

Package ‘authoritative’

July 22, 2025

Title Parse and Deduplicate Author Names

Version 0.2.0

Description Utilities to parse authors fields from DESCRIPTION files and general purpose functions to deduplicate names in database, beyond the specific case of R package authors.

License MIT + file LICENSE

URL <https://github.com/Bisaloo/authoritative>,
<https://hugogruson.fr/authoritative/>

BugReports <https://github.com/Bisaloo/authoritative/issues>

Depends R (>= 4.1.0)

Imports stringi, utils

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/Needs/website epiverse-trace/epiversetheme, tidyverse, igraph,
netUtils

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.2

Config/Needs/build moodymudskipper/devtag

NeedsCompilation no

Author Hugo Gruson [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-4094-1476>>),
Chris Hartgerink [rev] (ORCID: <<https://orcid.org/0000-0003-1050-6809>>),
data.org [fnd] (until version 0.2.0 included)

Maintainer Hugo Gruson <hugo.gruson+R@normalesup.org>

Repository CRAN

Date/Publication 2025-06-24 07:50:11 UTC

Contents

cran_epidemiology_packages	2
expand_names	2
invert_names	3
parse_authors	4
parse_authors_r	5

Index	7
--------------	----------

cran_epidemiology_packages	<i>A data.frame of historical metadata from CRAN packages epidemiology.</i>
----------------------------	---

Description

A data.frame of historical metadata from CRAN packages epidemiology.

Usage

```
cran_epidemiology_packages
```

Format

A data.frame with 5 variables:

Package package name

Version package version

Authors@R authors as listed in the Authors@R field from the DESCRIPTION file

Author authors as listed in the Author field from the DESCRIPTION file

Maintainer package maintainer

expand_names	<i>Expand names from abbreviated forms or initials</i>
--------------	--

Description

Expand names from abbreviated forms or initials

Usage

```
expand_names(short, expanded)
```

Arguments

short A character vector of potentially abbreviated names
 expanded A character vector of potentially expanded names

Details

When you have a list of abbreviated and non-abbreviated names and you want to deduplicate them, this function can be used as `expand_names(x, x)`, which will return the most expanded version available in `x` for each name

Value

A character vector with the same length as `short`

Examples

```
expand_names(
  c("W A Mozart", "Wolfgang Mozart", "Wolfgang A Mozart"),
  "Wolfgang Amadeus Mozart"
)

# Real-case application example
# Deduplicate names in list, as described in "details"
epi_pkg_authors <- cran_epidemiology_packages |>
  subset(!is.na(`Authors@R`), `Authors@R`, drop = TRUE) |>
  parse_authors_r() |>
  # Drop email, role, ORCID and format as string rather than person object
  lapply(function(x) format(x, include = c("given", "family"))) |>
  unlist()

# With all duplicates
length(unique(epi_pkg_authors))

# Deduplicate
epi_pkg_authors_normalized <- expand_names(epi_pkg_authors, epi_pkg_authors)

length(unique(epi_pkg_authors_normalized))
```

invert_names	<i>Invert 'LastName FirstName' to 'FirstName LastName' (or the reverse)</i>
--------------	---

Description

Invert 'LastName FirstName' to 'FirstName LastName' (or the reverse)

Usage

```
invert_names(names, correct_names)
```

Arguments

```
names          A character vector of potentially inverted names
correct_names  A character vector of correct names
```

Details

When you have a list `x` of mixed 'First Last' and 'Last First' names, but no source of truth and you want to deduplicate them, this function can be used as `expand_names(x, x)`, which will return the most common version available in `x` for each name.

Value

A character vector with the same length as `names`

Examples

```
invert_names(
  c("Wolfgang Mozart", "Mozart Wolfgang"),
  "Wolfgang Mozart"
)

# Real-case application example
# Deduplicate names in list, as described in "details"
epi_pkg_authors <- cran_epidemiology_packages |>
  subset(!is.na(`Authors@R`), `Authors@R`, drop = TRUE) |>
  parse_authors_r() |>
  # Drop email, role, ORCID and format as string rather than person object
  lapply(function(x) format(x, include = c("given", "family"))) |>
  unlist()

# With all duplicates
length(unique(epi_pkg_authors))

# Deduplicate
epi_pkg_authors_normalized <- invert_names(epi_pkg_authors, epi_pkg_authors)

length(unique(epi_pkg_authors_normalized))
```

parse_authors

Parse the Author field from a DESCRIPTION file

Description

Parse the Author field from a DESCRIPTION file into a person object

Usage

```
parse_authors(author_string)
```

Arguments

```
author_string  A character containing the Author or Maintainer field from a DESCRIPTION file
```

Value

A character vector, or a list of character vectors of length equals to the length of author_string

Examples

```
# Read from a DESCRIPTION file directly
utils_description <- system.file("DESCRIPTION", package = "utils")
utils_authors <- read.dcf(utils_description, "Author")

parse_authors(utils_authors)

# Read from a database of CRAN metadata
cran_epidemiology_packages$Author |>
  parse_authors() |>
  unlist() |>
  unique() |>
  sort()
```

parse_authors_r

Parse the Authors@R field from a DESCRIPTION file

Description

Parse the Authors@R field from a DESCRIPTION file into a person object

Usage

```
parse_authors_r(authors_r_string)
```

Arguments

```
authors_r_string  A character containing the Authors@R field from a DESCRIPTION file
```

Value

A person object, or a list of person objects of length equals to the length of authors_r_string

Examples

```
# Read from a DESCRIPTION file directly
pkg_description <- system.file("DESCRIPTION", package = "authoritative")
authors_r_pkg <- read.dcf(pkg_description, "Authors@R")

parse_authors_r(authors_r_pkg)

# Read from a database of CRAN metadata
cran_epidemiology_packages |>
  subset(!is.na(`Authors@R`), `Authors@R`, drop = TRUE) |>
  parse_authors_r() |>
  head()
```

Index

* **datasets**

cran_epidemiology_packages, [2](#)

cran_epidemiology_packages, [2](#)

expand_names, [2](#)

invert_names, [3](#)

parse_authors, [4](#)

parse_authors_r, [5](#)