Package: MRPATH (via r-universe)

October 1, 2024

Type Package

Title R Package for MR-PATH

Version 1.0

Date 2020-07-14

Author Daniel Iong

Maintainer Daniel <daniong@umich.edu>

Description This package implements methods for fitting the MR-PATH model.

License GPL-3

Depends R (>= 3.5.0)

Imports cowplot, ggplot2, plotly, Rcpp (>= 1.0.3), RcppArmadillo, RColorBrewer, Rdpack (>= 0.7), reshape2, stats, magrittr

Suggests loo

LinkingTo Rcpp, RcppArmadillo, RcppDist

RoxygenNote 7.1.0

RdMacros Rdpack

Encoding UTF-8

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

RemoteUrl https://github.com/remlapmot/MRPATH

RemoteRef suggestions

RemoteSha cbb795809199c1e41d18e0710f94a3beb47e7c98

Contents

`MRPATH-package` *MR-PATH: A Latent Mixture for Heterogenous Causal Mechanisms in Mendelian Randomization*

Description

Mendelian Randomization (MR) is a popular method in epidemiology and genetics that uses genetic variation as instrumental variables for causal inference. This package implements methods for fitting the MR-PATH model from (Iong et al. 2020).

Author(s)

Daniel Iong, daniong@umich.edu

Qingyuan Zhao, qyzhao@statslab.cam.ac.uk

Yang Chen, ychenang@umich.edu

Maintainer: Daniel Iong <daniong@umich.edu>

References

Iong D, Zhao Q, Chen Y (2020). "A Latent Mixture Model for Heterogeneous Causal Mechanisms in Mendelian Randomization." 2007.06476.

`bmi_t2d` *Effect of Body Mass Index (BMI) on Type-2 Diabetes (T2D)*

Description

This dataset is created from three genome-wide association studies using the three-sample summary-data MR design (Zhao et al. 2019):

1. **Selection**: Akiyama et al. (2017)
2. **Exposure**: Locke et al. (2015)
3. **Outcome**: Mahajan et al. (2018)

The 60 SNPs selected are independent (distance $\geq$ 10 mega base pairs, $R^2 \leq 0.001$ in a reference panel) and are associated with T2D (p-value less than $5 * 10^{-8}$).

Usage

```
data(bmi_t2d)
```

Format

A `data.frame` with 60 rows and 6 variables.

References

Akiyama M, Okada Y, Kanai M, Takahashi A, Momozawa Y, Ikeda M, Iwata N, Ikegawa S, Hirata M, Matsuda K, others (2017). "Genome-wide association study identifies 112 new loci for body mass index in the Japanese population." *Nature Genetics*, **49**(10), 1458.

Locke AE, Kahali B, Berndt SI, Justice AE, Pers TH, Day FR, Powell C, Vedantam S, Buchkovich ML, Yang J, others (2015). "Genetic studies of body mass index yield new insights for obesity biology." *Nature*, **518**(7538), 197–206.

Mahajan A, Taliun D, Thurner M, Robertson NR, Torres JM, Rayner NW, Payne AJ, Steinthorsdottir V, Scott RA, Grarup N, others (2018). "Fine-mapping type 2 diabetes loci to single-variant resolution using high-density imputation and islet-specific epigenome maps." *Nature genetics*, **50**(11), 1505–1513.

Zhao Q, Chen Y, Wang J, Small DS (2019). "Powerful three-sample genome-wide design and robust statistical inference in summary-data Mendelian randomization." *International Journal of Epidemiology*, **48**(5), 1478–1492. ISSN 14643685, doi:10.1093/ije/dyz142, https://academic.oup.com/ije/advance-article-abstract/doi/10.1093/ije/dyz142/5531250.

`computeClusterMembProb` *Cluster membership probabilities in MR-PATH*

Description

Computes SNP-specific cluster membership probabilities from the MR-PATH model.

Usage

```
computeClusterMembProb(data, Nsamples = 50000,
    impt_samples = NULL, MCEM_fit = NULL)
```

Arguments

`data`	A data frame. (see Details in `MR_PATH`).
`Nsamples`	Number of desired samples.
`impt_samples`	Importance samples from `getImportanceSamples`. If `NULL`, obtain importance samples within function.
`MCEM_fit`	MC-EM fit from `MR_PATH`. If `impt_samples = NULL`, use it for obtaining importance samples. Default is `NULL`.

Details

If `data` contains a column named `SNP`, the rows of the output matrix will be labeled by SNP name.

Value

Returns a `p` by `K` matrix containing cluster membership probabilities.

References

https://arxiv.org/abs/2007.06476

See Also

MR_PATH, sampleBetas

Examples

```
### Load data
data(hdl_chd)

### Filter weak instruments
hdl_chd = hdl_chd[hdl_chd$pval.selection < 5e-8,]

### Set your own K.
# For data-driven model selection, use MRPATH_selectModel
K = 2

### Set your own initial values.
# For automatic initial value selection, use MRPATH_optimizeInitVals
initVals = list("m_X" = mean(hdl_chd$beta.exposure),
                "lambdaX" = sd(hdl_chd$beta.exposure),
                "pis" = rep(1/K, K),
                "mus" = c(-.9, .4),
                "sds" = c(.9, .4))

### Run MC-EM algorithm
MCEM_fit = MR_PATH(K, hdl_chd, initVals)

### Compute SNP-specific cluster membership probabilities
clustermemb_prob = computeClusterMembProb(hdl_chd, MCEM_fit = MCEM_fit)
```

---

getImportanceSamples *Importance Sampling of SNP-specific latent variables in MR-PATH*

Description

Obtain importance samples of SNP-specific latent variables in the MR-PATH model

Usage

```
getImportanceSamples(data, MCEM_fit, Nsamples = 50000)
```

Arguments

Nsamples	Number of desired samples
data	A data frame. (see Details in MR_PATH).
MCEM_fit	Output from MR_PATH.

Value

thetaX	p by Nsamples matrix of importance samples of true marginal SNP-exposure effects
beta	p by Nsamples matrix of importance samples of SNP-specific causal effects
alpha	p by Nsamples by K array of importance samples of cluster membership probabilities.
W	p by Nsamples matrix of importance weights

See Also

sampleBetas, computeClusterMembProb

Examples

```
### Load data
data(hdl_chd)

### Filter weak instruments
hdl_chd = hdl_chd[hdl_chd$pval.selection < 5e-8,]

### Set your own K.
# For data-driven model selection, use MRPATH_selectModel
K = 2

### Set your own initial values.
# For automatic initial value selection, use MRPATH_optimizeInitVals
initVals = list("m_X" = mean(hdl_chd$beta.exposure),
                "lambdaX" = sd(hdl_chd$beta.exposure),
                "pis" = rep(1/K, K),
                "mus" = c(-.9, .4),
                "sds" = c(.9, .4))

### Run MC-EM algorithm
MCEM_fit = MR_PATH(K, hdl_chd, initVals)

### Obtain importance samples
impt_samples = getImportanceSamples(hdl_chd, MCEM_fit)
```

`hdl_chd` *Effect of HDL Cholesterol (HDL-C) on Coronary Heart Disease (CHD)*

Description

This dataset is created from three genome-wide association studies using the three-sample summary-data MR design (Zhao et al. 2019):

1. **Selection**: GWAS of HDL-C by Teslovich et al. (2010)
2. **Exposure**: GWAS of lipoprotein subfractions by Kettunen et al. (2016)
3. **Outcome**: The CARDIoGRAMplusC4D with 1000 Genome Project imputation GWAS of CAD (Nikpay et al. 2015).

The 151 SNPs selected are independent (distance ≥ 10 mega base pairs, $R^2 \leq 0.001$ in a reference panel) and are associated with at least one plasma lipid trait (the minimum p-value with HDl-C, LDL-C, and triglycerides is less than 10^{-4}).

Usage

```
data(hdl_chd)
```

Format

A `data.frame` with 151 rows and 6 variables.

References

Kettunen J, Demirkan A, Würtz P, Draisma HH, Haller T, Rawal R, Vaarhorst A, Kangas AJ, Lyytikäinen L, Pirinen M, others (2016). "Genome-wide study for circulating metabolites identifies 62 loci and reveals novel systemic effects of LPA." *Nature communications*, **7**(1), 1–9.

Nikpay M, Goel A, Won H, Hall LM, Willenborg C, Kanoni S, Saleheen D, Kyriakou T, Nelson CP, Hopewell JC, others (2015). "A comprehensive 1000 Genomes–based genome-wide association meta-analysis of coronary artery disease." *Nature Genetics*, **47**(10), 1121.

Teslovich TM, Musunuru K, Smith AV, Edmondson AC, Stylianou IM, Koseki M, Pirruccello JP, Ripatti S, Chasman DI, Willer CJ, others (2010). "Biological, clinical and population relevance of 95 loci for blood lipids." *Nature*, **466**(7307), 707–713.

Zhao Q, Chen Y, Wang J, Small DS (2019). "Powerful three-sample genome-wide design and robust statistical inference in summary-data Mendelian randomization." *International Journal of Epidemiology*, **48**(5), 1478–1492. ISSN 14643685, doi:10.1093/ije/dyz142, https://academic.oup.com/ije/advance-article-abstract/doi/10.1093/ije/dyz142/5531250.

MRPATH_barplot *MR-PATH Bar Plot*

Description

Plots a barplot for credible intervals and cluster membership probabilities with SNPs ordered by median of SNP-specific causal effects.

Usage

```
MRPATH_barplot(data, MCEM_fit, ret.snps = FALSE)
```

Arguments

data	A data frame. (see Details in MR_PATH).
MCEM_fit	Output from MR_PATH.
ret.snps	If TRUE, returns both plot and a list of ordered SNPs. Default is FALSE.

Value

Returns a 95% credible interval plot (top) and cluster membership probabilities barplot (bottom).

References

https://arxiv.org/abs/2007.06476

See Also

MRPATH_scatterplot

Examples

```
### Load data
data(hdl_chd)

### Filter weak instruments
hdl_chd = hdl_chd[hdl_chd$pval.selection < 5e-8,]

### Set your own K.
# For data-driven model selection, use MRPATH_selectModel
K = 2

### Set your own initial values.
# For automatic initial value selection, use MRPATH_optimizeInitVals
initVals = list("m_X" = mean(hdl_chd$beta.exposure),
                "lambdaX" = sd(hdl_chd$beta.exposure),
                "pis" = rep(1/K, K),
                "mus" = c(-.9, .4),
                "sds" = c(.9, .4))
```

```
### Run MC-EM algorithm
MCEM_fit = MR_PATH(K, hdl_chd, initVals)

### Plot barplots
MRPATH_barplot(hdl_chd, MCEM_fit)
```

MRPATH_optimizeInitVals

MR-PATH Initial Value Optimizer

Description

Runs MR_PATH multiple times and picks fit with highest complete-data log-likelihood.

Usage

```
MRPATH_optimizeInitVals(K, data, Nreps = 10,
    verbose = FALSE, altModel = FALSE, init_seed = 8686, ...)
```

Arguments

K	Number of desired clusters (see Details).
data	A data frame. (see Details in MR_PATH).
Nreps	Number of repetitions to run MR_PATH.
verbose	If TRUE, returns complete-data log-likelihood at each repetition.
altModel	If TRUE, fits alternative model *(This is still work in progress)*. Default is FALSE.
init_seed	Initial seed (for reproducibility).
...	Additional parameters to be passed to MR_PATH.

Details

K can be chosen by the user or by the modified BIC criterion with MRPATH_selectModel.

Value

Returns a list with

fit	MC-EM fit with optimal initial values.
initVals	Optimal initial values.

References

Iong D, Zhao Q, Chen Y (2020). "A Latent Mixture Model for Heterogeneous Causal Mechanisms in Mendelian Randomization." 2007.06476.

See Also

MR_PATH, MRPATH_selectModel

Examples

```
require(MRPATH)
### Load HDL-CHD data
data(hdl_chd)

### Filter weak instruments
hdl_chd = hdl_chd[hdl_chd$pval.selection < 5e-8,]

### Set your own K.
# For data-driven model selection, use MRPATH_selectModel
K = 2

Nreps = 10 # Testing
# Nreps = 30 # Takes longer, but more stable results
verbose = TRUE # Print output
initValsObj = MRPATH_optimizeInitVals(K = K, data = hdl_chd, Nreps = Nreps, verbose=verbose)

# See optimized initial values
initValsObj$initVals

# See MC-EM fit with optimized initial values
initValsObj$fit
```

MRPATH_scatterplot *MR-PATH Scatter Plot*

Description

Scatterplot with error bars and results from MR_PATH.

Usage

```
MRPATH_scatterplot(data, MCEM_fit = NULL,
exposure_name = "exposure", outcome_name = "outcome",
overDispersedY = FALSE, interactive = TRUE)
```

Arguments

data	A data frame. (see Details in MR_PATH).
MCEM_fit	Output from MR_PATH. If NULL, just plot the data and error bars.
exposure_name	Name of risk exposure variable.
outcome_name	Name of disease outcome variable.
overDispersedY	If TRUE, replaces se.exposure with tau * se.exposure.
interactive	Logical, if plot is interactive. Default TRUE.

Details

Each point is colored according to the cluster with the highest cluster membership probability. Solid lines represent `mus` and shaded regions represent 68% interval for each cluster.

Value

Returns a `ggplot` scatterplot with error bars and results from `MR_PATH`.

References

https://arxiv.org/abs/2007.06476

See Also

`MRPATH_barplot`

Examples

```
### Load data
data(hdl_chd)

### Filter weak instruments
hdl_chd = hdl_chd[hdl_chd$pval.selection < 5e-8,]

### Plot data without MC-EM fit results
MRPATH_scatterplot(hdl_chd)

### Set your own K.
# For data-driven model selection, use MRPATH_selectModel
K = 2

### Set your own initial values.
# For automatic initial value selection, use MRPATH_optimizeInitVals
initVals = list("m_X" = mean(hdl_chd$beta.exposure),
                "lambdaX" = sd(hdl_chd$beta.exposure),
                "pis" = rep(1/K, K),
                "mus" = c(-.9, .4),
                "sds" = c(.9, .4))

### Run MC-EM algorithm
MCEM_fit = MR_PATH(K, hdl_chd, initVals)

### Plot scatterplot with MC-EM fit results
MRPATH_scatterplot(hdl_chd, MCEM_fit)
```

MRPATH_selectModel *Model selection for MR-PATH model using a modified BIC criterion.*

Description

Runs MR_PATH multiple times and picks fit with highest complete-data log-likelihood.

Usage

```
MRPATH_selectModel(data, K_range = 1:3, Nreps = 20,
    altModel = FALSE, verbose=FALSE, ...)
```

Arguments

data	A data frame. (see Details in MR_PATH).
K_range	Range of K values to select from.
Nreps	Number of repetitions for MRPATH_optimizeInitVals.
altModel	If TRUE, fits alternative model *(This is still work in progress)*. Default is FALSE.
verbose	If TRUE, prints BIC for each value in K_range.
...	Additional parameters to be passed to MRPATH_optimizeInitVals.

Value

Returns a list with

bestK	K with highest BIC.
bestFit	MC-EM fit with highest BIC.
Q	Vector of complete-data log-likelihoods for each K in K_range.
BIC	Vector of BIC values for each K in K_range.

References

https://arxiv.org/abs/2007.06476

See Also

MR_PATH, MRPATH_optimizeInitVals

Examples

```
require(MRPATH)
### Load data
data(hdl_chd)

### Filter weak instruments
hdl_chd = hdl_chd[hdl_chd$pval.selection < 5e-8,]

Nreps = 10 # Testing
# Nrep = 30 # Takes longer, but more stable results
verbose = TRUE # Print output
K_range = 1:3 # Set range of K
modSelectionObj = MRPATH_selectModel(hdl_chd, K_range = K_range,
                                     Nreps = Nreps, verbose = verbose)

# See optimal K
modSelectionObj$bestK

# See optimal fit
modSelectionObj$bestFit

# See vector of complete-data log-likelihood values
modSelectionObj$Q

# See vector of BIC values
modSelectionObj$BIC
```

MR_PATH *MR-PATH Model Fitting*

Description

Fits MR-PATH model with MC-EM algorithm.

Usage

```
MR_PATH(K, data, initVals, overDispersedY = FALSE,
    equalSds = FALSE, computeSE = TRUE, Nstart_MC = 500L,
    M = 4L, max_Nsamples = 500000L, min_iters = 2L,
    max_iters = 100L, alpha = 0.05, gamma = 0.05, eps = 0.005,
    verbose = FALSE, saveTraj = FALSE)
```

Arguments

K	Number of desired clusters (see Details).
data	A data frame (see Details)
initVals	List of initial values (see Details).

overDispersedY	If TRUE, estimates multiplicative over-dispersion parameter for SNP-outcome effects. Default is FALSE. (This is still being tested.)
equalSds	If TRUE, assume mixture components have the same standard deviation. Default is FALSE.
computeSE	If TRUE, compute standard errors of parameter estimates. Default is TRUE.
Nstart_MC	Initial Monte-Carlo sample size for E-step MC approximation. Default is 500.
M	Geometric rate at which MC sample size is increased. Default is 4.
max_Nsamples	Max MC sample size for E-step MC approximation. Default is 500000.
min_iters	Min. number of iterations. Default is 2.
max_iters	Max number of iterations. Default is 100.
alpha	Threshold for Type 1 error rate in E-step approximation. Default is 0.05.
gamma	Threshold for Type 1 error rate in convergence criterion. Default is 0.05.
eps	Threshold for convergence criterion. Default is 0.005.
verbose	If TRUE, prints output at each iteration. Default is FALSE.
saveTraj	If TRUE, save MC-EM trajectory for each parameter. Default is FALSE.

Details

K can be chosen by the user or by the modified BIC criterion with MRPATH_selectModel.

The input data frame must contain the following variables:

1. beta.exposure
2. beta.outcome
3. se.exposure
4. se.outcome

initVals must be a list with the following elements:

1. m_X
2. lambdaX
3. pi
4. mus
5. sds

Since the MC-EM algorithm reaches local optima, it is recommended to run MR_PATH with a couple different initial values and pick the fit with the highest completeDataLogLik. One way to pick initial values is to first visualize the data with MRPATH_scatterplot. For automatic initial value selection, see MRPATH_optimizeInitVals.

MR_PATH returns the following information about convergence of the MC-EM algorithm:

1. Niters: Number of iterations it takes to converge.
2. N_MC_end: Number of MC samples at convergence.
3. completeDataLogLik: complete-data log-likelihood value at convergence.

Value

Returns a list containing

`paramEst`	List of parameter estimates.
`standardErrors`	Vector of standard errors.
`convergenceInfo`	List of information about convergence of MC-EM (see Details).
`paramTraj`	If `saveTraj = TRUE`, returns matrix containing MC-EM trajectory for each parameter.

References

Iong D, Zhao Q, Chen Y (2020). "A Latent Mixture Model for Heterogeneous Causal Mechanisms in Mendelian Randomization." 2007.06476.

See Also

`MRPATH_optimizeInitVals`, `MRPATH_selectModel`, `MRPATH_scatterplot`

Examples

```
require(MRPATH)
### Load HDL-CHD data
data(hdl_chd)

### Filter weak instruments
hdl_chd = hdl_chd[hdl_chd$pval.selection < 5e-8,]

### Set your own K.
# For data-driven model selection, use MRPATH_selectModel
K = 2

### Set your own initial values.
# For automatic initial value selection, use MRPATH_optimizeInitVals
initVals = list("m_X" = mean(hdl_chd$beta.exposure),
                "lambdaX" = sd(hdl_chd$beta.exposure),
                "pis" = rep(1/K, K),
                "mus" = c(-.9, .4),
                "sds" = c(.9, .4))

### Run MC-EM algorithm
MCEM_fit = MR_PATH(K, hdl_chd, initVals)
```

sampleBetas *Probabilistic inference for SNP-specific causal effects in MR-PATH*

Description

Re-samples importance samples to obtain samples of SNP-specific causal effects in MR-PATH

Usage

```
sampleBetas(data, Nsamples = 50000, impt_samples = NULL,
    MCEM_fit = NULL)
```

Arguments

data	A data frame. (see Details in MR_PATH).
Nsamples	Number of desired samples.
impt_samples	Importance samples from getImportanceSamples. If NULL, obtain importance samples within function.
MCEM_fit	MC-EM fit from MR_PATH. If impt_samples = NULL, use it for obtaining importance samples. Default is NULL.

Details

If data contains a column named SNP, the rows of the output matrix will be labeled by SNP name.

Value

Returns a p by Nsamples matrix containing samples of SNP-specific causal effects.

References

https://arxiv.org/abs/2007.06476

See Also

MR_PATH, computeClusterMembProb

Examples

```
### Load data
data(hdl_chd)

### Filter weak instruments
hdl_chd = hdl_chd[hdl_chd$pval.selection < 5e-8,]

### Set your own K.
# For data-driven model selection, use MRPATH_selectModel
K = 2
```

```
### Set your own initial values.
# For automatic initial value selection, use MRPATH_optimizeInitVals
initVals = list("m_X" = mean(hdl_chd$beta.exposure),
                "lambdaX" = sd(hdl_chd$beta.exposure),
                "pis" = rep(1/K, K),
                "mus" = c(-.9, .4),
                "sds" = c(.9, .4))

### Run MC-EM algorithm
MCEM_fit = MR_PATH(K, hdl_chd, initVals)

# Sample SNP-specific causal effects
beta_samples = sampleBetas(hdl_chd, MCEM_fit = MCEM_fit)
```

Index