Package ‘HIMA’

May 15, 2021

Type Package

Title High-Dimensional Mediation Analysis

Version 1.1.0

Date 2021-05-12


License GPL-3

Depends R (>= 3.4.0), ncvreg, glmnet

Imports utils, stats, MASS, survival, HDMT, iterators, parallel, foreach, doParallel


Encoding UTF-8

LazyData false

URL https://github.com/YinanZheng/HIMA/

BugReports https://github.com/YinanZheng/HIMA/issues/

RoxygenNote 7.1.1

NeedsCompilation no

Author Yinan Zheng [aut, cre], Haixiang Zhang [aut], Lifang Hou [aut], Lei Liu [aut, cph]

Maintainer Yinan Zheng <y-zheng@northwestern.edu>

Repository CRAN

Date/Publication 2021-05-15 05:50:03 UTC
R topics documented:

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIMA-package</td>
<td>High-dimensional Mediation Analysis</td>
</tr>
<tr>
<td>hima</td>
<td>High-dimensional Mediation Analysis</td>
</tr>
<tr>
<td>simHIMA</td>
<td>High-dimensional Mediation Analysis</td>
</tr>
<tr>
<td>survHIMA</td>
<td>High-dimensional Mediation Analysis</td>
</tr>
</tbody>
</table>

Index

**HIMA-package**

High-dimensional Mediation Analysis

Description

HIMA is an R package for estimating and testing high-dimensional mediation effects in omic studies.

<table>
<thead>
<tr>
<th>Package</th>
<th>Type</th>
<th>Version</th>
<th>Date</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIMA</td>
<td>Package</td>
<td>1.1.0</td>
<td>2021-05-12</td>
<td>GPL-3</td>
</tr>
</tbody>
</table>

Author(s)

Yinan Zheng <y-zheng@northwestern.edu>, Haixiang Zhang <haixiang.zhang@tju.edu.cn>, Lifang Hou <l-hou@northwestern.edu> Lei liu <lei.liu@wustl.edu>

Maintainer: Yinan Zheng <y-zheng@northwestern.edu>

References


**hima**

High-dimensional Mediation Analysis

Description

hima is used to estimate and test high-dimensional mediation effects.
Usage

```r
hima(
    X,
    Y,
    M,
    COV.XM = NULL,
    COV.MY = COV.XM,
    family = c("gaussian", "binomial"),
    penalty = c("MCP", "SCAD", "lasso"),
    topN = NULL,
    parallel = FALSE,
    ncore = 1,
    verbose = FALSE,
    ...
)
```

Arguments

- **X**: a vector of exposure.
- **Y**: a vector of outcome. Can be either continuous or binary (0-1).
- **M**: a `data.frame` or `matrix` of high-dimensional mediators. Rows represent samples, columns represent variables.
- **COV.XM**: a `data.frame` or `matrix` of covariates dataset for testing the association \( M \sim X \). Covariates specified here will not participate penalization. Default = `NULL`. If the covariates contain mixed data types, please make sure all categorical variables are properly formatted as factor type.
- **COV.MY**: a `data.frame` or `matrix` of covariates dataset for testing the association \( Y \sim M \). Covariates specified here will not participate penalization. If not specified, the same set of covariates for \( M \sim X \) will be applied. Using different sets of covariates is allowed but this needs to be handled carefully.
- **family**: either 'gaussian' or 'binomial', depending on the data type of outcome (Y). See `ncvreg`.
- **penalty**: the penalty to be applied to the model. Either 'MCP' (the default), 'SCAD', or 'lasso'. See `ncvreg`.
- **topN**: an integer specifying the number of top markers from sure independent screening. Default = `NULL`. If `NULL`, `topN` will be either \( \text{ceiling}(n/\log(n)) \) if `family` = 'gaussian', or \( \text{ceiling}(n/(2*\log(n))) \) if `family` = 'binomial', where \( n \) is the sample size. If the sample size is greater than `topN` (pre-specified or calculated), all mediators will be included in the test (i.e. low-dimensional scenario).
- **ncore**: number of cores to run parallel computing. Valid when `parallel == TRUE`. By default max number of cores available in the machine will be utilized.
- **verbose**: logical. Should the function be verbose? Default = `FALSE`.
- **...**: other arguments passed to `ncvreg`. 
Value

A data.frame containing mediation testing results of selected mediators.

- alpha: coefficient estimates of exposure (X) --> mediators (M).
- beta: coefficient estimates of mediators (M) --> outcome (Y) (adjusted for exposure).
- gamma: coefficient estimates of exposure (X) --> outcome (Y) (total effect).
- alpha*beta: mediation effect.
- % total effect: alpha*beta / gamma. Percentage of the mediation effect out of the total effect.
- Bonferroni.p: statistical significance of the mediator (Bonferroni procedure).
- BH.FDR: statistical significance of the mediator (Benjamini-Hochberg procedure).

References


Examples

```r
n <- 200 # sample size
p <- 200 # the dimension of covariates

# the regression coefficients alpha (exposure --> mediators)
alpha <- rep(0, p)

# the regression coefficients beta (mediators --> outcome)
beta1 <- rep(0, p) # for continuous outcome
beta2 <- rep(0, p) # for binary outcome

# the first four markers are true mediators
alpha[1:4] <- c(0.45, 0.5, 0.6, 0.7)
beta1[1:4] <- c(0.55, 0.6, 0.65, 0.7)
beta2[1:4] <- c(1.45, 1.5, 1.55, 1.6)

# these are not true mediators
alpha[7:8] <- 0.5
beta1[5:6] <- 0.8
beta2[5:6] <- 1.7

# Generate simulation data
simdat_cont = simHIMA(n, p, alpha, beta1, seed=1029)
simdat_bin = simHIMA(n, p, alpha, beta2, binaryOutcome = TRUE, seed=1029)

# Run HIMA with MCP penalty by default
# When Y is continuous (default)
hima.fit <- hima(simdat_cont$X, simdat_cont$Y, simdat_cont$M, verbose = TRUE)
hima.fit
```
# When Y is binary (should specify family)
hima.logistic.fit <- hima(simdat_bin$X, simdat_bin$Y, simdat_bin$M, 
family = "binomial", verbose = TRUE)
hima.logistic.fit

---

**simHIMA**

*Simulation Data Generator for High-dimensional Mediation Analysis*

## Description

simHIMA is used to generate simulation data for high-dimensional mediation analysis.

## Usage

```r
simHIMA(n, p, alpha, beta, binaryOutcome = FALSE, seed)
```

## Arguments

- **n**: an integer specifying sample size.
- **p**: an integer specifying the dimension of mediators.
- **alpha**: a numeric vector specifying the regression coefficients alpha (exposure --> mediators).
- **beta**: a numeric vector specifying the regression coefficients beta (mediators --> outcome).
- **binaryOutcome**: logical. Should the simulated outcome variable be binary?
- **seed**: an integer specifying a seed for random number generation.

## See Also

see [hima](#) to run HIMA.

## Examples

```r
n <- 200  # sample size
p <- 200  # the dimension of covariates

# the regression coefficients alpha (exposure --> mediators)
alpha <- rep(0, p)

# the regression coefficients beta (mediators --> outcome)
beta <- rep(0, p)

# the first four markers are true mediators.
alpha[1:4] <- c(0.45, 0.5, 0.55, 0.6)
beta[1:4] <- c(0.5, 0.45, 0.4, 0.35)

alpha[7:8] <- 0.5
```
# Generate simulation data
simdat = simHIMA(n, p, alpha, beta, seed=1029)

**survHIMA**

*High-dimensional mediation analysis for survival data*

**Description**

`survHIMA` is used to estimate and test high-dimensional mediation effects for survival data.

**Usage**

```r
survHIMA(X, Z, M, OT, status, FDRcut = 0.05, verbose = FALSE)
```

**Arguments**

- **X**: a vector of exposure.
- **Z**: a matrix of adjusting covariates. Rows represent samples, columns represent variables. Can be NULL.
- **M**: a data frame or matrix of high-dimensional mediators. Rows represent samples, columns represent variables.
- **OT**: a vector of observed failure times.
- **status**: a vector of censoring indicator (status = 1: uncensored; status = 0: censored).
- **FDRcut**: FDR cutoff applied to define and select significant mediators. Default = 0.05.
- **verbose**: logical. Should the function be verbose? Default = FALSE.

**Value**

A data.frame containing mediation testing results of selected mediators (FDR <0.05).

- **ID**: index of selected significant mediator.
- **alpha**: coefficient estimates of exposure (X) -> mediators (M).
- **alpha_se**: standard error for alpha.
- **beta**: coefficient estimates of mediators (M) -> outcome (Y) (adjusted for exposure).
- **beta_se**: standard error for beta.
- **p.joint**: joint p-value of selected significant mediator.

**References**

Examples

```r
## Generate simulated survival data
set.seed(100)
n <- 100  # sample size
p <- 100  # the dimension of mediators
q <- 1    # the dimension of covariate(s)

sigma_e <- matrix(0.25, p, p)
diag(sigma_e) <- 1
sigma_e[1, 3] <- 0.8
sigma_e[3, 1] <- 0.8
sigma_e[2, 4] <- 0.8
sigma_e[4, 2] <- 0.8

##
beta <- matrix(0, 1, p)
beta[1:5] <- c(0.6, -0.5, 0.4, -0.3, 0.25)

##
alpha <- matrix(0, 1, p)
alpha[1:5] <- c(0.6, -0.5, 0.4, -0.3, 0.25)

##
gamma <- matrix(0.5, 1, q)
etta <- matrix(0.3, p, q)
r <- matrix(0.5, 1, 1)

##
X <- matrix(rnorm(n, mean = 0, sd = 2), n, 1)  # exposure
Z <- matrix(rnorm(n * q, mean = 0, sd = 2), n, q)  # covariates
mu <- matrix(0, p, 1)
e <- MASS::mvrnorm(n, mu, sigma_e)  # the error terms

M <- X%*%(alpha) + Z%*%t(eta) + e
MZ <- cbind(M, Z, X)

beta_gamma <- cbind(beta, gamma, r)

## generate the failure time T
u <- runif(n, 0, 1)
T <- matrix(0, n, 1)
for (i in 1:n)
  T[i] <- -log(1 - u[i]) * exp(-sum(beta_gamma*MZ[i,]))

## generate censoring time 0.45 censoring rate
C <- runif(n, min = 0, max = 150)
status <- as.integer(T < C)

## the observed failure time
OT <- apply(cbind(C, T), 1, min)

## Not run:
```
survHIMA.fit <- survHIMA(X, Z, M, OT, status)
survHIMA.fit

## End(Not run)
Index

* package
  HIMA-package, 2

HIMA (HIMA-package), 2
hima, 2, 5
HIMA-package, 2

ncvreg, 3

simHIMA, 5
survHIMA, 6