Package 'matR'

May 4, 2018

Type Package

Title Metagenomics Analysis Tools

Version 0.9.1

Depends R (>= 2.10), MGRASTer, BIOM.utils, graphics, stats, utils

Imports

Suggests RJSONIO, qvalue, ecodist, gplots, scatterplot3d

URL https://github.com/MG-RAST/matR/

Date 2018-05-04

Description An analysis platform for metagenomics combining specialized tools and workflows, easy handling of the BIOM format, and transparent access to MG-RAST resources. Integrates easily with other R packages and non-R software.

License BSD_2_clause + file LICENSE

Copyright University of Chicago

LazyData yes

Collate utils.R biom-ext.R client.R graphics.R analysis-support.R analysis-misc.R distx.R rowstats.R transform.R boxplot.R princomp.R image.R init.R

NeedsCompilation no

Author Daniel T. Braithwaite [aut],
Kevin P. Keegan [aut],
William L. Trimble [cre, ctb],
University of Chicago [cph]

Maintainer William L. Trimble <trimble@anl.gov>

Repository CRAN

Date/Publication 2018-05-04 20:34:53 UTC

R topics documented:

`BIOMannotations` *Search and change row or column annotations of BIOM data*

Description

For an object of class `biom`, find row or column annotations (BIOM `metadata`) that match by name a given pattern, or append new annotations.

Usage

```
rows(x, pattern="*")
rows(x, name) <- value

columns(x, pattern="*")
columns(x, name) <- value
```

Arguments

`x`	an object (`biom`)
`pattern`	literal string or regular expression identifying metadata by name (`character`)
`name`	name for new metadata annotation (`character`)
`value`	new metadata, one value per row/column

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

For `rows()` and `columns()`, a `data.frame` consisting of the metadata of `x` matching `pattern` by name. For the replacement functions, the object `x` with updated metadata.

Author(s)

Daniel T. Braithwaite

See Also

`BIOM.utils::biom`, `regex`

Examples

```
####  exact sampling locations returned in a data.frame
columns (xx3, "latitude|longitude")

####  a data.frame is returned even in case of a single matching metadata field
is.data.frame (columns (xx1, "sample.data.biome"))

####  project IDs and environmental package metadata -- note regex here and above
colnames (columns (xx2, "project\\.id|^env_package"))

####  row metadata makes annotation hierarchy levels available,
####  so typical row metadata has few components, and here just two
names (rows (xx1))
rows (xx1, "ontology1")

####  here, the rownames and the (single) variable of the data.frame coincide
rows (xx1, "ontology2")

####  variables are almost always coded as factors
is.factor (columns (xx1, "sample.data.biome") [[1]])
```

`BIOMmerge` *Merge BIOM data*

Description

Merge two objects of class `biom`, maintaining metadata and other class structure.

Usage

```
## S3 method for class 'biom'
merge(x, y, ...)
```

Arguments

x	an object (biom)
y	an object (biom)
...	unused

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

A biom object resulting from merging x and y.

Author(s)

Daniel T. Braithwaite

See Also

BIOM.utils::biom

Examples

```
####  merging requires only that all colnames be unique, so nonsense can be performed
merge (xx1, xx4)

####  a more likely example, based on applying different normalizations
aa <- transform (xx4 [,1:8], t_Threshold, t_Log)
bb <- transform (xx4 [,9:16], t_Threshold=list(entry.min=5), t_Log)
xx4_norm <- merge (aa, bb)
```

BIOMrename *Change row and column identifiers of BIOM data*

Description

Change the dimnames (BIOM row and column ids) of an object of class biom.

Usage

```
## S3 replacement method for class 'biom'
dimnames(x) <- value
```

Arguments

x	an object (biom)
value	new row and column identifiers (list of character)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

The argument `x` with updated row and column `ids` (that is, `dimnames`).

Author(s)

Daniel T. Braithwaite

See Also

`BIOM.utils::biom`, `BIOM.utils::dimnames.biom`

Examples

```
####  even if not particularly useful, this is allowed
yy <- xx4
dimnames (yy) <- list (letters [1:nrow(yy)], LETTERS [1:ncol(yy)])

####  more useful: renaming columns by codes taken from metadata
colnames (yy) <- columns (yy, "sample.data.sample_name")
```

BIOMretrieval *Get annotation information of samples as BIOM data*

Description

Retrieve annotation pipeline information (such as abundance profiles) for specified metagenomes and projects into an object of class `biom`.

Usage

```
biomRequest(x, request=c("function", "organism"), ...,
  block, wait=TRUE, quiet=FALSE, file, outfile)

## S3 method for class 'environment'
biom(x, wait=TRUE, ..., quiet=FALSE)
```

Arguments

`x`	a set of metagenomes and/or projects (`character`) or a request ticket (`environment`)
`request`	choice of annotation type (string)
`...`	arguments specifying or qualifying the data desired (`biomRequest()` only)
`block`	number of metagenomes per API call (`integer`)

wait	return only when data is complete? (logical)
quiet	suppress messages and warnings? (logical)
file	file containing a set of metagenomes or projects (string)
outfile	file to save the retrieved data (string)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

For biomRequest(), an environment if wait=FALSE. Otherwise and for biom.environment(), a biom object.

Author(s)

Daniel T. Braithwaite

See Also

BIOM.utils::biom, MGRASTer::call.MGRAST

Examples

```
####  several files demonstrate valid formats for ID input
demoSets()

## Not run:
ff <- demoSets()

####  simple retrieval of annotation data
yy <- biomRequest (file=ff[1])
head (rows (yy))

####  many arguments can modify what is retrieved
yy <- biomRequest (file=ff[1], group_level="level1")
rownames (yy)

####  taxonomic annotations
yy <- biomRequest (file=ff[4], request="organism", group_level="phylum", source="Greengenes")

####  IDs can be given directly, while output can be to a file
biomRequest ("mgp9", request="function", outfile=file.path(tempdir("mgp9.biom")))
biomRequest ("mgm4441619.3 mgm4441620.3 mgm4441656.4",
  request="function", outfile=file.path(tempdir("mgp9.biom")))

####  place an asynchronous request...
yy <- biomRequest ("mgp9", wait=FALSE)
####  ...and receive the data when convenient
yy <- biom (yy)
```

```
## End(Not run)

####  full detail for available options
doc.MGRAST (3, head=c("matrix","function","parameters","options"))
doc.MGRAST (3, head=c("matrix","organism","parameters","options"))
```

BIOMsubset *Take part of (subset) BIOM data*

Description

Take part of (subset) an object of class `biom` by removing rows, columns, or both.

Usage

```
## S3 method for class 'biom'
x[i, j, ...]
```

Arguments

`x`	an object (`biom`)
`i`	row index (`integer`, `character`, or `logical`)
`j`	column index (`integer`, `character`, or `logical`)
`...`	unused

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

A `biom` object, the specified subset of `x`.

Author(s)

Daniel T. Braithwaite

See Also

`BIOM.utils::biom`, `BIOM.utils::dim.biom`

Examples

```
####  explicit subsetting
xx3 [ , 1:8]
xx4 [c ("Bacteria", "Eukaryota"), c ("mgm4575333.3", "mgm4575334.3", "mgm4575335.3")]

####  keep only metagenomes from one biome
xx3 [ , columns (xx3, "biome") == "Tundra biome"]

####  keep only rows matching a search term
xx1 [grepl ("Protein secretion system", rownames(xx1)), ]
```

`boxplot.biom` *Summarize BIOM data in boxplots*

Description

Summarize distribution of a `biom` object in columnwise boxplots.

Usage

```
## S3 method for class 'biom'
boxplot(x, y=NULL, ..., map=NULL, columns=TRUE)
```

Arguments

`x`	an object (`biom`)
`y`	optionally, a second object for comparison (`biom`)
`...`	arguments to `graphics::boxplot()`
`map`	assignment of `par` variables to metadata fields (`character`)
`columns`	subselection of columns (`integer`, `character`, or `logical`)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Author(s)

Daniel T. Braithwaite and Kevin P. Keegan

See Also

`graphics::boxplot`, `BIOM.utils::biom`

Examples

```
####  simple use
xx2t <- transform (xx2, t_Log)
boxplot (xx2t, main="log transformed data", notch=FALSE)

####  plotting raw and normalized against each other
columns (xx2t, "material")
boxplot (xx2t, xx2, x.main="log of data", y.main="raw data", map=c(col="material"),
  col=c("freshwater"="darkorange", "hot spring"="slateblue",
  "hot spring ; microbial mat"="chocolate4"), notch=FALSE)

####  label by metadata
columns (xx4, "sample_name")
boxplot (transform (xx4, t_Log), names="$$sample.data.sample_name", notch=FALSE)

####  two normalizations plotted against each other
xx2tt <- transform (xx2, t_Threshold=list(entry.min=5), t_Log)
boxplot (xx2t, xx2tt, notch=FALSE, x.main="log transformation",
  y.main="low counts removed, then log transformation")
```

`demoObjects` *BIOM annotation data for certain metagenomes and projects*

Description

Objects of class `biom` for demonstration purposes, containing annotation data for certain sets of metagenomes.

Usage

```
xx1
xx2
xx3
xx4
yy1
yy2
yy3
yy4
```

Details

`xx1` to `xx4` are packaged as examples to help users get started. They are objects of class `biom` that could be built with `biomRequest()`. They can be built in an automated way by `buildDemoSets()`.

`yy1` to `yy4` correspond to the example lists `li1` to `li4` from `BIOM.utils`. That is, `yy#` is equal to `biom(li#)`. These contain (all but the last) vacuous data.

See Also

`demoSets`, `buildDemoSets`

`dir.MGRAST` *List directory of projects*

Description

Get full or partial directory listing of projects, with minimal or detailed metadata per project.

Usage

```
dir.MGRAST(from, to, length.out=0, ..., quiet=TRUE)
```

Arguments

`from`	starting index for returned results (`numeric`)
`to`	ending index for returned results (`numeric`)
`length.out`	number of results to return; default zero means all (`numeric`)
`...`	arguments to `call.MGRAST()`
`quiet`	suppress messages and warnings? (`logical`)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

A `data.frame` of project information.

Author(s)

Daniel T. Braithwaite

See Also

`search.MGRAST`, `metadata.character`, `MGRASTer::call.MGRAST`

Examples

```
## Not run:
####  names of all public projects
dir.MGRAST()$name

####  ids of all public projects
rownames (dir.MGRAST())

####  investigators who have contributed public projects
unique (dir.MGRAST()$pi)
```

```
####  first 25 projects submitted to MG-RAST
dir.MGRAST (len=25, order="id")

####  detailed information about them
names (dir.MGRAST(len=25, order="id", verbosity="verbose"))

####  quick look at public projects
strtrim (dir.MGRAST (verbosity="verbose")$description, 70)

## End(Not run)

####  relevant documentation for the underlying API call
doc.MGRAST (3, head=c('project','query','parameters','options'))
```

`distx` *Calculate distances with optional grouping and other features*

Description

Calculate several distances and dissimilarities with optional grouping, by default columnwise and pairwise, or from an optionally specified common point.

Usage

```
distx(x, ...)

## S3 method for class 'matrix'
distx(x, method=c("euclidean", "bray-curtis", "jaccard", "mahalanobis",
  "sorensen", "difference", "maximum", "manhattan", "canberra", "binary", "minkowski"),
  groups=NULL, p=NULL, ..., bycol=TRUE)

## S3 method for class 'biom'
distx(x, method="euclidean", groups=NULL, ..., bycol=TRUE)
```

Arguments

`x`	a matrix-like object (`matrix` or `biom`)
`method`	name of distance or dissimilarity measure (`character`)
`groups`	a grouping of columns/rows (`character` or `factor`)
`p`	a single column/row (`numeric`)
`...`	unused
`bycol`	compute columnwise rather than rowwise? (`logical`)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Author(s)

Daniel T. Braithwaite and Kevin P. Keegan

See Also

`stats::dist`, `ecodist::distance`

Examples

```
####  Euclidean distance between samples based on raw counts
distx (xx1)

####  alternate dissimilarity measure
distx (xx1, method="bray-curtis")

####  distance in log-transformed data
distx (transform (xx2, t_Log))

####  mean pairwise distance between biomes
distx (xx3, groups="$$biome", method="bray-curtis")
```

IDsets *Handle sets of metagenome and project IDs*

Description

Utility functions to to establish a standard format and handle sets of metagenome and project IDs, possibly with metadata.

Usage

```
readSet(file)
expandSet(x)
scrapeSet(x)
scrubSet(x, resources="metagenome")
```

Arguments

`file`	a filename (`character`)
`x`	metagenome and project IDs, possibly with metadata (`character`, `numeric`, or `data.frame`)
`resources`	corresponding resource designation(s) (`character`)

Details

Internal and undocumented at present.

Value

Internal and undocumented at present.

Author(s)

Daniel T. Braithwaite

`image.biom` *Display heatmap of BIOM data with optional dendrograms*

Description

Display heatmap of a `biom` object with optional row and column dendrograms.

Usage

```
## S3 method for class 'biom'
image(x, ..., map=NULL, rows=TRUE, columns=TRUE, rerender=NULL)
```

Arguments

`x`	an object (`biom`)
`...`	arguments to `gplots::heatmap.2()`
`map`	assignment of `par` variables to metadata (`character`)
`rows`	subselection of rows (`integer`, `character`, or `logical`)
`columns`	subselection of columns (`integer`, `character`, or `logical`)
`rerender`	previous computation to reuse in this call (`heatmap`, `dclust`, `list`, or `dist`)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Note

The function `image()` is declared S3 generic in the base package `graphics`. The method documented here does not apply any existing methods, however, but rather relies on entirely different libraries for computation and graphical rendering. See reference below.

Author(s)

Daniel T. Braithwaite and Kevin P. Keegan

See Also

gplots::heatmap.2, BIOM.utils::biom

Examples

```
####  simple heatmap; using log transformation makes interesting things more apparent
image (xx2)
xx2t <- transform (xx2, t_Log)
image (xx2t, labCol="$$project.id")

####  clustering analysis restricted to Archaea
image (xx2t, labCol="$$project.id", rows=rows(xx2t,"taxonomy1")=="Archaea")

####  clustering analysis restricted by significance test p values
p <- rowstats (xx2t, test="t-test-unpaired", groups="$$material") $ p.value
p [is.na(p)] <- p [is.nan(p)] <- FALSE
image (xx2t [rows = p < 0.05, ], labCol="$$material")
```

---

metadata.character *Get metadata of projects and metagenomes*

Description

Get metadata of projects or metagenomes specified by ID, or simply look up correspondence of project and metagenome IDs.

Usage

```
## S3 method for class 'character'
metadata(x, detail=NULL, ..., quiet=TRUE, file)
```

Arguments

x	IDs of projects or metagenomes (character)
detail	level of metadata detail (NULL, TRUE, or character)
...	arguments to call.MGRAST()
quiet	suppress messages and warnings? (logical)
file	file containing project or metagenome IDs (string)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

A `list` (for projects) or `vector` (for metagenomes) when `detail=NULL`. Otherwise, a `data.frame`.

Note

The function `metadata()` is declared S3 generic in required package `BIOM.utils`, and a method is defined there for class `biom`. Although here is documented another method of the same function, the actual functionality is unrelated.

Author(s)

Daniel T. Braithwaite

See Also

`dir.MGRAST`, `search.MGRAST`, `biomRequest`, `MGRASTer::call.MGRAST`

Examples

```
## Not run:
####  three levels of detail for project metadata
xx <- "mgp21 mgp24 mgp30"
metadata (xx)
metadata (xx, detail=TRUE)
names (metadata (xx, detail="verbose"))

####  similar (but not identical) for metagenome metadata
yy <- "mgm4440066.3 mgm4440062.3 mgm4440055.3 mgm4441681.3 mgm4440463.3 mgm4440464.3"
metadata (yy)
metadata (yy, detail=TRUE)
names (metadata (yy, detail="metadata"))

## End(Not run)

####  relevant documentation for underlying API calls
doc.MGRAST (3, head=c('project','instance','parameters','options'))
doc.MGRAST (3, head=c('metagenome','instance','parameters','options'))
```

`princomp.biom` *Compute and plot principal coordinates of BIOM data*

Description

Compute principal coordinates of a `biom` object (columnwise), and plot selected coordinates.

Usage

```
## S3 method for class 'biom'
princomp(x, method="euclidean", dim=1:3, ...,
  map=NULL, rows=TRUE, columns=TRUE, rerender=NULL)
```

Arguments

x	an object (biom)
method	name of distance or dissimilarity measure (character)
dim	which principal coordinates to plot (integer)
...	arguments to scatterplot3d(), points(), or text()
map	assignment of par variables to metadata (character)
rows	subselection of rows (integer, character, or logical)
columns	subselection of columns (integer, character, or logical)
rerender	previous computation to reuse in this call (pco or dist)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Note

The function princomp() is declared S3 generic in the base package stats. The method documented here does not apply any existing methods, however, but rather relies on entirely different libraries for computation and graphical rendering. See references below.

Author(s)

Daniel T. Braithwaite and Kevin P. Keegan

See Also

ecodist::pco, scatterplot3d::scatterplot3d, graphics::points, graphics::text, distx, BIOM.utils::biom

Examples

```
####  quick two or three dimensional plots with choice of dissimilarity measure
princomp (xx1)
princomp (xx1, dim=2:3, method="bray-curtis")

####  graphical tweaks incorporating metadata
columns (xx1, "host_common_name|samp_store_temp")
princomp (xx1, dim=1:2, map=c(col="host_common_name", pch="samp_store_temp"),
  col=c(Mouse="brown", cow="red", "striped bass"="blue"),
  pch=c("-80"="+","NA"="*"), cex=2, label.pos=c(4,4,2,2,2,2,4), label.font=3)

####  transformed data, labeling from metadata, and modified perspective
```

```
columns (xx2, "material")
princomp (transform (xx2, t_Log), map=c(col="material"), labels="$$project.id",
  angle=50, mar=c(1,1,0,0))
```

`rowstats` *Apply selected significance test across rows*

Description

Apply selected significance test across rows to grouped columns, with optional q-value calculation.

Usage

```
rowstats(x, ...)

## S3 method for class 'matrix'
rowstats(x, groups,
  test=c("Kruskal-Wallis", "t-test-paired", "Wilcoxon-paired", "t-test-unpaired",
    "Mann-Whitney-unpaired-Wilcoxon", "ANOVA-one-way"),
  qvalue=FALSE, fdr.level=NULL, ...)

## S3 method for class 'biom'
rowstats(x, groups, ...)
```

Arguments

`x`	a matrix-like object (`matrix` or `biom`)
`groups`	a grouping of columns (`character` or `factor`)
`test`	name of statistical test (`character`)
`qvalue`	perform qvalue calculation? (`logical`)
`fdr.level`	false-discovery rate parameter, passed to `qvalue()` (`numeric`)
`...`	unused

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Author(s)

Kevin P. Keegan and Daniel T. Braithwaite

See Also

stats::t.test, stats::wilcox.test, stats::kruskal.test, stats::anova

Examples

```
####  Kruskal test applied, for the case of more than two metagenome groups
columns (xx1, "host_common_name")
str (rowstats (xx1, groups="$$host_common_name", test="Kruskal"))

####  force a desired grouping of metagenomes
gg <- columns (xx2, "material") [[1]]
gg
levels (gg) <- levels (gg) [c(1,2,2)]
str (rowstats (xx2, groups=gg, test="t-test-unpaired"))
```

---

search.MGRAST *Find metagenomes matching specified criteria*

Description

Find metagenomes matching search terms specified for metadata, annotations, and/or md5s, giving minimal or detailed metadata per metagenome.

Usage

```
search.MGRAST(public=NULL, detail=NULL, match.all=TRUE, ..., quiet=TRUE)
```

Arguments

public	optional restriction on sharing status (NULL or logical)
detail	level of metadata detail (NULL, TRUE, or character)
match.all	require match on all provided criteria? (logical)
...	arguments to call.MGRAST()
quiet	suppress messages and warnings? (logical)

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Author(s)

Daniel T. Braithwaite

See Also

dir.MGRAST, metadata.character, MGRASTer::call.MGRAST

Examples

```
####  relevant documentation for the underlying API call
doc.MGRAST (3, head=c('metagenome','query','parameters','options'))
```

---

transform.biom *Apply mathematical transformations to BIOM data*

Description

Prepare an object of class biom for further analysis by applying selected transformations with specified parameters.

Usage

```
## S3 method for class 'biom'
transform(`_data`, ...)

t_ColCenter(x, ...)
t_ColScale(x, ...)
t_Log(x, ...)
t_NA2Zero(x, ...)
t_Threshold(x, entry.min=2, row.min=2, col.min=2)
```

Arguments

_data	an object (biom)
x	a matrix
entry.min	minimum to retain an entry (numeric)
row.min	minimum sum to retain a row (numeric)
col.min	minimum sum to retain a column (numeric)
...	transformations to apply and arguments to them

Details

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Value

Complete technical documentation is forthcoming. For the current preliminary release, please refer to the examples provided.

Note

The function `transform()` is an S3 generic in base R. However, the method documented here does not apply any existing methods and offers rather different functionality.

Author(s)

Daniel T. Braithwaite and Kevin P. Keegan

See Also

`BIOM.utils::biom`, `transform`

Examples

```
####  simple log-transform
transform (xx1, t_Log)

####  additional filters
transform (xx1, t_NA2Zero, t_Threshold, t_Log)

####  what is lost with more stringent filtering of low-abundance annotations
yy <- transform (xx2, t_NA2Zero, t_Threshold, t_Log)
zz <- transform (xx2, t_NA2Zero, t_Threshold=list(entry.min=5, row.min=10), t_Log)
setdiff (rownames (yy), rownames (zz))

####  each sample centered around zero; scaling columnwise by standard deviation
transform (xx4, t_NA2Zero, t_Threshold, t_Log, t_ColCenter, t_ColScale)

####  defining a new transformation that indicates presence / absence
t_Indicator <- function (x, ...) { ifelse (x,1,0) }
transform (xx1, t_Threshold = list(entry.min=5), t_Indicator)
```

Index