Package ‘PPMR’

September 17, 2025

Title Probabilistic Two Sample Mendelian Randomization

Type Package

Version 1.0.1

Description Efficient statistical inference of two-sample MR (Mendelian Randomization) analysis. It can account for the correlated instruments and the horizontal pleiotropy, and can provide the accurate estimates of both causal effect and horizontal pleiotropy effect as well as the two corresponding p-values. There are two main functions in the 'PPMR' package. One is PMR_individual() for individual level data, the other is PMR_summary() for summary data.

License GPL-3

Encoding UTF-8

LazyData true

Imports Rcpp (>= 1.0.0)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

NeedsCompilation yes

BugReports `https://github.com/umich-biostatistics/PPMR/issues`

Author Zhongshang Yuan [aut],
Xiang Zhou [aut],
Michael Kleinsasser [cre]

Maintainer Michael Kleinsasser <`mkleinsa@umich.edu`>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-09-17 19:20:09 UTC

Contents

`Exampleindividual` *Individual level dataset*

Description

A simulated individual level dataset for PMR.

Usage

```
Exampleindividual
```

Format

A list contains the following objects:

zx the standardized genotype matrix for 465 individuals and 50 cis-SNPs in eQTL data.

zy the standardized genotype matrix for 2000 individuals and 50 cis-SNPs in GWAS data.

x the standarized gene expression vector for 465 individuals in eQL data.

y the standarized complex trait vector for 2000 individuals in GWAS data.

`Examplesummary` *Summary level dataset*

Description

A simulated summary level dataset for PMR

Usage

```
Examplesummary
```

Format

A list contains the following objects:

betax the cis-SNP effect size vector for one specific gene in eQTL data.

betay the cis-SNP effect size vector for one specific gene in GWAS data.

Sigma1 the LD matrix in eQTL data.

Sigma2 the LD matrix in GWAS data.

n1 the sample size of eQTL data.

n2 the sample size of GWAS data.

`PMR_individual` *PPMR Individual-level Analysis*

Description

PPMR Individual-level Analysis

Usage

```
PMR_individual(yin, zin, x1in, x2in, gammain, alphain, max_iterin, epsin)
```

Arguments

`yin`	Numeric vector of the outcome variable (length $n1$).
`zin`	Numeric vector of the mediator variable (length $n2$).
`x1in`	Numeric $n1 \times p$ matrix of SNP genotypes for the outcome model.
`x2in`	Numeric $n2 \times p$ matrix of SNP genotypes for the mediator model. Must have the same number of columns p as `x1in`.
`gammain`	Integer flag (0/1). If 1, constrains the gamma parameter to 0.
`alphain`	Integer flag (0/1). If 1, constrains the alpha parameter to 0.
`max_iterin`	Integer. Maximum number of EM iterations (default: 50 or more).
`epsin`	Numeric. Convergence tolerance for the log-likelihood.

Value

A named list with elements:

`alpha`	Estimated causal effect of the mediator on the outcome.
`gamma`	Estimated direct effect of the SNPs on the outcome.
`sigmaX`	Residual variance of the outcome model.
`sigmaY`	Residual variance of the mediator model.
`sigmabeta`	Variance of the genetic effects.
`loglik_seq`	Vector of log-likelihood values across iterations.
`loglik`	Final log-likelihood value.
`iteration`	Number of iterations before convergence.

Examples

```
# ---- Simulate simple example data ----
set.seed(456)
n1 <- 8
n2 <- 10
p  <- 3

# Outcome and mediator vectors
y <- c(0.5, -0.3, 0.1, 0.4, -0.2, 0.0, 0.6, -0.1)
z <- c(0.2, -0.4, 0.3, 0.1, -0.1, 0.5, 0.0, 0.4, -0.3, 0.2)

# Fixed genotype design matrices (n × p) with mild correlations
x1 <- matrix(c(
  1.0, 0.2, 0.1,
  0.2, 1.0, 0.3,
  0.1, 0.3, 1.0,
  0.4, 0.1, 0.2,
  0.2, 0.4, 0.3,
  0.3, 0.2, 0.4,
  0.5, 0.1, 0.3,
  0.1, 0.5, 0.2
), nrow = n1, byrow = TRUE)

x2 <- matrix(c(
  1.0, 0.3, 0.2,
  0.3, 1.0, 0.4,
  0.2, 0.4, 1.0,
  0.5, 0.1, 0.3,
  0.2, 0.5, 0.1,
  0.3, 0.2, 0.4,
  0.4, 0.3, 0.2,
  0.1, 0.4, 0.3,
  0.2, 0.1, 0.5,
  0.3, 0.2, 0.4
), nrow = n2, byrow = TRUE)

# Run PPMR individual-level analysis
PMR_individual(
  yin        = y,
  zin        = z,
  x1in       = x1,
  x2in       = x2,
  gammain    = 0,
  alphain    = 0,
  max_iterin = 50,
  epsin      = 1e-6
)
```

`PMR_summary` *PPMR Summary-level Analysis*

Description

PPMR Summary-level Analysis

Usage

```
PMR_summary(
  betaxin,
  betayin,
  Sigma1sin,
  Sigma2sin,
  samplen1,
  samplen2,
  gammain,
  alphain,
  max_iterin,
  epsin
)
```

Arguments

`betaxin`	Numeric vector of estimated SNP–exposure effects (length ‘p‘).
`betayin`	Numeric vector of estimated SNP–outcome effects (length ‘p‘).
`Sigma1sin`	Numeric ‘p x p‘ covariance matrix for the exposure SNP associations (typically an LD matrix).
`Sigma2sin`	Numeric ‘p x p‘ covariance matrix for the outcome SNP associations.
`samplen1`	Integer. Sample size used to estimate `betaxin`.
`samplen2`	Integer. Sample size used to estimate `betayin`.
`gammain`	Integer flag (0/1). If 1, constrains the gamma parameter to 0.
`alphain`	Integer flag (0/1). If 1, constrains the alpha parameter to 0.
`max_iterin`	Integer. Maximum number of EM iterations (default: 50 or more).
`epsin`	Numeric. Convergence tolerance for the log-likelihood.

Value

A named list with elements:

`alpha`	Estimated causal effect of the mediator on the outcome.
`gamma`	Estimated direct effect of the SNPs on the outcome.
`sigmaX`	Residual variance for the exposure model.
`sigmaY`	Residual variance for the outcome model.
`sigmabeta`	Variance of the genetic effects.
`loglik_seq`	Vector of log-likelihood values across iterations.
`loglik`	Final log-likelihood value.
`iteration`	Number of iterations used before convergence.

Examples

```
# ---- Simulate simple example data ----
set.seed(123)
p  <- 3
n1 <- 10
n2 <- 12
betax <- c(0.2, -0.1, 0.3)
betay <- c(0.1,  0.0, 0.2)
Sigma1 <- matrix(c(0.6, 0.2, 0.1,
                   0.2, 0.5, 0.1,
                   0.1, 0.1, 0.4), 3, 3)
Sigma2 <- matrix(c(0.5, 0.1, 0.0,
                   0.1, 0.6, 0.1,
                   0.0, 0.1, 0.5), 3, 3)

PMR_summary(
  betaxin   = betax,
  betayin   = betay,
  Sigma1sin = Sigma1,
  Sigma2sin = Sigma2,
  samplen1  = n1,
  samplen2  = n2,
  gammain   = 0,
  alphain   = 0,
  max_iterin = 50,
  epsin      = 1e-6
)
```

Index