Package 'glcdp'

August 6, 2026

Title Discover, Access, and Import Global Light Commons Data Packages

Version 1.0.0

Description Discovers Global Light Commons data packages through their registry, opens immutable passing revisions, and provides searchable inventories of package metadata. Selected metadata and measurement files can be downloaded or imported with metadata-defined columns, types, factor levels, date-time values, and time zones. 'Git Large File Storage' objects are resolved without requiring an external 'Git LFS' installation, and imported file groups can be explicitly collected into data suitable for personal light exposure analysis workflows. An included 'shiny' application supports interactive discovery, inspection, selection, preview, and reproducible handoff to 'R'.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports cli, digest, dplyr, httr2, jsonlite, lubridate, readr, tibble

Suggests bslib (>= 0.6.0), knitr, rmarkdown, shiny (>= 1.7.2), testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://tscnlab.github.io/glc-dp-r/, https://github.com/tscnlab/glc-dp-r

BugReports https://github.com/tscnlab/glc-dp-r/issues

Config/Needs/website pkgdown

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Johannes Zauner [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-2171-4566>), Salma M. Thalji [aut, cph] (ORCID: <https://orcid.org/0009-0002-7320-9289>), Manuel Spitschan [aut, cph] (ORCID: <https://orcid.org/0000-0002-8572-9268>)

Maintainer Johannes Zauner <johannes.zauner@tum.de>

Repository CRAN

Date/Publication 2026-08-06 13:00:02 UTC

Contents

`add_metadata` *Add metadata to imported data*

Description

Extracts requested metadata with `extract_metadata()` and joins it onto every matching observation in an imported dataset.

Usage

```
add_metadata(
  dataset,
  metadata,
  fields,
  by = "file_group_id",
  resource = NULL,
  overwrite = FALSE
)
```

Arguments

dataset	A data frame containing imported observations.
metadata	A metadata data frame, a local CSV or TSV path, or a package opened with glc_open().
fields	One or more exact, top-level metadata column names to select.
by	One common identifier column, or a named character mapping from the dataset column to the metadata column. The default is "file_group_id". Use "Id" for explicitly dataset-level extraction, or c(Id = "dataset_internal_id") for a differently named metadata key.
resource	An optional declared resource name when metadata is a glc_package. For file-group or dataset identifiers, omitting resource searches declared resources connected through the package's file-group, dataset, participant, study, and device relationships. Each requested field must resolve to exactly one connected resource. For other by mappings, exactly one declared resource must contain the metadata join column and a requested field.
overwrite	Replace existing dataset columns that have the same names as extracted metadata fields. The default is FALSE.

Value

add_metadata() returns the original dataset with the requested metadata columns added. Row order, row count, and dplyr grouping are preserved.

Examples

```
dataset <- tibble::tibble(
  file_group_id = c("DS1:1", "DS1:1", "DS2:1"),
  value = c(1, 2, 3)
)
metadata <- tibble::tibble(
  file_group_id = c("DS1:1", "DS2:1"),
  condition = c("control", "intervention")
)

add_metadata(dataset, metadata, fields = "condition")
```

---

extract_metadata *Extract metadata for imported data*

Description

Selects requested metadata fields for the identifiers represented in an imported dataset. By default, the result contains one row per unique file group and can be joined back with add_metadata().

Usage

```
extract_metadata(
  dataset,
  metadata,
  fields,
  by = "file_group_id",
  resource = NULL
)
```

Arguments

dataset	A data frame containing imported observations.
metadata	A metadata data frame, a local CSV or TSV path, or a package opened with glc_open().
fields	One or more exact, top-level metadata column names to select.
by	One common identifier column, or a named character mapping from the dataset column to the metadata column. The default is "file_group_id". Use "Id" for explicitly dataset-level extraction, or c(Id = "dataset_internal_id") for a differently named metadata key.
resource	An optional declared resource name when metadata is a glc_package. For file-group or dataset identifiers, omitting resource searches declared resources connected through the package's file-group, dataset, participant, study, and device relationships. Each requested field must resolve to exactly one connected resource. For other by mappings, exactly one declared resource must contain the metadata join column and a requested field.

Details

Identifiers are compared as character values, while the identifier column in the returned table retains its original class. Metadata-only identifiers are ignored. If metadata is a glc_package, the default file-group link may assemble fields from the linked dataset, participant, study, and device records. Dataset-level fields therefore repeat across file groups. Missing participant, study, or device links are retained as missing values with a warning. Use by = "Id" for one row per dataset; device fields then error when a dataset is linked to multiple devices.

For a grouped dataset, the grouping columns are placed before the extraction key and the original dplyr grouping (including its .drop setting) is retained. Each extraction-key value must map to exactly one combination of grouping-column values.

If only some input identifiers match, unmatched identifiers are retained with missing metadata and a warning is issued. If only some fields exist, the missing fields are omitted and a warning is issued.

Custom metadata are best stored in a declared data-package resource, for example data/metadata.csv, rather than discovered from the working directory or neighboring files.

Value

extract_metadata() returns a tibble with one row per unique value of by in first-occurrence order. The dataset's dplyr grouping columns and grouping are retained, with the by column added when it is not already a grouping column. With the default by, this is one row per file group.

Examples

```
dataset <- tibble::tibble(
  Id = c("DS1", "DS1", "DS2"),
  file_group_id = c("DS1:1", "DS1:1", "DS2:1"),
  value = c(1, 2, 3)
) |>
  dplyr::group_by(Id)
metadata <- tibble::tibble(
  file_group_id = c("DS1:1", "DS2:1"),
  condition = c("control", "intervention")
)

extract_metadata(dataset, metadata, fields = "condition")

if (interactive()) {
package <- glc_open("owner/repository")
imported <- glc_read(package, dataset_id = "DS1") |>
  glc_collect()
extract_metadata(
  imported,
  package,
  fields = c("participant_age", "study_title", "device_model")
)
extract_metadata(imported, package, "dataset_timezone", by = "Id")
}
```

`glc_collect` *Collect compatible file groups*

Description

Explicitly combines file-group tibbles after checking their columns, types, factor-level contracts, time zones, modalities, roles, data states, and relationship consistency. Multiple non-missing device links within one dataset are rejected.

Usage

```
glc_collect(x, standardize = c("lightlogr", "none"))
```

Arguments

`x`	A collection returned by `glc_read()`.
`standardize`	Either `"lightlogr"` to add the conventional `Id`, `file_group_id`, `participant_Id`, `Datetime`, and `file.name` columns and remove internal `.glc_*` provenance columns, or `"none"` to retain source and provenance columns unchanged.

Value

A combined tibble. In LightLogR-standardized output, `Id` contains the dataset id, `file_group_id` identifies the source file group, `participant_Id` contains the participant id, an existing source `file.name` column is retained, and the result is grouped by `Id`.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
collection <- glc_read(
  iztech,
  dataset_id = "MELIDOS_IZTECH_S001",
  file_group = "MELIDOS_IZTECH_S001:17",
  n_max = 10
)
glc_collect(collection)
```

---

glc_datasets *Inventory datasets*

Description

Inventory datasets

Usage

```
glc_datasets(x, dataset_id = NULL)
```

Arguments

`x`	A package opened with `glc_open()`.
`dataset_id`	Optional dataset id or ids.

Value

A tibble with one row per dataset.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
glc_datasets(iztech)
```

`glc_download` *Download package metadata or data*

Description

Downloads only descriptor-declared and dataset-referenced content, preserves repository-relative paths, and writes a reproducibility manifest.

Usage

```
glc_download(
  x,
  dest_dir,
  include = c("metadata", "data", "all"),
  dataset_id = NULL,
  file_group = NULL,
  resources = NULL,
  files = NULL,
  overwrite = FALSE
)
```

Arguments

`x`	A package opened with `glc_open()`.
`dest_dir`	Destination directory.
`include`	One of `"metadata"`, `"data"`, or `"all"`. Metadata is the safe default.
`dataset_id`	Optional dataset id selection for data downloads.
`file_group`	Optional file-group selection.
`resources`	Optional descriptor resource names.
`files`	Optional exact paths, declared paths, or basenames.
`overwrite`	Whether existing files may be replaced.

Value

A tibble recording downloaded paths, storage, size, and hashes.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
destination <- tempfile("glcdp-metadata-")
glc_download(iztech, destination)
```

`glc_explore` *Explore Global Light Commons data packages*

Description

Launches a local Shiny application for browsing the GLC registry, opening the immutable latest passing revision of a package, reviewing its contents, and filtering participants, devices, datasets, file groups, semantic terms, and source variables. The app can preview the resulting selection and export an annotated, reproducible R script without uploading package data to another service.

Usage

```
glc_explore(
  registry = NULL,
  launch.browser = getOption("shiny.launch.browser", interactive()),
  ...
)
```

Arguments

`registry`	Optional registry JSON URL or local path. Defaults to the official registry or the value of option `glcdp.registry_url`.
`launch.browser`	Whether to open the application in a browser, or a function that Shiny calls with the application URL. The default respects IDE viewer functions supplied through `shiny.launch.browser`.
`...`	Additional arguments passed to `shiny::runApp()`.

Value

Called for its side effect of running a Shiny application.

See Also

The Shiny app workflow.

Examples

```
glc_explore()
```

`glc_files` *Inventory declared data files*

Description

Inventory declared data files

Usage

```
glc_files(
  x,
  dataset_id = NULL,
  file_group = NULL,
  role = NULL,
  modality = NULL,
  available = NULL
)
```

Arguments

`x`	A package opened with `glc_open()`.
`dataset_id`	Optional dataset id or ids.
`file_group`	Optional group index or stable `dataset:group` id.
`role`	Optional file-group role.
`modality`	Optional modality.
`available`	Optional logical filter for file availability.

Value

A tibble with one row per concrete declared file, including the file-specific encoding declared by its file group.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
glc_files(iztech, dataset_id = "MELIDOS_IZTECH_S001")
```

glc_metadata *Load metadata resources*

Description

Loads core metadata by default. Additional descriptor resources can be requested explicitly by name.

Usage

```
glc_metadata(x, resources = NULL)
```

Arguments

x	A package opened with glc_open().
resources	Optional resource names. The default selects declared core metadata resources.

Value

A named list with one element per requested resource. Tabular resources are returned as tibbles; directory resources contain named sub-lists.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
glc_metadata(iztech, resources = "study")
```

glc_open *Open a Global Light Commons data package*

Description

Opens a local package or resolves a GitHub-hosted package at an immutable commit. Registered packages default to their latest passing revision.

Usage

```
glc_open(
  source,
  ref = "latest_pass",
  token = NULL,
  cache_dir = NULL,
  registry = NULL,
  quiet = FALSE
)
```

Arguments

source	Registry id, owner/repository, GitHub URL, local package directory, or local datapackage.json path.
ref	Remote revision: "latest_pass", "current", or an exact 40-character commit SHA.
token	Optional GitHub token. When omitted, GITHUB_PAT and then GITHUB_TOKEN are consulted.
cache_dir	Optional explicit persistent cache directory. The default uses session-temporary storage for remote reads.
registry	Optional registry object, URL, or local JSON path.
quiet	Suppress informational messages. Warnings remain visible.

Value

A glc_package handle.

Examples

```
package <- glc_open("tscnlab/melidos-iztech-glc-dataset")
package
```

---

glc_packages *List registered Global Light Commons data packages*

Description

Downloads and flattens the Global Light Commons registry. Both passing and non-passing current revisions are retained.

Usage

```
glc_packages(registry = glc_default_registry(), refresh = FALSE)
```

Arguments

registry	Registry JSON URL or path. Defaults to the official registry or the value of option glcdp.registry_url.
refresh	Whether to bypass the in-session registry cache.

Value

A glc_registry tibble with one row per registered repository.

Examples

```
packages <- glc_packages()
glc_search_packages("iztech", packages)
```

`glc_read` *Read metadata-described dataset files*

Description

Read metadata-described dataset files

Usage

```
glc_read(
  x,
  dataset_id,
  file_group = NULL,
  files = NULL,
  variables = NULL,
  terms = NULL,
  primary_only = FALSE,
  n_max = Inf,
  problems = c("error", "warn"),
  progress = interactive()
)
```

Arguments

`x`	A package opened with `glc_open()`.
`dataset_id`	Dataset id or ids. Use `"all"` explicitly to read every dataset.
`file_group`	Optional group index or stable id.
`files`	Optional declared paths, resolved paths, or basenames.
`variables`	Optional source variable names.
`terms`	Optional semantic variable terms.
`primary_only`	Select only declared primary variables.
`n_max`	Maximum rows read from each file.
`problems`	Whether metadata mismatches should be errors or warnings.
`progress`	Show progress while files are imported. Defaults to `TRUE` in interactive sessions and `FALSE` otherwise.

Details

When variable filters are used, source columns required to construct datetimes are used internally but omitted unless selected by the filters. Declared files that are absent from a local package subset are skipped. When a local package contains fewer datasets or files than declared, `glc_read()` reports the discrepancy and reads the available files.

Value

A `glc_data_collection` tibble with one data list-column per file group.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
glc_read(
  iztech,
  dataset_id = "MELIDOS_IZTECH_S001",
  file_group = "MELIDOS_IZTECH_S001:17",
  n_max = 10
)
```

`glc_resources` *Inventory data-package resources*

Description

Inventory data-package resources

Usage

```
glc_resources(x)
```

Arguments

`x`	A package opened with `glc_open()`.

Value

A tibble with one row per declared resource path.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
glc_resources(iztech)
```

glc_schema_versions *Report supported GLC schema versions*

Description

Report supported GLC schema versions

Usage

```
glc_schema_versions()
```

Value

A tibble describing support status for each schema version.

Examples

```
glc_schema_versions()
```

glc_search_metadata *Search metadata values or field paths*

Description

Search metadata values or field paths

Usage

```
glc_search_metadata(
  x,
  query,
  resources = NULL,
  fields = NULL,
  fixed = TRUE,
  ignore_case = TRUE,
  search_in = c("values", "fields", "both")
)
```

Arguments

x	A package opened with glc_open().
query	Text or regular expression to search for.
resources	Optional metadata resource names.
fields	Optional exact field names or complete field paths to include.
fixed	Treat query as fixed text rather than a regular expression.

ignore_case	Ignore letter case while matching.
search_in	Where to match query: scalar metadata "values", complete field paths "fields", or "both". The default is "values".

Details

Field searches return the same leaf-level rows as value searches. A field path that contains multiple scalar values therefore produces one row per value. The fields argument can be combined with any search_in mode to restrict which field paths are searched.

Value

A tibble of matching scalar metadata values and their field paths.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
glc_search_metadata(iztech, "Izmir", resources = "study")
```

---

glc_search_packages *Search registered data packages*

Description

Search registered data packages

Usage

```
glc_search_packages(
  query = NULL,
  packages = glc_packages(),
  status = NULL,
  has_pass = NULL
)
```

Arguments

query	Optional fixed, case-insensitive text searched in package ids and repository names.
packages	A registry returned by glc_packages().
status	Optional current validation status or statuses.
has_pass	Optional logical value selecting packages with or without a recorded passing revision.

Value

A filtered glc_registry tibble.

Examples

```
packages <- glc_packages()
glc_search_packages("iztech", packages)
glc_search_packages(packages = packages, status = "pass")
```

`glc_summary` *Summarize a Global Light Commons data package*

Description

Summarize a Global Light Commons data package

Usage

```
glc_summary(x)
```

Arguments

`x`	A package opened with `glc_open()`.

Value

A one-row `glc_summary` tibble. For local packages, declared and locally available dataset, file-group, and file counts are reported separately.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
glc_summary(iztech)
```

`glc_variables` *Inventory and search declared variables*

Description

Inventory and search declared variables

Usage

```
glc_variables(
  x,
  dataset_id = NULL,
  file_group = NULL,
  term = NULL,
  primary = NULL
)
```

Arguments

x	A package opened with glc_open().
dataset_id	Optional dataset id or ids.
file_group	Optional group index or stable id.
term	Optional semantic term or terms.
primary	Optional logical filter for primary variables.

Value

A tibble with one row per declared variable, including its declared type and factor values, labels, and descriptions.

Examples

```
iztech <- glc_open("tscnlab/melidos-iztech-glc-dataset")
glc_variables(
  iztech,
  file_group = "MELIDOS_IZTECH_S001:17",
  primary = TRUE
)
```

Index