

Package ‘csmGmm’

September 16, 2025

Type Package

Title Conditionally Symmetric Multidimensional Gaussian Mixture Model

Version 0.4.0

Description Implements the conditionally symmetric multidimensional Gaussian mixture model (csmGmm) for large-scale testing of composite null hypotheses in genetic association applications such as mediation analysis, pleiotropy analysis, and replication analysis. In such analyses, we typically have J sets of K test statistics where K is a small number (e.g. 2 or 3) and J is large (e.g. 1 million). For each one of the J sets, we want to know if we can reject all K individual nulls. Please see the vignette for a quickstart guide. The paper describing these methods is “Testing a Large Number of Composite Null Hypotheses Using Conditionally Symmetric Multidimensional Gaussian Mixtures in Genome-Wide Studies” by Sun R, McCaw Z, & Lin X (Journal of the American Statistical Association 2025, <doi:10.1080/01621459.2024.2422124>).

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

Imports dplyr, mvtnorm, rlang, magrittr

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Ryan Sun [aut, cre]

Maintainer Ryan Sun <ryansun.work@gmail.com>

Repository CRAN

Date/Publication 2025-09-16 21:40:02 UTC

Contents

calc_dens_cor	2
calc_dens_ind_2d	3
calc_dens_ind_3d	3

calc_dens_ind_multiple	4
check_incongruous	5
find_2d	5
find_3d	6
find_max_means	7
symm_fit_cor_EM	7
symm_fit_cor_EM_fulllik	8
symm_fit_cor_EM_noAssumption	10
symm_fit_cor_EM_rho	11
symm_fit_ind_EM	12
symm_fit_ind_EM_noAssumption	13

Index	15
--------------	-----------

calc_dens_cor	<i>calc_dens_cor.R</i>
---------------	------------------------

Description

For $J \times 2$ matrix of J sets, calculate the density of bivariate normal under fitted c-csmGmm.

Usage

```
calc_dens_cor(x, Zmat, corMat, log = FALSE)
```

Arguments

<code>x</code>	2*1 vector of means.
<code>Zmat</code>	$J \times 2$ matrix of test statistics.
<code>corMat</code>	2*2 matrix describing correlation structure of test statistics.
<code>log</code>	return logarithm of density

Value

A $J \times 1$ vector of densities for each row of `Zmat`.

Examples

```
x <- c(0, 0)
Zmat <- cbind(rnorm(10^5), rnorm(10^5))
calc_dens_cor(x, Zmat, corMat = cor(Zmat))
```

calc_dens_ind_2d	<i>calc_dens_ind.R</i>
------------------	------------------------

Description

Calculate J bivariate normal densities (both dimensions are independent) under fitted csmGmm.

Usage

```
calc_dens_ind_2d(x, Zmat)
```

Arguments

x	2*1 vector of means.
Zmat	J*2 matrix of test statistics.

Value

A $J*1$ vector of densities for each row of Zmat.

Examples

```
x <- c(0, 0)
Zmat <- cbind(rnorm(10^5), rnorm(10^5))
calc_dens_ind_2d(x, Zmat)
```

calc_dens_ind_3d	<i>Calculate J trivariate normal densities (all dimensions are independent) under fitted csmGmm.</i>
------------------	--

Description

Calculate J trivariate normal densities (all dimensions are independent) under fitted csmGmm.

Usage

```
calc_dens_ind_3d(x, Zmat)
```

Arguments

x	3*1 vector of means.
Zmat	J*3 matrix of test statistics.

Value

A $J*1$ vector of densities for each row of Zmat.

Examples

```
x <- c(0, 0)
Zmat <- cbind(rnorm(10^5), rnorm(10^5), rnorm(10^5))
calc_dens_ind_3d(x, Zmat)
```

calc_dens_ind_multiple

Calculate the density of K-dimensional multivariate normal (all dimensions are independent) under fitted acsGmm.

Description

Calculate the density of K-dimensional multivariate normal (all dimensions are independent) under fitted acsGmm.

Usage

```
calc_dens_ind_multiple(x, Zmat)
```

Arguments

x	K*1 vector of means.
Zmat	J*K matrix of test statistics.

Value

A J*1 vector of densities for each row of Zmat.

Examples

```
x <- c(0, 0)
Zmat <- cbind(rnorm(10^5), rnorm(10^5), rnorm(10^5), rnorm(10^5))
calc_dens_ind_multiple(x, Zmat)
```

check_incongruous	<i>check_incongruous.R</i>
-------------------	----------------------------

Description

Check the number of sets of test statistics that have a higher (less significant) lfdR value than other sets with test statistics of uniformly smaller magnitudes.

Usage

```
check_incongruous(zMatrix, lfdrVec)
```

Arguments

zMatrix	J*K vector of all test statistics.
lfdrVec	J*1 vector of lfdr values corresponding to each set of test statistics.

Value

A vector with all the indices of all sets that have a higher lfdr value those a set with smaller test statistic magnitudes.

Examples

```
zMatrix <- cbind(rnorm(10^4), rnorm(10^4))
lfdrVec <- runif(10^4)
check_incongruous(zMatrix = zMatrix, lfdrVec = lfdrVec)
```

find_2d	<i>Tells if row x if allTestStats is an incongruous result (has a higher lfdr than a set of test statistics with lower magnitudes). For K=2 case.</i>
---------	---

Description

Tells if row x if allTestStats is an incongruous result (has a higher lfdr than a set of test statistics with lower magnitudes). For K=2 case.

Usage

```
find_2d(x, allTestStats)
```

Arguments

x	Scalar, which row of allTestStats to check.
allTestStats	J*K vector of all test statistics.

Value

A scalar denoting the number of sets with lower lfdr and test statistics of lower magnitude. 0 means congruous result.

Examples

```
zMatrix <- cbind(rnorm(10^4), rnorm(10^4))
find_2d(x = 5, allTestStats = zMatrix)
```

find_3d	<i>Tells if row x if allTestStats is an incongruous result (has a higher lfdr than a set of test statistics with lower magnitudes). For K=3 case.</i>
---------	---

Description

Tells if row x if allTestStats is an incongruous result (has a higher lfdr than a set of test statistics with lower magnitudes). For K=3 case.

Usage

```
find_3d(x, allTestStats)
```

Arguments

x Scalar, which row of allTestStats to check.
allTestStats J*K vector of all test statistics.

Value

A scalar denoting the number of sets with lower lfdr and test statistics of lower magnitude. 0 means congruous result.

Examples

```
zMatrix <- cbind(rnorm(10^4), rnorm(10^4), rnorm(10^4))
find_3d(x = 5, allTestStats = zMatrix)
```

find_max_means	<i>find_max_means.R</i>
----------------	-------------------------

Description

Find maximum means for each dimension in null settings.

Usage

```
find_max_means(muInfo)
```

Arguments

muInfo	A list with 2^K elements, where each element is a matrix with K rows and M_b columns.
--------	---

Value

A $K \times 1$ vector of the maximum means for each dimension under the null.

Examples

```
initMuList <- list(matrix(data=0, nrow=2, ncol=1), matrix(data=c(0, 3, 0, 6), nrow=2),  
matrix(data=c(3, 0, 6, 0), nrow=2), matrix(data=c(8, 8), nrow=2))  
find_max_means(initMuList)
```

symm_fit_cor_EM	<i>symm_fit_cor.R</i>
-----------------	-----------------------

Description

Fit the correlated csmGmm for sets of correlated elements. Currently restricted to $K=2$.

Usage

```
symm_fit_cor_EM(  
  testStats,  
  corMat,  
  initMuList,  
  initPiList,  
  eps = 10(-5),  
  checkpoint = TRUE  
)
```

Arguments

testStats	J*K matrix of test statistics where J is the number of sets and K is number of elements in each set.
corMat	K*K matrix that describes the correlation structure of each set.
initMuList	List of 2^K elements where each element is a matrix with K rows and number of columns equal to the number of possible mean vectors for that binary group.
initPiList	List of 2^K elements where each element is a vector with number of elements equal to the number of possible mean vectors for that binary group.
eps	Scalar, stop the EM algorithm when L2 norm of difference in parameters is less than this value.
checkpoint	Boolean, set to TRUE to print iterations of EM.

Value

A list with the elements:

muInfo	List with same dimensions as initMuList, holds the final mean parameters.
piInfo	List with same dimensions as initPiList, holds the final probability parameters.
iter	Number of iterations run in EM algorithm.
lfdrResults	J*1 vector of all lfdr statistics.

Examples

```
set.seed(0)
corMat <- matrix(data=c(1, 0.3, 0.3, 1), nrow=2)
testStats <- rbind(mvtnorm::rmvnorm(n=200, mean=c(3, 0), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, 4), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=10^4 - 500, mean=c(0, 0), sigma=corMat))
initMuList <- list(matrix(data=0, nrow=2, ncol=1), matrix(data=c(0, 3), nrow=2),
matrix(data=c(4, 0), nrow=2), matrix(data=c(5, 5), nrow=2))
initPiList <- list(c(0.9), c(0.04), c(0.04), c(0.02))
results <- symm_fit_cor_EM(testStats = testStats, corMat = cor(testStats),
initMuList = initMuList, initPiList = initPiList)
```

symm_fit_cor_EM_fulllik

symm_fit_cor_fulllik.R

Description

Full likelihood, block correlation, blocks of size 2

Usage

```

symm_fit_cor_EM_fulllik(
  testStats,
  corMat,
  initMuList,
  initPiList,
  eps = 10^(-5),
  checkpoint = TRUE
)

```

Arguments

testStats	J*K matrix of test statistics where J is the number of sets and K is number of elements in each set.
corMat	K*K matrix that describes the correlation structure of each 2 by 2 block.
initMuList	List of 2^K elements where each element is a matrix with K rows and number of columns equal to the number of possible mean vectors for that binary group.
initPiList	List of 2^K elements where each element is a vector with number of elements equal to the number of possible mean vectors for that binary group.
eps	Scalar, stop the EM algorithm when L2 norm of difference in parameters is less than this value.
checkpoint	Boolean, set to TRUE to print iterations of EM.

Value

A list with the elements:

muInfo	List with same dimensions as initMuList, holds the final mean parameters.
piInfo	List with same dimensions as initPiList, holds the final probability parameters.
iter	Number of iterations run in EM algorithm.
lfdrResults	J*1 vector of all lfdr statistics.

Examples

```

set.seed(0)
testStats <- cbind(rnorm(10^4), rnorm(10^4))
testStats[1:100, 1] <- rnorm(100, mean=3)
testStats[101:200, 1] <- rnorm(100, mean=5)
testStats[201:300, 2] <- rnorm(100, mean=4)
testStats[301:400, 1:2] <- rnorm(200, mean=7)
initMuList <- list(matrix(data=0, nrow=2, ncol=1), matrix(data=c(0, 3), nrow=2, ncol=1),
matrix(data=c(3, 0), nrow=2, ncol=1), matrix(data=c(6, 6), nrow=2, ncol=1))
initPiList <- list(c(0.9), c(0.04),c(0.04), c(0.02))
results <- symm_fit_cor_EM_fulllik(testStats = testStats, corMat=diag(c(1,1)),
initMuList = initMuList, initPiList = initPiList)

```

```
symm_fit_cor_EM_noAssumption
      symm_fit_cor_noAssumption.R
```

Description

Fit the correlated csmGmm for sets of correlated elements, but we don't assume that the means in the composite alternative are greater in magnitude than those in the composite null.

Usage

```
symm_fit_cor_EM_noAssumption(
  testStats,
  corMat,
  initMuList,
  initPiList,
  eps = 10-5,
  checkpoint = TRUE
)
```

Arguments

testStats	J*K matrix of test statistics where J is the number of sets and K is number of elements in each set.
corMat	K*K matrix that describes the correlation structure of each set.
initMuList	List of 2^K elements where each element is a matrix with K rows and number of columns equal to the number of possible mean vectors for that binary group.
initPiList	List of 2^K elements where each element is a vector with number of elements equal to the number of possible mean vectors for that binary group.
eps	Scalar, stop the EM algorithm when L2 norm of difference in parameters is less than this value.
checkpoint	Boolean, set to TRUE to print iterations of EM.

Value

A list with the elements:

muInfo	List with same dimensions as initMuList, holds the final mean parameters.
piInfo	List with same dimensions as initPiList, holds the final probability parameters.
iter	Number of iterations run in EM algorithm.
lfdrResults	J*1 vector of all lfdr statistics.

Examples

```

set.seed(0)
corMat <- matrix(data=c(1, 0.3, 0.3, 1), nrow=2)
testStats <- rbind(mvtnorm::rmvnorm(n=200, mean=c(3, 0), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, 4), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=10^4 - 500, mean=c(0, 0), sigma=corMat))
initMuList <- list(matrix(data=0, nrow=2, ncol=1), matrix(data=c(0, 3), nrow=2),
matrix(data=c(4, 0), nrow=2), matrix(data=c(5, 5), nrow=2))
initPiList <- list(c(0.9), c(0.04), c(0.04), c(0.02))
results <- symm_fit_cor_EM_noAssumption(testStats = testStats,
corMat = cor(testStats), initMuList = initMuList, initPiList = initPiList)

```

symm_fit_cor_EM_rho *symm_fit_cor_rho.R*

Description

Fit the correlated csmGmm for sets of correlated elements. Also fits the correlation parameter in EM algorithm.

Usage

```

symm_fit_cor_EM_rho(
  testStats,
  initRho,
  initMuList,
  initPiList,
  eps = 10^(-5),
  checkpoint = TRUE
)

```

Arguments

testStats	J*K matrix of test statistics where J is the number of sets and K is number of elements in each set.
initRho	Initial value of rho, any reasonable guess should be ok.
initMuList	List of 2^K elements where each element is a matrix with K rows and number of columns equal to the number of possible mean vectors for that binary group.
initPiList	List of 2^K elements where each element is a vector with number of elements equal to the number of possible mean vectors for that binary group.
eps	Scalar, stop the EM algorithm when L2 norm of difference in parameters is less than this value.
checkpoint	Boolean, set to TRUE to print iterations of EM.

Value

A list with the elements:

<code>muInfo</code>	List with same dimensions as <code>initMuList</code> , holds the final mean parameters.
<code>piInfo</code>	List with same dimensions as <code>initPiList</code> , holds the final probability parameters.
<code>iter</code>	Number of iterations run in EM algorithm.
<code>lfdrResults</code>	$J \times 1$ vector of all lfdr statistics.

Examples

```
set.seed(0)
corMat <- matrix(data=c(1, 0.3, 0.3, 1), nrow=2)
testStats <- rbind(mvtnorm::rmvnorm(n=200, mean=c(3, 0), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, 4), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=10^4 - 500, mean=c(0, 0), sigma=corMat))
initMuList <- list(matrix(data=0, nrow=2, ncol=1), matrix(data=c(0, 3), nrow=2),
matrix(data=c(4, 0), nrow=2), matrix(data=c(5, 5), nrow=2))
initPiList <- list(c(0.9), c(0.04), c(0.04), c(0.02))
results <- symm_fit_cor_EM_rho(testStats = testStats,
initRho = 0.1, initMuList = initMuList, initPiList = initPiList)
```

`symm_fit_ind_EM`

symm_fit_ind.R

Description

Fit the conditionally symmetric multidimensional Gaussian mixture model for sets of independent elements

Usage

```
symm_fit_ind_EM(
  testStats,
  initMuList,
  initPiList,
  sameDirAlt = FALSE,
  eps = 10^(-5),
  checkpoint = TRUE
)
```

Arguments

testStats	J*K matrix of test statistics where J is the number of sets and K is number of elements in each set.
initMuList	List of 2^K elements where each element is a matrix with K rows and number of columns equal to the number of possible mean vectors for that binary representation.
initPiList	List of 2^K elements where each element is a vector with number of elements equal to the number of possible mean vectors for that binary representation.
sameDirAlt	Boolean, set to TRUE for replication testing, which uses a smaller alternative space.
eps	Scalar, stop the EM algorithm when L2 norm of difference in parameters is less than this value.
checkpoint	Boolean, set to TRUE to print iterations of EM

Value

A list with the elements:

muInfo	List with same dimensions as initMuList, holds the final mean parameters.
piInfo	List with same dimensions as initPiList, holds the final mixture proportions
iter	Number of iterations run in EM algorithm.
lfdrResults	J*1 vector of all lfdr statistics.

Examples

```
set.seed(0)
testStats <- cbind(rnorm(10^4), rnorm(10^4))
testStats[1:200, 1] <- rnorm(100, mean=3)
testStats[201:400, 1] <- rnorm(100, mean=5)
testStats[401:600, 2] <- rnorm(100, mean=3)
testStats[601:800, 2] <- rnorm(100, mean=5)
testStats[801:1000, 1:2] <- rnorm(200, mean=7)
initMuList <- list(matrix(data=0, nrow=2, ncol=1), matrix(data=c(0, 3, 0, 5), nrow=2, ncol=2),
matrix(data=c(3, 0, 5, 0), nrow=2, ncol=2), matrix(data=c(7, 7), nrow=2, ncol=1))
initPiList <- list(c(0.9), c(0.02, 0.02), c(0.02, 0.02), c(0.02))
results <- symm_fit_ind_EM(testStats = testStats, initMuList = initMuList, initPiList = initPiList)
```

symm_fit_ind_EM_noAssumption

symm_fit_ind_noAssumption.R

Description

Fit the conditionally symmetric multidimensional Gaussian mixture model for sets of independent elements, but we don't assume that the means in the composite alternative are greater in magnitude than those in the composite null.

Usage

```

symm_fit_ind_EM_noAssumption(
  testStats,
  initMuList,
  initPiList,
  sameDirAlt = FALSE,
  eps = 10(-5),
  checkpoint = TRUE
)

```

Arguments

testStats	J*K matrix of test statistics where J is the number of sets and K is number of elements in each set.
initMuList	List of 2^K elements where each element is a matrix with K rows and number of columns equal to the number of possible mean vectors for that binary representation.
initPiList	List of 2^K elements where each element is a vector with number of elements equal to the number of possible mean vectors for that binary representation.
sameDirAlt	Boolean, set to TRUE for replication testing, which uses a smaller alternative space.
eps	Scalar, stop the EM algorithm when L2 norm of difference in parameters is less than this value.
checkpoint	Boolean, set to TRUE to print iterations of EM

Value

A list with the elements:

muInfo	List with same dimensions as initMuList, holds the final mean parameters.
piInfo	List with same dimensions as initPiList, holds the final mixture proportions
iter	Number of iterations run in EM algorithm.
lfdrResults	J*1 vector of all lfdr statistics.

Examples

```

set.seed(0)
testStats <- cbind(rnorm(10^4), rnorm(10^4))
testStats[1:200, 1] <- rnorm(100, mean=3)
testStats[201:400, 1] <- rnorm(100, mean=5)
testStats[401:600, 2] <- rnorm(100, mean=3)
testStats[601:800, 2] <- rnorm(100, mean=5)
testStats[801:1000, 1:2] <- rnorm(200, mean=7)
initMuList <- list(matrix(data=0, nrow=2, ncol=1), matrix(data=c(0, 3, 0, 5), nrow=2, ncol=2),
matrix(data=c(3, 0, 5, 0), nrow=2, ncol=2), matrix(data=c(7, 7), nrow=2, ncol=1))
initPiList <- list(c(0.9), c(0.02, 0.02),c(0.02, 0.02), c(0.02))
results <- symm_fit_ind_EM_noAssumption(testStats = testStats,
initMuList = initMuList, initPiList = initPiList)

```

Index

[calc_dens_cor](#), [2](#)
[calc_dens_ind_2d](#), [3](#)
[calc_dens_ind_3d](#), [3](#)
[calc_dens_ind_multiple](#), [4](#)
[check_incongruous](#), [5](#)

[find_2d](#), [5](#)
[find_3d](#), [6](#)
[find_max_means](#), [7](#)

[symm_fit_cor_EM](#), [7](#)
[symm_fit_cor_EM_fulllik](#), [8](#)
[symm_fit_cor_EM_noAssumption](#), [10](#)
[symm_fit_cor_EM_rho](#), [11](#)
[symm_fit_ind_EM](#), [12](#)
[symm_fit_ind_EM_noAssumption](#), [13](#)