Package 'authordown'

January 22, 2026

Title Author Metadata Management and Manuscript Front Matter

Version 0.1.0

Author Zhipeng Zhang [aut, cre]

Maintainer Zhipeng Zhang <zhipeng30@foxmail.com>

Description Manage author metadata and generate manuscript front matter, including title pages, acknowledgements, conflicts of interest, and contributions, with support for large author lists.

License AGPL-3

URL https://github.com/zh1peng/authordown

BugReports https://github.com/zh1peng/authordown/issues

Encoding UTF-8

Depends R (>= 3.5)

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports openxlsx, rlang

Suggests rmarkdown, shiny, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Contents

Description

A convenience wrapper that calls the various section generators and combines them into one text block. Ideal for quick copy-paste into Word.

Usage

```
authordown(
  data,
  title = NULL,
  style = "default",
  show_degree = FALSE,
  acknowledgement_style = "paragraph",
  conflict_style = "paragraph",
  contribution_style = "paragraph"
)
```

Arguments

data	A data frame with the columns needed by each function: generate_title_page, generate_acknowledgement, generate_conflict, generate_contribution.
title	Optional paper title to pass to generate_title_page().
style	Title page style ("default", "APA", "Nature").
show_degree	Logical. If TRUE, include Degree after author names.
acknowledgement_style	Output style for acknowledgements.
conflict_style	Output style for conflicts of interest.
contribution_style	Output style for contributions.

Value

A character string containing all sections.

Examples

```
## Not run:
authors <- read.csv("authordown_template.csv")
cat(authordown(authors, title = "My Great Paper", style = "default"))

## End(Not run)
```

---

authordown_read_local *Read a local author metadata file*

Description

Reads a CSV/TSV/XLSX file and returns a standardized data frame.

Usage

```
authordown_read_local(path, sheet = NULL, validate = TRUE)
```

Arguments

path	Path to a local CSV, TSV, or XLSX file.
sheet	Optional sheet name or index for XLSX files.
validate	Logical. If TRUE, validate the data.

Value

A data frame with standardized columns.

Examples

```
## Not run:
authors <- authordown_read_local("authors.csv")

## End(Not run)
```

---

authordown_template *Write an authordown input template*

Description

Copies a standard author metadata template from the package into the requested path.

Usage

```
authordown_template(path = "authordown_template.csv", format = NULL)
```

Arguments

path	Output file path. Extension determines format if format is not supplied.
format	Optional format override: "csv" or "xlsx".

Value

Invisibly returns the output path.

Examples

```
## Not run:
authordown_template("authors.csv")
authordown_template("authors.xlsx")

## End(Not run)
```

authordown_validate *Validate author metadata*

Description

Checks required columns and key formats such as order, email, ORCID, and corresponding author rules.

Usage

```
authordown_validate(data, require_affiliations = FALSE)
```

Arguments

data	A data frame of author metadata.
require_affiliations	Logical. If TRUE, require at least one affiliation column.

Value

The validated, standardized data frame.

Examples

```
authors <- data.frame(FirstName = "Alice", LastName = "Smith")
authordown_validate(authors)
```

generate_acknowledgement *Generate an Acknowledgement Section*

Description

Combines acknowledgements from each author into a single formatted block.

Usage

```
generate_acknowledgement(data, style = c("paragraph", "bullets", "numbered"))
```

Arguments

data	A data frame containing at least the column Acknowledgement.
style	Output style: "paragraph", "bullets", or "numbered".

Value

A character string with the formatted acknowledgements.

Examples

```
authors <- data.frame(
  FirstName = c("Alice", "Bob"),
  LastName = c("Smith", "Johnson"),
  Acknowledgement = c("Thanks for funding A", "Supported by XYZ"),
  stringsAsFactors = FALSE
)
generate_acknowledgement(authors, style = "paragraph")
```

`generate_conflict` *Generate a Conflict of Interest Statement*

Description

Produces a readable conflict statement. If all authors report no conflict, it states so. Otherwise, it lists the authors reporting conflicts and then indicates that the remaining authors declare no conflict.

Usage

```
generate_conflict(data, style = c("paragraph", "bullets", "numbered"))
```

Arguments

`data`	A data frame containing at least the columns: FirstName, LastName, and Conflict.
`style`	Output style: "paragraph", "bullets", or "numbered".

Value

A character string with the formatted conflict statement.

Examples

```
authors <- data.frame(
  FirstName = c("Alice", "Bob"),
  LastName = c("Smith", "Johnson"),
  Conflict = c("No conflict", "Consultant at Company Z"),
  stringsAsFactors = FALSE
)
generate_conflict(authors, style = "paragraph")
```

generate_contribution *Generate Author Contributions*

Description

Combines each author's contribution statement into a clear paragraph. Each line indicates the author and their specific contribution.

Usage

```
generate_contribution(
  data,
  style = c("paragraph", "bullets", "numbered"),
  list_style = NULL
)
```

Arguments

data	A data frame containing at least the columns: FirstName, LastName, and Contribution.
style	Output style: "paragraph", "bullets", or "numbered".
list_style	Deprecated. Use style. When provided, TRUE maps to "bullets" and FALSE maps to "paragraph".

Value

A character string summarizing the author contributions.

Examples

```
authors <- data.frame(
  FirstName = c("Alice", "Bob"),
  LastName = c("Smith", "Johnson"),
  Contribution = c("Conceptualization; Data curation", "Supervision; Writing - review"),
  stringsAsFactors = FALSE
)
generate_contribution(authors, style = "paragraph")
```

generate_template *Generate a sample CSV template for authordown*

Description

Creates a sample CSV (or Excel) with columns for authors, affiliations, acknowledgments, conflicts, etc.

Usage

```
generate_template(file = "authordown_template.csv", excel = FALSE)
```

Arguments

file	A file path where the template should be written. Defaults to \"authordown_template.csv\" in the current directory.
excel	Logical. If TRUE, writes an Excel file (.xlsx) instead of CSV.

Value

Invisibly returns the data frame used for the template.

Examples

```
## Not run:
generate_template() # writes authordown_template.csv

## End(Not run)
```

---

generate_title_page *Generate a Title Page*

Description

Produces a formatted title page that includes the paper title, author names with affiliation indices, and an affiliation legend. Optionally adds a note for co-first authors.

Usage

```
generate_title_page(
  data,
  style = c("default", "APA", "Nature"),
  title = NULL,
  show_degree = FALSE,
  co_first_footnote = TRUE
)
```

Arguments

data	A data frame containing at least: FirstName, MiddleName, LastName, (optionally) Rank, Correspondence, and one or more Affiliation* columns.
style	A character string specifying the style (e.g., "default", "APA", "Nature").
title	An optional paper title.
show_degree	Logical. If TRUE, include Degree after author names.
co_first_footnote	Logical. If TRUE and Rank is provided, adds a note for co-first authors.

Value

A character string with the formatted title page.

Examples

```
authors <- data.frame(
  FirstName = c("Alice", "Bob"),
  MiddleName = c("M.", ""),
  LastName = c("Smith", "Johnson"),
  Degree = c("PhD", "MD"),
  Email = c("alice@example.com", "bob@example.com"),
  Rank = c(1, 2),
  Correspondence = c(TRUE, FALSE),
  Affiliation1 = c("University of X, Dept. of Y", "University of X, Dept. of Y"),
  Affiliation2 = c(NA, "Company Z, Research Division"),
  stringsAsFactors = FALSE
)
generate_title_page(authors, style = "default", title = "Example Paper", show_degree = TRUE)
```

---

render_section_html *Render a Manuscript Section to HTML*

Description

This function takes a section title, a content-generating function, and a data frame, then produces an HTML file that displays the section header and its content. This generic function works for any section (e.g. Conflict of Interest, Author Contributions, Acknowledgements).

Usage

```
render_section_html(
  section_title,
  content_function,
  data,
  output_file = tempfile(fileext = ".html"),
  ...
)
```

Arguments

section_title	A character string for the section header (e.g., "Conflict of Interest").
content_function	A function that accepts a data frame and returns a formatted character string.
data	A data frame containing the necessary columns.
output_file	The path to the output HTML file. Defaults to a temporary file.
...	Additional arguments passed to content_function.

Value

A character string with the path to the rendered HTML file.

Examples

```
## Not run:
  # To render the Conflict of Interest section:
  html_path <- render_section_html("Conflict of Interest", generate_conflict, authors)
  browseURL(html_path)

## End(Not run)
```