Package 'mmapcharr'

May 9, 2026

Title Memory-Map Character Files

Version 0.3.1

Date 2025-11-27

Description Uses memory-mapping to enable the random access of elements of a text file of characters separated by characters as if it were a simple R(cpp) matrix.

Encoding UTF-8

License GPL-3

ByteCompile TRUE

Depends R (>= 3.3.0)

Imports methods, Rcpp

LinkingTo Rcpp, rmio

Suggests testthat

RoxygenNote 7.3.3

URL https://github.com/privefl/mmapcharr

BugReports https://github.com/privefl/mmapcharr/issues

Collate 'RcppExports.R' 'extract.R' 'file-dim.R' 'mmapchar.R' 'mmapcharr-package.r' 'utils.R'

NeedsCompilation yes

Author Florian Privé [aut, cre]

Maintainer Florian Privé <florian.prive.21@gmail.com>

Repository CRAN

Date/Publication 2025-11-28 06:20:57 UTC

Contents

`dim_file` *File dimensions*

Description

Number of lines and columns of file (and extra 'return' characters).

Usage

```
dim_file(file)
```

Arguments

`file`	Path to file.

Value

The number of lines and columns of file (and extra 'return' characters).

Examples

```
tmpfile <- tempfile()
write(0:9, tmpfile, ncolumns = 2)
dim_file(tmpfile)
```

`Extract` *Create an Implementation of [For Custom Matrix-Like Types*

Description

`extract` is a function that converts different index types such as negative integer vectors or logical vectors passed to the `[` function as `i` (e.g. `X[i]`) or `i` and `j` (e.g. `X[i, j]`) into positive integer vectors. The converted indices are provided as the `i` parameter of `extract_vector` or `i` and `j` parameters of `extract_matrix` to facilitate implementing the extraction mechanism for custom matrix-like types.

Usage

```
Extract(extract_vector, extract_matrix)
```

Arguments

`extract_vector`	A function in the form of `function(x, i)` that takes a subset of `x` based on a single vector of indices `i` and returns a vector.
`extract_matrix`	A function in the form of `function(x, i, j)` that takes a subset of `x` based on two vectors of indices `i` and `j` and returns a matrix.

Details

The custom type must implement methods for `dim` for this function to work. Implementing methods for `nrow` and `ncol` is not necessary as the default method of those generics calls `dim` internally.

This idea initially comes from package crochet.

Value

A function in the form of `function(x, i, j, ..., drop = TRUE)` that is meant to be used as a method for `[` for a custom type.

`mmapchar-class` *Class mmapchar*

Description

A reference class for storing and accessing matrix-like data stored on disk in files containing only characters (digits) separated by a character.

Usage

```
mmapchar(file, code)
```

Arguments

`file`	Path of the file.
`code`	Integer vector of size 256 to access integers instead of `rawToChar(as.raw(0:255), multiple = TRUE)`. See `mmapcharr:::CODE_012` and `mmapcharr:::CODE_DIGITS`.

Examples

```
test_file <- system.file("testdata/test-windows.txt", package = "mmapcharr")
test <- mmapchar(test_file, code = mmapcharr:::CODE_012)
test[, 1:3]
test[]
readLines(test_file)
```

mmapchar-methods *Methods for the mmapchar class*

Description

Methods for the mmapchar class

Accessor methods for class `mmapchar`. You can use positive and negative indices, logical indices (that are recycled) and also a matrix of indices (but only positive ones).

Dimension and type methods for class `mmapchar`.

Usage

```
## S4 method for signature 'mmapchar'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'mmapchar'
dim(x)

## S4 method for signature 'mmapchar'
length(x)
```

Arguments

`x`	A mmapchar object.
`i`	A vector of indices (or nothing). You can use positive and negative indices, logical indices (that are recycled) and also a matrix of indices (but only positive ones).
`j`	A vector of indices (or nothing). You can use positive and negative indices, logical indices (that are recycled).
`...`	Not used. Just to make nargs works.
`drop`	Whether to delete the dimensions of a matrix which have one dimension equals to 1.

nelem *Size of line*

Description

Number of elements of each line of a file.

Usage

```
nelem(file)
```

Arguments

file	Path to file.

Value

The number of elements of each line of a file.

Examples

```
tmpfile <- tempfile()
write(1:10, tmpfile, ncolumns = 2)
nline(tmpfile)
```

---

nline *Number of lines*

Description

Number of lines of a file.

Usage

```
nline(file)
```

Arguments

file	Path to file.

Value

The number of lines of the file.

Examples

```
tmpfile <- tempfile()
write(1:5, tmpfile, ncolumns = 1)
nline(tmpfile)
```

Index