Package 'scholar'

July 3, 2018

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Version 0.1.7

License MIT + file LICENSE

Title Analyse Citation Data from Google Scholar

BugReports https://github.com/jkeirstead/scholar/issues

Description Provides functions to extract citation data from Google Scholar. Convenience functions are also provided for comparing multiple scholars and predicting future h-index values.

LazyData true

Depends R (>= 3.4.0)

Imports R.cache, dplyr, httr, rvest, stringr, xml2, tidygraph, ggraph, ggplot2

Suggests knitr, prettydoc, roxygen2

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Guangchuang Yu [aut, cre] (<https://orcid.org/0000-0002-6485-8781>),
James Keirstead [aut],
Gregory Jefferis [ctb],
Jorge Cimentada [ctb],
Max Czapanskiy [ctb]

Repository CRAN

Date/Publication 2018-07-03 08:40:02 UTC

R topics documented:

`compare_scholars` *Compare the citation records of multiple scholars*

Description

Compares the citation records of multiple scholars. This function compiles a data frame comparing the citations received by each of the scholar's publications by year of publication.

Usage

```
compare_scholars(ids, pagesize = 100)
```

Arguments

`ids`	a vector of Google Scholar IDs
`pagesize`	an integer specifying the number of articles to fetch for each scholar

Value

a data frame giving the ID of each scholar and the total number of citations received by work published in a year.

Examples

```
{

    ## How do Richard Feynmann and Stephen Hawking compare?
    ids <- c("B7vSqZsAAAAJ", "qj74uXkAAAAJ")
    df <- compare_scholars(ids)

}
```

compare_scholar_careers

Compare the careers of multiple scholars

Description

Compares the careers of multiple scholars based on their citation histories. The scholar's *career* is defined by the number of citations to his or her work in a given year (i.e. the bar chart at the top of a scholar's profile). The function has an `career` option that allows users to compare scholars directly, i.e. relative to the first year in which their publications are cited.

Usage

```
compare_scholar_careers(ids, career = TRUE)
```

Arguments

`ids`	a character vector of Google Scholar IDs
`career`	a boolean, should a column be added to the results measuring the year relative to the first citation year. Default = TRUE

Examples

```
{
    ## How do Richard Feynmann and Stephen Hawking compare?
    # Compare Feynman and Stephen Hawking
    ids <- c("B7vSqZsAAAAJ", "qj74uXkAAAAJ")
    df <- compare_scholar_careers(ids)
}
```

get_article_cite_history

Gets the citation history of a single article

Description

Gets the citation history of a single article

Usage

```
get_article_cite_history(id, article)
```

Arguments

`id`	a character string giving the id of the scholar
`article`	a character string giving the article id.

Value

a data frame giving the year, citations per year, and publication id

---

`get_citation_history` *Get historical citation data for a scholar*

Description

Gets the number of citations to a scholar's articles over the past nine years.

Usage

```
get_citation_history(id)
```

Arguments

`id`	a character string specifying the Google Scholar ID. If multiple ids are specified, only the first value is used and a warning is generated.

Details

This information is displayed as a bar plot at the top of a standard Google Scholar page and only covers the past nine years.

Value

a data frame giving the number of citations per year to work by the given scholar

---

`get_coauthors` *Gets the network of coauthors of a scholar*

Description

Gets the network of coauthors of a scholar

Usage

```
get_coauthors(id, n_coauthors = 5, n_deep = 1)
```

Arguments

id	a character string specifying the Google Scholar ID. If multiple ids are specified, only the first value is used and a warning is generated.
n_coauthors	Number of coauthors to explore. This number should usually be between 1 and 10 as choosing many coauthors can make the network graph too messy.
n_deep	The number of degrees that you want to go down the network. When n_deep is equal to 1 then grab_coauthor will only grab the coauthors of Joe and Mary, so Joe – > Mary –> All coauthors. This can get out of control very quickly if n_deep is set to 2 or above. The preferred number is 1, the default.

Details

Considering that scraping each publication for all coauthors is error prone, get_coauthors grabs only the coauthors listed on the google scholar profile (on the bottom right of the profile), not from all publications.

Value

A data frame with two columns showing all authors and coauthors.

See Also

plot_coauthors

Examples

```
## Not run:

library(scholar)
coauthor_network <- get_coauthors('amYIKXQAAAAJ&hl')
plot_coauthors(coauthor_network)

## End(Not run)
```

---

get_complete_authors *Get the Complete list of authors for a Publication*

Description

Found as Muhammad Qasim Pasta's solution here https://github.com/jkeirstead/scholar/issues/21

Usage

```
get_complete_authors(id, pubid)
```

Arguments

id	a Google Scholar ID
pubid	a Publication ID from a giving google Schalar ID

Value

a string containing the complete list of authors

Author(s)

Muhammad Qasim Pasta

Abram B. Fleishman

---

get_impactfactor *Get journal metrics.*

Description

Get journal metrics (impact factor) for a journal list.

Usage

```
get_impactfactor(journals, max.distance = 0.05)
```

Arguments

journals	a character list giving the journal list
max.distance	maximum distance allowed for a match bewteen journal and journal list. Expressed either as integer, or as a fraction of the pattern length times the maximal transformation cost (will be replaced by the smallest integer not less than the corresponding fraction), or a list with possible components

Value

Journal metrics data.

Author(s)

Dominique Makowski and Guangchuang Yu

Examples

```
library(scholar)

id <- get_publications("bg0BZ-QAAAAJ&hl")
impact <- get_impactfactor(journals=id$journal, max.distance = 0.1)

id <- cbind(id, impact)
```

get_num_articles *Calculates how many articles a scholar has published*

Description

Calculate how many articles a scholar has published.

Usage

```
get_num_articles(id)
```

Arguments

id	a character string giving the Google Scholar ID

Value

an integer value (max 100)

get_num_distinct_journals *Gets the number of distinct journals in which a scholar has published*

Description

Gets the number of distinct journals in which a scholar has published. Note that Google Scholar doesn't provide information on journals *per se*, but instead gives a title for the containing publication where applicable. So a *journal* here might actually be a journal, a book, a report, or some other publication outlet.

Usage

```
get_num_distinct_journals(id)
```

Arguments

id	a character string giving the Google Scholar id

Value

the number of distinct journals

get_num_top_journals *Gets the number of top journals in which a scholar has published*

Description

Gets the number of top journals in which a scholar has published. The definition of a 'top journal' comes from Acuna et al. and the original list was based on the field of neuroscience. This function allows users to specify that list for themselves, or use the default Acuna et al. list.

Usage

```
get_num_top_journals(id, journals)
```

Arguments

id	a character string giving a Google Scholar ID
journals	a character vector giving the names of the top journals. Defaults to Nature, Science, Nature Neuroscience, Proceedings of the National Academy of Sciences, and Neuron.

Source

DE Acuna, S Allesina, KP Kording (2012) Future impact: Predicting scientific success. Nature 489, 201-202. http://dx.doi.org/10.1038/489201a.

get_oldest_article *Gets the year of the oldest article for a scholar*

Description

Gets the year of the oldest article published by a given scholar.

Usage

```
get_oldest_article(id)
```

Arguments

id	a character string giving the Google Scholar ID

Value

the year of the oldest article

get_profile *Gets profile information for a scholar*

Description

Gets profile information for a researcher from Google Scholar. Each scholar profile page gives the researcher's name, affiliation, their homepage (if specified), and a summary of their key citation and impact metrics. The scholar ID can be found by searching Google Scholar at http://scholar.google.com.

Usage

```
get_profile(id)
```

Arguments

id	a character string specifying the Google Scholar ID. If multiple ids are specified, only the first value is used and a warning is generated. See the example below for how to profile multiple scholars.

Value

a list containing the scholar's name, affiliation, citations, impact metrics, fields of study, homepage and the author's list of coauthors provided by Google Scholar.

Examples

```
{
   ## Gets profiles of some famous physicists
   ids <- c("xJaxiEEAAAAJ", "qj74uXkAAAAJ")
   profiles <- lapply(ids, get_profile)
}
```

get_publications *Gets the publications for a scholar*

Description

Gets the publications of a specified scholar.

Usage

```
get_publications(id, cstart = 0, pagesize = 100, flush = FALSE)
```

Arguments

id	a character string specifying the Google Scholar ID. If multiple IDs are specified, only the publications of the first scholar will be retrieved.
cstart	an integer specifying the first article to start counting. To get all publications for an author, omit this parameter.
pagesize	an integer specifying the number of articles to fetch
flush	should the cache be flushed? Search results are cached by default to speed up repeated queries. If this argument is TRUE, the cache will be cleared and the data reloaded from Google.

Details

Google uses two id codes to uniquely reference a publication. The results of this method includes `cid` which can be used to link to a publication's full citation history (i.e. if you click on the number of citations in the main scholar profile page), and `pubid` which links to the details of the publication (i.e. if you click on the title of the publication in the main scholar profile page.)

Value

a data frame listing the publications and their details. These include the publication title, author, journal, number, cites, year, and two id codes (see details).

`plot_coauthors` *Plot a network of coauthors*

Description

Plot a network of coauthors

Usage

```
plot_coauthors(network, size_labels = 5)
```

Arguments

network	A data frame given by `get_coauthors`
size_labels	Size of the label names

Value

a `ggplot2` object but prints a plot as a side effect.

See Also

`get_coauthors`

Examples

```
## Not run:
library(scholar)
coauthor_network <- get_coauthors('amYIKXQAAAAJ&hl')
plot_coauthors(coauthor_network)

## End(Not run)
```

`predict_h_index` *Predicts the h-index for a researcher*

Description

Predicts the h-index for a researcher each year for ten years into the future using Acuna et al's method (see source). The model was fit to data from neuroscience researchers with an h-index greater than 5 and between 5 to 12 years since publishing their first article. So naturally if this isn't you, then the results should be taken with a large pinch of salt. For more caveats, see http://simplystatistics.org/2012/10/10/whats-wrong-with-the-predicting-h-index-paper/.

Usage

```
predict_h_index(id, journals)
```

Arguments

`id`	a character string giving the Google Scholar ID
`journals`	optional character vector of top journals. See `get_num_top_journals` for more details.

Details

Since the model is calibrated to neuroscience researchers, it is entirely possible that very strange (e.g. negative) h-indices will be predicted if you are a researcher in another field. A warning will be displayed if the sequence of predicted h-indices contains a negative value or is non-increasing.

Value

a data frame giving predicted h-index values in future

Note

A scientist has an h-index of n if he or she publishes n papers with at least n citations each. Values returned are fractional so it's up to your own vanity whether you want to round up or down.

Source

DE Acuna, S Allesina, KP Kording (2012) Future impact: Predicting scientific success. Nature 489, 201-202. http://dx.doi.org/10.1038/489201a. Thanks to DE Acuna for providing the full regression coefficients for each year ahead prediction.

Examples

```
{
   ## Predict h-index of original method author
  id <- "GAi23ssAAAAJ"
  df <- predict_h_index(id)
}
```

---

scholar *scholar*

Description

The scholar package provides functions to extract citation data from Google Scholar. There are also convenience functions for comparing multiple scholars and predicting h-index scores based on past publication records.

Note

A complementary set of Google Scholar functions can be found at http://biostat.jhsph.edu/~jleek/code/googleCite.r. The scholar package was developed independently.

Source

The package reads data from http://scholar.google.com. Dates and citation counts are estimated and are determined automatically by a computer program. Use at your own risk.

Index