easyPubMed ver:2.9 - New features

Damiano Fantini

September 17, 2018

Install new version of easyPubMed from GitHub

The following code is aimed at downloading an installing the *dev* version of `easyPubMed` from GitHub. You'll need to install the `devtools` library first (available on CRAN).

```
# Install easyPubMed v2.7
library(devtools)
install_github("dami82/easyPubMed")

# Load library
library(easyPubMed)
```

Getting started - prep some data

The following code is aimed at downloading ad pre-processing a list of ~6000 PubMed records. These records will be downloaded in a series of 1000-record batches.

```
# Query pubmed and fetch many results
my_query <- 'Interleukin[TI] AND "2015"[PDAT]:"2017"[PDAT]'
my_query <- get_pubmed_ids(my_query)

# Download by 1000-item batches
my_batches <- seq(from = 1, to = my_query$Count, by = 1000)
my_abstracts_xml <- lapply(my_batches,  function(i) {
  fetch_pubmed_data(my_query, retmax = 1000, retstart = i)
})

# Store Pubmed Records as elements of a list
all_xml <- list()
for(x in my_abstracts_xml) {
  xx <- articles_to_list(x)
  for(y in xx) {
    all_xml[[(1 + length(all_xml))]] <- y
  }
}
```

Demo 1: fast extraction of PMID, Title, and Abstract

The following code illustrates the use of `article_to_df(, getAuthors = FALSE)`, for fast extraction of PubMed record titles and abstracts. This function can process PubMed records quickly, and will return all info except author info. Here, ~6000 records were processed in less than 2 min.

```
# Starting time: record
t.start <- Sys.time()

# Perform operation (use lapply here, no further parameters)
```

```
final_df <- do.call(rbind, lapply(all_xml, article_to_df,
                                  max_chars = -1, getAuthors = FALSE))

# Final time: record
t.stop <- Sys.time()

# How long did it take?
print(t.stop - t.start)
```

```
## Time difference of 1.422024 mins
```

```
# Show an excerpt of the results
head(final_df[,c("pmid", "year", "abstract")])
```

```
##       pmid year                     abstract
## 1 30062159 2018                        NA...
## 2 29933861 2018 Cytokines are fundamental...
## 3 29849682 2018 To explore the clinical s...
## 4 29806425 2018 To investigate the interl...
## 5 29798960 2018 Multiple myeloma (MM) is ...
## 6 29798418 2018 <b>Objective:</b>To inves...
```

```
# If interested in specific information,
# you can subset the dataframe and save the
# desired columns/features
id_abst_df <- final_df[,c("pmid", "abstract")]
head(id_abst_df)
```

```
##       pmid                     abstract
## 1 30062159                        NA...
## 2 29933861 Cytokines are fundamental...
## 3 29849682 To explore the clinical s...
## 4 29806425 To investigate the interl...
## 5 29798960 Multiple myeloma (MM) is ...
## 6 29798418 <b>Objective:</b>To inves...
```

Demo 2: full info extraction, including keywords

The following code illustrates the use of `article_to_df(, getKeywords = TRUE)`, for recursive extraction of PubMed record info, including keywords. Author info extraction is a time-consuming process. Here, we are extracting info from the first 1000 PubMed records returned by the query. Here, 1000 records were processed in ~3 min.

```
# Starting time: record
t.start <- Sys.time()

# Perform operation (use lapply here, no further parameters)
keyword_df <- do.call(rbind, lapply(all_xml[1:1000],
                                    article_to_df, autofill = T,
                                    max_chars = 100, getKeywords = T))

# Final time: record
t.stop <- Sys.time()
```

```r
# How long did it take?
print(t.stop - t.start)
```

```
## Time difference of 3.385884 mins
```

```r
# Visualize Keywords extracted from PubMed records
# Keyword and MeSH Concepts are separated by semicolons
print(keyword_df$keywords[1])
```

```
## [1] "NF- B; co-receptors; context-dependent signaling; immune signaling; inflammation..."
```

```r
# Show an excerpt of the results
keyword_df[seq(1, 200, by = 10), c("lastname", "firstname", "keywords")]
```

```
##        lastname        firstname                          keywords
## 1   Mac Gabhann           Feilim NF- B; co-receptors; context-d...
## 11          Jin          Shengli Platelet-rich plasma; interleu...
## 21          Liu         Qi-Xiang benign prostatic hyperplasia; ...
## 31         Yang          Hongyao IL1; breast cancer; case-contr...
## 41      Chandra           Girish Cancer pain - inflammation - i...
## 51     Boghdadi      Ghada Samir allergen immunotherapy; allerg...
## 61         Shen             Ming PD-1/PD-L1 pathway; immune tol...
## 71          Qin            Renyi PD-1/PD-L1 pathway; immune tol...
## 81        Zhang               Yi Drug-naïve patients; First-epi...
## 91     Pourmand       Gholamreza Deceased donor; Delayed graft ...
## 101           F          Ghaly M                              <NA>
## 111     Umrania Vanali Vinodbhai Chronic periodontitis; enzyme-...
## 121     Jimenez         Cristina                              <NA>
## 131     Jeromin          Andreas Biomarkers; Inflammation; Inte...
## 141        Nair                S Genetics; inflammation; interl...
## 151     Bellini       Geoffrey A minimally invasive colorectal ...
## 161          Lu             Lian Pathology Section; asthma; int...
## 171       Zhang           Yan-Li Actins; Androstadienes; Animal...
## 181     Fischer        Katarzyna Adult; Autoantibodies; Female;...
## 191        Park       Byung-Hyun interferon- ; interleukin 6; i...
```

Demo 3: full info extraction via parallelization

The following code illustrates the use of `article_to_df()` in conjunction with parallelization. If multiple cores are available, splitting the job in multiple tasks can support faster info extraction from a large number of records. Here, ~6000 records were processed in ~11 min.

```r
# Load required packages (available from CRAN).
# This will work on UNIX/LINUX systems.
# Windows systems may not support the following code.
library(parallel)
library(foreach)
library(doParallel)

# Starting time: record
t.start <- Sys.time()

# Start a cluster with 5 cores
cl <- makeCluster(5)
registerDoParallel(cl)
```

```r
# Perform operation (use foreach)
# The .combine argument guides result aggregation
fullDF <- tryCatch(
  {foreach(x=all_xml,
           .packages = 'easyPubMed',
           .combine = rbind) %dopar% article_to_df(pubmedArticle = x,
                                                   autofill = T,
                                                   max_chars = 500,
                                                   getKeywords = T,
                                                   getAuthors = T)},
  error = function(e) {NULL},
  finally = {stopCluster(cl)})

# Final time: record
t.stop <- Sys.time()

# How long did it take?
print(t.stop - t.start)
```

```
## Time difference of 11.77222 mins
```

```r
# Show an excerpt of the results
fullDF[seq(1, 200, by = 10), c("lastname", "keywords", "abstract")]
```

```
##        lastname           keywords                abstract
## 1   Mac Gabhann NF- B; co-recep...                    <NA>
## 11          Jin Platelet-rich p... To investigate the i...
## 21          Liu benign prostati... To investigate the e...
## 31         Yang IL1; breast can... Cytokines are known ...
## 41      Chandra Cancer pain - i... Cytokines play an im...
## 51     Boghdadi allergen immuno... Introduction  Allerg...
## 61         Shen PD-1/PD-L1 path... Dysregulation of reg...
## 71          Qin PD-1/PD-L1 path... Dysregulation of reg...
## 81        Zhang Drug-naïve pati... Schizophrenia is acc...
## 91     Pourmand Deceased donor;... Ischemia reperfusion...
## 101           F                <NA> IL-22 plays a vital ...
## 111     Umrania Chronic periodo... Host modulation with...
## 121     Jimenez                <NA> To determine the rol...
## 131     Jeromin Biomarkers; Inf... Obstructive sleep ap...
## 141        Nair Genetics; infla... The objective of the...
## 151     Bellini minimally invas... Minimally invasive c...
## 161          Lu Pathology Secti... This study presents ...
## 171       Zhang Actins; Androst... Exemestane (EXE) is ...
## 181     Fischer Adult; Autoanti... To analyze the corre...
## 191        Park interferon- ; i... Synovitis of the aff...
```

Demo 4: Faster queries using API key.

The following code illustrates the use of the argument `api_key`, which was introduced in version 2.9. E-utils users are allowed 3 requests/second without an API key. However, users can obtain an API key to increase the e-utils limit to 10 requests/second. For more information, visit: https://www.ncbi.nlm.nih.gov/account/settings/. Three `easyPubMed` functions can take the `api_key` argument: `get_pubmed_ids()`, `fetch_pubmed_data()`, and `batch_pubmed_download()`. Requests submitted by the latter function are automatically paced, therefore the use of a key may speed the queries if records are retrieved in small batches.

```r
# define a PubMed Query: this should return 40 results
my_query <- '"immune checkpoint" AND 2010[DP]:2012[DP]'

# Monitor time, and proceed with record download -- USING API_KEY!
t_key1 <- Sys.time()
batch_pubmed_download(my_query,
                      api_key = "4ea263f0f8e9108aee96ace507afXXXXXXXX",
                      batch_size = 2, dest_file_prefix = "TMP_api_")
t_key2 <- Sys.time()

# Monitor time, and proceed with record download -- DO NOT USE API_KEY!
t_nok1 <- Sys.time()
batch_pubmed_download(my_query,
                      batch_size = 2, dest_file_prefix = "TMP_no_")
t_nok2 <- Sys.time()
```

```r
# Compute time differences
# The use of a key makes the process faster
print(paste("With key:", t_key2 - t_key1))
```

```
## [1] "With key: 16.8162899017334"
```

```r
print(paste("W/o key:", t_nok2 - t_nok1))
```

```
## [1] "W/o key: 24.5748829841614"
```

Acknowledgements

Thank you for using `easyPubMed`. This software was developed by Damiano Fantini. This vignette was created and built in September 2018. Please, acknowledge Damiano Fantini's work and http://www.data-pulse.com when using this code and/or `easyPubMed`. Thank you. (C-2018 - Damiano Fantini). More info available at http://www.data-pulse.com

Session Info

```r
# Session Info
print(sessionInfo())
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.3 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] doParallel_1.0.11 iterators_1.0.9   foreach_1.4.4     easyPubMed_2.9
## [5] XML_3.98-1.16
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.18     codetools_0.2-15 digest_0.6.15    rprojroot_1.3-2
##  [5] backports_1.1.2  magrittr_1.5     evaluate_0.10.1  stringi_1.2.2
##  [9] rmarkdown_1.9.15 tools_3.4.3      stringr_1.3.1    yaml_2.1.19
## [13] compiler_3.4.3   htmltools_0.3.6  knitr_1.20
```