Package 'armadillo4r'

June 12, 2026

Type Package

Title An 'Armadillo' Interface

Description Provides function declarations and inline function definitions that facilitate communication between R and the 'Armadillo' 'C++' library for linear algebra and scientific computing. This implementation is derived from Vargas Sepulveda and Schneider Malamud (2024) <doi:10.1016/j.softx.2025.102087>.

Version 1.0.0

Imports cpp4r

Suggests Matrix, roxygen2, tinytest

License Apache License (>= 2)

URL https://pacha.dev/armadillo4r/,
https://github.com/pachadotdev/armadillo4r

BugReports https://github.com/pachadotdev/armadillo4r/issues

Depends R(>= 4.0.0)

Encoding UTF-8

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Mauricio Vargas Sepulveda [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1017-7574>),
Jonathan Schneider Malamud [ctb],
Conrad Sanderson [aut] (Armadillo library (C++))

Maintainer Mauricio Vargas Sepulveda <m.vargas.sepulveda@gmail.com>

Repository CRAN

Date/Publication 2026-06-12 13:10:07 UTC

Contents

`armadillo_version` *Get Armadillo version*

Description

Provides the Armadillo C++ library version name and number included in the package.

Usage

```
armadillo_version()
```

Value

A string with the Armadillo version name and number

Examples

```
armadillo_version()
```

`pkg_template` *Start a new project with the armadillo4r package template*

Description

Copies a package template into a new directory. The template includes a DESCRIPTION file, a minimal R/ directory and placeholders with instructions. You can then edit these files to customize your new package.

Usage

```
pkg_template(path = NULL, pkgname = NULL)
```

Arguments

`path`	Path to the new project
`pkgname`	Name of the new package

Value

The file path to the copied template (invisibly).

Examples

```
# create a new directory
dir <- tempdir()
dir.create(dir)

# copy the package template into the directory
pkg_template(dir, "mynewpkg")
```

---

unvendor *Unvendor the armadillo4r headers*

Description

This function removes the vendored armadillo4r headers from your package by automatically finding the vendored headers.

Usage

```
unvendor(path = NULL)
```

Arguments

path	The directory with the vendored headers. It is recommended to use "./src/vendor". The default is NULL.

Value

The path to the unvendored code (invisibly).

Examples

```
# create a new directory
dir <- paste0(tempdir(), "/", gsub("\\s+|[[:punct:]]", "", Sys.time()))
dir.create(dir, recursive = TRUE)

# vendor the armadillo4r headers into the directory
vendor(dir)

# unvendor the armadillo4r headers from the directory
unvendor(dir)

# cleanup
unlink(dir, recursive = TRUE)
```

vendor *Vendor the cpp4r and armadillo4r headers*

Description

Vendoring is the act of making your own copy of the 3rd party packages your project is using. It is often used in the go language community.

This function vendors armadillo4r into your package by copying the armadillo4r headers into the `inst/include` folder of your package and adding 'armadillo4r version: XYZ' to the top of the files, where XYZ is the version of armadillo4r currently installed on your machine.

Note: vendoring places the responsibility of updating the code on **you**. Bugfixes and new features in armadillo4r will not be available for your code until you run `cpp_vendor()` again.

Usage

```
vendor(path = NULL)
```

Arguments

`path`	The directory with the vendored headers. It is recommended to use `"./src/vendor"`. The default is `NULL`.

Value

The path to the vendored code (invisibly).

Examples

```
# create a new directory
dir <- paste0(tempdir(), "/", gsub("\\s+|[[:punct:]]", "", Sys.time()))
dir.create(dir, recursive = TRUE)

# vendor the armadillo4r headers into the directory
vendor(dir)

# just a preview as Armadillo has over 100 files
head(list.files(dir, recursive = TRUE))

# cleanup
unlink(dir, recursive = TRUE)
```

Index