Package 'retraction'

July 10, 2026

Title Detect Retracted References in Documents and Bibliographies

Version 0.1.0

Description Scans manuscripts, bibliographies, and reference lists for citations to retracted publications so that authors can avoid citing retracted work. 'retraction' reads bibliographic formats (BibTeX, BibLaTeX, CSL-JSON, RIS, EndNote XML) and document formats (R Markdown, Quarto, 'LaTeX', Markdown, HTML, JATS XML, Word, and PDF), extracts and normalizes identifiers, and checks them against retraction data. The default data source is the Retraction Watch database served through the 'XeraRetractionTracker' API; 'Crossref', 'OpenAlex', 'Europe PMC', 'PubMed', 'DataCite', and a preprint source ('arXiv' and 'bioRxiv' withdrawals) are available as additional sources ('OpenAlex' retraction data is itself derived from Retraction Watch). Within each source, matching proceeds from exact Digital Object Identifier (DOI) and 'PubMed' identifier lookups to fuzzy title matching, and results are returned as a tidy table with a match-quality score and an optional self-contained HTML report.

License GPL-3

URL https://github.com/choxos/retraction, https://choxos.github.io/retraction/

BugReports https://github.com/choxos/retraction/issues

Depends R (>= 4.0)

Imports cli (>= 3.0.0), httr2, jsonlite, stringdist, tibble, tools, utils, xml2

Suggests arrow, covr, DBI, DT, furrr, future, httptest2, knitr, pdftools, rmarkdown, RSQLite, rstudioapi, shiny, stringi (>= 1.5.3), testthat (>= 3.0.0), withr, writexl

Config/testthat/edition 3

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.0.0

LazyData true

NeedsCompilation no

Author Ahmad Sofi-Mahmudi [aut, cre] (ORCID:
<https://orcid.org/0000-0001-6829-0823>)

Maintainer Ahmad Sofi-Mahmudi <a.sofimahmudi@gmail.com>

Repository CRAN

Date/Publication 2026-07-10 20:20:08 UTC

Contents

`annotate_bib` *Annotate a BibTeX/BibLaTeX file, marking retracted entries.*

Description

Inserts `note = {RETRACTED: <reason>}` into every flagged entry that does not already carry a RETRACTED note.

Usage

```
annotate_bib(bib_path, x, out_path = NULL)
```

Arguments

`bib_path`	Path to the source `.bib` file.
`x`	A `retraction_result` from checking that file (its `id` values must be the BibTeX keys).
`out_path`	Where to write the annotated file. Defaults to `"<name>-annotated.bib"` next to the source.

Value

Invisibly, `out_path`.

Examples

```
## Not run:
res <- check_file("refs.bib")
annotate_bib("refs.bib", res)

## End(Not run)
```

`as.data.frame.retraction_result` *Coerce a retraction result to a data frame.*

Description

Coerce a retraction result to a data frame.

Usage

```
## S3 method for class 'retraction_result'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A `retraction_result`.
`row.names, optional`	Passed to `base::as.data.frame()`.
`...`	Passed to `base::as.data.frame()`.

Value

A base data.frame.

`as_tibble.retraction_result` *Coerce a retraction result to a plain tibble.*

Description

Coerce a retraction result to a plain tibble.

Usage

```
## S3 method for class 'retraction_result'
as_tibble(x, ...)
```

Arguments

`x`	A `retraction_result`.
`...`	Unused.

Value

A plain tibble::tibble.

author_retractions *Retractions attributed to an author.*

Description

Retractions attributed to an author.

Usage

```
author_retractions(name, max_pages = 200L)
```

Arguments

name	Author name to search (family name is usually enough).
max_pages	Pagination cap; raise it to fetch more than `max_pages * 100` records.

Value

A data frame of matching retraction records, or `NULL` if none.

Examples

```
author_retractions("Wakefield")
```

badge_json *Write a shields.io endpoint badge for a checked result.*

Description

Produces a JSON file suitable for a shields.io endpoint badge, showing the number of retracted citations. Point a badge URL at the hosted file for a README or a paper's landing page.

Usage

```
badge_json(x, path = "retraction-badge.json")
```

Arguments

x	A `retraction_result`.
path	Output path for the JSON. Defaults to `"retraction-badge.json"`.

Value

Invisibly, `path`.

Examples

```
res <- check_dois("10.1016/S0140-6736(97)11096-0")
badge_json(res, tempfile(fileext = ".json"))
```

---

check_bib *Check a BibTeX or BibLaTeX bibliography for retracted references*

Description

A thin wrapper around `check_file()` that forces the BibTeX parser.

Usage

```
check_bib(
  path,
  sources = getOption("retraction.sources", "xera"),
  offline = FALSE,
  flag_nature = c("Retraction", "Expression of Concern"),
  allow_fuzzy = TRUE,
  resolve_ids = TRUE,
  progress = TRUE,
  strict = FALSE
)
```

Arguments

path	Path to a `.bib` file.
sources	Sources to query. A character vector of names from `list_backends()`, or `"all"`. Defaults to `getOption("retraction.sources", "xera")`.
offline	If `TRUE`, match against the local snapshot built by `retraction_sync()` instead of querying the network.
flag_nature	Notice labels that count as "flag this citation". Defaults to Retraction and Expression of Concern.
allow_fuzzy	Allow fuzzy title matching for references without a usable identifier.
resolve_ids	If `TRUE`, resolve PMID-only references to a DOI via OpenAlex before matching.
progress	Show a progress bar in interactive sessions.
strict	If `TRUE`, error when any reference cannot be checked (an `unchecked` row from a network or source failure) instead of returning it, so a failure cannot be mistaken for a clean result. Useful in CI.

Value

A `retraction_result` tibble.

Examples

```
bib <- system.file("extdata", "example.bib", package = "retraction")
if (nzchar(bib)) check_bib(bib)
```

`check_dois` *Check a set of DOIs or PMIDs for retraction*

Description

Check a set of DOIs or PMIDs for retraction

Usage

```
check_dois(
  x,
  sources = getOption("retraction.sources", "xera"),
  offline = FALSE,
  flag_nature = c("Retraction", "Expression of Concern"),
  allow_fuzzy = TRUE,
  resolve_ids = TRUE,
  progress = TRUE,
  strict = FALSE
)
```

Arguments

`x`	A character vector of DOIs and/or PMIDs (mixed is fine; each value is auto-detected).
`sources`	Sources to query. A character vector of names from `list_backends()`, or `"all"`. Defaults to `getOption("retraction.sources", "xera")`.
`offline`	If `TRUE`, match against the local snapshot built by `retraction_sync()` instead of querying the network.
`flag_nature`	Notice labels that count as "flag this citation". Defaults to Retraction and Expression of Concern.
`allow_fuzzy`	Allow fuzzy title matching for references without a usable identifier.
`resolve_ids`	If `TRUE`, resolve PMID-only references to a DOI via OpenAlex before matching.
`progress`	Show a progress bar in interactive sessions.
`strict`	If `TRUE`, error when any reference cannot be checked (an `unchecked` row from a network or source failure) instead of returning it, so a failure cannot be mistaken for a clean result. Useful in CI.

Value

A `retraction_result` tibble.

Examples

```
check_dois(c("10.1016/S0140-6736(97)11096-0", "10.1126/science.aac4716"))
```

`check_file` *Check a document or bibliography file for retracted references*

Description

Reads a file, extracts references and identifiers, and checks them. Supported formats: BibTeX/BibLaTeX (`.bib`), CSL-JSON (`.json`), RIS (`.ris`), EndNote XML and JATS XML (`.xml`), Word (`.docx`), PDF (`.pdf`), and any text-like document (`.Rmd`, `.qmd`, `.tex`, `.md`, `.txt`, `.html`), from which DOIs are scraped.

Usage

```
check_file(
  path,
  format = NULL,
  sources = getOption("retraction.sources", "xera"),
  offline = FALSE,
  flag_nature = c("Retraction", "Expression of Concern"),
  allow_fuzzy = TRUE,
  resolve_ids = TRUE,
  progress = TRUE,
  strict = FALSE
)
```

Arguments

`path`	One or more file paths.
`format`	Force a parser (e.g. "bib", "ris", "csljson", "endnote", "jats", "docx", "pdf", "text"). When `NULL`, inferred from the extension and, for `.xml`, the root element.
`sources`	Sources to query. A character vector of names from `list_backends()`, or `"all"`. Defaults to `getOption("retraction.sources", "xera")`.
`offline`	If `TRUE`, match against the local snapshot built by `retraction_sync()` instead of querying the network.
`flag_nature`	Notice labels that count as "flag this citation". Defaults to Retraction and Expression of Concern.
`allow_fuzzy`	Allow fuzzy title matching for references without a usable identifier.
`resolve_ids`	If `TRUE`, resolve PMID-only references to a DOI via OpenAlex before matching.
`progress`	Show a progress bar in interactive sessions.
`strict`	If `TRUE`, error when any reference cannot be checked (an `unchecked` row from a network or source failure) instead of returning it, so a failure cannot be mistaken for a clean result. Useful in CI.

Value

A retraction_result tibble.

Examples

```
bib <- system.file("extdata", "example.bib", package = "retraction")
if (nzchar(bib)) check_file(bib)
```

`check_included_studies` *Check the included studies of a systematic review or meta-analysis.*

Description

Check the included studies of a systematic review or meta-analysis.

Usage

```
check_included_studies(
  ids,
  sources = getOption("retraction.sources", "xera"),
  offline = FALSE,
  flag_nature = c("Retraction", "Expression of Concern"),
  ...
)
```

Arguments

`ids`	A character vector of DOIs or PMIDs for the included studies. Duplicates are collapsed and counted once.
`sources, offline, flag_nature, ...`	Passed to `check_dois()`.

Value

A `retraction_review` (a `retraction_result` with `n_included`, `n_unique`, `n_checked`, `n_unchecked`, `n_possible`, and `n_retracted` attributes).

Examples

```
check_included_studies(c("10.1016/S0140-6736(97)11096-0",
                         "10.1136/bmj.331.7531.1512"))
```

check_pmc *Check the reference lists of open-access PubMed Central articles*

Description

For each input, resolves it to a PubMed Central article, reports whether the open-access full text (with a reference list) is available, and if so checks every reference in that article for retractions. Inputs may be a PMID, PMCID, DOI, article title, or a whole reference string, in any mix.

Usage

```
check_pmc(
  x,
  sources = getOption("retraction.sources", "xera"),
  offline = FALSE,
  flag_nature = c("Retraction", "Expression of Concern"),
  allow_fuzzy = TRUE,
  resolve_ids = TRUE,
  cache = TRUE,
  progress = TRUE,
  strict = FALSE
)
```

Arguments

x	A character (or numeric) vector of PMIDs, PMCIDs, DOIs, titles, or reference strings.
sources	Sources to query. A character vector of names from `list_backends()`, or `"all"`. Defaults to `getOption("retraction.sources", "xera")`.
offline	If `TRUE`, match against the local snapshot built by `retraction_sync()` instead of querying the network.
flag_nature	Notice labels that count as "flag this citation". Defaults to Retraction and Expression of Concern.
allow_fuzzy	Allow fuzzy title matching for references without a usable identifier.
resolve_ids	If `TRUE`, resolve PMID-only references to a DOI via OpenAlex before matching.
cache	Cache fetched XML on disk (see `pmc_fetch_xml()`).
progress	Show a progress bar in interactive sessions.
strict	If `TRUE`, fail closed: abort when any input cannot be resolved to a PubMed Central article or its open-access full text cannot be retrieved, and (as in `check_dois()`) when any resulting reference cannot be checked, instead of returning a clean-looking partial result.

Details

Resolving and fetching the article always use the network, even when `offline = TRUE` (which controls only the retraction data source used for matching). Per-article open-access status is available via `pmc_articles()`.

Value

A `retraction_result` tibble of the references found across the open-access articles (the `source_file` column records the PMCID). The per-article open-access summary is attached and retrievable with `pmc_articles()`.

See Also

`pmc_articles()`, `pmc_fetch_xml()`

Examples

```
## Not run:
res <- check_pmc(c("PMC5334499", "10.1371/journal.pone.0000217", "29939664"))
pmc_articles(res)   # open-access status per input
retracted(res)      # any retracted references found

## End(Not run)
```

---

check_preprint *Check whether a preprint has been withdrawn*

Description

Accepts an arXiv identifier (e.g. `"2401.01234"` or `"arXiv:2401.01234"`), an arXiv DOI (`10.48550/arXiv.*`), or a bioRxiv DOI (`10.1101/*`).

Usage

```
check_preprint(x)
```

Arguments

x	A preprint identifier or DOI.

Value

A one-row tibble with `id`, `server`, `withdrawn`, and `title`; or `NULL` if `x` is not a recognized preprint identifier.

Examples

```
check_preprint("10.1101/2020.01.30.927871")
```

check_refs *Check a data frame of references for retraction*

Description

Check a data frame of references for retraction

Usage

```
check_refs(
  data,
  doi_col = NULL,
  pmid_col = NULL,
  title_col = NULL,
  author_col = NULL,
  year_col = NULL,
  sources = getOption("retraction.sources", "xera"),
  offline = FALSE,
  flag_nature = c("Retraction", "Expression of Concern"),
  allow_fuzzy = TRUE,
  resolve_ids = TRUE,
  progress = TRUE,
  strict = FALSE
)
```

Arguments

data	A data frame of references.
doi_col, pmid_col, title_col, author_col, year_col	Column names. When NULL (default) they are auto-detected from common names.
sources	Sources to query. A character vector of names from list_backends(), or "all". Defaults to getOption("retraction.sources", "xera").
offline	If TRUE, match against the local snapshot built by retraction_sync() instead of querying the network.
flag_nature	Notice labels that count as "flag this citation". Defaults to Retraction and Expression of Concern.
allow_fuzzy	Allow fuzzy title matching for references without a usable identifier.
resolve_ids	If TRUE, resolve PMID-only references to a DOI via OpenAlex before matching.
progress	Show a progress bar in interactive sessions.
strict	If TRUE, error when any reference cannot be checked (an unchecked row from a network or source failure) instead of returning it, so a failure cannot be mistaken for a clean result. Useful in CI.

Value

A retraction_result tibble.

Examples

```
df <- data.frame(doi = "10.1016/S0140-6736(97)11096-0", title = "Ileal-lymphoid...")
check_refs(df)
```

---

check_zotero *Check a Zotero library for retracted references*

Description

Reads a Zotero SQLite database directly (read-only) and checks every item's DOI and title. If Zotero has the library open the database may be locked; in that case export the library via Better BibTeX and use check_file().

Usage

```
check_zotero(
  path = zotero_default_db(),
  sources = getOption("retraction.sources", "xera"),
  offline = FALSE,
  flag_nature = c("Retraction", "Expression of Concern"),
  allow_fuzzy = TRUE,
  resolve_ids = TRUE,
  progress = TRUE,
  strict = FALSE
)
```

Arguments

path	Path to zotero.sqlite, or the directory containing it. Defaults to the standard per-user location.
sources	Sources to query. A character vector of names from list_backends(), or "all". Defaults to getOption("retraction.sources", "xera").
offline	If TRUE, match against the local snapshot built by retraction_sync() instead of querying the network.
flag_nature	Notice labels that count as "flag this citation". Defaults to Retraction and Expression of Concern.
allow_fuzzy	Allow fuzzy title matching for references without a usable identifier.
resolve_ids	If TRUE, resolve PMID-only references to a DOI via OpenAlex before matching.
progress	Show a progress bar in interactive sessions.
strict	If TRUE, error when any reference cannot be checked (an unchecked row from a network or source failure) instead of returning it, so a failure cannot be mistaken for a clean result. Useful in CI.

Value

A retraction_result tibble. Requires the suggested DBI and RSQLite packages.

Examples

```
## Not run:
check_zotero()

## End(Not run)
```

`classify_timing` *Classify each citation relative to the document's date.*

Description

Adds a `timing` column. The default date is the document's authoring date (git commit or mtime), which is **not** the date a specific citation was added, so the labels are deliberately conservative: `"document_after_retraction"` means the document was written after the work was retracted, not that the citation was knowingly added post-retraction. Pass `citation_dates` for true citation-level timing.

Usage

```
classify_timing(x, document_date = Sys.Date(), citation_dates = NULL)
```

Arguments

`x`	A `retraction_result`.
`document_date`	The document's date, `Date` or `YYYY-MM-DD`. Defaults to today; see `manuscript_date_of()` to derive it from a file.
`citation_dates`	Optional named vector (`Date`/string) of per-citation dates, named by `id` or DOI. When supplied for a row, that date is used and the label becomes `"cited_after_retraction"` / `"cited_before_retraction"`.

Value

`x` with an added character `timing` column (`NA` when no retraction date is known).

Examples

```
res <- check_dois("10.1016/S0140-6736(97)11096-0")
classify_timing(res, document_date = "2015-01-01")
```

compare_sources *Summarize cross-source (dis)agreement in a result*

Description

The rows where the selected sources disagreed, with which confirmed and which dissented. Meaningful only when more than one source was queried.

Usage

```
compare_sources(x)
```

Arguments

x	A retraction_result.

Value

A tibble of the disagreeing rows (id, identifier, status, confirmed_by, dissenting); zero rows when all sources agree.

Examples

```
res <- check_dois("10.1016/S0140-6736(97)11096-0",
                  sources = c("xera", "openalex"))
compare_sources(res)
```

evaluate_policy *Evaluate a checked result against a fail policy*

Description

Decides whether a fail_policy() is triggered by a result, for building custom gates (the CLI, knit gate, and GitHub Action use it).

Usage

```
evaluate_policy(res, policy, n_errors = 0L)
```

Arguments

res	A retraction_result.
policy	A fail_policy().
n_errors	Count of files that errored during the scan.

Value

A list: `fail` (logical), `triggered` (character reasons), `counts` (the per-state counts), and `n_errors`.

Examples

```
res <- check_dois("10.1016/S0140-6736(97)11096-0")
evaluate_policy(res, fail_policy("flagged"))$fail
```

`explain_result` *Explain the verdict for each reference in a result*

Description

A human-readable sentence per row: what matched, on which identifier, at what confidence, which sources confirmed, and any disagreement.

Usage

```
explain_result(x, rows = NULL)
```

Arguments

`x`	A `retraction_result`.
`rows`	Optional integer or logical row selector; defaults to all rows.

Value

A tibble with `id`, `status`, and `explanation`.

Examples

```
res <- check_dois("10.1016/S0140-6736(97)11096-0")
explain_result(res)
```

export_result *Export a checked result to CSV, JSON, or Excel.*

Description

Format is chosen from the file extension: `.csv`, `.json`, or `.xlsx` (Excel requires the suggested `writexl` package).

Usage

```
export_result(x, path)
```

Arguments

`x`	A `retraction_result`.
`path`	Output path; its extension selects the format.

Value

Invisibly, `path`.

Examples

```
## Not run:
res <- check_file("refs.bib")
export_result(res, "results.xlsx")
export_result(res, "results.csv")

## End(Not run)
```

exposure_score *Retraction-exposure summary, with denominator diagnostics.*

Description

A bare `flagged / n` headline is misleading when many rows are unchecked or only "possible", so this reports the full breakdown and two rates: per total, and per successfully-checked.

Usage

```
exposure_score(x)
```

Arguments

`x`	A `retraction_result`.

Value

A named list: `n_total`, `n_checked`, `n_flagged`, `n_possible`, `n_unchecked`, `flagged_per_total`, `flagged_per_checked`.

Examples

```
res <- check_dois("10.1016/S0140-6736(97)11096-0")
exposure_score(res)
```

`fail_policy` *A fail policy shared by the CLI, GitHub Action, and knit gate.*

Description

A fail policy shared by the CLI, GitHub Action, and knit gate.

Usage

```
fail_policy(on = "flagged")
```

Arguments

`on`	Which result states should cause failure. Any of `"flagged"`, `"possible"`, `"unchecked"`, `"error"` (a file that failed to parse or fetch). Default fails only on confirmed flags.

Value

A `retraction_fail_policy` object.

`journal_retractions` *Retraction summary for a journal.*

Description

Retraction summary for a journal.

Usage

```
journal_retractions(journal, max_pages = 200L)
```

Arguments

`journal`	Journal name to search.
`max_pages`	Pagination cap; journals can have thousands of retractions, so raise this if you need the complete set.

Value

A list: `n`, `by_reason`, `by_year`, `records`. `NULL` if none found.

Examples

```
journal_retractions("The Lancet")
```

---

list_backends *Names of all registered backends.*

Description

Names of all registered backends.

Usage

```
list_backends()
```

Value

A character vector of source names.

Examples

```
list_backends()
```

---

manuscript_date_of *Best-effort authoring date of a manuscript file (git commit date, else mtime).*

Description

Best-effort authoring date of a manuscript file (git commit date, else mtime).

Usage

```
manuscript_date_of(path)
```

Arguments

`path`	Path to the manuscript.

Value

A `Date`, or today's date if nothing better is available.

normalize_doi *Normalize a DOI.*

Description

Strips resolver prefixes (`https://doi.org/`, `doi:`), lowercases, and trims trailing punctuation, following Crossref's canonicalization guidance. Empty results become `NA`.

Usage

```
normalize_doi(x)
```

Arguments

`x`	A character vector of DOIs.

Value

A character vector of normalized DOIs, with `NA` for unparseable input.

Examples

```
normalize_doi("https://doi.org/10.1234/ABC. ")
normalize_doi("doi:10.1016/S0140-6736(97)11096-0")
```

normalize_pmcid *Normalize a PubMed Central identifier*

Description

Canonicalizes a PMCID to the `PMC#######` form, accepting `"PMC123"`, `"123"`, or `123`.

Usage

```
normalize_pmcid(x)
```

Arguments

`x`	A character or numeric vector of PMCIDs.

Value

A character vector of `PMC`-prefixed identifiers, `NA` where none is present.

Examples

```
normalize_pmcid(c("PMC5334499", "5334499", "pmc 5334499"))
```

normalize_pmid *Normalize a PubMed identifier to bare digits.*

Description

Accepts an optional `PMID:` prefix. Input that is not a bare run of digits after the prefix is rejected as `NA` rather than coerced, so a PMCID or DOI placed in a PMID field is not silently turned into a fabricated PMID.

Usage

```
normalize_pmid(x)
```

Arguments

`x`	A character (or numeric) vector of PMIDs, optionally `PMID:`-prefixed.

Value

A character vector of digit-only PMIDs, `NA` where none is present.

Examples

```
normalize_pmid("PMID: 12345678")
normalize_pmid("PMC12345")  # NA: a PMCID is not a PMID
```

normalize_title *Normalize a title for fuzzy comparison only.*

Description

Transliterates to lowercase ASCII Latin (so diacritics, ligatures, and full-width forms fold to a comparable base form), removes a leading "Retracted:" style marker, strips markup and punctuation, and collapses whitespace. The original title is retained elsewhere for reporting; this form is used solely for string distance. If the optional stringi package is installed, non-Latin scripts (such as CJK) are also romanized; otherwise they are dropped by the base fallback.

Usage

```
normalize_title(x)
```

Arguments

`x`	A character vector of titles.

Value

A character vector of normalized titles.

Examples

```
normalize_title("Résumé of a Study")   # accents folded
```

`pmc_articles` *Per-article open-access summary from a* `check_pmc()` *result*

Description

Per-article open-access summary from a `check_pmc()` result

Usage

```
pmc_articles(x)
```

Arguments

`x`	A `retraction_result` returned by `check_pmc()`.

Value

A tibble with `input`, `pmcid`, `doi`, `resolved` (a PMC article was found), `retrieved` (its full text was fetched), `is_open_access` (equal to `retrieved`), `n_references`, and `n_retracted`.

`pmc_fetch_xml` *Retrieve open-access PMC full-text XML for a PMCID*

Description

Tries NCBI EFetch (`db = pmc`) first, then the PMC OAI-PMH service. Results are cached on disk under `retraction_cache_dir()`; an existing non-empty cache file is reused unless `overwrite = TRUE`.

Usage

```
pmc_fetch_xml(pmcid, cache = TRUE, overwrite = FALSE)
```

Arguments

`pmcid`	A PMCID (any form accepted by `normalize_pmcid()`).
`cache`	Cache the XML on disk and reuse it on later calls.
`overwrite`	Re-fetch even if a cache file exists.

Value

An `xml_document`, or `NULL` if the article could not be retrieved.

Examples

```
## Not run:
doc <- pmc_fetch_xml("PMC5334499")

## End(Not run)
```

`primary_reason_bucket` *Primary (lossy) reason bucket: the first matching bucket.*

Description

Coarse and English-only; for reasons with several causes it keeps only the first match in bucket order (misconduct > plagiarism > error > ethical > process). Use `reason_buckets()` to keep all causes.

Usage

```
primary_reason_bucket(reason)
```

Arguments

`reason`	A character vector of reason strings.

Value

A character vector of primary buckets.

Examples

```
primary_reason_bucket(c("Fabrication of data", "Duplicate publication", "Unknown"))
```

print.retraction_result *Print a retraction result.*

Description

Print a retraction result.

Usage

```
## S3 method for class 'retraction_result'
print(x, ...)
```

Arguments

x	A retraction_result.
...	Unused.

Value

The input x, invisibly.

print.retraction_review *Print a systematic-review check, leading with denominators.*

Description

Print a systematic-review check, leading with denominators.

Usage

```
## S3 method for class 'retraction_review'
print(x, ...)
```

Arguments

x	A retraction_review.
...	Unused.

Value

x, invisibly.

reason_buckets *All matching reason buckets (multi-label).*

Description

All matching reason buckets (multi-label).

Usage

```
reason_buckets(reason)
```

Arguments

reason	A single free-text reason string.

Value

A character vector of every matching bucket (possibly several), or "other" if none match.

Examples

```
reason_buckets("Fabrication of data; Authorship disputes")
```

reason_summary *Tabulate primary reason buckets across a checked result.*

Description

Tabulate primary reason buckets across a checked result.

Usage

```
reason_summary(x)
```

Arguments

x	A retraction_result.

Value

A named integer table of primary-bucket counts over the matched rows.

Examples

```
res <- check_dois("10.1016/S0140-6736(97)11096-0")
reason_summary(res)
```

render_report *Render a retraction result as an HTML or Markdown report*

Description

Produces a shareable, self-contained report. HTML output embeds its own CSS and needs no other software to view.

Usage

```
render_report(
  x,
  output_file = NULL,
  format = c("html", "md"),
  title = "Retraction report",
  open = FALSE
)
```

Arguments

x	A retraction_result.
output_file	Output path. When NULL, a temporary file is used.
format	"html" (default) or "md".
title	Report title.
open	Open the report in a browser (interactive HTML only).

Value

Invisibly, the path to the written report.

Examples

```
res <- check_dois("10.1016/S0140-6736(97)11096-0")
report <- render_report(res, tempfile(fileext = ".html"))
```

retracted *Subset of references flagged as retracted (or otherwise notable).*

Description

Subset of references flagged as retracted (or otherwise notable).

Usage

```
retracted(x, which = c("flagged", "possible", "all_matched"))
```

Arguments

x	A retraction_result.
which	"flagged" (default; high-confidence flagged citations), "possible" (matched to a flagged record but below the confidence threshold), or "all_matched".

Value

A retraction_result with the selected rows.

---

retraction_addin_check_active *RStudio addin: check retractions in the active source document.*

Description

For a **saved** document, checks the file in place, so relative bibliography: paths in the YAML header resolve. For an **unsaved** buffer it can only scan inline identifiers (DOIs in the text), not an external bibliography, and says so. Prints the result and moves the cursor to the first flagged citation when a location is known.

Usage

```
retraction_addin_check_active()
```

Value

Invisibly, the retraction_result.

---

retraction_app *Launch the retraction triage Shiny app*

Description

Opens a small web app to upload a bibliography or document and browse the flagged, possible, and clean references interactively.

Usage

```
retraction_app(...)
```

Arguments

...	Passed to shiny::runApp().

Value

Runs the app; does not return a value. Requires the suggested shiny and DT packages.

Examples

```
## Not run:
retraction_app()

## End(Not run)
```

`retraction_cache_dir` *Location of the retraction cache directory*

Description

Location of the retraction cache directory

Usage

```
retraction_cache_dir(create = FALSE)
```

Arguments

`create`	Create the directory if it does not exist.

Value

The cache directory path (via `tools::R_user_dir()`).

Examples

```
retraction_cache_dir()
```

`retraction_clear_cache` *Clear the package cache*

Description

Removes the local retraction snapshot and any cached PubMed Central XML, and resets the in-memory cache.

Usage

```
retraction_clear_cache()
```

Value

Invisibly TRUE.

Examples

```
## Not run:
retraction_clear_cache()

## End(Not run)
```

`retraction_example` *Example references for demonstrating retraction checks*

Description

A small, stable set of references used in examples and tests. The first reference (Wakefield et al. 1998) was retracted by The Lancet in 2010; the other two are controls that have not been retracted. Bundled so that examples can run without network access.

Usage

```
retraction_example
```

Format

A data frame with 3 rows and 4 variables:

doi Digital Object Identifier of the reference.

title Article title.

year Publication year.

note Whether the item is retracted, for reference.

Source

Retraction Watch via the XeraRetractionTracker API, https://openscience.xera.ac/retractions.

Examples

```
retraction_example

check_refs(retraction_example)
```

`retraction_knit_check` *Fail (or warn) a knit when the document cites retracted work.*

Description

Call from a setup chunk. Checks the document and/or its bibliography; if the fail policy is triggered it aborts (or warns) the render.

Usage

```
retraction_knit_check(
  input = NULL,
  bib = NULL,
  on = "flagged",
  action = c("error", "warn"),
  sources = getOption("retraction.sources", "xera"),
  offline = FALSE
)
```

Arguments

`input`	Path to the document being knit. Defaults to `knitr::current_input()` when knitting.
`bib`	Optional path(s) to bibliography files to check as well.
`on`	Fail policy states (see `fail_policy()`): any of `"flagged"`, `"possible"`, `"unchecked"`, `"error"`. Default `"flagged"`.
`action`	`"error"` (default; abort the render) or `"warn"`.
`sources, offline`	Passed to `check_file()`.

Value

Invisibly, the combined `retraction_result`.

Examples

```
## Not run:
retraction::retraction_knit_check(bib = "refs.bib",
                                  on = c("flagged", "unchecked"))

## End(Not run)
```

retraction_main *CLI main: scan files and exit per a fail policy.*

Description

For `Rscript -e 'retraction::retraction_main()' [--fail-on=flagged,unchecked] file ...`. Prints per-state counts and any file errors, then exits: 0 = policy satisfied, 1 = policy triggered, 2 = usage/argument error.

Usage

```
retraction_main(args = commandArgs(trailingOnly = TRUE))
```

Arguments

`args`	Character vector of arguments (default: command-line trailing args).

Value

Does not return; exits the R session.

retraction_scan *Scan files for retracted citations.*

Description

The testable core of the CLI (no `quit()`). Missing files error by default.

Usage

```
retraction_scan(files, on_missing = c("error", "skip"), ...)
```

Arguments

`files`	Character vector of file paths.
`on_missing`	`"error"` (default) or `"skip"` (record as a file error).
`...`	Passed to `check_file()`.

Value

A list: `result` (a `retraction_result`), `n_errors` (files that could not be scanned), and `errors` (a named list of error messages).

retraction_snapshot_parquet *Export the local snapshot to a Parquet file*

Description

Writes the current offline snapshot (built by `retraction_sync()`) to Parquet so it can be queried at scale with the arrow package, e.g. `arrow::open_dataset(path)`.

Usage

```
retraction_snapshot_parquet(path = NULL)
```

Arguments

`path`	Output path. Defaults to `snapshot.parquet` in the cache directory (`retraction_cache_dir()`).

Value

Invisibly, the path written. Requires the suggested arrow package.

Examples

```
## Not run:
retraction_sync()
p <- retraction_snapshot_parquet()
arrow::open_dataset(p)

## End(Not run)
```

retraction_sync *Download or update a local snapshot of the retraction corpus*

Description

The first call downloads the full corpus (sliced by retraction year to respect the export endpoint's row cap and rate limit). Later calls are incremental by default: only recent years are re-fetched and merged into the existing snapshot by record id, so new retractions are added without re-downloading everything. Use `force = TRUE` for a complete refresh.

Usage

```
retraction_sync(force = FALSE, incremental = TRUE, quiet = FALSE)
```

Arguments

force	Re-download the entire corpus, replacing any existing snapshot.
incremental	When a snapshot exists, fetch only recent years and merge (default). Ignored when force = TRUE or no snapshot exists.
quiet	Suppress progress messages.

Details

Once a snapshot exists, pass offline = TRUE to any check_*() function to match locally without the network.

Value

Invisibly, the snapshot data frame.

Examples

```
## Not run:
retraction_sync()                # first run: full download
retraction_sync()                # later: incremental update
retraction_sync(force = TRUE)    # occasional full refresh
check_bib("refs.bib", offline = TRUE)

## End(Not run)
```

---

retraction_watch_diff *Report references newly flagged since the saved baseline.*

Description

Re-checks the same bibliography and returns rows that are flagged now but were not at save time, matched by normalized identifier.

Usage

```
retraction_watch_diff(x, name)
```

Arguments

x	A freshly checked retraction_result for the same bibliography.
name	The identifier used when saving the baseline.

Value

A retraction_result with only the newly-flagged rows (zero rows if nothing changed).

retraction_watch_save *Save a checked result as a named watch baseline.*

Description

Save a checked result as a named watch baseline.

Usage

```
retraction_watch_save(x, name)
```

Arguments

x	A retraction_result.
name	A short identifier for this bibliography (e.g. "my-review").

Value

Invisibly, the path written.

snapshot_info *Report the local snapshot's data version and freshness*

Description

Which retraction-database version an offline check runs against, for reproducible checks.

Usage

```
snapshot_info()
```

Value

Invisibly, a list with path, records, synced_at, and newest_retraction; or NULL if no snapshot exists. Also prints a summary.

Examples

```
## Not run:
snapshot_info()

## End(Not run)
```

suggest_alternatives *Suggest alternatives to a retracted work*

Description

Given a retracted work's DOI, return the records the corpus links to it, such as a later correction or reinstatement, to help decide what (if anything) to cite instead.

Usage

```
suggest_alternatives(doi)
```

Arguments

doi	A DOI (of the retracted work).

Value

A tibble of related records (`record_id`, `title`, `nature`, `date`), or `NULL` if the DOI is unknown or has no related records.

Examples

```
suggest_alternatives("10.1016/S0140-6736(97)11096-0")
```

summary.retraction_result *Summarize a retraction result as a status tally.*

Description

Summarize a retraction result as a status tally.

Usage

```
## S3 method for class 'retraction_result'
summary(object, ...)
```

Arguments

object	A `retraction_result`.
...	Unused.

Value

A tibble with one row per status category and its count.

Index