

Package ‘rfair’

September 23, 2026

Title Assess the FAIRness of Research Data Objects and Software

Version 0.2.0

Description A native R implementation of the F-UJI (FAIRsFAIR Research Data Object Assessment) and FRSM (FAIR for Research Software) metrics for evaluating how well a research data object or piece of research software satisfies the FAIR principles (Findable, Accessible, Interoperable, Reusable). The software metrics operationalize the FAIR Principles for Research Software (FAIR4RS) of Chue Hong et al. (2022) [doi:10.15497/RDA00068](https://doi.org/10.15497/RDA00068). Given a persistent identifier, URL, or code repository, 'rfair' resolves it, harvests metadata from landing pages and registries, and scores it against the FAIRsFAIR metrics of Devaraju and Huber (2020) [doi:10.5281/zenodo.3775793](https://doi.org/10.5281/zenodo.3775793) entirely in R, without requiring an external assessment server. 'rfair' began as a fork of the 'rfuji' F-UJI API client and reimplements the assessment engine natively.

License GPL-3

URL <https://github.com/choxos/rfair>, <https://choxos.github.io/rfair/>

BugReports <https://github.com/choxos/rfair/issues>

Depends R (>= 4.1)

Imports curl, digest, httr2 (>= 1.1.0), jsonlite, mime, parallel, rvest, stats, stringdist, utils, xml2, yaml

Suggests bslib, chromote, DT, jsonld, knitr, plumber, rdfliib, rmarkdown, shiny, testthat (>= 3.1.7), withr

Config/testthat/edition 3

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Ahmad Sofi-Mahmudi [aut, cre] (ORCID:
[<https://orcid.org/0000-0001-6829-0823>](https://orcid.org/0000-0001-6829-0823)),
 Steffen Neumann [ctb] (Author of the original rfujl F-UJI API client
 that rfair grew from),
 PANGAEA [cph] (Copyright holder of the F-UJI service whose metrics
 rfair reimplements)

Maintainer Ahmad Sofi-Mahmudi <a.sofimahmudi@gmail.com>

Repository CRAN

Date/Publication 2026-09-23 15:10:02 UTC

Contents

as.data.frame.fair_assessment	3
assess_data_code	3
assess_fair	5
assess_fair_batch	6
as_fuji_json	7
as_rdf	8
classify_access	8
fair4rs_principles	9
fair_assessment	10
fair_compare	10
fair_example	11
fair_principles	12
fair_recommendations	12
fair_tlc	13
frsm_agreement	14
identifier_hygiene	15
id_parse	15
launch_rfai	16
license_reuse	16
plot.fair_assessment	17
principle_definition	18
reusabledata_rating	19
rfair_metric_versions	19
split_identifiers	20
summary.fair_assessment	20

Index

21

as.data.frame.fair_assessment

Convert a FAIR assessment to a per-metric data frame.

Description

Convert a FAIR assessment to a per-metric data frame.

Usage

```
## S3 method for class 'fair_assessment'
as.data.frame(x, ...)
```

Arguments

x	A fair_assessment object.
...	Ignored.

Value

A data frame with one row per metric.

assess_data_code	<i>Assess the FAIRness of the data and code shared in articles (rtransparent)</i>
------------------	---

Description

Bridges **rtransparent** and **rfair**: takes the data/code identifiers **rtransparent** extracts from articles (its `open_data_links` and `open_code_links` columns) and scores each against the FAIR metrics. Data identifiers are scored with the FsF data metrics and code repositories with the FRSM software metrics.

Usage

```
assess_data_code(
  x,
  id_col = NULL,
  data_metric_version = "0.8",
  code_metric_version = "0.7_software",
  data_col = "open_data_links",
  code_col = "open_code_links",
  sep = " ; ",
  quiet = FALSE,
  workers = 1L,
```

```

    keep = FALSE,
    previous = NULL,
    ...
)

```

Arguments

<code>x</code>	One of: a data frame from <code>rtransparent::rt_data_code_pmc()</code> / <code>rt_all_pmc()</code> (with <code>open_data_links</code> / <code>open_code_links</code> columns); a named list with those elements; or a character vector of " ; "-joined data-link strings.
<code>id_col</code>	Optional name of a column in <code>x</code> identifying the source article (e.g. "pmid" or "doi"); used to label each result.
<code>data_metric_version</code>	Metric version for data identifiers (default "0.8").
<code>code_metric_version</code>	Metric version for code repositories (default "0.7_software").
<code>data_col, code_col</code>	Column/element names holding the joined links (defaults match <code>rtransparent</code> : "open_data_links", "open_code_links").
<code>sep</code>	Separator <code>rtransparent</code> uses to join identifiers (default " ; ").
<code>quiet</code>	If FALSE (default), print per-identifier progress.
<code>workers, keep</code>	See assess_fair_batch() . With <code>keep = TRUE</code> the "assessments" attribute holds the objects for data and code together.
<code>previous</code>	Optional result of an earlier <code>assess_data_code()</code> call, to resume an interrupted run (see assess_fair_batch()).
<code>...</code>	Passed to assess_fair() .

Value

A data frame with one row per (article, kind, identifier): source (article id), kind ("data" or "code"), and the columns of [assess_fair_batch\(\)](#). Each unique identifier is assessed once.

See Also

[assess_fair_batch\(\)](#), [split_identifiers\(\)](#), [assess_fair\(\)](#)

Examples

```

assess_data_code(list(open_data_links = "https://doi.org/10.5281/zenodo.8347772",
                     open_code_links = "https://github.com/pangaea-data-publisher/fuji"))

```

assess_fair	<i>Assess the FAIRness of a research data object.</i>
-------------	---

Description

Resolves a persistent identifier or URL, harvests its metadata, and scores it against the FAIRsFAIR metrics, entirely in R.

Usage

```
assess_fair(
  id,
  metric_version = "0.8",
  use_datacite = TRUE,
  metadata_service_endpoint = NULL,
  metadata_service_type = metadata_service_types(),
  test_debug = FALSE,
  resolve = TRUE,
  timeout = 15,
  use_headless = FALSE,
  max_time = Inf
)
```

Arguments

id	A persistent identifier or URL (DOI, Handle, ARK, URN, ...).
metric_version	Metric version to use (see rfair_metric_versions()).
use_datacite	Whether to query DataCite for registry metadata.
metadata_service_endpoint	Optional URL of an additional metadata document to harvest, or a ready protocol query URL (for example an OAI-PMH GetRecord URL, an OGC CSW GetRecordById URL, a SPARQL query URL, or a DCAT / schema.org JSON-LD / RO-Crate / DataCite / Crossref / CKAN document). The response is parsed with the same format-gated collectors used for content negotiation, so only a recognized metadata document contributes.
metadata_service_type	Type hint for metadata_service_endpoint. "schema_org" is harvested as JSON-LD; the others are tried as an XML metadata document, then RDF.
test_debug	If TRUE, collect debug log messages in the result.
resolve	If TRUE, resolve the identifier to its landing page.
timeout	Per-request timeout in seconds.
use_headless	If TRUE and the optional chromote package is installed, render JavaScript-heavy landing pages with a headless browser before harvesting embedded metadata.
max_time	Time budget for the whole assessment, in seconds. Metadata sources not yet harvested when it runs out are skipped and listed in harvest_errors. The default, Inf, has no limit.

Value

A [fair_assessment](#) object.

Examples

```
a <- assess_fair("https://doi.org/10.5281/zenodo.8347772")
summary(a)
```

assess_fair_batch	<i>Assess the FAIRness of a batch of identifiers</i>
-------------------	--

Description

Runs [assess_fair\(\)](#) over a vector of identifiers and returns one tidy row per identifier (deduplicated). Failures are captured in an error column rather than aborting the batch.

Usage

```
assess_fair_batch(
  ids,
  metric_version = "0.8",
  quiet = FALSE,
  workers = 1L,
  keep = FALSE,
  previous = NULL,
  ...
)
```

Arguments

<code>ids</code>	Character vector of DOIs, PIDs, URLs, or identifiers.org codes.
<code>metric_version</code>	Metric version (see rfair_metric_versions()).
<code>quiet</code>	If FALSE (default), print per-identifier progress.
<code>workers</code>	Number of identifiers to assess at once. Values above 1 fork worker processes with parallel::mclapply() , which is not available on Windows (there the batch runs serially with a warning). HTTP requests to one host stay rate limited per process (see <code>options(rfair.rate_per_host)</code>).
<code>keep</code>	If TRUE, keep the full fair_assessment objects, named by identifier, in the "assessments" attribute of the result.
<code>previous</code>	Optional result of an earlier <code>assess_fair_batch()</code> call. Identifiers it already scored without an error (for the same metric version) are reused instead of assessed again, so an interrupted batch can be resumed.
<code>...</code>	Passed to assess_fair() .

Value

A data frame with one row per unique identifier: identifier, metric_version, scheme, is_persistent, resolved (did the identifier resolve; NA when resolve = FALSE), http_status, resolved_url, fair_percent, F, A, I, R, maturity, n_pass, n_metrics, error.

See Also

[assess_data_code\(\)](#), [assess_fair\(\)](#)

Examples

```
res <- assess_fair_batch(c("https://doi.org/10.5281/zenodo.8347772", "geo:GSE12345"),
  keep = TRUE)
attr(res, "assessments")[[1]]
```

as_fuji_json

Convert a FAIR assessment to F-UJI-compatible JSON.

Description

Produces a payload matching the upstream F-UJI FAIRResults schema, so the output can be consumed by tools built for the F-UJI service.

Usage

```
as_fuji_json(x, pretty = TRUE)
```

Arguments

x	A fair_assessment object.
pretty	Whether to pretty-print the JSON.

Value

A JSON string (class json).

Examples

```
a <- assess_fair("https://doi.org/10.5281/zenodo.8347772")
cat(as_fuji_json(a))
```

as_rdf	<i>Serialize a FAIR assessment to RDF (DQV + schema.org Rating + FTR).</i>
--------	--

Description

Emits the assessment as W3C Data Quality Vocabulary quality measurements (one per FAIR category and one per metric) plus a schema.org Rating, the machine-readable form the F-UJI service publishes, and one FAIR Test Result per metric test using the OStrails FAIR Testing Resource vocabulary (<https://w3id.org/ftr/>, version 1.3.0): prov:value pass or fail, ftr:completion, the evidence as ftr:log, and for failed tests the [fair_recommendations\(\)](#) action as an ftr:suggestion.

Metrics and tests are identified within the metric specification (for example <https://doi.org/10.5281/zenodo.6461229#F>

Usage

```
as_rdf(x, format = c("jsonld", "turtle"))
```

Arguments

x	A fair_assessment object.
format	"jsonld" (default) or "turtle" (needs the optional rdflib and jsonld packages).

Value

A character scalar of serialized RDF.

Examples

```
a <- assess_fair("https://doi.org/10.5281/zenodo.8347772")
cat(as_rdf(a))
```

classify_access	<i>Classify the access level and sensitivity of a data object.</i>
-----------------	--

Description

Classify the access level and sensitivity of a data object.

Usage

```
classify_access(access_level = NULL, urls = NULL, source = NULL)
```

Arguments

access_level	Access codes/URIs harvested from metadata (character).
urls	Landing-page and content URLs (for host-based detection).
source	Optional source name/id.

Value

A list with access (public/embargoed/restricted/closed/ metadataonly/unknown), controlled_access, sensitive, the matched reusabledata record (or NULL), and a human-readable note.

Examples

```
classify_access(access_level = "info:eu-repo/semantics/openAccess")$access
```

fair4rs_principles	<i>The FAIR Principles for Research Software (FAIR4RS).</i>
--------------------	---

Description

The canonical FAIR4RS (sub)principles that rfair's software metrics (the FRSM metric set, `metric_version = "0.7_software"`) operationalize. Principle statements are reproduced from the FAIR4RS Principles version 1.0.

Usage

```
fair4rs_principles(category = NULL)
```

Arguments

category	Optional filter: one or more of "F", "A", "I", "R".
----------	---

Value

A data frame with id, category, statement (the principle text), and explanation. The four foundational F/A/I/R statements and the source citation are attached as the "foundational" and "source" attributes.

References

Chue Hong, N. P., Katz, D. S., Barker, M., Lamprecht, A.-L., Martinez, C., Psomopoulos, F. E., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., Honeyman, T., et al. (2022). FAIR Principles for Research Software (FAIR4RS Principles) (1.0). Research Data Alliance. doi:10.15497/RDA00068

See Also

[fair_principles\(\)](#) for the data FAIR principles.

Examples

```
fair4rs_principles()
fair4rs_principles("R")
```

fair_assessment	<i>The fair_assessment object</i>
-----------------	-----------------------------------

Description

[assess_fair\(\)](#) returns an object of class `fair_assessment`. It has `print()`, `format()`, [summary\(\)](#), and [as.data.frame\(\)](#) methods, and can be exported with [as_fuji_json\(\)](#) and [as_rdf\(\)](#).

Details

Useful list elements: `summary` (F/A/I/R scores), `results` (per-metric), `metadata` (harvested), `reuse` (license reusability), `access` (access/sensitivity), `identifier_hygiene`, `resolution` (the identifier's resolution attempt: `url`, `final_url`, HTTP status, `ok`, `error`), and `harvest_errors` (metadata sources that failed, for example a GitHub rate limit), and `software` (the repository signals the FRSM software metrics score, when a code repository was harvested), and `reference_data` (when and from which F-UJI release the bundled reference tables were built). `resolved_url` is NA when the identifier did not resolve.

See Also

```
assess\_fair\(\)
```

fair_compare	<i>Compare two FAIR assessments</i>
--------------	-------------------------------------

Description

Joins two assessments, for example of the same record before and after a metadata fix, or under two metric versions, and reports what changed.

Usage

```
fair_compare(a, b, level = c("metric", "test"))
```

Arguments

- `a, b` [fair_assessment](#) objects; `a` is the baseline.
- `level` "metric" (default) for one row per metric, or "test" for one row per metric test.

Value

A data frame with the identifiers, earned_a, earned_b, status_a, status_b, delta (earned_b - earned_a), and change ("improved", "worse", "same", or "only in a"/"only in b"), in the metric order of a.

See Also

[fair_recommendations\(\)](#)

Examples

```
data(fair_example)
fixed <- fair_example
fixed$results[[1]]$score$earned <- fixed$results[[1]]$score$total
fair_compare(fair_example, fixed)[1:3, ]
```

fair_example	<i>An example FAIR assessment</i>
--------------	-----------------------------------

Description

A stored [fair_assessment](#) object, produced by running [assess_fair\(\)](#) on a stable Zenodo deposit ([doi:10.5281/zenodo.8347772](https://doi.org/10.5281/zenodo.8347772)). It is bundled so the plotting examples and the vignette("illustrating-fairness") can run offline and reproducibly, without contacting any network service.

Usage

```
data(fair_example)
```

Format

A fair_assessment object (a list with S3 class fair_assessment); see [fair_assessment](#) for its structure.

Details

The verbose per-test debug log has been stripped to keep the installed size small; all elements used by the print, summary, as.data.frame, [plot](#), [as_fuji_json\(\)](#), and [as_rdf\(\)](#) methods are retained.

Source

[assess_fair\(\)](#) on [doi:10.5281/zenodo.8347772](https://doi.org/10.5281/zenodo.8347772), rebuilt by data-raw/06-build-example-assessment.R.

See Also

[assess_fair\(\)](#), [plot.fair_assessment\(\)](#)

Examples

```
data(fair_example)
summary(fair_example)
plot(fair_example)
```

fair_principles	<i>The canonical FAIR (sub)principles.</i>
-----------------	--

Description

The canonical FAIR (sub)principles.

Usage

```
fair_principles(category = NULL)
```

Arguments

category Optional filter: one or more of "F", "A", "I", "R".

Value

A data frame with id, label, category, definition, and uri (the w3id.org/fair/principles term URI).

Examples

```
fair_principles()
fair_principles("R")
```

fair_recommendations	<i>Recommend how to fix the failed tests of a FAIR assessment</i>
----------------------	---

Description

Lists every metric test that did not pass, with one concrete action a data curator or software maintainer can take to pass it (for example "Add a license to the metadata as a URL or SPDX identifier"). Actions come from a curated table covering the FsF data metrics and the FRSM software metrics; tests of legacy metric versions fall back to a metric-level action.

Usage

```
fair_recommendations(x)
```

Arguments

x A [fair_assessment](#) from [assess_fair\(\)](#).

Value

A data frame, most valuable fixes first, with columns `metric_identifier`, `test_identifier`, `test_name`, `points` (the score passing the test would add, given the metric's cap), and `recommendation`.

See Also

[fair_compare\(\)](#) to check the effect of a fix.

Examples

```
data(fair_example)
head(fair_recommendations(fair_example))
```

 fair_tlc

FAIR-TLC indicators (Traceable, Licensed, Connected)

Description

Computes the three "FAIR+" indicators proposed by Haendel and colleagues in the Monarch Initiative / NCATS TransMed response to the NIH RFI on biomedical repository value metrics ([doi:10.5281/zenodo.203295](https://doi.org/10.5281/zenodo.203295)), building on the (Re)usable Data Project ([doi:10.1371/journal.pone.0213090](https://doi.org/10.1371/journal.pone.0213090)). They extend FAIR with the provenance and legal dimensions that automated FAIR tools usually miss: whether data is **Traceable** (provenance, attribution), **Licensed** (clearly documented and actually reusable), and **Connected** (qualified links to related entities).

Usage

```
fair_tlc(x)
```

Arguments

`x` A [fair_assessment](#) from [assess_fair\(\)](#).

Value

A data frame with columns `dimension`, `indicator`, `met` (logical), and `detail`, plus a "source" attribute citing the framework.

Examples

```
a <- assess_fair("https://doi.org/10.5281/zenodo.8347772")
fair_tlc(a)
```

frsm_agreement

Agreement between rfair's FRSM scores and expert ratings

Description

rfair scores the FRSM software metrics from repository signals (file names, API fields, metadata files). These heuristics have not yet been validated against expert judgement, so every FRSM result carries `evidence_type = "heuristic"`. `frsm_agreement()` supports that validation: raters judge each FRSM test for a sample of repositories (start from the template in `system.file("extdata", "frsm_validation_template.csv", package = "rfair")`), and the function compares rfair's pass/fail result with the raters' majority judgement, test by test.

Usage

```
frsm_agreement(assessments, ratings)
```

Arguments

<code>assessments</code>	A list of fair_assessment objects scored with an FRSM metric version (for example the "assessments" attribute of <code>assess_fair_batch(..., metric_version = "0.7_software", keep = TRUE)</code>), or a single assessment.
<code>ratings</code>	A data frame with columns <code>identifier</code> (as passed to assess_fair()), <code>test_identifier</code> (for example "FRSM-14-R1-2"), <code>rater</code> , and <code>passed</code> (logical; did the software satisfy the test?).

Value

A data frame with one row per test: `test_identifier`, `n` (rated items), `rfair_pass` and `rater_pass` (pass rates), `agreement` (share of items where rfair matches the raters' majority), `kappa` (Cohen's kappa of rfair against the majority), and `rater_kappa` (kappa between the first two raters, when two or more rated the test). Ties between raters are dropped.

Examples

```
data(fair_example)
ratings <- data.frame(identifier = fair_example$id,
                      test_identifier = "FsF-F1-01MD-1", rater = "A", passed = TRUE)
# fair_example is a data assessment, so this only shows the shape of the output
frsm_agreement(fair_example, ratings)
```

identifier_hygiene	<i>Check an identifier against best-practice / hygiene heuristics.</i>
--------------------	--

Description

Check an identifier against best-practice / hygiene heuristics.

Usage

```
identifier_hygiene(id)
```

Arguments

id	A persistent identifier or URL.
----	---------------------------------

Value

A list with identifier, scheme, is_persistent, hygiene_ok, and a character vector of issues.

Examples

```
identifier_hygiene("RRID:MGI:5577054")$issues
identifier_hygiene("https://doi.org/10.5281/zenodo.8347772")$hygiene_ok
```

id_parse	<i>Parse a persistent identifier or URL.</i>
----------	--

Description

Resolves the identifier scheme, normalizes it, and constructs its resolver URL, mirroring IdentifierHelper in F-UJI.

Usage

```
id_parse(idstring)
```

Arguments

idstring	A DOI, Handle, ARK, URN, UUID, identifiers.org PID, or URL.
----------	---

Value

A list with identifier, normalized_id, identifier_url, preferred_schema, identifier_schemes, and is_persistent.

Examples

```
id_parse("https://doi.org/10.5281/zenodo.8347772")$preferred_schema
```

launch_rfair	<i>Launch the rfair Shiny app</i>
--------------	-----------------------------------

Description

Opens an interactive app to assess the FAIRness of a research data object and explore the per-metric results, license reusability, access/sensitivity, and identifier hygiene.

Usage

```
launch_rfair(...)
```

Arguments

... Passed to `shiny::runApp()`.

Value

Runs the app (called for its side effect); invisibly NULL.

Examples

```
if (interactive()) {
  launch_rfair()
}
```

license_reuse	<i>Assess the reuse permissions granted by a license.</i>
---------------	---

Description

Goes beyond detecting that a license exists: classifies whether it actually permits redistribution, commercial use, and derivative works, and whether it meets the Open Definition. Useful for judging real reusability of data.

Usage

```
license_reuse(license)
```

Arguments

license A license name, SPDX id, or URL (e.g. from an assessment).

Value

A list describing the license's reuse terms, including `is_open`, `permits_redistribution`, `permits_commercial`, `permits_derivatives`, `requires_attribution`, `requires_share_alike`, `category`, and `note`.

Examples

```
license_reuse("https://creativecommons.org/licenses/by-nc-nd/4.0/")$is_open
license_reuse("CC-BY-4.0")$is_open
```

```
plot.fair_assessment
```

Plot a FAIR assessment as a scorecard

Description

Draws a compact, readable scorecard of a [fair_assessment](#) using base graphics (no extra package dependencies). It is the quickest way to *see* an assessment: a horizontal progress bar per FAIR category (or per metric), each annotated with its score and CMMI maturity level. See `vignette("illustrating-fairness")` for worked examples.

Usage

```
## S3 method for class 'fair_assessment'
plot(
  x,
  type = c("category", "metric", "sunburst"),
  colors = .fair_cat_colors,
  show_maturity = (match.arg(type) == "category"),
  main = NULL,
  ...
)
```

Arguments

<code>x</code>	A <code>fair_assessment</code> object returned by <code>assess_fair()</code> .
<code>type</code>	What to draw. "category" (default) draws one bar per FAIR category (Findable, Accessible, Interoperable, Reusable) plus the overall score; "metric" draws one bar per individual metric, grouped and colored by category; "sunburst" draws a concentric sunburst (an inner ring of the F/A/I/R categories and an outer ring of the individual metrics, each filled in proportion to its score) with the overall FAIR percentage in the center.
<code>colors</code>	Named character vector of category fill colors, with names "F", "A", "I", "R".
<code>show_maturity</code>	Logical; annotate each bar with its maturity level. Defaults to TRUE for type = "category".
<code>main</code>	Title. Defaults to the resolved identifier (or the input id).
<code>...</code>	Ignored (for S3 method compatibility).

Value

`x`, invisibly. Called for the side effect of drawing a plot.

See Also

[assess_fair\(\)](#), [summary.fair_assessment\(\)](#), [fair_example](#)

Examples

```
# A stored example assessment (no network needed):
data(fair_example)
plot(fair_example)
plot(fair_example, type = "metric")
plot(fair_example, type = "sunburst")
```

`principle_definition` *Canonical definition of the FAIR principle a metric maps to.*

Description

For data metrics (FsF-*) this returns the FAIR Guiding Principle definition; for software metrics (FRSM-*) it returns the corresponding FAIR4RS Principle statement (see [fair4rs_principles\(\)](#)).

Usage

```
principle_definition(metric_identifier)
```

Arguments

`metric_identifier`
A metric identifier (e.g. "FsF-F1-01MD" or "FRSM-17-R1.2").

Value

The principle's definition string, or NA.

Examples

```
principle_definition("FsF-R1.1-01M")
principle_definition("FRSM-17-R1.2")
```

reusabledata_rating	<i>Look up a (Re)usable Data Project curation for a source.</i>
---------------------	---

Description

Look up a (Re)usable Data Project curation for a source.

Usage

```
reusabledata_rating(urls = NULL, source = NULL)
```

Arguments

urls	Character vector of URLs (e.g. landing page, content URLs).
source	Optional source name or id to match.

Value

The matched curation record (list) or NULL.

Examples

```
reusabledata_rating(source = "dbgap")$license_type
```

rfair_metric_versions	<i>List the metric versions bundled with rfair.</i>
-----------------------	---

Description

List the metric versions bundled with rfair.

Usage

```
rfair_metric_versions()
```

Value

Character vector of available metric versions (e.g. "0.8").

Examples

```
rfair_metric_versions()
```

split_identifiers	<i>Split a joined identifier string into individual identifiers.</i>
-------------------	--

Description

rtransparent joins the data/code identifiers it extracts with " ; ". This splits such a string (or a vector of them) into a trimmed character vector, dropping empties. rfair's `id_parse()` already understands the forms it emits (doi.org URLs, repository URLs, and identifiers.org prefix:accession codes such as geo:GSE123 or bioproject:PRJEB123).

Usage

```
split_identifiers(x, sep = " ; ")
```

Arguments

x	A character vector of identifier strings (each possibly joined).
sep	Separator used to join identifiers (default " ; ").

Value

A character vector of individual identifiers.

Examples

```
split_identifiers("https://doi.org/10.5061/dryad.x ; geo:GSE12345")
```

summary.fair_assessment	<i>Summarize a FAIR assessment as an F/A/I/R score table.</i>
-------------------------	---

Description

Summarize a FAIR assessment as an F/A/I/R score table.

Usage

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

Arguments

object	A fair_assessment object.
...	Ignored.

Value

A data frame with earned, total, percent, and maturity per FAIR category and overall.

Index

- * **datasets**
 - fair_example, 11
- as.data.frame(), 10
- as.data.frame.fair_assessment, 3
- as_fuji_json, 7
- as_fuji_json(), 10, 11
- as_rdf, 8
- as_rdf(), 10, 11
- assess_data_code, 3
- assess_data_code(), 7
- assess_fair, 5
- assess_fair(), 4, 6, 7, 10–14, 17, 18
- assess_fair_batch, 6
- assess_fair_batch(), 4
- classify_access, 8
- fair4rs_principles, 9
- fair4rs_principles(), 18
- fair_assessment, 6–8, 10, 10, 11–14, 17
- fair_compare, 10
- fair_compare(), 13
- fair_example, 11, 18
- fair_principles, 12
- fair_principles(), 9
- fair_recommendations, 12
- fair_recommendations(), 8, 11
- fair_tlc, 13
- frsm_agreement, 14
- id_parse, 15
- id_parse(), 20
- identifier_hygiene, 15
- launch_rfair, 16
- license_reuse, 16
- parallel::mclapply(), 6
- plot, 11
- plot.fair_assessment, 17
- plot.fair_assessment(), 11
- principle_definition, 18
- reusabledata_rating, 19
- rfair_metric_versions, 19
- rfair_metric_versions(), 5, 6
- shiny::runApp(), 16
- split_identifiers, 20
- split_identifiers(), 4
- summary(), 10
- summary.fair_assessment, 20
- summary.fair_assessment(), 18