Package: OmicsMLRepoR (via r-universe)

May 29, 2026

Type Package

Title Search harmonized metadata created under the OmicsMLRepo project

Version 1.6.0

Date 2026-3-19

Description This package provides functions to browse the harmonized metadata for large omics databases. This package also supports data navigation if the metadata incorporates ontology.

Depends R (>= 4.4.0)

Imports dplyr, stringr, rols, tidyr, methods, stats, tibble, data.tree, jsonlite, plyr, BiocFileCache, readr, DiagrammeR, rlang, lubridate

Suggests arrow, knitr, BiocStyle, curatedMetagenomicData, testthat (>= 3.0.0), cBioPortalData

License Artistic-2.0

biocViews Software, Infrastructure, DataRepresentation

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

URL `https://github.com/shbrief/OmicsMLRepoR`

BugReports `https://github.com/shbrief/OmicsMLRepoR/issues`

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev libssl-dev libx11-dev

Repository https://bioc-release.r-universe.dev

Date/Publication 2026-04-28 13:03:56 UTC

RemoteUrl https://github.com/bioc/OmicsMLRepoR

RemoteRef RELEASE_3_23

RemoteSha 75e8efd97d1079b727ac0719824a98d841450a28

Contents

availableTerms 2
get_ontologies 3
getLongMetaTb 3
getMetadata 4
getNarrowMetaTb 5
getOntoInfo 6
getShortMetaTb 7
getWideMetaTb 8
merge_vectors 9
mini_cbio 10
mini_cmd 10
mini_cmd2 11
mini_cmd3 11
OmicsMLRepoR 12
ontoSummarize 12
ontoTreePlot 13
sample_metadata 14
spreadMeta 14
tree_filter 15

Index **16**

`availableTerms` *Extract all the terms used in a quired attribute/column*

Description

Extract all the terms used in a quired attribute/column

Usage

```
availableTerms(attribute, db = "cMD")
```

Arguments

`attribute`	A character (1). Name of the attribute/column you want to extract the terms used under.
`db`	A character(1). Currently, ‘cMD‘ (curatedMetagenomicData) is only supported.

Value

A data frame with two columns - existing values under the quried attribute (‘allowedvalues‘ column) and their ontology term id (‘ontology‘ column).

Examples

```
availableTerms("age_group")
availableTerms("disease")
```

`get_ontologies` *Extract ontology from the ontology term ids*

Description

Extract ontology from the ontology term ids

Usage

```
get_ontologies(terms, delim = ":")
```

Arguments

`terms`	A character vector
`delim`	A character. Delimiter between ontology and its id. Default is ':'.

Value

A character vector containing the ontology names of the input 'terms'. The length of this is same as the 'terms' input.

Examples

```
terms <- c("HP:0001824", "MONDO:0010200", "NCIT:C122328", "4471000175100")
get_ontologies(terms = terms)
```

`getLongMetaTb` *Expands metadata with multiple values belong to a same attribute*

Description

Same as the 'getWideMetaTb' function, this function accepts single target column ('targetCol'). The target columns (and linked, accessory columns) should have the same number of elements separated by the 'delim', and the multiple values for each column belongs to the same column/attribute/field, i.e., no additional column name is required/provided.

Usage

```
getLongMetaTb(meta, targetCol = NULL, delim = NULL)
```

Arguments

`meta`	A data frame. Each column (and associated ‘ontology_term_id‘ column) should use the same delimiter to separate multiple, same-numbered values.
`targetCol`	A character. The column name(s) to expand if present. If multiple columns are specified here, they should be separated by the ‘sep‘ and the column name and its value should be separated by the provided ‘delim‘.
`delim`	Optional. A character (1) of a delimiter used to separate multiple values in the metadata table.

Value

A data frame of metadata expanded so that each individual treatment has its own row.

Examples

```
data(mini_cmd)
lmeta <- getLongMetaTb(mini_cmd, "hla")
dim(mini_cmd)
dim(lmeta)

data(mini_cmd2)
lmeta2 <- getLongMetaTb(mini_cmd2, "target_condition")
head(lmeta2, 3)

data(mini_cbio)
trt_cols <- grep("^treatment_", colnames(mini_cbio), value = TRUE)
lmeta3 <- getLongMetaTb(mini_cbio, targetCol = trt_cols)

short_tb <- data.frame(
    ind = c("A", "B", "C", "D", "E"),
    aval = c("cat;dog", "chicken", "horse", "frog;pig", "snake"),
    cval = c(1, NA, 3, 4, 5),
    bval = c("red;blue", "yellow", "NA", "green;NA", "brown"))

getLongMetaTb(short_tb, c("aval", "bval"), delim = ";")
```

`getMetadata` *Download a harmonized metadata table*

Description

Download a harmonized metadata table

Usage

```
getMetadata(database = NULL, load = TRUE)
```

Arguments

`database`	Name of the database to get the metadata from. Currently, there are two available options. • `cMD` : metadata for curatedMetagenomicData • `cBioPortal` : metadata for cBioPortalData
`load`	Default is `TRUE`. If it’s set to `FALSE`, the metadata table is downloaded to cache but not loaded into memory.

Value

Curated metadata table or file cache location, if ‘load = FALSE‘.

Examples

```
cmd <- getMetadata("cMD")
```

`getNarrowMetaTb` *Collapse values from multiple columns into one*

Description

Collapse values from multiple columns into one

Usage

```
getNarrowMetaTb(
  meta,
  newCol = NULL,
  targetCol = NULL,
  sep = ":",
  delim = ";",
  remove = TRUE,
  na.rm = TRUE,
  sort = TRUE
)
```

Arguments

`meta`	A data frame.
`newCol`	A character (1). Name of the new column to store collapsed values.
`targetCol`	A character vector. Names of the columns to be collapsed into one column.
`sep`	A character (1). Delimiter used to concatenate column name and its value. Default is double colons, ‘:‘.
`delim`	A character(1). Separator to use between values/columns. Default is ‘;‘.

`remove`	With the default, 'TRUE', this function will remove input columns from output data frame.
`na.rm`	With the default, 'TRUE', missing values will be removed prior to uniting each value.
`sort`	With the default, 'TRUE', the united columns will be ordered alphabetically.

Value

A data frame where target columns (`targetCol`) are collapsed into a single column. The original column name and its value are concatenated with the 'sep' input and the column:value pairs are separated by the 'delim' input. Target columns will be merged in the alphabetical order of their names.

Examples

```
wide_tb <- data.frame(fruit = c("apple", "banana", "pear", "watermelon",
                                 "grape"),
                      shape = c("round", "long", NA, "round", NA),
                      color = c("red", "yellow", NA, "green", "purple"),
                      size = c("medium", "medium", NA, "large", "small"))
getNarrowMetaTb(wide_tb,
                newCol = "feature",
                targetCol = c("color", "shape", "size"),
                sep = ":", delim = ";")
```

`getOntoInfo` *Query Ontology Lookup Service (OLS)*

Description

Extract identical or similar ontology terms across different ontologies

Usage

```
getOntoInfo(query, ontology = "", exact = FALSE, rows = 20)
```

Arguments

`query`	A character (1) containing the search query, either a term label or term id.
`ontology`	A character vector defining the ontology to be queried. Default is the empty character, to search all ontologies.
`exact`	A logical (1) defining if OLS search is restricted to exact matches. Defaults is 'FALSE'.
`rows`	An integer (1) defining the number of query returns. Default is 20L. Maximum number of values returned by the server is 1000.

Value

A tibble containing ontology term label and description

Examples

```
getOntoInfo("NCIT:C4872")
getOntoInfo("NCIT:C4872", ontology = c("EFO", "MONDO"))
getOntoInfo("Skin Infection")
getOntoInfo("Sitagliptin", ontology = c("Chebi", "apple"))

## Multiple query values
getOntoInfo("plasma,membrane")
getOntoInfo(c("plasma", "membrane"))
```

`getShortMetaTb` *Compresses expanded metadata columns to one row per sample*

Description

Compresses expanded metadata columns to one row per sample

Usage

```
getShortMetaTb(meta, idCols = NULL, targetCol = NULL, delim = "<;>")
```

Arguments

`meta`	A data frame with expanded treatment columns.
`idCols`	Optional. A character vector of columns that identify single samples, such as 'curation_id' and 'sampleId'. Defaults to standard ID columns.
`targetCol`	Optional. A character vector of columns to compress if present. Default is names of all cBioPortal treatment-related columns.
`delim`	Optional. A delimiter string. Default is '<;>'.

Value

A data frame where each sample gets a single row

Examples

```
data(mini_cmd)
lmeta <- getLongMetaTb(mini_cmd, "hla")
res <- getShortMetaTb(lmeta, targetCol = "hla")
dim(res) # 200 x 3 table

long_tb <- data.frame(ind = c("A", "A", "B", "C", "D", "D", "E"),
                      aval = c("cat", "dog", "chicken", "horse",
```

```
                                    "frog", "pig", "snake"),
                           cval = c(1, 1, NA, 3, 4, 4, 5),
                           bval = c("red", "blue", "yellow", NA, "green",
                                    NA, "brown"))
getShortMetaTb(long_tb, idCols = "ind", targetCol = c("aval", "bval"))
```

`getWideMetaTb` *Create individual columns for different attributes stored in one column*

Description

The values stored in one column should include their potential column names to use this function.

Usage

```
getWideMetaTb(meta, targetCol = NULL, sep = ":", delim = "<;>", remove = TRUE)
```

Arguments

`meta`	A data frame.
`targetCol`	A character. The column name(s) to expand if present. If multiple columns are specified here, they should be separated by the ‘sep‘ and the column name and its value should be separated by the provided ‘delim‘.
`sep`	A character (1). Delimiter used to concatenate column name and its value. Default is double colons, ‘:‘.
`delim`	A character(1). Separator used between values. Default ‘<;>‘.
`remove`	If ‘TRUE‘, remove input columns from output data frame.

Value

A data frame where the contents under ‘targetCol‘ is split into individual columns in an alphabetical order. Data type of the expanded columns is all character.

Examples

```
## Narrow-table example
narrow_tb <- data.frame(fruit = c("apple", "banana", "pear", "watermelon",
                                  "grape"),
                        feature = c("color:red;shape:round;size:medium",
                                    "color:yellow;shape:long;size:medium",
                                    "color:brown;shape:NA;size:NA",
                                    "color:green;shape:round;size:large",
                                    "color:purple;shape:NA;size:small"))
getWideMetaTb(narrow_tb, targetCol = "feature", sep = ":", delim = ";")

## Narrow-table example with missing columns
narrow_tb2 <- data.frame(fruit = c("apple", "banana", "pear",
```

```
                                 "watermelon", "grape"),
                      feature = c("color:red;shape:round;size:medium",
                                  "color:yellow;shape:long;size:medium",
                                  NA,
                                  "color:green;size:large",
                                  "color:purple;shape:NA;size:small"))
getWideMetaTb(narrow_tb2, targetCol = "feature", sep = ":", delim = ";")

## Subset of cMD metadata
data(mini_cmd3)
wtb <- getWideMetaTb(mini_cmd3, targetCol = "probing_pocket_depth")
head(wtb)
```

`merge_vectors` *Custom function to merge vectors*

Description

This function is designed for a group of, collapsible metadata attributes (e.g., 'biomarker' for curatedMetagenomicData).

Usage

```
merge_vectors(base, update, sep = ":", delim = ";")
```

Arguments

`base`	A character. A space-holder version of the key:value concatenates (e.g., ‘column1:NA;column2:NA;column3:NA‘)
`update`	A character. The target string to be compared and filled with ‘base‘ if there is missing pairs. (e.g., ‘column1:value1;column3:value3‘)
`sep`	A character string to separate the column name and value. Default is ‘:‘
`delim`	A character string to separate the column:value pairs. Default is ‘;‘

Value

A character updated the target string (‘update‘) to follow the reference string (‘base‘).

Examples

```
x <- "color:NA;shape:NA;size:NA"
y <- "color:green;size:large"
merge_vectors(x, y)
```

`mini_cbio` *A subset of cBioPortalData metadata*

Description

A subset of curated version of cBioPortal's clinical metadata.

Usage

```
mini_cbio
```

Format

A data frame with 10 samples and 9 columns ('curation_id', 'acronym', 'acronym_ontology_term_id', 'sex', 'package', 'treatment_name', 'treatment_name_ontology_term_id', 'treatment_type', 'treatment_type_ontology_term_id')

Author(s)

Sehyun Oh `<shbrief@gmail.com>`

`mini_cmd` *A subset of cMD metadata*

Description

A subset of curated version of ‘sampleMetadata‘ from the curatedMetagenomicData (cMD, ver.3.8.0) package.

Usage

```
mini_cmd
```

Format

A data frame with 200 samples and 3 columns ('curation_id', 'hla', and 'package')

Author(s)

Sehyun Oh `<shbrief@gmail.com>`

`mini_cmd2` *A subset of cMD metadata*

Description

A subset of curated version of ‘sampleMetadata‘ from the curatedMetagenomicData (cMD, ver.3.8.0) package.

Usage

```
mini_cmd2
```

Format

A data frame with 200 samples and 7 columns (’curation_id’, ’target_condition’, ’target_condition_ontology_term_id’, ’pmid’, ’disease’, ’disease_ontology_term_id’, ’package’). The two key exemplary attributes (’target_condition’ and ’disease’) selected here contain multiple values.

Author(s)

Sehyun Oh <`shbrief@gmail.com`>

`mini_cmd3` *A subset of cMD metadata*

Description

A subset of curated version of ‘sampleMetadata‘ from the curatedMetagenomicData (cMD, ver.3.8.0) package.

Usage

```
mini_cmd3
```

Format

A data frame with 200 samples and 7 columns (’curation_id’, ’pmid’, ’package’, ’target_condition’, ’feces_phenotype’, ’probing_pocket_depth’, ’target_condition_ontology_term_id’, ’feces_phenotype_ontology_term_id’, ’probing_pocket_depth_ontology_term_id’). The ’target_condition’ is multi-valued attribute, and ’feces_phenotype’ and ’probing_pocket_depth’ are composite attributes.

Author(s)

Sehyun Oh <`shbrief@gmail.com`>

`OmicsMLRepoR` *OmicsMLRepoR: A package for browsing harmonized metadata through ontology*

Description

The OmicsMLRepoR package provides functions to browse the harmonized metadata created under the OmicsMLRepo project. It supports data navigation if the metadata incorporates ontology.

Key Functions

- `getMetadata`: Download a harmonized metadata table
- `tree_filter`: Find samples including the queried terms and their descendants

Additional Information

For more detailed information, see the vignette: `vignette("Quickstart", package = "OmicsMLRepoR")`

`ontoSummarize` *Groups ontology terms by the child term of a provided "parent" they fall under*

Description

Groups ontology terms by the child term of a provided "parent" they fall under

Usage

```
ontoSummarize(parent, descendants, ontology)
```

Arguments

`parent`	Character; Term to use as parent of summarized children
`descendants`	Character vector; Terms to summarize as children of parent
`ontology`	Character; Ontology database ID

Value

A dataframe containing the descendants summarized into groups. Name of each group is the child of the parent that the descendants are grouped under. Both IDs and labels of the ontology terms are provided.

Examples

```
ontology <- "ncit"
parent <- "NCIT:C17049"
descendants <- c("NCIT:C44265", "NCIT:C77811", "NCIT:C43856",
                 "NCIT:C43672", "NCIT:C2991", "NCIT:C43860")
ontoSummarize(parent, descendants, ontology)
```

`ontoTreePlot` *Plot ontology tree*

Description

Plot ontology tree

Usage

```
ontoTreePlot(term, display = c("Term", "Text"))
```

Arguments

`term`	A character (1). Ontology term id (obo_id)
`display`	A character (1) specifying a node labeling option. Two available options are ‘Term‘ for ontology term or IRI (Internationalized Resource Identifier) and ‘Text‘ for the label or preferred name.

Value

A ontology tree plot. All the terms used in the output plot are ancestors of the queried term, so the queried term is the tip.

Examples

```
ontoTreePlot("NCIT:C2852", "Term")
```

`sample_metadata` *sample_metadata*

Description

A small data table to demonstrating the data reshaping functions in OmicsMLRepoR.

Usage

```
sample_metadata
```

Format

A data frame with 4 rows and 7 columns

Author(s)

Sehyun Oh <`shbrief@gmail.com`>

`spreadMeta` *Manipulate metadata into a spread format*

Description

This function speads gout multiple values per attribute (‘multi-valued’ attribute) or multiple features under a single generic attribute (‘composite’ attribute). Multi-valued attribute will return ’long’ table, while composite attribute will return ’wide’ table.

Usage

```
spreadMeta(meta, targetCol)
```

Arguments

`meta`	A data frame. Harmonized metadata available through the OmicsMLRepoR package. It should have the ‘package‘ column.
`targetCol`	A character. The column name(s) to expand if present. If multiple columns are specified here, they should be separated by the ‘sep‘ and the column name and its value should be separated by the provided ‘delim‘.

Value

A metadata table where the ‘targetCol‘ is updated to a long form (if it is multi-valued attribute) or spread into multiple columns (if it is composite attribute).

Examples

```
data(mini_cmd2) # multi-valued attribute
data(mini_cmd3) # composite attribute
spreadMeta(mini_cmd2, "target_condition")
spreadMeta(mini_cmd3, "probing_pocket_depth")

data(mini_cbio)
trt_cols <- grep("^treatment_", colnames(mini_cbio), value = TRUE)
spreadMeta(mini_cbio, targetCol = trt_cols)
```

`tree_filter` *Keep rows that include the queried terms and their descendants*

Description

Similar to `filter` function, while its filtering includes descendants and synonyms of the query term in addition to ontology terms and ids identical or similar to the query term across different ontologies collected through OLS search.

Usage

```
tree_filter(.data, col, query, logic = "OR", delim = NULL)
```

Arguments

`.data`	A data frame
`col`	A character (1). Column name to filter by.
`query`	A character vector containing words or ids to be used in the ontology search
`logic`	A character (1). Operator used to determine filtering method. Values allowed: "AND", "OR", "NOT". Defaults to "OR"
`delim`	A character (1) used to separate multiple values. If your '.data' input is obtained from `getMetadata` function, this input is automatically configured.

Value

Data frame filtered by provided queries along with child terms in the specified column

Examples

```
meta <- getMetadata("cMD")
tree_filter(meta, disease, c("pancreatic disease", "cancer"))
```

Index

∗ **data**
    `mini_cbio`, 10
    `mini_cmd`, 10
    `mini_cmd2`, 11
    `mini_cmd3`, 11
    `sample_metadata`, 14

`availableTerms`, 2

`filter`, *15*

`get_ontologies`, 3
`getLongMetaTb`, 3
`getMetadata`, 4, *12*
`getNarrowMetaTb`, 5
`getOntoInfo`, 6
`getShortMetaTb`, 7
`getWideMetaTb`, 8

`merge_vectors`, 9
`mini_cbio`, 10
`mini_cmd`, 10
`mini_cmd2`, 11
`mini_cmd3`, 11

`OmicsMLRepoR`, 12
`ontoSummarize`, 12
`ontoTreePlot`, 13

`sample_metadata`, 14
`spreadMeta`, 14

`tree_filter`, *12*, 15