

Package: buzzMed (via r-universe)

May 28, 2026

Version 0.1.2

Date 2026-05-20

Title Bayesian Understanding for Mediator Selection Framework

Description A collection of quantitative tools for selecting mediating effects within exploratory Bayesian mediation models. The package accommodates both continuous and dichotomous outcomes, including the dependent variables and the mediators for identifying and analyzing mediation pathways.

License GPL (>= 3)

Depends R (>= 4.0.0)

SystemRequirements JAGS (>= 4.0.0)

Imports coda, methods, rjags, stats, utils

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

ByteCompile true

VignetteBuilder knitr

URL <https://github.com/olfactorybulb/buzzMed>

BugReports <https://github.com/olfactorybulb/buzzMed/issues>

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

NeedsCompilation no

Author Dingjing Shi [aut, cre], Tansu Celikel [aut], Dexin Shi [aut],
Chih-Chia Hsing [ctb]

Maintainer Dingjing Shi <dshi32@gatech.edu>

Config/pak/sysreqs jags

Repository <https://cran.r-universe.dev>

Date/Publication 2026-05-28 12:47:42 UTC

RemoteUrl <https://github.com/cran/buzzMed>

RemoteRef HEAD

RemoteSha 88f4767739baac2a42bfc69c97b7f29cfce36ef3

Contents

buzzEBMcatMcatY	2
buzzEBMcatMcontY	4
buzzEBMcontMcatY	7
buzzEBMcontMcontY	10
run_parms_wizard	12

Index	15
--------------	-----------

buzzEBMcatMcatY	<i>Fit Bayesian Mediation Model (Binary M & Binary Y)</i>
-----------------	---

Description

This function implements variable selection for mediators when dealing with binary data. It utilizes a latent probit link formulation to model the probabilities of the binary responses within the JAGS framework. It automates data preparation, JAGS model construction, and MCMC sampling.

Usage

```

buzzEBMcatMcatY(
  model,
  dataset,
  my_prior = NULL,
  advanced = NULL,
  a.coef.mean = NULL,
  a.coef.prec = NULL,
  b.coef.mean = NULL,
  b.coef.prec = NULL,
  a.pip.hyperalpha = NULL,
  a.pip.hyperbeta = NULL,
  b.pip.hyperalpha = NULL,
  b.pip.hyperbeta = NULL,
  direct.coef.mean = NULL,
  direct.coef.precision = NULL,
  direct.coef.init = NULL,
  a.pip.hyperprior.init = NULL,
  b.pip.hyperprior.init = NULL,
  n_chains = NULL,
  n_adapt = NULL,
  n_burnin = NULL,
  n_iter = NULL,
  thin = NULL
)

```

Arguments

<code>model</code>	A description of the model to be fitted. This is typically a formula or a character string using lavaan syntax (e.g., $Y \sim M + X$).
<code>dataset</code>	A data.frame containing the variables specified in the model.
<code>my_prior</code>	Optional data.frame containing custom prior specifications. Run <code>parms <- run_parms_wizard()</code> to see the required structure.
<code>advanced</code>	Character. Use "interactive" for an interactive wizard to choose parameter distributions, or leave NULL for defaults.
<code>a.coef.mean</code> , <code>a.coef.prec</code>	Numeric scalar or vector. Mean and precision parameters of the Normal prior for the <i>a</i> path effects. By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>b.coef.mean</code> , <code>b.coef.prec</code>	Numeric scalar or vector. Mean and precision parameters of the Normal prior for the <i>b</i> path effects. By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>a.pip.hyperalpha</code> , <code>a.pip.hyperbeta</code>	Numeric scalar or vector. Alpha and beta parameters for the Beta hyperprior of the <i>a</i> path inclusion probabilities. By default, a Beta distribution is used. The default value is 3.
<code>b.pip.hyperalpha</code> , <code>b.pip.hyperbeta</code>	Numeric scalar or vector. Alpha and beta parameters for the Beta hyperprior of the <i>b</i> path inclusion probabilities. By default, a Beta distribution is used. The default value is 3.
<code>direct.coef.mean</code> , <code>direct.coef.precision</code>	Numeric scalar or vector. Mean and precision parameters for the Normal prior of the direct effects (<i>c'</i>). By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>direct.coef.init</code>	Numeric or NULL. Initial value for the direct effect (<i>c'</i>). Default is 0.
<code>a.pip.hyperprior.init</code> , <code>b.pip.hyperprior.init</code>	Numeric or NULL. Initial inclusion probabilities (PIP). Default is 0.5.
<code>n_chains</code>	Integer. Number of MCMC chains.
<code>n_adapt</code>	Integer. Number of adaptation iterations.
<code>n_burnin</code>	Integer. Number of burn-in iterations.
<code>n_iter</code>	Integer. Number of post-burn-in iterations.
<code>thin</code>	Integer. Thinning interval.

Details

Fits a Bayesian mediation model specifically designed for cases where both the mediators (*M*) and the outcome variable (*Y*) are binary (0/1).

Since both *M* and *Y* are binary, residual precisions are not estimated (they are fixed to 1 in the latent probit space). This function identifies *X*, *M*, *Y* via `.parse_buzz_syntax` and routes the data through an internal pipeline including `prepare_ebmed_data` and `build_ebmed_model_mcat_ycat`.

Value

An object of class `mcmc.list` containing posterior samples.

Note

This function is strictly for binary data. For variables with more than two levels, please convert them into dummy variables or ensure they are binary before running.

References

Shi, D., Shi, D., and Fairchild, A. J. (2023) "Variable Selection for Mediators under a Bayesian Mediation Model" <doi:10.1080/10705511.2022.2164285>

Examples

```
# Binary case: Both M and Y are Binary
set.seed(101)
n <- 100
toy_data <- data.frame(
  X = rbinom(n, 1, 0.5),
  M1 = rbinom(n, 1, 0.3),
  M2 = rbinom(n, 1, 0.7),
  Y = rbinom(n, 1, 0.5)
)

# Fit the model
results <- buzzEBMcatMcatY(
  model = "Y ~ M1 + M2 | X",
  dataset = toy_data,
  n_burnin = 200,
  n_iter = 1000
)

summary(results)
```

buzzEBMcatMcontY

Fit Bayesian Mediation Model (Binary M & Continuous Y)

Description

This function implements variable selection for mediators in a mixed data framework. It utilizes latent variable formulations for the binary mediators and a Gaussian likelihood for the continuous outcome. It automates data preparation, JAGS model construction, and MCMC sampling.

Usage

```

buzzEBMcatMcontY(
  model,
  dataset,
  my_prior = NULL,
  advanced = NULL,
  y.prec.shape = NULL,
  y.prec.rate = NULL,
  a.coef.mean = NULL,
  a.coef.prec = NULL,
  b.coef.mean = NULL,
  b.coef.prec = NULL,
  a.pip.hyperalpha = NULL,
  a.pip.hyperbeta = NULL,
  b.pip.hyperalpha = NULL,
  b.pip.hyperbeta = NULL,
  direct.coef.mean = NULL,
  direct.coef.precision = NULL,
  y.prec.init = NULL,
  direct.coef.init = NULL,
  a.pip.hyperprior.init = NULL,
  b.pip.hyperprior.init = NULL,
  n_chains = NULL,
  n_adapt = NULL,
  n_burnin = NULL,
  n_iter = NULL,
  thin = NULL
)

```

Arguments

<code>model</code>	A description of the model to be fitted. This is typically a formula or a character string using lavaan syntax (e.g., $Y \sim M + X$).
<code>dataset</code>	A data.frame containing the variables specified in the model.
<code>my_prior</code>	Optional data.frame containing custom prior specifications. Run <code>parms <- run_parms_wizard()</code> to see the required structure.
<code>advanced</code>	Character. Use "interactive" for an interactive wizard to choose parameter distributions, or leave NULL for defaults.
<code>y.prec.shape</code> , <code>y.prec.rate</code>	Numeric scalar. Shape and rate parameters of the Gamma hyperprior for the outcome residual precision. By default, a Gamma distribution is used. The default values are 1 and 0.001, respectively.
<code>a.coef.mean</code> , <code>a.coef.prec</code>	Numeric scalar or vector. Mean and precision parameters of the Normal prior for the <i>a</i> path effects. By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.

<code>b.coef.mean</code> , <code>b.coef.prec</code>	Numeric scalar or vector. Mean and precision parameters of the Normal prior for the b path effects. By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>a.pip.hyperalpha</code> , <code>a.pip.hyperbeta</code>	Numeric scalar or vector. Alpha and beta parameters for the Beta hyperprior of the a path inclusion probabilities. By default, a Beta distribution is used. The default value is 3.
<code>b.pip.hyperalpha</code> , <code>b.pip.hyperbeta</code>	Numeric scalar or vector. Alpha and beta parameters for the Beta hyperprior of the b path inclusion probabilities. By default, a Beta distribution is used. The default value is 3.
<code>direct.coef.mean</code> , <code>direct.coef.precision</code>	Numeric scalar or vector. Mean and precision parameters for the Normal prior of the direct effects (c'). By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>y.prec.init</code>	Numeric or NULL. Initial values for residual precisions. Default is 1.
<code>direct.coef.init</code>	Numeric or NULL. Initial value for the direct effect (c'). Default is 0.
<code>a.pip.hyperprior.init</code> , <code>b.pip.hyperprior.init</code>	Numeric or NULL. Initial inclusion probabilities (PIP). Default is 0.5.
<code>n_chains</code>	Integer. Number of MCMC chains.
<code>n_adapt</code>	Integer. Number of adaptation iterations.
<code>n_burnin</code>	Integer. Number of burn-in iterations.
<code>n_iter</code>	Integer. Number of post-burn-in iterations.
<code>thin</code>	Integer. Thinning interval.

Details

Fits a Bayesian mediation model specifically designed for cases where the mediators (M) are binary (0/1) and the outcome variable (Y) is continuous.

This function is a specific "worker" function. It identifies X, M, Y via `.parse_buzz_syntax`. While Y is modeled with residual precision parameters, the mediators M are binary and thus their residual precisions are fixed to 1 within the latent variable framework.

Value

An object of class `mcmc.list` containing posterior samples.

References

Shi, D., Shi, D., and Fairchild, A. J. (2023) "Variable Selection for Mediators under a Bayesian Mediation Model" <doi:10.1080/10705511.2022.2164285>

Examples

```

# Mixed case: Binary M, Continuous Y
set.seed(789)
n <- 100
toy_data <- data.frame(
  X = rnorm(n),
  M1 = rbinom(n, 1, 0.4),
  M2 = rbinom(n, 1, 0.6),
  Y = rnorm(n)
)

# Fit the model
results <- buzzEBMcatMcontY(
  model = "Y ~ M1 + M2 | X",
  dataset = toy_data,
  n_burnin = 200,
  n_iter = 1000
)

summary(results)

```

buzzEBMcontMcatY

Fit Bayesian Mediation Model (Continuous M & Binary Y)

Description

This function implements variable selection for mediators in a mixed data framework. It assumes a Gaussian likelihood for the mediators and a latent probit link for the binary outcome. It automates data preparation, JAGS model construction, and MCMC sampling.

Usage

```

buzzEBMcontMcatY(
  model,
  dataset,
  my_prior = NULL,
  advanced = NULL,
  m.prec.shape = NULL,
  m.prec.rate = NULL,
  a.coef.mean = NULL,
  a.coef.prec = NULL,
  b.coef.mean = NULL,
  b.coef.prec = NULL,
  a.pip.hyperalpha = NULL,
  a.pip.hyperbeta = NULL,
  b.pip.hyperalpha = NULL,
  b.pip.hyperbeta = NULL,

```

```

direct.coef.mean = NULL,
direct.coef.precision = NULL,
m.prec.init = NULL,
direct.coef.init = NULL,
a.pip.hyperprior.init = NULL,
b.pip.hyperprior.init = NULL,
n_chains = NULL,
n_adapt = NULL,
n_burnin = NULL,
n_iter = NULL,
thin = NULL
)

```

Arguments

<code>model</code>	A description of the model to be fitted. This is typically a formula or a character string using lavaan syntax (e.g., $Y \sim M + X$).
<code>dataset</code>	A <code>data.frame</code> containing the variables specified in the model.
<code>my_prior</code>	Optional <code>data.frame</code> containing custom prior specifications. Run <code>parms <- run_parms_wizard()</code> to see the required structure.
<code>advanced</code>	Character. Use "interactive" for an interactive wizard to choose parameter distributions, or leave NULL for defaults.
<code>m.prec.shape</code> , <code>m.prec.rate</code>	Numeric scalar or vector of length equal to the number of mediators. Shape and rate parameters of the Gamma hyperprior for the mediator residual precisions. By default, a Gamma distribution is used. The default values are 1 and 0.001, respectively.
<code>a.coef.mean</code> , <code>a.coef.prec</code>	Numeric scalar or vector. Mean and precision parameters of the Normal prior for the <i>a</i> path effects. By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>b.coef.mean</code> , <code>b.coef.prec</code>	Numeric scalar or vector. Mean and precision parameters of the Normal prior for the <i>b</i> path effects. By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>a.pip.hyperalpha</code> , <code>a.pip.hyperbeta</code>	Numeric scalar or vector. Alpha and beta parameters for the Beta hyperprior of the <i>a</i> path inclusion probabilities. By default, a Beta distribution is used. The default value is 3.
<code>b.pip.hyperalpha</code> , <code>b.pip.hyperbeta</code>	Numeric scalar or vector. Alpha and beta parameters for the Beta hyperprior of the <i>b</i> path inclusion probabilities. By default, a Beta distribution is used. The default value is 3.
<code>direct.coef.mean</code> , <code>direct.coef.precision</code>	Numeric scalar or vector. Mean and precision parameters for the Normal prior of the direct effects (<i>c'</i>). By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.

m.prec.init	Numeric or NULL. Initial values for residual precisions. Default is 1.
direct.coef.init	Numeric or NULL. Initial value for the direct effect (c'). Default is 0.
a.pip.hyperprior.init, b.pip.hyperprior.init	Numeric or NULL. Initial inclusion probabilities (PIP). Default is 0.5.
n_chains	Integer. Number of MCMC chains.
n_adapt	Integer. Number of adaptation iterations.
n_burnin	Integer. Number of burn-in iterations.
n_iter	Integer. Number of post-burn-in iterations.
thin	Integer. Thinning interval.

Details

Fits a Bayesian mediation model specifically designed for cases where the mediators (M) are continuous and the outcome variable (Y) is binary (0/1).

This function is a specific "worker" function. It identifies X, M, Y via `.parse_buzz_syntax`. While M is modeled with residual precision parameters, Y is binary and thus its residual precision is fixed to 1 within the latent variable framework.

Value

An object of class `mcmc.list` containing posterior samples.

References

Shi, D., Shi, D., and Fairchild, A. J. (2023) "Variable Selection for Mediators under a Bayesian Mediation Model" <doi:10.1080/10705511.2022.2164285>

Examples

```
# Mixed case: Continuous M, Binary Y
set.seed(456)
n <- 100
toy_data <- data.frame(
  X = rnorm(n),
  M1 = rnorm(n),
  M2 = rnorm(n),
  Y = rbinom(n, 1, 0.5)
)

# Fit the model
results <- buzzEBMcontMcatY(
  model = "Y ~ M1 + M2 | X",
  dataset = toy_data,
  n_burnin = 200,
  n_iter = 1000
)

summary(results)
```

buzzEBMcontMcontY *Fit Bayesian Mediation Model (Continuous M & Continuous Y)*

Description

This function implements variable selection for mediators in a fully continuous framework. It automates data preparation, JAGS model construction using Gaussian likelihoods, and MCMC sampling.

Usage

```
buzzEBMcontMcontY(
  model,
  dataset,
  my_prior = NULL,
  advanced = NULL,
  m.prec.shape = NULL,
  m.prec.rate = NULL,
  y.prec.shape = NULL,
  y.prec.rate = NULL,
  a.coef.mean = NULL,
  a.coef.prec = NULL,
  b.coef.mean = NULL,
  b.coef.prec = NULL,
  a.pip.hyperalpha = NULL,
  a.pip.hyperbeta = NULL,
  b.pip.hyperalpha = NULL,
  b.pip.hyperbeta = NULL,
  direct.coef.mean = NULL,
  direct.coef.precision = NULL,
  m.prec.init = NULL,
  y.prec.init = NULL,
  direct.coef.init = NULL,
  a.pip.hyperprior.init = NULL,
  b.pip.hyperprior.init = NULL,
  n_chains = NULL,
  n_adapt = NULL,
  n_burnin = NULL,
  n_iter = NULL,
  thin = NULL
)
```

Arguments

model	A description of the model to be fitted. This is typically a formula or a character string using lavaan syntax (e.g., $Y \sim M + X$).
dataset	A data.frame containing the variables specified in the model.

<code>my_prior</code>	Optional data.frame containing custom prior specifications. Run <code>parms <- run_parms_wizard()</code> to see the required structure.
<code>advanced</code>	Character. Use "interactive" for an interactive wizard to choose parameter distributions, or leave NULL for defaults.
<code>m.prec.shape, m.prec.rate</code>	Numeric scalar or vector of length equal to the number of mediators. Shape and rate parameters of the Gamma hyperprior for the mediator residual precisions. By default, a Gamma distribution is used. The default values are 1 and 0.001, respectively.
<code>y.prec.shape, y.prec.rate</code>	Numeric scalar. Shape and rate parameters of the Gamma hyperprior for the outcome residual precision. By default, a Gamma distribution is used. The default values are 1 and 0.001, respectively.
<code>a.coef.mean, a.coef.prec</code>	Numeric scalar or vector. Mean and precision parameters of the Normal prior for the <i>a</i> path effects. By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>b.coef.mean, b.coef.prec</code>	Numeric scalar or vector. Mean and precision parameters of the Normal prior for the <i>b</i> path effects. By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>a.pip.hyperalpha, a.pip.hyperbeta</code>	Numeric scalar or vector. Alpha and beta parameters for the Beta hyperprior of the <i>a</i> path inclusion probabilities. By default, a Beta distribution is used. The default value is 3.
<code>b.pip.hyperalpha, b.pip.hyperbeta</code>	Numeric scalar or vector. Alpha and beta parameters for the Beta hyperprior of the <i>b</i> path inclusion probabilities. By default, a Beta distribution is used. The default value is 3.
<code>direct.coef.mean, direct.coef.precision</code>	Numeric scalar or vector. Mean and precision parameters for the Normal prior of the direct effects (<i>c'</i>). By default, a Normal distribution is used. The default values are 0 and 1.0E-6, respectively.
<code>m.prec.init, y.prec.init</code>	Numeric or NULL. Initial values for residual precisions. Default is 1.
<code>direct.coef.init</code>	Numeric or NULL. Initial value for the direct effect (<i>c'</i>). Default is 0.
<code>a.pip.hyperprior.init, b.pip.hyperprior.init</code>	Numeric or NULL. Initial inclusion probabilities (PIP). Default is 0.5.
<code>n_chains</code>	Integer. Number of MCMC chains.
<code>n_adapt</code>	Integer. Number of adaptation iterations.
<code>n_burnin</code>	Integer. Number of burn-in iterations.
<code>n_iter</code>	Integer. Number of post-burn-in iterations.
<code>thin</code>	Integer. Thinning interval.

Details

Fits a Bayesian mediation model specifically designed for cases where both the mediators (M) and the outcome variable (Y) are continuous.

This function is a specific "worker" function. It identifies X, M, Y via `.parse_buzz_syntax` and performs Bayesian estimation assuming: $M \sim Normal(\dots)$ and $Y \sim Normal(\dots)$.

Value

An object of class `mcmc.list` containing posterior samples.

References

Shi, D., Shi, D., and Fairchild, A. J. (2023) "Variable Selection for Mediators under a Bayesian Mediation Model" <doi:10.1080/10705511.2022.2164285>

Examples

```
# 1. Create a continuous synthetic dataset
set.seed(123)
n <- 100
toy_data <- data.frame(
  X = rnorm(n),
  M1 = rnorm(n),
  M2 = rnorm(n),
  Y = rnorm(n)
)

# 2. Fit the model using lavaan-style syntax
# We define Y as the outcome, with M1, M2, and X as predictors
results <- buzzEBMcontMcontY(
  model = "Y ~ M1 + M2 | X",
  dataset = toy_data,
  n_burnin = 500,
  n_iter = 2000
)

# 3. Check results
summary(results)
```

Description

This function allows users to customize priors without needing to manually construct complex data frames. It displays current defaults, allows selection of specific parameters to change, and provides a menu of available JAGS distributions (e.g., Normal, Gamma, Beta).

Usage

```
run_parms_wizard(verbose = TRUE)
```

Arguments

`verbose` Logical. If TRUE (default), the wizard prints introductory text and the summary review table. If FALSE, only the interactive prompts are displayed.

Details

Launches an interactive console-based interface to guide users through viewing and modifying Bayesian prior distributions for mediation models.

The wizard handles two main types of customization:

- **Distribution Selection:** Choose from supported JAGS distributions.
- **Argument Specification:** Set numeric values for hyperparameters (e.g., shape, rate, mean, precision).

For coefficients like a and b , the wizard will note if they reference hyperpriors (back-references). It is generally recommended to modify the shape of the hyperprior rather than the coefficient's distribution directly to maintain the hierarchical structure.

Value

A data.frame with columns `priors`, `distribution`, and `arguments`. This dataframe is formatted to be passed directly to the `my_prior` argument in related fitting functions.

Note

This function requires an interactive R session. It will return `invisible(NULL)` if the user cancels the process.

Examples

```
# Create toy data
toy_data <- data.frame(
  X = rbinom(100, 1, 0.5),
  M = rbinom(100, 1, 0.3),
  Y = rbinom(100, 1, 0.5)
)

custom_priors <- NULL

if (interactive()){
  # Launch the wizard
  custom_priors <- run_parms_wizard()
}

# Use the resulting object in a model
fit <- buzzEBMcatMcatY(model = "Y ~ M | X",
```

```
dataset = toy_data,  
my_prior = custom_priors)
```

Index

[buzzEBMcatMcatY](#), [2](#)
[buzzEBMcatMcontY](#), [4](#)
[buzzEBMcontMcatY](#), [7](#)
[buzzEBMcontMcontY](#), [10](#)

[prepare_ebmed_data](#), [3](#)

[run_parms_wizard](#), [12](#)