Package 'hibayes'

January 13, 2022

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

Version 1.0.1

Date 2022-01-11

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: 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 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 Apache License 2.0

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, Rcpp

Depends R (>= 3.3.0), bigmemory, Matrix

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

RoxygenNote 7.1.1

SystemRequirements C++11

NeedsCompilation yes

Author Lilin Yin [aut, cre, cph], Haohao Zhang [aut, cph], Xiaolei Liu [aut, cph]

Repository CRAN

Date/Publication 2022-01-13 10:02:42 UTC

R topics documented:

bayes 2
ldmat 5
read_plink 7
sbayes 8
ssbayes 10

Index **15**

`bayes` *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

```
bayes(
  y,
  M,
  X = NULL,
  R = NULL,
 model = c("BayesCpi", "BayesA", "BayesL", "BSLMM", "BayesR", "BayesB", "BayesC",
    "BayesBpi", "BayesRR"),
  map = NULL,
```

```
  Pi = NULL,
  fold = NULL,
  niter = 20000,
  nburn = 14000,
  windsize = NULL,
  windnum = NULL,
  vg = NULL,
  dfvg = NULL,
  s2vg = NULL,
  ve = NULL,
  dfve = NULL,
  s2ve = NULL,
  lambda = 0,
  outfreq = 100,
  seed = 666666,
  threads = 4,
  verbose = TRUE
)
```

Arguments

y	vector of phenotype, use 'NA' for the missings. The number and order of individuals of y, M, X, R should be exactly the same.
M	numeric matrix of genotype with individuals in rows and markers in columns, NAs are not allowed.
X	(optional) covariate matrix of all individuals, all values should be in digits, characters are not allowed, please use 'model.matrix.lm' function to prepare it.
R	(optional) environmental random effects matrix of all individuals, NAs are not allowed for the individuals with phenotypic value.
model	bayes model 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.
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.
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.
outfreq	frequency of information output on console, the default is 100.
seed	seed for random sample.
threads	number of threads used for OpenMP.
verbose	whether to print the iteration information.

Value

the function returns a list 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

$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, known as posterior inclusive probability (PIP)

$g genomic estimated breeding value

$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

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", "pheno.txt", package = "hibayes")
pheno = read.table(pheno_file_path, header=TRUE)
bfile_path = system.file("extdata", "geno", package = "hibayes")
data = read_plink(bfile_path, out=tempfile())
fam = data$fam
geno = data$geno
map = data$map

# Adjust the order of phenotype by genotype id
geno.id = fam[, 2]
pheno = pheno[match(geno.id, pheno[, 1]), ]

# Add fixed effects, covariates, and random effect
X <- model.matrix.lm(~as.numeric(scale)+as.factor(sex), data=pheno, na.action = "na.pass")
X <- X[, -1] #remove the intercept
# then fit the model as: fit = bayes(..., X=X, R=pheno[,c("group")], ...)

# For GS/GP
fit = bayes(y=pheno[, 2], M=geno, model="BayesR", niter=200, nburn=100, outfreq=10)

# For GWAS
fit = bayes(y=pheno[, 2], M=geno, map=map, windsize=1e6, model="BayesCpi")
```

`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 = TRUE
)
```

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", "geno", package = "hibayes")
data = read_plink(bfile_path, out=tempfile())
geno = data$geno
map = data$map

xx = ldmat(geno, threads=4)   #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

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", "geno", package = "hibayes")
data = read_plink(bfile_path, out=tempfile(), mode="A")
fam = data$fam
geno = data$geno
map = data$map
```

sbayes *SBayes model*

Description

Bayes linear regression model using summary level data

Usage

```
sbayes(
  sumstat,
  ldm,
 model = c("BayesB", "BayesA", "BayesL", "BayesRR", "BayesBpi", "BayesC", "BayesCpi",
    "BayesR", "CG"),
  map = NULL,
  Pi = NULL,
  lambda = NULL,
  fold = NULL,
  niter = 20000,
  nburn = 14000,
  windsize = NULL,
  windnum = NULL,
  vg = NULL,
  dfvg = NULL,
  s2vg = NULL,
  ve = NULL,
  dfve = NULL,
  s2ve = NULL,
  outfreq = 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.
model	bayes model 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.

`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.

`outfreq` frequency of information output on console, the default is 100.

`seed` seed for random sample.

`threads` number of threads used for OpenMP.

`verbose` whether to print the iteration information.

Value

the function returns a list containing

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

$vg estimated genetic variance

$ve estimated residual variance

$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

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", "geno", package = "hibayes")
data = read_plink(bfile_path, out=tempfile())
geno = data$geno
map = data$map
head(map)
sumstat_path = system.file("extdata", "geno.ma", package = "hibayes")
sumstat = read.table(sumstat_path, header=TRUE)
head(sumstat)


# computate ld variance covariance matrix
ldm1 = ldmat(geno, threads=4)   #chromosome wide full ld matrix

# 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 = sbayes(sumstat=sumstat, ldm=ldm1, model="BayesR")
```

---

ssbayes *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

```
ssbayes(
  y,
  y.id,
  M,
  M.id,
  P,
  X = NULL,
  R = NULL,
 model = c("BayesCpi", "BayesA", "BayesL", "BayesR", "BayesB", "BayesC", "BayesBpi",
    "BayesRR"),
  map = NULL,
  Pi = NULL,
  fold = NULL,
  niter = 20000,
  nburn = 14000,
  windsize = NULL,
  windnum = NULL,
  vg = NULL,
  dfvg = NULL,
  s2vg = NULL,
  ve = NULL,
  dfve = NULL,
  s2ve = NULL,
  outfreq = 100,
  seed = 666666,
  threads = 4,
  verbose = TRUE
)
```

Arguments

y	vector of phenotype, use 'NA' for the missings.
y.id	vector of id for phenotype.
M	numeric matrix of genotype with individuals in rows and markers in columns, NAs are not allowed.
M.id	vector of id for genotype.
P	matrix of pedigree, 3 columns limited, the order of columns shoud be "id", "sir", "dam".
X	(optional) covariate matrix of all individuals, all values should be in digits, characters are not allowed, please use 'model.matrix.lm' function to prepare it.
R	(optional) environmental random effects matrix of all individuals, NAs are not allowed for the individuals with phenotypic value.
model	bayes model 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.
- "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.

`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.

`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.

`outfreq` frequency of information output on console, the default is 100.

`seed` seed for random sample.

`threads` number of threads used for OpenMP.

`verbose` whether to print the iteration information.

Value

the function returns a list containing

$J coefficient for 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

$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)

$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.

$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

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", "pheno.txt", package = "hibayes")
pheno = read.table(pheno_file_path, header=TRUE)
pedigree_file_path = system.file("extdata", "ped.txt", package = "hibayes")
ped = read.table(pedigree_file_path, header=TRUE)
bfile_path = system.file("extdata", "geno", package = "hibayes")
data = read_plink(bfile_path, out=tempfile())
fam = data$fam
geno = data$geno
map = data$map

# NOTE: for ssbayes model, there is no NEED to adjust the order of id in different files
geno.id = fam[, 2]
pheno.id = pheno[, 1]

# Add fixed effects, covariates, and random effect
X <- model.matrix.lm(~as.numeric(scale)+as.factor(sex), data=pheno, na.action = "na.pass")
```

```
X <- X[, -1] #remove the intercept
# then fit the model as: fit = ssbayes(..., X=X, R=pheno[,c("group")], ...)


# For GS/GP
fit = ssbayes(y=pheno[, 2], y.id=pheno.id, M=geno, M.id=geno.id, P=ped,
model="BayesR", niter=200, nburn=100, outfreq=10)
# For GWAS
fit = ssbayes(y=pheno[, 2], y.id=pheno.id, M=geno, M.id=geno.id, P=ped,
map=map, windsize=1e6, model="BayesCpi")
```

Index

bayes, 2

ldmat, 5

read_plink, 7

sbayes, 8
ssbayes, 10