Package 'puremoe'

April 21, 2026

Type Package

Title Pubmed Unified REtrieval for Multi-Output Exploration

Version 1.0.4

Maintainer Jason Timm <JaTimm@salud.unm.edu>

Description Access a variety of 'PubMed' data through a single, user-friendly interface, including abstracts, bibliometrics from 'iCite', pubtations from 'PubTator3', and full-text records from 'PMC'.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

Depends R (>= 3.5)

Imports rentrez, textshape, xml2, data.table, httr, pbapply, jsonlite, rappdirs

Suggests knitr, rmarkdown, DT, dplyr

VignetteBuilder knitr

RoxygenNote 7.3.3

URL https://github.com/jaytimm/puremoe, https://jaytimm.github.io/puremoe/

BugReports https://github.com/jaytimm/puremoe/issues

NeedsCompilation no

Author Jason Timm [aut, cre]

Repository CRAN

Date/Publication 2026-04-21 17:02:08 UTC

Contents

`data_mesh_frequencies` *MeSH Descriptor Frequencies Across PubMed*

Description

Baseline frequencies for MeSH descriptors computed from a local PostgreSQL mirror of PubMed (April 2026). For each descriptor, counts reflect the number of distinct PMIDs indexed with that term; proportions use the full PubMed corpus of 39,703,112 PMIDs as denominator. Descriptor UI and canonical name are joined from the NLM MeSH thesaurus. Intended as a baseline for MeSH term enrichment analyses against arbitrary PubMed subsets.

Usage

```
data_mesh_frequencies
```

Format

A data.table with 30,521 rows and 4 columns:

DescriptorUI MeSH descriptor unique identifier (e.g., `D000001`)

DescriptorName Canonical MeSH descriptor name

n_pmids Number of distinct PubMed records indexed with this descriptor

prop_total Proportion of all 39,703,112 PubMed PMIDs indexed with this descriptor

Source

Computed from `mesh_descriptor` table in a local PubMed PostgreSQL mirror; descriptor metadata from the NLM MeSH Thesaurus (April 2026).

`data_mesh_thesaurus` *Download and Combine 'MeSH' and Supplemental Thesauruses*

Description

This function downloads and combines the 'MeSH' (Medical Subject Headings) Thesaurus and a supplemental concept thesaurus. The data is sourced from specified URLs and stored locally for subsequent use. By default, the data is stored in a temporary directory. Users can opt into persistent storage by setting 'use_persistent_storage' to TRUE and optionally specifying a path.

Usage

```
data_mesh_thesaurus(
  path = NULL,
  use_persistent_storage = FALSE,
  force_install = FALSE
)
```

Arguments

`path`	A character string specifying the directory path where data should be stored. If not provided and persistent storage is requested, it defaults to a system-appropriate persistent location managed by 'rappdirs'.
`use_persistent_storage`	A logical value indicating whether to use persistent storage. If TRUE and no path is provided, data will be stored in a system-appropriate location. Defaults to FALSE, using a temporary directory.
`force_install`	A logical value indicating whether to force re-downloading of the data even if it already exists locally.

Value

A data.table containing the combined MeSH and supplemental thesaurus data.

Examples

```
if (interactive()) {
  data <- data_mesh_thesaurus()
}
```

`data_mesh_trees` *Download and Load 'MeSH' Trees Data*

Description

This function downloads and loads the 'MeSH' (Medical Subject Headings) Trees data.

Usage

```
data_mesh_trees(
  path = NULL,
  use_persistent_storage = FALSE,
  force_install = FALSE
)
```

Arguments

`path`	A character string specifying the directory path where data should be stored. If not provided and persistent storage is requested, it defaults to a system-appropriate persistent location managed by 'rappdirs'.
`use_persistent_storage`	A logical value indicating whether to use persistent storage. If TRUE and no path is provided, data will be stored in a system-appropriate location. Defaults to FALSE, using a temporary directory.
`force_install`	A logical value indicating whether to force re-downloading of the data even if it already exists locally.

Details

The data is sourced from specified URLs and stored locally for subsequent use. By default, the data is stored in a temporary directory. Users can opt into persistent storage by setting 'use_persistent_storage' to TRUE and optionally specifying a path.

Value

A data frame containing the MeSH Trees data.

Examples

```
if (interactive()) {
  data <- data_mesh_trees()
}
```

`data_pmc_list` *Download and Load 'PMC' Open Access File List*

Description

This function downloads and loads the 'PMC' (PubMed Central) Open Access file list. The file list contains mappings between PMC IDs, PMIDs, and file paths for open access articles available for download.

Usage

```
data_pmc_list(
  path = NULL,
  use_persistent_storage = FALSE,
  force_install = FALSE
)
```

Arguments

`path`	A character string specifying the directory path where data should be stored. If not provided and persistent storage is requested, it defaults to a system-appropriate persistent location managed by 'rappdirs'.
`use_persistent_storage`	A logical value indicating whether to use persistent storage. If TRUE and no path is provided, data will be stored in a system-appropriate location. Defaults to FALSE, using a temporary directory.
`force_install`	A logical value indicating whether to force re-downloading of the data even if it already exists locally.

Details

The data is sourced from NCBI's FTP server and stored locally for subsequent use. By default, the data is stored in a temporary directory. Users can opt into persistent storage by setting 'use_persistent_storage' to TRUE and optionally specifying a path.

Value

A data.table containing the PMC file list with columns: file_path, citation, pmcid, pmid, and license_code.

Examples

```
if (interactive()) {
  data <- data_pmc_list()
}
```

endpoint_info *Get Information About Available Endpoints*

Description

This function provides detailed information about the available endpoints in the package, including column descriptions, parameters, rate limits, and usage notes.

Usage

```
endpoint_info(endpoint = NULL, format = c("list", "json"))
```

Arguments

endpoint	Character string specifying which endpoint to get information about. If NULL (default), returns a list of all available endpoints.
format	Character string specifying the output format. Either "list" (default) or "json" for JSON-formatted output.

Value

If endpoint is NULL, returns a character vector of available endpoint names. If endpoint is specified, returns a list (or JSON string) with detailed information about that endpoint including description, columns, parameters, rate limits, and notes.

Examples

```
if (interactive()) {
  # List all available endpoints
  endpoint_info()

  # Get information about a specific endpoint
  endpoint_info("pubmed_abstracts")

  # Get information in JSON format
  endpoint_info("icites", format = "json")
}
```

get_records *Retrieve Data from 'NLM'/'PubMed' databases Based on PMIDs*

Description

This function retrieves different types of data (like 'PubMed' records, affiliations, 'iCites 'data, etc.) from 'PubMed' based on provided PMIDs. It supports parallel processing for efficiency.

Usage

```
get_records(
  pmids,
  endpoint = c("pubtations", "icites", "pubmed_affiliations", "pubmed_abstracts",
    "pmc_fulltext"),
  cores = 3,
  sleep = 1,
  ncbi_key = NULL
)
```

Arguments

pmids	A vector of PMIDs for which data is to be retrieved. For 'pmc_fulltext' endpoint, provide full URLs instead (e.g., from `pmid_to_pmc()$url`).
endpoint	A character vector specifying the type of data to retrieve ('pubtations', 'icites', 'pubmed_affiliations', 'pubmed_abstracts', 'pmc_fulltext').
cores	Number of cores to use for parallel processing (default is 3).
sleep	Duration (in seconds) to pause after each batch
ncbi_key	(Optional) NCBI API key for authenticated access.

Details

For the 'pmc_fulltext' endpoint, provide full URLs to PMC tar.gz files. Use `pmid_to_pmc` to convert PMIDs to PMC IDs and full URLs first.

Value

A data.table containing combined results from the specified endpoint.

Examples

```
pmids <- c("38136652")
results <- get_records(pmids, endpoint = "pubmed_abstracts", cores = 1)
```

`pmid_to_ftp` *Convert PubMed IDs (PMIDs) to PMC IDs and Full-Text URLs*

Description

This function converts PMIDs to PMC IDs, then fetches the full-text file URLs from the PMC Open Access service. It combines both steps into a single workflow.

Usage

```
pmid_to_ftp(
  pmids,
  batch_size = 200L,
  sleep = 0.5,
  verbose = FALSE,
  ncbi_key = NULL
)
```

Arguments

`pmids`	A character or numeric vector of PubMed IDs (PMIDs) to convert.
`batch_size`	An integer specifying the number of PMIDs to process per batch for ID conversion. Defaults to 200L. The NCBI API has limitations on batch sizes.
`sleep`	A numeric value specifying the number of seconds to pause between API requests for ID conversion (Step 1). Defaults to 0.5 seconds. For OA API calls (Step 2), sleep time is automatically adjusted based on rate limits: 0.11s with API key (10 req/sec), 0.34s without (3 req/sec).
`verbose`	Logical, whether to print progress messages. Defaults to FALSE.
`ncbi_key`	(Optional) NCBI API key for authenticated access.

Value

A data.table with columns:

- `pmid`: The input PubMed ID
- `pmcid`: The corresponding PMC ID
- `doi`: The corresponding DOI (NA if not available)
- `url`: The full HTTPS URL for downloading PMC full text

Results are filtered to only include rows with valid URLs (open access articles), ordered by PMID. Returns NULL with a message if the API is unavailable or returns invalid data.

Examples

```
if (interactive()) {
  # Convert PMIDs to PMC IDs and get full-text URLs
  result <- pmid_to_ftp(c("11250746", "11573492"))
}
```

`pmid_to_pmc` *Convert PubMed IDs (PMIDs) to PMC IDs*

Description

This function converts a vector of PubMed IDs (PMIDs) to their corresponding PubMed Central (PMC) IDs and DOIs using the NCBI ID Converter API.

Usage

```
pmid_to_pmc(pmids, batch_size = 200L, sleep = 0.5)
```

Arguments

`pmids`	A character or numeric vector of PubMed IDs (PMIDs) to convert.
`batch_size`	An integer specifying the number of PMIDs to process per batch. Defaults to 200L. The NCBI API has limitations on batch sizes.
`sleep`	A numeric value specifying the number of seconds to pause between API requests. Defaults to 0.5 seconds to respect API rate limits.

Value

A data.table with columns:

- `pmid`: The input PubMed ID
- `pmcid`: The corresponding PMC ID (NA if not available in PMC)
- `doi`: The corresponding DOI (NA if not available)

Results are ordered by PMID. Returns NULL with a message if the API is unavailable or returns invalid data.

Examples

```
if (interactive()) {
  # Convert a single PMID to PMC ID
  result <- pmid_to_pmc("12345678")

  # Convert multiple PMIDs
  pmids <- c("12345678", "23456789", "34567890")
  result <- pmid_to_pmc(pmids, batch_size = 10, sleep = 1)
```

```
}
```

`search_pubmed` *Search 'PubMed' Records*

Description

Performs a 'PubMed' search based on a query, optionally filtered by publication years. Returns a unique set of 'PubMed' IDs matching the query.

Usage

```
search_pubmed(
  x,
  start_year = NULL,
  end_year = NULL,
  retmax = 9999,
  use_pub_years = FALSE
)
```

Arguments

`x`	Character string, the search query.
`start_year`	Integer, the start year of publication date range (used if 'use_pub_years' is TRUE).
`end_year`	Integer, the end year of publication date range (used if 'use_pub_years' is TRUE).
`retmax`	Integer, maximum number of records to retrieve, defaults to 9999.
`use_pub_years`	Logical, whether to filter search by publication years, defaults to TRUE.

Value

Numeric vector of unique PubMed IDs.

Examples

```
ethnob1 <- search_pubmed("ethnobotany", 2010, 2012)
```

Index