Title   High Dimensional Bayesian Mediation Analysis
Version 1.3.0
URL https://github.com/umich-cphds/bama
BugReports https://github.com/umich-cphds/bama/issues
Description Perform mediation analysis in the presence of high-dimensional mediators based on the potential outcome framework. Bayesian Mediation Analysis (BAMA), developed by Song et al (2019) <doi:10.1111/biom.13189> and Song et al (2020) <arXiv:2009.11409>, relies on two Bayesian sparse linear mixed models to simultaneously analyze a relatively large number of mediators for a continuous exposure and outcome assuming a small number of mediators are truly active. This sparsity assumption also allows the extension of univariate mediator analysis by casting the identification of active mediators as a variable selection problem and applying Bayesian methods with continuous shrinkage priors on the effects.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.2.1
LinkingTo Rcpp, RcppArmadillo, RcppDist, BH
Imports Rcpp, parallel
Depends R (>= 3.5)
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation yes
Author Alexander Rix [aut],
        Mike Kleinsasser [aut, cre],
        Yanyi Song [aut]
Maintainer Mike Kleinsasser <mkleinsa@umich.edu>
Repository CRAN
Date/Publication 2023-01-24 18:10:02 UTC
**R topics documented:**

- `bama` ...................................................... 2
- `bama.data` .................................................. 6
- `fdr.bama` .................................................. 7
- `print.bama` ................................................ 9
- `print.fdr.bama` .......................................... 10
- `summary.bama` ............................................ 10
- `summary.fdr.bama` ....................................... 11

**Index** ................................................. 12

---

### bama

**Bayesian Mediation Analysis**

#### Description

`bama` is a Bayesian inference method that uses continuous shrinkage priors for high-dimensional Bayesian mediation analysis, developed by Song et al (2019, 2020). `bama` provides estimates for the regression coefficients as well as the posterior inclusion probability for ranking mediators.

#### Usage

```r
bama(
  Y,
  A,
  M,
  C1,
  C2,
  method,
  burnin,
  ndraws,
  weights = NULL,
  inits = NULL,
  control = list(k = 2, lm0 = 1e-04, lm1 = 1, lma1 = 1, l = 1, lambda0 = 0.04, lambda1 = 0.2, lambda2 = 0.2, phi0 = 0.01, phi1 = 0.01, a0 = 0.01 * ncol(M), a1 = 0.05 * ncol(M), a2 = 0.05 * ncol(M), a3 = 0.89 * ncol(M)),
  seed = NULL
)
```

#### Arguments

- **Y** Length n numeric outcome vector
- **A** Length n numeric exposure vector
- **M** n x p numeric matrix of mediators of Y and A
- **C1** n x nc1 numeric matrix of extra covariates to include in the outcome model
- **C2** n x nc2 numeric matrix of extra covariates to include in the mediator model
method String indicating which method to use. Options are
  • "BSLMM" - mixture of two normal components; Song et al. 2019
  • "PTG" - product threshold Gaussian prior; Song et al. 2020
  • "GMM" - NOTE: GMM not currently supported. Instead, use method = 'PTG'. four-component Gaussian mixture prior; Song et al. 2020

burnin number of iterations to run the MCMC before sampling

ndraws number of draws to take from MCMC (includes burnin draws)

weights Length n numeric vector of weights

inits list of initial values for the Gibbs sampler. Options are
  • beta.m - Length p numeric vector of initial beta.m in the outcome model. See details for equation
  • alpha.a - Length p numeric vector of initial alpha.a in the mediator model. See details for equation

control list of Gibbs algorithm control options. These include prior and hyper-prior parameters. Options vary by method selection. If method = "BSLMM"
  • k - Shape parameter prior for inverse gamma
  • lm0 - Scale parameter prior for inverse gamma for the small normal components
  • lm1 - Scale parameter prior for inverse gamma for the large normal components of beta.m
  • lma1 - Scale parameter prior for inverse gamma for the large normal component of alpha.a
  • l - Scale parameter prior for the other inverse gamma distributions
If method = "PTG"
  • lambda0 - threshold parameter for product of alpha.a and beta.m effect
  • lambda1 - threshold parameter for beta.m effect
  • lambda2 - threshold parameter for alpha.a effect
  • ha - inverse gamma shape prior for sigma.sq.a
  • la - inverse gamma scale prior for sigma.sq.a
  • h1 - inverse gamma shape prior for sigma.sq.e
  • l1 - inverse gamma scale prior for sigma.sq.e
  • h2 - inverse gamma shape prior for sigma.sq.g
  • l2 - inverse gamma scale prior for sigma.sq.g
  • km - inverse gamma shape prior for tau.sq.b
  • lm - inverse gamma scale prior for tau.sq.b
  • kma - inverse gamma shape prior for tau.sq.a
  • lma - inverse gamma scale prior for tau.sq.a
If method = "GMM". NOTE: GMM not currently supported. Instead, use method = 'PTG'.
  • phi0 - prior beta.m variance
  • phi1 - prior alpha.a variance
Details

`bama` uses two regression models for the two conditional relationships, $Y | A, M, C$ and $M | A, C$.
For the outcome model, `bama` uses

$$Y = M \beta_M + A \beta_A + C \beta_C + \epsilon_Y$$

For the mediator model, `bama` uses the model

$$M = A \alpha_A + C \alpha_C + \epsilon_M$$

For high dimensional tractability, `bama` employs continuous Bayesian shrinkage priors to select mediators and makes the two following assumptions: First, it assumes that all the potential mediators contribute small effects in mediating the exposure-outcome relationship. Second, it assumes that only a small proportion of mediators exhibit large effects ("active" mediators). `bama` uses a Metropolis-Hastings within Gibbs MCMC to generate posterior samples from the model.

NOTE: GMM not currently supported. Instead, use method = ‘PTG’.

Value

If method = "BSLMM", then `bama` returns a object of type "bama" with 12 elements:

- `beta.m` ndraws x p matrix containing outcome model mediator coefficients.
- `r1` ndraws x p matrix indicating whether or not each beta.m belongs to the larger normal component (1) or smaller normal component (0).
- `alpha.a` ndraws x p matrix containing the mediator model exposure coefficients.
- `r3` ndraws x p matrix indicating whether or not each alpha.a belongs to the larger normal component (1) or smaller normal component (0).
- `beta.a` Vector of length ndraws containing the beta.a coefficient.
- `pi.m` Vector of length ndraws containing the proportion of non zero beta.m coefficients.
- `pi.a` Vector of length ndraws containing the proportion of non zero alpha.a coefficients.
- `sigma.m0` Vector of length ndraws containing the standard deviation of the smaller normal component for mediator-outcome coefficients (beta.m).
\textbf{bama}  

\textbf{sigma.m1} Vector of length ndraws containing standard deviation of the larger normal component for mediator-outcome coefficients (beta.m).

\textbf{sigma.m0} Vector of length ndraws containing standard deviation of the smaller normal component for exposure-mediator coefficients (alpha.a).

\textbf{sigma.m1} Vector of length ndraws containing standard deviation of the larger normal component for exposure-mediator coefficients (alpha.a).

\textbf{call} The R call that generated the output.

NOTE: GMM not currently supported. Instead, use method = 'PTG' If method = "GMM", then bama returns a object of type "bama" with:

\textbf{beta.m} ndraws x p matrix containing outcome model mediator coefficients.

\textbf{alpha.a} ndraws x p matrix containing the mediator model exposure coefficients.

\textbf{beta.member} ndraws x p matrix of 1's and 0's where item = 1 only if beta.m is non-zero.

\textbf{alphaa_member} ndraws x p matrix of 1's and 0's where item = 1 only if alpha.a is non-zero.

\textbf{alpha.c} ndraws x (q2 + p) matrix containing alpha_c coefficients. Since alpha.c is a matrix of dimension q2 x p, the draws are indexed as alpha_c(w, j) = w * p + j

\textbf{beta.c} ndraws x q1 matrix containing beta_c coefficients. Since beta.c is a matrix of dimension q1 x p

\textbf{beta.a} Vector of length ndraws containing the beta.a coefficient.

\textbf{sigma.sq.a} Vector of length ndraws variance of beta.a effect

\textbf{sigma.sq.e} Vector of length ndraws variance of outcome model error

\textbf{sigma.sq.g} Vector of length ndraws variance of mediator model error

If method = "PTG", then bama returns a object of type "bama" with:

\textbf{beta.m} ndraws x p matrix containing outcome model mediator coefficients.

\textbf{alpha.c} ndraws x (q2 + p) matrix containing alpha_c coefficients. Since alpha.c is a matrix of dimension q2 x p, the draws are indexed as alpha_c(w, j) = w * p + j

\textbf{beta.c} ndraws x q1 matrix containing beta_c coefficients. Since beta.c is a matrix of dimension q1 x p

\textbf{beta.member} ndraws x p matrix of 1's and 0's where item = 1 only if beta.m is non-zero.

\textbf{alphaa_member} ndraws x p matrix of 1's and 0's where item = 1 only if alpha.a is non-zero.

\textbf{beta.a} Vector of length ndraws containing the beta.a coefficient.

\textbf{sigma.sq.a} Vector of length ndraws variance of beta.a effect

\textbf{sigma.sq.e} Vector of length ndraws variance of outcome model error

\textbf{sigma.sq.g} Vector of length ndraws variance of mediator model error

\textbf{References}


Examples

```r
library(bama)

Y <- bama.data$y
A <- bama.data$a

# grab the mediators from the example data.frame
M <- as.matrix(bama.data[, paste0("m", 1:100)], nrow(bama.data))

# We just include the intercept term in this example as we have no covariates
C1 <- matrix(1, 1000, 1)
C2 <- matrix(1, 1000, 1)
beta.m <- rep(0, 100)
alpha.a <- rep(0, 100)

out <- bama(Y = Y, A = A, M = M, C1 = C1, C2 = C2, method = "BSLMM", seed = 1234,
            burnin = 100, ndraws = 110, weights = NULL, inits = NULL,
            control = list(k = 2, lm0 = 1e-04, lm1 = 1, lma1 = 1, l = 1))

# The package includes a function to summarise output from 'bama'
summary <- summary(out)
head(summary)

# Product Threshold Gaussian
ptgmod = bama(Y = Y, A = A, M = M, C1 = C1, C2 = C2, method = "PTG", seed = 1234,
              burnin = 100, ndraws = 110, weights = NULL, inits = NULL,
              control = list(lambda0 = 0.04, lambda1 = 0.2, lambda2 = 0.2))

mean(ptgmod$beta.a)
apply(ptgmod$beta.m, 2, mean)
apply(ptgmod$alpha.a, 2, mean)
apply(ptgmod$beta_member, 2, mean)
apply(ptgmod$alpha_member, 2, mean)
```

bama.data

*Synthetic example data for bama*

**Description**

Synthetic example data for bama

**Usage**

bama.data
**fdr.bama**

**Format**

A data.frame with 1000 observations on 102 variables:

- **y**: Numeric response variable.
- **a**: Numeric exposure variable.
- **m[1-100]**: Numeric mediator variables

**Description**

*fdr.bama* uses the permutation test to estimate the null PIP distribution for each mediator and determines a threshold (based off of the *fdr* parameter) for significance.

**Usage**

```r
fdr.bama(
  Y,  # Length n numeric outcome vector
  A,  # Length n numeric exposure vector
  M,  # n x p numeric matrix of mediators of Y and A
  C1,  # n x nc1 numeric matrix of extra covariates to include in the outcome model
  C2,  #
  beta.m,  #
  alpha.a,  #
  burnin,  #
  ndraws,  #
  weights = NULL,  #
  npermutations = 200,  #
  fdr = 0.1,  #
  k = 2,  #
  lm0 = 1e-04,  #
  lm1 = 1,  #
  lma1 = 1,  #
  l = 1,  #
  mc.cores = 1,  #
  type = "PSOCK"  #
)
```

**Arguments**

- **Y**: Length \( n \) numeric outcome vector
- **A**: Length \( n \) numeric exposure vector
- **M**: \( n \times p \) numeric matrix of mediators of \( Y \) and \( A \)
- **C1**: \( n \times nc1 \) numeric matrix of extra covariates to include in the outcome model
fdr.bama

C2  \[ n \times nc2 \] numeric matrix of extra covariates to include in the mediator model

beta.m  Length \( p \) numeric vector of initial \( \beta_m \) in the outcome model

alpha.a  Length \( p \) numeric vector of initial \( \alpha_a \) in the mediator model

burnin  Number of iterations to run the MCMC before sampling

ndraws  Number of draws to take from MCMC after the burnin period

weights  Length \( n \) numeric vector of weights

npermutations  The number of permutations to generate while estimating the null pip distribution. Default is 200

fdr  False discovery rate. Default is 0.1

k  Shape parameter prior for inverse gamma. Default is 2.0

lm0  Scale parameter prior for inverse gamma for the small normal components. Default is 1e-4

lm1  Scale parameter prior for inverse gamma for the large normal component of \( \beta_m \). Default is 1.0

lma1  Scale parameter prior for inverse gamma for the large normal component of \( \alpha_a \). Default is 1.0

l  Scale parameter prior for the other inverse gamma distributions. Default is 1.0

mc.cores  The number of cores to use while running \( \text{fdr.bama} \). \( \text{fdr.bama} \) uses the \text{parallel} package for parallelization, so see that for more information. Default is 1 core

type  Type of cluster to make when \( \text{mc.cores} > 1 \). See \text{makeCluster} in the \text{parallel} package for more details. Default is "PSOCK"

Value

\text{fdr.bama} returns a object of type "fdr.bama" with 5 elements:

\text{bama.out}  Output from the \text{bama} run.

\text{pip.null}  A \( p \times npermutations \) matrices containing the estimated null PIP distribution for each mediator.

\text{threshold}  The cutoff significance threshold for each PIP controlling for the false discovery rate.

\text{fdr}  The false discovery rate used to calculate \text{threshold}.

\text{call}  The R call that generated the output.

Author(s)

Alexander Rix

References

Examples

```r
library(bama)

Y <- bama.data$y
A <- bama.data$a

# grab the mediators from the example data.frame
M <- as.matrix(bama.data[, paste0("m", 1:100)], nrow(bama.data))

# We just include the intercept term in this example as we have no covariates
C1 <- matrix(1, 1000, 1)
C2 <- matrix(1, 1000, 1)
beta.m <- rep(0, 100)
alpha.a <- rep(0, 100)

set.seed(12345)

out <- fdr.bama(Y, A, M, C1, C2, beta.m, alpha.a, burnin = 100,
    ndraws = 120, npermutations = 10)

# The package includes a function to summarise output from 'fdr.bama'
summary(out)
```

print.bama

## Printing bama objects

**Description**

Print a bama object.

**Usage**

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

**Arguments**

- `x` An object of class 'bama'.
- `...` Additional arguments to pass to print.data.frame or summary.bama
print.fdr.bama  

Printing bama objects

Description

Print a bama object.

Usage

```r
## S3 method for class 'fdr.bama'
print(x, ...)
```

Arguments

- `x` An object of class 'bama'.
- `...` Additional arguments to pass to print.data.frame or summary.bama

summary.bama  

Summarize objects of type “bama”

Description

summary.bama summarizes the 'beta.m' estimates from bama and generates an overall estimate, credible interval, and posterior inclusion probability.

Usage

```r
## S3 method for class 'bama'
summary(object, rank = FALSE, ci = c(0.025, 0.975), ...)
```

Arguments

- `object` An object of class "bama".
- `rank` Whether or not to rank the output by posterior inclusion probability. Default is TRUE.
- `ci` The credible interval to calculate. `ci` should be a length 2 numeric vector specifying the upper and lower bounds of the CI. By default, `ci = c(0.025, 0.975)`. Additional optional arguments to summary

Value

A data.frame with 4 elements. The beta.m estimates, the estimates’ credible interval (which by default is 95\% inclusion probability (pip) of each 'beta.m').
Summary

`summary.fdr.bama` summarizes the `beta.m` estimates from `fdr.bama` and for each mediator generates an overall estimate, credible interval, posterior inclusion probability (PIP), and PIP threshold for significance controlling for the specified false discovery rate (FDR).

Usage

```r
## S3 method for class 'fdr.bama'
summary(
  object,
  rank = F,
  ci = c(0.025, 0.975),
  fdr = object$fdr,
  filter = T,
  ...)
```

Arguments

- `object`: An object of class "bama".
- `rank`: Whether or not to rank the output by posterior inclusion probability. Default is TRUE.
- `ci`: The credible interval to calculate. `ci` should be a length 2 numeric vector specifying the upper and lower bounds of the CI. By default, `ci = c(0.025, 0.975)`.
- `fdr`: False discovery rate. By default, it is set to whatever the `fdr` of `object` is. However, it can be changed to recalculate the PIP cutoff threshold.
- `filter`: Whether or not to filter out mediators with PIP less than the PIP threshold.
- `...`: Additional optional arguments to `summary`

Value

A data.frame with 4 elements. The `beta.m` estimates, the estimates' credible interval (which by default is 95\% inclusion probability (pip)) of each 'beta.m'.
Index

* datasets
  bama.data, 6

bama, 2
bama.data, 6

fdr.bama, 7

print.bama, 9
print.fdr.bama, 10

summary.bama, 10
summary.fdr.bama, 11