Package 'medrxivr'

October 4, 2024

Title Access and Search MedRxiv and BioRxiv Preprint Data

Version 0.1.1

Description An increasingly important source of health-related bibliographic content are preprints - preliminary versions of research articles that have yet to undergo peer review. The two preprint repositories most relevant to health-related sciences are medRxiv <https://www.medrxiv.org/> and bioRxiv <https://www.biorxiv.org/>, both of which are operated by the Cold Spring Harbor Laboratory. 'medrxivr' provides programmatic access to the 'Cold Spring Harbour Laboratory (CSHL)' API <https://api.biorxiv.org/>, allowing users to easily download medRxiv and bioRxiv preprint metadata (e.g. title, abstract, publication date, author list, etc) into R. 'medrxivr' also provides functions to search the downloaded preprint records using regular expressions and Boolean logic, as well as helper functions that allow users to export their search results to a .BIB file for easy import to a reference manager and to download the full-text PDFs of preprints matching their search criteria.

License GPL-2

Encoding UTF-8

Language en-US

URL https://github.com/ropensci/medrxivr

BugReports https://github.com/ropensci/medrxivr/issues

Imports methods, dplyr, curl, jsonlite, httr, stringr, rlang, bib2df, tibble, progress, lubridate, purrr, data.table

Suggests testthat (>= 2.1.0), knitr, rmarkdown, covr, kableExtra, spelling

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Author Yaoxiang Li [aut, cre] (<https://orcid.org/0000-0001-9200-1016>),
Luke McGuinness [aut],
Lena Schmidt [aut],
Tuija Sonkkila [rev],
Najko Jahn [rev]

Maintainer Yaoxiang Li <liyaoxiang@outlook.com>

Repository CRAN

Date/Publication 2024-10-04 20:10:08 UTC

Contents

`mx_api_content` *Access medRxiv/bioRxiv data via the Cold Spring Harbour Laboratory API*

Description

Provides programmatic access to all preprints available through the Cold Spring Harbour Laboratory API, which serves both the medRxiv and bioRxiv preprint repositories.

Usage

```
mx_api_content(
  from_date = "2013-01-01",
  to_date = as.character(Sys.Date()),
  clean = TRUE,
  server = "medrxiv",
  include_info = FALSE
)
```

Arguments

`from_date`	Earliest date of interest, written as "YYYY-MM-DD". Defaults to 1st Jan 2013 ("2013-01-01"), ~6 months prior to earliest preprint registration date.
`to_date`	Latest date of interest, written as "YYYY-MM-DD". Defaults to current date.

clean	Logical, defaulting to TRUE, indicating whether to clean the data returned by the API. If TRUE, variables containing absolute paths to the preprints web-page ("link_page") and PDF ("link_pdf") are generated from the "server", "DOI", and "version" variables returned by the API. The "title", "abstract" and "authors" variables are converted to title case. Finally, the "type" and "server" variables are dropped.
server	Specify the server you wish to use: "medrxiv" (default) or "biorxiv"
include_info	Logical, indicating whether to include variables containing information returned by the API (e.g. API status, cursor number, total count of papers, etc). Default is FALSE.

Value

Dataframe with 1 record per row

See Also

Other data-source: `mx_api_doi()`, `mx_snapshot()`

Examples

```
if (interactive()) {
  mx_data <- mx_api_content(
    from_date = "2020-01-01",
    to_date = "2020-01-07"
  )
}
```

---

mx_api_doi *Access data on a single medRxiv/bioRxiv record via the Cold Spring Harbour Laboratory API*

Description

Provides programmatic access to data on a single preprint identified by a unique Digital Object Identifier (DOI).

Usage

```
mx_api_doi(doi, server = "medrxiv", clean = TRUE)
```

Arguments

doi	Digital object identifier of the preprint you wish to retrieve data on.
server	Specify the server you wish to use: "medrxiv" (default) or "biorxiv"

clean — Logical, defaulting to TRUE, indicating whether to clean the data returned by the API. If TRUE, variables containing absolute paths to the preprints web-page ("link_page") and PDF ("link_pdf") are generated from the "server", "DOI", and "version" variables returned by the API. The "title", "abstract" and "authors" variables are converted to title case. Finally, the "type" and "server" variables are dropped.

Value

Dataframe containing details on the preprint identified by the DOI.

See Also

Other data-source: `mx_api_content()`, `mx_snapshot()`

Examples

```
if (interactive()) {
  mx_data <- mx_api_doi("10.1101/2020.02.25.20021568")
}
```

---

mx_caps *Search term wrapper that allows for different capitalization of term*

Description

Inspired by the varying capitalization of "NCOV" during the corona virus pandemic (e.g. ncov, nCoV, NCOV, nCOV), this function allows for all possible configurations of lower- and upper-case letters in your search term.

Usage

```
mx_caps(x)
```

Arguments

x — Search term to be formatted

Value

The input string is return, but with each non-space character repeated in lower- and upper-case, and enclosed in square brackets. For example, mx_caps("ncov") returns "[Nn][Cc][Oo][Vv]"

See Also

Other helper: `mx_crosscheck()`, `mx_download()`, `mx_export()`

Examples

```
query <- c("coronavirus", mx_caps("ncov"))

mx_search(mx_snapshot("6c4056d2cccd6031d92ee4269b1785c6ec4d555b"), query)
```

`mx_crosscheck` *Check how up-to-date the maintained medRxiv snapshot is*

Description

Provides information on how up-to-date the maintained medRxiv snapshot provided by 'mx_snapshot()' is by checking whether there have been any records added to, or updated in, the medRxiv repository since the last snapshot was taken.

Usage

```
mx_crosscheck()
```

See Also

Other helper: `mx_caps()`, `mx_download()`, `mx_export()`

Examples

```
mx_crosscheck()
```

`mx_download` *Download PDF's of preprints returned by a search*

Description

Download PDF's of all the papers in your search results

Usage

```
mx_download(
  mx_results,
  directory,
  create = TRUE,
  name = c("ID", "DOI"),
  print_update = 10
)
```

Arguments

mx_results	Vector containing the links to the medRxiv PDFs
directory	The location you want to download the PDF's to
create	TRUE or FALSE. If TRUE, creates the directory if it doesn't exist
name	How to name the downloaded PDF. By default, both the ID number of the record and the DOI are used.
print_update	How frequently to print an update

See Also

Other helper: mx_caps(), mx_crosscheck(), mx_export()

Examples

```
mx_results <- mx_search(mx_snapshot(), query = "10.1101/2020.02.25.20021568")
mx_download(mx_results, directory = tempdir())
```

---

mx_export *Export references for preprints returning by a search to a .bib file*

Description

Export references for preprints returning by a search to a .bib file

Usage

```
mx_export(data, file = "medrxiv_export.bib")
```

Arguments

data	Dataframe returned by mx_search() or mx_api_*() functions
file	File location to save to. Must have the .bib file extension

Value

Exports a formatted .BIB file, for import into a reference manager

See Also

Other helper: mx_caps(), mx_crosscheck(), mx_download()

Examples

```
mx_results <- mx_search(mx_snapshot(), query = "brain")
mx_export(mx_results, tempfile(fileext = ".bib"))
```

mx_reporter *Search and print output for individual search items*

Description

Search and print output for individual search items

Usage

```
mx_reporter(mx_data, num_results, query, fields, deduplicate, NOT)
```

Arguments

mx_data	The mx_dataset filtered for the date limits
num_results	The number of results returned by the overall search
query	Character string, vector or list
fields	Fields of the database to search - default is Title, Abstract, Authors, Category, and DOI.
deduplicate	Logical. Only return the most recent version of a record. Default is TRUE.
NOT	Vector of regular expressions to exclude from the search. Default is "".

See Also

Other main: `mx_search()`, `print_full_results()`, `run_search()`

mx_search *Search preprint data*

Description

Search preprint data

Usage

```
mx_search(
  data = NULL,
  query = NULL,
  fields = c("title", "abstract", "authors", "category", "doi"),
  from_date = NULL,
  to_date = NULL,
  auto_caps = FALSE,
  NOT = "",
  deduplicate = TRUE,
  report = FALSE
)
```

Arguments

data	The preprint dataset that is to be searched, created either using mx_api_content() or mx_snapshot()
query	Character string, vector or list
fields	Fields of the database to search - default is Title, Abstract, Authors, Category, and DOI.
from_date	Defines earliest date of interest. Written in the format "YYYY-MM-DD". Note, records published on the date specified will also be returned.
to_date	Defines latest date of interest. Written in the format "YYYY-MM-DD". Note, records published on the date specified will also be returned.
auto_caps	As the search is case sensitive, this logical specifies whether the search should automatically allow for differing capitalisation of search terms. For example, when TRUE, a search for "dementia" would find both "dementia" but also "Dementia". Note, that if your term is multi-word (e.g. "systematic review"), only the first word is automatically capitalised (e.g your search will find both "systematic review" and "Systematic review" but won't find "Systematic Review". Note that this option will format terms in the query and NOT arguments (if applicable).
NOT	Vector of regular expressions to exclude from the search. Default is "".
deduplicate	Logical. Only return the most recent version of a record. Default is TRUE.
report	Logical. Run mx_reporter. Default is FALSE.

See Also

Other main: `mx_reporter()`, `print_full_results()`, `run_search()`

Examples

```
# Using the daily snapshot
mx_results <- mx_search(data = mx_snapshot(), query = "dementia")
```

---

mx_snapshot *Access a static snapshot of the medRxiv repository*

Description

[Available for medRxiv only] Rather than downloading a copy of the medRxiv database from the API, which can become unavailable at peak usage times, this allows users to import a maintained static snapshot of the medRxiv repository.

Usage

```
mx_snapshot(commit = "master")
```

Arguments

commit	Commit hash for the snapshot, taken from https://github.com/mcguinlu/medrxivr-data. Allows for reproducible searching by specifying the exact snapshot used to perform the searches. Defaults to "master", which will return the most recent snapshot.

Value

Formatted dataframe

See Also

Other data-source: mx_api_content(), mx_api_doi()

Examples

```
mx_data <- mx_snapshot()
```

---

print_full_results *Search for terms in the dataset*

Description

Search for terms in the dataset

Usage

```
print_full_results(num_results, deduplicate)
```

Arguments

num_results	number of searched terms returned
deduplicate	Logical. Only return the most recent version of a record. Default is TRUE.

See Also

Other main: mx_reporter(), mx_search(), run_search()

`run_search` *Search for terms in the dataset*

Description

Search for terms in the dataset

Usage

```
run_search(mx_data, query, fields, deduplicate, NOT = "")
```

Arguments

`mx_data`	The mx_dataset filtered for the date limits
`query`	Character string, vector or list
`fields`	Fields of the database to search - default is Title, Abstract, Authors, Category, and DOI.
`deduplicate`	Logical. Only return the most recent version of a record. Default is TRUE.
`NOT`	Vector of regular expressions to exclude from the search. Default is NULL.

See Also

Other main: `mx_reporter()`, `mx_search()`, `print_full_results()`

Index