustering*

Description

Hierarchical clustering of both samples and variables

Usage

```
clustering(
  x,
 dissym.c = c("euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski",
    "1-cor", "1-abs(cor)")[7],
  correl.c = c("pearson", "kendall", "spearman")[1],
 agglo.c = c("ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median",
    "centroid")[2],
  clusters.vi = c(2, 2),
  cex.vn = c(1, 1),
  palette.c = c("blueOrangeRed", "redBlackGreen")[1],
  scale_plot.l = TRUE,
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
clustering(
  x,
 dissym.c = c("euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski",
    "1-cor", "1-abs(cor)")[7],
  correl.c = c("pearson", "kendall", "spearman")[1],
 agglo.c = c("ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median",
    "centroid")[2],
  clusters.vi = c(2, 2),
  cex.vn = c(1, 1),
  palette.c = c("blueOrangeRed", "redBlackGreen")[1],
  scale_plot.l = TRUE,
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
```

```
)

## S4 method for signature 'SummarizedExperiment'
clustering(
  x,
 dissym.c = c("euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski",
    "1-cor", "1-abs(cor)")[7],
  correl.c = c("pearson", "kendall", "spearman")[1],
 agglo.c = c("ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median",
    "centroid")[2],
  clusters.vi = c(2, 2),
  cex.vn = c(1, 1),
  palette.c = c("blueOrangeRed", "redBlackGreen")[1],
  scale_plot.l = TRUE,
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
clustering(
  x,
 dissym.c = c("euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski",
    "1-cor", "1-abs(cor)")[7],
  correl.c = c("pearson", "kendall", "spearman")[1],
 agglo.c = c("ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median",
    "centroid")[2],
  clusters.vi = c(2, 2),
  cex.vn = c(1, 1),
  palette.c = c("blueOrangeRed", "redBlackGreen")[1],
  scale_plot.l = TRUE,
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
clustering(
  x,
 dissym.c = c("euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski",
    "1-cor", "1-abs(cor)")[7],
  correl.c = c("pearson", "kendall", "spearman")[1],
 agglo.c = c("ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median",
    "centroid")[2],
  clusters.vi = c(2, 2),
  cex.vn = c(1, 1),
  palette.c = c("blueOrangeRed", "redBlackGreen")[1],
  scale_plot.l = TRUE,
```

```
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

x	An S4 object of class SummarizedExperiment or MultiAssayExperiment (ExpressionSet and MultiDataSet are still supported)
dissym.c	character(1): dissymilarity to be used in the hierarchical clustering (as provided by the hclust package)
correl.c	character(1): correlation coefficient (in case '1-cor' or '1-abs(cor)' are selected as dissymilarity)
agglo.c	character(1): agglomeration method
clusters.vi	integer(2): number of sample and variable clusters, respectively; the default values (2) are only provided as starting guess (e.g. in case of two groups of samples)
cex.vn	numeric(2) [Plot parameter]; size of the sample and variable labels
palette.c	character(1) [Plot parameter]: color palette
scale_plot.l	logical(1) [Plot parameter]: scaling (mean-centering and unit variance scaling) to enhance contrast (for plotting only)
title.c	character(1) [Plot parameter]: Graphic the subtitle
figure.c	character(1): File name with '.pdf' extension for the figure; if 'interactive' (default), figures will be displayed interactively; if 'none', no figure will be generated
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

SummarizedExperiment or MultiAssayExperiment (or ExpressionSet and MultiDataSet) including columns indicating the clusters in rowData and colData if clusters.vi' has been specified

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine.se <- correcting(sacurine.se)
sacurine.se <- sacurine.se[, colData(sacurine.se)[, "sampleType"] != "pool"]
sacurine.se <- transforming(sacurine.se)
sacurine.se <- sacurine.se[, colnames(sacurine.se) != "HU_neg_096_b2"]
sacurine.se <- clustering(sacurine.se)
utils::head(rowData(sacurine.se))

# MultiAssayExperiment

prometis.mae <- reading(system.file("extdata/prometis", package="phenomis"))
prometis.mae <- clustering(prometis.mae)
```

correcting *correcting*

Description

Signal drift and batch effect correction. The normalization strategy relies on the measurements of a pooled (or QC) sample injected periodically: for each variable, a regression model is fitted to the values of the pool and subsequently used to adjust the intensities of the samples of interest (van der Kloet et al, 2009; Dunn et al, 2011). In case the number of pool observations is below 5, the linear method is used (for all variables) and a warning is generated. In case no pool is available, the samples themselves can be used to computed the regression model (Thevenot et al., 2015). The sample metadata of each datasets (e.g. colData Data Frames) must contain 3 columns: 1) 'sampleType' (character): only the 'sample' or 'pool' values can be used to indicate the reference samples for the correction, 2) 'injectionOrder' (integer): order of injection in the instrument, and 3) 'batch' (character): batch name(s).

Usage

```
correcting(
  x,
  method.vc = c("loess", "serrf")[1],
  reference.vc = c("pool", "sample")[1],
  loess_span.vn = 1,
  serrf_corvar.vi = 10,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
correcting(
  x,
  method.vc = c("loess", "serrf")[1],
  reference.vc = c("pool", "sample")[1],
  loess_span.vn = 1,
  serrf_corvar.vi = 10,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
correcting(
  x,
  method.vc = c("loess", "serrf")[1],
```

```
  reference.vc = c("pool", "sample")[1],
  loess_span.vn = 1,
  serrf_corvar.vi = 10,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
correcting(
  x,
  method.vc = c("loess", "serrf")[1],
  reference.vc = c("pool", "sample")[1],
  loess_span.vn = 1,
  serrf_corvar.vi = 10,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
correcting(
  x,
  method.vc = c("loess", "serrf")[1],
  reference.vc = c("pool", "sample")[1],
  loess_span.vn = 1,
  serrf_corvar.vi = 10,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

x	An S4 object of class SummarizedExperiment or MultiAssayExperiment (ExpressionSet and MultiDataSet are still supported)
method.vc	character of length 1 or the total number of datasets: method(s) to be used for each dataset (either 'serrf' or 'loess'); for the 'serrf' approach, the seed is internally set to 123 for reproducibility; in case the parameter is of length 1 and x contains multiple datasets, the same method will be used for all datasets
reference.vc	character of length 1 or the total number of datasets: sample type to be used as reference for the correction (as indicated in the 'sampleType' column from the colData(x); e.g. 'pool' [default]); should be set to 'pool' for the 'serrf' method; in case the parameter is of length 1 and x contains multiple datasets, the same reference sample type will be used for all datasets

loess_span.vn — character of length 1 or the total number of datasets: smoothing parameter for the loess regression; between 0 and 1; (default set to 1); in case the parameter is of length 1 and x contains multiple datasets, the same span value will be used for all datasets

serrf_corvar.vi — character of length 1 or the total number of datasets: number of correlated features for the random forest regression; (default set to 10); in case the parameter is of length 1 and x contains multiple datasets, the same value will be used for all datasets

sample_intensity.c — character(1): metric to be used when displaying the sample intensities

title.c — character(1): Graphic title: if NA [default] the 'title' slot from the experimentData will be used (metadata)

figure.c — character(1): File name with '.pdf' extension for the figure; if 'interactive' (default), figures will be displayed interactively; if 'none', no figure will be generated

report.c — character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

`SummarizedExperiment` or `MultiAssayExperiment` (or `ExpressionSet` and `MultiDataSet`) including the corrected intensities in the assay matrix (matrices)

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine.se <- correcting(sacurine.se)

# MultiDataSet (to be done)
```

---

filtering — *Filtering of the features (and/or samples) with a high proportion of NAs or a low variance*

Description

Filtering of the features (and/or samples) with a high proportion of NAs or a low variance

Usage

```
filtering(
  x,
  class.c = "",
  max_na_prop.n = 0.2,
  min_variance.n = .Machine$double.eps,
```

```
  dims.vc = c("features", "samples"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
filtering(
  x,
  class.c = "",
  max_na_prop.n = 0.2,
  min_variance.n = .Machine$double.eps,
  dims.vc = c("features", "samples"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
filtering(
  x,
  class.c = "",
  max_na_prop.n = 0.2,
  min_variance.n = .Machine$double.eps,
  dims.vc = c("features", "samples"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
filtering(
  x,
  class.c = "",
  max_na_prop.n = 0.2,
  min_variance.n = .Machine$double.eps,
  dims.vc = c("features", "samples"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
filtering(
  x,
  class.c = "",
  max_na_prop.n = 0.2,
  min_variance.n = .Machine$double.eps,
  dims.vc = c("features", "samples"),
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

x	An S4 object of class `SummarizedExperiment` or `MultiAssayExperiment` (`ExpressionSet` and `MultiDataSet` are still supported)

class.c	character(1): name of the column of the sample metadata giving the classification groups: the filtering will be applied on each class (default: "" meaning that there are no specific classes to consider)
max_na_prop.n	numeric(1): maximum proportion of NAs for a feature (or sample) to be kept (e.g. the default 20 values); in case 'class.c' is provided, the maximum proportion of NAs for a feature must be achieved in at least one sample class)
min_variance.n	numeric(1): minimum variance for a feature (or sample) to be kept (e.g. the default 0 value to discard constant features (or samples); in case 'class.c' is provided, the minimum variance for a feature must be achieved in all sample classes
dims.vc	Vector of one or two characters: dimension(s) to which the filtering should be applied; either 'features', 'samples', c('features', 'samples'), or c('samples', 'features'); in the two latter cases, the dimensions indicated in the dims.vc are filtered sequentially
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

SummarizedExperiment or MultiAssayExperiment (or ExpressionSet and MultiDataSet) including the filtered data and metadata

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
assay.mn <- assay(sacurine.se)
ropls::view(assay.mn)
filtering(sacurine.se)
assay.mn[assay.mn < 1e5] <- NA
ropls::view(assay.mn)
assay(sacurine.se) <- assay.mn
filtering(sacurine.se)
filtering(sacurine.se, class.c = "gender")
filtering(sacurine.se, class.c = "sampleType")

# MultiAssayExperiment

prometis.mae <- reading(system.file("extdata/prometis", package="phenomis"))
filtering(prometis.mae)
for (set.c in names(prometis.mae)) {
set.se <- prometis.mae[[set.c]]
assay.mn <- assay(set.se)
assay.mn[assay.mn < quantile(c(assay.mn), 0.2)] <- NA
assay(set.se) <- assay.mn
prometis.mae[[set.c]] <- set.se
}
filtering(prometis.mae)

# MultiDataSet
```

```
prometis.mset <- reading(system.file("extdata/prometis", package="phenomis"),
                         output.c = "set")
filtering(prometis.mset)
for (set.c in names(prometis.mset)) {
eset <- prometis.mset[[set.c]]
exprs.mn <- Biobase::exprs(eset)
exprs.mn[exprs.mn < quantile(c(exprs.mn), 0.2)] <- NA
Biobase::exprs(eset) <- exprs.mn
prometis.mset <- MultiDataSet::add_eset(prometis.mset, eset,
                                        dataset.type = set.c,
                                        GRanges = NA, overwrite = TRUE,
                                        warnings = FALSE)
}
filtering(prometis.mset)
```

---

gg_barplot *Barplot with ggplot2*

Description

Barplot with ggplot2

Usage

```
gg_barplot(
  data.mn,
  log10.l = FALSE,
  ylim.vn = c(NA, NA),
  title.c = "",
  xlab.c = "",
  ylab.c = "",
  row_levels.vc = NA,
  col_levels.vc = NA,
  palette.vc = "Set1",
 theme.c = c("default", "bw", "classic", "dark", "gray", "linedraw", "light", "minimal",
    "void")[3],
  flip.l = FALSE,
  legend_position.c = c("none", "bottom", "left", "top", "right")[2],
  cex_axis.i = 18,
  cex_bar.i = 10,
  cex_title.i = 28,
  bar_just.n = 0.9,
  figure.c = c("interactive", "my_barplot.pdf", "none")[1]
)
```

Arguments

data.mn	Matrix of numerics: values to be barplotted
log10.l	logical(1): should the intensities be log10 transformed?
ylim.vn	numeric(2): minimum and maximum values for the bars
title.c	Character: plot title
xlab.c	Character: x label
ylab.c	Character: y label
row_levels.vc	Vector of characters: levels of rownames (default: NA: alphabetical order will be used)
col_levels.vc	Vector of characters: levels of colnames (default: NA: alphabetical order will be used)
palette.vc	Character: either the name of an RColorBrewer palette (default: 'Set1'; 'Paired' can be useful for parallel plotting) or a vector manually defining the colors
theme.c	character(1): name of the ggplot theme
flip.l	logical(1): should the barplot be flipped (default: FALSE)
legend_position.c	character(1): position of the legend: either "none", "bottom" (default), "left", "top", "right"
cex_axis.i	Integer: size of axis text (default: 18)
cex_bar.i	Integer: size of bar value text (default: 10)
cex_title.i	Integer: size of title text (default: 28)
bar_just.n	Numeric: adjustment of bar value text (default : 0.9)
figure.c	Character: either 'interactive' for interactive display, 'my_barplot.pdf' for figure saving (only the extension matters), or 'none' to prevent plotting

Value

invisible ggplot2 object

Examples

```
prometis.mae <- reading(system.file("extdata/prometis", package = "phenomis"))
dims.mn <- vapply(names(prometis.mae),
                  function(set.c) { dim(prometis.mae[[set.c]])},
                  FUN.VALUE = integer(2))
dims.mn <- t(dims.mn)
colnames(dims.mn) <- c("features", "samples")
gg_barplot(dims.mn, title.c = "ProMetIS data",
           row_levels = c("proteo", "metabo"),
           col_levels = c("samples", "features"),
           ylim.vn = c(NA, 110),
           bar_just = -0.25,
           cex_bar.i = 6,
           cex_title.i = 15)
```

gg_boxplot *Boxplot with ggplot2*

Description

Boxplot with ggplot2

Usage

```
gg_boxplot(
  data.tb,
  x.c = "",
  y.c = "",
  color.c = "",
  title.c = NA,
  xlab.c = NA,
  ylab.c = "",
  label.vc = "",
  palette.vc = "Set1",
 theme.c = c("default", "bw", "classic", "dark", "gray", "linedraw", "light", "minimal",
    "void")[3],
  size.ls = list(dot.n = 0.7, lab.i = 20, tick.i = 20, title.i = 20),
  figure.c = c("interactive", "my_boxplot.pdf")[1]
)
```

Arguments

data.tb	Data frame (or tibble) containing the information
x.c	Character: name of the column with qualitative levels
y.c	Character: name of the column with quantitative values
color.c	Character: optional name of the column for color information
title.c	Character: plot title
xlab.c	Character: x label
ylab.c	Character: y label
label.vc	Character (vector): either the name of a character column from the data or a character vector of the same length as the rown number of the data, containing the feature labeling for outlier display
palette.vc	Character: either the name of an RColorBrewer palette (default: 'Set1'; 'Paired' can be useful for parallel plotting) or a vector manually defining the colors
theme.c	character(1): name of the ggplot theme
size.ls	List of sizes for dots (default is 0.7), labels (default is 16), ticks (14) and title (20)
figure.c	Character: either 'interactive' for interactive display or 'my_barplot.pdf' for figure saving (only the extension matters)

Value

character vector of outlier labels (same dimension as the number of rows from data.tb)

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine_pda.df <- as.data.frame(colData(sacurine.se))
sacurine_pda.df <- sacurine_pda.df[!grepl("QC", rownames(sacurine_pda.df)), ]
gg_boxplot(sacurine_pda.df, y.c = "age")
gg_boxplot(sacurine_pda.df, x.c = "gender", y.c = "bmi", color.c = "gender")
gg_boxplot(sacurine_pda.df, x.c = "gender", y.c = "bmi", color.c = "gender",
label.vc = rownames(sacurine_pda.df))
```

---

gg_pie *Pie with ggplot2*

Description

Pie with ggplot2

Usage

```
gg_pie(
  data.tb,
  y.c = "",
  color.c = "",
  title.c = "",
  palette.vc = "Set1",
  label.c = c("none", "value", "percent")[1],
 geom_text.ls = list(lab.i = 7, legend_title.i = 16, legend_text.i = 14, title.i = 16),
  figure.c = c("interactive", "my_pie.pdf", "none")[1]
)
```

Arguments

data.tb	Tibble (or data frame) containing the information
y.c	Character: name of the column with the factor to be displayed; alternatively, name of the column with the counts (in this case set the name of the column with the names of the factor levels with the 'color.c' argument)
color.c	Character: optional name of the column with the names of the factor levels
title.c	Character: plot title
palette.vc	Character: either the name of an RColorBrewer palette (default: 'Set1'; 'Paired' can be useful for parallel plotting) or a vector manually defining the colors
label.c	Character: (relative) counts to be displayed on the pie; either 'none' (default), 'value' or 'percent'

geom_text.ls	List of sizes for lab.i (default 7), legend_title.i (16), legend_text.i (14), and title.i (16)
figure.c	Character: either 'interactive' for interactive display, 'my_pie.pdf' for figure saving (only the extension matters), or 'none' to prevent plotting

Value

invisible ggplot2 object

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine_pda.df <- colData(sacurine.se)
sacurine_pda.df <- sacurine_pda.df[!grepl("QC", rownames(sacurine_pda.df)), ]
gg_pie(sacurine_pda.df, y.c = "gender", label.c = "value")
```

---

gg_volcanoplot *Volcano plot with ggplot2*

Description

Volcano plot with ggplot2

Usage

```
gg_volcanoplot(
  fold_change.vn,
  adjusted_pvalue.vn,
  adjust_method.c = "",
  adjust_thresh.n = 0.05,
  label.vc = "",
  title.c = "",
  xlab.c = "Fold Change",
  signif_palette.vc = c(yes = RColorBrewer::brewer.pal(9, "Greens")[8], no =
    RColorBrewer::brewer.pal(9, "Greys")[7]),
  signif_shape.vi = c(yes = 16, no = 1),
  class_name.vc = "",
  class_color.vc = "",
  size.ls = list(class.i = 5, lab.i = 16, point.i = 3, tick.i = 14, title.i = 20),
  figure.c = c("interactive", "interactive_plotly", "my_volcanoplot.pdf",
    "my_volcanoplot.html")[2]
)
```

Arguments

fold_change.vn	Numeric vector: fold changes
adjusted_pvalue.vn	Numeric vector: (adjusted) p-values
adjust_method.c	Character: method for multiple testing correction
adjust_thresh.n	Numeric: significance threshold
label.vc	Character (vector): either the name of a character column from the data or a character vector of the same length as the row number of the data, containing the feature labeling
title.c	Character: plot title
xlab.c	Character: x label (default: "Fold Change")
signif_palette.vc	Character vector: color palette (default 'green4' for significant features and 'gray' otherwise
signif_shape.vi	Integer vector: shapes for significant (respectively, non significant) features; default is 16 (respectively, 1)
class_name.vc	Character vector: names of the two compared class labels
class_color.vc	Character vector: colors of the two compared class labels
size.ls	List of sizes for classes (default: 5), xy labels (default: 16), points (default: 3), ticks (default: 14) and title (default: 20)
figure.c	Character: either 'interactive' (respectively, 'interactive_plotly') for interactive display with ggplot2 (respectively, with plotly::ggplotly [default]), or 'my_volcanoplot.pdf' (respectively 'my_volcanoplot.html') for figure saving (only the extension matters) with ggplot2 (respectively, with plotly::ggplotly)

Value

invisible ggplot2 object

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine.se <- correcting(sacurine.se, figure.c = "none")
sacurine.se <- sacurine.se[, colData(sacurine.se)[, "sampleType"] != "pool"]
sacurine.se <- transforming(sacurine.se)
sacurine.se <- hypotesting(sacurine.se, test.c = "wilcoxon",
                                    factor_names.vc = "gender",
                                    figure.c = "none", report.c = "none")
fold.vn <- rowData(sacurine.se)[, "wilcoxon_gender_Female.Male_diff"]
fdr.vn <- rowData(sacurine.se)[, "wilcoxon_gender_Female.Male_BH"]
feat.vc <- rownames(sacurine.se)
gg_volcanoplot(fold.vn,
               fdr.vn,
```

```
                label.vc = make.names(feat.vc),
                adjust_method.c = "BH")
feat_signif.vc <-  vapply(seq_along(feat.vc),
                          function(feat.i)
                           ifelse(fdr.vn[feat.i] <= 0.05, feat.vc[feat.i], ""),
                           FUN.VALUE = character(1))
gg_volcanoplot(fold.vn,
                fdr.vn,
                label.vc = make.names(feat_signif.vc),
                adjust_method.c = "BH",
                figure.c = "interactive")
```

---

hypotesting *Univariate hypothesis testing*

Description

The hypotesting method is a wrapper of the main R functions for hypothesis testing and corrections for multiple testing. The list of available tests includes two sample tests (t-test and Wilcoxon rank test, but also the limma test), analysis of variance (for one and two factors) and Kruskal-Wallis rank test, and correlation tests (by using either the pearson or the spearman correlation).

Usage

```
hypotesting(
  x,
 test.c = c("ttest", "limma", "wilcoxon", "anova", "kruskal", "pearson", "spearman",
    "limma2ways", "limma2waysInter", "anova2ways", "anova2waysInter")[2],
  factor_names.vc,
  factor_levels.ls = list(factor1.vc = "default", factor2.vc = "default"),
 adjust.c = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")[5],
  adjust_thresh.n = 0.05,
  signif_maxprint.i = NA,
  title.c = NA,
  display_signif.l = FALSE,
  prefix.c = "",
  figure.c = c("none", "interactive", "interactive_plotly", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
hypotesting(
  x,
 test.c = c("ttest", "limma", "wilcoxon", "anova", "kruskal", "pearson", "spearman",
    "limma2ways", "limma2waysInter", "anova2ways", "anova2waysInter")[2],
  factor_names.vc,
  factor_levels.ls = list(factor1.vc = "default", factor2.vc = "default"),
 adjust.c = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")[5],
```

```
  adjust_thresh.n = 0.05,
  signif_maxprint.i = NA,
  title.c = NA,
  display_signif.l = FALSE,
  prefix.c = "",
  figure.c = c("none", "interactive", "interactive_plotly", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
hypotesting(
  x,
 test.c = c("ttest", "limma", "wilcoxon", "anova", "kruskal", "pearson", "spearman",
    "limma2ways", "limma2waysInter", "anova2ways", "anova2waysInter")[2],
  factor_names.vc,
  factor_levels.ls = list(factor1.vc = "default", factor2.vc = "default"),
 adjust.c = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")[5],
  adjust_thresh.n = 0.05,
  signif_maxprint.i = NA,
  title.c = NA,
  display_signif.l = FALSE,
  prefix.c = "",
  figure.c = c("none", "interactive", "interactive_plotly", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
hypotesting(
  x,
 test.c = c("ttest", "limma", "wilcoxon", "anova", "kruskal", "pearson", "spearman",
    "limma2ways", "limma2waysInter", "anova2ways", "anova2waysInter")[2],
  factor_names.vc,
  factor_levels.ls = list(factor1.vc = "default", factor2.vc = "default"),
 adjust.c = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")[5],
  adjust_thresh.n = 0.05,
  signif_maxprint.i = NA,
  title.c = NA,
  display_signif.l = FALSE,
  prefix.c = "",
  figure.c = c("none", "interactive", "interactive_plotly", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
hypotesting(
  x,
 test.c = c("ttest", "limma", "wilcoxon", "anova", "kruskal", "pearson", "spearman",
    "limma2ways", "limma2waysInter", "anova2ways", "anova2waysInter")[2],
```

```
  factor_names.vc,
  factor_levels.ls = list(factor1.vc = "default", factor2.vc = "default"),
 adjust.c = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")[5],
  adjust_thresh.n = 0.05,
  signif_maxprint.i = NA,
  title.c = NA,
  display_signif.l = FALSE,
  prefix.c = "",
  figure.c = c("none", "interactive", "interactive_plotly", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

x	An S4 object of class SummarizedExperiment or MultiAssayExperiment (ExpressionSet and MultiDataSet are still supported)
test.c	character(1): One of the 9 available hypothesis tests can be selected (either 'ttest', 'limma', 'wilcoxon', 'anova', 'kruskal', 'pearson', 'spearman', limma2ways', 'limma2waysInter', 'anova2ways', 'anova2waysInter')
factor_names.vc	(Vector of) character(s): Factor(s) of interest (up to two), i.e. name(s) of a column from the pData(x)
factor_levels.ls	List: for each factor of interest (up to two), the levels of the factor can be specified (i.e. re-ordered) by including a character vector with those levels in the list; by default (no specification), the two vectors are set to "default".
adjust.c	character(1): Name of the method for correction of multiple testing (the p.adjust function is used)
adjust_thresh.n	numeric(1): Threshold for (corrected) p-values
signif_maxprint.i	integer(1): Maximum number of significant feature to display on the screen (by default, 'NA', all significant features are displayed)
title.c	character(1): Title of the graphics
display_signif.l	logical(1): In case of two sample tests (or correlation test), should individual boxplots (or scatterplots) of significant features be shown?
prefix.c	character(1): prefix to be added to the supplementary columns from the variableMetadata to prevent overwriting of pre-existing columns with identical names [default: ""]
figure.c	character(1): File name with '.pdf' extension for the figure (for venn diagrams, e.g. in the 'anova2ways' test, the extension will be internally changed to '.tiff' for compatibility with the VennDiagram package); if interactive' (default), figures will be displayed interactively; if 'none', no figure will be generated
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

SummarizedExperiment or MultiAssayExperiment (or ExpressionSet and MultiDataSet) including the difference in means/medians or correlations and the adjusted p-values in feature metadata

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine.se <- correcting(sacurine.se, figure.c = 'none')
sacurine.se <- sacurine.se[, colData(sacurine.se)[, "sampleType"] != "pool"]
sacurine.se <- transforming(sacurine.se)
sacurine.se <- sacurine.se[, colnames(sacurine.se) != "HU_neg_096_b2"]
# Student's T test
sacurine.se <- hypotesting(sacurine.se, "ttest", "gender")
# Pearson correlation test
sacurine.se <- hypotesting(sacurine.se, "pearson", "age")
# ANOVA
colData(sacurine.se)[, "ageGroup"] <- vapply(colData(sacurine.se)[, "age"],
                                             function(x) {
                                               if (x < 35) {
                                                 return("thirty")
                                               } else if (x < 50) {
                                                 return("fourty")
                                               } else {
                                                 return("fifty")}},
                                             FUN.VALUE = character(1))
sacurine.se <- hypotesting(sacurine.se, "anova", "ageGroup")

# MultiAssayExperiment

prometis.mae <- reading(system.file("extdata/prometis", package="phenomis"))
prometis.mae <- hypotesting(prometis.mae, "limma", "gene")

# MultiDataSet

prometis.mset <- reading(system.file("extdata/prometis", package="phenomis"),
                         output.c = "set")
prometis.mset <- hypotesting(prometis.mset, "limma", "gene")
```

---

inspecting *Inspecting*

Description

Provides numerical metrics and graphical overview of SummarizedExperiment, MultiAssayExperiment, ExpressionSet, or MultiDataSet instance

Usage

```
inspecting(
  x,
  pool_as_pool1.l = FALSE,
  pool_cv.n = 0.3,
  loess_span.n = 1,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  plot_dims.l = TRUE,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
inspecting(
  x,
  pool_as_pool1.l = FALSE,
  pool_cv.n = 0.3,
  loess_span.n = 1,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  plot_dims.l = TRUE,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
inspecting(
  x,
  pool_as_pool1.l = FALSE,
  pool_cv.n = 0.3,
  loess_span.n = 1,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  plot_dims.l = TRUE,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
inspecting(
  x,
  pool_as_pool1.l = FALSE,
  pool_cv.n = 0.3,
  loess_span.n = 1,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  plot_dims.l = TRUE,
```

```
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
inspecting(
  x,
  pool_as_pool1.l = FALSE,
  pool_cv.n = 0.3,
  loess_span.n = 1,
  sample_intensity.c = c("median", "mean", "sum")[2],
  title.c = NA,
  plot_dims.l = TRUE,
  figure.c = c("none", "interactive", "myfile.pdf")[2],
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

x	An S4 object of class SummarizedExperiment or MultiAssayExperiment (ExpressionSet and MultiDataSet are still supported)
pool_as_pool1.l	logical(1): should pool be included (as pool1) in the correlation with the dilution factor? [default = FALSE]
pool_cv.n	numeric(1): threshold for the coefficient of variation of the pools; the default value (30%) is often used in metabolomics
loess_span.n	numeric(1): span parameter used in the loess trend estimation; the default value is set to 1 to prevent overfitting
sample_intensity.c	Character: function to be used to display the global sample intensity; default: 'mean'
title.c	character(1): MultiAssayExperiment: title of the barplot showing the number of samples and variables in each dataset; ExpressionSet: title of the multipanel graphic displaying the metrics (if NA -default- the title slot from the experimentData will be used)
plot_dims.l	(MultiAssayExperiment) logical(1): should an overview of the number of samples and variables in all datasets be barplotted?
figure.c	character(1): File name with '.pdf' extension for the figure; if 'interactive' (default), figures will be displayed interactively; if 'none', no figure will be generated
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

SummarizedExperiment or MultiAssayExperiment (or ExpressionSet and MultiDataSet) including the computed sample and variable metrics in the rowData and colData metadata.

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine.se <- inspecting(sacurine.se)
sacurine.se <- correcting(sacurine.se)
sacurine.se <- inspecting(sacurine.se)
sacurine.se <- transforming(sacurine.se)
sacurine.se <- inspecting(sacurine.se)

# MultiAssayExperiment
prometis.mae <- reading(system.file("extdata/prometis",
                                    package = "phenomis"))
prometis.mae <- inspecting(prometis.mae)
```

---

normalizing *Normalization of the data matrix intensities*

Description

The matrix intensities may be normalized by using the Probabilistic Quotient Normalization to scale the spectra to the same virtual overall concentration

Usage

```
normalizing(
  x,
  method.vc = "pqn",
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
normalizing(
  x,
  method.vc = "pqn",
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
normalizing(
  x,
  method.vc = "pqn",
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
normalizing(
  x,
  method.vc = "pqn",
```

```
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
normalizing(
  x,
  method.vc = "pqn",
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

x	An S4 object of class SummarizedExperiment or MultiAssayExperiment (ExpressionSet and MultiDataSet are still supported)
method.vc	character of length 1 or the total number of datasets: method(s) to be used for each dataset (default is 'pqn'); in case the parameter is of length 1 and x contains multiple datasets, the same method will be used for all datasets
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

SummarizedExperiment or MultiAssayExperiment (or ExpressionSet and MultiDataSet) including the (list of) matrix with normalized intensities

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine.se <- sacurine.se[, colnames(sacurine.se) != 'HU_neg_096_b2']
sacurine.se <- transforming(sacurine.se, method.vc = "log10")
norm.se <- normalizing(sacurine.se, method.vc = "pqn")

# MultiDataSet
```

---

reading *reading*

Description

Reading dataset(s) in the 3 tables 'dataMatrix' (or 'DM'), sampleMetadata' (or 'SM') and variableMetadata' (or 'VM') tabular format. In case of a single dataset (3 tables in the specified directory), a SummarizedExperiment instance is returned. In case of a multiple dataset (several subfolders containing 3 tables), a MultiAssayExperiment instance is created.

Usage

```
reading(
  dir.c,
  files.ls = NULL,
  subsets.vc = NA,
  output.c = c("exp", "set")[1],
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

dir.c	character(1): directory containing the 3 tabular files (single dataset), or containing several subdirectories with 3 tabular files (multiple datasets)
files.ls	list: if dir.c is set to NA, the full names of the individual files can be provided; in case of a SummarizedExperiment, the names of the list must be 'dataMatrix', 'sampleMetadata', and 'variableMetadata' with the corresponding file full names; in case of a MultiAssayExperiment, the list must consists of one such sublist per dataset
subsets.vc	character(): specifying a subset of the subdirectories to be included in the MultiAssayExperiment (by default, all subdirectories containing the 3 tables will be considered as datasets)
output.c	character(1): Either 'exp' for SummarizedExperiment (or MultiAssayExperiment), or 'set' for ExpressionSet (or MultiDataSet) output formats (the latter are supported for convenience)
report.c	character(1): File name for the printed results (call to 'sink()'); if NA (default), messages will be printed on the screen; if NULL, no verbose will be generated

Value

`SummarizedExperiment` (one dataset) or `MultiAssayExperiment` (multiple datasets) instance containing the dataset(s)

Examples

```
data_dir.c <- system.file("extdata", package = "phenomis")
## 1) Single set
sacurine_dir.c <- file.path(data_dir.c, "sacurine")
sacurine.se <- reading(sacurine_dir.c)
# or
sacurine.se <- reading(NA,
                  files.ls = list(dataMatrix = file.path(sacurine_dir.c,
                                  "Galaxy1_dataMatrix.tabular"),
                                  sampleMetadata = file.path(sacurine_dir.c,
                                  "Galaxy2_sampleMetadata.tabular"),
                                  variableMetadata = file.path(sacurine_dir.c,
                                  "Galaxy3_variableMetadata.tabular")))
## 2) Multiple sets
prometis_dir.c <- file.path(data_dir.c, "prometis")
prometis.mae <- reading(prometis_dir.c)
```

```
metabo.mae <- reading(prometis_dir.c, subsets.vc = "metabo")
# or
prometis.mae <- reading(NA,
                    files.ls = list(metabo = list(dataMatrix = file.path(prometis_dir.c,
                                                                 "metabo", "dataMatrix.tsv"),
                                                  sampleMetadata = file.path(prometis_dir.c,
                                                                   "metabo", "sampleMetadata.tsv"),
                                                variableMetadata = file.path(prometis_dir.c,
                                                                   "metabo", "variableMetadata.tsv")),
                                    proteo = list(dataMatrix = file.path(prometis_dir.c,
                                                                 "proteo", "dataMatrix.tsv"),
                                                  sampleMetadata = file.path(prometis_dir.c,
                                                                   "proteo", "sampleMetadata.tsv"),
                                                variableMetadata = file.path(prometis_dir.c,
                                                                   "proteo", "variableMetadata.tsv"))))
```

`reducing` *Grouping chemically redundant MS1 features*

Description

This method groups chemically redundant features from a peak table, based on 1) correlation of sample profiles, 2) retention time window, 3) referenced m/z differences. The initial algorithm is named 'Analytic Correlation Filtration' (Monnerie et al., 2019; DOI:10.3390/metabo9110250) and is available in Perl and on the Workflow4Metabolomics platform. Here, the algorithm described in the paper was implemented in R as follows: An adjacency matrix of all pairs of features is built, containing a 1 when the features have a (Pearson) correlation above the (0.9) threshold, a retention time difference between the (6) seconds threshold, and an m/z difference belonging to referenced adducts, isotopes and fragments m/z difference, and containing a 0 otherwise. The connex components of this adjacency matrix are extracted ('igraph' package). Within each component, the features are ranked by decreasing average intensity in samples; all features except the first one are flagged as 'redundant'. Note: the algorithm relies on the 'mzdiff_db.tsv' file referencing the known adducts, isotopes, and fragments.

Usage

```
reducing(
  x,
  cor_method.c = "pearson",
  cor_threshold.n = 0.9,
  rt_tol.n = 6,
  rt_colname.c = "rt",
  mzdiff_tol.n = 0.005,
  mz_colname.c = "mz",
  return_adjacency.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

```
## S4 method for signature 'MultiAssayExperiment'
reducing(
  x,
  cor_method.c = "pearson",
  cor_threshold.n = 0.9,
  rt_tol.n = 6,
  rt_colname.c = "rt",
  mzdiff_tol.n = 0.005,
  mz_colname.c = "mz",
  return_adjacency.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
reducing(
  x,
  cor_method.c = "pearson",
  cor_threshold.n = 0.9,
  rt_tol.n = 6,
  rt_colname.c = "rt",
  mzdiff_tol.n = 0.005,
  mz_colname.c = "mz",
  return_adjacency.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
reducing(
  x,
  cor_method.c = "pearson",
  cor_threshold.n = 0.9,
  rt_tol.n = 6,
  rt_colname.c = "rt",
  mzdiff_tol.n = 0.005,
  mz_colname.c = "mz",
  return_adjacency.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
reducing(
  x,
  cor_method.c = "pearson",
  cor_threshold.n = 0.9,
  rt_tol.n = 6,
  rt_colname.c = "rt",
  mzdiff_tol.n = 0.005,
  mz_colname.c = "mz",
```

```
  return_adjacency.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

x	An S4 object of class SummarizedExperiment or MultiAssayExperiment (ExpressionSet and MultiDataSet are still supported): the dataset(s) must contain the dataMatrix and the variableMetadata (with the mz' and 'rt' columns)
cor_method.c	character(1): correlation method (default: 'pearson')
cor_threshold.n	numeric(1): correlation threshold (default: 0.9)
rt_tol.n	numeric(1): retention time width in seconds (default: 6 s); the time window may be increased when using hydrophilic interaction (HILIC) chromatography
rt_colname.c	character(1): column name for the retention time in the rowData/fData (default: 'rt')
mzdiff_tol.n	numeric(1): tolerance in Da for the matching of m/z differences and referenced adducts, isotopes, and fragments (default: 0.005 Da)
mz_colname.c	character(1): column name for the m/z in the rowData/fData (default: 'mz')
return_adjacency.l	logical(1): should the adjacency matrix be returned (in addition to the updated SummarizedExperiment/ExpressionSet)?
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

updated SummarizedExperiment or MultiAssayExperiment (or ExpressionSet and MultiDataSet): the SummarizedExperiment(s) (resp. ExpressionSet(s)) now include(s) 5 new columns in the rowData (resp. fData): redund_samp_mean', 'redund_is', redund_group', redund_iso_add_frag', redund_repres' and 'redund_relative' containing, respectively, the redundant features (coded by 1; i.e. features with a relative annotation distinct from '' and 'M'), the connected components, the m/z diff. chemical annotations, the representative ion of each group, and the annotations relative to this representative ion within each group

Examples

```
metabo.se <- reading(system.file("extdata/prometis/metabo",
                                 package = "phenomis"),
                     report.c = "none")
metabo.se <- reducing(metabo.se,
                      rt_tol.n = 15)
# Note: in the 'prometis' example data set from this package, the chemical
# redundancy has already been filtered out
```

`transforming` *Transformation of the data matrix intensities*

Description

A logarithmic or square root transformation may be applied to the data matrix intensities in (each of) the data set (e.g. to stabilize the variance)

Usage

```
transforming(
  x,
  method.vc = c("log2", "log10", "sqrt", "none")[1],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
transforming(
  x,
  method.vc = c("log2", "log10", "sqrt", "none")[1],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
transforming(
  x,
  method.vc = c("log2", "log10", "sqrt", "none")[1],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
transforming(
  x,
  method.vc = c("log2", "log10", "sqrt", "none")[1],
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
transforming(
  x,
  method.vc = c("log2", "log10", "sqrt", "none")[1],
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

`x`	An S4 object of class `SummarizedExperiment` or `MultiAssayExperiment` (`ExpressionSet` and `MultiDataSet` are still supported)

method.vc	character of length 1 or the total number of datasets: transformation to be used for each dataset (either 'log2', 'log10', 'sqrt', or 'none')
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

SummarizedExperiment or MultiAssayExperiment (or ExpressionSet and MultiDataSet) including the (list of) matrix with transformed intensities

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine.se <- correcting(sacurine.se)
sacurine.se <- sacurine.se[, colData(sacurine.se)[, "sampleType"] != "pool"]
sacurine.se <- transforming(sacurine.se)
# MultiAssayExperiment
prometis.mae <- reading(system.file("extdata/prometis",
                                    package = "phenomis"))
prometis.mae <- transforming(prometis.mae, method.vc = c("log2", "none"))
# Note: in the 'prometis' example data set from the package, the data are
# already log2 transformed
```

---

vennplot *Venn diagram with VennDiagram*

Description

Venn diagram with VennDiagram

Usage

```
vennplot(
  input.ls,
  palette.vc = RColorBrewer::brewer.pal(9, "Set1")[seq_len(5)],
  title.c = NA,
  sub.c = "",
  cat_pos.vi = NA,
  label_col.c = "black",
  lwd.i = 2,
  inverted.l = FALSE,
  figure.c = "none"
)
```

Arguments

input.ls	Named list of vectors to be compared
palette.vc	Character vector: Color palette
title.c	Character: Plot title
sub.c	Character: Plot subtitle
cat_pos.vi	Integer vector giving the position (in degrees) of each category name along the circle, with 0 at 12 o'clock; if NA, (-50, 50), (-40, 40, 180), (-15, 15, 0, 0), and (0, 287.5, 215, 145, 70) values are used
label_col.c	Character: Label color
lwd.i	Integer: Width of the circle's circumference
inverted.l	Logical: Should the Venn diagram be flipped along its vertical axis (pairwise venn only)
figure.c	Character: Filename for image output (with either .tiff, .png, or .svg extensions); if 'none' (default) the grid object is displayed interactively

Value

invisible grid object

Examples

```
sacurine.se <- reading(system.file("extdata/sacurine", package = "phenomis"))
sacurine.se <- correcting(sacurine.se, figure.c = 'none')
sacurine.se <- sacurine.se[, colData(sacurine.se)[, "sampleType"] != "pool"]
sacurine.se <- transforming(sacurine.se)
sacurine.se <- sacurine.se[, colnames(sacurine.se) != "HU_neg_096_b2"]
# Student's T test
sacurine.se <- hypotesting(sacurine.se, "ttest", "gender")
# Wilcoxon T test
sacurine.se <- hypotesting(sacurine.se, "wilcoxon", "gender")
signif.ls <- list(ttest = which(rowData(sacurine.se)[, "ttest_gender_Female.Male_signif"] > 0),
wilcoxon =  which(rowData(sacurine.se)[, "wilcoxon_gender_Female.Male_signif"] > 0))
vennplot(signif.ls, label_col.c = "black",
title.c = "Signif. features\nwith Student or Wilcoxon tests")
```

---

writing *Exporting a SummarizedExperiment (or MultiAssayExperiment) instance into (subfolders with) the 3 tabulated files 'dataMatrix.tsv', sampleMetadata.tsv', 'variableMetadata.tsv'*

Description

Note that the dataMatrix is transposed before export (e.g., the samples are written column wise in the 'dataMatrix.tsv' exported file).

Usage

```
writing(
  x,
  dir.c,
  prefix.c = "",
  files.ls = NULL,
  overwrite.l = FALSE,
  metadata.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiAssayExperiment'
writing(
  x,
  dir.c,
  prefix.c = "",
  files.ls = NULL,
  overwrite.l = FALSE,
  metadata.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'SummarizedExperiment'
writing(
  x,
  dir.c,
  prefix.c = "",
  files.ls = NULL,
  overwrite.l = FALSE,
  metadata.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'MultiDataSet'
writing(
  x,
  dir.c,
  prefix.c = "",
  files.ls = NULL,
  overwrite.l = FALSE,
  metadata.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)

## S4 method for signature 'ExpressionSet'
writing(
  x,
  dir.c,
```

```
  prefix.c = "",
  files.ls = NULL,
  overwrite.l = FALSE,
  metadata.l = FALSE,
  report.c = c("none", "interactive", "myfile.txt")[2]
)
```

Arguments

x	An S4 object of class SummarizedExperiment or MultiAssayExperiment (ExpressionSet and MultiDataSet are still supported)
dir.c	character(1): directory where each dataset should be written
prefix.c	character(1): prefix to be used (followed by '_') in the 'dataMatrix.tsv', 'sampleMetadata.tsv', and 'variableMetadata.tsv' file names
files.ls	list: alternatively to the dir.c argument, the full names of the files can be provided as a list
overwrite.l	logical(1): should existing files be overwritten?
metadata.l	logical(1): should the metadata be saved (as an additional .rds file)?
report.c	character(1): File name with '.txt' extension for the printed results (call to sink()'); if 'interactive' (default), messages will be printed on the screen; if 'none', no verbose will be generated

Value

No object returned.

Examples

```
metabo.se <- reading(system.file("extdata/prometis/metabo",
                     package = "phenomis"))

writing(metabo.se, dir.c = file.path(getwd(), "metabo"))

# MultiAssayExperiment
prometis.mae <- reading(system.file("extdata/prometis",package="phenomis"))

writing(prometis.mae, dir.c = file.path(getwd(), "prometis"))
# alternatively
writing(prometis.mae,
         dir.c = NA,
         files.ls = list(metabo = list(dataMatrix = file.path(getwd(),
                                              "met_dataMatrix.tsv"),
                                       sampleMetadata = file.path(getwd(),
                                       "met_sampleMetadata.tsv"),
                                       variableMetadata = file.path(getwd(),
                                       "met_variableMetadata.tsv")),
                         proteo = list(dataMatrix = file.path(getwd(),
                                       "pro_dataMatrix.tsv"),
                                       sampleMetadata = file.path(getwd(),
```

```
"pro_sampleMetadata.tsv"),
variableMetadata = file.path(getwd(),
"pro_variableMetadata.tsv"))))
```

Index