Package 'grumpy'

November 16, 2026

Title Read 'NumPy' '.npy' and '.npz' Files

Version 0.2.0

Description Lightweight way to read 'NumPy' '.npy' and '.npz' files in R. All data types supported by 'NumPy', with all sizes (converted internally to R native size), both C and 'Fortran' order, and any shape, up to an arbitrary number of dimensions, are supported.

License MIT + file LICENSE

Encoding UTF-8

Suggests microbenchmark, quarto, Rarr, reticulate, rmarkdown, testthat (>= 3.0.0), withr

VignetteBuilder quarto

Config/testthat/edition 3

URL https://hugogruson.fr/grumpy/, https://github.com/Bisaloo/grumpy

BugReports https://github.com/Bisaloo/grumpy/issues

Config/Needs/website Huber-group-EMBL/Rarr

Config/roxygen2/version 8.1.0

Depends R (>= 4.2.0)

NeedsCompilation yes

Author Hugo Gruson [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-4094-1476>), Mike Smith [aut, cph] (Original author of portions of the C code migrated from the Rarr package), German Network for Bioinformatics Infrastructure - de.NBI [fnd]

Maintainer Hugo Gruson <hugo.gruson+R@normalesup.org>

Repository CRAN

Date/Publication 2026-11-16 15:00:02 UTC

Contents

convert_bytes_to_array 2
parse_npy_datatype 3
read_npy 4
read_npz 5

Index **6**

`convert_bytes_to_array` *Convert raw bytes to an R array based on the specified data type information*

Description

This is a replacement for `readBin()` that can handle the various data types and endianness specified in the .npy file header.

Usage

```
convert_bytes_to_array(bytes, what, shape, size, endian)
```

Arguments

`bytes`	A raw vector containing the bytes to convert
`what`	A character specifying the base type to convert to (e.g., `"float"`, `"int"`, `"string"`, etc.)
`shape`	A numeric vector with desired shape of the output array
`size`	A numeric value with the number of bytes per element for the specified type
`endian`	The endianness of the data (`"little"`, `"big"`, or `NA` for single-byte types)

Value

An R array containing the converted data, with the specified shape and data type.

Examples

```
x <- matrix(c(3L, 6L, 2L, 1L, 12L, 0L), nrow = 2, ncol = 3)
x

y <- writeBin(c(x), raw()) |>
  convert_bytes_to_array(
     "int",
     shape = c(2L, 3L),
     size = 4L,
     endian = "little"
  )
```

```
y
dim(y)
is.array(y)
storage.mode(y)
```

`parse_npy_datatype` *Parse a NumPy Array-protocol type string*

Description

Parse a NumPy Array-protocol type string

Usage

```
parse_npy_datatype(descr)
```

Arguments

`descr`	A NumPy dtype description string, or a list of such strings for structured dtypes

Details

If a `list` is passed to `descr`, each element can be of length 1, or of length 2 in which case the first element corresponds to the name of the field and the second to its dtype.

Value

A list containing the parsed data type information, including the base type, the number of bytes, and the endianness

Examples

```
parse_npy_datatype(">i8")
parse_npy_datatype("|b1")
# A structured datatype where each element has 3 components, all integers,
# named "r", "g" and "b".
parse_npy_datatype(list(c("r", "<i8"), c("g", "<i8"), c("b", "<i8")))
```

read_npy *Read a* .npy *file*

Description

Read a .npy file

Usage

```
read_npy(file, lazy = FALSE, ...)
```

Arguments

file	Path to the .npy file
lazy	If TRUE, and file is a path (not a connection), the data payload is memory-mapped rather than read into memory upfront via readBin(). This limits the number of copies in memory when the npy file contains types native to R. Requires a POSIX platform; ignored (with a warning) otherwise. Defaults to FALSE.
...	Ignored. Reserved for future use.

Value

An array containing the data from the .npy file

Examples

```
# Array of integers. NumPy "<i4" dtype
read_npy(
  system.file("extdata", "test_int32.npy", package = "grumpy")
)

# Array of logicals. NumPy "|b1" dtype
read_npy(
  system.file("extdata", "test_bool.npy", package = "grumpy")
)

# Array of numerics. NumPy "<f8" dtype
read_npy(
  system.file("extdata", "test_float64.npy", package = "grumpy")
)

# Array of strings. NumPy "<U" dtype
read_npy(
  system.file("extdata", "test_str_unicode.npy", package = "grumpy")
)

# "Array" of lists. Numpy structured dtype
```

```
read_npy(
  system.file("extdata", "test_structured.npy", package = "grumpy")
)
```

---

read_npz *Read a* .npz *file*

Description

Read a .npz file

Usage

```
read_npz(file, arrays = NULL)
```

Arguments

file	Path to the .npz file
arrays	Optional character vector of array names or numeric vector of indices to read from the .npz file. If NULL (the default), all arrays are read.

Value

A named list of arrays containing the data from the .npz file

Examples

```
read_npz(
  system.file("extdata", "test.npz", package = "grumpy")
)

read_npz(
  system.file("extdata", "test.npz", package = "grumpy"),
  arrays = "x"
)
```

Index

convert_bytes_to_array, 2

parse_npy_datatype, 3

read_npy, 4
read_npz, 5
readBin(), *2*, *4*