Package 'biobouncer'

September 3, 2026

Title Validate Biological Identifiers and Inputs

Version 0.1.4

Description Gene symbols, ontology terms, variant formats, and database accessions are validated through one small application programming interface (API), either offline against identifier patterns and bundled snapshots or live against a source web service. Results are designed to match those of the companion 'Python' package for the same inputs. Input order and length are preserved, and errors are reported explicitly rather than as a silent failure. Identifier patterns follow the registry and nomenclature described in Hoyt et al. (2022) <doi:10.1038/s41597-022-01807-3> and den Dunnen et al. (2016) <doi:10.1002/humu.22981>.

License MIT + file LICENSE

URL https://github.com/samuelbharti/biobouncer, https://www.samuelbharti.com/biobouncer/r/

BugReports https://github.com/samuelbharti/biobouncer/issues

Encoding UTF-8

Depends R (>= 4.0)

Imports checkmate, cli, curl, jsonlite, tibble, tools, utils, yaml

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

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Samuel Bharti [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-4190-7058>)

Maintainer Samuel Bharti <samuelbharti.io@gmail.com>

Repository CRAN

Date/Publication 2026-09-03 11:20:15 UTC

Contents

`assert_valid_id` *Assert that identifiers are valid*

Description

A checkmate-style assertion. Throws an error when any element of `x` is not a valid identifier for `source_db`, otherwise returns `x` invisibly.

Usage

```
assert_valid_id(
  x,
  source_db,
  how = "pattern",
  species = NULL,
  version = NULL,
  .var.name = checkmate::vname(x),
  add = NULL
)
```

Arguments

`x`	A vector of identifiers. Coerced to character.
`source_db`	Source key, for example `"mondo"`. See `sources()`.
`how`	Checking mode: `"pattern"` (offline, shape only), `"cache"` (offline existence against a snapshot), `"remote"` (live existence against the source API), or `"existence"` (cache when a snapshot is available for `version`, otherwise remote, or pattern for a source with no resolver).

species	Optional species context, echoed in the result. A name such as "homo_sapiens" or an NCBI taxon id such as 9606. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in pattern and remote modes), and UniProt from the entry's organism in remote mode. A species outside the source map is not checked.
version	Snapshot version. In cache mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for existence mode; ignored in pattern and remote modes.
.var.name	Name for x to use in error messages.
add	A checkmate AssertCollection to push messages onto, or NULL.

Value

x invisibly on success.

See Also

check_valid_id(), test_valid_id().

Examples

```
assert_valid_id(c("MONDO:0005148", "MONDO:0018076"), "mondo")
```

---

bioboucer_cache_dir *Snapshot cache directory*

Description

The directory where downloaded snapshots are stored. Set the environment variable BIOBOUNCER_CACHE_DIR to override the default.

Usage

```
bioboucer_cache_dir()
```

Value

A path to the cache directory.

Examples

```
bioboucer_cache_dir()
```

`biobouncer_pull` *Download a snapshot for cache mode*

Description

Dispatches on the source's `cache$builder`: `obo` fetches the ontology release, `hgnc_tsv` fetches the HGNC complete set. Identifiers that match the source pattern are written to the cache directory as a snapshot, and a retired-id map, when the builder produces one, to the matching `<version>.retired.tsv` sidecar. An OBO version defaults to the ontology's own data-version; an HGNC version defaults to the source's `default_version`.

Usage

```
biobouncer_pull(source_db, version = NULL, quiet = FALSE)
```

Arguments

`source_db`	Source key, for example `"mondo"`.
`version`	Snapshot version label. Defaults to the builder's own version.
`quiet`	Suppress progress messages.

Value

The path to the written snapshot, invisibly.

See Also

`biobouncer_snapshots()`, `check_id()`.

Examples

```
## Not run:
# Downloads a full snapshot into the cache directory (needs a network).
biobouncer_pull("mondo")

## End(Not run)
```

biobouncer_snapshots *List installed snapshots*

Description

Reports snapshots available for `cache` mode, both downloaded ones in the cache directory and the small bundled samples.

Usage

```
biobouncer_snapshots()
```

Value

A tibble with columns `source`, `version`, `n_ids`, and `location` (`"cache"` or `"bundled"`).

Examples

```
biobouncer_snapshots()
```

check_id *Check biological identifiers*

Description

Validate a vector of identifiers against a source. `pattern` mode checks that each identifier is well-formed. `cache` mode also checks that it exists in a pinned local snapshot (see `biobouncer_snapshots()`). `remote` mode checks live existence against the source API. `existence` mode uses a snapshot when one is available for `version`, otherwise falls back to `remote`, and for a source with no resolver falls back to `pattern`.

Usage

```
check_id(
  x,
  source_db,
  how = "pattern",
  species = NULL,
  version = NULL,
  refresh = FALSE,
  on_error = "raise"
)
```

Arguments

x	A vector of identifiers. Coerced to character.
source_db	Source key, for example "mondo". See sources().
how	Checking mode: "pattern" (offline, shape only), "cache" (offline existence against a snapshot), "remote" (live existence against the source API), or "existence" (cache when a snapshot is available for version, otherwise remote, or pattern for a source with no resolver).
species	Optional species context, echoed in the result. A name such as "homo_sapiens" or an NCBI taxon id such as 9606. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in pattern and remote modes), and UniProt from the entry's organism in remote mode. A species outside the source map is not checked.
version	Snapshot version. In cache mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for existence mode; ignored in pattern and remote modes.
refresh	In remote checks, skip any cached response and refetch. Ignored by the offline modes.
on_error	How a per-id remote failure is handled. "raise" (the default) lets the failure unwind the whole call. "indeterminate" leaves just that id NA with the reason in its error column and checks the rest of the batch, so one unreachable id does not lose the others. Ignored by the offline modes.

Value

A tibble with one row per element of x and the columns input, valid, normalized, suggestion, source_db, version, species, how, and error.

See Also

is_valid_id(), sources(), source_info(), biobouncer_snapshots().

Examples

```
check_id(c("MONDO:0005148", "mondo:5148"), source_db = "mondo")
check_id("MONDO:0005148", source_db = "mondo", how = "cache", version = "sample")
```

---

check_valid_id *Check that identifiers are valid*

Description

A checkmate-style check function. Returns TRUE when every element of x is a valid identifier for source_db, otherwise a message describing the failure. A missing element (NA) is not a failure. Pairs with assert_valid_id() and test_valid_id().

Usage

```
check_valid_id(x, source_db, how = "pattern", species = NULL, version = NULL)
```

Arguments

x	A vector of identifiers. Coerced to character.
source_db	Source key, for example "mondo". See sources().
how	Checking mode: "pattern" (offline, shape only), "cache" (offline existence against a snapshot), "remote" (live existence against the source API), or "existence" (cache when a snapshot is available for version, otherwise remote, or pattern for a source with no resolver).
species	Optional species context, echoed in the result. A name such as "homo_sapiens" or an NCBI taxon id such as 9606. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in pattern and remote modes), and UniProt from the entry's organism in remote mode. A species outside the source map is not checked.
version	Snapshot version. In cache mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for existence mode; ignored in pattern and remote modes.

Value

TRUE if all elements are valid, otherwise a message string.

See Also

assert_valid_id(), test_valid_id(), is_valid_id().

Examples

```
check_valid_id(c("MONDO:0005148", "MONDO:0018076"), "mondo")
check_valid_id("mondo:5148", "mondo")
```

---

id_predicate *Build a vectorized predicate for data-frame validation*

Description

Returns a function of one argument that reports, element by element, whether each value is a valid identifier for source_db. The predicate is the shape that data-frame validation packages expect, so it drops straight into assertr::assert(), a validate::validator() rule, or a pointblank col_vals_expr() step, or into base Filter() and vapply(). It is the R counterpart of the pandera check in the Python package.

Usage

```
id_predicate(source_db, how = "pattern", species = NULL, version = NULL)
```

Arguments

source_db	Source key, for example "mondo". See sources().
how	Checking mode: "pattern" (offline, shape only), "cache" (offline existence against a snapshot), "remote" (live existence against the source API), or "existence" (cache when a snapshot is available for version, otherwise remote, or pattern for a source with no resolver).
species	Optional species context, echoed in the result. A name such as "homo_sapiens" or an NCBI taxon id such as 9606. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in pattern and remote modes), and UniProt from the entry's organism in remote mode. A species outside the source map is not checked.
version	Snapshot version. In cache mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for existence mode; ignored in pattern and remote modes.

Value

A function of one argument that returns a logical vector the same length as its input. A missing cell (NA) passes, so it is never dropped by a filter or flagged by a data-frame rule; only a malformed or non-existent id is FALSE.

See Also

is_valid_id(), check_valid_id().

Examples

```
is_mondo <- id_predicate("mondo")
ids <- c("MONDO:0005148", "mondo:5148", "MONDO:0018076")
is_mondo(ids)
ids[is_mondo(ids)]

# The predicate takes the shape a data-frame validator expects, so it also
# applies to a column directly.
df <- data.frame(term = ids)
df[is_mondo(df$term), , drop = FALSE]
```

---

is_valid_id *Test biological identifiers*

Description

A convenience wrapper over check_id() that returns only the verdict.

Usage

```
is_valid_id(
  x,
  source_db,
  how = "pattern",
  species = NULL,
  version = NULL,
  refresh = FALSE
)
```

Arguments

x	A vector of identifiers. Coerced to character.
source_db	Source key, for example "mondo". See sources().
how	Checking mode: "pattern" (offline, shape only), "cache" (offline existence against a snapshot), "remote" (live existence against the source API), or "existence" (cache when a snapshot is available for version, otherwise remote, or pattern for a source with no resolver).
species	Optional species context, echoed in the result. A name such as "homo_sapiens" or an NCBI taxon id such as 9606. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in pattern and remote modes), and UniProt from the entry's organism in remote mode. A species outside the source map is not checked.
version	Snapshot version. In cache mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for existence mode; ignored in pattern and remote modes.
refresh	In remote checks, skip any cached response and refetch. Ignored by the offline modes.

Value

A logical vector, one element per element of x.

See Also

check_id().

Examples

```
is_valid_id(c("MONDO:0005148", "mondo:5148"), source_db = "mondo")
```

repair_id *Repair a column of identifiers*

Description

Returns `x` with every fixable value substituted by its suggestion: an invalid value that maps to a successor or a well-formed alternative. Valid values, invalid values with no suggestion, and missing values are returned unchanged, so the result is the same length and order as `x`. It is designed to drop into `dplyr::mutate()`.

Usage

```
repair_id(
  x,
  source_db,
  how = "pattern",
  species = NULL,
  version = NULL,
  refresh = FALSE,
  on_error = "raise"
)
```

Arguments

`x`	A vector of identifiers. Coerced to character.
`source_db`	Source key, for example `"mondo"`. See `sources()`.
`how`	Checking mode: `"pattern"` (offline, shape only), `"cache"` (offline existence against a snapshot), `"remote"` (live existence against the source API), or `"existence"` (cache when a snapshot is available for `version`, otherwise remote, or pattern for a source with no resolver).
`species`	Optional species context, echoed in the result. A name such as `"homo_sapiens"` or an NCBI taxon id such as `9606`. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in `pattern` and `remote` modes), and UniProt from the entry's organism in `remote` mode. A species outside the source map is not checked.
`version`	Snapshot version. In `cache` mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for `existence` mode; ignored in `pattern` and `remote` modes.
`refresh`	In remote checks, skip any cached response and refetch. Ignored by the offline modes.
`on_error`	How a per-id remote failure is handled. `"raise"` (the default) lets the failure unwind the whole call. `"indeterminate"` leaves just that id `NA` with the reason in its `error` column and checks the rest of the batch, so one unreachable id does not lose the others. Ignored by the offline modes.

Value

A character vector the same length as `x`.

See Also

`report_id()`, `check_id()`.

Examples

```
repair_id(c("MONDO:0005148", "mondo:5148"), "mondo")
```

`report_id` *Validate and report on a column of identifiers*

Description

`report_id()` runs `check_id()` over a column and returns its result table with the extra class `bioboucer_report`, so it prints with a one-line summary of how many values are valid, repairable, invalid, or missing. It is the recommended entry point for inspecting and cleaning a column. Use `repair_id()` inside `dplyr::mutate()` to substitute the fixable values. For enforcing validity inside a framework, reach for the adapters such as `id_predicate()`.

Usage

```
report_id(
  x,
  source_db,
  how = "pattern",
  species = NULL,
  version = NULL,
  refresh = FALSE,
  on_error = "raise"
)

## S3 method for class 'biobouncer_report'
summary(object, ...)

## S3 method for class 'biobouncer_report'
print(x, ...)
```

Arguments

`x`	A column of identifiers for `report_id()`, or a `bioboucer_report` for the `print()` method.
`source_db`	Source key, for example `"mondo"`. See `sources()`.

how	Checking mode: "pattern" (offline, shape only), "cache" (offline existence against a snapshot), "remote" (live existence against the source API), or "existence" (cache when a snapshot is available for version, otherwise remote, or pattern for a source with no resolver).
species	Optional species context, echoed in the result. A name such as "homo_sapiens" or an NCBI taxon id such as 9606. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in pattern and remote modes), and UniProt from the entry's organism in remote mode. A species outside the source map is not checked.
version	Snapshot version. In cache mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for existence mode; ignored in pattern and remote modes.
refresh	In remote checks, skip any cached response and refetch. Ignored by the offline modes.
on_error	How a per-id remote failure is handled. "raise" (the default) lets the failure unwind the whole call. "indeterminate" leaves just that id NA with the reason in its error column and checks the rest of the batch, so one unreachable id does not lose the others. Ignored by the offline modes.
object	A biobouncer_report, as returned by report_id().
...	Ignored.

Value

A tibble, as returned by check_id(), with the extra class biobouncer_report. Calling summary() on it returns a one-row tibble of counts.

See Also

repair_id(), check_id().

Examples

```
report_id(c("MONDO:0005148", "mondo:5148"), "mondo")
```

---

sources *List available source databases*

Description

List available source databases

Usage

```
sources()
```

Value

A character vector of source keys, sorted.

See Also

`source_info()` for a table with more detail.

Examples

```
sources()
```

---

source_info *Describe the available sources*

Description

Answers "what does a valid id look like and how can I check it?" for each source.

Usage

```
source_info()
```

Value

A tibble with one row per source and the columns `key`, `name`, `example` (a valid identifier for the source), `modes` (the checking modes it supports), `species_aware`, and `version_aware`.

Examples

```
source_info()
```

---

sv_biobouncer *Create a shinyvalidate rule*

Description

Returns a rule function for use with `shinyvalidate::InputValidator`'s `add_rule()`. The rule returns `NULL` for a valid input and a message otherwise. This adapter does not depend on shinyvalidate; it only produces a rule the package can consume.

Usage

```
sv_biobouncer(
  source_db,
  how = "pattern",
  species = NULL,
  version = NULL,
  message = NULL
)
```

Arguments

source_db	Source key, for example "mondo". See sources().
how	Checking mode: "pattern" (offline, shape only), "cache" (offline existence against a snapshot), "remote" (live existence against the source API), or "existence" (cache when a snapshot is available for version, otherwise remote, or pattern for a source with no resolver).
species	Optional species context, echoed in the result. A name such as "homo_sapiens" or an NCBI taxon id such as 9606. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in pattern and remote modes), and UniProt from the entry's organism in remote mode. A species outside the source map is not checked.
version	Snapshot version. In cache mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for existence mode; ignored in pattern and remote modes.
message	Optional custom message for an invalid input.

Value

A function of one argument suitable for add_rule().

See Also

check_valid_id().

Examples

```
rule <- sv_biobouncer("mondo")
rule("MONDO:0005148")
rule("mondo:5148")
```

synthesize_ids *Generate a synthetic column of identifiers*

Description

`synthesize_ids()` builds a small "messy column" for a source: a mix of well-formed ids, repairable ids (a wrong-case or unpadded form that suggests a valid one), hard-invalid ids, and missing cells. Every value is labeled by running the checker, so the labels are always correct and match what the Python `synthesize()` produces for the same source. It works in `pattern` mode (the shape) for any source and in `cache` mode (the snapshot) for a source that ships one. It is useful for exercising a validation pipeline (feed the column to `report_id()`, `repair_id()`, or an adapter) without hand-writing test data. The generation is deterministic and offline.

Usage

```
synthesize_ids(
  source_db,
  how = "pattern",
  version = NULL,
  n_valid = 2,
  n_repairable = 1,
  n_invalid = 1,
  missing = 1,
  seed = 0
)
```

Arguments

`source_db`	Source key, for example `"mondo"`. See `sources()`.
`how`	Checking mode to label against: `"pattern"` (the shape, any source) or `"cache"` (the snapshot; the source must ship one).
`version`	In cache mode, the snapshot version. Defaults to `"sample"`.
`n_valid`	How many well-formed or in-snapshot ids to include. A source with no numeric part yields just the example.
`n_repairable`	How many repairable ids (a wrong-case or unpadded form that suggests a valid id, or in cache mode a retired id that maps to a successor). `ec`, `hgvs`, and `hgnc` have no pattern-mode repairable form, so they yield none there.
`n_invalid`	How many hard-invalid ids (neither valid nor suggestible).
`missing`	How many missing cells (`NA`).
`seed`	Shifts the numeric variants, for a different but still deterministic column (pattern mode).

Value

A tibble with the columns `input`, `category` (`"valid"`, `"repairable"`, `"invalid"`, or `"missing"`), and the `valid`, `normalized`, and `suggestion` the checker returned for that input. Categories a source cannot produce are simply absent.

See Also

report_id(), check_id().

Examples

```
synthesize_ids("mondo")
```

---

test_valid_id *Test whether identifiers are valid*

Description

A checkmate-style test. Returns a single TRUE when every element of x is valid for source_db, otherwise FALSE.

Usage

```
test_valid_id(x, source_db, how = "pattern", species = NULL, version = NULL)
```

Arguments

x	A vector of identifiers. Coerced to character.
source_db	Source key, for example "mondo". See sources().
how	Checking mode: "pattern" (offline, shape only), "cache" (offline existence against a snapshot), "remote" (live existence against the source API), or "existence" (cache when a snapshot is available for version, otherwise remote, or pattern for a source with no resolver).
species	Optional species context, echoed in the result. A name such as "homo_sapiens" or an NCBI taxon id such as 9606. When given, an id of a different species is invalid: Ensembl is checked from its id prefix (in pattern and remote modes), and UniProt from the entry's organism in remote mode. A species outside the source map is not checked.
version	Snapshot version. In cache mode it selects the snapshot and defaults to the latest installed one when omitted; it selects the snapshot for existence mode; ignored in pattern and remote modes.

Value

A single logical.

See Also

check_valid_id(), assert_valid_id().

Examples

```
test_valid_id(c("MONDO:0005148", "MONDO:0018076"), "mondo")
test_valid_id("mondo:5148", "mondo")
```

Index