

# Package: authordown (via r-universe)

May 25, 2026

**Title** Author Metadata Management and Manuscript Front Matter

**Version** 0.1.0

**Author** Zhipeng Cao [aut, cre]

**Maintainer** Zhipeng Cao <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://zhipeng.github.io/authordown/>,  
<https://github.com/zhipeng/authordown>

**BugReports** <https://github.com/zhipeng/authordown/issues>

**Encoding** UTF-8

**Depends** R (>= 3.5)

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Imports** openxlsx, rlang

**Suggests** knitr, pkgdown, rmarkdown, shiny, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Config/pak/sysreqs** libicu-dev

**Repository** <https://zhipeng.r-universe.dev>

**Date/Publication** 2026-02-05 08:46:35 UTC

**RemoteUrl** <https://github.com/zhipeng/authordown>

**RemoteRef** HEAD

**RemoteSha** 40fa9cf411ca41f789c82228a7ccb666ac0e678d

## Contents

authordown . . . . .	2
authordown_read_local . . . . .	3
authordown_template . . . . .	4
authordown_validate . . . . .	4
generate_acknowledgement . . . . .	5
generate_conflict . . . . .	6
generate_contribution . . . . .	6
generate_template . . . . .	7
generate_title_page . . . . .	8
render_section_html . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

authordown	<i>Generate all sections (title page, acknowledgement, conflict, contribution)</i>
------------	--

---

## 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. By default, the data are validated for required columns and basic formats.

**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. Use this to start a new project with the expected columns and example values.

### 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 (Updated)*

---

**Description**

Combines acknowledgements provided by the authors into a single, coherent sentence. The function ignores author names (as they are assumed to be included in the acknowledgement text itself) and simply concatenates each non-empty acknowledgement entry using a semicolon.

**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 combined acknowledgements.

**Examples**

```
authors <- data.frame(
  Acknowledgement = c("This work received support from resource X",
                     "Resource Y was instrumental",
                     "Alice Smith received support from resource ZZ"),
  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 (Journal-Style)*

---

## Description

Produces a journal-style title page text with:

1. An optional paper title
2. Author line (with superscript affiliation indices)
3. Affiliation list (labeled by superscript numbers)
4. Co-first author footnote (dagger symbol) if multiple authors share the same minimal rank
5. Corresponding author footnote (\*) and contact line

The function expects columns:

- **FirstName, MiddleName, LastName** (strings)
- **Rank** (numeric), used to detect co-first authors (lowest rank)
- **Correspondence** (logical or "TRUE"/"FALSE" string)
- **Email** (for corresponding authors)
- **Affiliation1, Affiliation2**, etc. (any number of these)

## Usage

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

## Arguments

<code>data</code>	A data frame with the columns listed above.
<code>style</code>	Title page style. Supported values: "default", "APA", "Nature".
<code>title</code>	Optional character string for the paper title.
<code>show_degree</code>	Logical. If TRUE, include Degree after author names.
<code>co_first_footnote</code>	Logical. If TRUE, adds a footnote for co-first authors if multiple authors share the same <b>lowest</b> rank.

## Value

A single character string suitable for copy-pasting into Word.

**Examples**

```

authors <- data.frame(
  FirstName     = c("Alice", "Bob", "Charlie"),
  MiddleName    = c("M.", "", "Q."),
  LastName      = c("Smith", "Johnson", "Lee"),
  Degree        = c("PhD", "MD", "PhD"),
  Email         = c("alice@example.com", "bob@example.com", "charlie@example.org"),
  Rank          = c(1, 1, 2), # Alice and Bob are co-first authors
  Correspondence = c(TRUE, FALSE, FALSE),
  Affiliation1  = c("University of X, Dept. of Y",
                  "University of X, Dept. of Y",
                  "Institute of Z"),
  Affiliation2  = c(NA, "Company W", NA),
  stringsAsFactors = FALSE
)
cat(generate_title_page(authors, title = "A Great 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)
```

# Index

authordown, [2](#)  
authordown\_read\_local, [3](#)  
authordown\_template, [4](#)  
authordown\_validate, [4](#)  
  
generate\_acknowledgement, [5](#)  
generate\_conflict, [6](#)  
generate\_contribution, [6](#)  
generate\_template, [7](#)  
generate\_title\_page, [8](#)  
  
render\_section\_html, [9](#)