Package: TransHDM (via r-universe)

October 23, 2026

Title High-Dimensional Mediation Analysis via Transfer Learning

Version 1.1.3

Description Provides a framework for high-dimensional mediation analysis using transfer learning. The main function TransHDM() integrates large-scale source data to improve the detection power of potential mediators in small-sample target studies. It addresses data heterogeneity via transfer regularization and debiased estimation while controlling the false discovery rate. The package also includes utilities for data generation (gen_simData_homo(), gen_simData_hetero()), baseline methods such as lasso() and dblasso(), sure independence screening via SIS(), and model diagnostics through source_detection(). The methodology is described in Pan et al. (2025) <doi:10.1093/bib/bbaf460>.

License GPL (>= 3)

URL https://github.com/Gaohuer/TransHDM

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.0.0)

Imports glmnet (>= 4.1-10), caret (>= 7.0-1), MASS (>= 7.3-61), doParallel (>= 1.0.17), foreach (>= 1.5.2), HDMT (>= 1.0.5), ggplot2 (>= 3.4.0)

Suggests knitr (>= 1.50), rmarkdown (>= 2.30), spelling (>= 2.3.2), dplyr (>= 1.0.0), tidyr (>= 1.0.0), ggrepel (>= 0.9.0)

Language en-US

VignetteBuilder knitr

LazyData true

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libicu-dev

Repository https://gaohuer.r-universe.dev

Date/Publication 2026-08-24 08:11:34 UTC

RemoteUrl https://github.com/gaohuer/transhdm

RemoteRef HEAD

RemoteSha b6fabb41c5e0a1bead03505f7b2520f10240f9de

Contents

gen_simData_hetero . . . 2
gen_simData_homo . . . 4
inflam_effect . . . 5
inflam_external1 . . . 6
inflam_external2 . . . 7
inflam_target . . . 8
joint_test . . . 9
mediation_inference . . . 10
plot.source_detection . . . 12
plot.TransHDM . . . 14
print.dblasso . . . 16
print.lasso . . . 16
print.mediation_inference . . . 17
print.SIS . . . 17
print.source_detection . . . 18
print.summary.dblasso . . . 18
print.TransHDM . . . 19
SIS . . . 19
source_detection . . . 21
summary.dblasso . . . 23
summary.lasso . . . 24
summary.mediation_inference . . . 25
summary.source_detection . . . 25
summary.TransHDM . . . 26
TransHDM . . . 26

Index **30**

`gen_simData_hetero` *Simulated Dataset Generation for High-Dimensional Mediation Analysis*

Description

Generates synthetic datasets mimicking high-dimensional mediation structures, optionally incorporating transferable source data under varying covariate correlation and heterogeneity levels. This function supports heterogeneous settings for data generation.

Usage

```
gen_simData_hetero(
  n = 100,
  p_x = 5,
  rho = 0,
  p_m = 100,
  h = 0,
  source = FALSE,
  transferable = TRUE,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations (sample size). Default is 100.
p_x	Integer. Number of covariates (confounders). Default is 5.
rho	Numeric. Correlation coefficient (0–1) controlling correlation between mediators. Default is 0 (no correlation).
p_m	Integer. Number of mediators. Default is 100.
h	Integer. Degree of heterogeneity (for source data). Default is 0.
source	Logical. If TRUE, generate source (external) dataset. Default is FALSE.
transferable	Logical. If TRUE, generates a transferable source dataset sharing mediator-outcome structure with the target. Default is TRUE.
seed	Integer. Optional random seed for reproducibility. Default is NULL.

Details

This function generates data according to a structural equation model (SEM):

- Treatment D depends linearly on covariates X.
- Mediators M depend on D and X, with residual correlation controlled by rho.
- Outcome Y depends on D, M, and X.
- If source = TRUE, then a source dataset is simulated with potentially transferable mechanisms.

Value

A list with the following components:

- data: A data.frame of dimension n × (2 + p_m + p_x) containing outcome (Y), treatment (D), mediators (M1–Mp_m), and covariates (X1–Xp_x).
- effect: A named list of true total, direct, and indirect effects, exposure-mediator effects and mediator-outcome effects.

Examples

```
source_data <- gen_simData_hetero(
  n = 100, p_x = 5, rho = 0, p_m = 100, h = 0,
  source = TRUE, transferable = TRUE, seed = 123
)
source_data <- gen_simData_hetero(
  n = 100, p_x = 5, rho = 0, p_m = 100, h = 0,
  source = FALSE, transferable = TRUE, seed = 123
)
```

---

gen_simData_homo *Simulated Dataset Generation for High-Dimensional Mediation Analysis*

Description

Generates synthetic datasets mimicking high-dimensional mediation structures, optionally incorporating transferable source data under varying covariate correlation and heterogeneity levels. This function supports homogeneous settings for data generation.

Usage

```
gen_simData_homo(
  n = 100,
  p_x = 5,
  rho = 0,
  p_m = 100,
  h = 0,
  source = FALSE,
  transferable = TRUE,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations (sample size). Default is 100.
p_x	Integer. Number of covariates (confounders). Default is 5.
rho	Numeric. Correlation coefficient (0–1) controlling correlation between mediators. Default is 0 (no correlation).
p_m	Integer. Number of mediators. Default is 100.
h	Integer. Degree of heterogeneity (for source data). Default is 0.
source	Logical. If TRUE, generate source (external) dataset. Default is FALSE.
transferable	Logical. If TRUE, generates a transferable source dataset sharing mediator-outcome structure with the target. Default is TRUE.
seed	Integer. Optional random seed for reproducibility. Default is NULL.

Details

This function generates data according to a structural equation model (SEM):

- Treatment D depends linearly on covariates X.
- Mediators M depend on D and X, with residual correlation controlled by `rho`.
- Outcome Y depends on D, M, and X.
- If `source = TRUE`, then a source dataset is simulated with potentially transferable mechanisms.

Value

A list with the following components:

- `data`: A data.frame of dimension `n * (2 + p_m + p_x)` containing outcome (Y), treatment (D), mediators (M1–Mp_m), and covariates (X1–Xp_x).
- `effect`: A named list of true total, direct, and indirect effects, exposure-mediator effects and mediator-outcome effects.

Examples

```
source_data <- gen_simData_homo(
  n = 100, p_x = 5, rho = 0, p_m = 100, h = 0,
  source = TRUE, transferable = TRUE, seed = 123
)
target_data <- gen_simData_homo(
  n = 100, p_x = 5, rho = 0, p_m = 100, h = 0,
  source = FALSE, transferable = TRUE, seed = 123
)
```

`inflam_effect` *Inflammatory Mediation Benchmark Data: Ground Truth Effects*

Description

Ground-truth mediation effects for the target cohort of the inflammatory mediation benchmark dataset. Maps each candidate mediator to its true exposure–mediator effect (α), mediator–outcome effect (β), and indirect effect ($\alpha \times \beta$).

Usage

```
data(inflam_effect)
```

Format

A data frame with 20 rows (one per mediator) and 4 columns:

- `mediator`: mediator name.
- `alpha`: true exposure–mediator effect.
- `beta`: true mediator–outcome effect.
- `alpha_beta`: true indirect (mediation) effect.

Details

Four mediators have non-zero $\alpha \times \beta$ and are the true active mediators. Because the mediator columns were randomly permuted during data generation, these active mediators are scattered across the candidate names — as in a real study, where the ground truth is unknown at analysis time. This table lets users benchmark `TransHDM` against the truth.

See Also

`inflam_target`, `inflam_external1`, `inflam_external2` for the associated cohorts.

Examples

```
data(inflam_effect)
inflam_effect[inflam_effect$alpha_beta != 0, ]
```

---

inflam_external1 *Inflammatory Mediation Benchmark Data: Transferable External Cohort*

Description

Simulated **transferable** external cohort for high-dimensional mediation analysis. It shares the mediator–outcome structure with the target cohort, making it suitable for transfer learning.

Usage

```
data(inflam_external1)
```

Format

A data frame with 200 rows (patients) and 25 columns, with the same structure as `inflam_target`:

- `DSI`: outcome — disease severity index.
- `IB`: exposure — inflammatory biomarker.
- `CYT_01`–`SIG_04`: 20 candidate mediators.
- `BIS`, `PRS`, `BRS`: 3 covariates.

Details

This large external cohort ($n = 200$) is generated under a transferable setting: its exposure–mediator (α) and mediator–outcome (β) coefficients match those of the target for the active mediators. It is intended as the source in `TransHDM` with `transfer = TRUE`.

Note

Demonstration data only. Simulated for methodological demonstration of the `TransHDM` workflow.

Source

Generated with gen_simData_homo (n = 200, p_x = 3, p_m = 20, rho = 0.1, seed = 123, source = TRUE, transferable = TRUE).

See Also

inflam_target for the target cohort, inflam_external2 for the non-transferable external cohort.

Examples

```
data(inflam_external1)
str(inflam_external1)
```

---

inflam_external2 *Inflammatory Mediation Benchmark Data: Non-Transferable External Cohort*

Description

Simulated **non-transferable** external cohort for high-dimensional mediation analysis. Its mediator structure differs from the target cohort, and the covariate distribution is shifted, so it should not be used for transfer.

Usage

```
data(inflam_external2)
```

Format

A data frame with 100 rows (patients) and 25 columns, with the same structure as inflam_target:

- DSI: outcome — disease severity index.
- IB: exposure — inflammatory biomarker.
- CYT_01–SIG_04: 20 candidate mediators.
- BIS, PRS, BRS: 3 covariates.

Details

This external cohort (n = 100) is generated under a non-transferable setting with covariate shift. Its exposure–mediator and mediator–outcome coefficients differ from the target, so naive transfer would degrade performance. It is used with source_detection to demonstrate detection of non-transferable sources.

Note

Demonstration data only. Simulated for methodological demonstration of the TransHDM workflow.

Source

Generated with gen_simData_hetero (n = 100, p_x = 3, p_m = 20, rho = 0.1, seed = 999, source = TRUE, transferable = FALSE).

See Also

inflam_target for the target cohort, inflam_external1 for the transferable external cohort.

Examples

```
data(inflam_external2)
str(inflam_external2)
```

---

inflam_target *Inflammatory Mediation Benchmark Data: Target Cohort*

Description

Simulated **target** cohort for a high-dimensional mediation analysis in which an inflammatory biomarker (exposure) influences a disease severity index (outcome) through a panel of candidate molecular mediators, adjusting for clinical covariates.

Usage

```
data(inflam_target)
```

Format

A data frame with 50 rows (patients) and 25 columns:

- DSI: outcome — disease severity index.
- IB: exposure — inflammatory biomarker.
- CYT_01–SIG_04: 20 candidate mediators across five functional groups (cytokines CYT, immune markers IMM, oxidative stress OXI, metabolic MET, signalling SIG).
- BIS, PRS, BRS: 3 covariates (baseline inflammatory status, physiological reserve score, baseline risk score).

Details

Mediation model:

$$IB \rightarrow Molecular\ Mediators \rightarrow DSI$$

This is the small-sample target cohort (n = 50) that benefits from transfer learning using the larger external cohorts. The mediator columns have been randomly permuted, so the identity of the truly active mediators is hidden; the ground truth is stored in inflam_effect.

Note

Demonstration data only. Simulated for methodological demonstration of the TransHDM workflow, not for drawing biological conclusions.

Source

Generated with gen_simData_homo (n = 50, p_x = 3, p_m = 20, rho = 0.1, seed = 123). See gen_data.R in the package source.

See Also

inflam_external1, inflam_external2 for the external cohorts, and inflam_effect for the ground truth.

Examples

```
data(inflam_target)
str(inflam_target)
```

---

joint_test *Joint Multiple Testing for Mediation Analysis*

Description

Step 3 (final step) of the TransHDM pipeline: joint multiple testing on the alpha and beta p-values obtained from mediation_inference (Step 2). It estimates the null proportion, controls the false discovery rate via HDMT::fdr_est, identifies significant mediators, and summarises the direct, indirect, and total effects into a "TransHDM" object.

Together with SIS (Step 1) and mediation_inference (Step 2), this completes the three-step TransHDM workflow.

Usage

```
joint_test(inference_result, p_cutoff = 0.05, verbose = TRUE)
```

Arguments

inference_result	An object of class "mediation_inference" returned by mediation_inference.
p_cutoff	A numeric value (default: 0.05) specifying the maximum FDR threshold for identifying significant mediators.
verbose	A logical value (default: TRUE) controlling whether progress messages are printed.

Value

An object of class "TransHDM" — a list with:

- contributions: Data frame of identified mediators with columns mediator, alpha, alpha_pv, beta, beta_pv, alpha_beta, ab_pv, pa.
- effects: Data frame summarising indirect, direct, total effects and proportion mediated (pe).
- IDE_est: Named numeric vector of indirect effect estimates for all mediators (non-selected set to zero).
- DE_est: Estimated direct effect.
- all_mediators: Data frame of all SIS-screened mediators with columns mediator, alpha, alpha_pv, beta, beta_pv, significant (logical).

See Also

TransHDM, mediation_inference, SIS

Examples

```
set.seed(123)
target_data <- gen_simData_homo(n = 50, p_x = 3, p_m = 20, rho = 0.1)$data
source_data <- gen_simData_homo(n = 100, p_x = 3, p_m = 20, rho = 0.1,
  source = TRUE, transferable = TRUE)$data
M_vars <- paste0("M", 1:20)
X_vars <- paste0("X", 1:3)

# Full pipeline
sis <- SIS(target_data, source_data, Y = "Y", D = "D", M = M_vars,
  X = X_vars, transfer = TRUE, topN = 10)
infer <- mediation_inference(sis, M = M_vars, D = "D", X = X_vars,
  Y = "Y", transfer = TRUE)
result <- joint_test(infer)
plot(result)
```

---

mediation_inference *Mediation Inference via De-biased Lasso*

Description

Step 2 of the TransHDM pipeline: de-biased lasso estimation following screening (SIS, Step 1). It consists of two sub-steps:

- **Step 2.1 — Outcome model (beta):** estimates mediator-outcome effects by regressing the outcome Y on all retained mediators, the exposure D, and covariates X.
- **Step 2.2 — Mediator models (alpha):** estimates exposure-mediator effects by regressing each mediator on D and X.

Both steps support transfer learning. Results are passed to joint_test (Step 3) for multiple testing.

Usage

```
mediation_inference(
  screen_result = NULL,
  target_data = NULL,
  source_data = NULL,
  Y = NULL,
  D = NULL,
  M = NULL,
  X = NULL,
  transfer = FALSE,
  verbose = TRUE,
  ncore = 1
)
```

Arguments

screen_result	An object of class "SIS" returned by SIS. When provided, Y, D, M, and X are automatically extracted from it. If NULL, target_data (and optionally source_data) are used directly without screening, and Y, D, M, X must be supplied explicitly.
target_data	A data frame containing the target dataset. Required when screen_result = NULL. Ignored otherwise.
source_data	A data frame containing the source dataset (optional, default: NULL). Used only when transfer = TRUE and screen_result = NULL.
Y	Character string specifying the outcome variable name. Required only when screen_result = NULL; otherwise extracted from screen_result.
D	Character string specifying the exposure variable name. Required only when screen_result = NULL.
M	Character vector specifying the full set of mediator variable names (before screening). Required only when screen_result = NULL; otherwise extracted from screen_result.
X	Character vector specifying covariate variable names. Required only when screen_result = NULL.
transfer	A logical value (default: FALSE) indicating whether to enable transfer learning.
verbose	A logical value (default: TRUE) controlling whether progress messages are printed.
ncore	An integer (default: 1) specifying the number of CPU cores for parallel computation of mediator models.

Value

An object of class "mediation_inference" — a list with:

- DE_est: Estimated direct effect of D on Y.
- alpha_SIS_est: Named numeric vector of alpha estimates (exposure-mediator effects) for all mediators in M. Non-selected mediators are set to zero.
- P_alpha_SIS: Named numeric vector of p-values for alpha.

- `beta_SIS_est`: Named numeric vector of beta estimates (mediator-outcome effects) for all mediators in `M`. Non-selected mediators are set to zero.
- `P_beta_SIS`: Named numeric vector of p-values for beta.
- `M`: The full mediator name vector.
- `M_SIS`: Mediators retained after screening.
- `D`: The exposure variable name.

See Also

`TransHDM`, `joint_test`, `SIS`

Examples

```
set.seed(123)
target_data <- gen_simData_homo(n = 50, p_x = 3, p_m = 20, rho = 0.1)$data
source_data <- gen_simData_homo(n = 100, p_x = 3, p_m = 20, rho = 0.1,
  source = TRUE, transferable = TRUE)$data
M_vars <- paste0("M", 1:20)
X_vars <- paste0("X", 1:3)

# With screening (SIS)
sis <- SIS(target_data, source_data, Y = "Y", D = "D", M = M_vars,
  X = X_vars, transfer = TRUE, topN = 10)
infer <- mediation_inference(sis, transfer = TRUE)

# Without screening (use all mediators directly)
infer_ns <- mediation_inference(
  target_data = target_data, source_data = source_data,
  Y = "Y", D = "D", M = M_vars, X = X_vars,
  transfer = TRUE)
```

`plot.source_detection` *Visualize Source Detection Results*

Description

S3 plot method for objects of class `"source_detection"`. Produces a bar chart of the T-index for each source dataset, with bars extending from a solid baseline at zero and a horizontal reference line at the transferability threshold. Sources whose T-index falls on or below the threshold are considered transferable.

Usage

```
## S3 method for class 'source_detection'
plot(
  x,
  fill_transferable = "white",
  fill_nontransferable = "white",
  color = "black",
  linewidth = 0.5,
  size.text = 3,
  show.values = TRUE,
  digits = 3,
  base_size = 12,
  threshold.linetype = "dashed",
  threshold.color = "gray30",
  ...
)
```

Arguments

x	An object of class "source_detection", returned by source_detection.
fill_transferable	Fill color for transferable-source bars. Default is "white".
fill_nontransferable	Fill color for non-transferable-source bars. Default is "white".
color	Bar border color. Default is "black".
linewidth	Bar border line width. Default is 0.5.
size.text	Numeric; size of value labels on bars. Default is 3.5.
show.values	Logical; if TRUE (default), T-index values are displayed on each bar.
digits	Integer; number of decimal places for displayed values. Default is 3.
base_size	Numeric; base font size passed to theme_classic. Default is 12.
threshold.linetype	Linetype for the threshold reference line. Default is "dashed".
threshold.color	Color for the threshold reference line. Default is "gray30".
...	Additional arguments (currently unused).

Value

A ggplot object.

See Also

source_detection

Examples

```
set.seed(123)
target_data <- data.frame(
  Y  = rnorm(200), D  = rnorm(200),
  M1 = rnorm(200), M2 = rnorm(200), X1 = rnorm(200)
)
source1 <- data.frame(
  Y  = rnorm(300), D  = rnorm(300),
  M1 = rnorm(300), M2 = rnorm(300), X1 = rnorm(300)
)
source2 <- data.frame(
  Y  = rnorm(250), D  = rnorm(250),
  M1 = rnorm(250), M2 = rnorm(250), X1 = rnorm(250)
)
result <- source_detection(
  target_data = target_data,
  source_data = list(source1, source2),
  Y = "Y", D = "D", M = c("M1", "M2"), X = "X1",
  kfold = 5, verbose = FALSE
)
plot(result)
```

---

plot.TransHDM *Visualize TransHDM Mediation Analysis Results*

Description

S3 plot method for objects of class "TransHDM". Produces a bar chart of the overall effects (indirect, direct, total, proportion mediated), individual mediator effects, or p-values.

Usage

```
## S3 method for class 'TransHDM'
plot(
  x,
  type = c("overall", "mediator", "pvalue", "alpha_beta"),
  fill = "white",
  color = "black",
  linewidth = 0.5,
  size.text = 3,
  show.values = TRUE,
  digits = 3,
  base_size = 12,
  label = c("significant", "all", "none"),
  ...
)
```

Arguments

x	An object of class "TransHDM", returned by TransHDM().
type	Character string specifying the type of plot: "overall" (default) for overall effect estimates, "mediator" for individual mediator contributions ($\hat{\alpha} \times \hat{\beta}$), "pvalue" for joint p-values ($-\log_{10}$(ab_pv)), or "alpha_beta" for a scatter plot of exposure–mediator versus mediator–outcome effects, coloured by significance.
fill	Bar fill color. Default is "white".
color	Bar border color. Default is "black".
linewidth	Bar border line width. Default is 0.5.
size.text	Numeric; size of value labels on bars. Default is 3.5.
show.values	Logical; if TRUE (default), numeric values are displayed on each bar.
digits	Integer; number of decimal places for displayed values. Default is 3.
base_size	Numeric; base font size passed to theme_classic. Default is 12.
label	For type = "alpha_beta" only: "significant" (default) labels only significant mediators, "all" labels every mediator, "none" suppresses labels. Requires the **ggrepel** package.
...	Additional arguments (currently unused).

Value

A ggplot object. The plot is printed automatically when called interactively.

Examples

```
set.seed(123)
target_data <- gen_simData_homo(n = 50, p_x = 3, p_m = 20, rho = 0.1)$data
source_data <- gen_simData_homo(n = 100, p_x = 3, p_m = 20, rho = 0.1,
  source = TRUE, transferable = TRUE)$data
result <- TransHDM(
  target_data = target_data, source_data = source_data,
  Y = "Y", D = "D", M = paste0("M", 1:20), X = paste0("X", 1:3),
  transfer = TRUE, ncore = 1, topN = 10
)
plot(result)
plot(result, type = "mediator")
plot(result, type = "pvalue")
plot(result, type = "alpha_beta")
```

print.dblasso *Print method for dblasso objects*

Description

Print method for dblasso objects

Usage

```
## S3 method for class 'dblasso'
print(x, ...)
```

Arguments

x	An object of class "dblasso".
...	Further arguments (currently not used).

Value

Invisibly returns x.

print.lasso *Print method for lasso objects*

Description

Print method for lasso objects

Usage

```
## S3 method for class 'lasso'
print(x, ...)
```

Arguments

x	An object of class "lasso".
...	Further arguments (currently not used).

Value

Invisibly returns x.

print.mediation_inference *Print method for mediation_inference objects*

Description

Print method for mediation_inference objects

Usage

```
## S3 method for class 'mediation_inference'
print(x, ...)
```

Arguments

x	An object of class "mediation_inference".
...	Further arguments (currently not used).

Value

Invisibly returns x.

print.SIS *Print method for SIS objects*

Description

Print method for SIS objects

Usage

```
## S3 method for class 'SIS'
print(x, ...)
```

Arguments

x	An object of class "SIS".
...	Further arguments (currently not used).

Value

Invisibly returns x.

print.source_detection *Print method for source_detection objects*

Description

Print method for source_detection objects

Usage

```
## S3 method for class 'source_detection'
print(x, ...)
```

Arguments

x	An object of class "source_detection".
...	Further arguments (currently not used).

Value

Invisibly returns x.

print.summary.dblasso *Print method for summary.dblasso objects*

Description

Displays the coefficient table from a debiased Lasso model summary. Supports fixed decimal places and scientific notation formatting.

Usage

```
## S3 method for class 'summary.dblasso'
print(x, digits = 3, fixed = FALSE, ...)
```

Arguments

x	An object of class "summary.dblasso".
digits	Number of digits to display (default is 3).
fixed	Logical. If TRUE, numbers are printed with fixed decimal places (no scientific notation). If FALSE, numbers are printed in scientific notation with the specified digits.
...	Further arguments (currently not used).

Value

Invisibly returns x.

`print.TransHDM` *Print method for TransHDM objects*

Description

Print method for TransHDM objects

Usage

```
## S3 method for class 'TransHDM'
print(x, ...)
```

Arguments

`x`	An object of class `"TransHDM"`.
`...`	Further arguments (currently not used).

Value

Invisibly returns `x`.

`SIS` *Sure Independence Screening for High-Dimensional Mediation Analysis*

Description

Step 1 of the `TransHDM` pipeline: dimension reduction via Sure Independence Screening (SIS). Mediators are ranked based on the product of their marginal associations with the exposure (alpha) and the outcome (beta), and the top-ranked mediators are retained for downstream de-biased lasso estimation (`mediation_inference`, Step 2) and joint testing (`joint_test`, Step 3).

The function supports transfer learning, allowing information from a source dataset to be leveraged to improve screening stability and robustness in the target dataset.

Usage

```
SIS(
  target_data,
  source_data = NULL,
  Y,
  D,
  M,
  X,
  topN = NULL,
  transfer = FALSE,
```

```
  verbose = TRUE,
  ncore = 1,
  dblasso_method = FALSE
)
```

Arguments

target_data	A data frame containing the target dataset. All variables must be numeric.
source_data	A list of data frames containing source datasets (optional, default: NULL). All variables must be numeric and have the same column names as target_data.
Y	Character string specifying the outcome variable name.
D	Character string specifying the exposure (treatment) variable name.
M	Character vector specifying mediator variable names.
X	Character vector specifying covariate variable names.
topN	An integer specifying the number of mediators to retain after screening. If NULL, the number is automatically determined as $\lceil 2n/\log(n) \rceil$, where n is the target sample size.
transfer	A logical value (default: FALSE) indicating whether to apply transfer learning by incorporating the source dataset in the screening procedure.
verbose	A logical value (default: TRUE) controlling whether progress messages are printed to the console.
ncore	An integer (default: 1) specifying the number of CPU cores for parallel computation.
dblasso_method	A logical value (default: FALSE). If TRUE, the debiased lasso (dblasso) is used to estimate marginal effects. If FALSE, standard linear or generalized linear models are used.

Value

A list of class "SIS" with the following components:

- target_SIS: A data frame containing the outcome, exposure, selected mediators, and covariates from the target dataset.
- source_SIS: A data frame containing the same variables from the source dataset if transfer = TRUE; otherwise NULL.
- M_ID_name_SIS: A character vector of selected mediator names.
- Y: The resolved outcome variable name.
- D: The resolved exposure variable name.
- M: The full mediator variable names (before screening).
- X: The resolved covariate variable names.

See Also

TransHDM, mediation_inference, joint_test

Examples

```
set.seed(123)

# Target data
M_target <- matrix(rnorm(200 * 50), nrow = 200)
colnames(M_target) <- paste0("M", 1:50)

target_data <- data.frame(
  Y = rnorm(200),
  D = rnorm(200),
  M_target,
  X1 = rnorm(200)
)

# Source data
M_source <- matrix(rnorm(300 * 50), nrow = 300)
colnames(M_source) <- paste0("M", 1:50)

source_data <- data.frame(
  Y = rnorm(300),
  D = rnorm(300),
  M_source,
  X1 = rnorm(300)
)

# Run SIS
result <- SIS(
  target_data = target_data,
  source_data = source_data,
  Y = "Y",
  D = "D",
  M = paste0("M", 1:50),
  X = "X1",
  transfer = TRUE,
  topN = 10
)

result$M_ID_name_SIS
```

`source_detection` *Detect Transferable Source Data via Cross-Validation*

Description

Determines whether external source datasets can be effectively transferred to the target data by comparing predictive performance using LASSO regression under a transfer learning framework.

Usage

```
source_detection(
  target_data,
  source_data = NULL,
  Y,
  D,
  M,
  X,
  kfold = 5,
  t = 0.02,
  verbose = TRUE
)
```

Arguments

target_data	A data frame containing the target dataset. All variables must be numeric.
source_data	A list of data frames containing source datasets (optional, default: NULL). All variables must be numeric and have the same column names as target_data.
Y	Character string specifying the outcome variable name.
D	Character string specifying the exposure (treatment) variable name.
M	Character vector specifying mediator variable names.
X	Character vector specifying covariate variable names.
kfold	Integer (default: 5). Number of folds for cross-validation.
t	Numeric (default: 0.02). Threshold constant for determining transferability. Larger values make the criterion more lenient.
verbose	Logical (default: TRUE). Whether to print progress messages.

Value

A list containing:

- transfer.source.id: Indices of source datasets deemed transferable
- source.loss: Mean validation loss for each source dataset
- target.valid.loss: Mean validation loss using target-only model
- T_index: Difference between source loss and target-only loss for each source
- threshold: Calculated transferability threshold
- loss.cv: Full k-fold cross-validation loss matrix

Examples

```
## Reproducible example
set.seed(123)

# Generate synthetic target data
target_data <- data.frame(
```

```
  Y  = rnorm(200),
  D  = rnorm(200),
  M1 = rnorm(200),
  M2 = rnorm(200),
  X1 = rnorm(200)
)

# Generate synthetic source data
source1 <- data.frame(
  Y  = rnorm(300),
  D  = rnorm(300),
  M1 = rnorm(300),
  M2 = rnorm(300),
  X1 = rnorm(300)
)

source2 <- data.frame(
  Y  = rnorm(250),
  D  = rnorm(250),
  M1 = rnorm(250),
  M2 = rnorm(250),
  X1 = rnorm(250)
)

# Run source detection
result <- source_detection(
  target_data = target_data,
  source_data = list(source1, source2),
  Y = "Y",
  D = "D",
  M = c("M1", "M2"),
  X = "X1",
  kfold = 5,
  t = 0.05,
  verbose = FALSE
)

# Get Summary
summary(result)

# Transferable source indices
result$transfer.source.id

# Compare validation losses
data.frame(
  Source = c(paste0("Source", seq_along(result$source.loss)), "Target"),
  Loss   = c(result$source.loss, result$target.valid.loss)
)
```

---

summary.dblasso *Summary of Debiased Lasso Inference*

Description

Summary of Debiased Lasso Inference

Usage

```
## S3 method for class 'dblasso'
summary(object, ...)
```

Arguments

object	An object of class "dblasso".
...	Further arguments (currently not used).

Value

An object of class "summary.dblasso".

---

summary.lasso *Summary of Lasso Regression*

Description

Summary of Lasso Regression

Usage

```
## S3 method for class 'lasso'
summary(object, ...)
```

Arguments

object	A numeric vector of lasso coefficients with names.
...	Further arguments (currently not used).

Value

An object of class "summary.lasso".

summary.mediation_inference *Summary of Mediation Inference*

Description

Summary of Mediation Inference

Usage

```
## S3 method for class 'mediation_inference'
summary(object, top = 10, digits = 4, ...)
```

Arguments

object	An object of class "mediation_inference" returned by mediation_inference.
top	Integer, maximum number of mediators to display (default: 10).
digits	Number of digits for rounding estimates (default: 4).
...	Further arguments (unused).

Value

An object of class "summary.mediation_inference".

summary.source_detection *Summary of Source Detection Results*

Description

Summary of Source Detection Results

Usage

```
## S3 method for class 'source_detection'
summary(object, ...)
```

Arguments

object	An object of class "source_detection".
...	Further arguments (unused).

Value

An object of class "summary.source_detection".

summary.TransHDM *Summary of TransHDM Mediation Analysis*

Description

Summary of TransHDM Mediation Analysis

Usage

```
## S3 method for class 'TransHDM'
summary(object, top = 10, digits = 4, ...)
```

Arguments

object	An object of class "TransHDM".
top	Integer, maximum number of mediators to display.
digits	Number of digits for rounding estimates.
...	Further arguments (unused).

Value

An object of class "summary.TransHDM".

TransHDM *TransHDM: High-Dimensional Mediation Analysis with Transfer Learning*

Description

The TransHDM function performs high-dimensional mediation analysis under a transfer learning framework. It identifies and estimates indirect (mediation) effects of a high-dimensional set of mediators between an exposure and an outcome by integrating a target dataset and a source datasets.

Usage

```
TransHDM(
  target_data,
  source_data = NULL,
  Y,
  D,
  M,
  X,
  transfer = FALSE,
  verbose = TRUE,
  ncore = 1,
```

```
  topN = NULL,
  use_SIS = TRUE,
  dblasso_SIS = FALSE,
  p_cutoff = 0.05
)
```

Arguments

target_data	A data frame containing the target dataset. All variables must be numeric.
source_data	A list of data frames containing source datasets (optional, default: NULL). All variables must be numeric and have the same column names as target_data.
Y	Character string specifying the outcome variable name.
D	Character string specifying the exposure (treatment) variable name.
M	Character vector specifying mediator variable names.
X	Character vector specifying covariate variable names..
transfer	A logical value (default: FALSE) indicating whether to enable transfer learning by incorporating information from source_data.
verbose	A logical value (default: TRUE) controlling whether progress messages are printed to the console.
ncore	An integer (default: 1) specifying the number of CPU cores to use for parallel computation when fitting mediator models.
topN	An integer (default: NULL) specifying the number of mediators to retain after Sure Independence Screening (SIS). If NULL, the number is determined automatically based on the data dimensions.
use_SIS	A logical value (default: TRUE) indicating whether to apply Sure Independence Screening (SIS) for initial variable screening. When FALSE, the full set of mediators is used in estimation and inference.
dblasso_SIS	A logical value (default: FALSE) indicating whether to apply a two-stage procedure combining SIS and debiased Lasso. When TRUE, mediators are first screened via SIS and then debiased Lasso is applied to the reduced set, which is recommended for ultra-high-dimensional settings.
p_cutoff	A numeric value (default: 0.05) specifying the maximum p-value threshold for determining significant mediators.

Details

TransHDM implements a three-step workflow that can also be executed step-by-step using the individual components:

1. **Step 1 — Screening:** SIS performs Sure Independence Screening to reduce dimensionality by ranking mediators based on their marginal associations.
2. **Step 2 — De-biased estimation:** mediation_inference fits de-biased lasso models to estimate exposure-mediator (alpha) and mediator-outcome (beta) effects.
3. **Step 3 — Multiple testing:** joint_test applies joint multiple testing with FDR control to identify significant mediators and computes effect summaries.

Set use_SIS = FALSE to skip Step 1 and use all mediators directly.

Value

A list with the following components:

- `contributions`: A data frame of identified mediators containing:
  - `mediator`: Mediator name
  - `alpha`: Estimated exposure mediator effect
  - `alpha_pv`: P-value for the exposure mediator effect
  - `beta`: Estimated mediator outcome effect
  - `beta_pv`: P-value for the mediator outcome effect
  - `alpha_beta`: Estimated indirect (mediation) effect
  - `ab_pv`: Joint p-value for the indirect effect
  - `pa`: Proportion of the total effect mediated
- `effects`: A data frame summarizing the total indirect effect, direct effect, total effect, and proportion mediated.
- `IDE_est`: A numeric vector of indirect effect estimates for all specified mediators, with non-selected mediators set to zero.
- `DE_est`: The estimated direct effect of the exposure on the outcome.
- `all_mediators`: A data frame of all SIS-screened mediators (including non-significant ones) with columns `mediator`, `alpha`, `alpha_pv`, `beta`, `beta_pv`, and `significant` (logical).

References

Pan L, Liu Y, Huang C, Lin R, Yu Y, Qin G. Transfer learning reveals the mediating mechanisms of cross-ethnic lipid metabolic pathways in the association between APOE gene and Alzheimer's disease. Brief Bioinform. 2025;26(5):bbaf460. doi:10.1093/bib/bbaf460

See Also

`SIS`, `mediation_inference`, `joint_test`

Examples

```
set.seed(123)

# Target data
target_data <- gen_simData_homo(n = 50, p_x = 3, p_m = 20, rho = 0.1)$data

# Source data
source_data <- gen_simData_homo(n = 100, p_x = 3, p_m = 20, rho = 0.1, source = TRUE,
transferable = TRUE)$data

# Run TransHDM
result <- TransHDM(
  target_data = target_data,
  source_data = source_data,
  Y = "Y",
  D = "D",
  M = paste0("M", 1:20),
```

```
  X = paste0("X", 1:3),
  transfer = TRUE,
  ncore = 1,
  topN = 10
)
summary(result)
```

Index

∗ **datasets**
inflam_effect, 5
inflam_external1, 6
inflam_external2, 7
inflam_target, 8

gen_simData_hetero, 2, *8*
gen_simData_homo, 4, *7*, *9*
ggplot, *13*, *15*

inflam_effect, 5, *8, 9*
inflam_external1, 6, *6*, *8, 9*
inflam_external2, *6*, 7, *7*, *9*
inflam_target, *6, 7*, 8, *8*

joint_test, 9, *10*, *12*, *19, 20*, *27, 28*

mediation_inference, *9*, 10, *10*, *19, 20*, *25*, *27, 28*

plot.source_detection, 12
plot.TransHDM, 14
print.dblasso, 16
print.lasso, 16
print.mediation_inference, 17
print.SIS, 17
print.source_detection, 18
print.summary.dblasso, 18
print.TransHDM, 19

SIS, *9–12*, 19, *27, 28*
source_detection, *7*, *13*, 21
summary.dblasso, 23
summary.lasso, 24
summary.mediation_inference, 25
summary.source_detection, 25
summary.TransHDM, 26

theme_classic, *13*, *15*
TransHDM, *6*, *9, 10*, *12*, *19, 20*, 26