Package: GCPBayes (via r-universe)

October 20, 2026

Type Package

Title Bayesian Meta-Analysis of Pleiotropic Effects Using Group Structure

Version 4.3.0

Description Run a Gibbs sampler for a multivariate Bayesian sparse group selection model with Dirac, continuous and hierarchical spike prior for detecting pleiotropy on the traits. This package is designed for summary statistics containing estimated regression coefficients and its estimated covariance matrix. The methodology is available from: Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021) <doi:10.1002/sim.8855>.

URL https://github.com/tbaghfalaki/GCPBayes

License GPL (>= 2.0)

Encoding UTF-8

Imports MASS, mvtnorm, invgamma, gdata, truncnorm, abind, posterior, Rcpp (>= 1.0.9)

LinkingTo Rcpp, RcppArmadillo

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

NeedsCompilation yes

Author Yazdan Asgari [aut, cre], Taban Baghfalaki [aut], Benoit Liquet [aut], Pierre-Emmanuel Sugier [aut], Mohammed Sedki [aut], Therese Truong [aut]

Maintainer Yazdan Asgari <yazdan.asgari@inserm.fr>

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

Date/Publication 2025-11-28 19:20:14 UTC

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

RemoteRef HEAD

RemoteSha 0c6b879b42345ea02f33f5b2c287bd09f8e76079

Contents

CS . . . 2
DNAJC1 . . . 8
DS . . . 9
e2_Monte_Carlo_EM . . . 14
GCPBayes . . . 15
HS . . . 16
MCMCplot . . . 21
PARP2 . . . 25
PARP2_summary . . . 26
Simulated_individual . . . 27
Simulated_individual_survival . . . 27
Simulated_summary . . . 28
summaryCS . . . 29
summaryDS . . . 30
summaryHS . . . 32

Index **34**

`CS` *Continuous Spike*

Description

Utilize a Gibbs sampler to conduct analysis on a multivariate Bayesian sparse group selection model, employing a Continuous spike prior to identify pleiotropic effects across traits. This function is tailored for summary statistics comprising estimated regression coefficients and their corresponding covariance matrices.

Usage

```
CS(
  Betah,
  Sigmah,
  kappa0,
  tau20,
  zeta0,
  m,
  K,
  niter = 1000,
  burnin = 500,
  nthin = 2,
  nchains = 2,
  a1 = a1,
  a2 = a2,
  c1 = c1,
  c2 = c2,
```

```
  sigma2 = 10^-3,
  snpnames = snpnames,
  genename = genename
)
```

Arguments

Betah	A list containing m-dimensional vectors of the regression coefficients for K studies.
Sigmah	A list containing the positive definite covariance matrices (m*m-dimensional) which is the estimated covariance matrices of K studies.
kappa0	Initial value for kappa (its dimension is equal to nchains).
tau20	Initial value for tau2 (its dimension is equal to nchains).
zeta0	Initial value for zeta.
m	Number of variables in the group.
K	Number of traits.
niter	Number of iterations for the Gibbs sampler.
burnin	Number of burn-in iterations.
nthin	The lag of the iterations used for the posterior analysis is defined (or thinning rate).
nchains	Number of Markov chains, when nchains>1, the function calculates the Gelman-Rubin convergence statistic, as modified by Brooks and Gelman (1998).
a1, a2	Hyperparameters of kappa. Default is a1=0.1 and a2=0.1.
c1, c2	Hyperparameters of tau2. Default is c1=0.1 and c2=0.1.
sigma2	Variance of spike (multivariate normal distribution with a diagonal covariance matrix with small variance) representing the null effect distribution. Default is 10^-3.
snpnames	Names of variables for the group.
genename	Name of group.

Details

Let betah_k, k=1,...,K be a m-dimensional vector of the regression coefficients for the kth study and Sigmah_k be its estimated covariance matrix. The hierarchical set-up of CS prior, by considering summary statistics (betah_k and Sigmah_k, k=1,...,K) as the input of the method, is given by:

betah_k ~ (1 - zeta_k) N_m(0,sigma2 I_m) + zeta_k N_m(0,tau2 I_m),

zeta_k ~ Ber(kappa),

kappa ~ Beta(a_1,a_2),

tau2 ~ inverseGamma (c_1,c_2).

Value

- mcmcchain: The list of simulation output for all parameters.
- Summary: Summary statistics for regression coefficients in each study.
- Criteria: genename, snpnames, PPA, log10BF, lBFDR, theta.
- Indicator: A table containing m rows of binary indicators for each study, the number of studies with nonzero signal and having pleiotropic effect by credible interval (CI). The first K columns show nonzero signals, K+1 th column includes the number of studies with nonzero signal and the last column shows an indicator for having pleiotropic effect of each SNP.

Author(s)

Taban Baghfalaki.

References

Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. Statistics in Medicine, 40(6), 1498-1518.

Examples

```
############################## Gene DNAJC1 ################################################
data(DNAJC1)
Breast <- DNAJC1$Breast
Thyroid <- DNAJC1$Thyroid
genename <- "DNAJC1"
snpnames <- Breast$snp
Betah <- list(Breast$beta, Thyroid$beta)
Sigmah <- list(diag(Breast$se^2), diag(Thyroid$se^2))
K <- 2
m <- 14

pvalue <- matrix(0, K, m)
for (k in 1:K) {
  pvalue[k, ] <- 2 * pnorm(-abs(Betah[[k]] / sqrt(diag(Sigmah[[k]]))))
}

zinit <- rep(0, K)
for (j in 1:K) {
  index <- 1:m
  PVALUE <- p.adjust(pvalue[j, ])
  SIGNALS <- index[PVALUE < 0.05]
  modelf1 <- rep(0, m)
  modelf1[SIGNALS] <- 1
  if (max(modelf1) == 1) (zinit[j] <- 1)
}

set.seed(123)
RES <- CS(Betah, Sigmah,
  kappa0 = 0.5, tau20 = 1, zeta0 = zinit,
```

```
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1, a1 = 0.1, a2 = 0.1,
  c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames = snpnames, genename = genename
)
## Not run:
  print(RES)


  RES1 <- CS(Betah, Sigmah,
    kappa0 = c(0.2, 0.5), tau20 = c(1, 2), zeta0 = zinit,
    m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
    a1 = 0.1, a2 = 0.1, c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames, genename
  )
  print(RES1)
 ################### Simulated summary level data with K=5 ################################
  data(Simulated_summary)
  genename <- Simulated_summary$genename
  snpnames <- Simulated_summary$snpnames
  Betah <- Simulated_summary$simBeta
  Sigmah <- Simulated_summary$simSIGMA
  K <- 5
  m <- 10

  pvalue <- matrix(0, K, m)
  for (k in 1:K) {
    pvalue[k, ] <- 2 * pnorm(-abs(Betah[[k]] / sqrt(diag(Sigmah[[k]]))))
  }

  zinit <- rep(0, K)
  for (j in 1:K) {
    index <- 1:m
    PVALUE <- p.adjust(pvalue[j, ])
    SIGNALS <- index[PVALUE < 0.05]
    modelf1 <- rep(0, m)
    modelf1[SIGNALS] <- 1
    if (max(modelf1) == 1) (zinit[j] <- 1)
  }


  RES <- CS(Betah, Sigmah,
    kappa0 = 0.5, tau20 = 1, zeta0 = zinit,
    m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1, a1 = 0.1, a2 = 0.1,
    c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames = snpnames, genename = genename
  )
  print(RES)


  RES1 <- CS(Betah, Sigmah,
    kappa0 = c(0.2, 0.5), tau20 = c(1, 2), zeta0 = zinit,
    m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
    a1 = 0.1, a2 = 0.1, c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames, genename
  )
  print(RES1)
 ################################### Gene PARP2 ########################################
```

```
library(BhGLM)
data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
genename <- "PARP2"
snpnames <- c("rs3093872", "rs3093921", "rs1713411", "rs3093926", "rs3093930", "rs878156")


Fit1 <- BhGLM::bglm(y1 ~ ., family = binomial(link = "logit"), data = Breast)
Betah1 <- Fit1$coefficients[-1]
Sigmah1 <- cov(coef(arm::sim(Fit1)))[-1, -1]

Fit2 <- BhGLM::bglm(y2 ~ ., family = binomial(link = "logit"), data = Thyroid)
Betah2 <- Fit2$coefficients[-1]
Sigmah2 <- cov(coef(arm::sim(Fit2)))[-1, -1]

Betah <- list(Betah1, Betah2)
Sigmah <- list(Sigmah1, Sigmah2)
K <- 2
m <- 6

pvalue <- matrix(0, K, m)
for (k in 1:K) {
  pvalue[k, ] <- 2 * pnorm(-abs(Betah[[k]] / sqrt(diag(Sigmah[[k]]))))
}

zinit <- rep(0, K)
for (j in 1:K) {
  index <- 1:m
  PVALUE <- p.adjust(pvalue[j, ])
  SIGNALS <- index[PVALUE < 0.05]
  modelf1 <- rep(0, m)
  modelf1[SIGNALS] <- 1
  if (max(modelf1) == 1) (zinit[j] <- 1)
}


RES <- CS(Betah, Sigmah,
  kappa0 = 0.5, tau20 = 1, zeta0 = zinit,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1, a1 = 0.1, a2 = 0.1,
  c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames = snpnames, genename = genename
)
print(RES)

RES1 <- CS(Betah, Sigmah,
  kappa0 = c(0.2, 0.5), tau20 = c(1, 2), zeta0 = zinit,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames, genename
)
print(RES1)
########### Simulated individual level data with K=3 and continuous phynotype ###########
library(BhGLM)
data(Simulated_individual)
```

```
Study1 <- Simulated_individual$Study1
Study2 <- Simulated_individual$Study2
Study3 <- Simulated_individual$Study3
K <- 3
m <- 30
genename <- "Simulated"
snpnames <- sprintf("SNP%s", seq(1:m))


Fit1 <- BhGLM::bglm(Y1 ~ ., family = gaussian, data = data.frame(Study1))
Betah1 <- Fit1$coefficients[-1]
Sigmah1 <- cov(coef(arm::sim(Fit1)))[-1, -1]

Fit2 <- BhGLM::bglm(Y2 ~ ., family = gaussian, data = data.frame(Study2))
Betah2 <- Fit2$coefficients[-1]
Sigmah2 <- cov(coef(arm::sim(Fit2)))[-1, -1]

Fit3 <- BhGLM::bglm(Y3 ~ ., family = gaussian, data = data.frame(Study3))
Betah3 <- Fit3$coefficients[-1]
Sigmah3 <- cov(coef(arm::sim(Fit3)))[-1, -1]

Betah <- list(Betah1, Betah2, Betah3)
Sigmah <- list(Sigmah1, Sigmah2, Sigmah3)


pvalue <- matrix(0, K, m)
for (k in 1:K) {
  pvalue[k, ] <- 2 * pnorm(-abs(Betah[[k]] / sqrt(diag(Sigmah[[k]]))))
}

zinit <- rep(0, K)
for (j in 1:K) {
  index <- 1:m
  PVALUE <- p.adjust(pvalue[j, ])
  SIGNALS <- index[PVALUE < 0.05]
  modelf1 <- rep(0, m)
  modelf1[SIGNALS] <- 1
  if (max(modelf1) == 1) (zinit[j] <- 1)
}

RES <- CS(Betah, Sigmah,
  kappa0 = 0.5, tau20 = 1, zeta0 = zinit,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1, a1 = 0.1, a2 = 0.1,
  c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames = snpnames, genename = genename
)
print(RES)

RES1 <- CS(Betah, Sigmah,
  kappa0 = c(0.2, 0.5), tau20 = c(1, 2), zeta0 = zinit,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames, genename
)
print(RES1)
```

```
########### Simulated individual level data with K=2 and gene expression data ###########
library(BhGLM)
data(Simulated_individual_survival)
Study1 <- Simulated_individual_survival$Study1
Study2 <- Simulated_individual_survival$Study2
K <- 2
m <- 10
genename <- "Simulated"
snpnames <- sprintf("G%s", seq(1:m))


Fit1 <- BhGLM::bcoxph(Study1$T ~ Study1$X)
Betah1 <- Fit1$coefficients
Sigmah1 <- Fit1$var


Fit2 <- BhGLM::bcoxph(Study2$T ~ Study2$X)
Betah2 <- Fit2$coefficients
Sigmah2 <- Fit2$var

Betah <- list(Betah1, Betah2)
Sigmah <- list(Sigmah1, Sigmah2)

pvalue <- matrix(0, K, m)
for (k in 1:K) {
  pvalue[k, ] <- 2 * pnorm(-abs(Betah[[k]] / sqrt(diag(Sigmah[[k]]))))
}

zinit <- rep(0, K)
for (j in 1:K) {
  index <- 1:m
  PVALUE <- p.adjust(pvalue[j, ])
  SIGNALS <- index[PVALUE < 0.05]
  modelf1 <- rep(0, m)
  modelf1[SIGNALS] <- 1
  if (max(modelf1) == 1) (zinit[j] <- 1)
}


RES1 <- CS(Betah, Sigmah,
  kappa0 = c(0.2, 0.5), tau20 = c(1, 2), zeta0 = zinit,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames, genename
)
print(RES1)

## End(Not run)
```

DNAJC1 *Gene DNAJC1 from BCAC and Epithyr studies*

Description

The summary statistics data for DNAJC1 protein coding gene including beta and standard error for pleiotropy investigation of breast and thyroid cancers. The summary statistics of breast and thyroid cancer are extracted from BCAC and Epithyr (Baghgalaki et al., 2021b) studies, respectively.

Usage

```
DNAJC1
```

Format

A list which contains two matrices for the summary statistics of each study.

Breast Summary statitics of breast cancer including the name of SNPs, beta and se

Thyroid Summary statitics of thyroid cancer including the name of SNPs, beta and se

References

1. Baghfalaki, T., Sugier, Y. Asgari, P. E., Truong, & Liquet, B. (2021). GCPBayes: An R Package for Studying Cross-Phenotype Genetic Associations with Group-Level Bayesian Meta-Analysis. *Submitted*.

See Also

GCPBayes

DS *Dirac Spike*

Description

Utilize a Gibbs sampler to conduct analysis on a multivariate Bayesian sparse group selection model, employing a Dirac spike prior to identify pleiotropic effects across traits. This function is tailored for summary statistics comprising estimated regression coefficients and their corresponding covariance matrices.

Usage

```
DS(
  Betah,
  Sigmah,
  kappa0,
  sigma20,
  m,
  K,
  niter = 1000,
  burnin = 500,
```

```
  nthin = 2,
  nchains = 2,
  a1 = 0.1,
  a2 = 0.1,
  d1 = 0.1,
  d2 = 0.1,
  snpnames,
  genename
)
```

Arguments

Betah	A list containing m-dimensional vectors of the regression coefficients for K studies.
Sigmah	A list containing the positive definite covariance matrices (m*m-dimensional) which is the estimated covariance matrices of K studies.
kappa0	Initial value for kappa (its dimension is equal to nchains).
sigma20	Initial value for sigma2 (its dimension is equal to nchains).
m	Number of variables in the group.
K	Number of traits.
niter	Number of iterations for the Gibbs sampler.
burnin	Number of burn-in iterations.
nthin	The lag of the iterations used for the posterior analysis is defined (or thinning rate).
nchains	Number of Markov chains, when nchains>1, the function calculates the Gelman-Rubin convergence statistic, as modified by Brooks and Gelman (1998).
a1, a2	Hyperparameters of kappa. Default is a1=0.1 and a2=0.1.
d1, d2	Hyperparameters of sigma2. Default is d1=0.1 and d2=0.1.
snpnames	Names of variables for the group.
genename	Name of group.

Details

Let betah_k, k=1,...,K be a m-dimensional vector of the regression coefficients for the kth study and Sigmah_k be its estimated covariance matrix. The hierarchical set-up of DS prior, by considering summary statistics (betah_k and Sigmah_k, k=1,...,K) as the input of the method, is given by:

betah _k ~ (1 - kappa) delta_0(betah_k) + kappa N_m(0,sigma2 I_m),

kappa ~ Beta(a_1,a_2),

sigma2 ~ inverseGamma (d_1,d_2).

where delta_0(betah_k) denotes a point mass at 0, such that delta_0(betah_k)=1 if beta_k=0 and delta_0(betah_k)=0 if at least one of the m components of beta_k is non-zero.

Value

- mcmcchain: The list of simulation output for all parameters.
- Summary: Summary statistics for regression coefficients in each study.
- Criteria: genename, snpnames, PPA, log10BF, lBFDR, theta.
- Indicator: A table containing m rows of binary indicators for each study, the number of studies with nonzero signal and having pleiotropic effect by credible interval (CI). The first K columns show nonzero signals, K+1 th column includes the number of studies with nonzero signal and the last column shows an indicator for having pleiotropic effect of each SNP.

Author(s)

Taban Baghfalaki.

References

1. Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. *Statistics in Medicine*, **40**(6), 1498-1518.

Examples

```
############################## Gene DNAJC1 ################################################
data(DNAJC1)
Breast <- DNAJC1$Breast
Thyroid <- DNAJC1$Thyroid
genename <- "DNAJC1"
snpnames <- Breast$snp
Betah <- list(Breast$beta, Thyroid$beta)
Sigmah <- list(diag(Breast$se^2), diag(Thyroid$se^2))
K <- 2
m <- 14

set.seed(123)
RES <- DS(Betah, Sigmah,
  kappa0 = 0.5, sigma20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)
## Not run:
  print(RES)

  RES1 <- DS(Betah, Sigmah,
    kappa0 = c(0.2, 0.5), sigma20 = c(1, 2),
    m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
    a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
  )
  print(RES1)
 ################### Simulated summary level data with K=5 ##############################
  data(Simulated_summary)
  genename <- Simulated_summary$genename
```

```
snpnames <- Simulated_summary$snpnames
Betah <- Simulated_summary$simBeta
Sigmah <- Simulated_summary$simSIGMA
K <- 5
m <- 10

RES <- DS(Betah, Sigmah,
  kappa0 = 0.5, sigma20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)
print(RES)

RES1 <- DS(Betah, Sigmah,
  kappa0 = c(0.2, 0.5), sigma20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)
print(RES1)
################################### Gene PARP2 ##########################################
library(BhGLM)
data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
genename <- "PARP2"
snpnames <- c("rs3093872", "rs3093921", "rs1713411", "rs3093926", "rs3093930", "rs878156")


Fit1 <- BhGLM::bglm(y1 ~ ., family = binomial(link = "logit"), data = Breast)
Betah1 <- Fit1$coefficients[-1]
Sigmah1 <- cov(coef(arm::sim(Fit1)))[-1, -1]


Fit2 <- BhGLM::bglm(y2 ~ ., family = binomial(link = "logit"), data = Thyroid)
Betah2 <- Fit2$coefficients[-1]
Sigmah2 <- cov(coef(arm::sim(Fit2)))[-1, -1]

Betah <- list(Betah1, Betah2)
Sigmah <- list(Sigmah1, Sigmah2)
K <- 2
m <- 6

RES <- DS(Betah, Sigmah,
  kappa0 = 0.5, sigma20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)
print(RES)

RES1 <- DS(Betah, Sigmah,
  kappa0 = c(0.2, 0.5), sigma20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
```

```
)
print(RES1)
############ Simulated individual level data with K=3 and continuous phynotype ############
library(BhGLM)
data(Simulated_individual)
Study1 <- Simulated_individual$Study1
Study2 <- Simulated_individual$Study2
Study3 <- Simulated_individual$Study3
K <- 3
m <- 30
genename <- "Simulated"
snpnames <- sprintf("SNP%s", seq(1:m))


Fit1 <- BhGLM::bglm(Y1 ~ ., family = gaussian, data = data.frame(Study1))
Betah1 <- Fit1$coefficients[-1]
Sigmah1 <- cov(coef(arm::sim(Fit1)))[-1, -1]


Fit2 <- BhGLM::bglm(Y2 ~ ., family = gaussian, data = data.frame(Study2))
Betah2 <- Fit2$coefficients[-1]
Sigmah2 <- cov(coef(arm::sim(Fit2)))[-1, -1]


Fit3 <- BhGLM::bglm(Y3 ~ ., family = gaussian, data = data.frame(Study3))
Betah3 <- Fit3$coefficients[-1]
Sigmah3 <- cov(coef(arm::sim(Fit3)))[-1, -1]

Betah <- list(Betah1, Betah2, Betah3)
Sigmah <- list(Sigmah1, Sigmah2, Sigmah3)


RES <- DS(Betah, Sigmah,
  kappa0 = 0.5, sigma20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)
print(RES)


RES1 <- DS(Betah, Sigmah,
  kappa0 = c(0.2, 0.5), sigma20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)
print(RES1)

############ Simulated individual level data with K=2 and gene expression data ############
library(BhGLM)
data(Simulated_individual_survival)
Study1 <- Simulated_individual_survival$Study1
Study2 <- Simulated_individual_survival$Study2
K <- 2
```

```
m <- 10
genename <- "Simulated"
snpnames <- sprintf("G%s", seq(1:m))


Fit1 <- BhGLM::bcoxph(Study1$T ~ Study1$X)
Betah1 <- Fit1$coefficients
Sigmah1 <- Fit1$var


Fit2 <- BhGLM::bcoxph(Study2$T ~ Study2$X)
Betah2 <- Fit2$coefficients
Sigmah2 <- Fit2$var

Betah <- list(Betah1, Betah2)
Sigmah <- list(Sigmah1, Sigmah2)



RES1 <- DS(Betah, Sigmah,
  kappa0 = c(0.2, 0.5), sigma20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)
print(RES1)

## End(Not run)
```

`e2_Monte_Carlo_EM` *Internal: e2_Monte_Carlo_EM*

Description

Internal: e2_Monte_Carlo_EM

Usage

```
e2_Monte_Carlo_EM(
  Betah,
  Sigmah,
  kappa0 = kappa0,
  kappastar0 = kappastar0,
  sigma20 = sigma20,
  s20 = s20,
  m,
  K,
  a1 = a1,
  a2 = a2,
  d1 = d1,
```

```
  d2 = d2,
  c1 = c1,
  c2 = c2,
  e2 = e2,
  snpnames,
  genename
)
```

Arguments

Betah	A matrix of dimension K*m represents the regression coefficients. Each row of this matrix includes the regression coefficients for each trait.
Sigmah	A symmetric block-diagonal matrix of dimension K*m is used. Each block of this matrix shows a positive definite covariance matrix which is an estimated covariance matrix of each trait.
kappa0	Initial value for kappa.
kappastar0	Initial value for kappastar.
sigma20	Initial value for sigma2.
s20	Initial value for s2.
m	Number of variables in the group.
K	Number of traits.
a1, a2	Hyperparameters of kappa. Default is a1=0.1 and a2=0.1.
d1, d2	Hyperparameters of sigma2. Default is d1=0.1 and d2=0.1.
c1, c2	Hyperparameters of kappastar. Default is c1=0.1 and c2=0.1.
e2	Initial value for doing Monte Carlo EM algorithm to estimate hyperparameter of s2.
snpnames	Names of variables for the group.
genename	Name of group.

GCPBayes *GCPBayes Package*

Description

Execute a Gibbs sampler to analyze a multivariate Bayesian sparse group selection model, incorporating Dirac, continuous, and hierarchical spike priors to detect pleiotropic effects across multiple traits. This package is tailored for summary statistics comprising estimated regression coefficients and their corresponding covariance matrices.

Author(s)

Taban Baghfalaki <t.baghfalaki@gmail.com>, <t.baghfalaki@modares.ac.ir>

References

1. Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. *Statistics in Medicine*, **40**(6), 1498-1518.
2. Baghfalaki, T., Sugier, Y. Asgari, P. E., Truong, & Liquet, B. (2023). GCPBayes: An R Package for Studying Cross-Phenotype Genetic Associations with Group-Level Bayesian Meta-Analysis. *RJournal*, **15**(1), 122-141.

See Also

Useful links:

- https://github.com/tbaghfalaki/GCPBayes

HS *Hierarchical Spike*

Description

Utilize a Gibbs sampler to conduct analysis on a multivariate Bayesian sparse group selection model, employing a hierarchical spike prior to identify pleiotropic effects across traits. This function is tailored for summary statistics comprising estimated regression coefficients and their corresponding covariance matrices.

Usage

```
HS(
  Betah,
  Sigmah,
  kappa0 = kappa0,
  kappastar0 = kappastar0,
  sigma20 = sigma20,
  s20 = s20,
  m,
  K,
  niter = 1000,
  burnin = 500,
  nthin = 2,
  nchains = 2,
  a1 = 0.1,
  a2 = 0.1,
  d1 = 0.1,
  d2 = 0.1,
  c1 = 1,
  c2 = 1,
  e2 = 1,
  snpnames,
```

```
  genename
)
```

Arguments

Betah	A list containing m-dimensional vectors of the regression coefficients for K studies.
Sigmah	A list containing the positive definite covariance matrices (m*m-dimensional) which is the estimated covariance matrices of K studies.
kappa0	Initial value for kappa (its dimension is equal to nchains).
kappastar0	Initial value for kappastar (its dimension is equal to nchains).
sigma20	Initial value for sigma2 (its dimension is equal to nchains).
s20	Initial value for s2 (its dimension is equal to nchains).
m	Number of variables in the group.
K	Number of traits.
niter	Number of iterations for the Gibbs sampler.
burnin	Number of burn-in iterations.
nthin	The lag of the iterations used for the posterior analysis is defined (or thinning rate).
nchains	Number of Markov chains, when nchains>1, the function calculates the Gelman-Rubin convergence statistic, as modified by Brooks and Gelman (1998).
a1, a2	Hyperparameters of kappa. Default is a1=0.1 and a2=0.1.
d1, d2	Hyperparameters of sigma2. Default is d1=0.1 and d2=0.1.
c1, c2	Hyperparameters of kappastar. Default is c1=0.1 and c2=0.1.
e2	Initial value for doing Monte Carlo EM algorithm to estimate hyperparameter of s2.
snpnames	Names of variables for the group.
genename	Name of group.

Details

For considering the HS prior, a reparameterization of betah_k is considered as betah_k = V_k^0.5 b_k, V_k^0.5 = diag(tau_k1,...,tau_km)$. Therfore, we have the following hirarchical model:

b_k ~ (1 - kappa) delta_0(b_k) + kappa N_m(0,sigma2 I_m),

tau_kj ~ (1 - kappa^*) delta_0(tau_kj) + kappa TN(0,s2),

kappa ~ Beta(a_1,a_2),

kappa^* ~ Beta(c_1,c_2),

sigma2 ~ inverseGamma (d_1,d_2).

s2 ~ inverseGamma (e_1,e_2).

where delta_0(betah_k) denotes a point mass at 0, such that delta_0(betah_k)=1 if beta_k=0 and delta_0(betah_k)=0 if at least one of the m components of beta_k is non-zero and TN(0,s2) denotes a univariate truncated normal distribution at zero with mean 0 and variance s2.

Value

- mcmcchain: The list of simulation output for all parameters.
- Summary: Summary statistics for regression coefficients in each study.
- Indicator 1: A table containing m rows of binary indicators for each study, the number of studies with nonzero signal and having pleiotropic effect by credible interval (CI). The first K columns show nonzero signals, K+1 th column includes the number of studies with nonzero signal and the last column shows an indicator for having pleiotropic effect of each SNP.
- Indicator 2: A table containing m rows of binary indicators for each study, the number of studies with nonzero signal and having pleiotropic effect by median. The first K columns show nonzero signals, K+1 th column includes the number of studies with nonzero signal and the last column shows an indicator for having pleiotropic effect of each SNP.

Author(s)

Taban Baghfalaki.

References

Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. Statistics in Medicine, 40(6), 1498-1518.

Examples

```
############################## PARP2_summary ##############################################
data(PARP2_summary)
Breast <- PARP2_summary$Breast
Thyroid <- PARP2_summary$Thyroid
Betah <- list(Breast$beta, Thyroid$beta)
Sigmah <- list(diag(Breast$se), diag(Thyroid$se))
genename <- "PARP2"
snpnames <- Breast$snp
K <- 2
m <- 6
set.seed(123)
RES <- HS(Betah, Sigmah,
  kappa0 = 0.5, kappastar0 = 0.5, sigma20 = 1, s20 = 1,
  m = m, K = K, niter = 800, burnin = 400, nthin = 1, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
## Not run:
  print(RES)
 ############################## Gene DNAJC1 ##############################################
  data(DNAJC1)
  Breast <- DNAJC1$Breast
  Thyroid <- DNAJC1$Thyroid
  genename <- "DNAJC1"
  snpnames <- Breast$snp
  Betah <- list(Breast$beta, Thyroid$beta)
  Sigmah <- list(diag(Breast$se^2), diag(Thyroid$se^2))
```

```
K <- 2
m <- 14

RES <- HS(Betah, Sigmah,
  kappa0 = 0.5, kappastar0 = 0.5, sigma20 = 1, s20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 1, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES)

RES1 <- HS(Betah, Sigmah,
  kappa0 = c(0.5, 0.3), kappastar0 = c(0.5, 0.3), sigma20 = c(2, 1), s20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES1)
################### Simulated summary level data with K=5 ##############################
data(Simulated_summary)
genename <- Simulated_summary$genename
snpnames <- Simulated_summary$snpnames
Betah <- Simulated_summary$simBeta
Sigmah <- Simulated_summary$simSIGMA
K <- 5
m <- 10

RES <- HS(Betah, Sigmah,
  kappa0 = 0.5, kappastar0 = 0.5, sigma20 = 1, s20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES)

RES1 <- HS(Betah, Sigmah,
  kappa0 = c(0.5, 0.3), kappastar0 = c(0.5, 0.3), sigma20 = c(2, 1), s20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES1)
################################### Gene PARP2 ##########################################
library(BhGLM)
data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
genename <- "PARP2"
snpnames <- c("rs3093872", "rs3093921", "rs1713411", "rs3093926", "rs3093930", "rs878156")


Fit1 <- BhGLM::bglm(y1 ~ ., family = binomial(link = "logit"), data = Breast)
Betah1 <- Fit1$coefficients[-1]
Sigmah1 <- cov(coef(arm::sim(Fit1)))[-1, -1]


Fit2 <- BhGLM::bglm(y2 ~ ., family = binomial(link = "logit"), data = Thyroid)
```

```
Betah2 <- Fit2$coefficients[-1]
Sigmah2 <- cov(coef(arm::sim(Fit2)))[-1, -1]

Betah <- list(Betah1, Betah2)
Sigmah <- list(Sigmah1, Sigmah2)
K <- 2
m <- 6


RES <- HS(Betah, Sigmah,
  kappa0 = 0.5, kappastar0 = 0.5, sigma20 = 1, s20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES)

RES1 <- HS(Betah, Sigmah,
  kappa0 = c(0.5, 0.3), kappastar0 = c(0.5, 0.3), sigma20 = c(2, 1), s20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES1)
########### Simulated individual level data with K=3 and continuous phynotype ###########
library(BhGLM)
data(Simulated_individual)
Study1 <- Simulated_individual$Study1
Study2 <- Simulated_individual$Study2
Study3 <- Simulated_individual$Study3
K <- 3
m <- 30
genename <- "Simulated"
snpnames <- sprintf("SNP%s", seq(1:m))


Fit1 <- BhGLM::bglm(Y1 ~ ., family = gaussian, data = data.frame(Study1))
Betah1 <- Fit1$coefficients[-1]
Sigmah1 <- cov(coef(arm::sim(Fit1)))[-1, -1]


Fit2 <- BhGLM::bglm(Y2 ~ ., family = gaussian, data = data.frame(Study2))
Betah2 <- Fit2$coefficients[-1]
Sigmah2 <- cov(coef(arm::sim(Fit2)))[-1, -1]


Fit3 <- BhGLM::bglm(Y3 ~ ., family = gaussian, data = data.frame(Study3))
Betah3 <- Fit3$coefficients[-1]
Sigmah3 <- cov(coef(arm::sim(Fit3)))[-1, -1]

Betah <- list(Betah1, Betah2, Betah3)
Sigmah <- list(Sigmah1, Sigmah2, Sigmah3)


RES <- HS(Betah, Sigmah,
```

```
  kappa0 = 0.5, kappastar0 = 0.5, sigma20 = 1, s20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES)

RES1 <- HS(Betah, Sigmah,
  kappa0 = c(0.5, 0.3), kappastar0 = c(0.5, 0.3), sigma20 = c(2, 1), s20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES1)

########### Simulated individual level data with K=2 and gene expression data ###########
library(BhGLM)
data(Simulated_individual_survival)
Study1 <- Simulated_individual_survival$Study1
Study2 <- Simulated_individual_survival$Study2
K <- 2
m <- 10
genename <- "Simulated"
snpnames <- sprintf("G%s", seq(1:m))


Fit1 <- BhGLM::bcoxph(Study1$T ~ Study1$X)
Betah1 <- Fit1$coefficients
Sigmah1 <- Fit1$var


Fit2 <- BhGLM::bcoxph(Study2$T ~ Study2$X)
Betah2 <- Fit2$coefficients
Sigmah2 <- Fit2$var

Betah <- list(Betah1, Betah2)
Sigmah <- list(Sigmah1, Sigmah2)

RES1 <- HS(Betah, Sigmah,
  kappa0 = c(0.5, 0.3), kappastar0 = c(0.5, 0.3), sigma20 = c(2, 1), s20 = c(1, 2),
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
print(RES1)

## End(Not run)
```

MCMCplot *MCMC plot*

Description

Trace plot, density plot and ACF plot for the output of CS/DS/HS. The plot is able to draw at most ten SNPs.

Usage

```
MCMCplot(
  Result = Result,
  k = k,
  nchains = nchains,
  whichsnps = whichsnps,
  betatype = "l",
  acftype = "correlation",
  dencol = "white",
  denlty = 1,
  denbg = "white"
)
```

Arguments

Result	All the generated results by CS/DS/HS function.
k	The number of study for drawing plots, k=1,2,...,K.
nchains	Number of Markov chains run in Result.
whichsnps	The name of SNPs.
betatype	The type of plot desired. The following values are possible: "p" for points, "l" for lines, "b" for both points and lines, "c" for empty points joined by lines, "o" for overplotted points and lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
acftype	String giving the type of ACF to be computed. Allowed values are "correlation" (the default), "covariance" or "partial". Will be partially matched.
dencol	The color for filling the density plot.
denlty	The line type to be used in the density plot.
denbg	The color to be used for the background of the density plot.

Details

Trace plot, density plot and ACF plot for the output of CS/DS/HS for checking convergence of MCMC chains.

Author(s)

Taban Baghfalaki.

References

Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. Statistics in Medicine, 40(6), 1498-1518.

Examples

```
############################Gene DNAJC1 ##############################################
data(DNAJC1)
Breast <- DNAJC1$Breast
Thyroid <- DNAJC1$Thyroid
genename <- "DNAJC1"
snpnames <- Breast$snp
Betah <- list(Breast$beta, Thyroid$beta)
Sigmah <- list(diag(Breast$se^2), diag(Thyroid$se^2))
K <- 2
m <- 14


RES1 <- DS(Betah, Sigmah,
           kappa0 = 0.5, sigma20 = 1,
           m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
           a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)


MCMCplot(Result = RES1, k = 2, nchains = 1, whichsnps = sample(snpnames, 7),
                    betatype = "1",
                    acftype = "correlation",
                    dencol = "white", denlty = 1, denbg = "white")
###################Simulated summary level data with K=5 ###############################
## Not run:
data(Simulated_summary)
genename <- Simulated_summary$genename
snpnames <- Simulated_summary$snpnames
Betah <- Simulated_summary$simBeta
Sigmah <- Simulated_summary$simSIGMA
K <- 5
m <- 10

RES1 <- DS(Betah, Sigmah,
 kappa0 = c(0.2, 0.5), sigma20 = c(1, 2),
 m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 2,
 a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename)

MCMCplot(Result = RES1, k = 3, nchains = 2, whichsnps = sample(snpnames, 3),
         betatype = "1",
         acftype = "partial",
         dencol = "blue", denlty = 1, denbg = "black")

RES1 <- DS(Betah, Sigmah,
 kappa0 = c(0.2, 0.5, 0.6), sigma20 = c(1, 2, 1.5),
 m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 3,
 a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename)

MCMCplot(Result = RES1, k = 3, nchains = 3, whichsnps = sample(snpnames, 5),
         betatype = "1",
         acftype = "partial",
```

```
         dencol = "white", denlty = 1, denbg = "white")
##############################Gene DNAJC1 ################################################
pvalue <- matrix(0, K, m)
for (k in 1:K) {
  pvalue[k, ] <- 2 * pnorm(-abs(Betah[[k]] / sqrt(diag(Sigmah[[k]]))))
}

zinit <- rep(0, K)
for (j in 1:K) {
  index <- 1:m
  PVALUE <- p.adjust(pvalue[j, ])
 SIGNALS <- index[PVALUE < 0.05]
 modelf1 <- rep(0, m)
 modelf1[SIGNALS] <- 1
  if (max(modelf1) == 1) (zinit[j] <- 1)
}

RES <- CS(Betah, Sigmah,
  kappa0 = 0.5, tau20 = 1, zeta0 = zinit,
 m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1, a1 = 0.1, a2 = 0.1,
c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames = snpnames, genename = genename)

MCMCplot(Result = RES1, k = 1, nchains = 1, whichsnps = sample(snpnames, 7),
         betatype = "1",
         acftype = "correlation",
         dencol = "white", denlty = 1, denbg = "white")
###################################Gene PARP2 ##########################################
library(BhGLM)
data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
genename <- "PARP2"
snpnames <- c("rs3093872", "rs3093921", "rs1713411", "rs3093926", "rs3093930", "rs878156")


Fit1 <- BhGLM::bglm(y1~ ., family=binomial(link="logit"),data=Breast)
Betah1 <-  Fit1$coefficients[-1]
Sigmah1 <- cov(coef(arm::sim(Fit1)))[-1,-1]

Fit2 <- BhGLM::bglm(y2~ ., family=binomial(link="logit"),data=Thyroid)
Betah2 <-  Fit2$coefficients[-1]
Sigmah2 <- cov(coef(arm::sim(Fit2)))[-1,-1]

Betah <- list(Betah1,Betah2)
Sigmah <- list(Sigmah1,Sigmah2)
K <- 2
m <- 6


RES1 <- DS(Betah, Sigmah, kappa0=c(0.2,0.5), sigma20=c(1,2),
          m=m, K=K, niter=1000, burnin=500, nthin=1, nchains=2,
          a1=0.1, a2=0.1, d1=0.1, d2=0.1, snpnames, genename)
```

```
MCMCplot(Result=RES1, k=1, nchains=2, whichsnps=snpnames,
         betatype = "l",
         acftype = "correlation",
         dencol = "red", denlty = 1, denbg = "white")


RES1 <- DS(Betah, Sigmah, kappa0=c(0.2,0.5), sigma20=c(1,2),
           m=m, K=K, niter=2000, burnin=1000, nthin=2, nchains=2,
           a1=0.1, a2=0.1, d1=0.1, d2=0.1, snpnames, genename)

MCMCplot(Result=RES1, k=1, nchains=2, whichsnps=snpnames,
         betatype = "l",
         acftype = "correlation",
         dencol = "white", denlty = 1, denbg = "white")

## End(Not run)
```

PARP2 *Gene PARP2 from CECILE study*

Description

A list containing the individual level data for gene PARP2 including genotypes and phenotypes for pleiotropy investigation of breast and thyroid cancers. It is from CECILE study, a French population-based case-control study on breast cancer and from the French studies included in the EPITHYR consortium on thyroid cancer.

Usage

```
PARP2
```

Format

A list which contains two matrices.

Breast Summary statitics of breast cancer including the name of SNPs, beta and se

Thyroid Summary statitics of thyroid cancer including the name of SNPs, beta and se

References

1. Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. *Statistics in Medicine*, **40**(6), 1498-1518.
2. Baghfalaki, T., Sugier, Y. Asgari, P. E., Truong, & Liquet, B. (2021). GCPBayes: An R Package for Studying Cross-Phenotype Genetic Associations with Group-Level Bayesian Meta-Analysis. *Submitted*.

See Also

GCPBayes

PARP2_summary *Summary statistics of gene PARP2 from CECILE study*

Description

The summary statistics data for PARP2 protein coding gene including beta and standard error for pleiotropy investigation of breast and thyroid cancers.

Usage

```
PARP2_summary
```

Format

A list which contains two matrices for the summary statistics of each study.

Breast Summary statitics of breast cancer including the name of SNPs, beta and se

Thyroid Summary statitics of thyroid cancer including the name of SNPs, beta and se

References

1. Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. *Statistics in Medicine*, **40**(6), 1498-1518.
2. Baghfalaki, T., Sugier, Y. Asgari, P. E., Truong, & Liquet, B. (2021). GCPBayes: An R Package for Studying Cross-Phenotype Genetic Associations with Group-Level Bayesian Meta-Analysis. *Submitted*.

See Also

GCPBayes

Simulated_individual *Simulated individual level data*

Description

A list containing the individual level data including genotypes and phenotypes for pleiotropy investigation of three studies.

Usage

```
Simulated_individual
```

Format

A list which contains the name of gene, the name of the SNPs, the vectors and the matrices for the summary statistics of each study.

Study1 The inividual level data including genotypes and phenotypes of Study 1

Study2 The inividual level data including genotypes and phenotypes of Study 2

Study3 The inividual level data including genotypes and phenotypes of Study 3

References

,

Baghfalaki, T., Sugier, Y. Asgari, P. E., Truong, & Liquet, B. (2021). GCPBayes: An R Package for Studying Cross-Phenotype Genetic Associations with Group-Level Bayesian Meta-Analysis. *Submitted*.

See Also

1. GCPBayes

Simulated_individual_survival *Simulated individual level survival data*

Description

A list containing the individual level data including gene expression data for survival outcomes for pleiotropy investigation of two studies.

Usage

```
Simulated_individual_survival
```

Format

A list which contains the name of gene, the name of the SNPs, the vectors and the matrices for the summary statistics of each study.

Study1 The inividual level data including survival outcomes and gene expression data of Study 1

Study2 The inividual level data including survival outcomes and gene expression data of Study 2

References

,

Baghfalaki, T., Sugier, Y. Asgari, P. E., Truong, & Liquet, B. (2021). GCPBayes: An R Package for Studying Cross-Phenotype Genetic Associations with Group-Level Bayesian Meta-Analysis. *Submitted*.

See Also

1. GCPBayes

`Simulated_summary` *Simulated summary statistics for K=5 traits*

Description

A list containing the summary statistics including regression coefficients and covariance matrices for K=5 studies.

Usage

```
Simulated_summary
```

Format

A list which contains the name of gene, the name of the SNPs, the vectors and the matrices for the summary statistics of each study.

genename The name of gene

snpnames The name of the SNPs

simBeta The regression coeffiecnts of the studies

simSIGMA The covariance matrices for the studies

References

,

Baghfalaki, T., Sugier, Y. Asgari, P. E., Truong, & Liquet, B. (2021). GCPBayes: An R Package for Studying Cross-Phenotype Genetic Associations with Group-Level Bayesian Meta-Analysis. *Submitted*.

See Also

1. GCPBayes

---

summaryCS *Summary function of Continuous Spike*

Description

The summaryCS function serves as a versatile tool for generating result summaries derived from the output of the CS function.

Usage

```
summaryCS(object)
```

Arguments

object	a result of a call to CS

Value

- Name of Gene: the component from object.
- Number of SNPs: the component from object.
- Name of SNPs: the component from object.
- log10BF: the component from object.
- lBFDR: the component from object.
- theta: the component from object.
- Significance based on CI: the component from object.

Author(s)

Taban Baghfalaki.

References

Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. Statistics in Medicine, 40(6), 1498-1518.

Examples

```
############################## Gene DNAJC1 ################################################
data(DNAJC1)
Breast <- DNAJC1$Breast
Thyroid <- DNAJC1$Thyroid
genename <- "DNAJC1"
snpnames <- Breast$snp
Betah <- list(Breast$beta, Thyroid$beta)
Sigmah <- list(diag(Breast$se^2), diag(Thyroid$se^2))
K <- 2
m <- 14

pvalue <- matrix(0, K, m)
for (k in 1:K) {
  pvalue[k, ] <- 2 * pnorm(-abs(Betah[[k]] / sqrt(diag(Sigmah[[k]]))))
}

zinit <- rep(0, K)
for (j in 1:K) {
  index <- 1:m
  PVALUE <- p.adjust(pvalue[j, ])
  SIGNALS <- index[PVALUE < 0.05]
  modelf1 <- rep(0, m)
  modelf1[SIGNALS] <- 1
  if (max(modelf1) == 1) (zinit[j] <- 1)
}

set.seed(123)
RES <- CS(Betah, Sigmah,
  kappa0 = 0.5, tau20 = 1, zeta0 = zinit,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1, a1 = 0.1, a2 = 0.1,
  c1 = 0.1, c2 = 0.1, sigma2 = 10^-3, snpnames = snpnames, genename = genename
)
summaryCS(RES)
```

`summaryDS` *Summary function of Dirac Spike*

Description

The summaryDS function serves as a versatile tool for generating result summaries derived from the output of the DS function.

Usage

```
summaryDS(object)
```

Arguments

`object`	a result of a call to DS

Value

- Name of Gene: the component from object.
- Number of SNPs: the component from object.
- Name of SNPs: the component from object.
- log10BF: the component from object.
- lBFDR: the component from object.
- theta: the component from object.
- Significance based on CI: the component from object.
- Significance based on median thresholding: the component from object.

Author(s)

Taban Baghfalaki.

References

Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. Statistics in Medicine, 40(6), 1498-1518.

Examples

```
############################## Gene DNAJC1 ################################################
data(DNAJC1)
Breast <- DNAJC1$Breast
Thyroid <- DNAJC1$Thyroid
genename <- "DNAJC1"
snpnames <- Breast$snp
Betah <- list(Breast$beta, Thyroid$beta)
Sigmah <- list(diag(Breast$se^2), diag(Thyroid$se^2))
K <- 2
m <- 14

set.seed(123)
RES <- DS(Betah, Sigmah,
  kappa0 = 0.5, sigma20 = 1,
  m = m, K = K, niter = 2000, burnin = 1000, nthin = 2, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, snpnames, genename
)
summaryDS(RES)
```

summaryHS *Summary function of Hierarchical Spike*

Description

The summaryHS function serves as a versatile tool for generating result summaries derived from the output of the HS function.

Usage

```
summaryHS(object)
```

Arguments

object	a result of a call to HS

Value

- Name of Gene: the component from object.
- Number of SNPs: the component from object.
- Name of SNPs: the component from object.
- Pleiotropic effect based on CI: the component from object.
- Pleiotropic effect based on median thresholding: the component from object.

Author(s)

Taban Baghfalaki.

References

Baghfalaki, T., Sugier, P. E., Truong, T., Pettitt, A. N., Mengersen, K., & Liquet, B. (2021). Bayesian meta analysis models for cross cancer genomic investigation of pleiotropic effects using group structure. Statistics in Medicine, 40(6), 1498-1518.

Examples

```
# PARP2_summary
data(PARP2_summary)
Breast <- PARP2_summary$Breast
Thyroid <- PARP2_summary$Thyroid
Betah <- list(Breast$beta, Thyroid$beta)
Sigmah <- list(diag(Breast$se), diag(Thyroid$se))
genename <- "PARP2"
snpnames <- Breast$snp
K <- 2
m <- 6
```

```
set.seed(123)
RES <- HS(Betah, Sigmah,
  kappa0 = 0.5, kappastar0 = 0.5, sigma20 = 1, s20 = 1,
  m = m, K = K, niter = 800, burnin = 400, nthin = 1, nchains = 1,
  a1 = 0.1, a2 = 0.1, d1 = 0.1, d2 = 0.1, c1 = 1, c2 = 1, e2 = 1, snpnames, genename
)
summaryHS(RES)
```

Index

∗ **CS**
    GCPBayes, 15
∗ **DS**
    GCPBayes, 15
∗ **HS**
    GCPBayes, 15
∗ **MCMCplot**
    GCPBayes, 15
∗ **datasets**
    DNAJC1, 8
    PARP2, 25
    PARP2_summary, 26
    Simulated_individual, 27
    Simulated_individual_survival, 27
    Simulated_summary, 28
∗ **summaryCS**
    GCPBayes, 15
∗ **summaryDS**
    GCPBayes, 15
∗ **summaryHS**
    GCPBayes, 15

CS, 2

DNAJC1, 8
DS, 9

e2_Monte_Carlo_EM, 14

GCPBayes, *9*, 15, *26–29*
GCPBayes-package *(*GCPBayes*)*, 15

HS, 16

MCMCplot, 21

PARP2, 25
PARP2_summary, 26

Simulated_individual, 27
Simulated_individual_survival, 27
Simulated_summary, 28
summaryCS, 29
summaryDS, 30
summaryHS, 32