Package: mr.mashr (via r-universe)

May 17, 2026

Encoding UTF-8

Type Package

Version 0.3.44

Date 2025-12-21

Title Multiple Regression with Multivariate Adaptive Shrinkage

Description Provides an implementation of methods for multivariate multiple regression with adaptive shrinkage priors as described in F. Morgante et al (2023) <doi:10.1371/journal.pgen.1010539>.

URL https://github.com/stephenslab/mr.mashr

License MIT + file LICENSE

Depends R (>= 3.1.0)

SystemRequirements GNU make

Imports methods, stats, Matrix, Rcpp (>= 1.1.0), RcppParallel (>= 5.1.10), mvtnorm, matrixStats, mashr (>= 0.2.73), ebnm, flashier (>= 1.0.7), parallel

Suggests testthat, varbvs, knitr, rmarkdown,

LinkingTo Rcpp, RcppArmadillo (>= 0.10.4.0.0), RcppParallel

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/pak/sysreqs cmake make libgsl0-dev libicu-dev libuv1-dev libssl-dev

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

Date/Publication 2026-02-16 16:00:01 UTC

RemoteUrl https://github.com/stephenslab/mr.mashr

RemoteRef HEAD

RemoteSha be538c3856d12a127104058708ba58e0af2da670

Contents

`autoselect.mixsd` *Compute a grid of standard deviations to scale the canonical covariance matrices.*

Description

Function to compute a grid of standard deviations from univariate simple linear regression summary statistics

Usage

```
autoselect.mixsd(data, mult = 2)
```

Arguments

`data`	a list with two elements. 1 - Bhat, a numeric vector of regression coefficients. 2 - Shat, a numeric vector of of standard erros for the regression coefficients.
`mult`	a scalar affecting how dense the resulting grid of standard deviations will be.

Value

A numeric vector of standard deviations.

coef.mr.mash *Extract coefficients from mr.mash fit.*

Description

Extract coefficients from mr.mash fit.

Usage

```
## S3 method for class 'mr.mash'
coef(object, ...)
```

Arguments

object	a mr.mash fit.
...	Other arguments (not used).

Value

(p+1) x r matrix of coefficients.

coef.mr.mash.rss *Extract coefficients from mr.mash.rss fit.*

Description

Extract coefficients from mr.mash.rss fit.

Usage

```
## S3 method for class 'mr.mash.rss'
coef(object, ...)
```

Arguments

object	a mr.mash fit.
...	Other arguments (not used).

Value

p x r or (p+1) x r matrix of coefficients, depending on whether an intercept was computed.

compute_canonical_covs *Compute canonical covariance matrices.*

Description

Function to compute canonical covariance matrices scaled.

Usage

```
compute_canonical_covs(
  r,
  singletons = TRUE,
  hetgrid = c(0, 0.25, 0.5, 0.75, 1)
)
```

Arguments

r	number of responses.
singletons	if TRUE, the response-specific effect matrices will be included.
hetgrid	scalar or numeric vector of positive correlation [0, 1] of the effects across responses.

Value

A list containing the canonical covariance matrices.

compute_data_driven_covs *Compute data-driven covariance matrices.*

Description

Function to compute data-driven covariance matrices from summary statistics using PCA, FLASH and the sample covariance. These matrices are de-noised using Extreme Deconvolution.

Usage

```
compute_data_driven_covs(
  sumstats,
  subset_thresh = NULL,
  n_pcs = 3,
  flash_factors = c("default", "nonneg"),
  flash_remove_singleton = FALSE,
  Gamma = diag(ncol(sumstats$Bhat))
)
```

Arguments

sumstats	a list with two elements. 1 - Bhat, a numeric vector of regression coefficients. 2 - Shat, a numeric vector of of standard erros for the regression coefficients.
subset_thresh	scalar indicating the threshold for selecting the effects to be used for computing the covariance matrices based on false local sign rate (lfsr) for a response-by-response ash analysis.
n_pcs	indicating the number of principal components to be selected.
flash_factors	factors "default" to use flashr default function to initialize factors, currently udv_si. "nonneg" to implement a non-negative constraint on the factors
flash_remove_singleton	whether or not factors corresponding to singleton matrices should be removed from output.
Gamma	an r x r correlation matrix for the residuals; must be positive definite.

Value

A list containing the (de-noised) data-driven covariance matrices.

---

compute_univariate_sumstats

Compute summary statistics from univariate simple linear regression.

Description

Function to compute regression coefficients and their standard errors from univariate simple linear regression.

Usage

```
compute_univariate_sumstats(
  X,
  Y,
  standardize = FALSE,
  standardize.response = FALSE,
  mc.cores = 1
)
```

Arguments

X	n x p matrix of covariates.
Y	n x r matrix of responses.
standardize	If TRUE, X is "standardized" using the sample means and sample standard deviations.
standardize.response	If TRUE, Y is "standardized" using the sample means and sample standard deviations.
mc.cores	Number of cores to use. Parallelization is done over responses.

Value

A list with following elements:

Bhat	p x r matrix of the regression coeffcients.
Shat	p x r matrix of the standard errors for regression coeffcients.

`expand_covs` *Expand covariance matrices by a grid of variances for use in* `mr.mash`.

Description

Function to scale the covariance matrices by a grid of variances.

Usage

```
expand_covs(mats, grid, zeromat = TRUE)
```

Arguments

mats	a list of covariance matrices.
grid	scalar or numeric vector of variances of the effects.
zeromat	if TRUE, the no-effect matrix will be included.

Value

A list containing the scaled covariance matrices.

`mr.mash` *Multiple Regression with Multivariate Adaptive Shrinkage.*

Description

Performs multivariate multiple regression with mixture-of-normals prior.

Usage

```
mr.mash(
  X,
  Y,
  S0,
  w0 = rep(1/(length(S0)), length(S0)),
  V = NULL,
  mu1_init = matrix(0, nrow = ncol(X), ncol = ncol(Y)),
  tol = 0.0001,
  convergence_criterion = c("mu1", "ELBO"),
```

```
  max_iter = 5000,
  update_w0 = TRUE,
  update_w0_method = "EM",
  w0_penalty = rep(1, length(S0)),
  update_w0_max_iter = Inf,
  w0_threshold = 0,
  compute_ELBO = TRUE,
  standardize = TRUE,
  verbose = TRUE,
  update_V = FALSE,
  update_V_method = c("full", "diagonal"),
  version = c("Rcpp", "R"),
  e = 1e-08,
 ca_update_order = c("consecutive", "decreasing_logBF", "increasing_logBF", "random"),
  nthreads = as.integer(NA)
)
```

Arguments

X	n x p matrix of covariates.
Y	n x r matrix of responses.
S0	List of length K containing the desired r x r prior covariance matrices on the regression coefficients.
w0	K-vector with prior mixture weights, each associated with the respective covariance matrix in S0.
V	r x r residual covariance matrix.
mu1_init	p x r matrix of initial estimates of the posterior mean regression coefficients. These should be on the same scale as the X provided. If standardize=TRUE, mu1_init will be scaled appropriately after standardizing X.
tol	Convergence tolerance.
convergence_criterion	Criterion to use for convergence check.
max_iter	Maximum number of iterations for the optimization algorithm.
update_w0	If TRUE, prior weights are updated.
update_w0_method	Method to update prior weights. Only EM is currently supported.
w0_penalty	K-vector of penalty terms (>=1) for each mixture component. Default is all components are unpenalized.
update_w0_max_iter	Maximum number of iterations for the update of w0.
w0_threshold	Drop mixture components with weight less than this value. Components are dropped at each iteration after 15 initial iterations. This is done to prevent from dropping some poetentially important components prematurely.
compute_ELBO	If TRUE, ELBO is computed.

standardize	If TRUE, X is "standardized" using the sample means and sample standard deviations. Standardizing X allows a faster implementation, but the prior has a different interpretation. Coefficients and covariances are returned on the original scale.
verbose	If TRUE, some information about the algorithm's process is printed at each iteration.
update_V	if TRUE, residual covariance is updated.
update_V_method	Method to update residual covariance. So far, "full" and "diagonal" are supported. If update_V=TRUE and V is not provided by the user, this option will determine how V is computed (and fixed) internally from mu1_init.
version	Whether to use R or C++ code to perform the coordinate ascent updates.
e	A small number to add to the diagonal elements of the prior matrices to improve numerical stability of the updates.
ca_update_order	The order with which coordinates are updated. So far, "consecutive", "decreasing_logBF", "increasing_logBF", "random" are supported.
nthreads	Number of RcppParallel threads to use for the updates. When nthreads is NA, the default number of threads is used; see defaultNumThreads. This setting is ignored when version = "R".

Value

A mr.mash fit, stored as a list with some or all of the following elements:

mu1	p x r matrix of posterior means for the regression coeffcients.
S1	r x r x p array of posterior covariances for the regression coeffcients.
w1	p x K matrix of posterior assignment probabilities to the mixture components.
V	r x r residual covariance matrix
w0	K-vector with (updated, if update_w0=TRUE) prior mixture weights, each associated with the respective covariance matrix in S0
.	
S0	r x r x K array of prior covariance matrices on the regression coefficients
.	
intercept	r-vector containing posterior mean estimate of the intercept.
fitted	n x r matrix of fitted values.
G	r x r covariance matrix of fitted values.
pve	r-vector of proportion of variance explained by the covariates.
ELBO	Evidence Lower Bound (ELBO) at last iteration.
progress	A data frame including information regarding convergence criteria at each iteration.

converged — TRUE or FALSE, indicating whether the optimization algorithm converged to a solution within the chosen tolerance level.

Y — n x r matrix of responses at last iteration (only relevant when missing values are present in the input Y).

Examples

```
###Set seed
set.seed(123)

###Simulate X and Y
##Set parameters
n  <- 1000
p <- 100
p_causal <- 20
r <- 5

###Simulate data
out <- simulate_mr_mash_data(n, p, p_causal, r, pve=0.5, B_cor=1,
                             B_scale=1, X_cor=0, X_scale=1, V_cor=0)

###Split the data in training and test sets
Ytrain <- out$Y[-c(1:200), ]
Xtrain <- out$X[-c(1:200), ]
Ytest <- out$Y[c(1:200), ]
Xtest <- out$X[c(1:200), ]

###Specify the covariance matrices for the mixture-of-normals prior.
univ_sumstats <- compute_univariate_sumstats(Xtrain, Ytrain,
                   standardize=TRUE, standardize.response=FALSE)
grid <- autoselect.mixsd(univ_sumstats, mult=sqrt(2))^2
S0 <- compute_canonical_covs(ncol(Ytrain), singletons=TRUE,
                             hetgrid=c(0, 0.25, 0.5, 0.75, 1))
S0 <- expand_covs(S0, grid, zeromat=TRUE)

###Fit mr.mash
# Note that max_iter was set to 4 in this example to shorten
# the running time. In practice, you should allow the model-fitting
# algorithm to run for many more iterations to obtain better estimates.
fit <- mr.mash(Xtrain, Ytrain, S0, update_V=TRUE, max_iter = 4,
               nthreads = 1)

# Compare the "fitted" values of Y against the true Y in the training set.
plot(fit$fitted,Ytrain,pch = 20,col = "darkblue",xlab = "true",
     ylab = "fitted")
abline(a = 0,b = 1,col = "magenta",lty = "dotted")

# Predict the multivariate outcomes in the test set using the fitted model.
Ytest_est <- predict(fit,Xtest)
plot(Ytest_est,Ytest,pch = 20,col = "darkblue",xlab = "true",
     ylab = "predicted")
abline(a = 0,b = 1,col = "magenta",lty = "dotted")
```

`mr.mash.rss` *Multiple Regression with Multivariate Adaptive Shrinkage from summary data.*

Description

Performs multivariate multiple regression with mixture-of-normals prior.

Usage

```
mr.mash.rss(
  Bhat,
  Shat,
  Z,
  R,
  covY,
  n,
  S0,
  w0 = rep(1/(length(S0)), length(S0)),
  V,
  mu1_init = NULL,
  tol = 0.0001,
  convergence_criterion = c("mu1", "ELBO"),
  max_iter = 5000,
  update_w0 = TRUE,
  update_w0_method = "EM",
  w0_penalty = rep(1, length(S0)),
  update_w0_max_iter = Inf,
  w0_threshold = 0,
  compute_ELBO = TRUE,
  standardize = FALSE,
  verbose = TRUE,
  update_V = FALSE,
  update_V_method = c("full", "diagonal"),
  version = c("Rcpp", "R"),
  e = 1e-08,
 ca_update_order = c("consecutive", "decreasing_logBF", "increasing_logBF", "random"),
  X_colmeans = NULL,
  Y_colmeans = NULL,
  check_R = TRUE,
  R_tol = 1e-08,
  nthreads = as.integer(NA)
)
```

Arguments

Bhat	p x r matrix of regression coefficients from univariate simple linear regression.
Shat	p x r matrix of standard errors of the regression coefficients from univariate simple linear regression.
Z	p x r matrix of Z-scores from univariate simple linear regression.
R	p x p dense or sparse correlation matrix among the variables.
covY	r x r covariance matrix across responses.
n	scalar indicating the sample size.
S0	List of length K containing the desired r x r prior covariance matrices on the regression coefficients.
w0	K-vector with prior mixture weights, each associated with the respective covariance matrix in S0.
V	r x r residual covariance matrix.
mu1_init	p x r matrix of initial estimates of the posterior mean regression coefficients. These should be on the same scale as the X provided. If standardize=TRUE, mu1_init will be scaled appropriately after standardizing X.
tol	Convergence tolerance.
convergence_criterion	Criterion to use for convergence check.
max_iter	Maximum number of iterations for the optimization algorithm.
update_w0	If TRUE, prior weights are updated.
update_w0_method	Method to update prior weights. Only EM is currently supported.
w0_penalty	K-vector of penalty terms (>=1) for each mixture component. Default is all components are unpenalized.
update_w0_max_iter	Maximum number of iterations for the update of w0.
w0_threshold	Drop mixture components with weight less than this value. Components are dropped at each iteration after 15 initial iterations. This is done to prevent from dropping some potentially important components prematurely.
compute_ELBO	If TRUE, ELBO is computed.
standardize	If TRUE, X is "standardized" using the sample means and sample standard deviations. Standardizing X allows a faster implementation, but the prior has a different interpretation. Coefficients and covariances are returned on the original scale.
verbose	If TRUE, some information about the algorithm's process is printed at each iteration.
update_V	if TRUE, residual covariance is updated.
update_V_method	Method to update residual covariance. So far, "full" and "diagonal" are supported. If update_V=TRUE and V is not provided by the user, this option will determine how V is computed (and fixed) internally from mu1_init.

version	Whether to use R or C++ code to perform the coordinate ascent updates.
e	A small number to add to the diagonal elements of the prior matrices to improve numerical stability of the updates.
ca_update_order	The order with which coordinates are updated. So far, "consecutive", "decreasing_logBF", "increasing_logBF", "random" are supported.
X_colmeans	a p-vector of variable means.
Y_colmeans	a r-vector of response means.
check_R	If TRUE, R is checked to be positive semidefinite.
R_tol	tolerance to declare positive semi-definiteness of R.
nthreads	Number of RcppParallel threads to use for the updates. When nthreads is NA, the default number of threads is used; see defaultNumThreads. This setting is ignored when version = "R".

Value

A mr.mash.rss fit, stored as a list with some or all of the following elements:

mu1	p x r matrix of posterior means for the regression coeffcients.
S1	r x r x p array of posterior covariances for the regression coeffcients.
w1	p x K matrix of posterior assignment probabilities to the mixture components.
V	r x r residual covariance matrix
w0	K-vector with (updated, if update_w0=TRUE) prior mixture weights, each associated with the respective covariance matrix in S0
.	
S0	r x r x K array of prior covariance matrices on the regression coefficients
.	
intercept	r-vector containing posterior mean estimate of the intercept, if X_colmeans and Y_colmeans are provided. Otherwise, NA is output.
ELBO	Evidence Lower Bound (ELBO) at last iteration.
progress	A data frame including information regarding convergence criteria at each iteration.
converged	TRUE or FALSE, indicating whether the optimization algorithm converged to a solution within the chosen tolerance level.
Y	n x r matrix of responses at last iteration (only relevant when missing values are present in the input Y).

Examples

```
###Set seed
set.seed(123)

###Simulate X and Y
##Set parameters
n  <- 1000
p <- 100
p_causal <- 20
r <- 5

###Simulate data
out <- simulate_mr_mash_data(n, p, p_causal, r, pve=0.5, B_cor=1,
                             B_scale=1, X_cor=0, X_scale=1, V_cor=0)

###Split the data in training and test sets
Ytrain <- out$Y[-c(1:200), ]
Xtrain <- out$X[-c(1:200), ]
Ytest <- out$Y[c(1:200), ]
Xtest <- out$X[c(1:200), ]

###Specify the covariance matrices for the mixture-of-normals prior.
univ_sumstats <- compute_univariate_sumstats(Xtrain, Ytrain,
                   standardize=TRUE, standardize.response=FALSE)
grid <- autoselect.mixsd(univ_sumstats, mult=sqrt(2))^2
S0 <- compute_canonical_covs(ncol(Ytrain), singletons=TRUE,
                             hetgrid=c(0, 0.25, 0.5, 0.75, 1))
S0 <- expand_covs(S0, grid, zeromat=TRUE)

###Fit mr.mash
# Note that max_iter was set to 4 in this example to shorten
# the running time. In practice, you should allow the model-fitting
# algorithm to run for many more iterations to obtain better estimates.
covY <- cov(Ytrain)
corX <- cor(Xtrain)
n_train <- nrow(Ytrain)
fit <- mr.mash.rss(Bhat=univ_sumstats$Bhat, Shat=univ_sumstats$Shat,
                   S0=S0, covY=covY, R=corX, n=n_train, V=covY,
                   update_V=TRUE, X_colmeans=colMeans(Xtrain),
                   Y_colmeans=colMeans(Ytrain), max_iter = 4,
                   nthreads = 1)

# Predict the multivariate outcomes in the test set using the fitted model.
Ytest_est <- predict(fit,Xtest)
plot(Ytest_est,Ytest,pch = 20,col = "darkblue",xlab = "true",
     ylab = "predicted")
abline(a = 0,b = 1,col = "magenta",lty = "dotted")
```

`predict.mr.mash` *Predict future observations from mr.mash fit.*

Description

Predict future observations from mr.mash fit.

Usage

```
## S3 method for class 'mr.mash'
predict(object, newx, ...)
```

Arguments

object	a mr.mash fit.
newx	a new value for X at which to do predictions.
...	Additional arguments (not used).

Value

Matrix of predicted values.

---

predict.mr.mash.rss *Predict future observations from mr.mash.rss fit.*

Description

Predict future observations from mr.mash.rss fit.

Usage

```
## S3 method for class 'mr.mash.rss'
predict(object, newx, ...)
```

Arguments

object	a mr.mash.rss fit.
newx	a new value for X at which to do predictions.
...	Additional arguments (not used).

Value

Matrix of predicted values.

`simulate_mr_mash_data` *Simulate data to test* `mr.mash`.

Description

Function to simulate data from $MN_{nxr}(XB, I, V)$, where $X \sim N_p(0, Gamma)$, $B \sim \sum_k w_k N_r(0, Sigma_k)$, with $Gamma$, w_k, $Sigma_k$, and V defined by the user.

Usage

```
simulate_mr_mash_data(
  n,
  p,
  p_causal,
  r,
  r_causal = list(1:r),
  intercepts = rep(1, r),
  pve = 0.2,
  B_cor = 1,
  B_scale = 1,
  w = 1,
  X_cor = 0,
  X_scale = 1,
  V_cor = 0
)
```

Arguments

`n`	scalar indicating the number of samples.
`p`	scalar indicating the number of variables.
`p_causal`	scalar indicating the number of causal variables.
`r`	scalar indicating the number of responses.
`r_causal`	a list of numeric vectors (one element for each mixture component) indicating in which responses the causal variables have an effect.
`intercepts`	numeric vector of intercept for each response.
`pve`	per-response proportion of variance explained by the causal variables.
`B_cor`	scalar or numeric vector (one element for each mixture component) with positive correlation [0, 1] between causal effects.
`B_scale`	scalar or numeric vector (one element for each mixture component) with the diagonal value for Sigma_k;
`w`	scalar or numeric vector (one element for each mixture component) with mixture proportions associated to each mixture component.
`X_cor`	scalar indicating the positive correlation [0, 1] between variables.
`X_scale`	scalar indicating the diagonal value for Gamma.
`V_cor`	scalar indicating the positive correlation [0, 1] between residuals

Value

A list with some or all of the following elements:

- `X` — n x p matrix of variables.
- `Y` — n x r matrix of responses.
- `B` — p x r matrix of effects.
- `V` — r x r residual covariance matrix among responses.
- `Sigma` — list of r x r covariance matrices among the effects.
- `Gamma` — p x p covariance matrix among the variables.
- `intercepts` — r-vector of intercept for each response.
- `causal_responses` — a list of numeric vectors of indexes indicating which responses have causal effects for each mixture component.
- `causal_variables` — p_causal-vector of indexes indicating which variables are causal.
- `causal_vars_to_mixture_comps` — p_causal-vector of indexes indicating from which mixture components each causal effect comes.

Examples

```
set.seed(1)
dat <- simulate_mr_mash_data(n=50, p=40, p_causal=20, r=5,
                             r_causal=list(1:2, 3:4), intercepts=rep(1, 5),
                             pve=0.2, B_cor=c(0, 1), B_scale=c(0.5, 1),
                             w=c(0.5, 0.5), X_cor=0.5, X_scale=1,
                             V_cor=0)
```

Index