

Package ‘BMIselect’

May 6, 2026

Title Bayesian MI-LASSO for Variable Selection on Multiply-Imputed Datasets

Version 1.0.3

Author Jungang Zou [aut, cre],
Sijian Wang [aut],
Qixuan Chen [aut]

Maintainer Jungang Zou <jungang.zou@gmail.com>

Description Provides a suite of Bayesian MI-LASSO for variable selection methods for multiply-imputed datasets. The package includes four Bayesian MI-LASSO models using shrinkage (Multi-Laplace, Horseshoe, ARD) and Spike-and-Slab (Spike-and-Laplace) priors, along with tools for model fitting via MCMC, four-step projection predictive variable selection, and hyperparameter calibration. Methods are suitable for both continuous and binary covariates under missing-at-random or missing-completely-at-random assumptions. See Zou, J., Wang, S. and Chen, Q. (2025), Bayesian MI-LASSO for Variable Selection on Multiply-Imputed Data. ArXiv, 2211.00114. <[doi:10.48550/arXiv.2211.00114](https://doi.org/10.48550/arXiv.2211.00114)> for more details. We also provide the frequentist’s MI-LASSO function.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports MCMCpack, mvnfast, GIGrv, MASS, Rfast, foreach, doParallel,
arm, mice, abind, stringr, stats, posterior

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2025-08-25 13:20:13 UTC

Contents

ARD_mcmc	2
BMI_LASSO	3
horseshoe_mcmc	5
MI_LASSO	7
multi_laplace_mcmc	8
projection_mean	9
projection_posterior	10
sim_A	11
sim_B	12
sim_C	14
spike_laplace_partially_mcmc	15
Index	17

ARD_mcmc	<i>ARD MCMC Sampler for Multiply-Imputed Regression</i>
----------	---

Description

Implements Bayesian variable selection using the Automatic Relevance Determination (ARD) prior across multiply-imputed datasets. The ARD prior imposes feature-specific shrinkage by placing a prior proportional to inverse of precision of each coefficient.

Usage

```
ARD_mcmc(
  X,
  Y,
  intercept = TRUE,
  nburn = 4000,
  npost = 4000,
  seed = NULL,
  verbose = TRUE,
  printevery = 1000,
  chain_index = 1
)
```

Arguments

X	A 3-D array of predictors with dimensions $D \times n \times p$.
Y	A matrix of outcomes with dimensions $D \times n$.
intercept	Logical; include an intercept? Default TRUE.
nburn	Integer; number of burn-in MCMC iterations. Default 4000.
npost	Integer; number of post-burn-in samples to retain. Default 4000.
seed	Integer or NULL; random seed for reproducibility. Default NULL.

verbose Logical; print progress messages? Default TRUE.
 printevery Integer; print progress every this many iterations. Default 1000.
 chain_index Integer; index of this MCMC chain (for labeling messages). Default 1.

Value

A named list with components:

post_beta Array $n_{\text{post}} \times D \times p$ of sampled regression coefficients.
 post_alpha Matrix $n_{\text{post}} \times D$ of sampled intercepts (if used).
 post_sigma2 Numeric vector length n_{post} , sampled residual variances.
 post_psi2 Matrix $n_{\text{post}} \times p$ of sampled precision parameters for each coefficient.
 post_fitted_Y Array $n_{\text{post}} \times D \times n$ of posterior predictive draws (with noise).
 post_pool_beta Matrix $(n_{\text{post}} * D) \times p$ of pooled coefficient draws.
 post_pool_fitted_Y Matrix $(n_{\text{post}} * D) \times n$ of pooled predictive draws (with noise).
 hat_matrix_proj Matrix $D \times n \times n$ of averaged projection hat-matrices. To avoid recalculate for estimating degree of freedom.

Examples

```

sim <- sim_B(n = 100, p = 20, type = "MAR", SNP = 1.5, corr = 0.5,
low_missing = TRUE, n_imp = 5, seed = 123)
X <- sim$data_MI$X
Y <- sim$data_MI$Y
fit <- ARD_mcmc(X, Y, nburn = 100, npost = 100)
  
```

BMI_LASSO

Bayesian MI-LASSO for Multiply-Imputed Regression

Description

Fit a Bayesian multiple-imputation LASSO (BMI-LASSO) model across multiply-imputed datasets, using one of four priors: Multi-Laplace, Horseshoe, ARD, or Spike-Laplace. Automatically standardizes data, runs MCMC in parallel, performs variable selection via four-step projection predictive variable selection, and selects a final submodel by BIC.

Usage

```

BMI_LASSO(
  X,
  Y,
  model,
  standardize = TRUE,
  SNC = TRUE,
  grid = seq(0, 1, 0.01),
  
```

```

orthogonal = FALSE,
nburn = 4000,
npost = 4000,
seed = NULL,
nchains = 1,
ncores = 1,
output_verbose = TRUE,
printevery = 1000,
...
)

```

Arguments

X	A numeric matrix or array of predictors. If a matrix $n \times p$, it is taken as one imputation; if an array $D \times n \times p$, each slice along the first dimension is one imputed dataset.
Y	A numeric vector or matrix of outcomes. If a vector of length n , it is recycled for each imputation; if a $D \times n$ matrix, each row is the response for one imputation.
model	Character; which prior to use. One of "Multi_Laplace", "Horseshoe", "ARD", or "Spike_Laplace".
standardize	Logical; whether to normalize each X and centralize Y within each imputation before fitting. Default TRUE.
SNC	Logical; if TRUE, use scaled neighborhood criterion; otherwise apply thresholding or median-based selection. Default TRUE.
grid	Numeric vector; grid of scaled neighborhood criterion (or thresholding) to explore. Default <code>seq(0, 1, 0.01)</code> .
orthogonal	Logical; if TRUE, using orthogonal approximations for degrees-of-freedom estimations. Default FALSE.
nburn	Integer; number of burn-in MCMC iterations per chain. Default 4000.
npost	Integer; number of post-burn-in samples to retain per chain. Default 4000.
seed	Optional integer; base random seed. Each chain adds its index.
nchains	Integer; number of MCMC chains to run in parallel. Default 1.
ncores	Integer; number of parallel cores to use. Default 1.
output_verbose	Logical; print progress messages. Default TRUE.
printevery	Integer; print status every so many iterations. Default 1000.
...	Additional model-specific hyperparameters: <ul style="list-style-type: none"> • For "Multi_Laplace": h (shape) and v (scale) of Gamma hyperprior. • For "Spike_Laplace": a (shape) and b (scale) of Gamma hyperprior.

Value

A named list with elements:

`posterior` List of length `nchains` of MCMC outputs (posterior draws).

`select` List of length `nchains` of logical matrices showing which variables are selected at each grid value.

`best_select` List of length `nchains` of the single best selection (by BIC) for each chain.

`posterior_best_models` List of length `nchains` of projected posterior draws for the best sub-model.

`bic_models` List of length `nchains` of BIC values and degrees-of-freedom for each candidate sub-model.

`summary_table_full` A data frame summarizing rank-normalized split-Rhat and other diagnostics for the full model.

`summary_table_selected` A data frame summarizing diagnostics for the selected submodel after projection.

Examples

```
sim <- sim_A(n = 100, p = 20, type = "MAR", SNP = 1.5, low_missing = TRUE, n_imp = 5, seed = 123)
X <- sim$data_MI$X
Y <- sim$data_MI$Y
fit <- BMI_LASSO(X, Y, model = "Horseshoe",
                nburn = 100, npost = 100,
                nchains = 1, ncores = 1)
str(fit$best_select)
```

horseshoe_mcmc

Horseshoe MCMC Sampler for Multiply-Imputed Regression

Description

Implements Bayesian variable selection using the hierarchical Horseshoe prior across multiply-imputed datasets. This model applies global–local shrinkage to regression coefficients via a global scale (τ^2), local scales (λ^2), and auxiliary hyperpriors (κ , η).

Usage

```
horseshoe_mcmc(
  X,
  Y,
  intercept = TRUE,
  nburn = 4000,
  npost = 4000,
  seed = NULL,
  verbose = TRUE,
  printevery = 1000,
  chain_index = 1
)
```

Arguments

<code>X</code>	A 3-D array of predictors with dimensions $D \times n \times p$.
<code>Y</code>	A matrix of outcomes with dimensions $D \times n$.
<code>intercept</code>	Logical; include an intercept term? Default TRUE.
<code>nburn</code>	Integer; number of burn-in MCMC iterations. Default 4000.
<code>npost</code>	Integer; number of post-burn-in samples to retain. Default 4000.
<code>seed</code>	Integer or NULL; random seed for reproducibility. Default NULL.
<code>verbose</code>	Logical; print progress messages? Default TRUE.
<code>printevery</code>	Integer; print progress every this many iterations. Default 1000.
<code>chain_index</code>	Integer; index of this MCMC chain (for labeling prints). Default 1.

Value

A named list with components:

<code>post_beta</code>	Array $npost \times D \times p$ of sampled regression coefficients.
<code>post_alpha</code>	Matrix $npost \times D$ of sampled intercepts (if used).
<code>post_sigma2</code>	Numeric vector of length $npost$, sampled residual variances.
<code>post_lambda2</code>	Matrix $npost \times p$ of local shrinkage parameters λ_j^2 .
<code>post_kappa</code>	Matrix $npost \times p$ of auxiliary local hyperparameters κ_j .
<code>post_tau2</code>	Numeric vector of length $npost$, sampled global scale τ^2 .
<code>post_eta</code>	Numeric vector of length $npost$, sampled auxiliary global hyperparameter η .
<code>post_fitted_Y</code>	Array $npost \times D \times n$ of posterior predictive draws (with noise).
<code>post_pool_beta</code>	Matrix $(npost * D) \times p$ of pooled coefficient draws.
<code>post_pool_fitted_Y</code>	Matrix $(npost * D) \times n$ of pooled predictive draws (with noise).
<code>hat_matrix_proj</code>	Matrix $D \times n \times n$ of averaged projection hat-matrices. To avoid recalculate for estimating degree of freedom.

Examples

```
sim <- sim_B(n = 100, p = 20, type = "MAR", SNP = 1.5, corr = 0.5,
low_missing = TRUE, n_imp = 5, seed = 123)
X <- sim$data_MI$X
Y <- sim$data_MI$Y
fit <- horseshoe_mcmc(X, Y, nburn = 100, npost = 100)
```

MI_LASSO

*Multiple-Imputation LASSO (MI-LASSO)***Description**

Fit a LASSO-like penalty across D multiply-imputed datasets by iteratively reweighted ridge regressions (Equation (4) of the manuscript). For each tuning parameter in `lamvec`, it returns the pooled coefficient estimates, the BIC, and the selected variables.

Usage

```
MI_LASSO(
  X,
  Y,
  lamvec = (2^(seq(-1, 4, by = 0.05)))^2/2,
  maxiter = 200,
  eps = 1e-20,
  ncores = 1
)
```

Arguments

<code>X</code>	A matrix $n \times p$ or an array $D \times n \times p$ of imputed predictor sets. If a matrix is supplied, it is treated as a single imputation ($D = 1$).
<code>Y</code>	A vector length n or a $D \times n$ matrix of outcomes. If a vector, it is reused across imputations.
<code>lamvec</code>	Numeric vector of penalty parameters λ to search. Default $(2^{(\text{seq}(-1, 4, \text{by}=0.05))})^2/2$.
<code>maxiter</code>	Integer; maximum number of ridge-update iterations per <code>lambda</code> . Default 200.
<code>eps</code>	Numeric; convergence tolerance on coefficient change. Default $1e-20$.
<code>ncores</code>	Integer; number of cores for parallelizing over <code>lamvec</code> . Default 1.

Value

If `length(lamvec) > 1`, a list with elements:

`best` List for the *lambda* with minimal BIC containing: `coefficients` ($(p+1) \times D$ intercept + slopes), `bic` (BIC scalar), `varsel` (logical length- p vector of selected predictors), `lambda` (the chosen penalty).

`lambda_path` `length(lamvec) \times 2` matrix of each `lambda` and its corresponding BIC.

If `length(lamvec) == 1`, returns a single list (as above) for that penalty.

Examples

```
sim <- sim_A(n = 100, p = 20, type = "MAR", SNP = 1.5, low_missing = TRUE, n_imp = 5, seed = 123)
X <- sim$data_MI$X
Y <- sim$data_MI$Y
fit <- MI_LASSO(X, Y, lamvec = c(0.1))
```

multi_laplace_mcmc *Multi-Laplace MCMC Sampler for Multiply-Imputed Regression*

Description

Implements Bayesian variable selection under the Multi-Laplace prior on regression coefficients across multiply-imputed datasets. The prior shares local shrinkage parameters (λ_2) across imputations and places a $\text{Gamma}(h, \nu)$ hyperprior on the global parameter ρ .

Usage

```
multi_laplace_mcmc(
  X,
  Y,
  intercept = TRUE,
  h = 2,
  v = NULL,
  nburn = 4000,
  npost = 4000,
  seed = NULL,
  verbose = TRUE,
  printevery = 1000,
  chain_index = 1
)
```

Arguments

X	A 3-D array of predictors with dimensions $D \times n \times p$.
Y	A matrix of outcomes with dimensions $D \times n$.
intercept	Logical; include an intercept? Default TRUE.
h	Numeric; shape parameter of the Gamma prior on ρ . Default 2.
v	Numeric or NULL; scale parameter of the Gamma prior on ρ . If NULL, defaults to $(D+1)/(D*(h-1))$.
nburn	Integer; number of burn-in iterations. Default 4000.
npost	Integer; number of post-burn-in samples to store. Default 4000.
seed	Integer or NULL; random seed for reproducibility. Default NULL.
verbose	Logical; print progress messages? Default TRUE.
printevery	Integer; print progress every this many iterations. Default 1000.
chain_index	Integer; index of this MCMC chain (for messages). Default 1.

Value

A named list with elements:

post_beta Array $n_{\text{post}} \times D \times p$ of sampled regression coefficients.

post_alpha Matrix $n_{\text{post}} \times D$ of sampled intercepts (if used).

post_sigma2 Numeric vector of length n_{post} , sampled residual variances.

post_lambda2 Matrix $n_{\text{post}} \times p$ of sampled local shrinkage parameters.

post_rho Numeric vector of length n_{post} , sampled global parameters.

post_fitted_Y Array $n_{\text{post}} \times D \times n$ of posterior predictive draws (with noise).

post_pool_beta Matrix $(n_{\text{post}} * D) \times p$ of pooled coefficient draws.

post_pool_fitted_Y Matrix $(n_{\text{post}} * D) \times n$ of pooled predictive draws (with noise).

hat_matrix_proj Matrix $D \times n \times n$ of averaged projection hat-matrices. To avoid recalculate for estimating degree of freedom.

h, v Numeric; the shape and scale hyperparameters used.

Examples

```
sim <- sim_B(n = 100, p = 20, type = "MAR", SNP = 1.5, corr = 0.5, low_missing = TRUE,
n_imp = 5, seed = 123)
X <- sim$data_MI$X
Y <- sim$data_MI$Y
fit <- multi_laplace_mcmc(X, Y, intercept = TRUE, nburn = 100, npost = 100)
```

projection_mean	<i>Projecting Posterior Means of Full-Model Coefficients onto a Reduced Subset Model</i>
-----------------	--

Description

Given posterior means of beta1_mat (and optional intercepts alpha1_vec) from a full model fitted on D imputed datasets, compute the predictive projection onto the submodel defined by xs_vec. Returns the projected coefficients (and intercepts, if requested).

Usage

```
projection_mean(X_arr, beta1_mat, xs_vec, sigma2, alpha1_vec = NULL)
```

Arguments

X_arr	A 3-D array of predictors, of dimension $D * n * p$.
beta1_mat	A $D * p$ matrix of full-model coefficients, one row per imputation.
xs_vec	Logical vector of length p ; TRUE for predictors to keep in the submodel.
sigma2	Numeric scalar; the residual variance from the full model (pooled across imputations).
alpha1_vec	Optional numeric vector of length D ; full-model intercepts per imputation. If NULL (the default), the projection omits an intercept term.

Value

A list with components:

beta2_mat A $D \times p$ matrix of projected submodel coefficients.

alpha2_vec (If alpha1_vec provided) numeric vector length D of projected intercepts.

Examples

```
# Simulate a single imputation with n=50, p=5:
D <- 3; n <- 50; p <- 5
X_arr <- array(rnorm(D * n * p), c(D, n, p))
beta1_mat <- matrix(rnorm(D * p), nrow = D)
# Suppose full-model sigma2 pooled is 1.2
sigma2 <- 1.2
# Project onto predictors 1 and 4 only:
xs_vec <- c(TRUE, FALSE, FALSE, TRUE, FALSE)
proj <- projection_mean(X_arr, beta1_mat, xs_vec, sigma2)
str(proj)

# With intercept:
alpha1_vec <- rnorm(D)
proj2 <- projection_mean(X_arr, beta1_mat, xs_vec, sigma2, alpha1_vec)
str(proj2)
```

projection_posterior *Projection of Full-Posterior Draws onto a Reduced-Subset Model*

Description

Given posterior draws beta1_arr (and optional intercepts alpha1_arr) from a full model fitted on D imputed datasets, compute the predictive projection of each draw onto the submodel defined by xs_vec. Returns the projected coefficients (and intercepts, if requested) plus the projected residual variance for each posterior draw.

Usage

```
projection_posterior(X_arr, beta1_arr, sigma1_vec, xs_vec, alpha1_arr = NULL)
```

Arguments

X_arr	A 3-D array of predictors, of dimension $D \times n \times p$.
beta1_arr	A $npost \times D \times p$ array of full-model coefficient draws.
sigma1_vec	Numeric vector of length npost, full-model residual variances.
xs_vec	Logical vector of length p ; TRUE indicates predictors to keep.
alpha1_arr	Optional $npost \times D$ matrix of full_model intercept draws. If NULL (the default), the projection omits an intercept term.

Value

A list with components:

beta2_arr Array $n_{\text{post}} \times D \times p$ of projected submodel coefficients.

alpha2_arr (If alpha1_arr provided) matrix $n_{\text{post}} \times D$ of projected intercepts.

sigma2_opt Numeric vector length n_{post} of projected residual variances.

Examples

```
D <- 3; n <- 50; p <- 5; npost <- 100
X_arr <- array(rnorm(D*n*p), c(D, n, p))
beta1_arr <- array(rnorm(npost*D*p), c(npost, D, p))
sigma1_vec <- runif(npost, 0.5, 2)
xs_vec <- c(TRUE, FALSE, TRUE, FALSE, TRUE)
# Without intercept
proj <- projection_posterior(X_arr, beta1_arr, sigma1_vec, xs_vec)
str(proj)
# With intercept draws
alpha1_arr <- matrix(rnorm(npost*D), nrow = npost, ncol = D)
proj2 <- projection_posterior(X_arr, beta1_arr, sigma1_vec, xs_vec, alpha1_arr)
str(proj2)
```

sim_A	<i>Simulate dataset A: Independent continuous covariates with MCAR/MAR missingness</i>
-------	--

Description

Generates a dataset for Scenario A used in Bayesian MI-LASSO benchmarking. Covariates are iid standard normal, with a fixed true coefficient vector, linear outcome, missingness imposed on specified columns under MCAR or MAR, and multiple imputations via predictive mean matching.

Usage

```
sim_A(
  n = 100,
  p = 20,
  type = "MAR",
  SNP = 1.5,
  low_missing = TRUE,
  n_imp = 5,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations.
p	Integer. Number of covariates (columns). Takes values in {20, 40}.
type	Character. Missingness mechanism: "MCAR" or "MAR".
SNP	Numeric. Signal-to-noise ratio controlling error variance.
low_missing	Logical. If TRUE, use low missingness rates; if FALSE, higher missingness.
n_imp	Integer. Number of multiple imputations to generate.
seed	Integer or NULL. Random seed for reproducibility.

Value

A list with components:

data_O A list of complete covariate matrix and outcomes before missingness.

data_mis A list of covariate matrix and outcomes with missing values.

data_MI A list of array of imputed covariates ($n_imp \times n \times p$) and a matrix of imputed outcomes ($n_imp \times n$).

data_CC A list of complete-case covariate matrix and outcomes.

important Logical vector of true nonzero coefficient indices.

covmat True covariance matrix used for X.

beta True coefficient vector.

Examples

```
sim <- sim_A(n = 100, p = 20, type = "MAR", SNP = 1.5,
            low_missing = TRUE, n_imp = 5, seed = 123)
str(sim)
```

sim_B *Simulate dataset B: AR(1)-correlated continuous covariates with MCAR/MAR missingness*

Description

Generates a dataset for Scenario B used in Bayesian MI-LASSO benchmarking. Covariates are multivariate normal with AR(1) covariance, with a fixed true coefficient vector, linear outcome, missingness imposed on specified columns under MCAR or MAR, and multiple imputations via predictive mean matching.

Usage

```
sim_B(
  n = 100,
  p = 20,
  low_missing = TRUE,
  type = "MAR",
  SNP = 1.5,
  corr = 0.5,
  n_imp = 5,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations.
p	Integer. Number of covariates (columns). Takes values in {20, 40}.
low_missing	Logical. If TRUE, use low missingness rates; if FALSE, higher missingness.
type	Character. Missingness mechanism: "MCAR" or "MAR".
SNP	Numeric. Signal-to-noise ratio controlling error variance.
corr	Numeric. AR(1) correlation parameter
n_imp	Integer. Number of multiple imputations to generate.
seed	Integer or NULL. Random seed for reproducibility.

Value

A list with components:

data_O A list of complete covariate matrix and outcomes before missingness.

data_mis A list of covariate matrix and outcomes with missing values.

data_MI A list of array of imputed covariates ($n_imp \times n \times p$) and a matrix of imputed outcomes ($n_imp \times n$).

data_CC A list of complete-case covariate matrix and outcomes.

important Logical vector of true nonzero coefficient indices.

covmat True covariance matrix used for X.

beta True coefficient vector.

Examples

```
sim <- sim_B(n = 100, p = 20, type = "MAR", SNP = 1.5, corr = 0.5,
            low_missing = TRUE, n_imp = 5, seed = 123)
str(sim)
```

sim_C	<i>Simulate dataset C: AR(1)-latent Gaussian dichotomized to binary covariates with MCAR/MAR missingness</i>
-------	--

Description

Generates binary covariates by thresholding an AR(1) latent Gaussian, then proceeds as in sim_B.

Usage

```
sim_C(
  n = 100,
  p = 20,
  low_missing = TRUE,
  type = "MAR",
  SNP = 1.5,
  corr = 0.5,
  n_imp = 5,
  seed = NULL
)
```

Arguments

n	Integer. Number of observations.
p	Integer. Number of covariates (columns). Takes values in {20, 40}.
low_missing	Logical. If TRUE, use low missingness rates; if FALSE, higher missingness.
type	Character. Missingness mechanism: "MCAR" or "MAR".
SNP	Numeric. Signal-to-noise ratio controlling error variance.
corr	Numeric. AR(1) correlation parameter
n_imp	Integer. Number of multiple imputations to generate.
seed	Integer or NULL. Random seed for reproducibility.

Value

A list with components:

data_O A list of complete covariate matrix and outcomes before missingness.

data_mis A list of covariate matrix and outcomes with missing values.

data_MI A list of array of imputed covariates ($n_imp \times n \times p$) and a matrix of imputed outcomes ($n_imp \times n$).

data_CC A list of complete-case covariate matrix and outcomes.

important Logical vector of true nonzero coefficient indices.

covmat True covariance matrix used for X.

beta True coefficient vector.

Examples

```
sim <- sim_C(n = 100, p = 20, type = "MAR", SNP = 1.5, corr = 0.5,
            low_missing = TRUE, n_imp = 5, seed = 123)
str(sim)
```

```
spike_laplace_partially_mcmc
```

Spike-and-Laplace MCMC Sampler for Multiply-Imputed Regression

Description

Implements Bayesian variable selection using a spike-and-slab prior with a Laplace (double-exponential) slab on nonzero coefficients. Latent inclusion indicators γ follow Bernoulli(θ), and their probabilities follow independent Beta(a , b) priors.

Usage

```
spike_laplace_partially_mcmc(
  X,
  Y,
  intercept = TRUE,
  a = 2,
  b = NULL,
  nburn = 4000,
  npost = 4000,
  seed = NULL,
  verbose = TRUE,
  printevery = 1000,
  chain_index = 1
)
```

Arguments

X	A 3-D array of predictors with dimensions $D * n * p$.
Y	A matrix of outcomes with dimensions $D * n$.
intercept	Logical; include an intercept term? Default TRUE.
a	Numeric; shape parameter of the Gamma prior. Default 2.
b	Numeric or NULL; scale parameter of the Gamma prior. If NULL, defaults to $0.5 * (D+1) / (D * (a-1))$.
nburn	Integer; number of burn-in MCMC iterations. Default 4000.
npost	Integer; number of post-burn-in samples to retain. Default 4000.
seed	Integer or NULL; random seed for reproducibility. Default NULL.
verbose	Logical; print progress messages? Default TRUE.
printevery	Integer; print progress every this many iterations. Default 1000.
chain_index	Integer; index of this MCMC chain (for labeling messages). Default 1.

Value

A named list with components:

post_rho Numeric vector length npost, sampled global scale ρ .
 post_gamma Matrix npost * p of sampled inclusion indicators.
 post_theta Matrix npost * p of sampled Beta parameters θ_j .
 post_alpha Matrix npost * D of sampled intercepts (if used).
 post_lambda2 Matrix npost * p of sampled local scale parameters λ_j^2 .
 post_sigma2 Numeric vector length npost, sampled residual variances.
 post_beta Array npost * D * p of sampled regression coefficients.
 post_fitted_Y Array npost * D * n of posterior predictive draws (including noise).
 post_pool_beta Matrix (npost * D) * p of pooled coefficient draws.
 post_pool_fitted_Y Matrix (npost * D) * n of pooled predictive draws (with noise).
 hat_matrix_proj Matrix D * n * n of averaged projection hat-matrices. To avoid recalculate for estimating degree of freedom.
 a, b Numeric values of the rho hyperparameters used.

Examples

```
sim <- sim_B(n = 100, p = 20, type = "MAR", SNP = 1.5, corr = 0.5,
low_missing = TRUE, n_imp = 5, seed = 123)
X <- sim$data_MI$X
Y <- sim$data_MI$Y
fit <- spike_laplace_partially_mcmc(X, Y, nburn = 10, npost = 10)
```

Index

[ARD_mcmc](#), [2](#)

[BMI_LASSO](#), [3](#)

[horseshoe_mcmc](#), [5](#)

[MI_LASSO](#), [7](#)

[multi_laplace_mcmc](#), [8](#)

[projection_mean](#), [9](#)

[projection_posterior](#), [10](#)

[sim_A](#), [11](#)

[sim_B](#), [12](#)

[sim_C](#), [14](#)

[spike_laplace_partially_mcmc](#), [15](#)