Package: hibayes (via r-universe)

October 25, 2026

Title Individual-Level, Summary-Level and Single-Step Bayesian Regression Model

Version 3.1.0

Date 2025-09-05

Description A user-friendly tool to fit Bayesian regression models. It can fit 3 types of Bayesian models using individual-level, summary-level, and individual plus pedigree-level (single-step) data for both Genomic prediction/selection (GS) and Genome-Wide Association Study (GWAS), it was designed to estimate joint effects and genetic parameters for a complex trait, including: (1) fixed effects and coefficients of covariates, (2) environmental random effects, and its corresponding variance, (3) genetic variance, (4) residual variance, (5) heritability, (6) genomic estimated breeding values (GEBV) for both genotyped and non-genotyped individuals, (7) SNP effect size, (8) phenotype/genetic variance explained (PVE) for single or multiple SNPs, (9) posterior probability of association of the genomic window (WPPA), (10) posterior inclusive probability (PIP). The functions are not limited, we will keep on going in enriching it with more features. References: Lilin Yin et al. (2025) <doi:10.18637/jss.v114.i06>; Meuwissen et al. (2001) <doi:10.1093/genetics/157.4.1819>; Gustavo et al. (2013) <doi:10.1534/genetics.112.143313>; Habier et al. (2011) <doi:10.1186/1471-2105-12-186>; Yi et al. (2008) <doi:10.1534/genetics.107.085589>; Zhou et al. (2013) <doi:10.1371/journal.pgen.1003264>; Moser et al. (2015) <doi:10.1371/journal.pgen.1004969>; Lloyd-Jones et al. (2019) <doi:10.1038/s41467-019-12653-0>; Henderson (1976) <doi:10.2307/2529339>; Fernando et al. (2014) <doi:10.1186/1297-9686-46-50>.

License GPL-3

Maintainer Lilin Yin <ylilin@163.com>

URL https://github.com/YinLiLin/hibayes

BugReports https://github.com/YinLiLin/hibayes/issues

Encoding UTF-8

Imports utils, stats, methods, stringr, CMplot

Depends R (>= 3.3.0), bigmemory, Matrix

LinkingTo Rcpp, RcppArmadillo (>= 0.9.600.0.0), RcppProgress, BH, bigmemory, Matrix

RoxygenNote 7.2.3

NeedsCompilation yes

Author Lilin Yin [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-4413-7976>), Haohao Zhang [aut, cph] (ORCID: <https://orcid.org/0000-0002-7913-5228>), Xiaolei Liu [aut, cph] (ORCID: <https://orcid.org/0000-0002-9954-1426>)

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

Date/Publication 2025-09-05 16:40:20 UTC

RemoteUrl https://github.com/cran/hibayes

RemoteRef HEAD

RemoteSha 725bb1ca429d9cb5f6a5ce5190748f2a63d54694

Contents

ibrm 2
ldmat 7
read_plink 8
sbrm 9
ssbrm 12

Index **17**

`ibrm` *Bayes model*

Description

Bayes linear regression model using individual level data

$$y = X\beta + Rr + M\alpha + e$$

where β is a vector of estimated coefficient for covariates, and r is a vector of environmental random effects. M is a matrix of genotype covariate, α is a vector of estimated marker effect size. e is a vector of residuals.

Usage

```
ibrm(
  formula,
  data = NULL,
  M = NULL,
  M.id = NULL,
 method = c("BayesCpi", "BayesA", "BayesL", "BSLMM", "BayesR", "BayesB", "BayesC",
    "BayesBpi", "BayesRR"),
  map = NULL,
  Pi = NULL,
  fold = NULL,
  niter = NULL,
  nburn = NULL,
  thin = 5,
  windsize = NULL,
  windnum = NULL,
  dfvr = NULL,
  s2vr = NULL,
  vg = NULL,
  dfvg = NULL,
  s2vg = NULL,
  ve = NULL,
  dfve = NULL,
  s2ve = NULL,
  lambda = 0,
  printfreq = 100,
  seed = 666666,
  threads = 4,
  verbose = TRUE
)
```

Arguments

formula	a two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a '~' operator and the terms, separated by '+' operators, on the right. Random-effects terms are distinguished by vertical bars (1|') separating expressions for design matrices from grouping factors.
data	the data frame containing the variables named in 'formula', NOTE that the first column in 'data' should be the individual id.
M	numeric matrix of genotype with individuals in rows and markers in columns, NAs are not allowed.
M.id	vector of id for genotyped individuals, NOTE that no need to adjust the order of id to be the same between 'data' and 'M', the package will do it automatically.
method	bayes methods including: "BayesB", "BayesA", "BayesL", "BayesRR", "BayesBpi", "BayesC", "BayesCpi", "BayesR", "BSLMM".

- "BayesRR": Bayes Ridge Regression, all SNPs have non-zero effects and share the same variance, equals to RRBLUP or GBLUP.
- "BayesA": all SNPs have non-zero effects, and take different variance which follows an inverse chi-square distribution.
- "BayesB": only a small proportion of SNPs (1-Pi) have non-zero effects, and take different variance which follows an inverse chi-square distribution.
- "BayesBpi": the same with "BayesB", but 'Pi' is not fixed.
- "BayesC": only a small proportion of SNPs (1-Pi) have non-zero effects, and share the same variance.
- "BayesCpi": the same with "BayesC", but 'Pi' is not fixed.
- "BayesL": BayesLASSO, all SNPs have non-zero effects, and take different variance which follows an exponential distribution.
- "BSLMM": all SNPs have non-zero effects, and take the same variance, but a small proportion of SNPs have additional shared variance.
- "BayesR": only a small proportion of SNPs have non-zero effects, and the SNPs are allocated into different groups, each group has the same variance.

`map` (optional, only for GWAS) the map information of genotype, at least 3 columns are: SNPs, chromosome, physical position.

`Pi` vector, the proportion of zero effect and non-zero effect SNPs, the first value must be the proportion of non-effect markers.

`fold` proportion of variance explained for groups of SNPs, the default is c(0, 0.0001, 0.001, 0.01).

`niter` the number of MCMC iteration.

`nburn` the number of iterations to be discarded.

`thin` the number of thinning after burn-in. Note that smaller thinning frequency may have higher accuracy of estimated parameters, but would result in more memory for collecting process, on contrary, bigger frequency may have negative effect on accuracy of estimations.

`windsize` window size in bp for GWAS, the default is NULL.

`windnum` fixed number of SNPs in a window for GWAS, if it is specified, 'windsize' will be invalid, the default is NULL.

`dfvr` the number of degrees of freedom for the distribution of environmental variance.

`s2vr` scale parameter for the distribution of environmental variance.

`vg` prior value of genetic variance.

`dfvg` the number of degrees of freedom for the distribution of genetic variance.

`s2vg` scale parameter for the distribution of genetic variance.

`ve` prior value of residual variance.

`dfve` the number of degrees of freedom for the distribution of residual variance.

`s2ve` scale parameter for the distribution of residual variance.

`lambda` value of ridge regression for inverting a matrix.

`printfreq` frequency of printing iterative details on console.

`seed` seed for random sample.

`threads` number of threads used for OpenMP.

`verbose` whether to print the iteration information on console.

Details

- the fixed effects and covariates in 'formula' must be in factors and numeric, respectively. if not, please remember to use 'as.factor' and 'as.numeric' to transform.
- the package has the automatical function of taking the intersection and adjusting the order of id between 'data' and the genotype 'M', thus the first column in 'data' should be the individual id.
- if any one of the options 'windsize' and 'windnum' is specified, the GWAS results will be returned, and the 'map' information must be provided, in which the physical positions should be all in digital values.
- the 'windsize' or 'windnum' option only works for the methods of which the assumption has a proportion of zero effect markers, e.g., BayesB, BayesBpi, BayesC, BayesCpi, BSLMM, and BayesR.

Value

the function returns a 'blrMod' object containing

$mu the regression intercept

$pi estimated proportion of zero effect and non-zero effect SNPs

$beta estimated coefficients for all covariates

$r estimated environmental random effects

$Vr estimated variance for all environmental random effect

$Vg estimated genetic variance

$Ve estimated residual variance

$h2 estimated heritability (h2 = Vg / (Vr + Vg + Ve))

$alpha estimated effect size of all markers

$g genomic estimated breeding value

$e residuals of the model

$pip the frequency for markers to be included in the model during MCMC iteration, known as posterior inclusive probability (PIP)

$gwas WPPA is defined to be the window posterior probability of association, it is estimated by counting the number of MCMC samples in which

$$\alpha$$

is nonzero for at least one SNP in the window

$MCMCsamples the collected samples of posterior estimation for all the above parameters across MCMC iterations

References

Meuwissen, Theo HE, Ben J. Hayes, and Michael E. Goddard. "Prediction of total genetic value using genome-wide dense marker maps." Genetics 157.4 (2001): 1819-1829.
de los Campos, G., Hickey, J. M., Pong-Wong, R., Daetwyler, H. D., and Calus, M. P. (2013). Whole-genome regression and prediction methods applied to plant and animal breeding. Genetics, 193(2), 327-345.
Habier, David, et al. "Extension of the Bayesian alphabet for genomic selection." BMC bioinformatics 12.1 (2011): 1-12.
Yi, Nengjun, and Shizhong Xu. "Bayesian LASSO for quantitative trait loci mapping." Genetics 179.2 (2008): 1045-1055.
Zhou, Xiang, Peter Carbonetto, and Matthew Stephens. "Polygenic modeling with Bayesian sparse linear mixed models." PLoS genetics 9.2 (2013): e1003264.
Moser, Gerhard, et al. "Simultaneous discovery, estimation and prediction analysis of complex traits using a Bayesian mixture model." PLoS genetics 11.4 (2015): e1004969.

Examples

```
# Load the example data attached in the package
pheno_file_path = system.file("extdata", "demo.phe", package = "hibayes")
pheno = read.table(pheno_file_path, header=TRUE)

bfile_path = system.file("extdata", "demo", package = "hibayes")
bin = read_plink(bfile_path, threads=1)
fam = bin$fam
geno = bin$geno
map = bin$map

# For GS/GP
## no environmental effects:
fit = ibrm(T1~1, data=pheno, M=geno, M.id=fam[,2], method="BayesCpi",
 niter=2000, nburn=1200, thin=5, threads=1)

## overview of the returned results
summary(fit)



## add fixed effects or covariates:
fit = ibrm(T1~sex+season+day+bwt, data=pheno, M=geno, M.id=fam[,2],
 method="BayesCpi")

## add environmental random effects:
fit = ibrm(T1~sex+(1|loc)+(1|dam), data=pheno, M=geno, M.id=fam[,2],
 method="BayesCpi")

# For GWAS
fit = ibrm(T1~sex+bwt+(1|dam), data=pheno, M=geno, M.id=fam[,2],
 method="BayesCpi", map=map, windsize=1e6)
```

```
# get the SD of estimated SNP effects for markers
summary(fit)$alpha
# get the SD of estimated breeding values
summary(fit)$g
```

`ldmat` *LD variance-covariance matrix calculation*

Description

To calculate density or sparse LD variance-covariance matrix with genotype in bigmemory format.

Usage

```
ldmat(
  geno,
  map = NULL,
  gwas.geno = NULL,
  gwas.map = NULL,
  chisq = NULL,
  ldchr = FALSE,
  threads = 4,
  verbose = FALSE
)
```

Arguments

`geno`	the reference genotype panel in bigmemory format.
`map`	the map information of reference genotype panel, columns are: SNPs, chromosome, physical position.
`gwas.geno`	(optional) the genotype of gwas samples which were used to generate the summary data.
`gwas.map`	(optional) the map information of the genotype of gwas samples, columns are: SNPs, chromosome, physical position.
`chisq`	chi-squre value for generating sparse matrix, if n*r2 < chisq, it would be set to zero.
`ldchr`	lpgical, whether to calulate the LD between chromosomes.
`threads`	the number of threads used in computation.
`verbose`	whether to print the information.

Value

For full ld matrix, it returns a standard R matrix, for sparse matrix, it returns a 'dgCMatrix'.

Examples

```
bfile_path = system.file("extdata", "demo", package = "hibayes")
data = read_plink(bfile_path)
geno = data$geno
map = data$map

xx = ldmat(geno, threads=4, verbose=FALSE)   #chromosome wide full ld matrix
# xx = ldmat(geno, chisq=5, threads=4)   #chromosome wide sparse ld matrix
# xx = ldmat(geno, map, ldchr=FALSE, threads=4)   #chromosome block ld matrix
# xx = ldmat(geno, map, ldchr=FALSE, chisq=5, threads=4)  #chromosome block + sparse ld matrix
```

`read_plink` *data load*

Description

To load plink binary data

Usage

```
read_plink(
  bfile = "",
  maxLine = 10000,
  impute = TRUE,
  mode = c("A", "D"),
  out = NULL,
  threads = 4
)
```

Arguments

`bfile`	character, prefix of Plink binary format data.
`maxLine`	number, set the number of lines to read at a time.
`impute`	logical, whether to impute missing values in genotype by major alleles.
`mode`	"A" or "D", additive effect or dominant effect.
`out`	character, path and prefix of output file
`threads`	number, the number of used threads for parallel process

Value

four files will be generated in the directed folder: "xx.desc", "xx.bin", "xx.id, "xx.map", where 'xx' is the prefix of the argument 'out', the memory-mapping files can be fast loaded into memory by 'geno = attach.big.matrix("xx.desc")'. Note that hibayes will code the genotype A1A1 as 2, A1A2 as 1, and A2A2 as 0, where A1 is the first allele of each marker in ".bim" file, therefore the estimated effect size is on A1 allele, users should pay attention to it when a process involves marker effect.

Examples

```
bfile_path = system.file("extdata", "demo", package = "hibayes")
data = read_plink(bfile_path, out=tempfile(), mode="A")
fam = data$fam
geno = data$geno
map = data$map
```

---

sbrm *SBayes model*

Description

Bayes linear regression model using summary level data

Usage

```
sbrm(
  sumstat,
  ldm,
 method = c("BayesB", "BayesA", "BayesL", "BayesRR", "BayesBpi", "BayesC", "BayesCpi",
    "BayesR", "CG"),
  map = NULL,
  Pi = NULL,
  lambda = NULL,
  fold = NULL,
  niter = NULL,
  nburn = NULL,
  thin = 5,
  windsize = NULL,
  windnum = NULL,
  vg = NULL,
  dfvg = NULL,
  s2vg = NULL,
  ve = NULL,
  dfve = NULL,
  s2ve = NULL,
  printfreq = 100,
  seed = 666666,
  threads = 4,
  verbose = TRUE
)
```

Arguments

sumstat	matrix of summary data, details refer to https://cnsgenomics.com/software/gcta/#COJO.
ldm	dense or sparse matrix, ld for reference panel (m * m, m is the number of SNPs). NOTE that the order of SNPs should be consistent with summary data.

`method` bayes methods including: "BayesB", "BayesA", "BayesL", "BayesRR", "BayesBpi", "BayesC", "BayesCpi", "BayesR", "CG".

- "BayesRR": Bayes Ridge Regression, all SNPs have non-zero effects and share the same variance, equals to RRBLUP or GBLUP.
- "BayesA": all SNPs have non-zero effects, and take different variance which follows an inverse chi-square distribution.
- "BayesB": only a small proportion of SNPs (1-Pi) have non-zero effects, and take different variance which follows an inverse chi-square distribution.
- "BayesBpi": the same with "BayesB", but 'Pi' is not fixed.
- "BayesC": only a small proportion of SNPs (1-Pi) have non-zero effects, and share the same variance.
- "BayesCpi": the same with "BayesC", but 'Pi' is not fixed.
- "BayesL": BayesLASSO, all SNPs have non-zero effects, and take different variance which follows an exponential distribution.
- "BayesR": only a small proportion of SNPs have non-zero effects, and the SNPs are allocated into different groups, each group has the same variance.
- "CG": conjugate gradient algorithm with assigned lambda.

`map` (optional, only for GWAS) the map information of genotype, at least 3 columns are: SNPs, chromosome, physical position.

`Pi` vector, the proportion of zero effect and non-zero effect SNPs, the first value must be the proportion of non-effect markers.

`lambda` value or vector, the ridge regression value for each SNPs.

`fold` percentage of variance explained for groups of SNPs, the default is c(0, 0.0001, 0.001, 0.01).

`niter` the number of MCMC iteration.

`nburn` the number of iterations to be discarded.

`thin` the number of thinning after burn-in. Note that smaller thinning frequency may have higher accuracy of estimated parameters, but would result in more memory for collecting process, on contrary, bigger frequency may have negative effect on accuracy of estimations.

`windsize` window size in bp for GWAS, the default is 1e6.

`windnum` fixed number of SNPs in a window for GWAS, if it is specified, 'windsize' will be invalid, the default is NULL.

`vg` prior value of genetic variance.

`dfvg` the number of degrees of freedom for the distribution of genetic variance.

`s2vg` scale parameter for the distribution of genetic variance.

`ve` prior value of residual variance.

`dfve` the number of degrees of freedom for the distribution of residual variance.

`s2ve` scale parameter for the distribution of residual variance.

`printfreq` frequency of collecting the estimated parameters and printing on console. Note that smaller frequency may have higher accuracy of estimated parameters, but would result in more time and memory for collecting process, on contrary, bigger frequency may have an negative effect on accuracy of estimations.

seed	seed for random sample.
threads	number of threads used for OpenMP.
verbose	whether to print the iteration information on console.

Details

- if any one of the options 'windsize' and 'windnum' is specified, the GWAS results will be returned, and the 'map' information must be provided, in which the physical positions should be all in digital values.
- the 'windsize' or 'windnum' option only works for the methods of which the assumption has a proportion of zero effect markers, e.g., BayesB, BayesBpi, BayesC, BayesCpi, BSLMM, and BayesR.

Value

the function returns a 'blrMod' object containing

$pi estimated proportion of zero effect and non-zero effect SNPs

$Vg estimated genetic variance

$Ve estimated residual variance

$h2 estimated heritability (h2 = Vg / (Vg + Ve))

$alpha estimated effect size of all markers

$pip the frequency for markers to be included in the model during MCMC iteration, also known as posterior inclusive probability (PIP)

$gwas WPPA is defined to be the window posterior probability of association, it is estimated by counting the number of MCMC samples in which

$$\alpha$$

is nonzero for at least one SNP in the window

$MCMCsamples the collected samples of posterior estimation for all the above parameters across MCMC iterations

References

Lloyd-Jones, Luke R., et al. "Improved polygenic prediction by Bayesian multiple regression on summary statistics." Nature communications 10.1 (2019): 1-11.

Examples

```
bfile_path = system.file("extdata", "demo", package = "hibayes")
bin = read_plink(bfile_path, threads=1)
fam = bin$fam
geno = bin$geno
map = bin$map

sumstat_path = system.file("extdata", "demo.ma", package = "hibayes")
sumstat = read.table(sumstat_path, header=TRUE)
```

```
head(sumstat)


# computate ld variance covariance matrix
## construct genome wide full variance-covariance matrix
ldm1 <- ldmat(geno, threads=4)
## construct genome wide sparse variance-covariance matrix
# ldm2 <- ldmat(geno, chisq=5, threads=4)
## construct chromosome wide full variance-covariance matrix
# ldm3 <- ldmat(geno, map, ldchr=FALSE, threads=4)
## construct chromosome wide sparse variance-covariance matrix
# ldm4 <- ldmat(geno, map, ldchr=FALSE, chisq=5, threads=4)

# if the order of SNPs in genotype is not consistent with the order in sumstat file,
# prior adjusting is necessary.
indx = match(map[, 1], sumstat[, 1])
sumstat = sumstat[indx, ]

# fit model
fit = sbrm(sumstat=sumstat, ldm=ldm1, method="BayesCpi", Pi = c(0.95, 0.05),
 niter=20000, nburn=12000, seed=666666, map=map, windsize=1e6, threads=1)

# overview of the returned results
summary(fit)

# get the SD of estimated SNP effects for markers
summary(fit)$alpha
```

---

ssbrm *Single-step Bayes model*

Description

Single-step Bayes linear regression model using individual level data and pedigree information

$$y = X\beta + Rr + M\alpha + U\epsilon + e$$

where y is the vector of phenotypic values for both genotyped and non-genotyped individuals, β is a vector of estimated coefficient for covariates, M contains the genotype (M_2) for genotyped individuals and the imputed genotype ($M_1 = A_{12}A_{22}^{-1}M_2$) for non-genotyped individuals, ϵ is the vector of genotype imputation error, e is a vector of residuals.

Usage

```
ssbrm(
  formula,
  data = NULL,
  M = NULL,
```

```
  M.id = NULL,
  pedigree = NULL,
 method = c("BayesCpi", "BayesA", "BayesL", "BayesR", "BayesB", "BayesC", "BayesBpi",
    "BayesRR"),
  map = NULL,
  Pi = NULL,
  fold = NULL,
  niter = NULL,
  nburn = NULL,
  thin = 5,
  windsize = NULL,
  windnum = NULL,
  maf = 0.01,
  dfvr = NULL,
  s2vr = NULL,
  vg = NULL,
  dfvg = NULL,
  s2vg = NULL,
  ve = NULL,
  dfve = NULL,
  s2ve = NULL,
  printfreq = 100,
  seed = 666666,
  threads = 4,
  verbose = TRUE
)
```

Arguments

formula	a two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a '~' operator and the terms, separated by '+' operators, on the right. Random-effects terms are distinguished by vertical bars (1|') separating expressions for design matrices from grouping factors.
data	the data frame containing the variables named in 'formula', NOTE that the first column in 'data' should be the individual id.
M	numeric matrix of genotype with individuals in rows and markers in columns, NAs are not allowed.
M.id	vector of id for genotype.
pedigree	matrix of pedigree, 3 columns limited, the order of columns shoud be "id", "sir", "dam".
method	bayes methods including: "BayesB", "BayesA", "BayesL", "BayesRR", "BayesBpi", "BayesC", "BayesCpi", "BayesR". • "BayesRR": Bayes Ridge Regression, all SNPs have non-zero effects and share the same variance, equals to RRBLUP or GBLUP. • "BayesA": all SNPs have non-zero effects, and take different variance which follows an inverse chi-square distribution.

- "BayesB": only a small proportion of SNPs (1-Pi) have non-zero effects, and take different variance which follows an inverse chi-square distribution.
- "BayesBpi": the same with "BayesB", but 'Pi' is not fixed.
- "BayesC": only a small proportion of SNPs (1-Pi) have non-zero effects, and share the same variance.
- "BayesCpi": the same with "BayesC", but 'Pi' is not fixed.
- "BayesL": BayesLASSO, all SNPs have non-zero effects, and take different variance which follows an exponential distribution.
- "BayesR": only a small proportion of SNPs have non-zero effects, and the SNPs are allocated into different groups, each group has the same variance.

`map` (optional, only for GWAS) the map information of genotype, at least 3 columns are: SNPs, chromosome, physical position.

`Pi` vector, the proportion of zero effect and non-zero effect SNPs, the first value must be the proportion of non-effect markers.

`fold` proportion of variance explained for groups of SNPs, the default is c(0, 0.0001, 0.001, 0.01).

`niter` the number of MCMC iteration.

`nburn` the number of iterations to be discarded.

`thin` the number of thinning after burn-in. Note that smaller thinning frequency may have higher accuracy of estimated parameters, but would result in more memory for collecting process, on contrary, bigger frequency may have negative effect on accuracy of estimations.

`windsize` window size in bp for GWAS, the default is NULL.

`windnum` fixed number of SNPs in a window for GWAS, if it is specified, 'windsize' will be invalid, the default is NULL.

`maf` the effects of markers whose MAF is lower than the threshold will not be estimated.

`dfvr` the number of degrees of freedom for the distribution of environmental variance.

`s2vr` scale parameter for the distribution of environmental variance.

`vg` prior value of genetic variance.

`dfvg` the number of degrees of freedom for the distribution of genetic variance.

`s2vg` scale parameter for the distribution of genetic variance.

`ve` prior value of residual variance.

`dfve` the number of degrees of freedom for the distribution of residual variance.

`s2ve` scale parameter for the distribution of residual variance.

`printfreq` frequency of printing iterative details on console.

`seed` seed for random sample.

`threads` number of threads used for OpenMP.

`verbose` whether to print the iteration information on console.

Value

the function returns a a 'blrMod' object containing

$J coefficient for genotype imputation residuals

$Veps estimated variance of genotype imputation residuals

$epsilon genotype imputation residuals

$mu the regression intercept

$pi estimated proportion of zero effect and non-zero effect SNPs

$beta estimated coefficients for all covariates

$r estimated environmental random effects

$Vr estimated variance for all environmental random effect

$Vg estimated genetic variance

$Ve estimated residual variance

$h2 estimated heritability (h2 = Vg / (Vr + Vg + Ve))

$g data.frame, the first column is the list of individual id, the second column is the genomic estimated breeding value for all individuals, including genotyped and non-genotyped.

$alpha estimated effect size of all markers

$e residuals of the model

$pip the frequency for markers to be included in the model during MCMC iteration, also known as posterior inclusive probability (PIP)

$gwas WPPA is defined to be the window posterior probability of association, it is estimated by counting the number of MCMC samples in which

$$\alpha$$

is nonzero for at least one SNP in the window

$MCMCsamples the collected samples of posterior estimation for all the above parameters across MCMC iterations

References

Fernando, Rohan L., Jack CM Dekkers, and Dorian J. Garrick. "A class of Bayesian methods to combine large numbers of genotyped and non-genotyped animals for whole-genome analyses." Genetics Selection Evolution 46.1 (2014): 1-13.
Henderson, C.R.: A simple method for computing the inverse of a numerator relationship matrix used in prediction of breeding values. Biometrics 32(1), 69-83 (1976).

Examples

```
# Load the example data attached in the package
pheno_file_path = system.file("extdata", "demo.phe", package = "hibayes")
pheno = read.table(pheno_file_path, header=TRUE)

bfile_path = system.file("extdata", "demo", package = "hibayes")
bin = read_plink(bfile_path, threads=1)
```

```
fam = bin$fam
geno = bin$geno
map = bin$map

pedigree_file_path = system.file("extdata", "demo.ped", package = "hibayes")
ped = read.table(pedigree_file_path, header=TRUE)

# For GS/GP
## no environmental effects:
fit = ssbrm(T1~1, data=pheno, M=geno, M.id=fam[,2], pedigree=ped,
 method="BayesCpi", niter=1000, nburn=600, thin=5, printfreq=100, threads=1)

## overview of the returned results
summary(fit)



## add fixed effects or covariates:
fit = ssbrm(T1~sex+bwt, data=pheno, M=geno, M.id=fam[,2], pedigree=ped,
 method="BayesCpi")

## add environmental random effects:
fit = ssbrm(T1~(1|loc)+(1|dam), data=pheno, M=geno, M.id=fam[,2],
 pedigree=ped, method="BayesCpi")

# For GWAS
fit = ssbrm(T1~sex+bwt+(1|dam), data=pheno, M=geno, M.id=fam[,2],
 pedigree=ped, method="BayesCpi", map=map, windsize=1e6)


# get the SD of estimated SNP effects for markers
summary(fit)$alpha
# get the SD of estimated breeding values
summary(fit)$g
```

Index

ibrm, 2

ldmat, 7

read_plink, 8

sbrm, 9
ssbrm, 12