Package ‘phoenics’

January 29, 2025

Type Package

Title Pathways Longitudinal and Differential Analysis in Metabolomics

Version 0.5

Date 2025-01-29

Depends R (>= 4.0.0)

Imports lme4, blme, tibble, FactoMineR, factoextra, tidyr, stats

Suggests knitr, KEGGREST

Maintainer Camille Guilmineau <`camille.guilmineau@inrae.fr`>

Description Perform a differential analysis at pathway level based on metabolite quantifications and information on pathway metabolite composition.
The method is based on a Principal Component Analysis step and on a linear mixed model.
Automatic query of metabolic pathways is also implemented.

License GPL-3

Repository CRAN

BugReports `https://forgemia.inra.fr/panoramics/phoenics/-/issues`

Encoding UTF-8

NeedsCompilation no

URL `https://forgemia.inra.fr/panoramics/phoenics`

LazyData true

RoxygenNote 7.3.2

VignetteBuilder knitr

Language en-US

Author Camille Guilmineau [aut, cre],
Remi Servien [aut] (<`https://orcid.org/0000-0003-1270-1843`>),
Nathalie Vialaneix [aut] (<`https://orcid.org/0000-0003-1156-0639`>)

Date/Publication 2025-01-29 14:20:06 UTC

Contents

`from_ASICS_to_PHOENICS` *Prepare quantification data from ASICS outputs*

Description

Prepare quantification data from `ASICS` outputs for `test_pathway`. In short, it replaces metabolite names by metabolites KEGG codes and transposes the matrix to have samples in rows and metabolites in columns.

Usage

```
from_ASICS_to_PHOENICS(quantif)
```

Arguments

`quantif`	output matrix of `ASICS` quantification

Value

A matrix of quantification with samples in rows and metabolites in columns, properly formatted for `test_pathway`

Author(s)

Camille Guilmineau <camille.guilmineau@inrae.fr>
Remi Servien <remi.servien@inrae.fr>
Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

References

Lefort G., Liaubet L., Canlet C., Tardivel P., P\'ere M.C., Quesnel H., Paris A., Iannuccelli N., Vialaneix N. Servien R. (2019). ASICS: an R package for a whole analysis workflow of 1D 1H NMR spectra. *Bioinformatics*, **35**(21): 4356–4363. doi:10.1093/bioinformatics/btz248

Tardivel P., Canlet C., Lefort G., Tremblay-Franco M., Debrauwer L., Concordet D., Servien R. (2017). ASICS: an automatic method for identification and quantification of metabolites in complex 1D 1H NMR spectra. *Metabolomics*, **13**(10): 109. doi:10.1007/s1130601712445

Examples

```
data("MTBLS422")
quantif <- from_ASICS_to_PHOENICS(quantif)
```

MTBLS422 *Dataset "MTBLS422"*

Description

Metabolites quantifications, associated metabolic pathways and experimental design to characterize the effects of two clinically important antibiotic treatments, ciprofloxacin and vancomycin-imipenem on mice.

Format

3 datasets are provided:

- `quantif`: a data frame with 10 rows (metabolites name) and 11 columns (samples id), which corresponds to the metabolites quantifications in the samples.
- `pathways`: a data frame with 11 rows and 4 columns, which contains informations about pathways and their metabolites.
- `design`: a data frame with 11 rows (samples id) and 4 columns (id and effects to be added in the model).

Details

The raw dataset have been made available on MetaboLights (with the id MTBLS422 `https://www.ebi.ac.uk/metabolights/editor/MTBLS422`) by [Choo *et al.*, 2017]. Metabolite quantifications were obtained based on the raw signal using `ASICS` package. Pathways were obtained using `KEGGREST` package. The datasets provided for the example are a subset of the original dataset.

References

Choo J. M., Kanno T., Zain N. M. M., Leong L. E. X., Abell G. C. J., Keeble J. E., Bruce K. D., Mason A. J., Rogers G. B. (2017). Divergent relationships between fecal microbiota and metabolome following distinct antibiotic-induced disruptions. *mSphere*, **2**(1) doi:10.1128/msphere.0000517

Lefort G., Liaubet L., Canlet C., Tardivel P., P\‘ere M.C., Quesnel H., Paris A., Iannuccelli N., Vialaneix N. Servien R. (2019). ASICS: an R package for a whole analysis workflow of 1D 1H NMR spectra. *Bioinformatics*, **35**(21): 4356–4363. doi:10.1093/bioinformatics/btz248

Tardivel P., Canlet C., Lefort G., Tremblay-Franco M., Debrauwer L., Concordet D., Servien R. (2017). ASICS: an automatic method for identification and quantification of metabolites in complex 1D 1H NMR spectra. *Metabolomics*, **13**(10): 109. doi:10.1007/s1130601712445

Tenenbaum D., Maintainer B. (2022). KEGGREST: Client-side REST access to the Kyoto Encyclopedia of Genes and Genomes (KEGG). R package version 1.38.0.

Examples

```
data(MTBLS422)

design[1:5, ]
pathways[1:5, ]
quantif[1:5, 1:5]
```

`overlap_coefficient` *Calculate overlap coefficient between pathways*

Description

Calculate overlap coefficient between pathways

Usage

```
overlap_coefficient(pathwayA, pathwayB, pathways = NULL, organism = NULL)
```

Arguments

`pathwayA`	a character string of pathway name or pathway code or a character vector of metabolite names or metabolite codes
`pathwayB`	a character string of pathway name or pathway code or a character vector of metabolite names or metabolite codes
`pathways`	data.frame or matrix with metabolites in rows and the following information in columns: • `metabolite_code` metabolite code • `metabolite_name` metabolite name • `pathway_code` pathway code (identifier) • `pathway_name` name of the pathway Used if `pathwayA` and `pathwayB` are pathway names or pathway codes.
`organism`	organism code in KEGG database. Required if `pathways` = `NULL` and `pathwayA` and `pathwayB` are pathway names and ignored otherwise.

Value

A value between 0 and 1 calculated with the formula:

$$\mathrm{OC}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}$$

An overlap coefficient of 1 means that one pathway is included in the other. An overlap coefficient of 0 means that there is no overlap between the pathways.

References

Wieder C., Lai R.P.J., Ebbels T.M.D. (2022). Single sample pathway analysis in metabolomics: performance evaluation and application. *BMC Bioinformatics*, **23**, 481. doi:10.1186/s12859022-050051

Examples

```
data("MTBLS422")
pathwayA <- "Galactose metabolism"
pathwayB <- "Vitamin digestion and absorption"
overlap_coefficient(pathwayA, pathwayB, pathways)

if (requireNamespace("KEGGREST", quietly = TRUE)) {
  pathwayA <- "Galactose metabolism"
  pathwayB <- "Vitamin digestion and absorption"

  overlap_coefficient(pathwayA, pathwayB, organism = "mmu")

}

pathwayA <- "mmu00052"
pathwayB <- "mmu00562"
overlap_coefficient(pathwayA, pathwayB, pathways)

pathwayA <- c("C00029", "C00116", "C00137", "C00794", "C00984", "C01697")
pathwayB <- c("C00191", "C00092", "C00137")
overlap_coefficient(pathwayA, pathwayB)
```

`pathwayRes` *Class pathwayRes*

Description

S3 class for pathway differential analysis results

Usage

```
## S3 method for class 'pathwayRes'
summary(object, ...)

## S3 method for class 'pathwayRes'
print(x, ...)

## S3 method for class 'pathwayRes'
plot(
  x,
  ...,
```

```
  pathway_id = NULL,
  plot = c("eig", "var", "ind", "group"),
  habillage = "none"
)

## S3 method for class 'pathwayRes'
head(x, ...)

extract(object, pathway_id)

adjust_pval(object, method = p.adjust.methods, n = length(object))
```

Arguments

`object, x`	object of class `pathwayRes`
`...`	not used
`pathway_id`	a character string or vector of pathway codes or names
`plot`	a character string indicating the type of plot to return. Default to `"eig"` (the screegraph of the PCA or MFA is displayed)
`habillage`	a character string indicating the column of the design used to color the individuals. Only used when `plot = "ind"`. Default to `"none"` (no color)
`method`	a character string indicating the correction method to be used for multiple testing correction (authorized values are those of `p.adjust.methods`)
`n`	number of comparisons for multiple testing correction

Details

Methods for the class pathwayRes

Value

The function `extract` returns an object of class `pathwayRes` which is a list of pathway results, containing only the pathways in `pathway_id`.

The function `adjust_pval` returns a data.frame with pathways in rows and the following information in columns:

`pathway_name`	name of the pathway
`pathway_code`	pathway code (identifier)
`Fixed_effect`	tested effect
`pval`	raw p-value of the pathway
`adjusted_pval`	adjusted p-value of the pathway

Examples

```
data("MTBLS422")
quantif <- from_ASICS_to_PHOENICS(quantif)
out_test <- test_pathway(quantif, design, pathways,
                         fixed = c("Age", "Treatment"), random = "Mouse",
                         npc = 2, model = "blmer")
summary(out_test)
out_test2 <- test_pathway(quantif, design, pathways,
                         fixed = c("Age", "Treatment"), random = "Mouse",
                         npc = 2, model = "blmer", analysis = "MFA")
summary(out_test2)
print(out_test)
print(out_test2)
plot(out_test)
plot(out_test, pathway_id = "mmu00052", plot = "var")
plot(out_test, pathway_id = "mmu00052", plot = "ind", habillage = "Age")
plot(out_test2, pathway_id = "mmu00562", plot = "eig")
plot(out_test2, pathway_id = "mmu00562", plot = "var")
plot(out_test2, pathway_id = "mmu00562", plot = "ind")
plot(out_test2, pathway_id = "mmu00562", plot = "group")
extract(out_test, "mmu00562")
adj_pval <- adjust_pval(out_test)
```

`pathway_search` *Query KEGG pathways for a given set of metabolites*

Description

Query KEGG pathways for a given set of metabolites

Usage

```
pathway_search(metab, organism, min_size = 2)
```

Arguments

`metab`	vector of metabolite KEGG codes
`organism`	organism code in KEGG database
`min_size`	minimal number of metabolites required for a pathway to be returned

Value

a data.frame with metabolites in rows and the following information in columns:

- `metabolite_code` metabolite code
- `metabolite_name` metabolite name
- `pathway_code` pathway code (identifier)
- `pathway_name` name of the pathway

Author(s)

Camille Guilmineau <camille.guilmineau@inrae.fr>
Remi Servien <remi.servien@inrae.fr>
Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

References

Kanehisa M., Goto S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes, *Nucleic Acids Research*, Volume 28, Issue 1, Pages 27–30, doi:10.1093/nar/28.1.27

Tenenbaum D., Maintainer B. (2022). KEGGREST: Client-side REST access to the Kyoto Encyclopedia of Genes and Genomes (KEGG). R package version 1.38.0.

Examples

```
if (requireNamespace("KEGGREST", quietly = TRUE)) {
  data("MTBLS422")
  quantif <- from_ASICS_to_PHOENICS(quantif)

  pathways <- pathway_search(metab = colnames(quantif), organism = "mmu")

}
```

`test_pathway` *Pathway differential analysis based on longitudinal metabolomics data*

Description

Perform a differential analysis at pathway level based on metabolite quantifications and information on pathway metabolite composition. The method relies on a PCA or a MFA step.

Usage

```
test_pathway(
  quantif,
  design,
  pathways = "auto",
  fixed,
  random,
  npc = 1L,
  model = c("lmer", "blmer"),
  organism = NULL,
  min_size = 2,
  analysis = c("PCA", "MFA")
)
```

Arguments

`quantif`	data.frame or matrix of the metabolite quantification with samples in rows (sample identifiers must be row names) and metabolites in columns (metabolite codes must be column names)
`design`	data.frame or matrix with samples in rows (sample identifiers must be row names) and the different effects to be included in the model in columns. Column names must be provided and are used as arguments for `fixed` and `random`
`pathways`	data.frame or matrix with metabolites in rows (all metabolites in columns of `quantif` must have a row in this input) and the following information in columns: • `metabolite_code` metabolite code • `metabolite_name` metabolite name • `pathway_code` pathway code (identifier) • `pathway_name` name of the pathway
`fixed`	character vector of fixed effects to be included in the model. They must correspond to column names of `design`. If `analysis = "MFA"`, the first fixed effect must correspond to the time effect
`random`	character vector of random effects to be included in the model. They must correspond to column names of `design`
`npc`	number of PCs for the PCA step
`model`	a character string indicating if the model has to be fitted using lmer or blmer. Default to `"lmer"`
`organism`	organism code in KEGG database. Required if `pathways = "auto"` and ignored otherwise
`min_size`	minimal number of metabolites in a pathway. Required if `pathways = "auto"` and ignored otherwise. Default to `2`
`analysis`	character string indicating if the pathway scores are obtained using PCA or MFA

Details

If `pathways = "auto"`, information on pathways in which metabolites are present is automatically obtained by the function pathway_search using `KEGGREST` that queries KEGG database. Results are specific to a given organism (passed in `organism`). Pathways containing less than `min_size` metabolites are filtered out.

Value

an object of class `PCApath` or `MFApath` that inherits from class `pathwayRes` (a list of pathway results). Each element of the list contains the following entries:

`pathway_name`	a character corresponding to the pathway name
`pathway_code`	a character corresponding to the pathway code
`metabolites`	a data.frame with the names and codes of the quantified metabolites in the pathway
`PCA`	the result of the pathway PCA or MFA (a `PCA` object as obtained from PCA or a `MFA` object as obtained from MFA)

`model`	the output of the mixed model fit
`test_pathway`	a data.frame with the p-values for each tested fixed effect

Author(s)

Camille Guilmineau <camille.guilmineau@inrae.fr>
Remi Servien <remi.servien@inrae.fr>
Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

See Also

pathway_search for the automatic annotation of metabolites in KEGG pathways.

Examples

```
data("MTBLS422")
quantif <- from_ASICS_to_PHOENICS(quantif)
out_test <- test_pathway(quantif, design, pathways,
                         fixed = c("Age", "Treatment"), random = "Mouse",
                         npc = 2, model = "blmer")
out_test

out_test2 <- test_pathway(quantif, design, pathways,
                         fixed = c("Age", "Treatment"), random = "Mouse",
                         npc = 2, model = "blmer", analysis = "MFA")
out_test2

if (requireNamespace("KEGGREST", quietly = TRUE)) {

  out_test3 <- test_pathway(quantif, design, pathways = "auto",
                            fixed = c("Age", "Treatment"), random = "Mouse",
                            npc = 2, model = "blmer", organism = "mmu")
  out_test3

}
```

Index