Package 'fastMatMR'

March 2, 2026

Title High-Performance Matrix Market File Operations

Version 1.2.7

Description An interface to the 'fast_matrix_market' 'C++' library, this package offers efficient read and write operations for Matrix Market files in R. It supports both sparse and dense matrix formats. Peer-reviewed at 'rOpenSci' (<https://github.com/ropensci/software-review/issues/606>).

Author Rohit Goswami [aut, cre] (ORCID: <https://orcid.org/0000-0002-2393-8056>), Ildiko Czeller [rev] (ORCID: <https://orcid.org/0000-0002-9418-4589>), Adam Lugowski [ctb] (ORCID: <https://orcid.org/0009-0004-0922-4067>)

Maintainer Rohit Goswami <rgoswami@ieee.org>

Depends R (>= 3.1.0)

License MIT + file LICENSE

SystemRequirements C++17

Encoding UTF-8

RoxygenNote 7.2.3

LinkingTo cpp11 (>= 0.5.0)

Suggests ggplot2, knitr, Matrix, methods, microbenchmark, rmarkdown, spam, SparseM, testthat (>= 3.0.0)

URL https://github.com/ropensci/fastMatMR

BugReports https://github.com/ropensci/fastMatMR/issues

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-03-02 09:10:20 UTC

Contents

`fmm_to_mat` *Convert Matrix Market File to Matrix*

Description

This function reads a Matrix Market file and converts it to a matrix in R.

Usage

```
fmm_to_mat(filename)
```

Arguments

`filename`	The name of the input Matrix Market file to be read.

Value

A matrix containing the data read from the Matrix Market file.

Examples

```
# Create
sample_mat <- matrix(c(1, 2, 3, 4), nrow = 2)
temp_file_mat <- tempfile(fileext = ".mtx")
write_fmm(sample_mat, temp_file_mat)
# Read
mat <- fmm_to_mat(temp_file_mat)
```

fmm_to_spam *Convert Matrix Market File to spam Sparse Matrix*

Description

This function reads a Matrix Market file and converts it to a sparse matrix using the spam package.

Usage

```
fmm_to_spam(filename)
```

Arguments

filename	The name of the input Matrix Market file to be read.

Value

A spam object containing the data read from the Matrix Market file.

Examples

```
sample_sparse <- Matrix::Matrix(c(1, 0, 0, 2), nrow = 2, sparse = TRUE)
tmp <- tempfile(fileext = ".mtx")
write_fmm(sample_sparse, tmp)
sp <- fmm_to_spam(tmp)
```

fmm_to_SparseM *Convert Matrix Market File to SparseM matrix.csr*

Description

This function reads a Matrix Market file and converts it to a matrix.csr object using the SparseM package.

Usage

```
fmm_to_SparseM(filename)
```

Arguments

filename	The name of the input Matrix Market file to be read.

Value

A matrix.csr object containing the data read from the Matrix Market file.

Examples

```
sample_sparse <- Matrix::Matrix(c(1, 0, 0, 2), nrow = 2, sparse = TRUE)
tmp <- tempfile(fileext = ".mtx")
write_fmm(sample_sparse, tmp)
csr <- fmm_to_SparseM(tmp)
```

---

fmm_to_sparse_Matrix *Convert Matrix Market File to Sparse Matrix*

Description

This function reads a Matrix Market file and converts it to a sparse matrix in R using the Matrix package.

Usage

```
fmm_to_sparse_Matrix(filename)
```

Arguments

filename	The name of the input Matrix Market file to be read.

Value

A dgCMatrix object containing the data read from the Matrix Market file.

Examples

```
# Create
sample_sparse_mat <- Matrix::Matrix(c(1, 0, 0, 2), nrow = 2, sparse = TRUE)
temp_file <- tempfile(fileext = ".mtx")
write_fmm(sample_sparse_mat, temp_file)
# Read
sparse_mat <- fmm_to_sparse_Matrix(temp_file)
```

---

fmm_to_vec *Convert Matrix Market File to Numeric Vector*

Description

This function reads a Matrix Market file and converts it to a numeric vector in R.

Usage

```
fmm_to_vec(filename)
```

Arguments

filename	The name of the input Matrix Market file to be read.

Value

A numeric vector containing the data read from the Matrix Market file.

Examples

```
# Create
sample_vec <- c(1, 2, 3)
temp_file_vec <- tempfile(fileext = ".mtx")
write_fmm(sample_vec, temp_file_vec)
# Read
vec <- fmm_to_vec(temp_file_vec)
```

---

intmat_to_fmm *Convert a Numeric Matrix to Matrix Market Format*

Description

This function takes a numeric matrix and converts it into a Matrix Market file.

Arguments

input	A numeric matrix to be converted.
filename	The name of the output file where the Matrix Market formatted data will be saved.

Value

A boolean indicating success or failure. Writes a MTX file to disk.

Examples

```
intmat <- matrix(c(1L, 2L, 3L, 4L), nrow = 2)
intmat_to_fmm(intmat, tempfile(fileext = ".mtx"))
```

intvec_to_fmm *Convert a numeric integer vector to Matrix Market Format*

Description

This function takes a numeric intvector and converts it into a Matrix Market output file.

Arguments

input	A numeric integer vector to be converted.
filename	The name of the output file where the Matrix Market formatted data will be saved.

Value

A boolean indicating success or failure. Writes a MTX file to disk.

Examples

```
intvec <- c(1L, 2L, 3L)
intvec_to_fmm(intvec, tempfile(fileext = ".mtx"))
```

mat_to_fmm *Convert a Numeric Matrix to Matrix Market Format*

Description

This function takes a numeric matrix and converts it into a Matrix Market file.

Arguments

input	A numeric matrix to be converted.
filename	The name of the output file where the Matrix Market formatted data will be saved.

Value

A boolean indicating success or failure. Writes a MTX file to disk.

Examples

```
mat <- matrix(c(1, 2, 3, 4), nrow = 2)
mat_to_fmm(mat, tempfile(fileext = ".mtx"))
```

spam_to_fmm *Convert a spam Sparse Matrix to Matrix Market Format*

Description

This function takes a spam sparse matrix and converts it into a Matrix Market file.

Usage

```
spam_to_fmm(input, filename)
```

Arguments

input	A spam sparse matrix to be converted.
filename	The name of the output file where the Matrix Market formatted data will be saved.

Value

A boolean indicating success or failure. Writes a MTX file to disk.

Examples

```
sp <- spam::spam(c(1, 0, 0, 2), nrow = 2)
spam_to_fmm(sp, tempfile(fileext = ".mtx"))
```

SparseM_to_fmm *Convert a SparseM matrix.csr to Matrix Market Format*

Description

This function takes a SparseM matrix.csr and converts it into a Matrix Market file.

Arguments

input	A matrix.csr object to be converted.
filename	The name of the output file where the Matrix Market formatted data will be saved.

Value

A boolean indicating success or failure. Writes a MTX file to disk.

Examples

```
csr <- SparseM::as.matrix.csr(matrix(c(1, 0, 0, 2), nrow = 2))
SparseM_to_fmm(csr, tempfile(fileext = ".mtx"))
```

---

sparse_Matrix_to_fmm *Convert a Sparse Numeric Matrix to Matrix Market Format*

Description

This function takes a sparse numeric matrix and converts it into a Matrix Market file.

Arguments

input	A sparse numeric matrix to be converted.
filename	The name of the output file where the Matrix Market formatted data will be saved.

Value

A boolean indicating success or failure. Writes a MTX file to disk.

Examples

```
sparse_mat <- Matrix::Matrix(c(1, 0, 0, 2), nrow = 2, sparse = TRUE)
sparse_Matrix_to_fmm(sparse_mat, tempfile(fileext = ".mtx"))
```

---

vec_to_fmm *Convert a Numeric Vector to Matrix Market Format*

Description

This function takes a numeric vector and converts it into a Matrix Market output file.

Arguments

input	A numeric vector to be converted.
filename	The name of the output file where the Matrix Market formatted data will be saved.

Value

A boolean indicating success or failure. Writes a MTX file to disk.

Examples

```
vec <- c(1, 2, 3)
vec_to_fmm(vec, tempfile(fileext = ".mtx"))
```

write_fmm *Convert Various Numeric Types to Matrix Market Format*

Description

This function takes different types of numeric inputs—vectors, matrices, and sparse matrices— and converts them into Matrix Market files. The output file is written to disk.

Usage

```
write_fmm(input, filename = "out.mtx")
```

Arguments

input	A numeric object to be converted. This can be a numeric vector, a matrix, or a sparse matrix.
filename	The name of the output file where the Matrix Market formatted data will be saved. It is recommended to use a filename ending with ".mtx" for clarity.

Value

A boolean indicating success or failure. Writes a MTX file to disk.

Examples

```
vec <- c(1, 2, 3)
mat <- matrix(c(1, 2, 3, 4), nrow = 2)
sparse_mat_diag <- Matrix::Matrix(c(1, 0, 0, 2), nrow = 2, sparse = TRUE)
## Diagonal ^-
sparse_mat <- Matrix::Matrix(c(1, 1, 0, 2), nrow = 2, sparse = TRUE)
## And not diagonal -^
write_fmm(vec, tempfile(fileext = ".mtx"))
write_fmm(mat, tempfile(fileext = ".mtx"))
write_fmm(sparse_mat_diag, tempfile(fileext = ".mtx"))
write_fmm(sparse_mat, tempfile(fileext = ".mtx"))
```

Index