Getting started with R package `ctrdata` for clinical trial information

Ralf Herold

2019-10-09

Contents

Install package `ctrdata` on a R system

The R Project website (https://www.r-project.org/) provides installers for the R system.

Alternatively, the R system can be used from software products such as R Studio (https://www.rstudio.com/products/RStudio/), which includes an open source integrated development environment (IDE), or Microsoft R Open (https://mran.microsoft.com/open/).

General information on the `ctrdata` package is available here: https://github.com/rfhb/ctrdata.

```
install.packages("ctrdata")
```

For using the development version of package `ctrdata`, install from GitHub:

```
# install preparatory package
install.packages(c("devtools", "httr"))
devtools::install_github("rfhb/ctrdata")
```

Either of the above should install package `ctrdata` into the user's library.

Internet access via proxy

Functions in package `ctrdata` that start with `ctr...` require access to trial registers over the internet via the `https` protocol. Package `ctrdata` checks and *automatically uses* the proxy that is set under MS Windows in system settings. However, proxy settings need to be set by the user for other operating systems and for authenticating proxies, such as follows:

```
Sys.setenv(https_proxy = "your_proxy.server.domain:8080")
Sys.setenv(https_proxy_user = "userid:password") # if needed
```

Additional installation aspects for MS Windows

On MS Windows, it seems recommended to not use UNC notation (such as `\\server\directory`) for specifying the user's library location, but the following notation using a local disk or mapped network share:

```
.libPaths("D:/my/directory/")
```

Package `ctrdata` requires on MS Windows the cygwin environment to be installed, into the local directory `c:\cygwin` (or any folder corresponding to `c:\cygw*`). The applications php, bash, perl, cat and sed in the cygwin environment are required for function `ctrLoadQueryIntoDb()` of package `ctrdata`. No other function in the package has this requirement. The installation of a minimal cygwin environment on MS Windows can be done from package `ctrdata` as follows:

```
ctrdata::installCygwinWindowsDoInstall()
```

If internet accces has to be via a proxy and this was not correctly detected automatically, it can be specified:

```
ctrdata::installCygwinWindowsDoInstall(proxy = "proxy.server.domain:8080")
```

Users who want or need to install cygwin manually can download the setup executable from here. In a MS Windows command window or Powershell window, use the following command line. The parameters are explained here.

```
setup-x86_64.exe --no-admin --quiet-mode --verbose --upgrade-also --root c:/cygwin
--site http://www.mirrorservice.org/sites/sourceware.org/pub/cygwin/
--packages perl,php-jsonc,php-simplexml
```

Database

So far, a RSQLite or a remote or a local MongoDB database can be used with the package `ctrdata`. Suggested installation instructions for a local MongoDB server are here; a remote MongoDB database server is accessible here.

Attach package `ctrdata`

```
library(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"
)
```

Click search parameters and execute search in browser

Refine the search until the trials of interest are listed in the browser. Currently, the total number of trials that can be retrieved with package `ctrdata` is intentionally limited to 5000.

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 <- "https://www.clinicaltrialsregister.eu/ctr-search/search?
# query=cancer&age=under-18&status=completed&phase=phase-one"

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

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

# Open browser with this query
# Note the register needs to be specified
# when it cannot be deduced from the query
ctrOpenSearchPagesInBrowser(
  input = q
)
```

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

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

# Count number of records
ctrLoadQueryIntoDb(
  queryterm = q,
  only.count = TRUE,
  con = db
)

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

# With file-base SQLite, takes 5 minutes for 1000 records.
# Speed is higher when using MongoDB (or memory-based SQLite).

# Show which queries have been downloaded into database so far
dbQueryHistory(con = db)
#       query-timestamp query-register query-records
# 1 2019-10-13 20:23:11          EUCTR           173
#                                                 query-term
# 1 query=cancer&age=under-18&status=completed&phase=phase-one
```

Analyse information on clinical trials

```r
# find names of fields of interest in database:
dbFindFields(
  namepart = "status",
  con = db
)
# Finding fields on server (may take some time)
# Field names cached for this session.
# [1] "x5_trial_status"
# [2] "b1_sponsor.b31_and_b32_status_of_the_sponsor"
# [3] "p_end_of_trial_status"

# Get all records that have values in all specified fields.
# Note that b31_... is a field within the array b1_...

# NOTE for SQLite:
# For SQLite, but NOT for MongoDB, the element b1_... at the
# root has to be retrieved and only from the nested list in
# this element, the sub-element b31_and_b32_status... can
# then be extracted, using the helpfer function below.
result <- dbGetFieldsIntoDf(
```

```
  fields = c("b1_sponsor",
             "p_end_of_trial_status"),
  con = db
)

# Show nested list in the element
str(result[1, "b1_sponsor"])
# List of 1
#  $ :'data.frame': 1 obs. of  4 variables:
#   ..$ _b1_sponsor                     : chr "1"
#   ..$ b11_name_of_sponsor             : chr "Wyeth Research
#   ..$ b134_country                    : chr "United States"
#   ..$ b31_and_b32_status_of_the_sponsor: chr "Commercial"

# Extract sought element from root element
tmp <- dfListExtractKey(
  result,
  list(c(
    "b1_sponsor", "b31_and_b32_status_of_the_sponsor")
))
result <- merge(result, tmp)

# Tabulate the status of the clinical trial
with(result,
     table(
       "Status"       = p_end_of_trial_status,
       "Sponsor type" = value)
)
#
#                     Sponsor type
# Status               Commercial Non-Commercial
#   Completed                 114             10
#   Not Authorised              1              0
#   Ongoing                    31              2
#   Prematurely Ended           6              0
#   Temporarily Halted          1              0
```