

Package: blocklist (via r-universe)

October 15, 2026

Title Harvest Byte-References of Compressed Array Data Chunks Native to HDF5-Accessible Files

Version 0.0.1.9002

Description Generate byte-reference stores from chunked array storage, on disk, on s3 storage or at urls. Array sources are logically mosaiced with gdal multidimensional tool 'mdim mosaic' and this is used to extract and organize raw byte references (path, offset, size) written to Parquet with semantic grouping of manually-sharded, positionally-indexed parquet reference store. The structure is a directory of parquet files, manually partitioned by variable (subdirectory) and by flat-chunk-index range ('refs.N.parq', 'record_size' rows each), with the partition keys implied positionally rather than stored as columns (i.e. _not_ hive-encoded). This is compatible with ``kerchunk-parquet Zarr'', and is used as an generic and intermediate convenience before generating 'icechunk' versions.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Remotes mdsumner/rhdf5@H5D-chunk-iter

Imports arrow, jsonlite, mirai, rhdf5, xml2

URL <https://github.com/hypertidy/blocklist>

BugReports <https://github.com/hypertidy/blocklist/issues>

Config/pak/sysreqs cmake make libxml2-dev libssl-dev

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

Date/Publication 2026-07-16 06:08:44 UTC

RemoteUrl <https://github.com/hypertidy/blocklist>

RemoteRef HEAD

RemoteSha 9f5ede94906ba1ba5d8f18a715c3d7cde3309b3f

Contents

mosaic_sources	2
parse_mosaic_vrt	3
scan_source_chunks	3
virtualize_mosaic	4
write_kerchunk_parquet	5
Index	7

mosaic_sources	Build a sources table for virtualize_mosaic()
----------------	---

Description

Pairs each source file’s *access* path (what gdal mdim mosaic is given and what rhdf5 opens to scan) with its *public* URL (what is written into the reference store). Chunk byte offsets are identical in any byte-identical copy of a file, so the two may differ freely: scan a cheap local or NFS path while referencing a durable remote URL.

Usage

```
mosaic_sources(public, access = public)
```

Arguments

public	Character vector of durable URLs to record in the references (e.g. https:// or s3://), one per source file.
access	Character vector, same order as public, of the paths actually opened during the build (handed to gdal mdim mosaic and opened by rhdf5). Defaults to public for the all-remote case.

Value

A data.frame with columns access and public, one row per source.

See Also

```
virtualize\_mosaic\(\)
```

Examples

```
## Not run:
mosaic_sources(
  public = c("https://host/day01.nc", "https://host/day02.nc"),
  access = c("/nfs/day01.nc",          "/nfs/day02.nc"))

## End(Not run)
```

parse_mosaic_vrt	<i>Parse a GDAL multidimensional mosaic VRT</i>
------------------	---

Description

Reads the logical manifest produced by `gdal mdim mosaic`: dimension sizes, and for each data array its C-order dimension names, dtype, shape, GDAL block size, CF scale/offset/nodata/unit, and the ordered list of sources (each with SourceFilename, SourceArray, and DestSlab offset). Coordinate arrays are returned with their composed values (from the VRT's InlineValues or RegularlySpacedValues). No data or chunk bytes are read.

Usage

```
parse_mosaic_vrt(vrt)
```

Arguments

vrt	Path to a .vrt written by <code>gdal mdim mosaic</code> .
-----	---

Value

A list with `dim_size` (named integer vector of dimension extents), `arrays` (named list of data-variable descriptions, each with `dim_names`, `dtype`, `shape`, `chunks`, `scale`, `offset`, `nodata`, `unit`, `sources`), and `coords` (named list with `dim`, `dtype`, `values`).

See Also

[virtualize_mosaic\(\)](#)

scan_source_chunks	<i>Extract chunk byte references for one array in one source file</i>
--------------------	---

Description

Opens `source_array` in `scan_path` read-only and walks its HDF5 chunk index (`h5getAllChunkInfo`) to recover, per chunk, the grid coordinate (C order) and the byte address and length of the stored, compressed data. Coordinates are local to the source; the caller shifts them to global position using the mosaic DestSlab offset. A non-zero filter mask on any chunk aborts with an error - it means the chunk does not share the array-level codec and cannot be virtualised uniformly.

Usage

```
scan_source_chunks(scan_path, source_array, ref_path, A, contiguous)
```

Arguments

scan_path	Path opened to read the chunk index (the source's access).
source_array	Array name within the file, e.g. <code>"/anom"</code> .
ref_path	Durable URL written into the path column (the public URL).
A	Array description from <code>parse_mosaic_vrt()</code> with at least shape and chunks (C order); chunks must be the authoritative HDF5 storage chunk.
contiguous	Logical; if TRUE the dataset is unchunked and a single whole-array reference is emitted instead of iterating a chunk index.

Value

A `data.frame`, one row per chunk: ndim integer coordinate columns (`c1..cN`), offset (byte address), size (byte length), path.

<code>virtualize_mosaic</code>	<i>Virtualise a GDAL mosaic into a kerchunk-Parquet reference store</i>
--------------------------------	---

Description

End-to-end driver. Parses a `gdal mdim mosaic` VRT for the logical layout, probes each array's codec and storage chunk shape once via `rhdf5`, scans every source file for chunk byte references, composes them across files using the VRT `DestSlab` offsets, inlines coordinate variables (values from the VRT, CF attributes read from a source), and writes the store. The output is both a portable reference store and the input to a `VirtualiZarr -> Icechunk` step.

Usage

```
virtualize_mosaic(vrt, root, sources, record_size = 100000L)
```

Arguments

vrt	Path to a VRT written by <code>gdal mdim mosaic</code> over <code>sources\$access</code> .
root	Output directory for the store.
sources	A <code>data.frame</code> with columns <code>access</code> and <code>public</code> , typically from <code>mosaic_sources()</code> . Joined to the VRT by <code>access</code> (the <code>SourceFilename</code>), with a <code>basename</code> fallback.
record_size	Integer chunk references per parquet shard (default <code>1e5</code>).

Value

The store root, invisibly.

See Also

`mosaic_sources()`, `parse_mosaic_vrt()`, `write_kerchunk_parquet()`

Examples

```
## Not run:
src <- mosaic_sources(public = urls, access = nfs_paths)
system(sprintf("gdal mdim mosaic %s mosaic.vrt", paste(src$access, collapse = " ")))
virtualize_mosaic("mosaic.vrt", "oisst.zarr", sources = src)

## End(Not run)
```

write_kerchunk_parquet

Writes the fsspec LazyReferenceMapper layout (Zarr v2): a root .zmetadata holding consolidated array/group metadata plus record_size, and a <var>/refs.<N>.parq shard per array. Each row is either a chunk reference (path/offset/size) or inline data (raw); rows are placed at the C-order flat chunk index and padded to record_size. The result is readable by VirtualiZarr (KerchunkParquetParser), the GDAL Zarr driver, and any fsspec reference filesystem.

Description

Writes the fsspec LazyReferenceMapper layout (Zarr v2): a root .zmetadata holding consolidated array/group metadata plus record_size, and a <var>/refs.<N>.parq shard per array. Each row is either a chunk reference (path/offset/size) or inline data (raw); rows are placed at the C-order flat chunk index and padded to record_size. The result is readable by VirtualiZarr (KerchunkParquetParser), the GDAL Zarr driver, and any fsspec reference filesystem.

Usage

```
write_kerchunk_parquet(
  root,
  vars_meta,
  ref_tables,
  inline = list(),
  record_size = 100000L,
  root_attrs = list()
)
```

Arguments

root	Output directory for the store (created, overwriting any existing).
vars_meta	Named list of per-variable metadata; each entry has shape, chunks, dtype, compressor, filters, fill_value, dim_names, attrs.
ref_tables	Named list (matching vars_meta) of chunk-reference data.frames from scan_source_chunks() ; omit for variables given via inline.

<code>inline</code>	Named list of variables stored inline; each entry has <code>raw</code> , the bytes of a single whole-array chunk.
<code>record_size</code>	Integer chunk references per parquet shard (default 1e5).
<code>root_attrs</code>	Named list of group-level attributes for the root <code>.zattrs</code> .

Value

The store root, invisibly.

See Also

[`virtualize\_mosaic\(\)`](#)

Index

mosaic_sources, [2](#)
mosaic_sources(), [4](#)

parse_mosaic_vrt, [3](#)
parse_mosaic_vrt(), [4](#)

scan_source_chunks, [3](#)
scan_source_chunks(), [5](#)

virtualize_mosaic, [4](#)
virtualize_mosaic(), [2](#), [3](#), [6](#)

write_kerchunk_parquet, [5](#)
write_kerchunk_parquet(), [4](#)