Package ‘HIMA’

February 3, 2022

Type Package

Title High-Dimensional Mediation Analysis

Version 2.0.0

Date 2022-02-02


License GPL-3

Depends R (>= 3.4.0), ncvreg, glmnet

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


Encoding UTF-8

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

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

RoxygenNote 7.1.2

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 2022-02-03 07:40:17 UTC
\textbf{R topics documented:}

<table>
<thead>
<tr>
<th>HIMA-package</th>
<th>hima</th>
<th>microHIMA</th>
<th>simHIMA</th>
<th>survHIMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

\textbf{Index} 11

\begin{tabular}{ll}
\textbf{HIMA-package} & \textit{High-Dimensional Mediation Analysis for 'Omic' Data} \\
\end{tabular}

\textbf{Description}

HIMA is an R package for estimating and testing high-dimensional mediation effects in omic studies. HIMA can perform high-dimensional mediation analysis on a wide range of omic data types as potential mediators, including epigenetics, transcriptomics, proteomics, and metabolomics using function \texttt{hima} and microbiome data (function \texttt{microHIMA}). HIMA can also handle survival data (function \texttt{survHIMA}).

\begin{center}
\begin{tabular}{lrr}
Package: & HIMA & \\
Type: & Package & \\
Version: & 2.0.0 & \\
Date: & 2022-02-02 & \\
License: & GPL-3 & \\
\end{tabular}
\end{center}

\textbf{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>

\textbf{References}


**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"),
  screen.family = c("gaussian", "negbin"),
  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`
- **screen.family**
  either 'gaussian' (default) or 'negbin' (i.e., negative binomial). This parameter is taking effect only when parameter family = 'binomial'. When family = 'binomial', the screening step is based on X (exposure, independent variable) and the omic mediator (dependent variable). This is useful when handling RNA sequencing count data as mediators.
penalty an integer specifying the number of top markers from sure independent screening. Default = NULL. If NULL, topN will be either ceiling(n/log(n)) if family = 'gaussian', or 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).

parallel logical. Enable parallel computing feature? Default = TRUE.

cmp logical. Should the function be verbose? Default = FALSE.

Value

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

• alpha: coefficient estimates of exposure (X) \rightarrow mediators (M).
• beta: coefficient estimates of mediators (M) \rightarrow outcome (Y) (adjusted for exposure).
• gamma: coefficient estimates of exposure (X) \rightarrow 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

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

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

# the regression coefficients beta (mediators \rightarrow 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

---

microHIMA

High-dimensional mediation analysis for compositional microbiome data

Description

microHIMA is used to estimate and test high-dimensional mediation effects for compositional microbiome data.

Usage

microHIMA(X, Y, OTU, COV = NULL, FDPcut = 0.05)

Arguments

X a vector of exposure.
Y a vector of outcome.
OTU a data.frame or matrix of high-dimensional compositional OTUs (mediators). Rows represent samples, columns represent variables.
COV a data.frame or matrix of adjusting covariates. Rows represent samples, columns represent microbiome variables. Can be NULL.
FDPcut FDP (false discovery proportions) cutoff applied to define and select significant mediators. Default = 0.05.
Value

A data.frame containing mediation testing results of selected mediators (FDP < FDPcut).

- **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_FDP**: false discovery proportions of selected significant mediator.

References


Examples

```r
## Generate simulated survival data
n <- 200 # Sample size
p <- 25 # Number of microbiome
Treatment = rbinom(n, 1, 0.2) # binary outcome

## Generate two covariates, one binary, one continuous
covariates = cbind(sample(c(1,0), n, replace = TRUE), rnorm(n))

## parameters
beta0 = as.numeric(matrix(0, 1, p))
betaT = rep(0, p)
betaT[c(1, 2, 3)] = c(1, 1.2, 1.5) # let the first three are non-zero
betaX = matrix(0, p, 2)
alpha0 = 0
alphaT = 1
alphaZ = alphaC = rep(0, p)
alphaZ[c(1, 2, 3)] = c(1.3, -0.7, -0.6) # let the first three are non-zero for response
alphaX = c(0.5, 0.5)

## Generate microbiome data
X = cbind(rep(1, n), covariates, Treatment) # n * (1 + q + p)
b = cbind(beta0, betaX, betaT) # p * (1 + q + p)
gamma.simu = exp(X %*% t(b)) # n * p
otu.com = t(apply(gamma.simu, 1, HIMA:::rdirichlet, n = 1)) # Dirichlet distribution

## Generate outcome data
X = cbind(rep(1, n), Treatment, covariates, log(otu.com), log(otu.com) * Treatment)
```
b = c(alpha0, alphaT, alphaX, alphaZ, alphaC)
outcome = c(b %*% t(X) + rnorm(n, mean = 0, sd = 1))
exposure = t(t(Treatment))

## Not run:
microHIMA.fit <- microHIMA(X = exposure, Y = outcome, OTU = otu.com, COV = covariates)
microHIMA.fit

## End(Not run)

# simHIMA

Simulates data for high-dimensional mediation analysis.

## Usage

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
beta[5:6] <- 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 mediator 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 < FDPcut).

- **ID**: index of selected significant mediator.
- **alpha**: coefficient estimates of exposure (X) \rightarrow mediators (M).
- **alpha_se**: standard error for alpha.
- **beta**: coefficient estimates of mediators (M) \rightarrow 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 <- 300 # 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)
## eta <- 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%*%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, 3, 7
HIMA-package, 2

microHIMA, 5
ncvreg, 3, 4
simHIMA, 7
survHIMA, 8