Package: rfair (via r-universe)

July 2, 2026

Title Assess the FAIRness of Research Data Objects and Software

Version 0.1.0

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>. 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> 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 digest, httr2, jsonlite, mime, rvest, stats, stringdist, utils, xml2, yaml

Suggests bslib, chromote, covr, DT, httptest2, jqr, knitr, plumber, rdflib, rmarkdown, shiny, testthat (>= 3.0.0), wand

Config/testthat/edition 3

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Ahmad Sofi-Mahmudi [aut, cre] (ORCID:
<https://orcid.org/0000-0001-6829-0823>), Steffen Neumann [ctb]
(Author of the original rfuji 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 https://cran.r-universe.dev

Date/Publication 2026-07-01 09:10:02 UTC

RemoteUrl https://github.com/cran/rfair

RemoteRef HEAD

RemoteSha 024de049d3547c75dbf39905170a6fc838b0fd33

Contents

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.

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 *Serialize a FAIR assessment to RDF (DQV + schema.org Rating).*

Description

Emits the assessment as W3C Data Quality Vocabulary quality measurements plus a schema.org Rating, the machine-readable form the F-UJI service publishes.

Usage

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

Arguments

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

Value

A character scalar of serialized RDF.

Examples

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

assess_data_code *Assess the FAIRness of the data and code shared in articles (rtransparent)*

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,
  ...
)
```

Arguments

x	One of: a data frame from `rtransparent::rt_data_code_pmc()` / `rt_all_pmc()` (with `open_data_links` / `open_code_links` columns); a named list with those elements; or a character vector of `" ; "`-joined data-link strings.
id_col	Optional name of a column in `x` identifying the source article (e.g. `"pmid"` or `"doi"`); used to label each result.
data_metric_version	Metric version for data identifiers (default `"0.8"`).
code_metric_version	Metric version for code repositories (default `"0.7_software"`).
data_col, code_col	Column/element names holding the joined links (defaults match rtransparent: `"open_data_links"`, `"open_code_links"`).
sep	Separator rtransparent uses to join identifiers (default `" ; "`).
quiet	If `FALSE` (default), print per-identifier progress.
...	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` *Assess the FAIRness of a research data object.*

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
)
```

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.

Value

A fair_assessment object.

Examples

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

`assess_fair_batch` *Assess the FAIRness of a batch of identifiers*

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, ...)
```

Arguments

`ids`	Character vector of DOIs, PIDs, URLs, or identifiers.org codes.
`metric_version`	Metric version (see `rfair_metric_versions()`).
`quiet`	If `FALSE` (default), print per-identifier progress.
`...`	Passed to `assess_fair()`.

Value

A data frame with one row per unique identifier: `identifier`, `metric_version`, `scheme`, `is_persistent`, `resolved_url`, `fair_percent`, `F`, `A`, `I`, `R`, `maturity`, `n_pass`, `n_metrics`, `error`.

See Also

`assess_data_code()`, `assess_fair()`

Examples

```
assess_fair_batch(c("https://doi.org/10.5281/zenodo.8347772", "geo:GSE12345"))
```

classify_access *Classify the access level and sensitivity of a data object.*

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
```

fair_assessment *The* fair_assessment *object*

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), and identifier_hygiene.

See Also

assess_fair()

`fair_example` *An example FAIR assessment*

Description

A stored fair_assessment object, produced by running `assess_fair()` on a stable Zenodo deposit (doi: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, 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 *The canonical FAIR (sub)principles.*

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_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), building on the (Re)usable Data Project (doi: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)
```

`fair4rs_principles` *The FAIR Principles for Research Software (FAIR4RS).*

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")
```

id_parse *Parse a persistent identifier or URL.*

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
```

identifier_hygiene *Check an identifier against best-practice / hygiene heuristics.*

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
```

launch_rfair *Launch the rfair Shiny app*

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 *Assess the reuse permissions granted by a license.*

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

`x`	A `fair_assessment` object returned by `assess_fair()`.
`type`	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.
`colors`	Named character vector of category fill colors, with names `"F"`, `"A"`, `"I"`, `"R"`.
`show_maturity`	Logical; annotate each bar with its maturity level. Defaults to `TRUE` for `type = "category"`.
`main`	Title. Defaults to the resolved identifier (or the input id).
`...`	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 *Look up a (Re)usable Data Project curation for a source.*

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 *List the metric versions bundled with rfair.*

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 *Split a joined identifier string into individual identifiers.*

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

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