

Package: journalabbr (via r-universe)

October 10, 2026

Type Package

Title Journal Abbreviations for BibTeX Documents

Version 0.4.4

Description Since the reference management software (such as 'Zotero', 'Mendeley') exports Bib file journal abbreviation is not detailed enough, the 'journalabbr' package only abbreviates the journal field of Bib file, and then outputs a new Bib file for generating reference format with journal abbreviation on other software (such as 'texstudio'). The abbreviation table is from 'JabRef'. At the same time, 'Shiny' application is provided to generate 'thebibliography', a reference format that can be directly used for latex paper writing based on 'Rmd' files.

License GPL (>= 3)

URL <https://github.com/zoushucai/journalabbr>

BugReports <https://github.com/zoushucai/journalabbr/issues>

Encoding UTF-8

Depends R (>= 4.3.0)

Imports data.table (>= 1.14.0), stringr (>= 1.4.0), purrr (>= 1.0.0),
httr (>= 1.4.0), shiny (>= 1.7.0), stringi (>= 1.7.0), rlang(>= 1.0.0)

Suggests testthat (>= 3.0.0), rmarkdown, knitr, tinytex, covr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/testthat/edition 3

Date 2024-02-21

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev zlib1g-dev

Repository <https://zoushucai.r-universe.dev>

Date/Publication 2024-12-05 07:17:34 UTC

RemoteUrl <https://github.com/zoushucai/journalabbr>

RemoteRef HEAD

RemoteSha 666a5425e2103bdd2a28de665972ca2713533789

Contents

bib2bib	2
get_abbrtable	3
get_delcolumns	4
read_bib2dt	5
replace_field	6
run_example	8
single_add_space	8
single_cites_convert	9
single_cites_extract	11
single_replace_author	12
single_replace_order	14
write_dt2bib	15
Index	17

bib2bib	<i>Load .bib, Process .bib and Output .bib File.</i>
---------	--

Description

The bib2bib function reads a BibTeX file, replaces the journal field using a custom or system-provided abbreviation table, optionally applies a function to the author field, and writes the result back to a new BibTeX file.

Usage

```
bib2bib(  
  file,  
  out.file,  
  user_table = NULL,  
  use_sys_table = TRUE,  
  fun = NULL,  
  ...  
)
```

Arguments

file	A string representing the path to the input .bib file to be processed.
out.file	A string representing the path where the output .bib file should be saved.
user_table	A user-provided abbreviation table. It can be a file path (string), a data.frame, or a data.table. The table must contain columns journal_lower and journal_abbr. Default is NULL.
use_sys_table	Logical. Whether to include the system abbreviation table (abbrtable_sys) in the merged result. Default is TRUE.

fun	A function (optional) to apply to the "author" field in the .bib file. If provided, this function will be applied to the "author" field
...	Additional arguments passed to the custom function fun, if provided.

Details

The function works in the following steps:

1. Reads a custom journal abbreviation table from a CSV file (if provided).
2. Reads the input BibTeX file and converts it into a `data.table`.
3. Replaces the JOURNAL field with abbreviations from the provided or system abbreviation table.
4. Optionally applies the provided function to the AUTHOR field.
5. Writes the modified data back to a new BibTeX file.

Value

None. The function writes the processed data to the specified output file.

Examples

```
bib_file <- system.file("extdata", "testfile_2.bib", package = "journalabbr", mustWork = TRUE)
csv_file <- system.file("extdata", "myabbr.csv", package = "journalabbr", mustWork = TRUE)
output_file <- tempfile(fileext = ".bib")

dt <- bib2bib(file = bib_file,
  out.file = output_file,
  user_table = csv_file,
  use_sys_table = TRUE,
  fun = function(x) {
    gsub(" and ", " & ", x, perl = TRUE, ignore.case = TRUE)
  })
```

get_abbrrtable	<i>Create an abbreviation table by merging user-provided and system tables</i>
----------------	--

Description

This function processes a user-provided abbreviation table and optionally merges it with a system abbreviation table. The user table can be provided as a file path, `data.frame`, or `data.table`. The system table is included by default but can be excluded.

Usage

```
get_abbrrtable(user_table = NULL, use_sys_table = TRUE)
```

Arguments

user_table	A user-provided abbreviation table. It can be a file path (string), a data.frame, or a data.table. The table must contain columns journal_lower and journal_abbr. Default is NULL.
use_sys_table	Logical. Whether to include the system abbreviation table (abbrtable_sys) in the merged result. Default is TRUE.

Details

The function first processes the user-provided table by checking its type:

- If it's a file path, it reads the file using `data.table::fread`.
- If it's a data.frame, it is converted to a data.table.
- If it's already a data.table, no conversion is needed.

After processing the user table, the system table (`abbrtable_sys`) is optionally included in the final merged table. Duplicate entries based on `journal_lower` are removed.

Value

A data.table containing the merged abbreviation table, with any duplicates (by `journal_lower`) removed. The resulting table contains two columns: `journal_lower` and `journal_abbr`.

Examples

```
# Example 1: Provide a user table as a CSV file path and merge with system table
user_table_path <- system.file("extdata", "myabbr.csv", package = "journalabbr", mustWork = TRUE)
abbr_table <- get_abbrtable(user_table = user_table_path, use_sys_table = TRUE)

# Example 2: Provide a user table as a data.frame
user_df <- data.frame(journal_lower = c("example journal", "sample journal"),
                     journal_abbr = c("EJ", "SJ"))
abbr_table <- get_abbrtable(user_table = user_df, use_sys_table = FALSE)

# Example 3: Use only the system table
abbr_table <- get_abbrtable(use_sys_table = TRUE)

abbr_table <- get_abbrtable(user_table=NULL, use_sys_table = FALSE)
```

get_delcolumns	<i>Get default columns to delete</i>
----------------	--------------------------------------

Description

Returns the default list of columns that will be deleted.

Usage

```
get_delcolumns()
```

Value

A character vector containing the default column names to be excluded.

read_bib2dt	<i>Parse a BibTeX file to a data.table.</i>
-------------	---

Description

The BibTeX file is read, parsed, tidied and written to a data.table.

Usage

```
read_bib2dt(file, encoding = "UTF-8")
```

Arguments

file	character, path or URL to a bib file.
encoding	character, encoding to be passed to base::readLines. Default is "UTF-8".

Details

Read, parse and collate bibtex file to form a data.table. Different BIB may produce different data.table columns.

Value

A data.table.

Author(s)

ShuCai Zou

See Also

[readLines](#) for more details on reading text files.

Examples

```
# Read from .bib file:
require(journalabbr)
require(purrr)
require(data.table)
file1 <- system.file("extdata", "testfile_1.bib", package = "journalabbr", mustWork = TRUE)
dt1 <- read_bib2dt(file1)
colnames(dt1)

file2 <- system.file("extdata", "testfile_2.bib", package = "journalabbr", mustWork = TRUE)
dt2 <- read_bib2dt(file2)
```

```

colnames(dt2)

# A valid BibTeX entry as a character vector
item1 <- c(
  "@Article{switalski2003general,",
  "author = {Switalski, Zbigniew}," ,
  "journal = {Fuzzy Sets and Systems}," ,
  "title = {General transitivity conditions for fuzzy reciprocal preference matrices}," ,
  "year = {2003}," ,
  "number = {1}," ,
  "pages = {85--100}," ,
  "volume = {137}," ,
  "publisher = {Elsevier}," ,
  "}"
)

item2 <- c(
  "@Article{switalski2003general,",
  "author = {Switalski, Zbigniew}," ,
  "journal = {Fuzzy Sets and Systems}," ,
  "title = {General transitivity conditions for fuzzy reciprocal preference matrices}," ,
  "year = {2003}}"
)

item3 <- c(
  "@Article{switalski2003general,",
  "author = {Switalski, Zbigniew}," ,
  "journal = {Fuzzy Sets and Systems}," ,
  "title = {General transitivity conditions for fuzzy reciprocal preference matrices}," ,
  "year = {2003}}"
)

item = list(item1, item2, item3)

# Check validity of the entries (inner function)
map(item, journalabbr::checkitem_valid)
map(item, journalabbr::extract_fields)
rbindlist(map(item, journalabbr::extract_fields), fill=TRUE, use.names = TRUE)
rbindlist(map(item, journalabbr::extract_fields, check=TRUE), fill=TRUE, use.names = TRUE)

```

replace_field

Replace or process a specified field in a data.table

Description

This function processes a specified field in a `data.table` by using a journal abbreviation table (either user-specified, system-provided, or both) to abbreviate journal names in the `oldfield`. Additionally, a custom function can be applied to the processed field if specified.

Usage

```
replace_field(
  dt,
  oldfield,
  newfield,
  user_table = NULL,
  use_sys_table = TRUE,
  fun = NULL,
  ...
)
```

Arguments

<code>dt</code>	A <code>data.table</code> . The input data table that contains the field to be processed.
<code>oldfield</code>	A character string. The name of the field to be processed, typically in uppercase (e.g., "JOURNAL"). Must be a valid column name in <code>dt</code> .
<code>newfield</code>	A character string. The name of the new field where the processed result will be stored. If this field does not exist in <code>dt</code> , it will be created.
<code>user_table</code>	A <code>data.table</code> . Optional. A user-provided journal abbreviation table with at least two columns: <code>journal_lower</code> and <code>journal_abbr</code> . Defaults to <code>NULL</code> . If provided, it will be merged with the system abbreviation table if <code>use_sys_table = TRUE</code> .
<code>use_sys_table</code>	A logical. Whether to use the system-provided journal abbreviation table. Defaults to <code>TRUE</code> . If <code>TRUE</code> , the system abbreviation table is used alongside the user-provided one.
<code>fun</code>	A function. Optional. A custom function to apply to the processed field after abbreviation (if applicable). Defaults to <code>NULL</code> . The function should accept a column from <code>dt</code> as its first argument and return the processed values.
<code>...</code>	Additional arguments passed to the custom function <code>fun</code> , if provided.

Details

If the `oldfield` is "JOURNAL", the function will attempt to apply journal abbreviations using the provided abbreviation table(s). The abbreviation process involves converting the journal names to lowercase and removing excess whitespace before matching against the abbreviation table.

The `user_table` and `use_sys_table` parameters allow flexibility in choosing which abbreviation tables to use. If both are used, they will be merged, and duplicates will be removed.

If a custom function is provided via `fun`, it will be applied to the processed field after any abbreviations.

Value

The function returns the modified `data.table` with the processed field stored in `newfield`.

Examples

```
csvpath <- system.file("extdata", "myabbr.csv", package = "journalabbr", mustWork = TRUE)
file <- system.file("extdata", "testfile_2.bib", package = "journalabbr", mustWork = TRUE)
dt <- read_bib2dt(file)

abbrtable_user = get_abbrtable(user_table = csvpath, use_sys_table = TRUE)
print(head(abbrtable_user))
dm1 = replace_field(dt,
  oldfield = "JOURNAL",
  newfield = "JOURNAL",
  user_table = csvpath,
  use_sys_table = TRUE,
  fun = NULL)

myauthor = function(x){ gsub(" and ", " & ", x, perl = TRUE, ignore.case = TRUE) }
dm2 = replace_field(dm1,
  oldfield = "AUTHOR",
  newfield = "AUTHOR",
  user_table = NULL,
  use_sys_table = FALSE,
  fun = myauthor)
print(head(dt)[, c("JOURNAL", "AUTHOR"), with = FALSE])
print(head(dm1)[, c("JOURNAL", "AUTHOR"), with = FALSE])
print(head(dm2)[, c("JOURNAL", "AUTHOR"), with = FALSE])
```

run_example

The Shiny Program for the 'journalabbr' Package

Description

The Shiny Program for the 'journalabbr' Package

Usage

```
run_example()
```

single_add_space

Clean up and standardize spacing in a given text

Description

This function processes a given string by cleaning up extra spaces, handling punctuation, and adjusting mathematical symbols according to specific formatting rules.

This function takes a vector of LaTeX-formatted strings and applies the single_add_space function to each string to standardize spacing throughout the document.

Usage

```
single_add_space(text)

tex_funs(tex, fun = single_add_space, ...)
```

Arguments

<code>text</code>	A single character string (input text) that needs to be processed.
<code>tex</code>	A vector of strings (LaTeX document or text) to be processed.
<code>fun</code>	The function to be applied to each string in the vector (default is <code>single_add_space</code>).
<code>...</code>	Additional arguments to be passed to the function.

Details

see function `single_add_space` source code for details.

Value

A cleaned-up string where multiple spaces are reduced to one, unnecessary spaces are removed, and spaces are appropriately added around certain punctuation and mathematical symbols.

A vector of processed strings where unnecessary spaces have been removed or adjusted.

Examples

```
tex = c("This is a citation,and.",
        "This is a citation2 $x ^ 2+y^ 2 = z^2.$ and this is equ",
        "this is no , citation")
tex = tex_funs(tex)
print(tex)
```

`single_cites_convert` *Convert citations between LaTeX and Markdown formats*

Description

This function converts citations in a given text between LaTeX and Markdown formats. It can handle multiple LaTeX citation styles and produces the appropriate format based on the specified direction. Note that the input text must have a length of 1.

the `cites_convert` function applies `single_cites_convert` to each element of the input vector.

Usage

```
single_cites_convert(
  text,
  pattern = NULL,
  tex2md = TRUE,
  latex_prefix = c("cite", "upcite", "citep", "citet")
)

cites_convert(x, ...)
```

Arguments

<code>text</code>	A character string containing the text with citations, Note that the input text must have a length of 1.
<code>pattern</code>	A regular expression pattern for matching citations. If NULL, defaults to a pattern based on the citation direction.
<code>tex2md</code>	Logical; if TRUE, converts from LaTeX to Markdown; if FALSE, converts from Markdown to LaTeX.
<code>latex_prefix</code>	A character vector of LaTeX citation prefixes to use. Defaults to <code>c("cite", "upcite", "citep", "citet")</code> .
<code>x</code>	A character vector containing texts with citations.
<code>...</code>	Additional arguments passed to <code>single_cites_convert</code> .

Value

the `single_cites_convert` function return a character string with citations converted to the specified format.

the `cites_convert` function return a character vector with citations converted to the specified format.

Examples

```
# Example text containing various LaTeX citations
# Convert the LaTeX citations in the text to Markdown format
text = "This is a citation \\upcite{key1,key2} and \\citep{key3-a }.
       paper \\cite{key3-b, cky3-c}, paper \\citet{key4-a, cky4-b}"
converted_md = single_cites_convert(text, tex2md = TRUE)
print(converted_md)
```

```
# Example text containing Markdown citations
# Convert the Markdown citations back to LaTeX format

text = "This is a citation [@key1; @key2] and [@key3-a].
       paper [@key3-b; @cky3-c], paper [@key4-a; @cky4-b]"
# default prefix "cite"
converted_latex = single_cites_convert(text, tex2md = FALSE)
```

```

print(converted_latex)

# Convert Markdown citations to LaTeX using a custom prefix
custom_prefix = single_cites_convert(text, tex2md = FALSE, latex_prefix = "upcite")
print(custom_prefix)

# A vector of texts with Markdown citations
tex = c("This is a citation [@key1; @key2] and [@key3-a].",
        "This is a citation2 [@key1; @key2] and [@key3-a].",
        "this is no citation",
        "",
        "This is a citation3 [@key1; @key2] and [@key3-a].")

# vector version
tex = cites_convert(tex, tex2md = FALSE)
print(tex)

# Another vector of texts containing LaTeX citations
tex2= c("This is a citation \\upcite{key1,key2} and \\citep{key3-a }",
        "This is a citation \\citett{key1,key2} and \\citett{key3-a }",
        "this is no citation",
        "",
        "This is a citation \\upcite{key1,key2} and \\cite{key3-a }")
# Convert the vector of LaTeX citations to Markdown format
tex2 = cites_convert(tex2, tex2md = TRUE)
print(tex2)

```

single_cites_extract *Extract Single Citation Keys from LaTeX Text*

Description

single_cites_extract function extracts all citation keys from a LaTeX string based on specific citation commands.

cites_extract function extracts all unique citation keys from a LaTeX document.

Usage

```
single_cites_extract(text)
```

```
cites_extract(tex)
```

Arguments

text	A character vector containing LaTeX content.
tex	A character vector where each element represents a line or section of LaTeX content.

Details

The function identifies citation commands such as `cite`, `upcite`, `citep`, and `citet`. It extracts the content within braces following these commands, and splits multiple keys separated by commas.

This function uses `single_cites_extract` to extract citation keys from each line or section of the LaTeX document. It removes duplicates and trims unnecessary whitespace.

Value

A character vector of citation keys, or NULL if no citations are found.

A unique character vector of citation keys.

Examples

```
tex= c("This is a citation \\upcite{key1,key2} and \\citep{key3-a }",
      "This is a citation \\citet{key1,key2} and \\citet{key3-a }",
      "this is no citation",
      "",
      "This is a citation \\upcite{key1,key4} and \\cite{key3-a }")
cites_extract(tex)
```

single_replace_author *Add Parentheses to Citation Commands in LaTeX Text*

Description

`single_replace_author` function modifies LaTeX citation commands to add parentheses around them, transforming commands such as `\cite{...}` into `(\cite{...})`.

`cites_replace_author` function processes a LaTeX document to transform all numeric citations into author-year style by wrapping citation commands in parentheses.

`single_replace_number` function transforms LaTeX citation commands from an author-year style (e.g., `(\cite{...})`) back to their original form (e.g., `\cite{...}`).

`cites_replace_number` function processes a LaTeX document to remove parentheses around citation commands, converting citations from an author-year style back to numeric style.

Usage

```

single_replace_author(
  text,
  latex_prefix = c("cite", "upcite", "citep", "citet")
)

cites_replace_author(tex)

single_replace_number(
  text,
  latex_prefix = c("cite", "upcite", "citep", "citet")
)

cites_replace_number(tex)

```

Arguments

<code>text</code>	A single string of LaTeX content.
<code>latex_prefix</code>	A character vector of LaTeX citation commands to be matched (default: <code>c("cite", "upcite", "citep", "citet")</code>).
<code>tex</code>	A character vector where each element represents a line or section of LaTeX content.

Details

The function searches for LaTeX citation commands using regular expressions. For each matched command, it appends and prepends parentheses to create the author-year citation style. Excessive whitespace in the resulting text is removed to ensure formatting consistency.

If no matching citation commands are found, the input text is returned unchanged.

This function applies `single_replace_author` to each line or section of the LaTeX document, ensuring that citation commands such as `\cite{...}` and `\citet{...}` are converted into `(\cite{...})` and `(\citet{...})`, respectively.

The function identifies citation commands enclosed in parentheses and removes the parentheses, restoring the original citation format. Excessive whitespace in the resulting text is removed for formatting consistency.

If no matching citation commands are found, the input text is returned unchanged.

This function applies `single_replace_number` to each line or section of the LaTeX document, ensuring that citations like `(\cite{...})` are transformed back into `\cite{...}`.

Value

A string where all matched citation commands are wrapped in parentheses.

A character vector with LaTeX content where all citation commands are converted to author-year style.

A string where parentheses around matched citation commands are removed.

A character vector with LaTeX content where parentheses around citation commands are removed.

See Also[single_replace_author](#)[single_replace_number](#)**Examples**

```
tex= c("This is a citation \\upcite{key1,key2} and \\citep{key3-a }",
      "This is a citation \\citet{key1,key2} and \\citet{key3-a }",
      "this is no citation",
      "",
      "This is a citation \\upcite{key1,key4} and \\cite{key3-a }")
```

```
tex = cites_replace_author(tex)
```

```
print(tex)
```

```
tex2 = cites_replace_number(tex)
print(tex2)
```

single_replace_order *Replace Citation Order in LaTeX Text*

Description

single_replace_order function modifies LaTeX citation commands to reorder citation keys based on a provided order.

cites_replace_order function processes a LaTeX document to reorder all citation keys within citation commands based on their first appearance in the document.

Usage

```
single_replace_order(
  text,
  citesall = NULL,
  latex_prefix = c("cite", "upcite", "citep", "citet")
)
```

```
cites_replace_order(tex, citesall = NULL)
```

Arguments

text	A single string of LaTeX content.
citesall	A character vector containing all citation keys in the desired order. If not provided, default applies cites_extract.

latex_prefix	A character vector of LaTeX citation commands to be matched (default: c("cite", "upcite", "citep", "citet")).
tex	A character vector where each element represents a line or section of LaTeX content.

Details

The function identifies LaTeX citation commands and reorders their arguments (e.g., \cite{key1,key2}) to match the order specified in citesall. If a citation contains only one key, it remains unchanged.

Matched citation keys not found in citesall are excluded from the reordered result.

The function extracts all citation keys using cites_extract, determines their order based on first occurrence, and applies single_replace_order to reorder keys in each citation command.

Value

A string with citation keys reordered within the specified citation commands.

A character vector with updated LaTeX content where citation keys in commands are reordered.

See Also

[single_replace_order](#), [cites_extract](#)

Examples

```
tex= c("This is a citation \\upcite{key1,key2} and \\citep{key3-a }",
      "This is a citation \\citet{key1,key2} and \\citet{key3-a }",
      "this is no citation",
      "",
      "This is a citation \\upcite{key1,key4} and \\cite{key3-a }")
citesall = cites_extract(tex)

citesall = c("key2", "key1", "key3-a", "key4")

cites_replace_order(tex, citesall)
```

write_dt2bib

Export a BibTeX data.table to a .bib file.

Description

The BibTeX data.table is written to a .bib file.

Usage

```
write_dt2bib(  
  dt,  
  file = tempfile(fileext = ".bib"),  
  columns = get_delcolumns(),  
  indent = "\t"  
)
```

Arguments

dt	data.table, in the format as returned by read_bib2dt .
file	character, file path to write the .bib file.
columns	character vector, names of the columns to exclude from the output. a default list of columns will be used from get_delcolumns .
indent	character, the indentation to use for the .bib file. Default is "\t"

Value

file as a character string, invisibly.

Examples

```
# Read from .bib file:  
require(journalabbr)  
file <- system.file("extdata", "testfile_2.bib", package = "journalabbr", mustWork = TRUE)  
bib <- read_bib2dt(file)  
  
# Write to .bib file:  
write_dt2bib(bib, file = tempfile(fileext = ".bib"))
```

Index

bib2bib, [2](#)

cites_convert (single_cites_convert), [9](#)
cites_extract, [15](#)
cites_extract (single_cites_extract), [11](#)
cites_replace_author
 (single_replace_author), [12](#)
cites_replace_number
 (single_replace_author), [12](#)
cites_replace_order
 (single_replace_order), [14](#)

get_abbrtable, [3](#)
get_delcolumns, [4](#), [16](#)

read_bib2dt, [5](#), [16](#)
readLines, [5](#)
replace_field, [6](#)
run_example, [8](#)

single_add_space, [8](#)
single_cites_convert, [9](#)
single_cites_extract, [11](#)
single_replace_author, [12](#), [14](#)
single_replace_number, [14](#)
single_replace_number
 (single_replace_author), [12](#)
single_replace_order, [14](#), [15](#)

tex_funs (single_add_space), [8](#)

write_dt2bib, [15](#)