Package 'sdafilter'

February 13, 2026

Title Symmetrized Data Aggregation

Version 1.0.1

Author Lilun Du [aut, cre],
Xu Guo [aut],
Wenguang Sun [aut],
Changliang Zou [aut]

Maintainer Lilun Du <lilundu@cityu.edu.hk>

Description We develop a new class of distribution free multiple testing rules for false discovery rate (FDR) control under general dependence. A key element in our proposal is a symmetrized data aggregation (SDA) approach to incorporating the dependence structure via sample splitting, data screening and information pooling. The proposed SDA filter first constructs a sequence of ranking statistics that fulfill global symmetry properties, and then chooses a data driven threshold along the ranking to control the FDR. For more information, see the website below and the accompanying paper: Du et al. (2023), ``False Discovery Rate Control Under General Dependence By Symmetrized Data Aggregation", <doi:10.1080/01621459.2021.1945459>. Some optional functionality uses the archived R packages 'huge' and 'pfa', which are not available from CRAN's main repositories. Users who need this optional functionality can obtain them from the CRAN Archive as follows: 'huge' at <https://cran.r-project.org/src/contrib/Archive/huge/>; 'pfa' at <https://cran.r-project.org/src/contrib/Archive/pfa/>.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

Imports glmnet, glasso, POET, stats, selectiveInference,

Suggests testthat (>= 2.1.0), huge, pfa, MASS

Repository CRAN

NeedsCompilation no

Date/Publication 2026-02-13 16:30:02 UTC

Contents

SDA_2S 2
SDA_M 3

Index **5**

SDA_2S *Symmetrized Data Aggregation for two-sample t-test*

Description

Symmetrized Data Aggregation for two-sample t-test

Usage

```
SDA_2S(dat_I, dat_II, alpha, Sigma_I, Sigma_II, stable = TRUE)
```

Arguments

dat_I	a n_1 by p data matrix, the first part of data
dat_II	a n_2 by p data matrix, the second part of data
alpha	the FDR level
Sigma_I	the covariance matrix of sample 1; if it is missing, it will be estimated by the glasso package.
Sigma_II	the covariance matrix of sample 2; if it is missing, it will be estimated by the glasso package.
stable	If it is TRUE, the sample will be randomly splitted $B = 10$ times for stability performance; otherwise, only single sample splitting is used.

Value

the indices of the hypotheses rejected

Examples

```
p = 100
n = 30
dat_I = matrix(rnorm(n*p),nrow = n)
mu = rep(0, p)
mu[1:10] = 1.5
dat_I = dat_I = rep(1, n)%*%t(mu)

dat_II = matrix(rnorm(n*p), nrow = n)
Sigma_I = diag(p)
Sigma_II = diag(p)
out = SDA_2S(dat_I, dat_II, alpha=0.05, Sigma_I, Sigma_II)
print(out)
```

SDA_M *Symmetrized Data Aggregation for one-sample t-test*

Description

This is the main function in the SDA paper. Other commonly used test statistics for the first part of data are also allowed in this function.

Usage

```
SDA_M(
  dat,
  alpha,
  Omega,
  nonsparse = FALSE,
  stable = TRUE,
  kwd = c("lasso", "de-lasso", "innovate", "pfa"),
  scale = TRUE
)
```

Arguments

dat	a n by p data matrix
alpha	the FDR level
Omega	the inverse covariance matrix; if it is missing, it will be estimated by the glasso package.
nonsparse	If it is TRUE, the covariance matrix will be estimated by the POET package; otherwise it will be fitted by glasso by default.
stable	If it is TRUE, the sample will be randomly splitted $B = 10$ times for stability performance; otherwise, only single sample-splitting is used.
kwd	various methods for calculating the test statistics from the first part of data
scale	If it is TRUE, the test statistic from the first part of data will be standardized.

Details

We provide other commonly used test statistics for the first part of sample. These include the de-biased lasso, innovated transformation, and factor-adjusted test statistics.

Value

the indices of the hypotheses rejected by the SDA method

Examples

```
n = 50
p = 100
rho = 0.8
Sig = matrix(rho, p, p)
diag(Sig) = 1
dat <- MASS::mvrnorm(n, rep(0, p), Sig)
mu = rep(0, p)
mu[1:as.integer(0.1*p)]=0.5
dat = dat+rep(1, n)%*%t(mu)
alpha = 0.2
out = SDA_M(dat, alpha, solve(Sig), kwd='lasso')
print(out)
```

Index

SDA_2S, 2
SDA_M, 3