

Package: sdafilter (via r-universe)

September 6, 2024

Title Symmetrized Data Aggregation

Version 1.0.0

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

Maintainer Lilun Du <dulilun@ust.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. (2020), "False Discovery Rate Control Under General Dependence By Symmetrized Data Aggregation", <[arXiv:2002.11992](https://arxiv.org/abs/2002.11992)>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Imports glmnet, glasso, huge, POET, stats

Suggests testthat (>= 2.1.0)

Repository CRAN

NeedsCompilation no

Date/Publication 2020-03-19 13:30:05 UTC

Contents

SDA_M	2
Index	3

`SDA_M`*Symmetrized Data Aggregation*

Description

This is the core function for the paper posted in arXiv preprint arXiv:2002.11992

Usage

```
SDA_M(dat, alpha, Omega, nonsparse = FALSE, stable = TRUE)
```

Arguments

<code>dat</code>	a n by p data matrix
<code>alpha</code>	the FDR level
<code>Omega</code>	the inverse covariance matrix; if missing, it will be estimated by the glasso package
<code>nonsparse</code>	if TRUE, the covariance matrix will be estimated by the POET package
<code>stable</code>	if 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

```
n = 50
p = 100
dat = matrix(rnorm(n*p), nrow=n)
mu = rep(0, p)
mu[1:as.integer(0.1*p)] = 0.3
dat = dat + rep(1, n) %*% t(mu)
alpha = 0.2
out = SDA_M(dat, alpha, diag(p))
print(out)
```

Index

SDA_M, [2](#)