Package 'grumpy'

May 19, 2026

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

Version 0.1.1

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 knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

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

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

Imports jsonlite

Config/roxygen2/version 8.0.0

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 2026-05-19 09:30:08 UTC

Contents

`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 strings*

Description

Parse a NumPy Array-protocol type strings

Usage

```
parse_npy_datatype(descr)
```

Arguments

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

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")
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)
```

Arguments

file	Path to the .npy file

Value

An array containing the data from the .npy file

Examples

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

---

read_npz *Read a .npz file*

Description

Read a .npz file

Usage

```
read_npz(file)
```

Arguments

file	Path to the .npz file

Value

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

Examples

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

Index