Package: easyRef (via r-universe)

June 8, 2026

Type Package

Title Easy Reference Generation for R Packages

Version 0.1.0

Author Rasmus Rydbirk [aut, cre]

Maintainer Rasmus Rydbirk <rrydbirk@outlook.dk>

Description Generate citations and references for R packages from CRAN or Bioconductor. Supports RIS and BibTeX formats with automatic DOI retrieval from GitHub repositories and published papers. Includes command-line interface for batch processing.

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/rrydbirk/easyRef

BugReports https://github.com/rrydbirk/easyRef/issues

Depends R (>= 3.5.0)

Imports utils, xml2, rvest

Suggests BiocManager, stringr, testthat (>= 3.0.0)

RoxygenNote 7.3.2

Config/pak/sysreqs libicu-dev libxml2-dev libssl-dev

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

Date/Publication 2025-10-07 08:14:15 UTC

RemoteUrl https://github.com/rrydbirk/easyref

RemoteRef HEAD

RemoteSha f780ac95181b91c7f4a038baf7f89c233f839e1e

Contents

`authors_from_bibentry` *Extract authors from bibentry object*

Description

Extract authors from bibentry object

Usage

```
authors_from_bibentry(be)
```

Arguments

be	Bibentry object

Value

Character vector of author names

`bibtex_from_bibentry` *Convert bibentry to BibTeX string*

Description

Convert bibentry to BibTeX string

Usage

```
bibtex_from_bibentry(be, key_hint = NULL)
```

Arguments

`be`	Bibentry object
`key_hint`	Optional key hint (not used currently)

Value

BibTeX string or NULL if conversion fails

`bioc_meta_for` *Get Bioconductor metadata for a package without installing it*

Description

Get Bioconductor metadata for a package without installing it

Usage

```
bioc_meta_for(pkg)
```

Arguments

`pkg`	Package name

Value

List with package metadata or NULL if not found

clean_author_name *Clean author names by removing bracketed content*

Description

Clean author names by removing bracketed content

Usage

```
clean_author_name(x)
```

Arguments

x	Author name string

Value

Cleaned author name string

collect_for_package_internal *Collect package information and generate citations (internal)*

Description

Collect package information and generate citations (internal)

Usage

```
collect_for_package_internal(pkg, database = "auto", verbose = FALSE)
```

Arguments

pkg	Package name
database	Repository to search: "auto", "cran", or "bioconductor"
verbose	Logical. If TRUE, prints detailed information about each step

Value

List with package information and formatted citations

cran_meta_for *Get CRAN metadata for a package without installing it*

Description

Get CRAN metadata for a package without installing it

Usage

```
cran_meta_for(pkg)
```

Arguments

pkg	Package name

Value

List with package metadata or NULL if not found

createBibtex *Create BibTeX format citation for software*

Description

Creates a properly formatted BibTeX citation entry for software packages.

Usage

```
createBibtex(
  key,
  title,
  authors,
  year,
  url = NULL,
  version = NULL,
  filename = NULL,
  overwrite = TRUE
)
```

Arguments

key	BibTeX key for the entry
title	Software title
authors	Character vector of author names
year	Publication year
url	Software URL (optional)
version	Software version (optional)
filename	Output file path. If NULL, returns the BibTeX string without writing to file
overwrite	Allow overwriting existing files (default: TRUE)

Value

Character string with BibTeX formatted citation, or writes to file if filename provided

Examples

```
createBibtex(
  key = "ggplot2_2016",
  title = "ggplot2: Create Elegant Data Visualisations",
  authors = c("Wickham, Hadley"),
  year = "2016",
  url = "https://ggplot2.tidyverse.org",
  filename = tempfile() # Omit filename
)

# Write to file
createBibtex(
  key = "ggplot2_2016",
  title = "ggplot2: Create Elegant Data Visualisations",
  authors = c("Wickham, Hadley"),
  year = "2016",
  filename = tempfile() # Replace with e.g. "ggplot2.bib"
)
```

---

createBiocRef *Create reference citation for Bioconductor packages*

Description

Convenience function specifically for Bioconductor packages. Automatically detects Bioconductor packages and retrieves metadata from Bioconductor repositories.

Usage

```
createBiocRef(
  pkg,
  format = "ris",
  filename = NULL,
  overwrite = TRUE,
  verbose = FALSE,
  database = "auto"
)
```

Arguments

pkg	Character vector of Bioconductor package names to process
format	Output format: "ris", "bib", "bibtex", or "both" (default: "ris")
filename	Output file path. If NULL, creates a default filename based on package name(s). If no file extension is provided, one will be added based on the format.
overwrite	Allow overwriting existing files (default: TRUE)
verbose	Logical. If TRUE, prints detailed information about each step (default: FALSE)
database	Repository to search: "auto" (default), "cran", or "bioconductor". For Bioconductor packages, "bioconductor" is recommended.

Value

Invisible list of results with package information and formatted citations. Always writes to file.

Examples

```
# Generate citation for Bioconductor packages (requires BiocManager)
if (requireNamespace("BiocManager", quietly = TRUE)) {
  result <- createBiocRef("Biobase", filename = tempfile())

  # Generate citations for multiple Bioconductor packages
  bioc_packages <- c("Biobase", "limma", "edgeR")
 results <- createBiocRef(bioc_packages, format = "both", filename = tempfile()) # Omit filename

  # Write Bioconductor package citations to file (extension added automatically)
  createBiocRef("Biobase", filename = tempfile()) # Replace with e.g. "biobase_citation"

  # Verbose output for Bioconductor packages (writes to default file)
 createBiocRef("Biobase", verbose = TRUE, filename = tempfile()) # Omit filename = tempfile()

  # Force search in Bioconductor repository
  createBiocRef("Biobase", database = "bioconductor", verbose = TRUE,
  filename = tempfile()) # Omit filename = tempfile()
}
```

createRef *Create reference citation for R packages*

Description

This is the main function to collect information about R packages and generate citations in RIS or BibTeX format. Supports both CRAN and Bioconductor packages with automatic DOI retrieval from GitHub repositories and published papers.

Usage

```
createRef(
  pkg,
  format = "ris",
  filename = NULL,
  overwrite = TRUE,
  verbose = FALSE,
  database = "auto"
)
```

Arguments

pkg	Character vector of package names to process
format	Output format: "ris", "bib", "bibtex", or "both" (default: "ris")
filename	Output file path. If NULL, creates a default filename based on package name(s). If no file extension is provided, one will be added based on the format.
overwrite	Allow overwriting existing files (default: TRUE)
verbose	Logical. If TRUE, prints detailed information about each step (default: FALSE)
database	Repository to search: "auto" (default), "cran", or "bioconductor". "auto" automatically detects the repository, "cran" searches only CRAN, "bioconductor" searches only Bioconductor.

Value

Invisible list of results with package information and formatted citations. Always writes to file.

Examples

```
# Generate RIS citation for a CRAN package
result <- createRef("ggplot2", filename = tempfile()) # Omit filename

# Generate citation for a Bioconductor package (requires BiocManager)
if (requireNamespace("BiocManager", quietly = TRUE)) {
  result <- createRef("Biobase", filename = tempfile())
}
```

```
# Generate both RIS and BibTeX for multiple packages
results <- createRef(c("ggplot2", "dplyr"), format = "both", filename = tempfile())

# Write to file (extension will be added automatically)
createRef("ggplot2", filename = tempfile()) # Replace with e.g. "ggplot2_citation"

# Verbose output showing each step (writes to default file)
createRef("ggplot2", verbose = TRUE, filename = tempfile()) # Omit filename

# Force search in specific repository
createRef("ggplot2", database = "cran", filename = tempfile()) # Omit filename
```

`createRis` *Create RIS format citation for software*

Description

Creates a properly formatted RIS citation entry for software packages.

Usage

```
createRis(
  title,
  authors,
  year,
  url = NULL,
  version = NULL,
  doi = NULL,
  notes = NULL,
  publisher = "Comprehensive R Archive Network (CRAN)",
  filename = NULL,
  overwrite = TRUE
)
```

Arguments

`title`	Software title
`authors`	Character vector of author names
`year`	Publication year
`url`	Software URL (optional)
`version`	Software version (optional)
`doi`	DOI (optional)
`notes`	Additional notes (optional)
`publisher`	Publisher name (default: "Comprehensive R Archive Network (CRAN)")
`filename`	Output file path. If NULL, returns the RIS string without writing to file
`overwrite`	Allow overwriting existing files (default: TRUE)

Value

Character string with RIS formatted citation, or writes to file if filename provided

Examples

```
createRis(
  title = "ggplot2: Create Elegant Data Visualisations",
  authors = c("Wickham, Hadley"),
  year = "2016",
  url = "https://ggplot2.tidyverse.org",
  filename = tempfile() # Omit filename
)

# Write to file
createRis(
  title = "ggplot2: Create Elegant Data Visualisations",
  authors = c("Wickham, Hadley"),
  year = "2016",
  filename = tempfile() # Replace with e.g. "ggplot2.ris"
)
```

---

emit_outputs_internal *Emit citation outputs in specified formats (internal)*

Description

Emit citation outputs in specified formats (internal)

Usage

```
emit_outputs_internal(results, format, out, split, overwrite)
```

Arguments

results	List of package citation results
format	Output format: "ris", "bib", "bibtex", or "both"
out	Output file path or directory
split	Write one file per package when multiple packages or format="both"
overwrite	Allow overwriting existing files

Value

Invisible TRUE

`ensure_dir` *Ensure directory exists*

Description

Ensure directory exists

Usage

```
ensure_dir(path)
```

Arguments

`path`	Directory path

`extract_doi_from_reference_sections` *Extract DOI from reference sections in README content*

Description

Extract DOI from reference sections in README content

Usage

```
extract_doi_from_reference_sections(readme_content)
```

Arguments

`readme_content`	Character vector of README lines

Value

DOI string or NULL if not found

get_bioc_author_from_web

Get author information from BioC package website

Description

Get author information from BioC package website

Usage

```
get_bioc_author_from_web(pkg)
```

Arguments

pkg	Package name

Value

Character string with author information or NULL if not available

get_cran_author_from_web

Get author information from CRAN package website

Description

Get author information from CRAN package website

Usage

```
get_cran_author_from_web(pkg)
```

Arguments

pkg	Package name

Value

Character string with author information or NULL if not available

get_cran_title_from_web *Get title information from CRAN package website*

Description

Get title information from CRAN package website

Usage

```
get_cran_title_from_web(pkg)
```

Arguments

pkg	Package name

Value

Character string with title information or NULL if not available

get_doi_from_github *Get DOI from GitHub repository*

Description

Get DOI from GitHub repository

Usage

```
get_doi_from_github(repo)
```

Arguments

repo	GitHub repository in format "owner/repo"

Value

DOI string or NULL if not found

get_doi_from_package *Get DOI from package citation or metadata*

Description

Get DOI from package citation or metadata

Usage

```
get_doi_from_package(pkg)
```

Arguments

pkg	Package name

Value

DOI string or NULL if not found

get_github_info *Get GitHub repository information and DOI*

Description

Get GitHub repository information and DOI

Usage

```
get_github_info(pkg, url = NULL)
```

Arguments

pkg	Package name
url	Package URL (optional)

Value

List with GitHub info and DOI or NULL if not found

get_repository_doi *Generate CRAN/Bioconductor DOI for a package*

Description

Generate CRAN/Bioconductor DOI for a package

Usage

```
get_repository_doi(pkg, repository)
```

Arguments

pkg	Package name
repository	Repository name ("CRAN" or "Bioconductor")

Value

DOI string or NULL if not found

is_bioc_package *Check if a package is from Bioconductor*

Description

Check if a package is from Bioconductor

Usage

```
is_bioc_package(pkg)
```

Arguments

pkg	Package name

Value

Logical indicating if package is from Bioconductor

is_installed *Check if a package is installed*

Description

Check if a package is installed

Usage

```
is_installed(pkg)
```

Arguments

pkg	Package name to check

Value

Logical indicating if package is installed

make_ris_for_software_internal *Generate RIS format citation for software (internal)*

Description

Generate RIS format citation for software (internal)

Usage

```
make_ris_for_software_internal(
  title,
  authors,
  year,
  url = NULL,
  version = NULL,
  doi = NULL,
  notes = NULL,
  publisher = "Comprehensive R Archive Network (CRAN)"
)
```

Arguments

title	Software title
authors	Character vector of author names
year	Publication year
url	Software URL (optional)
version	Software version (optional)
doi	DOI (optional)
notes	Additional notes (optional)
publisher	Publisher name (default: "Comprehensive R Archive Network (CRAN)")

Value

Character string with RIS formatted citation

---

normalize_key *Normalize string for use as BibTeX key*

Description

Normalize string for use as BibTeX key

Usage

```
normalize_key(x)
```

Arguments

x	String to normalize

Value

Normalized string

package_exists_on_repos *Check if a package exists on CRAN or Bioconductor repositories*

Description

Check if a package exists on CRAN or Bioconductor repositories

Usage

```
package_exists_on_repos(pkg)
```

Arguments

pkg	Package name

Value

Logical indicating if package exists on any repository

parse_author_text *Parse author text from DESCRIPTION file*

Description

Parse author text from DESCRIPTION file

Usage

```
parse_author_text(x)
```

Arguments

x	Author text string

Value

Character vector of author names

person_to_string *Convert person object to string format*

Description

Convert person object to string format

Usage

```
person_to_string(p)
```

Arguments

p	Person object

Value

Character string with formatted name

ris_sanitize *Sanitize text for RIS format (single-line)*

Description

Sanitize text for RIS format (single-line)

Usage

```
ris_sanitize(x)
```

Arguments

x	Text to sanitize

Value

Sanitized text

safely_get *Safely get element from list with default value*

Description

Safely get element from list with default value

Usage

```
safely_get(x, name, default = NULL)
```

Arguments

x	List or object to access
name	Name of element to get
default	Default value if element doesn't exist

Value

Element value or default

synthesize_bibtex_internal *Generate BibTeX format citation for software (internal)*

Description

Generate BibTeX format citation for software (internal)

Usage

```
synthesize_bibtex_internal(
  key,
  title,
  authors,
  year,
  url = NULL,
  version = NULL,
  doi = NULL
)
```

Arguments

key	BibTeX key for the entry
title	Software title
authors	Character vector of author names
year	Publication year
url	Software URL (optional)
version	Software version (optional)
doi	DOI (optional)

Value

Character string with BibTeX formatted citation

---

try_case_insensitive_search *Try case-insensitive search for package name*

Description

Try case-insensitive search for package name

Usage

```
try_case_insensitive_search(pkg, database, verbose = FALSE)
```

Arguments

pkg	Package name to search for
database	Repository to search: "auto", "cran", or "bioconductor"
verbose	Logical. If TRUE, prints detailed information

Value

List with origin and correct_name if found, NULL otherwise

write_text *Write text to file with overwrite control*

Description

Write text to file with overwrite control

Usage

```
write_text(path, text, overwrite = FALSE)
```

Arguments

path	File path
text	Text to write
overwrite	Whether to allow overwriting existing files

year_from_bibentry *Extract year from bibentry object*

Description

Extract year from bibentry object

Usage

```
year_from_bibentry(be)
```

Arguments

be	Bibentry object

Value

Character string with year

Index