Retrieve clinical trial information

Ralf Herold

2025-03-02

Get started

Attach package `ctrdata`

```
library(ctrdata)
citation("ctrdata")
```

Remember to respect the registers' terms and conditions (see `ctrOpenSearchPagesInBrowser(copyright = TRUE)`). Please cite this package in any publication as follows: Ralf Herold (2025). ctrdata: Retrieve and Analyze Clinical Trials in Public Registers. R package version 1.21.0. https://cran.r-project.org/package=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(
  url = "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 10,000 result records.

Copy address from browser address bar to clipboard

Use functions or keyboard shortcuts according to the operating system. See here for our automation to copy the URLs of a user's queries in any of the supported clinical trial registers.

Get address from clipboard

The next steps are executed in the R environment:

```
q <- ctrGetQueryUrl()
# * Using clipboard content as register query URL: https://www.clinicaltrialsregister.eu/
# ctr-search/search?query=cancer&age=under-18&resultsstatus=trials-with-results
# * Found search query from EUCTR: query=cancer&age=under-18&resultsstatus=trials-with-results

q
#                                                    query-term  query-register
# 1 query=cancer&age=under-18&resultsstatus=trials-with-results           EUCTR

# To check, this opens a browser with the query
ctrOpenSearchPagesInBrowser(url = q)
```

Retrieve trial information, transform, save to database and check queries

Note that in addition to protocol- and results-related information, also all trial documents that made publicly available by the registers, including any protocols, consent forms and results reports, can be downloaded by `ctrdata`, by specifying parameter `documents.path`, see `help(ctrLoadQueryIntoDb)`.

```
# Count number of trial records
ctrLoadQueryIntoDb(
  queryterm = q,
  only.count = TRUE
)$n
# * Checking trials in EUCTR...
# Retrieved overview, multiple records of 376 trial(s) from 19 page(s) to be
# downloaded (estimate: 50 MB)
# [1] 376

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

# Retrieve records, download into database
ctrLoadQueryIntoDb(
  queryterm = q,
  con = db
)

# Show which queries have been downloaded into database
dbQueryHistory(con = db)
#       query-timestamp query-register query-records
# 1 2025-03-02 13:05:08          EUCTR          1470
#                                                    query-term
# 1 query=cancer&age=under-18&resultsstatus=trials-with-results
```

With any database, this takes about 80 seconds for 1470 records of 376 trials (~55 ms/record), with most of the time needed for internet-retrieval which is slow from this register.

Repeat and update a previous query

Previously executed queries can be repeated by specifying "last" or an integer number for parameter `querytoupdate`, where the number corresponds to the row number of the query shown with `dbQueryHistory()`. Where possible, the query to update first checks for new records in the register. Depending on the register and time since running the query last, an update (differential update) is possible or the original query is executed fully again.

```
# Show all queries
dbQueryHistory(con = db)

# Repeat last query
ctrLoadQueryIntoDb(
  querytoupdate = "last",
  only.count = TRUE,
  con = db
)
# First result page empty - no (new) trials found?
# Updated history ("meta-info" in "test")
```

Results-related trial information

For CTGOV and CTGOV2, any results are always included in the retrieval. Only for EUCTR, result-related trial information has to be requested to be retrieved, because it will take longer to download and store. No results in structured electronic format are foreseeably available from ISRCTN and CTIS, thus `ctrdata` cannot load them, see `help(ctrLoadQueryIntoDb)`. The download or presence of results is not recorded in `dbQueryHistory()` because the availability of results increases over time. The following takes about 90 seconds for about 1470 records (~ 65 ms/record) of 376 trials.

```
Sys.time(); ctrLoadQueryIntoDb(
  querytoupdate = "last",
  forcetoupdate = TRUE,
  euctrresults = TRUE,
  con = db
); Sys.time();
# * Checking trials in EUCTR...
# Retrieved overview, multiple records of 376 trial(s) from 19 page(s) to be downloaded (estimate: 50 M
# (1/3) Downloading trials...
# Note: register server cannot compress data, transfer takes longer (estimate: 500 s)
# (2/3) Converting to NDJSON (estimate: 8 s)...
# (3/3) Importing records into database...
# = Imported or updated 1470 records on 376 trial(s)
# * Checking results if available from EUCTR for 376 trials:
# (1/4) Downloading results...
# Download status: 376 done; 0 in progress. Total size: 45.77 Mb (100%)... done!
# - extracting results (. = data, F = file[s] and data, x = none):
# . . . . . F . F . . . F . F . . F . F . . . . . . . . . . . . . . F . . . .
# . . . . . . . . . . . . . . . . . . . F . . . . . . . . F . . . . . . . . . F . .
# . . F . F . . . . . F . . . . . F . . . F . . . . . . . . . F F . F . . . . . F
# . . . . . F F . F F . . . . F F . . . F F . . . . . . . . . . . . . F . . . . . .
# F . F . . . . . . F . . . . . . F F . . . F . . F . . . . . F F F . . . . . .
# . . F . F . . . . . F . . F . . . . F F F . . F . . . . F F . F . . . . . . . .
# . . . . . F . F . . . . . . . . . . . F F . . . . . . . . F . F . . . . . . . . F
```

```
# . . F F . . . . F F . . F . . . . F F . . . F . . . . . . . . . . F . . . . . .
# . . . . . . . F . F . . . . . . . F . F . . . F . . . . . . . . . . . F . F . F .
# . . F F F . . . . . . . . . . . . F
# (2/4) Converting to NDJSON (estimate: 40 s)...
# (3/4) Importing results into database (may take some time)...
# (4/4) Results history: not retrieved (euctrresultshistory = FALSE)
# = Imported or updated results for 376 trials
# Updated history ("meta-info" in "test")
# $n
# [1] 1470
```

Add trial information from several registers

The same collection can be used to store (and analyse) trial information from different registers, thus can include different and complementary sets of trials. The registers currently supported include CTIS, EUCTR, CTGOV, CTGOV2 and ISRCTN. This can be achieved by loading queries that the user defines specifically or that function `ctrGenerateQueries()` provides, as follows:

```
# Loading specific query into same collection
ctrLoadQueryIntoDb(
  queryterm = "cond=neuroblastoma&aggFilters=phase:2,ages:child,status:com",
  register = "CTGOV2",
  con = db
)

# Use same query details to obtain queries
queries <- ctrGenerateQueries(
  condition = "neuroblastoma",
  recruitment = "completed",
  phase = "phase 2",
  population = "P"
)

# Open queries in registers' web interfaces
sapply(queries, ctrOpenSearchPagesInBrowser)

# Load all queries into database collection
result <- lapply(queries, ctrLoadQueryIntoDb, con = db)

# Show results of loading
sapply(result, "[[", "n")
# EUCTR CTGOV2 ISRCTN   CTIS
#   180    111      0      1

# Overview of queries
dbQueryHistory(con = db)
#       query-timestamp query-register query-records
# 1 2025-03-02 13:05:08          EUCTR          1470
# 2 2025-03-02 13:26:49          EUCTR             0
# 3 2025-03-02 13:31:53          EUCTR          1470
# 4 2025-03-02 13:49:09         CTGOV2           111
# 5 2025-03-02 13:56:01          EUCTR           180
```

```
# 6 2025-03-02 13:56:02         CTGOV2           111
# 7 2025-03-02 13:56:03           CTIS             1
#                                                                                                    que
# 1                                                  query=cancer&age=under-18&resultsstatus=trials-with-
# 2                                                  query=cancer&age=under-18&resultsstatus=trials-with-
# 3                                                  query=cancer&age=under-18&resultsstatus=trials-with-
# 4                                                  cond=neuroblastoma&aggFilters=phase:2,ages:child,sta
# 5                                query=neuroblastoma&phase=phase-two&age=children&age=under-18&status=co
# 6                                                  cond=neuroblastoma&aggFilters=phase:2,ages:child,sta
# 7 searchCriteria={"medicalCondition":"neuroblastoma","trialPhaseCode":[4],"ageGroupCode":[2],"status"
```

Add personal annotations

When loading trial information, the user can specify an annotation string to each of the records that are loaded when calling `ctrLoadQueryIntoDb()`. By default, new annotations are appended to any existing annotation of the trial record; alternatively, annotations can be replaced. Annotations are useful for analyses, for example to specially identify subsets of records and trials of interest in the collection

```
# Annotate a query in CTGOV2 defined above
ctrLoadQueryIntoDb(
  queryterm = queries["CTGOV2"],
  annotation.text = "site_DE ",
  annotation.mode = "append",
  con = db
)
# * Appears specific for CTGOV REST API 2.0
# * Found search query from CTGOV2: cond=neuroblastoma&aggFilters=phase:2,ages:child,status:com
# * Checking trials using CTGOV REST API 2.0, found 111 trials
# (1/3) Downloading in 1 batch(es) (max. 1000 trials each; estimate: 11 Mb total)
# (2/3) Converting to NDJSON...
# (3/3) Importing records into database...
# JSON file #: 1 / 1
# = Imported or updated 111 trial(s)
# = Annotated retrieved records (111 records)
# Updated history ("meta-info" in "test")
# $n
# [1] 111
```

Load information using trial identifiers

When identifiers of clinical trials of interest are already known, this example shows how they can be processed to import the trial information into a database collection. This involves constructing a query that combines the identifiers and then iterating over the sets of identifiers. Note to combine identifiers into the `queryterm` depends on the specific register:

```
# ids of trials of interest
ctIds <- c(
  "NCT00001209", "NCT00001436", "NCT00187109", "NCT01516567", "NCT01471782",
  "NCT00357084", "NCT00357500", "NCT00365755", "NCT00407433", "NCT00410657",
  "NCT00436852", "NCT00445965", "NCT00450307", "NCT00450827", "NCT00471679",
  "NCT00492167", "NCT00499616", "NCT00503724"
```

```
)

# split into sets of each 10 trial ids
# (larger sets e.g. 50 may still work)
idSets <- split(ctIds, ceiling(seq_along(ctIds) / 10))

# variable to collect import results
result <- NULL

# iterate over sets of trial ids
for (idSet in idSets) {

  setResult <- ctrLoadQueryIntoDb(
    queryterm = paste0("term=", paste0(idSet, collapse = " ")),
    register = "CTGOV2",
    con = db
  )

  # check that queried ids have
  # successfully been loaded
  stopifnot(identical(
    sort(setResult$success), sort(idSet)))

  # append result
  result <- c(result, list(setResult))
}

# inspect results
as.data.frame(do.call(rbind, result))[, c("n", "failed")]
#    n failed
# 1 10   NULL
# 2  8   NULL

# queryterms for other registers for retrieving trials by their identifier:
#
# CTIS (note the comma separated values):
# https://euclinicaltrials.eu/ctis-public/search#searchCriteria=
# {"containAny":"2025-521008-22-00, 2024-519446-67-00, 2024-517647-31-00"}
#
# EUCTR (note the country suffix os to be removed, values separated with OR):
# https://www.clinicaltrialsregister.eu/ctr-search/search?
# query=2008-001606-16+OR+2008-001721-34+OR+2008-002260-33
#
```

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. The synonymous names can be used in queries in a register that does not offer search expansion. To obtain a character vector of synonyms for an active substance name:

```
# Search for synonyms
ctrFindActiveSubstanceSynonyms(
```

```
  activesubstance = "imatinib"
)
#  [1] "imatinib"          "CGP 57148"         "CGP 57148B"
#  [4] "CGP57148B"         "Gleevec"           "GLIVEC"
#  [7] "Imatinib"          "Imatinib Mesylate" "NSC 716051"
# [10] "ST1571"            "STI 571"           "STI571"
```

Using a MongoDB 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 at https://docs.mongodb.com/manual/reference/connection-string/. For recommended databases, see vignette Install R package ctrdata.

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

# Since the above access is read-only,
# just obtain fields of interest:
dbGetFieldsIntoDf(
  fields = c("a2_eudract_number",
             "e71_human_pharmacology_phase_i"),
  con = db)
#                   _id a2_eudract_number e71_human_pharmacology_phase_i
# 1 2010-024264-18-3RD    2010-024264-18                           TRUE
# 2  2010-024264-18-AT    2010-024264-18                           TRUE
# 3  2010-024264-18-DE    2010-024264-18                           TRUE
# 4  2010-024264-18-GB    2010-024264-18                           TRUE
# 5  2010-024264-18-IT    2010-024264-18                           TRUE
# 6  2010-024264-18-NL    2010-024264-18                           TRUE
```