Package: rustycogs (via r-universe)

March 7, 2026

Title TIFF Virtualization via Rust

Version 0.1.0.9001

Description Extract byte-range chunk references from cloud-hosted TIFF and COG files without reading pixel data. Uses the Rust 'async-tiff' and 'object_store' crates for async I/O across S3, GCS, Azure, HTTP, and local storage. Returns a data frame of tile offsets and metadata suitable for constructing Kerchunk or Zarr virtual stores.

License MIT + file LICENSE

URL https://github.com/mdsumner/rustycogs

BugReports https://github.com/mdsumner/rustycogs/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

SystemRequirements Cargo (Rust's package manager), rustc

Imports jsonlite

Suggests arrow, testthat (>= 3.0.0)

Config/rextendr/version 0.4.2

Config/testthat/edition 3

Config/pak/sysreqs libclang-dev

Repository https://hypertidy.r-universe.dev

Date/Publication 2026-02-17 03:34:15 UTC

RemoteUrl https://github.com/hypertidy/rustycogs

RemoteRef HEAD

RemoteSha 62d9800af7153896e3e3aefe180bb328ace08b88

Contents

`refs_to_kerchunk` *Convert TIFF references to Kerchunk V1 JSON*

Description

Takes the data frame from `tiff_refs()` and reshapes it into the Kerchunk reference specification (version 1) as a list suitable for writing with `jsonlite::write_json()`.

Usage

```
refs_to_kerchunk(refs, var_name = "data")
```

Arguments

`refs`	Data frame from `tiff_refs()`.
`var_name`	Name for the Zarr variable. Default `"data"`.

Value

A list in Kerchunk V1 format.

Examples

```
## Not run:
refs <- tiff_refs("s3://bucket/file.tif", anon = TRUE)
kc <- refs_to_kerchunk(refs)
jsonlite::write_json(kc, "references.json", auto_unbox = TRUE)

## End(Not run)
```

tiff_refs *Extract tile byte-range references from TIFF/COG files.*

Description

Extract tile byte-range references from TIFF/COG files.

Usage

```
tiff_refs(paths, region, anon, concurrency)
```

Arguments

paths	Character vector of file paths or URLs (s3://, gs://, az://, http://, https://, or local paths).
region	Optional AWS region string (e.g. "us-west-2").
anon	Logical, use anonymous/unsigned requests. Default FALSE.
concurrency	Integer, max concurrent file scans. Default 16.

Value

A data.frame with columns: path, ifd, tile_col, tile_row, offset, length, image_w, image_h, tile_w, tile_h, dtype, compression, bits_per_sample, samples_per_pixel, crs_epsg.

tiff_tile *Fetch and decode a single tile from a TIFF/COG file.*

Description

Fetch and decode a single tile from a TIFF/COG file.

Usage

```
tiff_tile(path, ifd_index, col, row, region, anon)
```

Arguments

path	File path or URL to the TIFF.
ifd_index	IFD index (0-based). Default 0 (full resolution).
col	Tile column (0-based).
row	Tile row (0-based).
region	Optional AWS region string.
anon	Logical, use anonymous requests. Default FALSE.

Value

A named list with components:

- `data`: numeric vector of decoded pixel values
- `dim`: integer vector c(height, width, bands)
- `dtype`: character string (e.g. "<f4", "<u2")

`tiff_tiles` *Fetch and decode multiple tiles as a batch.*

Description

Fetch and decode multiple tiles as a batch.

Usage

```
tiff_tiles(path, ifd_index, cols, rows, region, anon)
```

Arguments

`path`	File path or URL to the TIFF.
`ifd_index`	IFD index (0-based). Default 0 (full resolution).
`cols`	Integer vector of tile columns (0-based).
`rows`	Integer vector of tile rows (0-based).
`region`	Optional AWS region string.
`anon`	Logical, use anonymous requests. Default FALSE.

Value

A list of tile results, each with data, dim, and dtype.

`tile_to_array` *Convert tile data to a matrix or array*

Description

Convert tile data to a matrix or array

Usage

```
tile_to_array(tile)
```

Arguments

`tile`	A tile result from `tiff_tile()`.

Value

A matrix (single band) or 3D array (multi-band).

Index