

# Package ‘StructureMC’

September 20, 2019

**Type** Package

**Title** Structured Matrix Completion

**Version** 1.0

**Date** 2019-09-12

**Author** Yifu Liu and Anru Zhang

**Maintainer** Yifu Liu <2012johnnyliu@gmail.com>

**Description** Provides an efficient method to recover the missing block of an approximately low-rank matrix. Current literature on matrix completion focuses primarily on independent sampling models under which the individual observed entries are sampled independently. Motivated by applications in genomic data integration, we propose a new framework of structured matrix completion (SMC) to treat structured missingness by design [Cai T, Cai TT, Zhang A (2016) <doi:10.1080/01621459.2015.1021005>]. Specifically, our proposed method aims at efficient matrix recovery when a subset of the rows and columns of an approximately low-rank matrix are observed. The main function in our package, `smc.FUN()`, is for recovery of the missing block  $A_{22}$  of an approximately low-rank matrix  $A$  given the other blocks  $A_{11}$ ,  $A_{12}$ ,  $A_{21}$ .

**Depends** R ( $\geq 3.1.0$ ), MASS, matrixcalc

**License** GPL ( $\geq 2$ )

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-09-20 07:40:02 UTC

## R topics documented:

StructureMC-package . . . . .	2
smc.FUN . . . . .	2

<b>Index</b>	<b>4</b>
--------------	----------

---

StructureMC-package     *Structured Matrix Completion*

---

### Description

Current literature on matrix completion focuses primarily on independent sampling models under which the individual observed entries are sampled independently. Motivated by applications in genomic data integration, we propose a new framework of structured matrix completion (SMC) to treat structured missingness by design. Specifically, our proposed method aims at efficient matrix recovery when a subset of the rows and columns of an approximately low-rank matrix are observed. The main function in our package, smc.FUN, is for recovery of the missing block A22 of an approximately low-rank matrix A given the other blocks A11, A12, A21.

### Author(s)

Yifu Liu and Anru Zhang

Maintainer: Yifu Liu (2012johnnyliu@gmail.com)

### References

Cai, T., Cai, T. T., & Zhang, A. (2015). Structured Matrix Completion with Applications to Genomic Data Integration. *Journal of the American Statistical Association*.

---

smc.FUN     *Structured Matrix Completion*

---

### Description

The main function in our package, smc.FUN, is for recovery of the missing block A22 of an approximately low-rank matrix A given the other blocks A11, A12, A21.

### Usage

```
smc.FUN(A.mat, c_T, row_thresh, m1, m2)
```

### Arguments

A.mat	The approximately low-rank matrix that we want to recover
c_T	c_T is the thresholding level, the default value is 2.
row_thresh	is row thresholding
m1	number of rows of block A11
m2	number of columns of block A11

**Value**

smc.FUN()	returns the missing block of initial input matrix which has the following components:
A.mat	the approximately low-rank matrix with given value of other three blocks and NA values for the block that user want to recover.
c_T	is the thresholding level, the default value is 2.
row_thresh	it is the row thresholding which is "True" under default. Otherwise, it can be "False" to make it column thresholding.
m1	number of rows of block A11. It can be used to calculate the rest of dimension of blocks in A.mat with m2.
m2	number of columns of block A11. It can be used to calculate the rest of dimension of blocks in A.mat with m1.

**Author(s)**

Yifu Liu and Anru Zhang

**References**

Cai, T., Cai, T. T., & Zhang, A. (2015). Structured Matrix Completion with Applications to Genomic Data Integration. *Journal of the American Statistical Association*.

**Examples**

```
#dimension of matrix A with row number p1 = 10 and column number p2 = 9
p1 = 60
p2 = 50
m1 = 55 #row number of A11
m2 = 45 #column number of A11
A = matrix(rnorm(300, mean = 0.05, sd = 0.1), p1, p2)

#different blocks of our matrix A that are used to approximate missing block A22
A11 = A[1:m1, 1:m2]
A12 = A[1:m1, (1+m2):p2]
A21 = A[(1+m1):p1, 1:m2]
Arecovey = rbind(cbind(A11,A12),cbind(A21,matrix(NA,nrow=p1-m1,ncol=p2-m2)))

#recover the missing block A22 where A22 is now filled with approximate values by our algorithm.
A22 = smc.FUN(Arecovey, 2, "True", m1, m2)
```

# Index

\*Topic **Genomic**

smc.FUN, [2](#)

\*Topic **approximate**

smc.FUN, [2](#)

\*Topic **completion**

smc.FUN, [2](#)

\*Topic **low**

smc.FUN, [2](#)

\*Topic **matrix**

smc.FUN, [2](#)

\*Topic **rank**

smc.FUN, [2](#)

\*Topic **structure**

smc.FUN, [2](#)

smc.FUN, [2](#)

StructureMC (StructureMC-package), [2](#)

StructureMC-package, [2](#)