Package: MVMR.CUE (via r-universe)

May 15, 2026

Type Package

Title Multivariable Mendelian randomization accounting for complex correlated horizontal pleiotropy while elucidating shared genetic etiology

Version 1.0

Description Multivariable Mendelian randomization accounting for complex correlated horizontal pleiotropy while elucidating shared genetic etiology.

License GPL (>= 3)

Depends R (>= 2.10)

Imports coda, glmnet, MASS, MendelianRandomization, mvtnorm, quantreg, R2jags, Rcpp (>= 1.0.13), robustbase, stats

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

Config/pak/sysreqs cmake libgmp3-dev make jags libicu-dev libuv1-dev libssl-dev

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

Date/Publication 2026-03-16 15:28:01 UTC

RemoteUrl https://github.com/remlapmot/MVMR.CUE

RemoteRef suggestions

RemoteSha b6b793182fe8b57023fcc94aed615b2be2b0a744

Contents

`MVMR.CUE-package` *Multivariable Mendelian Randomization Accounting for Correlated Horizontal Pleiotropy*

Description

Multivariable Mendelian randomization accounting for complex correlated horizontal pleiotropy while elucidating shared genetic etiology.

Author(s)

Qing Cheng <`qingcheng0218@gmail.com`>

`gamma1` *Example MVMR Summary Statistics*

Description

Example summary statistics for multivariable Mendelian randomization, containing association estimates and standard errors for 200 genetic variants with two exposures and one outcome.

Usage

```
data(gamma1)
data(Gamma2)
data(se1)
data(se2)
```

Format

`gamma1` A 200 x 2 matrix of SNP-exposure association estimates.

`Gamma2` A matrix of 200 SNP-outcome association estimates.

`se1` A 200 x 2 matrix of standard errors for SNP-exposure associations.

`se2` A matrix of 200 standard errors for SNP-outcome associations.

`genRaw` *Generate Raw Genetic Data*

Description

Simulates SNP genotype data with a specified minor allele frequency range and correlation structure.

Usage

```
genRaw(L, M, rho, n)
```

Arguments

`L`	Number of independent LD blocks.
`M`	Number of SNPs per LD block.
`rho`	Correlation parameter controlling linkage disequilibrium within blocks.
`n`	Sample size.

Value

An $n \times (L \times M)$ matrix of simulated genotype values (0, 1, or 2).

Examples

```
  ## Not run:
    X <- genRaw(L = 10, M = 5, rho = 0.3, n = 500)

## End(Not run)
```

`mr_horse` *HORSE Mendelian Randomization via MCMC*

Description

Implements the HORSE (Heteroscedastic Outlier-Robust causal effect estimation with Shrinkage for plEiotropy) model for univariate (`mr_horse`) and multivariable (`mvmr_horse`) Mendelian randomization using JAGS.

Usage

```
mr_horse(D, no_ini = 3, variable.names = "theta", n.iter = 10000,
         n.burnin = 10000)
mvmr_horse(D, no_ini = 3, variable.names = "theta", n.iter = 10000,
           n.burnin = 10000)
```

Arguments

`D`	A data frame containing columns `betaY`, `betaX`, `betaYse`, `betaXse` (and `betaX1`, `betaX1se`, `betaX2`, `betaX2se`, etc. for `mvmr_horse`).
`no_ini`	Number of MCMC chains (default 3).
`variable.names`	Character vector of parameters to monitor (default `"theta"`).
`n.iter`	Number of MCMC iterations per chain after burn-in (default 10000).
`n.burnin`	Number of burn-in iterations (default 10000).

Value

A list with elements `MR_Estimate` (a data frame of point estimates and diagnostics) and `MR_Coda` (the MCMC samples as a `mcmc.list`).

Examples

```
  ## Not run:
    D <- data.frame(betaY = Gamma2, betaYse = se2,
                    betaX = gamma1, betaXse = se1)
    res <- mr_horse(D)

## End(Not run)
```

`mvmr_lasso` *Multivariable LASSO Mendelian Randomization*

Description

Implements multivariable MR with LASSO-based invalid instrument selection. `cv.mvmr_lasso` performs cross-validation to select the LASSO penalty parameter; `mvmr_lasso` uses this to obtain post-LASSO causal estimates.

Usage

```
mvmr_lasso(bx, by, seby)
cv.mvmr_lasso(bx, by, seby)
```

Arguments

`bx`	Matrix of SNP-exposure association estimates (SNPs x exposures).
`by`	Vector of SNP-outcome association estimates.
`seby`	Vector of standard errors for SNP-outcome associations.

Value

`cv.mvmr_lasso` returns a list with elements `fit` (LASSO coefficients at the selected lambda) and `lambda`.

`mvmr_lasso` returns a list with elements `thest`, `a`, `lambda`, `th_post`, and `se_post`.

Examples

```
  ## Not run:
    res <- mvmr_lasso(bx = gamma1, by = Gamma2, seby = se2)

## End(Not run)
```

`mvmr_median` *Multivariable Weighted Median Mendelian Randomization*

Description

Implements multivariable weighted median MR using quantile regression. `mvmr_med_boot` computes bootstrap standard errors.

Usage

```
mvmr_median(bx, sebx, by, seby, boot = FALSE, boot_it = 1000)
mvmr_med_boot(bx, sebx, by, seby, N)
```

Arguments

`bx`	Matrix of SNP-exposure association estimates (SNPs x exposures).
`sebx`	Matrix of standard errors for SNP-exposure associations.
`by`	Vector of SNP-outcome association estimates.
`seby`	Vector of standard errors for SNP-outcome associations.
`boot`	Logical; if `TRUE`, bootstrap standard errors are computed.
`boot_it`	Number of bootstrap iterations (default 1000).
`N`	Number of bootstrap iterations in `mvmr_med_boot`.

Value

`mvmr_median` returns a list with element `coefficients` and, if `boot = TRUE`, element `se`.

`mvmr_med_boot` returns a vector of bootstrap standard errors.

Examples

```
  ## Not run:
    res <- mvmr_median(bx = gamma1, sebx = se1, by = Gamma2, seby = se2)

## End(Not run)
```

`mvmr_robust` *Multivariable Robust Regression Mendelian Randomization*

Description

Implements multivariable MR using robust regression (MM-estimator) with inverse-variance weighting.

Usage

```
mvmr_robust(bx, by, seby, k.max = 500, maxit.scale = 500)
```

Arguments

`bx`	Matrix of SNP-exposure association estimates (SNPs x exposures).
`by`	Vector of SNP-outcome association estimates.
`seby`	Vector of standard errors for SNP-outcome associations.
`k.max`	Maximum number of iterations for the M-step (default 500).
`maxit.scale`	Maximum number of iterations for the scale estimation (default 500).

Value

A list with elements `coefficients` and `se`.

Examples

```
  ## Not run:
    res <- mvmr_robust(bx = gamma1, by = Gamma2, seby = se2)

## End(Not run)
```

`MVMRCUE` *Multivariable Mendelian Randomization Accounting for Correlated Pleiotropy*

Description

Core functions implementing the MVMR-CUE algorithm for multivariable Mendelian randomization that accounts for complex correlated horizontal pleiotropy.

Usage

```
MVMRCUE(gammahM, Gammah, se1M, se2, Re, opts = NULL)
MVMRCUEfun(gammahM, Gammah, se1M, se2)
MVMRCUEoverlapfun(gammahM, Gammah, se1M, se2, Re, Eta)
MVMRCUEIndepSample(gammahM, Gammah, se1M, se2, opts = NULL)
```

Arguments

`gammahM`	Matrix of SNP-exposure association estimates.
`Gammah`	Vector of SNP-outcome association estimates.
`se1M`	Matrix of standard errors for SNP-exposure associations.
`se2`	Vector of standard errors for SNP-outcome associations.
`Re`	Correlation matrix for overlapping samples.
`Eta`	Eta parameter for overlapping samples.
`opts`	Optional list of algorithm options.

Value

A list containing the estimated causal effects and associated statistics.

Examples

```
  ## Not run:
    result <- MVMRCUE(gamma1, Gamma2, se1, se2, Re = diag(nrow(gamma1)))

## End(Not run)
```

`RcppArmadillo-Functions` *Set of functions in example RcppArmadillo package*

Description

These four functions are created when `RcppArmadillo.package.skeleton()` is invoked to create a skeleton packages.

Usage

```
rcpparma_hello_world()
rcpparma_outerproduct(x)
rcpparma_innerproduct(x)
rcpparma_bothproducts(x)
```

Arguments

`x`	a numeric vector

Details

These are example functions which should be largely self-explanatory. Their main benefit is to demonstrate how to write a function using the Armadillo C++ classes, and to have to such a function accessible from R.

Value

`rcpparma_hello_world()` does not return a value, but displays a message to the console.

`rcpparma_outerproduct()` returns a numeric matrix computed as the outer (vector) product of `x`.

`rcpparma_innerproduct()` returns a double computer as the inner (vector) product of `x`.

`rcpparma_bothproducts()` returns a list with both the outer and inner products.

Author(s)

Dirk Eddelbuettel

References

See the documentation for Armadillo, and RcppArmadillo, for more details.

Examples

```
x <- sqrt(1:4)
rcpparma_innerproduct(x)
rcpparma_outerproduct(x)
```

Index