Package 'LitFetchR'

November 12, 2026

Type Package

Title Automatically Fetching References Metadata from Literature Databases

Version 1.1.0

Maintainer Thomas Dumond <thomas.dumond@adelaide.edu.au>

Description Provides functions to automatically retrieve and deduplicate reference metadata based on saved search strings. Access to Web of Science and Scopus requires personal API keys, while PubMed can be queried without one. The optional deduplication functionality requires the package 'ASySD' available from <https://github.com/camaradesuk/ASySD>.

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr, httr, jsonlite, openxlsx, purrr, xml2

Suggests ASySD, cronR, knitr, rmarkdown, testthat (>= 3.0.0)

RoxygenNote 7.3.3

Config/testthat/edition 3

URL https://github.com/thomasdumond/LitFetchR, https://thomasdumond.github.io/LitFetchR/

BugReports https://github.com/thomasdumond/LitFetchR/issues

VignetteBuilder knitr

NeedsCompilation no

Author Thomas Dumond [cre, aut, cph] (ORCID: <https://orcid.org/0000-0002-9427-8649>), Charles Caraguel [ctb] (ORCID: <https://orcid.org/0000-0003-0019-4813>), Torben Nielsen [ctb] (ORCID: <https://orcid.org/0000-0002-8941-7376>)

Repository CRAN

Date/Publication 2026-11-12 16:00:09 UTC

Contents

auto_LitFetchR_setup 2
create_save_search 4
dedup_refs 5
manual_fetch 6
remove_scheduled_task 7
save_api_keys 8

Index **10**

`auto_LitFetchR_setup` *Automating the retrieval of references based on a saved search string(s).*

Description

Creates a read-only Rscript and a task to run the code automatically at a specified frequency and time, to retrieve references corresponding to the saved search string(s) on up to three platforms (e.g. Web of Science, Scopus and PubMed).

Usage

```
auto_LitFetchR_setup(
  task_id = "task_id",
  when = "DAILY",
  time = "08:00",
  wos = FALSE,
  scp = FALSE,
  pmd = FALSE,
  directory,
  dedup = FALSE,
  open_file = FALSE,
  dry_run = FALSE,
  day = NULL,
  output_format = c("csv", "ris")
)
```

Arguments

`task_id`	Name of the automated reference retrieval task (e.g. one keyword describing your review). Spaces are replaced by "-". A task with the same name must not exist already: use `remove_scheduled_task()` first to replace it.
`when`	Frequency of the automated reference retrieval task (DAILY, WEEKLY or MONTHLY).
`time`	Time of the automated reference retrieval task (must be HH:MM 24-hour clock format).
`wos`	Runs the search on Web of Science (TRUE or FALSE).

scp	Runs the search on Scopus (TRUE or FALSE).
pmd	Runs the search on PubMed (TRUE or FALSE).
directory	Choose the directory in which the search string is saved (Project's directory). That is also where the references metadata will be saved.
dedup	Deduplicates the retrieved references (TRUE or FALSE).
open_file	Automatically opens the output file after reference retrieval.
dry_run	Simulation run option.
day	Day on which a WEEKLY or MONTHLY task runs. For WEEKLY, one of MON, TUE, WED, THU, FRI, SAT or SUN. For MONTHLY, a day of the month between 1 and 28. If NULL (default), the weekday (WEEKLY) or day of the month (MONTHLY) of today's date is used. Ignored for DAILY.
output_format	Format of the file of references written at each run: "csv" (default) or "ris" (see manual_fetch()). The choice is fixed in the script of the task. To change it, remove the task with remove_scheduled_task() and create it again.

Details

The function reports "Task scheduled!" only after checking that the scheduler accepted the task and that the task exists. On Windows the registered task is read back from Task Scheduler and compared with the requested schedule. A short summary (recurrence, next run time and status) is then printed. If the scheduler returns an error, the function stops and shows the scheduler's own message. If the task cannot be confirmed, a warning is given instead of the success message. Check the summary: it is the only evidence that automatic retrieval will run.

The scheduled task runs only while the computer is on and you are logged in. Each run is recorded in run_log.csv (see manual_fetch()) with the name of the task in the column run_mode. When a run fails, or finishes with problems on one platform, it ends with a non-zero exit status, which the Windows Task Scheduler shows as the Last Run Result of the task.

Value

NULL (invisibly). Called for its side effects: writes an R script and schedules a task (Windows Task Scheduler or cron) to run the script automatically. The function stops with the scheduler's error message if the task cannot be created, and gives a warning if the new task cannot be confirmed.

Examples

```
# This is a "dry run" example.
# No task will actually be scheduled,
# it only shows how the function should react.
auto_LitFetchR_setup(task_id = "fish_vibrio",
                     when = "WEEKLY",
                     time = "14:00",
                     day = "MON",
                     wos = TRUE,
                     scp = TRUE,
                     pmd = TRUE,
                     directory,
                     dedup = FALSE,
```

```
open_file = FALSE,
dry_run = TRUE
)
```

`create_save_search` *Creates and saves search string(s).*

Description

An interactive function that ask the user to enter a search string and provide the number of results from 3 platforms: Web of Science, Scopus and PubMed. You can then save one or more search strings to retrieve the references later.

Usage

```
create_save_search(
  wos = FALSE,
  scp = FALSE,
  pmd = FALSE,
  directory,
  dry_run = FALSE
)
```

Arguments

`wos`	Runs the search on Web of Science (TRUE or FALSE).
`scp`	Runs the search on Scopus (TRUE or FALSE).
`pmd`	Runs the search on PubMed (TRUE or FALSE).
`directory`	Choose the directory in which the search string and the search history will be saved.
`dry_run`	Simulation run option.

Details

Type `summary` to display the searches run so far and save one of them (the saved search strings are stored in "search_list.txt", which is required by the retrieval functions). Type `exit` to leave the tool. If searches were run but none was saved during the session, `exit` first offers to save one.

Value

`NULL` (invisibly). Called for its side effects: interactive querying and writing search history files.

Examples

```
# This is a "dry run" example.
# No search will be created and no database will be accessed.
# It only shows how the function should react.
create_save_search(wos = TRUE,
                   scp = TRUE,
                   pmd = TRUE,
                   directory,
                   dry_run = TRUE)
```

`dedup_refs` *Deduplicates the references from up to three dataframes.*

Description

Deduplicates the references from up to three dataframes.

Usage

```
dedup_refs(
  df1 = NULL,
  df2 = NULL,
  df3 = NULL,
  directory,
  open_file = FALSE,
  dry_run = FALSE,
  output_format = c("csv", "ris")
)
```

Arguments

`df1`	Dataframe 1 (can be NULL)
`df2`	Dataframe 2 (can be NULL)
`df3`	Dataframe 3 (can be NULL)
`directory`	Choose the directory in which the references deduplication history will be saved.
`open_file`	Automatically opens the output file after reference retrieval.
`dry_run`	Simulation run option.
`output_format`	Format of the file of deduplicated references: `"csv"` (default) or `"ris"` (see `manual_fetch()`).

Value

`NULL` (invisibly). Called for its side effects: writes a CSV or RIS file of deduplicated citations and an Excel workbook recording the deduplication history.

Examples

```
# This is a "dry run" example.
# No deduplication will happen.
# It only shows how the function should react.
dedup_refs(df1 = df_vibrio_wos,
           df2 = df_vibrio_scp,
           df3 = df_vibrio_pmd,
           directory = tempdir(),
           open_file = FALSE,
           dry_run = TRUE
           )
```

`manual_fetch` *Manual literature retrieval.*

Description

Retrieves references corresponding to the saved search string(s) on up to three platforms (e.g. Web of Science, Scopus and PubMed).

Usage

```
manual_fetch(
  wos = FALSE,
  scp = FALSE,
  pmd = FALSE,
  directory,
  dedup = FALSE,
  open_file = FALSE,
  dry_run = FALSE,
  output_format = c("csv", "ris")
)
```

Arguments

`wos`	Runs the search on Web of Science (TRUE or FALSE).
`scp`	Runs the search on Scopus (TRUE or FALSE).
`pmd`	Runs the search on PubMed (TRUE or FALSE).
`directory`	Choose the directory in which the search string is saved (Project's directory). That is also where the references metadata will be saved.
`dedup`	Deduplicates the retrieved references (TRUE or FALSE).
`open_file`	Automatically opens the output file after reference retrieval.
`dry_run`	Simulation run option.

`output_format` Format of the file of references: `"csv"` (default) or `"ris"`. RIS is read by most reference managers and screening tools, for example Covidence, which does not accept CSV files. A RIS file has one `AU` line per author and always describes journal articles. With `dedup = TRUE` the deduplicated references are written in the chosen format.

Details

Every retrieval is recorded in the file `run_log.csv` of `directory` (one row per saved search and platform, or one row with platform "all" if the run could not start, for example because `search_list.txt` is invalid). It records whether the run was started by hand or by a scheduled task (`run_mode`). It stores the exact query submitted to each platform, the date, the package and R versions, the number of results and new records, the status of the request and any error message. For PubMed, it also stores the query translation reported by PubMed. For Web of Science, it also stores the number of records retrieved from each source database.

The history of the references already retrieved (`history_id.xlsx`) is only updated once the output file has been written. If a platform fails (for example an expired API key), the other platforms are still saved, a warning names the failed platform(s), and the failed platform is not added to the history, so its references are retrieved again by the next run. If all the selected platforms fail, an error is raised. When there is no new record, no output file is written.

Value

`NULL` (invisibly). Called for its side effects: Create a CSV or RIS file with the references metadata, a history file of the references retrieved, a history file of the deduplication (if the option is selected) and a run log (`run_log.csv`).

Examples

```
# This is a "dry run" example.
# No references will actually be scheduled, it only shows how the function should react.
manual_fetch(wos = TRUE,
             scp = TRUE,
             pmd = TRUE,
             directory,
             dedup = TRUE,
             open_file = FALSE,
             dry_run = TRUE
             )
```

---

remove_scheduled_task *Removes a scheduled task.*

Description

Removes a scheduled task using the "task_id" from Task Scheduler (Windows) or Cron (Mac/Linux).

Usage

```
remove_scheduled_task(task_id, dry_run = FALSE)
```

Arguments

task_id	Name/ID of the scheduled task (Windows Task Scheduler or Cron).
dry_run	Simulation run option.

Details

The function checks whether the task exists first, removes it, then checks that it is gone. It reports which of these happened. If the task does not exist, nothing is removed and a message says so. If the scheduler returns an error, or the task is still there afterwards, the function stops and shows the scheduler's own message.

Value

NULL (invisibly). Called for its side effects: removes a scheduled task saved using the function 'auto_LitFetchR_setup'.

Examples

```
# This is a "dry run" example.
# No task will actually be removed, it only shows how the function should react.
remove_scheduled_task("fish_vibrio",
                      dry_run = TRUE
                      )
```

---

save_api_keys *Saves Web of Science, Scopus, and/or NCBI/PubMed API keys in .Renviron.*

Description

You can set wos_api_key, scp_api_key, scp_insttoken, ncbi_api_key, or any combination. Remember to restart the R session after saving your API keys.

Usage

```
save_api_keys(
  wos_api_key = NULL,
  scp_api_key = NULL,
  scp_insttoken = NULL,
  ncbi_api_key = NULL,
  dry_run = FALSE
)
```

Arguments

wos_api_key	The API key value for Web of Science (use quotation marks).
scp_api_key	The API key value for Scopus (use quotation marks).
scp_insttoken	The institutional token for Scopus (use quotation marks). Required by some institutional Scopus subscriptions alongside the API key.
ncbi_api_key	The API key value for NCBI/PubMed (use quotation marks). Optional; when set, PubMed requests run at a higher rate limit.
dry_run	Simulation run option.

Value

Logical. TRUE if at least one value was written, FALSE if left unchanged.

Examples

```
save_api_keys(wos_api_key = "abcd01234",
              scp_api_key = "efgh5678",
              dry_run = TRUE
              )
```

Index

auto_LitFetchR_setup, 2

create_save_search, 4

dedup_refs, 5

manual_fetch, 6
manual_fetch(), *3*, *5*

remove_scheduled_task, 7
remove_scheduled_task(), *2*, *3*

save_api_keys, 8