Package 'jlview'

March 23, 2026

Title Zero-Copy Julia to R Array Bridge via ALTREP

Version 0.1.0

Description Provides zero-copy R views of Julia-owned arrays by implementing ALTREP (Alternative Representations) classes that return pointers directly into Julia's memory. The package integrates with 'JuliaCall' and uses C-level finalizers for safe cross-runtime garbage collection.

License MIT + file LICENSE

URL https://github.com/tanaylab/jlview

BugReports https://github.com/tanaylab/jlview/issues

Depends R (>= 4.0.0)

SystemRequirements Julia (>= 1.6, https://julialang.org/downloads/)

Imports JuliaCall, Matrix, methods

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

NeedsCompilation yes

Author Aviezer Lifshitz [aut, cre] (ORCID:
<https://orcid.org/0000-0002-8458-9507>),
Weizmann Institute of Science [cph]

Maintainer Aviezer Lifshitz <aviezer.lifshitz@weizmann.ac.il>

Repository CRAN

Date/Publication 2026-03-23 17:40:08 UTC

Contents

is_jlview 2
jlview 2
jlview_gc_pressure 3
jlview_info 4
jlview_named_matrix 4
jlview_named_vector 5
jlview_release 6
jlview_set_gc_threshold 6
jlview_sparse 7
with_jlview 7

Index **9**

`is_jlview` *Check if an object is a jlview ALTREP vector*

Description

Check if an object is a jlview ALTREP vector

Usage

```
is_jlview(x)
```

Arguments

`x`	An R object

Value

`TRUE` if `x` is a jlview ALTREP vector, `FALSE` otherwise

`jlview` *Create a zero-copy R view of a Julia array*

Description

Creates an ALTREP vector (or matrix, if 2D+) backed by Julia memory. The resulting R object shares the same memory as the Julia array, avoiding data copying. Modifications to the R object trigger copy-on-write (unless `writeable = TRUE`).

Usage

```
jlview(julia_array, writeable = FALSE, names = NULL, dimnames = NULL)
```

Arguments

julia_array	A JuliaObject referencing a Julia array
writeable	If TRUE, allow R to write directly to Julia's memory. Use with caution — this enables shared mutation. Default FALSE.
names	Optional character vector of names to attach to the result. Attached atomically during construction to avoid ALTREP materialization.
dimnames	Optional list of dimnames to attach to the result. Attached atomically during construction to avoid ALTREP materialization.

Value

An ALTREP vector backed by Julia memory, or a standard R vector if the Julia type is not supported for zero-copy.

Examples

```
## Not run:
JuliaCall::julia_setup()
# Create a Julia array and view it in R without copying
JuliaCall::julia_command("x = randn(1000)")
x <- jlview(JuliaCall::julia_eval("x"))
sum(x) # operates directly on Julia memory

## End(Not run)
```

---

jlview_gc_pressure *Get current GC pressure information*

Description

Returns the current amount of Julia memory pinned by jlview objects and the threshold at which forced GC is triggered.

Usage

```
jlview_gc_pressure()
```

Value

A list with pinned_bytes and threshold

jlview_info *Get information about a jlview object*

Description

Returns metadata about a jlview ALTREP vector including the Julia element type, length, writeability, and release status.

Usage

```
jlview_info(x)
```

Arguments

x	A jlview ALTREP vector

Value

A named list with components:

type Julia element type (e.g., "Float64")

length Number of elements

writeable Whether the view allows direct writes

released Whether the view has been released

materialized Whether COW materialization has occurred

jlview_named_matrix *Create a zero-copy R view of a named Julia matrix*

Description

Creates a zero-copy ALTREP view of a Julia NamedArray matrix, preserving row and column names as dimnames.

Usage

```
jlview_named_matrix(julia_named_matrix)
```

Arguments

julia_named_matrix	A JuliaObject referencing a Julia NamedArray matrix

Value

An ALTREP matrix with dimnames set from the Julia NamedArray

Examples

```
## Not run:
JuliaCall::julia_setup()
JuliaCall::julia_command("using NamedArrays")
m <- JuliaCall::julia_eval('NamedArray(randn(3,2), (["a","b","c"], ["x","y"]))')
x <- jlview_named_matrix(m)
rownames(x) # returns c("a", "b", "c")
colnames(x) # returns c("x", "y")

## End(Not run)
```

`jlview_named_vector` *Create a zero-copy R view of a named Julia vector*

Description

Creates a zero-copy ALTREP view of a Julia NamedArray vector, preserving the axis names.

Usage

```
jlview_named_vector(julia_named_array)
```

Arguments

`julia_named_array`
: A JuliaObject referencing a Julia NamedArray vector

Value

An ALTREP vector with names set from the Julia NamedArray

Examples

```
## Not run:
JuliaCall::julia_setup()
JuliaCall::julia_command("using NamedArrays")
v <- JuliaCall::julia_eval('NamedArray([1.0, 2.0, 3.0], (["a", "b", "c"],))')
x <- jlview_named_vector(v)
names(x) # returns c("a", "b", "c")

## End(Not run)
```

jlview_release *Explicitly release a jlview object*

Description

Unpins the Julia array immediately, freeing memory without waiting for R's garbage collector. After release, accessing the data will error.

Usage

```
jlview_release(x)
```

Arguments

x	A jlview ALTREP vector

Value

Invisible NULL

jlview_set_gc_threshold *Set the GC pressure threshold*

Description

When total pinned bytes exceeds this threshold, jlview forces an R garbage collection to reclaim stale ALTREP objects. Default is 10GB.

Usage

```
jlview_set_gc_threshold(bytes)
```

Arguments

bytes	Threshold in bytes (numeric)

Value

Invisible NULL

jlview_sparse *Create a zero-copy R sparse matrix view of a Julia SparseMatrixCSC*

Description

Creates a dgCMatrix-class backed by a zero-copy ALTREP vector for the nonzero values (x slot). The row indices (i slot) and column pointers (p slot) are copied and shifted from 1-based (Julia) to 0-based (R) indexing in Julia, then returned as plain R integer vectors.

Usage

```
jlview_sparse(julia_sparse_matrix, lazy_indices = FALSE)
```

Arguments

julia_sparse_matrix	A JuliaObject referencing a Julia SparseMatrixCSC. The value type must be supported for zero-copy (Float64, Float32, Int32, Int64, Int16, UInt8).
lazy_indices	Ignored. Retained for API compatibility only. Previously controlled lazy vs eager materialization of ALTREP index vectors, which have been removed in favor of simple copy+shift in Julia.

Value

A dgCMatrix-class sparse matrix.

Examples

```
## Not run:
JuliaCall::julia_setup()
JuliaCall::julia_command("using SparseArrays")
m <- JuliaCall::julia_eval("sprand(Float64, 100, 50, 0.1)")
s <- jlview_sparse(m)
class(s) # "dgCMatrix"

## End(Not run)
```

with_jlview *Use a jlview object within a scope, releasing it on exit*

Description

Creates a jlview object and ensures it is released when the scope exits, even if an error occurs. This prevents memory leaks from forgotten releases.

Usage

```
with_jlview(
  julia_array,
  expr,
  writeable = FALSE,
  names = NULL,
  dimnames = NULL
)
```

Arguments

julia_array	A JuliaObject referencing a Julia array
expr	An expression to evaluate with the jlview object bound to .x
writeable	Passed to jlview
names	Passed to jlview
dimnames	Passed to jlview

Value

The result of evaluating expr

Examples

```
## Not run:
JuliaCall::julia_setup()
JuliaCall::julia_command("big = randn(100000)")
result <- with_jlview(JuliaCall::julia_eval("big"), {
    c(mean(.x), sd(.x))
})
# .x is automatically released here

## End(Not run)
```

Index

is_jlview, 2

jlview, 2, *8*
jlview_gc_pressure, 3
jlview_info, 4
jlview_named_matrix, 4
jlview_named_vector, 5
jlview_release, 6
jlview_set_gc_threshold, 6
jlview_sparse, 7

with_jlview, 7