Package 'scholid'

June 4, 2026

Type Package

Title Scholarly and Academic Identifier Utilities

Version 0.2.0

Language en-US

Description Detects, normalizes, classifies, and extracts scholarly identifier strings. Provides lightweight, dependency-free helpers for twenty identifier types, including DOIs, ORCID iDs, ISBNs, ISSNs, arXiv and PubMed identifiers, ROR and ISNI, OpenAlex and ADS bibcodes, RRID, ARK, SWHID, and selected life-science accessions (UniProt, RefSeq, SRA, GEO, BioProject, and genome assemblies). Functions are vectorized, predictable, and suitable as low-level building blocks for other R packages and data workflows. Use 'scholid_types()' for the authoritative type list. For online lookup, conversion, metadata retrieval, and linked identifier discovery, see 'scholidonline'.

License MIT + file LICENSE

URL https://thomas-rauter.github.io/scholid/, https://thomas-rauter.github.io/scholidonline/

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 <rauterthomas0@gmail.com>

Repository CRAN

Date/Publication 2026-06-04 16:20:02 UTC

Contents

`scholid-package` *Scholarly and Academic Identifier Utilities*

Description

`scholid` provides lightweight, dependency-free utilities for detecting, normalizing, classifying, and extracting scholarly identifier strings. The package supports twenty identifier types; see `scholid_types()` for the authoritative list and classification order.

Vignettes

- Getting started introduces the exported functions and typical workflows for mixed identifier data.
- scholid_definitions (*About identifiers*) documents per-type formats, validation rules, and classification precedence.

Author(s)

Maintainer: Thomas Rauter <rauterthomas0@gmail.com> (ORCID) [funder]

See Also

`is_scholid()`, `normalize_scholid()`, `extract_scholid()`, `classify_scholid()`, `detect_scholid_type()`, `scholid_types()`

classify_scholid *Classify scholarly identifiers*

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. Types are checked in the order returned by `scholid_types()` (most specific first); the first match wins. 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.

See Also

`detect_scholid_type()`, `scholid_types()`, scholid_definitions

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. PMID is checked last as a fallback when no more specific type matches.

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 *Extract scholarly identifiers from text*

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 extracted identifier tokens from the text. Surrounding prose punctuation or markup fragments may be removed where necessary to isolate the identifier. 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 scholid_types() 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 *Test scholarly identifier validity*

Description

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

For identifier types with checksum algorithms (e.g., ORCID, ROR, ISNI, ISBN, ISSN), checksum correctness is verified. The same checksum rules apply to normalize_scholid().

The main difference from normalization is input form: is_scholid() expects values in canonical (or near-canonical) form. Wrapped values such as URLs or prefixed labels should be normalized first with normalize_scholid().

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

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 requires that inputs match the expected identifier structure. For identifier types with checksum algorithms (ORCID, ROR, ISNI, ISBN, ISSN), normalization also requires checksum-valid values. Inputs that do not meet these requirements yield NA_character_.

Normalized outputs are canonical, type-specific representations of valid identifiers.

Use is_scholid() to test whether already-canonical values are valid identifiers of a given type. Both functions apply checksum verification where applicable; normalization additionally accepts wrapped input forms and returns canonical strings.

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, checksum- failing, 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 in classification priority order (most specific first). The package currently supports twenty types (from DOI and ORCID through life-science and archive identifiers). For per-type formats, validation rules, and classification precedence, see the *How Scholarly Identifiers Are Defined* vignette (`vignette("scholid_definitions", package = "scholid")`).

Usage

```
scholid_types()
```

Value

A character vector of supported identifier type strings.

Examples

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

Index