

Package: Rgguf (via r-universe)

July 21, 2026

Title Official GGUF Parsing and Typed Storage Views

Version 0.1.0

Description Reads 'GGUF' model files, the file format used by the 'llama.cpp' project (<<https://github.com/ggerganov/llama.cpp>>) to store large language model tensors and metadata. Tensors can be decoded into file-backed 'Rfmalloc' arrays, copied at their native codec density, or borrowed as zero-copy read-only spans over the original model mapping. Parsing, writing, and quantized decoding use the official 'GGML' GGUF implementation exposed by the sibling 'Rggml' package. Metadata key-value pairs and the tensor directory are exposed as ordinary R lists and data frames. A minimal writer is included to create 'GGUF' files containing 32-bit floating point tensors, primarily to support round-trip testing.

License GPL (>= 2)

Encoding UTF-8

Imports Rfmalloc, Rggml

LinkingTo Rfmalloc, Rggml

Suggests tinytest

Remotes Rfmalloc=souunkou-bioinfo/Rfmalloc/packages/Rfmalloc,
Rggml=souunkou-bioinfo/Rfmalloc/packages/Rggml

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

SystemRequirements GNU make

URL <https://github.com/souunkou-bioinfo/Rfmalloc>,
<https://souunkou-bioinfo.github.io/Rfmalloc/Rgguf/>

Depends R (>= 4.4.0)

BugReports <https://github.com/souunkou-bioinfo/Rfmalloc/issues>

Config/pak/sysreqs make

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

Date/Publication 2026-07-18 21:34:09 UTC
RemoteUrl https://github.com/RGenomicsETL/Rfmalloc
RemoteRef HEAD
RemoteSha 6e05ad03613e89cab2f8bf8eb3a49d553d5a3bed
RemoteSubdir packages/Rgguf

Contents

gguf_import	2
gguf_metadata	3
gguf_open	4
gguf_tensor	5
gguf_tensors	6
gguf_write_tensors	7
Index	8

gguf_import	<i>Import GGUF Tensors</i>
-------------	----------------------------

Description

Reads some or all tensors from a GGUF file into a named list of **Rfmalloc**-backed matrices/arrays, dequantizing as needed. This is a thin convenience wrapper around repeated `gguf_tensor()` calls that shares a single open file handle across all of them.

Usage

```
gguf_import(  
  path_or_ctx,  
  tensors = NULL,  
  runtime = NULL,  
  as = c("numeric", "native", "view")  
)
```

Arguments

path_or_ctx	Either a path to a GGUF file, or a "gguf_ctx" object returned by <code>gguf_open()</code> .
tensors	Optional character vector of tensor names to import. If NULL (the default), every tensor in the file is imported.
runtime	Optional Rfmalloc runtime handle passed through to every <code>gguf_tensor()</code> call, so all imported tensors share the same backing file. If NULL, Rfmalloc's own default-runtime resolution is used for each tensor.
as	Passed through to <code>gguf_tensor()</code> : "numeric" dequantizes to fmalloc double matrices/arrays, "native" copies the encoded bytes into fmalloc storage, and "view" borrows the original read-only GGUF spans.

Value

A named list of tensors, in the order requested (or file order, if `tensors` is `NULL`). Numeric and native imports own `fmalloc` allocations; views are read-only spans over the original GGUF mapping.

Examples

```
tmp <- tempfile(fileext = ".gguf")
gguf_write_tensors(tmp, list(
  w1 = matrix(as.double(1:12), nrow = 4, ncol = 3),
  w2 = matrix(as.double(1:6), nrow = 3, ncol = 2)
))
rt <- Rfmalloc::open_fmmalloc(tempfile(fileext = ".bin"))
mats <- gguf_import(tmp, runtime = rt)
prod <- mats$w1 %*% mats$w2
Rfmalloc::is_fmmalloc_vector(prod)
unlink(tmp)
```

gguf_metadata

Read GGUF Metadata

Description

Reads every metadata key-value pair from a GGUF file into a named R list.

Usage

```
gguf_metadata(x)
```

Arguments

`x` Either a path to a GGUF file, or a "gguf_ctx" object returned by [gguf_open\(\)](#).

Details

Scalar values are converted to the closest native R type (integer for 8/16/32-bit signed and unsigned-but-int-safe integers, numeric for 32-bit unsigned and 64-bit integers/floats, logical for booleans, and character for strings). Arrays of a supported scalar type become an R vector of that type. A metadata value of a type this package does not represent (there are none in the current GGUF spec, but the check is defensive) is returned as `NULL` while keeping its name, rather than raising an error.

Value

A named list of metadata values, in file order.

Examples

```
tmp <- tempfile(fileext = ".gguf")
gguf_write_tensors(tmp, list(w = matrix(1:6 + 0.5, nrow = 2)),
  metadata = list(name = "example", version = 1))
gguf_metadata(tmp)
unlink(tmp)
```

gguf_open

Open a GGUF File

Description

Opens a 'GGUF' model file and returns a lightweight context handle used by [gguf_metadata\(\)](#), [gguf_tensors\(\)](#), [gguf_tensor\(\)](#), and [gguf_import\(\)](#). Metadata and the tensor directory are parsed by Rggml's official GGUF implementation. Tensor bytes are mapped read-only and unmapped automatically when the returned object is garbage collected, or earlier if you call [gguf_import\(\)/friends](#) with a plain file path (which open and close their own short-lived context internally).

Usage

```
gguf_open(path)
```

Arguments

path Character string giving the path to a GGUF file.

Value

An object of class "gguf_ctx": an external pointer to the underlying parser context, with a finalizer that closes it.

Examples

```
tmp <- tempfile(fileext = ".gguf")
gguf_write_tensors(tmp, list(w = matrix(1:6 + 0.5, nrow = 2)))
ctx <- gguf_open(tmp)
gguf_tensors(ctx)
unlink(tmp)
```

gguf_tensor

Read and Dequantize a GGUF Tensor

Description

Reads a single named tensor from a GGUF file, dequantizing it if needed, into a fresh **Rfmalloc**-backed ALTREP matrix or array of doubles.

Usage

```
gguf_tensor(x, name, runtime = NULL, as = c("numeric", "native", "view"))
```

Arguments

x	Either a path to a GGUF file, or a "gguf_ctx" object returned by gguf_open() .
name	Tensor name, as it appears in gguf_tensors() 's name column.
runtime	Optional Rfmalloc runtime handle (see Rfmalloc::open_fmmalloc()). If NULL, Rfmalloc's own default-runtime resolution is used.
as	"numeric" (default) dequantizes the whole tensor into an fmalloc double matrix/array. "native" copies the tensor's raw, still-encoded payload into fmalloc storage and returns an Rfmalloc::create_fmmalloc_tensor() typed tensor: it keeps the GGUF storage density (e.g. 4.5 bits/weight for q4_k) and is decoded in bounded panels only when used in matrix products. "view" returns the same typed tensor without copying: it borrows the exact read-only span in the original GGUF mapping and keeps that mapping alive. Native mode requires two dimensions; view mode accepts any rank. Both require a registered Rfmalloc codec.

Details

GGUF tensor dimensions are stored with `dim[0]` as the fastest-varying (contiguous) dimension. R arrays are column-major, i.e. the first dimension varies fastest, so `dim[0]` maps directly onto R's first dimension with no transposition needed: `dim(result) == c(dim[0], dim[1], ...)`. See `inst/tinytest/test_gguf_roundtrip.R` for a test that verifies this mapping by writing known matrices/arrays and reading them back.

Value

For `as = "numeric"`, an Rfmalloc-backed ALTREP matrix (if the tensor has 2 dimensions) or array (otherwise) of doubles, with `dim()` equal to the tensor's GGUF dimensions in `c(dim[0], dim[1], ...)` order. For `as = "native"` or `"view"`, an `fmalloc_tensor` with the same dims.

Examples

```
tmp <- tempfile(fileext = ".gguf")
gguf_write_tensors(tmp, list(w = matrix(1:6 + 0.5, nrow = 2)))
rt <- Rfmalloc::open_fmalloc(tempfile(fileext = ".bin"))
w <- gguf_tensor(tmp, "w", runtime = rt)
w
Rfmalloc::is_fmalloc_vector(w)
unlink(tmp)
```

gguf_tensors

List GGUF Tensors

Description

Lists the tensor directory of a GGUF file as a data frame, without reading or dequantizing any tensor payload.

Usage

```
gguf_tensors(x)
```

Arguments

x Either a path to a GGUF file, or a "gguf_ctx" object returned by [gguf_open\(\)](#).

Value

A data frame with one row per tensor and columns:

name Tensor name.

type GGUF tensor type name, e.g. "f32", "q4_k".

n_dims Number of dimensions.

dims A list column; `dims[[i]]` is an integer vector of tensor `i`'s dimensions, GGUF-order (`dim[0]` first, the fastest-varying/ contiguous dimension, which [gguf_tensor\(\)](#) maps to R's first, column-major dimension).

n_elements Total number of scalar elements.

nbytes Size of the tensor's raw (possibly quantized) on-disk representation, in bytes.

offset Byte offset of the tensor data from the start of the file.

Examples

```
tmp <- tempfile(fileext = ".gguf")
gguf_write_tensors(tmp, list(w = matrix(1:6 + 0.5, nrow = 2), b = 1:3 + 0.5))
gguf_tensors(tmp)
unlink(tmp)
```

gguf_write_tensors *Write a Minimal GGUF File*

Description

Writes a named list of numeric vectors/matrices/arrays to a GGUF file as 32-bit floating point (f32) tensors, with optional simple metadata. This is a minimal writer, primarily meant to build small GGUF fixtures for this package's own round-trip tests without shipping binary test fixtures, but it is exported since it is also useful on its own for producing test fixtures for other GGUF consumers.

Usage

```
gguf_write_tensors(path, tensors, metadata = list())
```

Arguments

path	Character string giving the output file path.
tensors	A non-empty named list of numeric vectors, matrices, or arrays. Names become tensor names and must be unique and non-empty.
metadata	A named list of metadata key-value pairs to write. Values may be non-missing character or numeric vectors. Vectors of length one are written as scalar strings or 64-bit floats; other lengths become GGUF arrays of the corresponding type. Defaults to <code>list()</code> (no metadata).

Details

Tensor dimensions are taken from each object's `dim()` (or its `length()` for a plain vector, written as a 1-dimensional tensor) and stored in GGUF `dim[0]`-fastest-varying order directly from R's column-major storage, so `gguf_tensor()/gguf_import()` read back exactly the matrix/array you wrote (see `inst/tinytest/test_gguf_roundtrip.R`).

Existing files at path are silently overwritten.

Value

path, invisibly.

Examples

```
tmp <- tempfile(fileext = ".gguf")
gguf_write_tensors(tmp,
  tensors = list(weight = matrix(1:6 + 0.5, nrow = 2)),
  metadata = list(name = "example-model", version = 1)
)
gguf_tensors(tmp)
unlink(tmp)
```

Index

gguf_import, [2](#)
gguf_import(), [4](#)
gguf_metadata, [3](#)
gguf_metadata(), [4](#)
gguf_open, [4](#)
gguf_open(), [2](#), [3](#), [5](#), [6](#)
gguf_tensor, [5](#)
gguf_tensor(), [2](#), [4](#), [6](#)
gguf_tensors, [6](#)
gguf_tensors(), [4](#), [5](#)
gguf_write_tensors, [7](#)

Rfmalloc::create_fmalloc_tensor(), [5](#)
Rfmalloc::open_fmalloc(), [5](#)