Package 'csmGmm'

November 4, 2026

Type Package

Title Conditionally Symmetric Multidimensional Gaussian Mixture Model

Version 0.6.0

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

Depends R (>= 4.1.0)

Imports dplyr, mvtnorm, curl, data.table, ggplot2 (>= 3.5.0), rlang, magrittr, stats, utils

Suggests knitr, rmarkdown, R.utils

VignetteBuilder knitr

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Ryan Sun [aut, cre], Emily Kim [aut]

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

Repository CRAN

Date/Publication 2026-11-04 06:10:02 UTC

Contents

calc_dens_cor . . . 2
calc_dens_ind_2d . . . 3
calc_dens_ind_3d . . . 4
calc_dens_ind_multiple . . . 4
check_incongruous . . . 5
create_plots . . . 6
find_2d . . . 6
find_3d . . . 7
find_4d . . . 8
find_max_means . . . 8
generate_init_lists . . . 9
get_marginal_mixture . . . 10
plot_fit_hist . . . 10
prepare_csmgmm_data . . . 11
process_lfdr_results . . . 13
read_gwas_one . . . 14
read_gwas_set . . . 15
symm_fit_cor_EM . . . 16
symm_fit_cor_EM_fulllik . . . 17
symm_fit_cor_EM_noAssumption . . . 18
symm_fit_cor_EM_rho . . . 20
symm_fit_ind_EM . . . 21
symm_fit_ind_EM_noAssumption . . . 23

Index **25**

`calc_dens_cor` *calc_dens_cor.R*

Description

For J*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

`x`	2*1 vector of means.
`Zmat`	J*2 matrix of test statistics.
`corMat`	2*2 matrix describing correlation structure of test statistics.
`log`	return logarithm of density

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_cor(x, Zmat, corMat = cor(Zmat))
```

`calc_dens_ind_2d` *calc_dens_ind.R*

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

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 *check_incongruous.R*

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)
```

create_plots *create_plots.R*

Description

Creates lfdr-value Manhttan plots and z-score plots to visualize composite null hypothesis testing results

Usage

```
create_plots(plotData)
```

Arguments

plotData	data.frame with columns labeled (exactly) Chr, BP, cum_avg_lfdr, Z_Dataset1, Z_Dataset2

Value

A list with the elements:

manPlot	A ggplot object of the Manhattan plot of the lfdr-values at which each SNP is significant
zPlot	A ggplot object of a scatterplot comparing the z-scores for the first two datasets

Examples

```
set.seed(0)
plotData <- data.frame(Chr = rep(1:22, each=100), BP=runif(n=2200, min=1, max=10^8),
cum_avg_lfdr=runif(n=2200, min=0, max=22),
Z_Dataset1=rnorm(n=2200), Z_Dataset2=rnorm(n=2200))
create_plots(plotData = plotData)
```

find_2d *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.*

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 — *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.*

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_4d` *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=4 case.*

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=4 case.

Usage

```
find_4d(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), rnorm(10^4))
find_4d(x = 5, allTestStats = zMatrix)
```

`find_max_means` *find_max_means.R*

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 Mb columns.

Value

A K*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)
```

`generate_init_lists` *generate_init_lists.R*

Description

Generates initMuList and initPiList parameters for symm_fit_ind_EM() function.

Usage

```
generate_init_lists(K)
```

Arguments

`K`	Scalar, number of datasets in the analysis.

Value

A list with the elements:

`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.

Examples

```
K <- 2
generate_init_lists(K)
```

`get_marginal_mixture` *diagnostic_plot.R*

Description

Collapse a fitted csmGmm into the one-dimensional marginal mixture for each dimension.

Usage

```
get_marginal_mixture(fitResults)
```

Arguments

`fitResults`	The list returned by symm_fit_ind_EM() (only muInfo and piInfo are used).

Details

Under the independence model, the marginal distribution of column k of testStats is a univariate Gaussian mixture with unit variance. Every configuration where dimension k is null contributes mass at mean 0, and every configuration where dimension k is non-null contributes half its mass at +mu and half at -mu.

Value

A data.frame with columns dim (dimension index), comp ("null", "negAlt", or "posAlt"), mean, and weight. Weights sum to 1 within each dimension.

Examples

```
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))
get_marginal_mixture(list(muInfo = initMuList, piInfo = initPiList))
```

`plot_fit_hist` *Histogram of the observed test statistics with the fitted marginal mixture density overlaid.*

Description

For each dimension, plots a histogram of the observed test statistics along with the fitted marginal mixture density (scaled to counts), the null N(0,1) component, and the alternative component (all non-null components combined). Pink bars show observed counts minus the counts expected under the fitted null component, similar to the diagnostics in repfdr::ztobins().

Usage

```
plot_fit_hist(testStats, fitResults, nBins = 100, zLim = NULL, yLim = NULL)
```

Arguments

testStats	J*K matrix of test statistics that was passed to symm_fit_ind_EM().
fitResults	The list returned by symm_fit_ind_EM().
nBins	Number of histogram bins.
zLim	Optional length-2 vector of x-axis limits; defaults to the range of each column. Statistics outside zLim are dropped from the histogram but still count toward the total J.
yLim	Optional length-2 vector of y-axis limits. The alternative components are usually tiny relative to the null, so setting e.g. yLim = c(0, 50) zooms in on the tails without dropping any bars.

Value

A list of K ggplot objects, one for each dimension.

Examples

```
set.seed(0)
testStats <- cbind(rnorm(10^4), rnorm(10^4))
testStats[1:200, 1] <- rnorm(200, mean=3)
testStats[201:400, 1] <- rnorm(200, mean=5)
testStats[401:600, 2] <- rnorm(200, mean=3)
testStats[601:800, 2] <- rnorm(200, mean=5)
testStats[801:1000, 1:2] <- rnorm(400, 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, sameDirAlt = TRUE,
initMuList = initMuList, initPiList = initPiList)
histPlots <- plot_fit_hist(testStats = testStats, fitResults = results)
histPlots[[1]]
```

---

prepare_csmgmm_data *prepare_csmgmm_data.R*

Description

Prepares J*K matrix of summary statistics for GWAS replication studies where J is the number of SNPs and K is the number of cohorts.

Usage

```
prepare_csmgmm_data(
  datasets,
  dataset_names = NULL,
  z_cap = 8.1,
  strand_aligned = FALSE
)
```

Arguments

datasets	list holding K datasets to use for replication analysis
dataset_names	optional vector of names for each dataset; otherwise function uses Dataset1,.., DatasetK as default.
z_cap	optional number to cap the most extreme z-scores, we recommend setting at 8.1 as R is not very accurate for very large z-scores
strand_aligned	Boolean that describes whether the strands are pre-aligned. If not, removes all palindromic SNPs.

Value

A list with the elements:

extra_dat	A J*K matrix with J matching SNPs across K datasets, holds relevant SNP information.
clean_dat	J*K matrix of test statistics, holds J Z-scores for each dataset.
n_snps	A scalar returning the number of matching SNPs across all K datasets.

Examples

```
library(dplyr)
set.seed(0)
J <- 1000
alleles <- c("A", "C", "T", "G")
dat1 <- data.frame(zstat = rnorm(J)) %>% mutate(se = runif(n=J, min=0.0005, max=0.0015)) %>%
mutate(beta = zstat * se, chr=1, pos=1:J, a1=sample(x=alleles, J, replace=TRUE),
a2=sample(x=alleles, J, replace=TRUE))
dat2 <- data.frame(zstat = rnorm(J)) %>% mutate(se = runif(n=J, min=0.0005, max=0.0015)) %>%
mutate(beta = zstat * se, chr=1, pos=1:J, a1=sample(x=alleles, J, replace=TRUE),
a2=sample(x=alleles, J, replace=TRUE))
dat3 <- data.frame(zstat = rnorm(J)) %>% mutate(se = runif(n=J, min=0.0005, max=0.0015)) %>%
mutate(beta = zstat * se, chr=1, pos=1:J, a1=sample(x=alleles, J, replace=TRUE),
a2=sample(x=alleles, J, replace=TRUE))
datasets = list(dat1, dat2, dat3)
prepare_csmgmm_data(datasets = datasets)
```

process_lfdr_results *process_lfdr_results.R*

Description

Sorts lfdrResults and calculates cumulative averages to return significant SNPs.

Usage

```
process_lfdr_results(orig_data, lfdrResults, fdr_threshold = 0.1)
```

Arguments

orig_data	data.frame with J rows, one for each composite null hypothesis being tested, holds data such as position, z-statistics, etc.
lfdrResults	J*1 vector of all lfdr-values, the jth value corresponds to the jth row of orig_data.
fdr_threshold	Scalar between 0 and 1 determining the percentage of false discoveries, default set to 0.1.

Value

A list with elements:

all_snps_info	A data.frame of all SNPs, sorted by the lfdr-value at which they are significant (cum_avg_lfdr column)
sig_snps_info	A data.frame of the significant SNPs, sorted by the lfdr-value at which they are significant (cum_avg_lfdr column)
n_significant	A scalar returning the number of significant SNPs.

Examples

```
set.seed(0)
orig_data <- data.frame(
variant_id = sample(c("A","B","C","D"), 10^4, replace = TRUE),
Chr = sample(1:22, 10^4, replace = TRUE),
BP = sample(1e6:2e6, 10^4),
Z_Dataset1  = rnorm(10^4),
Z_Dataset2    = rnorm(10^4)
)
lfdrResults <- runif(10^4)
process_lfdr_results(orig_data = orig_data, lfdrResults = lfdrResults, fdr_threshold = 0.05)
```

read_gwas_one *read_gwas.R*

Description

Read 1 GWAS Catalog summary-statistics file into the current environment.

Usage

```
read_gwas_one(
  url,
  cache_dir = tempdir(),
  overwrite = FALSE,
  max_tries = 5,
  fread_args = list()
)
```

Arguments

url	Full https URL to a .tsv.gz file on the GWAS Catalog FTP.
cache_dir	Directory to keep downloaded files (default: tempdir()). Use a persistent path (e.g. "~/gwas_cache") to avoid re-downloading across R sessions.
overwrite	If TRUE, redownload even if a cached file exists.
max_tries	Number of download attempts before giving up.
fread_args	Named list of extra args forwarded to data.table::fread (e.g. list(select = c("variant_id", "p_value"))).

Details

Strategy: download each .tsv.gz to a local cache directory, then parse with data.table::fread (which transparently handles gzip via local path). This is more robust than streaming over HTTPS for files of this size (~1 GB each): downloads can be resumed/retried, and re-reads are cheap once cached.

Value

A data.table with the downloaded data

Examples

```
# Ensure internet is available before running the example
if (curl::has_internet()) {
  try({
   dat <- read_gwas_one(paste0("http://ftp.ebi.ac.uk/pub/databases/gwas/summary_statistics/",
   "GCST005001-GCST006000/GCST005527/tsoi_2012_23143594_pso_efo0000676_1_ichip.sumstats.tsv.gz"))
    head(dat)
  })
}
```

read_gwas_set *Read 2-5 GWAS Catalog summary-statistics files into the current environment.*

Description

Read 2-5 GWAS Catalog summary-statistics files into the current environment.

Usage

```
read_gwas_set(urls, ...)
```

Arguments

urls	Character vector of 2-5 https URLs to .tsv.gz files. If the vector has names, those names are used in the returned list; otherwise entries are named gwas1, gwas2, ...
...	Forwarded to read_gwas_one (cache_dir, overwrite, max_tries, fread_args).

Value

Named list of data.tables, one per input URL.

Examples

```
# Ensure internet is available before running the example
if (curl::has_internet()) {
  try({
    dat <- read_gwas_set(c(
      paste0("http://ftp.ebi.ac.uk/pub/databases/gwas/summary_statistics/",
    "GCST005001-GCST006000/GCST005527/tsoi_2012_23143594_pso_efo0000676_1_ichip.sumstats.tsv.gz"),
      paste0("http://ftp.ebi.ac.uk/pub/databases/gwas/summary_statistics/",
    "GCST003001-GCST004000/GCST003044/CD_trans_ethnic_association_summ_stats_b37.txt.gz")
    ))
    gwas1 <- dat$gwas1
    gwas2 <- dat$gwas2
    head(gwas1)
  })
}
```

`symm_fit_cor_EM` *symm_fit_cor.R*

Description

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

Usage

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

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.
`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 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(-3, 0), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, 3), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, -3), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, -7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(-7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(-7, -7), sigma=corMat),
mvtnorm::rmvnorm(n=10^4 - 1200, 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, sameDirAlt = TRUE)
```

`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,
  sameDirAlt,
  eps = 10^(-5),
  checkpoint = FALSE
)
```

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.
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 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(-3, 0), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, 3), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, -3), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, -7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(-7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(-7, -7), sigma=corMat),
mvtnorm::rmvnorm(n=10^4 - 1200, mean=c(0, 0), sigma=corMat))
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, sameDirAlt = TRUE)
```

---

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,
  sameDirAlt,
  eps = 10^(-5),
  checkpoint = FALSE
)
```

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.
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 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(-3, 0), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, 3), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, -3), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, -7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(-7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(-7, -7), sigma=corMat),
```

```
mvtnorm::rmvnorm(n=10^4 - 1200, 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, sameDirAlt = TRUE)
```

`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,
  sameDirAlt,
  eps = 10^(-5),
  checkpoint = FALSE
)
```

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.
`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 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(-3, 0), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, 3), sigma=corMat),
mvtnorm::rmvnorm(n=200, mean=c(0, -3), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(7, -7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(-7, 7), sigma=corMat),
mvtnorm::rmvnorm(n=100, mean=c(-7, -7), sigma=corMat),
mvtnorm::rmvnorm(n=10^4 - 1200, 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, sameDirAlt = TRUE)
```

---

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,
  eps = 10^(-5),
  checkpoint = FALSE
)
```

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(200, mean=3)
testStats[201:400, 1] <- rnorm(200, mean=-3)
testStats[401:600, 2] <- rnorm(200, mean=3)
testStats[601:800, 2] <- rnorm(200, mean=-3)
testStats[801:900, 1:2] <- rnorm(200, mean=-7)
testStats[901:1000, 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, sameDirAlt = TRUE,
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,
  eps = 10^(-5),
  checkpoint = FALSE
)
```

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(200, mean=3)
testStats[201:400, 1] <- rnorm(200, mean=-3)
testStats[401:600, 2] <- rnorm(200, mean=3)
testStats[601:800, 2] <- rnorm(200, mean=-3)
testStats[801:900, 1:2] <- rnorm(200, mean=-7)
testStats[901:1000, 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, sameDirAlt = TRUE,
initMuList = initMuList, initPiList = initPiList)
```

Index

calc_dens_cor, 2
calc_dens_ind_2d, 3
calc_dens_ind_3d, 4
calc_dens_ind_multiple, 4
check_incongruous, 5
create_plots, 6

find_2d, 6
find_3d, 7
find_4d, 8
find_max_means, 8

generate_init_lists, 9
get_marginal_mixture, 10

plot_fit_hist, 10
prepare_csmgmm_data, 11
process_lfdr_results, 13

read_gwas_one, 14
read_gwas_set, 15

symm_fit_cor_EM, 16
symm_fit_cor_EM_fulllik, 17
symm_fit_cor_EM_noAssumption, 18
symm_fit_cor_EM_rho, 20
symm_fit_ind_EM, 21
symm_fit_ind_EM_noAssumption, 23