Package: fairpub (via r-universe)

May 31, 2026

Type Package

Title How Fair Are You When You Publish/Cite Scientific Works?

Version 1.0.0

Description Provides a user-friendly way to compute the non-profit and academic friendly ratio of the bibliographic reference list before submitting a manuscript for peer review.

URL https://github.com/frbcesab/fairpub,https://frbcesab.github.io/fairpub

BugReports https://github.com/frbcesab/fairpub/issues

License GPL (>= 2)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends R (>= 4.1)

Imports bibtex, openalexR, stringdist

Suggests knitr, rmarkdown, testthat (>= 3.0.0), vcr, withr

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libssl-dev

Repository https://frbcesab.r-universe.dev

Date/Publication 2026-05-31 09:35:05 UTC

RemoteUrl https://github.com/FRBCesab/fairpub

RemoteRef HEAD

RemoteSha 2a5e8b3f7db4b5e7f1960fa07c9c4438fba6eb07

Contents

`fp_clean_doi` *Clean a DOI vector*

Description

This helper cleans DOIs (Digital Object Identifier) by removing prefix (`doi:`, `https://doi.org/` and `http://dx.doi.org/`) and using lower case.

Usage

```
fp_clean_doi(doi = NULL)
```

Arguments

`doi`	a `character` vector with Digital Object Identifiers (DOI).

Value

A `character` of DOI without prefix and in lower case.

Examples

```
dois <- c(
  "10.1098/rsos.160384",
  "10.1098/RSOS.160384",
  "doi: 10.1098/rsos.160384",
  "http://dx.doi.org/10.1098/rsos.160384",
  "https://doi.org/10.1098/rsos.160384",
  "HTTPS://DOI.ORG/10.1098/RSOS.160384",
  NA
)

fp_clean_doi(dois)
```

`fp_compute_citation_ratio` *Non profit & academic friendly ratio of citations*

Description

Scientific journals operate over a broad spectrum of publishing strategies, from strictly for-profit, to non-profit, and in-between business models (e.g. for-profit but academic friendly journals).

From a list of references, this function computes three citation ratios: the proportion of non-profit citations, the proportion of for-profit and academic friendly citations, and the proportion of for-profit and non-academic friendly citations (Beck *et al.* 2026).

It uses the OpenAlex bibliographic database (`https://openalex.org`) to retrieve journal names from article DOI and the DAFNEE database (`https://dafnee.isem-evolution.fr/`) to get the business model and the academic friendly status of journals.

Usage

```
fp_compute_citation_ratio(doi = NULL)
```

Arguments

`doi`	a `character` vector of Digital Object Identifiers (DOI). Can contain `NA` (book, book chapter, etc.).

Value

A `list` of two elements:

- `summary`, a `data.frame` with two columns (`metric` and `value`) reporting the following statistics:
  - number of total references (length of `doi` argument)
  - number of references with DOI
  - number of deduplicated references
  - number of references found in the OpenAlex database
  - number of references whose journal is indexed in the DAFNEE database
  - number of non-profit and academic friendly references
  - number of for-profit and academic friendly references
  - number of for-profit and non academic friendly references
- `ratios`, a vector of three ratios:
  - non-profit and academic friendly ratio
  - for-profit and academic friendly ratio
  - for-profit and non academic friendly ratio

References

Beck M et al. (2026) Citation self-awareness for a fairer academic publishing landscape. **BioScience**. DOI: doi:10.1093/biosci/biag028

Examples

```
# Be polite and send your email to OpenAlex API ----
options(openalexR.mailto = 'anonymous@mail.com')

# Path to the BibTeX provided by <fairpub> ----
filename <- system.file(
  file.path("extdata", "references.bib"),
  package = "fairpub"
)

# Extract DOI from BibTeX ----
doi_list <- fp_extract_doi(file = filename)

# Print DOI ----
head(doi_list)

## Not run:
# Compute citation ratio ----
fp_compute_citation_ratio(doi_list)
#> $summary
#>                                           metric value
#> 1                                Total references    38
#> 2                              References with DOI    33
#> 3                          Deduplicated references    33
#> 4                     References found in OpenAlex    33
#> 5                       References found in DAFNEE    11
#> 6     Non-profit and academic friendly references     9
#> 7     For-profit and academic friendly references     2
#> 8 For-profit and non-academic friendly references     0
#>
#> $ratios
#>     Non-profit and academic friendly     For-profit and academic friendly
#>                                 0.82                                 0.18
#> For-profit and non-academic friendly
#>                                 0.00

## End(Not run)
```

`fp_extract_doi` *Extract DOI from a BibTeX file or a string*

Description

This function detects and extracts DOI from bibliographic records. User can provides either a `character` vector (argument `x`) or the path to a BibTex file (argument `file`).

Usage

```
fp_extract_doi(x = NULL, file = NULL)
```

Arguments

x	a character vector. A string containing bibliographic records.
file	a character of length 1. The path to the BibTeX file to open.

Value

A character vector with extracted DOI. Some values can be NA in case of books, chapters, etc. or if references are malformed in the BibTeX.

Examples

```
# Argument 'x' (one DOI per element) ----
string <- c(
  "Beck M (2026) Citation self-awareness... 10.1093/biosci/biag028.",
  "Galtier N (2026) Time to publish... DOI: 10.32942/X24933",
  "Doe J (9999) Title... http://dx.doi.org/10.1162/qss(c)_00305",
  "Receveur A (2024) David vs Goliath... https://doi.org/10.1111/ele.14395",
  "Smith J (9999) This is a fake article."
)

## Extract DOI from a vector ----
fp_extract_doi(x = string)

# Argument 'x' (many DOI per element) ----
string <- paste(string, collapse = "\n")
cat(string)

## Extract DOI from a vector ----
fp_extract_doi(x = string)

# Argument 'file' ----

## Path to the BibTeX provided by <fairpub> ----
filename <- system.file(
  file.path("extdata", "references.bib"),
  package = "fairpub"
)

## Extract DOI from BibTeX ----
fp_extract_doi(file = filename)
```

`fp_get_article_fairness` *Get the fairness status of an article*

Description

By querying the OpenAlex bibliographic database (https://openalex.org) and the DAFNEE database (https://dafnee.isem-evolution.fr/), this function returns the business model and the academic friendly status of an article (more precisely the fairness status of the journal).

Usage

```
fp_get_article_fairness(doi = NULL)
```

Arguments

`doi`	a `character` of length 1. The Digital Object Identifiers (DOI) of the article.

Value

A `data.frame` with two columns: `journal`, the journal name, and `fairness`, the fairness status with the following possible values:

- Non-profit and academic friendly
- For-profit and academic friendly
- For-profit and non academic friendly
- Record not found in OpenAlex
- Record not found in DAFNEE database

Examples

```
# Be polite and send your email to OpenAlex API ----
options(openalexR.mailto = 'anonymous@mail.com')

## Not run:
# Fairness status ----
fp_get_article_fairness(doi = "10.1126/science.162.3859.1243")
#>   journal                          fairness
#> 1 Science   Non-profit and academic friendly

fp_get_article_fairness(doi = "10.1111/j.1461-0248.2005.00792.x")
#>           journal                        fairness
#> 1 Ecology Letters For-profit and academic friendly

fp_get_article_fairness(doi = "10.1038/35002501")
#>   journal                                fairness
#> 1  Nature   For-profit and non-academic friendly
```

```
# Article not found in OA ----
fp_get_article_fairness(doi = "10.xxxx/xxxx")
#>   journal                       fairness
#> 1      NA   Record not found in OpenAlex

# Journal not found in the DAFNEE database ----
fp_get_article_fairness(doi = "10.21105/joss.05753")
#>                               journal                              fairness
#> 1 The Journal of Open Source Software Record not found in DAFNEE database

## End(Not run)
```

`fp_get_journal_fairness` *Get the fairness status of a journal*

Description

By querying the OpenAlex bibliographic database (https://openalex.org) and the DAFNEE database (https://dafnee.isem-evolution.fr/), this function returns the business model and the academic friendly status of a journal.

Usage

```
fp_get_journal_fairness(journal = NULL)
```

Arguments

`journal`	a `character` of length 1. The name of the journal. Do not use journal abbreviation.

Value

A `data.frame` with two columns: `journal`, the journal name, and `fairness`, the fairness status with the following possible values:

- Non-profit and academic friendly
- For-profit and academic friendly
- For-profit and non academic friendly
- Record not found in OpenAlex
- Record not found in DAFNEE database

Examples

```
# Be polite and send your email to OpenAlex API ----
options(openalexR.mailto = 'anonymous@mail.com')

## Not run:
# Fairness status ----
fp_get_journal_fairness("Science")
#>   journal                          fairness
#> 1 Science   Non-profit and academic friendly

# Fuzzy search ----
fp_get_journal_fairness("Science of Nature")
#> No exact match found!
#> The fuzzy search returns these three best candidates:
#>   'The Science of Nature'
#>   'Science Advances'
#>   'People and Nature'

fp_get_journal_fairness("The Science of Nature")
#>                 journal                                fairness
#> 1 The Science of Nature   For-profit and non-academic friendly

## End(Not run)
```

`fp_get_openalex_author_id`

Get OpenAlex author ID

Description

Queries the OpenAlex bibliographic database (https://openalex.org) to retrieve an author's identifier.

Usage

```
fp_get_openalex_author_id(author = NULL, n = 10)
```

Arguments

`author`	a `character` vector of length 1. Name of the author.
`n`	an `integer` of length 1. Number of results to return (between 1 and 200, default is `10`).

Value

A data frame with the following columns:

id OpenAlex author ID

display_name Author name in OpenAlex

orcid ORCID identifier

works_count Number of publications

Examples

```
## Not run:
# Be polite and send your email to OpenAlex API ----
options(openalexR.mailto = 'anonymous@mail.com')

fp_get_openalex_author_id("Nicolas Casajus")
#>            id    display_name               orcid works_count
#> 1 A5004806463 Nicolas Casajus 0000-0002-5537-5294         102

fp_get_openalex_author_id("Nicolas Mouquet")
#>            id    display_name               orcid works_count
#> 1 A5001034207 Nicolas Mouquet 0000-0003-1840-6984         210

## End(Not run)
```

---

fp_get_openalex_author_works

Get and filter an author's works from OpenAlex

Description

Queries the OpenAlex bibliographic database (https://openalex.org) to retrieve works associated with an OpenAlex author identifier. Optionally filters publication types and incomplete records.

Usage

```
fp_get_openalex_author_works(
  author_id = NULL,
  select = c("article", "review", "letter"),
  drop_na = TRUE
)
```

Arguments

author_id	a character of length 1. OpenAlex author ID. This identifier can be retrieved with fp_get_openalex_author_id().
select	a character vector of work types to retain. Use fp_list_openalex_work_types() to list valid work types. Defaults to c("article", "review", "letter"). Set to NULL to keep all work types.
drop_na	a logical. If TRUE (default), works with missing DOI or missing source information are removed.

Details

This function is a wrapper around the OpenAlex API using the `openalexR` package. Results are automatically standardized and cleaned for downstream bibliometric analyses.

Some repositories and preprint servers (e.g. Zenodo, HAL, bioRxiv, figshare) may be excluded depending on the selected work types.

Value

A data frame containing one row per work with the following columns:

id OpenAlex work identifier

authors Work (first) author

title Work title

publication_year Year of publication

source_display_name Journal or source name

source_id OpenAlex source identifier

doi Digital Object Identifier

cited_by_count Citation count in OpenAlex

type OpenAlex work type

Examples

```
## Not run:
# Be polite and send your email to OpenAlex API ----
options(openalexR.mailto = 'anonymous@mail.com')

fp_get_openalex_author_works("A5004806463")
#>          id                    authors                                title
#> 1 W7143431770  Brunno F. Oliveira et al.     Species range shifts often...
#> 2 W7153879999         Miriam Beck et al.    Citation self-awareness for...
#> 3 W4406766122 Érica Rievrs Borges et al.         Road-river intersectio...
#> 4 W4415048605   Jonathan Bonfanti et al. Geographic, taxonomic and metr...
#> 5 W4411408576      Matthew McLean et al.       Conserving the beauty of...
#> 6 W4415113473     Nicolas Casajus et al.           forcis: An R package...
#>   publication_year                              source_display_name
#> 1             2026 Proceedings of the National Academy of Sciences
#> 2             2026                                     BioScience
#> 3             2025                     Applied Vegetation Science
#> 4             2025                                 Ecology Letters
#> 5             2025 Proceedings of the National Academy of Sciences
#> 6             2025             The Journal of Open Source Software
#>     source_id                      doi cited_by_count    type
#> 1  S125754415 10.1073/pnas.2515903123              1 article
#> 2  S121830084  10.1093/biosci/biag028              0 article
#> 3  S179963793      10.1111/avsc.70011              0 article
#> 4   S80967739       10.1111/ele.70220              1  review
#> 5  S125754415 10.1073/pnas.2415931122              1 article
#> 6 S4210214273     10.21105/joss.09217              0 article
```

```
## End(Not run)
```

`fp_get_openalex_doi` *Get OpenAlex publication DOI*

Description

Queries the OpenAlex bibliographic database (https://openalex.org) to retrieve metadata about publications matching a given title, including their DOI.

Usage

```
fp_get_openalex_doi(title = NULL, n = 10)
```

Arguments

`title`	a `character` vector of length 1. The title of the publication.
`n`	an `integer` of length 1. Number of results to return (between 1 and 200, default is `10`).

Value

A data frame with the following columns:

display_name Title of the publication

publication_year Year of publication

source_display_name Journal or source name

doi Digital Object Identifier (DOI)

Examples

```
## Not run:
# Be polite and send your email to OpenAlex API ----
options(openalexR.mailto = 'anonymous@mail.com')

# Search for an full title ----
fp_get_openalex_doi(
  "Citation self-awareness for a fairer academic publishing landscape"
)
#>                                     display_name publication_year
#> 1 Citation self-awareness for a fairer academic...             2026
#>   source_display_name                   doi
#> 1          BioScience 10.1093/biosci/biag028

# Search for a partial title ----
fp_get_openalex_doi("Citation fairer academic landscape")
#>                                     display_name publication_year
```

```
#> 1     Strategic citations for a fairer academic...             2025
#> 2 Citation self-awareness for a fairer academic...             2026
#>                        source_display_name                       doi
#> 1 bioRxiv (Cold Spring Harbor Laboratory) 10.1101/2025.08.06.668908
#> 2                              BioScience    10.1093/biosci/biag028

## End(Not run)
```

`fp_identify_duplicate_works` *Identify duplicate works based on title similarity*

Description

Groups potentially duplicate bibliographic records by computing pairwise string distances between work titles and clustering similar items.

Usage

```
fp_identify_duplicate_works(
  data = NULL,
  string_dist = "lv",
  hclust_method = "single",
  threshold = 0.2
)
```

Arguments

`data`	a `data.frame` containing at least a `title` column.
`string_dist`	a `character` string specifying the distance metric used by `stringdist::stringdistmatrix()`. Defaults to `"lv"` (Levenshtein distance).
`hclust_method`	a `character` string specifying the hierarchical clustering method used by `stats::hclust()`. Defaults to `"single"`.
`threshold`	a `numeric` value controlling cluster separation. Lower values produce more fine-grained clusters (stricter matching), while higher values merge more records into the same group.

Details

Title similarity is computed after basic text normalization (lowercasing, punctuation removal, whitespace trimming). Distances are calculated using `stringdist::stringdistmatrix()` and normalized by title length before hierarchical clustering.

This function does not remove duplicates but assigns a cluster identifier that can be used for downstream deduplication or grouping.

Value

The input `data.frame` with an additional column:

ref_id Integer cluster identifier grouping similar titles.

Examples

```
## Not run:
df <- data.frame(
  title = c(
    "Deep Learning for NLP",
    "Deep learning for natural language processing",
    "Quantum Computing Basics"
  )
)

fp_identify_duplicate_works(df)

## End(Not run)
```

`fp_list_dafnee_journals` *List DAFNEE journals*

Description

The DAFNEE database (Database of Academia-Friendly Journals in Ecology and Evolution, https://dafnee.isem-evolution.fr/) provides the business model and the academic friendly status of several journals in the field of Ecology and Evolution.

The `fairpub` package provides a selection of 287 DAFNEE journals and this function returns information about these journals.

Usage

```
fp_list_dafnee_journals()
```

Value

A `data.frame` with three columns:

- `journal`, the name of the journal
- `business_model`, the business model of the journal (non-profit or for-profit)
- `academic_friendly`, the academic friendly status of the journal (yes or no)

Examples

```
# List DAFNEE journals in fairpub ----
journals <- fp_list_dafnee_journals()

# Number of journals ----
nrow(journals)

# Preview of the outputs ----
head(journals)
```

`fp_list_openalex_work_types` *List valid OpenAlex work types*

Description

Returns the set of work types recognized by the OpenAlex database and used for filtering bibliographic records.

Usage

```
fp_list_openalex_work_types()
```

Details

These work types correspond to the classification system used by OpenAlex to describe scholarly outputs. They can be used to filter results in functions such as `fp_get_openalex_author_works()`.

Value

a `character` vector of valid OpenAlex work types.

Examples

```
fp_list_openalex_work_types()
```

Index