Package ‘MR.RGM’

March 2, 2024

Title Multivariate Bidirectional Mendelian Randomization Networks

Version 0.0.2

Description Addressing a central challenge encountered in Mendelian randomization (MR) studies, where MR primarily focuses on discerning the effects of individual exposures on specific outcomes and establishes causal links between them. Using a network-based methodology, the intricacy involving interdependent outcomes due to numerous factors has been tackled through this routine. Based on Ni et al. (2018) <doi:10.1214/17-BA1087>, 'MR.RGM' extends to a broader exploration of the causal landscape by leveraging on network structures and involves the construction of causal graphs that capture interactions between response variables and consequently between responses and instrument variables. 'MR.RGM' facilitates the navigation of various data availability scenarios effectively by accommodating three input formats, i.e., individual-level data and two types of summary-level data. In the process, causal effects, adjacency matrices, and other essential parameters of the complex biological networks, are estimated. Besides, 'MR.RGM' provides uncertainty quantification for specific network structures among response variables.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

URL https://github.com/bitansa/MR.RGM

BugReports https://github.com/bitansa/MR.RGM/issues

Imports Rcpp, stats

Suggests MASS, igraph

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Bitan Sarkar [aut, cre],
Yang Ni [aut]

Maintainer Bitan Sarkar <bitan@tamu.edu>

Repository CRAN

Date/Publication 2024-03-02 07:12:37 UTC

R topics documented:

NetworkMotif *Estimating the uncertainty of a specified network*

Description

The NetworkMotif function facilitates uncertainty quantification. Specifically, it determines the proportion of posterior samples that contains the given network structure. To use this function, users may use the Gamma_Pst output obtained from the RGM function.

Usage

```
NetworkMotif(Gamma, Gamma_Pst)
```

Arguments

Gamma	A matrix of dimension p * p that signifies a specific network structure among the response variables, where p represents the number of response variables. This matrix is the focus of uncertainty quantification.
Gamma_Pst	An array of dimension p * p * n_pst, where n_pst is the number of posterior samples and p denotes the number of response variables. It comprises the posterior samples of the causal network among the response variables. This input might be obtained from the RGM function. Initially, execute the RGM function and save the resulting Gamma_Pst. Subsequently, utilize this stored Gamma_Pst as input for this function.

Value

The NetworkMotif function calculates the uncertainty quantification for the provided network structure. A value close to 1 indicates that the given network structure is frequently observed in the posterior samples, while a value close to 0 suggests that the given network structure is rarely observed in the posterior samples.

References

Ni, Y., Ji, Y., & Müller, P. (2018). Reciprocal graphical models for integrative gene regulatory network analysis. *Bayesian Analysis*, **13(4)**, 1095-1110. doi:10.1214/17BA1087.

Examples

```
#' # --------------------------------------------------------

# Example 1:
# Run NetworkMotif to do uncertainty quantification for a given network among the response variable

# Data Generation
set.seed(9154)

# Number of data points
n = 10000

# Number of response variables and number of instrument variables
p = 3
k = 4

# Initialize causal interaction matrix between response variables
A = matrix(sample(c(-0.1, 0.1), p^2, replace = TRUE), p, p)

# Diagonal entries of A matrix will always be 0
diag(A) = 0

# Make the network sparse
A[sample(which(A!=0), length(which(A!=0))/2)] = 0

# Initialize causal interaction matrix between response and instrument variables
B = matrix(0, p, k)

# Create d vector
d = c(2, 1, 1)


# Initialize m
m = 1

# Calculate B matrix based on d vector
for (i in 1:p) {

 # Update ith row of B
 B[i, m:(m + d[i] - 1)] = 1

 # Update m
 m = m + d[i]

}

Sigma = diag(p)

Mult_Mat = solve(diag(p) - A)
```

```
Variance = Mult_Mat %*% Sigma %*% t(Mult_Mat)

# Generate instrument data matrix
X = matrix(rnorm(n * k, 0, 1), nrow = n, ncol = k)

# Initialize response data matrix
Y = matrix(0, nrow = n, ncol = p)

# Generate response data matrix based on instrument data matrix
for (i in 1:n) {

    Y[i, ] = MASS::mvrnorm(n = 1, Mult_Mat %*% B %*% X[i, ], Variance)

}


# Apply RGM on individual level data for Threshold Prior
Output = RGM(X = X, Y = Y, d = c(2, 1, 1), prior = ”Threshold”)

# Store Gamma_Pst
Gamma_Pst = Output$Gamma_Pst

# Define a function to create smaller arrowheads
smaller_arrowheads = function(graph) {
    igraph::E(graph)$arrow.size = 1  # Adjust the arrow size value as needed
    return(graph)
}

# Start with a random subgraph
Gamma = matrix(0, nrow = p, ncol = p)
Gamma[2, 1] = 1

# Plot the subgraph to get an idea about the causal network
plot(smaller_arrowheads(igraph::graph.adjacency(Gamma,
         mode = ”directed”)), layout = igraph::layout_in_circle,
            main = ”Subgraph”)


# Do uncertainty quantification for the subgraph
NetworkMotif(Gamma = Gamma, Gamma_Pst = Gamma_Pst)
```

RGM — *Fitting Bayesian Multivariate Bidirectional Mendelian Randomization Networks*

Description

The RGM function transforms causal inference by merging Mendelian randomization and network-based methods, enabling the creation of comprehensive causal graphs within complex biological systems. RGM accommodates varied data contexts with three input options: individual-level data (X, Y matrices), summary-level data including S_YY, S_YX, and S_XX matrices, and intricate data with challenging cross-correlations, utilizing S_XX, Beta, and Sigma_Hat matrices. For the latter input, data centralization is necessary. Users can select any of these data formats to suit their needs and don't have to specify all of them, allowing flexibility based on data availability. Crucial inputs encompass "d" (instrument count per response) and "n" (total observations, only required for summary level data), amplified by customizable parameters that refine analysis. Additionally, users can tailor the analysis by setting parameters such as "nIter" (number of MCMC iterations), "nBurnin" (number of discarded samples during burn-in for convergence), and "Thin" (thinning of posterior samples). These customizable parameters enhance the precision and relevance of the analysis. RGM provides essential causal effect/strength estimates between response variables and between response and instrument variables. Moreover, it furnishes adjacency matrices, visually mapping causal graph structures. These outputs empower researchers to untangle intricate relationships within biological networks, fostering a holistic understanding of complex systems.

Usage

```
RGM(
  X = NULL,
  Y = NULL,
  S_YY = NULL,
  S_YX = NULL,
  S_XX = NULL,
  Beta = NULL,
  Sigma_Hat = NULL,
  d,
  n,
  nIter = 10000,
  nBurnin = 2000,
  Thin = 1,
  prior = c("Threshold", "Spike and Slab"),
  a_rho = 3,
  b_rho = 1,
  nu_1 = 0.001,
  a_psi = 0.5,
  b_psi = 0.5,
  nu_2 = 1e-04,
  a_sigma = 0.01,
  b_sigma = 0.01,
  Prop_VarA = 0.01,
  Prop_VarB = 0.01
)
```

Arguments

`X` A matrix of dimension n * k. In this matrix, each row signifies a distinct observation, while each column represents a specific instrument variable. The default value is set to NULL.

`Y` A matrix of dimension n * p. In this matrix, each row corresponds to a specific observation, and each column pertains to a particular response variable. The default value is set to NULL.

`S_YY` A matrix of dimensions p * p. Here, "p" signifies the count of response variables. This matrix is derived through the operation t(Y) %*% Y / n, where "Y" denotes the response data matrix and "n" stands for the total number of observations.

`S_YX` A matrix of dimensions p * k. Here, "p" signifies the number of response variables, and "k" represents the count of instrument variables. This matrix is calculated using the operation t(Y) %*% X / n, where "Y" is the response data matrix, "X" is the instrument data matrix and "n" is the total number of observations.

`S_XX` A matrix of dimensions k * k. Here, "k" signifies the count of instrument variables. This matrix is derived through the operation t(X) %*% X / n, where "X" denotes the instrument data matrix and "n" stands for the total number of observations.

`Beta` A matrix of dimensions p * k. In this matrix, each row corresponds to a specific response variable, and each column pertains to a distinct instrument variable. Each entry within the matrix represents the regression coefficient of the individual response variable on the specific instrument variable. To use Beta as an input, ensure you centralize each column of Y i.e. response data matrix and X i.e. instrument data matrix before calculating Beta, S_XX, and Sigma_Hat.

`Sigma_Hat` A matrix of dimensions p * k. In this matrix, each row corresponds to a specific response variable, and each column pertains to an individual instrument variable. Each entry in this matrix represents the mean square error associated with regressing the particular response on the specific instrument variable. To employ Sigma_Hat as an input, ensure that you centralize each column of Y i.e. response data matrix and X i.e. instrument data matrix before calculating Beta, S_XX, and Sigma_Hat.

`d` A vector input with a length of p i.e. number of response variables. Each element within this vector is a positive integer denoting the count of instrument variables influencing a specific response variable. The sum of all elements in the vector should be equal to the total count of instrument variables, represented as k.

`n` A positive integer input representing the count of data points or observations in the dataset. This input is only required when summary level data is used as input.

`nIter` A positive integer input representing the number of MCMC (Markov Chain Monte Carlo) sampling iterations. The default value is set to 10,000.

`nBurnin` A non-negative integer input representing the number of samples to be discarded during the burn-in phase of MCMC sampling. It's important that nBurnin is less than nIter. The default value is set to 2000.

`Thin` A positive integer input denoting the thinning factor applied to posterior samples. Thinning reduces the number of samples retained from the MCMC process for efficiency. Thin should not exceed (nIter - nBurnin). The default value is set to 1.

`prior` A parameter representing the prior assumption on the graph structure. It offers two options: "Threshold" or "Spike and Slab". The default value is "Spike and Slab".

`a_rho` A positive scalar input representing the first parameter of a Beta distribution. The default value is set to 3.

`b_rho` A positive scalar input representing the second parameter of a Beta distribution. The default value is set to 1.

`nu_1` A positive scalar input representing the multiplication factor in the variance of the spike part in the spike and slab distribution of matrix A. The default value is set to 0.001.

`a_psi` A positive scalar input corresponding to the first parameter of a Beta distribution. The default value is set to 0.5.

`b_psi` A positive scalar input corresponding to the second parameter of a Beta distribution. The default value is set to 0.5.

`nu_2` A positive scalar input corresponding to the multiplication factor in the variance of the spike part in the spike and slab distribution of matrix B. The default value is set to 0.0001.

`a_sigma` A positive scalar input corresponding to the first parameter of an Inverse Gamma distribution, which is associated with the variance of the model. The default value is set to 0.01.

`b_sigma` A positive scalar input corresponding to the second parameter of an Inverse Gamma distribution, which is associated with the variance of the model. The default value is set to 0.01.

`Prop_VarA` A positive scalar input representing the variance of the normal distribution used for proposing terms within the A matrix. The default value is set to 0.01.

`Prop_VarB` A positive scalar input representing the variance of the normal distribution used for proposing terms within the B matrix. The default value is set to 0.01.

Value

`A_Est` A matrix of dimensions p * p, representing the estimated causal effects or strengths between the response variables.

`B_Est` A matrix of dimensions p * k, representing the estimated causal effects or strengths between the response variables and the instrument variables. Each row corresponds to a specific response variable, and each column corresponds to a particular instrument variable.

`zA_Est` A binary adjacency matrix of dimensions p * p, indicating the graph structure between the response variables. Each entry in the matrix represents the presence (1) or absence (0) of a causal link between the corresponding response variables.

`zB_Est` A binary adjacency matrix of dimensions p * k, illustrating the graph structure between the response variables and the instrument variables. Each row corresponds to a specific response variable, and each column corresponds to a particular instrument variable. The presence of a causal link is denoted by 1, while the absence is denoted by 0.

A0_Est A matrix of dimensions p * p, representing the estimated causal effects or strengths between response variables before thresholding. This output is particularly relevant for cases where the "Threshold" prior assumption is utilized.

B0_Est A matrix of dimensions p * k, representing the estimated causal effects or strengths between the response variables and the instrument variables before thresholding. This output is particularly relevant for cases where the "Threshold" prior assumption is utilized. Each row corresponds to a specific response variable, and each column corresponds to a particular instrument variable.

Gamma_Est A matrix of dimensions p * p, representing the estimated probabilities of edges between response variables in the graph structure. Each entry in the matrix indicates the probability of a causal link between the corresponding response variables.

Tau_Est A matrix of dimensions p * p, representing the estimated variances of causal interactions between response variables. Each entry in the matrix corresponds to the variance of the causal effect between the corresponding response variables.

Phi_Est A matrix of dimensions p * k, representing the estimated probabilities of edges between response and instrument variables in the graph structure. Each row corresponds to a specific response variable, and each column corresponds to a particular instrument variable.

Eta_Est A matrix of dimensions p * k, representing the estimated variances of causal interactions between response and instrument variables. Each row corresponds to a specific response variable, and each column corresponds to a particular instrument variable.

tA_Est A scalar value representing the estimated thresholding value of causal interactions between response variables. This output is relevant when using the "Threshold" prior assumption.

tB_Est A scalar value representing the estimated thresholding value of causal interactions between response and instrument variables. This output is applicable when using the "Threshold" prior assumption.

Sigma_Est A vector of length p, representing the estimated variances of each response variable. Each element in the vector corresponds to the variance of a specific response variable.

AccptA The percentage of accepted entries in the A matrix, which represents the causal interactions between response variables. This metric indicates the proportion of proposed changes that were accepted during the sampling process.

AccptB The percentage of accepted entries in the B matrix, which represents the causal interactions between response and instrument variables. This metric indicates the proportion of proposed changes that were accepted during the sampling process.

Accpt_tA The percentage of accepted thresholding values for causal interactions between response variables when using the "Threshold" prior assumption. This metric indicates the proportion of proposed thresholding values that were accepted during the sampling process.

Accpt_tB The percentage of accepted thresholding values for causal interactions between response and instrument variables when using the "Threshold" prior assumption.

This metric indicates the proportion of proposed thresholding values that were accepted during the sampling process.

LL_Pst A vector containing the posterior log-likelihoods of the model. Each element in the vector represents the log-likelihood of the model given the observed data and the estimated parameters.

Rho_Est A matrix of dimensions p * p, representing the estimated Bernoulli success probabilities of causal interactions between response variables when using the "Spike and Slab" prior assumption. Each entry in the matrix corresponds to the success probability of a causal interaction between the corresponding response variables.

Psi_Est A matrix of dimensions p * k, representing the estimated Bernoulli success probabilities of causal interactions between response and instrument variables when using the "Spike and Slab" prior assumption. Each row in the matrix corresponds to a specific response variable, and each column corresponds to a particular instrument variable.

Gamma_Pst An array containing the posterior samples of the network structure among the response variables.

References

Ni, Y., Ji, Y., & Müller, P. (2018). Reciprocal graphical models for integrative gene regulatory network analysis. *Bayesian Analysis*, **13(4)**, 1095-1110. doi:10.1214/17BA1087.

Examples

```
#' # ---------------------------------------------------------

# Example 1:
# Run RGM based on individual level data with Threshold prior based on the model Y = AY + BX + E

# Data Generation
set.seed(9154)

# Number of data points
n = 10000

# Number of response variables and number of instrument variables
p = 3
k = 4

# Initialize causal interaction matrix between response variables
A = matrix(sample(c(-0.1, 0.1), p^2, replace = TRUE), p, p)

# Diagonal entries of A matrix will always be 0
diag(A) = 0

# Make the network sparse
A[sample(which(A!=0), length(which(A!=0))/2)] = 0
```

```
# Initialize causal interaction matrix between response and instrument variables
B = matrix(0, p, k)

# Create d vector
d = c(2, 1, 1)


# Initialize m
m = 1

# Calculate B matrix based on d vector
for (i in 1:p) {

 # Update ith row of B
 B[i, m:(m + d[i] - 1)] = 1

 # Update m
 m = m + d[i]

}

Sigma = diag(p)

Mult_Mat = solve(diag(p) - A)

Variance = Mult_Mat %*% Sigma %*% t(Mult_Mat)

# Generate instrument data matrix
X = matrix(rnorm(n * k, 0, 1), nrow = n, ncol = k)

# Initialize response data matrix
Y = matrix(0, nrow = n, ncol = p)

# Generate response data matrix based on instrument data matrix
for (i in 1:n) {

    Y[i, ] = MASS::mvrnorm(n = 1, Mult_Mat %*% B %*% X[i, ], Variance)

}

# Define a function to create smaller arrowheads
smaller_arrowheads = function(graph) {
    igraph::E(graph)$arrow.size = 1  # Adjust the arrow size value as needed
    return(graph)
}

# Print true causal interaction matrices between response variables
# and between response and instrument variables
A
B


# Plot the true graph structure between response variables
```

```
plot(smaller_arrowheads(igraph::graph.adjacency(((A != 0) * 1),
 mode = ”directed”)), layout = igraph::layout_in_circle, main = ”True Graph”)

# Apply RGM on individual level data for Threshold Prior
Output = RGM(X = X, Y = Y, d = c(2, 1, 1), prior = ”Threshold”)

# Get the graph structure between response variables
Output$zA_Est

# Plot the estimated graph structure between response variables
plot(smaller_arrowheads(igraph::graph.adjacency(Output$zA_Est,
 mode = ”directed”)), layout = igraph::layout_in_circle, main = ”Estimated Graph”)

# Get the estimated causal strength matrix between response variables
Output$A_Est

# Get the graph structure between response and instrument variables
Output$zB_Est

# Get the estimated causal strength matrix between response and instrument variables
Output$B_Est

# Plot posterior log-likelihood
plot(Output$LL_Pst, type = 'l', xlab = ”Number of Iterations”, ylab = ”Log-likelihood”)




# -----------------------------------------------------------------
# Example 2:
# Run RGM based on summary level data with Spike and Slab prior based on the model Y = AY + BX + E

# Data Generation
set.seed(9154)

# Number of data points
n = 10000

# Number of response variables and number of instrument variables
p = 3
k = 4

# Initialize causal interaction matrix between response variables
A = matrix(sample(c(-0.1, 0.1), p^2, replace = TRUE), p, p)

# Diagonal entries of A matrix will always be 0
diag(A) = 0

# Make the network sparse
A[sample(which(A!=0), length(which(A!=0))/2)] = 0

# Initialize causal interaction matrix between response and instrument variables
B = matrix(0, p, k)
```

```
# Create d vector
d = c(2, 1, 1)


# Initialize m
m = 1

# Calculate B matrix based on d vector
for (i in 1:p) {

 # Update ith row of B
 B[i, m:(m + d[i] - 1)] = 1

 # Update m
 m = m + d[i]

}

Sigma = diag(p)

Mult_Mat = solve(diag(p) - A)

Variance = Mult_Mat %*% Sigma %*% t(Mult_Mat)

# Generate instrument data matrix
X = matrix(rnorm(n * k, 0, 1), nrow = n, ncol = k)

# Initialize response data matrix
Y = matrix(0, nrow = n, ncol = p)

# Generate response data matrix based on instrument data matrix
for (i in 1:n) {

    Y[i, ] = MASS::mvrnorm(n = 1, Mult_Mat %*% B %*% X[i, ], Variance)

}


# Calculate summary level data
S_YY = t(Y) %*% Y / n
S_YX = t(Y) %*% X / n
S_XX = t(X) %*% X / n


# Print true causal interaction matrices between response variables
# and between response and instrument variables
A
B

# Plot the true graph structure between response variables
plot(smaller_arrowheads(igraph::graph.adjacency(((A != 0) * 1),
 mode = "directed")), layout = igraph::layout_in_circle, main = "True Graph")
```

```
# Apply RGM on summary level data for Spike and Slab Prior
Output = RGM(S_YY = S_YY, S_YX = S_YX, S_XX = S_XX,
          d = c(2, 1, 1), n = 10000, prior = "Spike and Slab")

# Get the graph structure between response variables
Output$zA_Est

# Plot the estimated graph structure between response variables
plot(smaller_arrowheads(igraph::graph.adjacency(Output$zA_Est,
 mode = "directed")), layout = igraph::layout_in_circle, main = "Estimated Graph")

# Get the estimated causal strength matrix between response variables
Output$A_Est

# Get the graph structure between response and instrument variables
Output$zB_Est

# Get the estimated causal strength matrix between response and instrument variables
Output$B_Est

# Plot posterior log-likelihood
plot(Output$LL_Pst, type = 'l', xlab = "Number of Iterations", ylab = "Log-likelihood")




# --------------------------------------------------------------------
# Example 3:
# Run RGM based on Beta and Sigma_Hat with Spike and Slab prior based on the model Y = AY + BX + E

# Data Generation
set.seed(9154)

# Number of datapoints
n = 10000

# Number of response variables and number of instrument variables
p = 3
k = 4

# Initialize causal interaction matrix between response variables
A = matrix(sample(c(-0.1, 0.1), p^2, replace = TRUE), p, p)

# Diagonal entries of A matrix will always be 0
diag(A) = 0

# Make the network sparse
A[sample(which(A!=0), length(which(A!=0))/2)] = 0

# Initialize causal interaction matrix between response and instrument variables
B = matrix(0, p, k)
```

```
# Create d vector
d = c(2, 1, 1)


# Initialize m
m = 1

# Calculate B matrix based on d vector
for (i in 1:p) {

 # Update ith row of B
 B[i, m:(m + d[i] - 1)] = 1

 # Update m
 m = m + d[i]

}

Sigma = diag(p)

Mult_Mat = solve(diag(p) - A)

Variance = Mult_Mat %*% Sigma %*% t(Mult_Mat)

# Generate DNA expressions
X = matrix(rnorm(n * k, 0, 1), nrow = n, ncol = k)

# Initialize response data matrix
Y = matrix(0, nrow = n, ncol = p)

# Generate response data matrix based on instrument data matrix
for (i in 1:n) {

    Y[i, ] = MASS::mvrnorm(n = 1, Mult_Mat %*% B %*% X[i, ], Variance)

}


# Centralize Data
Y = t(t(Y) - colMeans(Y))
X = t(t(X) - colMeans(X))

# Calculate S_XX
S_XX = t(X) %*% X / n

# Generate Beta matrix and Sigma_Hat
Beta = matrix(0, nrow = p, ncol = k)
Sigma_Hat = matrix(0, nrow = p, ncol = k)

for (i in 1:p) {

   for (j in 1:k) {
       fit = lm(Y[, i] ~ X[, j])

       Beta[i, j] =  fit$coefficients[2]

       Sigma_Hat[i, j] = sum(fit$residuals^2) / n

       }

   }


# Print true causal interaction matrices between response variables
# and between response and instrument variables
A
B


# Plot the true graph structure between response variables
plot(smaller_arrowheads(igraph::graph.adjacency(((A != 0) * 1),
 mode = "directed")), layout = igraph::layout_in_circle, main = "True Graph")


# Apply RGM based on S_XX, Beta and Sigma_Hat for Spike and Slab Prior
Output = RGM(S_XX = S_XX, Beta = Beta, Sigma_Hat = Sigma_Hat,
          d = c(2, 1, 1), n = 10000, prior = "Spike and Slab")

# Get the graph structure between response variables
Output$zA_Est

# Plot the estimated graph structure between response variables
plot(smaller_arrowheads(igraph::graph.adjacency(Output$zA_Est,
 mode = "directed")), layout = igraph::layout_in_circle, main = "Estimated Graph")

# Get the estimated causal strength matrix between response variables
Output$A_Est

# Get the graph structure between response and instrument variables
Output$zB_Est

# Get the estimated causal strength matrix between response and instrument variables
Output$B_Est

# Plot posterior log-likelihood
plot(Output$LL_Pst, type = 'l', xlab = "Number of Iterations", ylab = "Log-likelihood")
```

Index