Retrieve clinical trial information

Ralf Herold

2020-10-11

Attach package `ctrdata`

```
library(ctrdata)
```

Remember to respect the registers' terms and conditions (see `ctrOpenSearchPagesInBrowser(copyright = TRUE)`). Please cite this package in any publication as follows: Ralf Herold (2020). ctrdata: Retrieve and Analyze Clinical Trials in Public Registers. R package version 1.4, https://github.com/rfhb/ctrdata

Open register's advanced search page in browser

These functions open the browser, where the user can start searching for trials of interest.

```
# Please review and respect register copyrights:
ctrOpenSearchPagesInBrowser(
  copyright = TRUE
)
# Open browser with example search:
ctrOpenSearchPagesInBrowser(
  input = "cancer&age=under-18",
  register = "EUCTR"
)
```

Adjust search parameters and execute search in browser

Refine the search until the trials of interest are listed in the browser. The total number of trials that can be retrieved with package `ctrdata` is intentionally limited to queries with at most 10000 result records.

Copy address from browser address bar to clipboard

Use functions or keyboard shortcuts according to the operating system.

Get address from clipboard

The next steps are executed in the R environment:

```
q <- ctrGetQueryUrlFromBrowser()
# Found search query from EUCTR.

q
#                                                    query-term  query-register
# 1 query=cancer&age=under-18&status=completed&phase=phase-one            EUCTR

# To check, a browser with this query
# is opened with this command
ctrOpenSearchPagesInBrowser(
  input = q
)
```

Retrieve protocol-related information, transform, save to database, check

```
# Connect to a database and chose a table / collection
db <- nodbi::src_sqlite(
  dbname = "sqlite_file.sql",
  collection = "test"
)

# Count number of trial records
ctrLoadQueryIntoDb(
  queryterm = q,
  only.count = TRUE,
  con = db
)$n
# [1] 60

# Use search q (defined in previous step):
ctrLoadQueryIntoDb(
  queryterm = q,
  con = db
)

# Show which queries have been downloaded into database
dbQueryHistory(con = db)
#       query-timestamp query-register query-records
# 1 2020-10-11 20:23:11          EUCTR           173
#                                                    query-term
# 1 query=cancer&age=under-18&status=completed&phase=phase-one
```

With file-base SQLite, it takes about 5 minutes for 1000 records.

Speed is higher when using MongoDB (or memory-based SQLite).

Repeat and update a previous query

```
ctrLoadQueryIntoDb(
  querytoupdate = "last",
```

```
  con = db
)
```

Instead of "last", an integer number can be specified for `querytoupdate` that corresponds to the number when using `dbQueryHistory()`.

Depending on the register, an update (differential update) is possible or the original query is executed fully again.

Retrieve results

For EUCTR, result-related trial information has to be requested to be retrieved, because it will take longer to download and store. For CTGOV, any results are always included in the retrieval.

```
ctrLoadQueryIntoDb(
  queryterm = q,
  euctrresults = TRUE,
  con = db
)
```

The download or presence of results is not recorded in `dbQueryHistory()` because the availability of results increased, also for previously retrieved trials.

Add trial information from other register

The same database and table / collection can be used to store (and analyse) trial information from different registers. Example:

```
ctrLoadQueryIntoDb(
  queryterm = "https://clinicaltrials.gov/ct2/results?cond=neuroblastoma&recrs=e&age=0&intr=Drug",
  con = db
)
# [...]
#       query-timestamp query-register query-records
# 1 2020-10-11 21:27:58          EUCTR           230
# 2 2020-10-11 21:39:37          CTGOV           190
#                                                 query-term
# 1 query=cancer&age=under-18&status=completed&phase=phase-one
# 2                 cond=neuroblastoma&recrs=e&age=0&intr=Drug
```

Add personal annotations when retrieving trial information

(Only) When downloading trial information, the user can specify an annotation to all records that are downloaded. By default, annotations are accumulated if trial records are loaded again or updated; alternatively, annotations can be replaced.

Annotations are useful for analyses, for example to specially identify subsets of records in the database.

```
ctrLoadQueryIntoDb(
  queryterm = "https://clinicaltrials.gov/ct2/results?cond=neuroblastoma&recrs=e&age=0&intr=Drug&cntry=
  annotation.text = "site_DE ",
  annotation.mode = "append",
  con = db
)
# [...]
#       query-timestamp query-register query-records
# 1 2020-10-11 21:27:58          EUCTR           230
# 2 2020-10-11 21:39:37          CTGOV           190
#                                                 query-term
# 1 query=cancer&age=under-18&status=completed&phase=phase-one
# 2                 cond=neuroblastoma&recrs=e&age=0&intr=Drug
```

Find synonyms of active substance names

Not all registers automatically expand search terms to include alternative terms, such as codes and other names of active substances. To obtain a character vector of synonyms for any active substance name, use:

```
ctrFindActiveSubstanceSynonyms(
  activesubstance = "imatinib"
)
```

These names can then be used in queries in any register.

Using a mongo database

This example works with a free service here. Note that the user name and password need to be encoded. The format of the connection string is documented here.

```
# Specify base uri for remote mongodb server as part of the encoded connection string
db <- nodbi::src_mongo(
  # This provides a read-only access
  url = "mongodb+srv://DWbJ7Wh:bdTHh5cS@cluster0-b9wpw.mongodb.net",
  db = "dbperm",
  collection = "dbperm")

# Database connection can just be used to retrieve data
result <- dbGetFieldsIntoDf(
  fields = c("a2_eudract_number",
             "overall_status",
             "record_last_import",
             "primary_completion_date",
             "x6_date_on_which_this_record_was_first_entered_in_the_eudract_database",
             "e71_human_pharmacology_phase_i"),
  con = db)
```