

# Package ‘scholid’

February 13, 2026

**Type** Package

**Title** Tools for Scholarly and Academic Identifiers

**Version** 0.1.0

**Language** en-US

**Description** Tools for detecting, normalizing, classifying, and extracting scholarly identifier strings. The package provides lightweight, dependency-free helpers for common identifier systems such as DOIs, ORCID iDs, ISBNs, ISSNs, arXiv identifiers, and PubMed identifiers. Functions are designed to be vectorized, predictable, and suitable as low-level building blocks for other R packages and data workflows.

**License** MIT + file LICENSE

**URL** <https://thomas-rauter.github.io/scholid/>

**BugReports** <https://github.com/Thomas-Rauter/scholid/issues>

**Depends** R (>= 3.5.0)

**Suggests** testthat (>= 3.0.0), knitr (>= 1.30), rmarkdown

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Thomas Rauter [aut, cre, fnd] (ORCID:  
<<https://orcid.org/0009-0004-5578-3628>>)

**Maintainer** Thomas Rauter <rauterthomas@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-02-13 16:20:02 UTC

## Contents

classify_scholid . . . . .	2
detect_scholid_type . . . . .	3
extract_scholid . . . . .	4
is_scholid . . . . .	4
normalize_scholid . . . . .	5
scholid_types . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

classify_scholid	<i>Classify scholarly identifiers</i>
------------------	---------------------------------------

---

### Description

Performs best-guess classification of scholarly identifier strings. For each element of the input, the function returns the first matching identifier type, or `NA_character_` if no supported type matches.

Classification is based on canonical identifier syntax. Wrapped forms (e.g., URLs or labels) should be normalized first with `normalize_scholid()`.

### Usage

```
classify_scholid(x)
```

### Arguments

`x`                    A vector of candidate identifier values.

### Value

A character vector of the same length as `x`, giving the detected identifier type for each element, or `NA_character_` if no match is found.

### Examples

```
classify_scholid(c("10.1000/182", "0000-0002-1825-0097", "not an id"))
classify_scholid(normalize_scholid("https://doi.org/10.1000/182", "doi"))
```

---

detect\_scholid\_type     *Detect scholarly identifier types*

---

### Description

Performs best-effort detection of scholarly identifier types from possibly wrapped identifier strings (e.g., URLs or labels).

For each element of the input, the function returns the first matching identifier type, or `NA_character_` if no supported type matches.

Detection first attempts classification based on canonical identifier syntax (see `classify_scholid()`). If no match is found, the function attempts per-type normalization (see `normalize_scholid()`) and returns the first type for which normalization yields a non-missing result.

Use `normalize_scholid()` to convert detected values to canonical form once the identifier type is known.

### Usage

```
detect_scholid_type(x)
```

### Arguments

`x`                     A vector of candidate identifier values.

### Value

A character vector of the same length as `x`, giving the detected identifier type for each element, or `NA_character_` if no match is found.

### See Also

[classify\\_scholid\(\)](#), [normalize\\_scholid\(\)](#), [scholid\\_types\(\)](#)

### Examples

```
detect_scholid_type(c(
  "https://doi.org/10.1000/182",
  "doi:10.1000/182",
  "https://orcid.org/0000-0002-1825-0097",
  "arXiv:2101.12345v2",
  "PMID: 12345678",
  "PMCID: PMC1234567",
  "not an id"
))
```

---

extract_scholid	<i>Extract scholarly identifiers from text</i>
-----------------	--

---

### Description

Extract identifiers of a single supported type from free text.

The result is a list with one element per input element. Each element is a character vector of matches (possibly length 0). NA inputs yield an empty character vector.

Matches are returned as found in the text; use `normalize_scholid()` to convert identifiers to canonical form.

### Usage

```
extract_scholid(text, type)
```

### Arguments

text	A character vector of text.
type	A single string giving the identifier type. See <code>scholid_types()</code> for supported values.

### Value

A list of character vectors of extracted identifiers.

### Examples

```
extract_scholid("See https://doi.org/10.1000/182.", "doi")
extract_scholid("ORCID 0000-0002-1825-0097", "orcid")
```

---

is_scholid	<i>Test scholarly identifier validity</i>
------------	---

---

### Description

Vectorized predicate that tests whether values are valid scholarly identifiers of a given supported type.

Validation is stricter than normalization. Values must conform to the canonical identifier syntax, and for identifier types with checksum algorithms (e.g., ORCID, ISBN, ISSN), checksum correctness is verified.

Inputs that are NA yield NA. Non-matching values return FALSE.

Use `normalize_scholid()` to convert structurally plausible identifiers to canonical form without performing checksum validation.

**Usage**

```
is_scholid(x, type)
```

**Arguments**

**x** A vector of values to test.

**type** A single string giving the identifier type. See [scholid\\_types\(\)](#) for supported values.

**Value**

A logical vector of the same length as `x`, indicating whether each element is a valid identifier of the specified type.

**See Also**

[normalize\\_scholid\(\)](#), [scholid\\_types\(\)](#)

**Examples**

```
is_scholid("10.1000/182", "doi")
is_scholid("0000-0002-1825-0097", "orcid")
```

---

normalize\_scholid      *Normalize scholarly identifiers*

---

**Description**

Vectorized normalizer that converts supported scholarly identifier values to a canonical form (e.g., removing URL prefixes, labels, or separators).

Normalization is structural: inputs that conform to the expected identifier syntax are converted to a canonical representation. Inputs that do not match the required structure yield `NA_character_`.

For identifier types with checksum algorithms (e.g., ORCID, ISBN, ISSN), normalization does not verify checksum correctness. It only enforces structural plausibility and canonical formatting.

Use [is\\_scholid\(\)](#) to test whether values are fully valid identifiers, including checksum verification where applicable.

**Usage**

```
normalize_scholid(x, type)
```

**Arguments**

**x** A vector of values to normalize.

**type** A single string giving the identifier type. See [scholid\\_types\(\)](#) for supported values.

**Value**

A character vector with the same length as `x`. Invalid or structurally non-matching inputs yield `NA_character_`.

**See Also**

[is\\_scholid\(\)](#), [scholid\\_types\(\)](#)

**Examples**

```
normalize_scholid("https://doi.org/10.1000/182", "doi")
normalize_scholid("https://orcid.org/0000-0002-1825-0097", "orcid")
```

---

scholid\_types

*Supported scholid identifier types*

---

**Description**

Returns the set of identifier types supported by the scholid package.

**Usage**

```
scholid_types()
```

**Value**

A character vector of supported identifier type strings.

**Examples**

```
scholid_types()
"orcid" %in% scholid_types()
```

# Index

`classify_scholid`, 2  
`classify_scholid()`, 3  
`detect_scholid_type`, 3  
`extract_scholid`, 4  
`is_scholid`, 4  
`is_scholid()`, 5, 6  
`normalize_scholid`, 5  
`normalize_scholid()`, 3–5  
`scholid_types`, 6  
`scholid_types()`, 3, 5, 6