GCPBayes: An R package for studying Cross-Phenotype Genetic Associations with Group-level Bayesian Meta-Analysis

by Taban Baghfalaki, Pierre-Emmanuel Sugier, Yazdan Asgari, Thérèse Truong, and Benoit Liquet

Abstract Several R packages have been developed to study cross-phenotypes associations (or pleiotropy) at SNP-level based on summary statistics data from genome-wide association study (GWAS). However, none of them allows to consider an underlying group structure of the data. We developed an R package, entitled **GCPBayes** (Group level Bayesian Meta-Analysis for Studying Cross-Phenotype Genetic Associations) introduced by Baghfalaki et al. (2021) that implements continuous and Dirac spike priors for group selection, and also a Bayesian sparse group selection approach with hierarchical spike and slab priors to select important variables at the group level and within the groups. The methods use summary statistics data from association studies or individual level data as inputs, and perform Bayesian meta-analysis approaches across multiple phenotypes to detect pleiotropy at both group-level (e.g. at the gene or pathway level) and within group (e.g. at the SNP level).

Introduction

Genome-wide association study (GWAS) has identified many cross-phenotypes associations underlying the evidence that pleiotropy, the fact that a single genetic variant, usually a Single-Nucleotide Polymorphism (SNP), is affecting multiple traits, is a widespread phenomenon in human complex traits (Watanabe et al. 2019). As a result, pleiotropy has attracted a great deal of attention among genetic epidemiologists (see, Solovieff et al. 2013; Verma et al. 2016; Hackinger and Zeggini 2017; Liu and Lin 2018; Majumdar et al. 2018; Krapohl et al. 2018; Lu et al. 2017; Trochet et al. 2019; Ray and Chatterjee 2020; Baghfalaki et al. 2021; Broc, Truong, and Liquet 2021), and several R packages have been developed to perform meta-analysis methods adapted for cross-phenotype association detection based on summary statistics data from GWAS as inputs (i.e, estimated effect size, standard error and p-value of the association between each SNP and a trait)(Ray and Chatterjee 2020). It should be noted that pleiotropy is only one possible explanation of cross-phenotype association, but for simplicity we will use the term pleiotropy for cross phenotype association in this paper. We detailed below a selection of methods that are most commonly used to detect pleiotropy between several traits.

ASSET (Bhattacharjee et al. 2012) is a frequentist method which is an extension of standard fixed effects meta-analysis that considers the effects of a variable (such as a SNP) in each study (representing a phenotype) to be in either the same direction or opposite directions allowing detection of antagonistic pleiotropic effects. This approach uses summary statistics data and does not take into account group structure. In the case of GWAS, the outputs of this approach are a p-value of global association of each SNP with all studies (or phenotypes) and an optimal subset of non-null studies or phenotypes that are associated with each SNP.

GPA (Chung et al. 2014) implements a flexible statistical framework for the joint analysis of multiple GWAS and its integration with various genetic and genomic data. It implements a flexible parametric mixture modeling approach for such integrative analysis and also provides hypothesis testing procedures for pleiotropy and annotation enrichment. **PLACO** is a recently created R package using GWAS summary statistics data to identify pleiotropic signals between two traits (Ray and Chatterjee 2020). The computed method derives a composite null hypothesis, that at most one trait is associated with the genetic variant vs the alternative that both traits are associated, based on the product of the Z-statistics of the genetic variants across two studies. **CPBayes** (Majumdar et al. 2018) is a Bayesian meta-analysis approach which uses univariate spike and slab prior and performs the MCMC technique via a Gibbs sampler. **CPBayes** uses the summary statistics data for a SNP across multiple traits. Two different measures are estimated for evaluating the overall pleiotropic association: the local false discovery rate and the Bayes factor. The optimal subset of associated traits underlying a pleiotropic signal is defined as the maximum a posteriori (MAP) estimate. The marginal trait-specific posterior probability of association (PPA), the direction of associations, and the credible interval of true genetic effects, are some examples of additional insights into pleiotropic signals that are provided by **CPBayes**.

All packages developed for these summary statistics-based methods are only designed to test pleiotropy at variable-level (SNP-level). Though, the structure of the common mechanisms shared by multiple phenotypes can be more complex as for example, different variants in the same locus can be associated with multiple traits, affecting the same gene and therefore can have an impact on the same.

Thus, extending CPBayes to the gene or pathway level which takes into account the group structure of the data could provide additional power to detect pleiotropic signals between multiple diseases. By incorporating prior biological information in GWAS such as group structure information (gene or pathway), our approaches could uncover new pleiotropic signals (Li et al. 2020; Baghfalaki et al. 2021).

In this paper, we present the GCPBayes R package (Bayesian meta-analysis of pleiotropic effects using group structure) for studying pleiotropy between multiples phenotypes using GWAS data by considering group structure information from prior biological knowledge. This package is able to consider and detect pleiotropy at both variable-level and group-level. The inputs of the developed functions are SNP-level summary statistics data derived from GWAS, by considering all the estimated regression coefficients of the variables (SNPs) in a group (a gene or a pathway) and its covariance matrix.

The methods that can be performed by the package offer a Bayesian paradigm using multivariate spike and slab priors for group-level pleiotropy using either continuous spike (CS, George and McCulloch 1993) or Dirac spike (DS, Mitchell and Beauchamp 1988) formulations. Also, a hierarchical sparsity prior (HS, Xu, Ghosh, et al. 2015) using two levels of Dirac spike and slab components to achieve group and within-group selection can be applied. In the package, both tests for the global null hypothesis of no association and for detection of pleiotropy at group-level and variable-level are considered. The free RStudio interface is advised to use GCPBayes in the R environment. The GCPBayes package can be installed by typing:

```
install.packages("GCPBayes")
```

To be used, the GCPBayes package has to be loaded each time R or RStudio are opened, via the following code:

```
library(GCPBayes)
```

The paper is organized as follows. The next section ("Data structure") describes how a data should look like in order to be used by functions of the GCPBayes package. The section "Usage" presents the different functions of GCPBayes. Some details for using different case studies, and a practical guideline for detection of pleiotropic effects are provided. A "Guidelines" section provides a statistical inferences pipeline and some recommendations to perform the successive stages of a pleiotropy analysis with GCPBayes. The computational time of the functions of the GCPBayes package is explored by using real data in the "Computational time for GCPBayes" section. Finally, in the "Concluding remarks" section, we discuss some remarks and limitations of GCPBayes. Besides, future works for a newer version of the package are discussed at the end of the section. In addition, the paper includes three appendices. The first is "Material and method", where notations and models are described briefly. The last appendix describes the design of the simulated data and their corresponding R codes which are also embedded in the package.

Data structure

GCPBayes is designed to work on summary statistics level data. Though, summary statistics can be derived from individual level data when available, allowing to take into account correlation between variables to improve the accuracy of the method.

It should be noted that GCPBayes is not restricted to a specific type of an outcome. Summary statistics data can be estimated through various kinds of models (linear, logistic, Cox, etc) and based on different types of phenotypes (categorical or continuous traits). Therefore, it is possible to use the GCPBayes package for genetic studies, gene expression analysis and other omics studies.

In this paper, we will illustrate the use of our method on genetic studies on breast and thyroid cancers. We will explore the cross phenotype association between these two traits and SNPs located in one gene in particular, *PARP2*. So, this section includes two different scenarios; first part deals with an input of summary statistics data of the association between breast cancer risk or thyroid cancer risk and SNPs located in the *PARP2* gene (Example 1). The second part considers the individual level data from which the summary statistics of Example 1 were derived as input, then exploring multicollinearity through the data (Example 2), and computes summary statistics and the covariance matrix from the individual level data for *PARP2* gene (Example 3).

We will consider the estimated regression coefficients and their covariance matrices from all studies:

$$\hat{\beta}_k,\ \hat{\Sigma}_k = \text{cov}(\hat{\beta}_k),\ \ k = 1, \cdots, K.$$

where $\hat{\beta}_k$ is a $m-$dimensional vector of the regression coefficients of the k^{th} study for a specified gene (or pathway) and $\hat{\Sigma}_k$ is its corresponding covariance matrix. If only summary statistics are available, the covariance matrix is replaced by a diagonal matrix with estimated variance of the regression coefficients.

Summary statistics level data

GWAS summary statistics data are available through various public databases such as GWAS Catalog. In most cases, a summary statistics data includes effect size estimate (beta), standard error and p-value for each SNP. The **GCPBayes** package accepts inputs related to beta and standard errors as lists: a list of regression coefficients (betas) and another list containing the matrices of the computed estimates of the variance of the regression coefficients.

Example 1: Inputs of GCPBayes using summary statistics from two case-control studies (on breast and thyroid cancers) for *PARP2* gene GWAS summary statistics data are available through various public databases such as GWAS Catalog. In most cases, a summary statistics data includes effect size estimate (beta), standard error and p-value for each SNP. The **GCPBayes** package accepts inputs related to beta and standard errors as lists: a list of regression coefficients (betas) and another list containing the matrices of the computed estimates of the variance of the regression coefficients.

```
library(GCPBayes)
data(PARP2_summary)
Breast <- PARP2_summary$Breast
Thyroid <- PARP2_summary$Thyroid
genename <- "PARP2"
snpnames <- rownames(Breast)
Betah <- list(Breast$beta,Thyroid$beta)
Sigmah <- list(diag(Breast$se^2),diag(Thyroid$se^2))
print(Betah,digits=2)

#> [[1]]
#> [1] -0.081  0.073 -0.346 -0.222  0.095  0.166
#>
#> [[2]]
#> [1] -0.033 -0.470  0.398  0.276  0.160  0.040

print(Sigmah,digits=2)

#> [[1]]
#>       [,1]  [,2]  [,3]  [,4]   [,5]  [,6]
#> [1,] 0.029 0.000 0.000 0.000 0.0000 0.000
#> [2,] 0.000 0.056 0.000 0.000 0.0000 0.000
#> [3,] 0.000 0.000 0.061 0.000 0.0000 0.000
#> [4,] 0.000 0.000 0.000 0.031 0.0000 0.000
#> [5,] 0.000 0.000 0.000 0.000 0.0064 0.000
#> [6,] 0.000 0.000 0.000 0.000 0.0000 0.051
#>
#> [[2]]
#>       [,1] [,2]  [,3] [,4]   [,5]  [,6]
#> [1,] 0.031 0.00 0.000 0.00 0.0000 0.000
#> [2,] 0.000 0.16 0.000 0.00 0.0000 0.000
#> [3,] 0.000 0.00 0.068 0.00 0.0000 0.000
#> [4,] 0.000 0.00 0.000 0.04 0.0000 0.000
#> [5,] 0.000 0.00 0.000 0.00 0.0063 0.000
#> [6,] 0.000 0.00 0.000 0.00 0.0000 0.043
```

This gene included six SNPs (rs3093872, rs3093921, rs1713411, rs3093926, rs3093930, and rs878156). So, the Betah list must include summary statistics of six elements for every study and Sigmah list must contain diagonal matrix of $[6 \times 6]$ for each study. A user could check the data using a "print" function.

Individual level data

While individual level data are available, the summary statistics should be computed by using external packages. Here, we explain about the available approaches and R packages for computing the summary statistics. Let $Y_1, \cdots, Y_K$ be the response variables of K studies such that $Y_k = (Y_{k1}, \cdots, Y_{kn_k})'$ denotes the n_k observations of the corresponding phenotype, $k = 1, \cdots, K$. For study k, let the genetic information be structured into G groups. For example this can be a set of SNPs belonging to a gene, or

SNPs belonging to a set of genes acting together in the same biological pathway. We denote Z_{kg} as a $n_k \times m_g-$ matrix of the m_g covariates for group g, $g = 1, 2, \cdots, G$. As all inferences are performed gene by gene, we remove the index g and consider m as the number of variables in each specific group for simplicity. Considering one of the groups, we assume that a generalized linear model (GLM, McCullagh 2019) is fitted separately for each study as follows:

$$\eta\left(\mathrm{E}\left(Y_k\right)\right) = \alpha_k 1_{n_k} + Z_k \beta_k, \quad k = 1, \cdots, K, \tag{1}$$

where $\eta(\cdot)$ is a link function, α_k is the intercept of the model and 1_{n_k} is a vector of n_k ones. Here $\beta_k = (\beta_{k1}, \cdots, \beta_{km})'$ denotes the m-dimensional regression coefficients for the group and the k^{th} study.

The "glm" function from R can be applied to fit generalized linear models and so to get the summary statistics or parameter estimation of the model 1 (Dunn and Smyth 2018). If the phenotype is binary then 1 is the usual logistic or probit model (Agresti 2018). If the phenotype is continuous, the model 1 is reduced to multiple linear regression and can also be fitted by the "lm" function from R that will provide the same results as the "glm" function. The "vcov" function can be applied to get the estimated covariance matrix of the regression coefficients, where the "glm" and "lm" functions calculates standard errors which yield only the diagonal elements.

Though, another strategy should be considered in the case of multicollinearity. Multicollinearity is the existence of near-linear relationships among variables of a group (Malo, Libiger, and Schork 2008). This phenomenon is widely spread in genetic data were non-random association of alleles at different loci in a given population are frequent, introducing large structures of correlation between SNPs. It is known as linkage disequilibrium (LD). This can create inaccurate estimates of the regression coefficients, and also inflate the standard errors of the regression coefficients (Saleh, Arashi, and Kibria 2019). This phenomenon can be visualized by drawing pairwise scatter plots of variables or to consider the correlation matrix. The variance inflation factor (VIF) can be used to asses the presence multicollinearity (Fox and Weisberg 2018). It can be computed by the "vif" function of car package of R. Most research papers consider a VIF > 10 as an indicator of multicollinearity (Menard 2002; Johnston, Jones, and Manley 2018; Gareth et al. 2013; Vittinghoff et al. 2011), but some choose a more conservative threshold of 5 Gareth et al. (2013) or even 2.5 (Johnston, Jones, and Manley 2018).

Example 2: Exploring multicollinearity/linkage disequilibrium in *PARP2* gene using individual level data In this example, we show the existence of multicollinearity in the data. Here, we consider the individual level data from which the summary statistics of the Example 1 was derived. The summary statistics and the corresponding VIFs for both studies could be obtained using the usual "glm" function as follow:

```
library(GCPBayes)
library(car)

#> Loading required package: carData

data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
Fit1 <- glm(y1~ ., family=binomial(link="logit"),data=Breast)
print(vif(Fit1))

#> rs3093872 rs3093921 rs1713411 rs3093926 rs3093930  rs878156
#>  1.696966  1.584775  3.095049  2.700594  1.347861  4.438011

Fit2 <- glm(y2~ ., family=binomial(link="logit"),data=Thyroid)
print(vif(Fit2))

#> rs3093872 rs3093921 rs1713411 rs3093926 rs3093930  rs878156
#>  1.565507  1.478632  3.552804  3.465883  1.397995  5.490866
```

A high VIF value (>5) for one SNP indicates multicollinearity in the data. Hence, a user should avoid to consider the "glm" function to obtain correct estimates of regression coefficients and their standard errors. Thereby, a user could perform Ridge regression (Hilt and Seegrist 1977). Although many R packages are available to perform Ridge regression, only **lrmest**, **ridge** and **lmridge** can estimate the standard errors of the regression coefficients. Especially, only **lmridge** can compute their covariance matrix and for the linear model only, by using the "vcov" or "vcov.lmridge" functions.

Since a usual approach for obtaining the covariance matrix is to apply a very time-consuming Bootstrap method (Efron 1992), we recommend to perform a Bayesian hierarchical GLM by using Gaussian priors to get summary statistics of each group (gene/pathway). Thus, a user could apply the **BhGLM** package (Yi et al. 2019) that provides fast and stable algorithms to estimate parameters for high-dimensional clinical and genomic data as well as highly correlated variables.

Example 3: Inputs of GCPBayes using individual level data from two case-control studies (on breast and thyroid cancers) for *PARP2* gene We propose to estimate the regression coefficients and their covariance matrices using Bayesian hierarchical logistic model which would lead to a more powerful inputs for **GCPBayes** (as we mentioned earlier) than the inputs used in Example 1:

```
library(GCPBayes)
library(BhGLM)
library(arm)

#> Loading required package: MASS

#> Loading required package: Matrix

#> Loading required package: lme4

#>
#> arm (Version 1.13-1, built: 2022-8-25)

#> Working directory is /Users/taban/Desktop/GCPBayes-new-2021/paper

#>
#> Attaching package: 'arm'

#> The following object is masked from 'package:car':
#>
#>     logit

data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
Fit1 <- bglm(y1~ ., family=binomial(link="logit"),data=Breast)
Betah1 <-  Fit1$coefficients[-1]
Sigmah1 <- cov(coef(sim(Fit1)))[-1,-1]
Fit2 <- bglm(y2~ ., family=binomial(link="logit"),data=Thyroid)
Betah2 <-  Fit2$coefficients[-1]
Sigmah2 <- cov(coef(sim(Fit2)))[-1,-1]
Betah <- list(Betah1,Betah2)
Sigmah <- list(Sigmah1,Sigmah2)
#print(Betah,digits=1)
#print(Sigmah,digits=1)
```

After running the above commands, the regression coefficients and their covariance matrices would be created based on the individual level data of two studies. The results can be checked using the last two "print" commands.

As a final remark in this section, we showed in a previous study (Baghfalaki et al. 2021) that running GCPBayes on summary statistics data could lead to a loss of power of the approaches as it leads to consider a diagonal covariance matrix of the effects (without information on off-diagonal components). So, statistically speaking, when individual level data are available, we recommend to compute the covariance matrix of the effects and to use it as inputs of GCPBayes in order to increase the power of the method.

Usage

In this section, we describe how to use the **GCPBayes** package that includes four functions : DS, CS, HS, and MCMCplot. First, Examples 4-7 shows the outputs of each of the four functions while

applying GCPBayes to the summary statistics described in Example 1 to test whether the gene *PARP2* is associated to both thyroid cancer and breast cancer risk. Then, in Example 8, we compared results obtained with DS function applied to summary statistics of *PARP2* gene versus those obtained with individual level data (as described in Examples 2-3) as input. Finally, in Example 9, we show an example of output for the gene *DNAJC1* which was significantly associated to breast and thyroid cancers using the DS function. The inputs are summary statistics from two larger studies than those used in Example 1 and will be detailed below. In addition to the real datasets described in Examples 1-3 and Example 9, three more simulated data are provided in Appendix B and also are embedded in the **GCPBayes** package including summary statistics level data for $K = 5$ studies with binary outcome, individual level data for $K = 3$ studies with continuous outcome and individual level data for $K = 2$ studies for survival outcomes and gene expression data. All commands used for the examples considered in this section are available through the R documentation of the **GCPBayes** package.

DS function

DS function runs a Gibbs sampler for a multivariate Bayesian sparse group selection model with Dirac spike prior for the detection of pleiotropic effects on the traits. As we mentioned in the previous section, the DS function is designed to use estimated regression coefficients and their estimated covariance matrices from K studies. The statistical details of the DS approach are given in Appendix A. Besides, more details can be found in (Baghfalaki et al. 2021). The DS function and its parameters has the following format:

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

where

- Betah: A list containing m-dimensional vectors of the regression coefficients for K studies.
- Sigmah: A list containing the $m \times m$-dimensional positive definite covariance matrices which is the estimated covariance matrices of K studies. If individual level data are not available, it corresponds to the diagonal matrices with estimated variance of the regression coefficients for the K studies.
- kappa0: Initial value for kappa such that its dimension is equal to nchains. Although, the best strategy for considering the initial values are the previous studies, but, the domain for choosing any value is equal to the domain of the priors which is given in the hierarchical setup A.2 in Appendix A. The initial values ranges kappa0 could be in $(0, 1)$ range.
- sigma20: Initial value for sigma2 such that its dimension is equal to nchains. The domain for initial values of sigma2 is $\Re^+$.
- 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. Default value is burnin=1000.
- nthin: The lag of the iterations used for the posterior analysis (or thinning rate). Default value is nthin=2.
- nchains: Number of Markov chains. When nchains > 1, the function calculates the Gelman-Rubin convergence statistic (Brooks and Gelman 1998; Gelman, Rubin, et al. 1992). Default value is nchains=2.
- a1, a2: Hyperparameters of kappa. Default value is a1=0.1 and a2=0.1.
- d1, d2: Hyperparameters of sigma2. Default value is d1=0.1 and d2=0.1.
- snpnames: Names of variables for the group.
- genename: Name of the group.

Example 4: Usage of the DS function to analyse pleiotropy between breast and thyroid cancers at *PARP2* gene Here we consider the individual data for *PARP2* gene from two case-control studies on breast and thyroid cancer (Examples 2-3) as input. So, as we explained earlier, first we estimated the effect size of the six SNPs of *PARP2* and the corresponding covariance matrix for both studies. Then, we applied the DS function to the results.

```
library(GCPBayes)
data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
genename <- "PARP2"
snpnames <- names(PARP2$Breast)[-1]
```

```
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)
```

To detect pleiotropic effects for this gene, 2000 iterations are considered with a burn-in period equals to 1000, and two chains are applied.

```
set.seed(123)
RES1 <- DS(Betah, Sigmah, kappa0=c(0.2,0.5), sigma20=c(1,2),
           m=6, K=2, niter=2000, burnin=1000, nthin=2, nchains=2,
           a1=0.1, a2=0.1, d1=0.1, d2=0.1, snpnames, genename)
```

The result of the DS function will be save in the "RES1" list which includes four lists (MCMCChain, Summary, Criteria, and Indicator). The "MCMCChain" list contains posterior samples for unknown parameters while the "Summary" list includes summary statistics of the posterior samples for unknown parameters (including name of SNP, posterior mean, posterior standard deviation, quantile 2.5%, median, quantile 97.5% and Gelman-Rubin convergence statistic) Gelman, Rubin, et al. (1992).

Using a "print" command for the "Summary" list would result in two lists showing Bayesian estimation for all analyzed SNPs in both studies.

```
print(RES1$Summary,digits=2)
```

```
#> $Beta
#> $Beta[[1]]
#>      Name of SNP    Mean    SD val2.5pc Median val97.5pc  BGR
#> var1   rs3093872 -0.0051 0.046   -0.145      0    0.0797 1.11
#> var2   rs3093921  0.0067 0.060   -0.097      0    0.1859 0.00
#> var3   rs1713411 -0.0259 0.084   -0.296      0    0.0000 0.99
#> var4   rs3093926 -0.0210 0.067   -0.239      0    0.0069 0.87
#> var5   rs3093930  0.0131 0.036    0.000      0    0.1368 1.02
#> var6    rs878156  0.0073 0.057   -0.100      0    0.1547 1.14
#>
#> $Beta[[2]]
#>      Name of SNP   Mean    SD val2.5pc Median val97.5pc  BGR
#> var1   rs3093872 -0.076 0.139    -0.37 -0.051      0.17 1.00
#> var2   rs3093921 -0.143 0.231    -0.68 -0.123      0.27 1.00
#> var3   rs1713411  0.154 0.192    -0.20  0.152      0.57 1.00
#> var4   rs3093926  0.139 0.152    -0.10  0.125      0.47 1.00
#> var5   rs3093930  0.126 0.083     0.00  0.133      0.27 0.99
#> var6    rs878156  0.124 0.143    -0.11  0.112      0.42 1.01
```

In order to check if the chains converged to a stationary distribution or not, a user could look at the BGR column. In our example, as the values of BGR are not close to one for some of the variables, it means that the chains do not converge to a stationary distribution. To check this criteria, we also repeat the DS function with $niter = 20000$ and $burnin = 10000$.

```
set.seed(123)
RES1 <- DS(Betah, Sigmah, kappa0=c(0.2,0.5), sigma20=c(1,1.5),
           m=6, K=2, niter=20000, burnin=10000, nthin=2, nchains=2,
           a1=0.1, a2=0.1, d1=0.1, d2=0.1, snpnames, genename)
print(RES1$Summary,digits=2)
```

As it is clear, after running DS function with new parameters, the values of the BGR column now confirm convergence to the stationary distributions. Note that in the case of $nchain = 1$, the BGR column could not be computed.

The two other output lists of the "RES1" ("Criteria" and "Indicator") contain information for checking group and variable pleiotropy, respectively.

The "Criteria" list contains the name of Gene, name of SNPs, the posterior probability of association (PPA), the logarithm (to base 10) of the Bayes factor (BF), the local false discovery rate (locFDR) and θ of equation (4) of the Guidelines section. A user could check it by the following command:

```
print(RES1$Criteria,digits=2)

#> $`Name of Gene`
#> [1] "PARP2"
#>
#> $`Name of SNPs`
#> [1] "rs3093872" "rs3093921" "rs1713411" "rs3093926" "rs3093930" "rs878156"
#>
#> $PPA
#> $PPA[[1]]
#> [1] 0.2
#>
#> $PPA[[2]]
#> [1] 0.84
#>
#>
#> $log10BF
#> [1] 0.25
#>
#> $lBFDR
#> [1] 0.16
#>
#> $theta
#> [1] 0.2
```

Based on the PPA1 and PPA2 values, the gene *PARP2* is significantly associated to the trait in the second study (thyroid cancer) but not to the other trait (breast cancer). This is also confirmed by the values of lBFDR and BF. As an interpretation of these statistics, one can conclude that the global null hypothesis of no association at the group level cannot be rejected. Now, we can check the results of the pleiotropic effect at group level by considering the value of theta which is close to 0.21. By considering the threshold 0.5, it shows that there is no group-level pleiotropic effect for textit{PARP2} gene. However, this value of theta is substantive and could be of interest.

Also, an "Indicator" list is given for checking variable pleiotropy using two different statistics (95% credible interval and median threshold). Each row of the lists includes the name of SNP, a binary indicator for significant regression coefficient of each study (0=no,1=yes) and the total number of significant regression coefficients in the studies. Finally, the last column indicates whether the test for pleiotropy at the variable level is significant or not.

```
print(RES1$Indicator,digits=2)
```

Also, *summaryDS* is a generic function used to produce result summaries of DS function:

```
summaryDS(RES1)

#> $`Name of Gene`
#> [1] "PARP2"
#>
#> $`Number of SNPs`
#> [1] 6
#>
#> $`Name of SNPs`
#> [1] "rs3093872" "rs3093921" "rs1713411" "rs3093926" "rs3093930" "rs878156"
#>
#> $log10BF
#> [1] 0.2523829
#>
#> $lBFDR
#> [1] 0.1571292
#>
#> $theta
#> [1] 0.2000668
#>
#> $`Significance based on CI`
```

```
#>           Study 1 Study 2 Total Pleiotropic effect
#> rs3093872       0       0     0                 No
#> rs3093921       0       0     0                 No
#> rs1713411       0       0     0                 No
#> rs3093926       0       0     0                 No
#> rs3093930       0       0     0                 No
#> rs878156        0       0     0                 No
#>
#> $`Significance based on median thresholding`
#>           Study 1 Study 2 Total Pleiotropic effect
#> rs3093872       0       1     1                 No
#> rs3093921       0       1     1                 No
#> rs1713411       0       1     1                 No
#> rs3093926       0       1     1                 No
#> rs3093930       0       1     1                 No
#> rs878156        0       1     1                 No
```

As it is seen, there is no variable-level pleiotropic effect for this gene which means that none of its SNPs was significant in both studies.

CS function

CS function run a Gibbs sampler for a multivariate Bayesian sparse group selection model with continuous spike prior for detection of pleiotropic effects on K traits. CS uses the same inputs than DS function. The details of the approach are given in the material and methods section (Appendix A). More details could be found in (Baghfalaki et al. 2021). The CS function has the following format in general:

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)

where Betah, Sigmah, kappa0, m, K, niter, burnin, nthin, nchains, a1, a2, snpnames and genename are the same as those introduced for the DS function. For the other arguments we have

- zeta0: Initial value for zeta. For elicitation of initial values of zeta, first the regression coefficients and their standard errors for each study is considered. After that, by adjusting their p-values, we can use them as a tool for finding initial values of zeta,~$k = 1, \cdots, K$. In this way, if one group has at least one non-null signal, the group may be a non-null and zeta0$[k] = 1$, otherwise zeta0$[k] = 0$.
- c1, c2: Hyperparameters of tau2. Default values are 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 value is 10^{-3}.

Example 5: Using the CS method to analyse pleiotropy between breast and thyroid cancers at *PARP2* gene For this example, we consider the same data used for Example 4 as input. So, first we estimated the regression coefficients for each SNP and corresponding covariance matrix for both studies (the same estimated regression coefficients and covariance matrices were computed in Example 4), and then, we performed the CS function on the results. To obtain the values for $\zeta_k^{(0)}$, a user needs to calculate P-values and initial values of ζ_k corresponding to each regression coefficients using the following code (for more information, see (Baghfalaki et al. 2021):

```
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)
}
```

The initial value is a two-dimensional vector (as K=2) of 0 or 1:

```
print(zinit)

#> [1] 0 0
```

Now, we could use the CS function to look for a pleiotropic signal:

```
set.seed(123)
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)
```

The output of CS (here RES1) is the same as DS and has similar interpretations, *summaryCS* is a generic function used to produce result summaries of CS function.

HS function

The HS function runs a Gibbs sampler for a multivariate Bayesian sparse group selection model with hierarchical spike prior for the detection of pleiotropic effects associated to the traits at group-level and variable-level. As for CS and DS, this function is designed to use summary statistics as inputs, containing estimated regression coefficients and their estimated covariance matrices. The following is a general usage for the HS function:

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)

where Betah, Sigmah, kappa0, sigma20, m, K, niter, burnin, nthin, nchains, a1, a2, d1, d2, snpnames, and genename have similar definitions as the DS function. For the other arguments we have

- kappastar0: Initial value for kappastar such that its dimension is equal to nchains. The domain for initial values of kappastar0 is $(0, 1)$.
- s20: Initial value for s2 such that its dimension is equal to nchains such that the domain for initial values of s2 is $\Re^+$.
- 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.

Example 6: Usage of the HS method to analyse pleiotropy between breast and thyroid cancers at *PARP2* gene The same data as Example 4 is used as input. We used the same estimated regression coefficients and their covariance matrices.

```
library(GCPBayes)
data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
genename <- "PARP2"
snpnames <- names(PARP2$Breast)[-1]
Fit1 <- BhGLM::bglm(y1~ ., family=binomial(link="logit"),data=Breast)
Betah1 <-  Fit1$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)

set.seed(123)
RES <- HS(Betah, Sigmah, kappa0=c(0.5,0.3), kappastar0=c(0.5,0.3), sigma20=c(2,1), s20=c(1,2),
           m=6, K=2, 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)
```

In this example, the HS function is applied to detect group and variable pleiotropic signals. The outputs of the HS function based on median threshold are as follows:

```
summaryHS(RES1)
```

The overall output of HS (RES1) is the same as DS and has similar interpretations.

MCMCplot function

In addition to the Gelman–Rubin convergence diagnostic, monitoring the convergence of the MCMC algorithm is essential for producing results from the posterior distribution of interest. MCMCplot function presents some visual plots such as trace, auto-correlation function (ACF) and posterior density plots to check the convergence of the MCMC chains. The trace plot is the plot of the iterations versus the generated values. If all values are within a zone without strong periodicities and tendencies, then we can assume convergence. The density plot is the estimated posterior distribution of signal. Also, monitoring ACF plots is very useful since low or high values indicate fast or slow convergence, respectively. In fact, we need to monitor the ACF of the MCMC generated sample and select a sampling lag larger than 1 [=L]. Then, we can produce an independent sample by keeping the first generated values in every batch of L iterations which is called thinning rate. In practice, the ACF should be close to zero for a sufficiently large number of lags.

The inputs of the MCMCplot function is the generated results by DS, CS, or HS function, the number of study for drawing plots, the number of MCMC chains in the result, and the names of favorable SNPs for drawing the plots. The function is able to plot Maximum of 10 SNPs separately. In general, a usage for the MCMCplot function is as follow:

MCMCplot(Result, k, nchains, whichsnps, betatype, acftype, dencol, denlty, denbg)

where

- Result: All the generated results by DS/CS/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.

Example 7: Using of the MCMCplot function to check convergence of the DS method For this example, the output of DS function for *PARP2* gene is consider to draw the MCMCplot. We select this gene because it is a special case as it is required more MCMC iterations to converge. To run this example, a user needs to run the first fourteen commands in the Example 4 (briefly, these commands computes regression coefficients and covariance matrices based on the individual data). Then, it is needed to run the DS function as follow:

```
library(GCPBayes)
data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
genename <- "PARP2"
snpnames <- names(PARP2$Breast)[-1]
Fit1 <- BhGLM::bglm(y1~ ., family=binomial(link="logit"),data=Breast)
Betah1 <-  Fit1$coefficients[-1]
Sigmah1 <- cov(coef(arm::sim(Fit1)))[-1,-1]
Fit2 <- BhGLM::bglm(y2~ ., family=binomial(link="logit"),data=Thyroid)
Betah2 <-  Fit2$coefficients[-1]
```

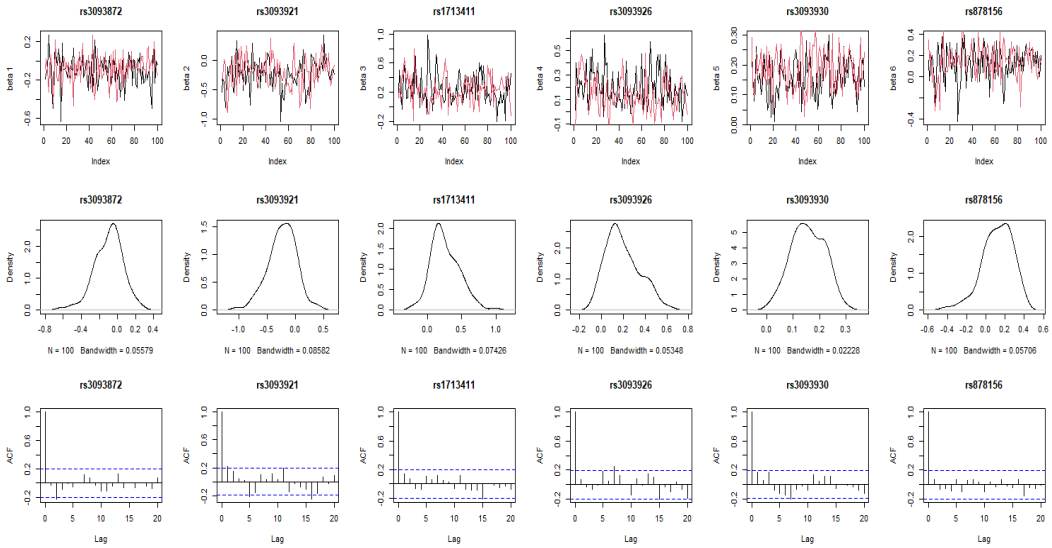

Figure 1: Trace, density and ACF plot for the posterior samples of the regression coefficients (niter=200, burnin=100, nthin=2) for thyroid study. Based on the top and last rows, the results do not show a convergence.

```
Sigmah2 <- cov(coef(arm::sim(Fit2)))[-1,-1]
Betah <- list(Betah1,Betah2)
Sigmah <- list(Sigmah1,Sigmah2)

set.seed(123)
RES1 <- DS(Betah, Sigmah, kappa0=c(0.2,0.5), sigma20=c(1,2),
           m=6, K=2, niter=200, burnin=100, nthin=1, nchains=2,
           a1=0.1, a2=0.1, d1=0.1, d2=0.1, snpnames, genename)
MCMCplot(Result = RES1, k = 2, nchains = 2, whichsnps = snpnames,
                    betatype = "l",
                    acftype = "correlation",
                    dencol = "white", denlty = 1, denbg = "white")

set.seed(123)
RES1 <- DS(Betah, Sigmah, kappa0=c(0.2,0.5), sigma20=c(1,2),
           m=6, K=2, 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 = 2, nchains = 2, whichsnps = snpnames,
                    betatype = "l",
                    acftype = "correlation",
                    dencol = "white", denlty = 1, denbg = "white")
```

The plots for thyroid cancer study with 200 MCMC iterations and 100 burn-in is given in Figure 2. The top row of the figure shows strong periodicities of the posterior samples of some SNPs which is confirmed by their corresponding ACF plots (last row). Thus, more samples are required to get into a convergence for the *PARP2* gene. In the second DS command, the number of iterations was increased to 2000, half of this number was considered as burn-in and thinning rate was equal to 2. The results are shown in Figure 3 which imply the convergence with increasing the number of iterations.

```
knitr::include_graphics("p1.png")

knitr::include_graphics("p2.png")
```

Comparing GCPBayes results using summary statistics level versus individual level data as input

As we mentioned earlier, we showed that the performance of the **GCPBayes** package is better in the case of considering non-diagonal covariance matrices (individual level data) compared to diagonal

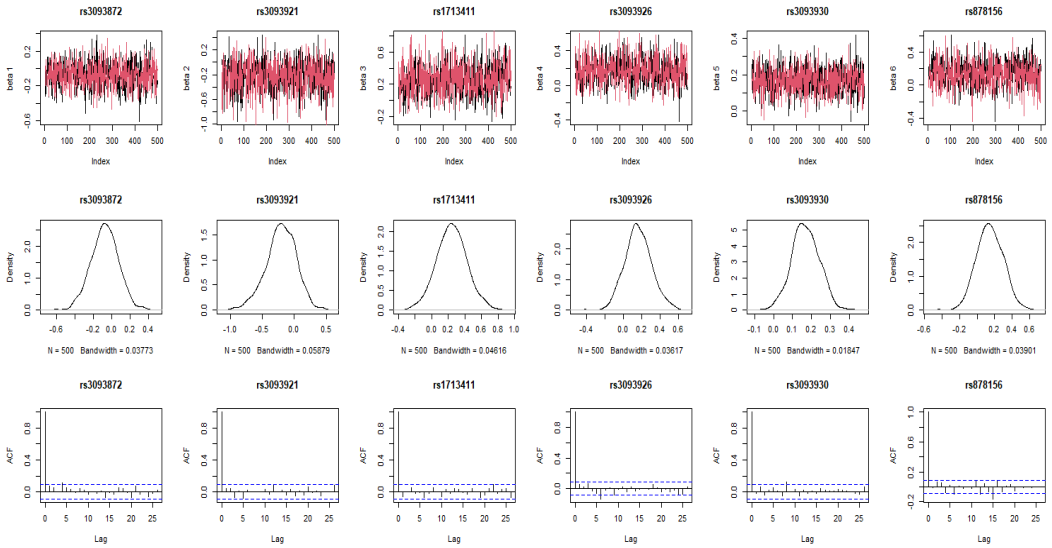

Figure 2: Trace, density and ACF plot for the posterior samples of the regression coefficients (niter=2000, burnin=1000, nthin=5) for thyroid study. Based on the top and last rows, the results imply a convergence.

covariance matrices (summary level data) (Baghfalaki et al. 2021). In this section, we show two examples to deal with such difference.

Example 8: Comparing DS function outputs using summary statistics level versus individual level data of *PARP2* gene as inputs First, we consider summary statistics level data for *PARP2* gene as an input of the **GCPBayes** package and run the DS function for this data as follow:

```
library(GCPBayes)
data(PARP2_summary)
Breast <- PARP2_summary$Breast
Thyroid <- PARP2_summary$Thyroid
genename <- "PARP2"
snpnames <- rownames(Breast)
Betah <- list(Breast$beta,Thyroid$beta)
Sigmah <- list(diag(Breast$se^2),diag(Thyroid$se^2))
print(Betah,digits=2)
print(Sigmah,digits=2)
set.seed(123)
RES_sum <- DS(Betah, Sigmah, kappa0=c(0.2,0.5), sigma20=c(1,2),
                          m=6, K=2, niter=2000, burnin=1000, nthin=2, nchains=2,
                          a1=0.1, a2=0.1, d1=0.1, d2=0.1, snpnames, genename)
print(RES_sum$Criteria,digits=2)
```

Now, we use individual data for *PARP2* gene, then computed regression coefficients and covariance matrices based on the individual data, and use it as an input of the **GCPBayes** package. Finally, we run the DS function for this data as follow:

```
library(GCPBayes)
data(PARP2)
Breast <- PARP2$Breast
Thyroid <- PARP2$Thyroid
genename <- "PARP2"
snpnames <- names(PARP2$Breast)[-1]
Fit1 <- BhGLM::bglm(y1~ ., family=binomial(link="logit"),data=Breast)
Betah1 <-  Fit1$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)
set.seed(123)
RES_ind <- DS(Betah, Sigmah, kappa0=c(0.2,0.5), sigma20=c(1,2),
                          m=6, K=2, niter=2000, burnin=1000, nthin=2, nchains=2,
                          a1=0.1, a2=0.1, d1=0.1, d2=0.1, snpnames, genename)
#print(RES_ind$Criteria,digits=2)
```

It is clear by comparing the results for these two scenarios that using the individual level data which includes more information about the SNPs correlations would lead to a less bias results and hence to get a better signals at the end.

Example 9: Investigation of pleiotropic effect between breast cancer and thyroid cancer at *DNAJC1* gene in large datasets For the final example, we consider the summary statistics for *DNAJC1* protein coding gene including beta and standard error for fourteen SNPs from two studies (on breast and thyroid cancers) and try to investigate any potential pleiotropic signal at group and individual levels. The summary statistics of the breast and thyroid cancer studies were extracted from Breast Cancer Association Consortium (BCAC) (Zhang et al. 2020) and EPITHYR consortium (Truong et al. 2021), respectively. The data is embeded in the **GCPBayes** package and a user could run the DS function as follow:

```
library(GCPBayes)
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)
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)
```

Now, we run the following command for testing the global null hypothesis:

```
print(RES1$Criteria)

#> $`Name of Gene`
#> [1] "DNAJC1"
#>
#> $`Name of SNPs`
#>  [1] "rs10740997" "rs10764330" "rs10828266" "rs12570400" "rs1970467"
#>  [6] "rs2666762"  "rs2807967"  "rs35759613" "rs3951780"  "rs4747438"
#> [11] "rs60773921" "rs6650129"  "rs7075508"  "rs77353976"
#>
#> $PPA
#> $PPA[[1]]
#> [1] 1
#>
#> $PPA[[2]]
#> [1] 1
#>
#>
#> $log10BF
#> [1] 47.29012
#>
#> $lBFDR
#> [1] 1.709075e-48
#>
```

```
#> $theta
#> [1] 0.9998224
```

Based on the values of log10BF and lBFDR, the null hypothesis is rejected. Thus, we check the value of theta which is equal to 0.999, and conclude that the gene has a group pleiotropic effect.

For checking variable pleiotropic effect, a user should run the following command:

```
print(RES1$Indicator$`Significant studies and Pleiotropic effect based on CI`)
```

```
#>            Study 1 Study 2 Total Pleiotropic effect
#> rs10740997       1       0     1                 No
#> rs10764330       1       0     1                 No
#> rs10828266       1       0     1                 No
#> rs12570400       1       0     1                 No
#> rs1970467        1       0     1                 No
#> rs2666762        1       1     2                Yes
#> rs2807967        1       0     1                 No
#> rs35759613       1       0     1                 No
#> rs3951780        1       0     1                 No
#> rs4747438        1       1     2                Yes
#> rs60773921       1       0     1                 No
#> rs6650129        1       0     1                 No
#> rs7075508        1       0     1                 No
#> rs77353976       0       0     0                 No
```

Based on the result, four SNPs are significantly associated to both traits (rs10740997, rs10828266, rs2666762, and rs6650129).

Guidelines

In this section, we provide some remarks for a user in order to work with the **GCPBayes** package and extract potential pleiotropic effects at group and variable-levels. We first give some practical advice on which methods to consider and in what order, and then we provide a decision pipeline based on statistical inferences, firstly for a primary analysis by using DS (or CS), and secondly for a precision analysis by using HS.

Practical hints

We do not recommend to a user to perform the initial running by directly using the HS function because it might have a long computational time when using a large datasets. Besides, the HS function does not show a better power in detection of a group pleiotropy compared to the other methods (DS and CS). In addition, DS and CS methods have similar strategies but we showed that the DS performs better than CS during the simulations, for all tested scenarios (Baghfalaki et al. 2021). That is why we suggest to a user to first perform the pleiotropic analysis on all groups of the data by using the DS function (Figure 3). Thus, in order to select groups with probable pleiotropic effects, we recommend to perform a first batch of analyses by using the DS method, for a reasonable amount of iterations for which most of the groups should have converged, using 2 chains to compute the BGR and test the convergence of the method. We suggest to perform the method for 2,000 iterations and 1,000 of burn-in. Then, if the BGR values for one SNP of the group is significantly different from 1 (a BGR between 0.9 and 1.1 as a convergence threshold), the method should be run again for the group by using higher values of iterations and burn-in. We recommend to double these values until the convergence threshold are verified.

Statistical inferences pipeline for primary analysis

We provide a diagram in Figure 3 which help a user to detect group and variable pleiotropic effects using the **GCPBayes** package. The first step for detection of a pleiotropic signal is to test the global null hypothesis of no association against the global alternative hypothesis of association with at least one trait by considering DS (or CS).

Considering the CS method of the package (please refer to the corresponding equations in Appendix A), we define the following global null hypothesis of no association as follow:

$$\mathrm{H}_0 : \xi_1 = \xi_2 = \cdots = \xi_K = 0 \quad \text{versus } \mathrm{H}_1 : \text{at least one } \xi_k \neq 0, k = 1, \cdots, K. \tag{2}$$

Two quantities are computed by the package to perform this test. First, let $P(\mathrm{H}_0|\mathcal{D})$ be the posterior probability of the null hypothesis, i.e., the probability of making a false discovery for a non-null effect which is called the local Bayesian false discovery rate for $\mathcal{D}$ (denoted by $lBFDR$) by Efron (Efron 2012). The phrase "local" comes from a single point $\{0\}$ as the domain of the null hypothesis tested (Efron 2012). Note that small values of $lBFDR$ show strong evidence for the existence of a substantive effect. We recommand to use the usual value of 0.05 as a threshold to consider a non-null effect or not. A user could also consider the Bayes factor (BF) which describes the evidence of H_1 against H_0. Unlike the $lBFDR$, a large value of a Bayes factor (BF) indicates strong evidence in favour of H_1. We recommand to consider a value of 1 as a threshold (or log10BF > 0).

The marginal study-specific posterior probability of association (PPA) proposed by Majumdar et al. (Majumdar et al. 2018) is also computed to quantify the relative contribution of a study underlying a signal. This is defined for each study k by $PPA_k = \frac{1}{M}\sum_{r=1}^{M} \xi_k^{(r)}$. This correspond for the study k to the draw proportion from the slab distribution, i.e. the proportion of time the group was considered to have a non-zero effect based on posterior results.

For checking any association by using DS, the global test is defined by:

$$\mathrm{H}_0 : \beta_1 = \cdots = \beta_K = 0, \text{versus } \mathrm{H}_1 : \text{at least one fi}_\mathrm{k} \neq 0, \ \mathrm{k} = 1, \cdots, \mathrm{K}. \tag{3}$$

The existence of an association by using DS can be defined in the same way as for CS.

If there is no sufficient evidence for H_1 (i.e. a non-null effect in at least one study), then there is no need to further consider group pleiotropy neither variable pleiotropy.

If there is evidence for H_1, then a user can evaluate the value of θ of the DS function. The strategy to detect group pleiotropic effects based on CS is to compute θ as follow:

$$\theta = P\left(\sum_{k=1}^{K} \xi_k \geq 2 | \hat{\beta}_k, \hat{\Sigma}_k, \ k = 1, \cdots, K\right). \tag{4}$$

Note that for computing θ for DS, we can follow the same strategy as for CS. Hence, θ represente the probability to have a non-zero effect in at least two studies. Thus, it is reasonable to consider a threshold t of 0.5 to statistically determine whether there is pleiotropy or not. However, a user wishing to generate hypotheses could choose a less stringent threshold (i.e. 0.1). Or conversely, a user could choose a very strict threshold (i.e. 0.9) to select only the most certain groups. If $\theta \leq t$, then no pleiotropic effect is detected at the group-level and so there is no pleiotropy signal at variable level. On the other hand, if $\theta > t$, we can consider the group has a pleiotropic effect. It should be mentioned that when a pleiotropic effect is detected for a group, we can investigate the pleiotropy for each variable in the group. In this case, as CS and DS are defined group by group, the use of Bayesian point estimations (the means of the posterior distributions) and 95% credible intervals from DS function can be used as criteria for variable selection. If at least two β_{kj}s are selected, the j^{th} variable has a pleiotropic effect. For example, assume that a pleiotropic effect is detected for an analysis on two studies. Then, the posterior estimates $\hat{\beta}_{1j}$ and $\hat{\beta}_{2j}$ can be used as a decision tool for pleiotropy for the j^{th} variable $j = 1, 2, \cdots, m$. Hence, if β_{1j} and β_{2j} are two non-zero signals, then the j^{th} variable is declared to have a pleiotropic effect, otherwise there is no pleiotropic effect for the j^{th} variable.

Statistical inferences for analysis by using HS

As HS have shown a better power to detect pleiotropy at variable-level in simulations, we then recommend to run the HS approach for the groups detected as pleiotropic at first by using the DS function. A user can move down the θ threshold in order to explore potential pleiotropic effects which could have been detected with a larger sample size (i.e. to consider $\theta \geq 0.1$ from DS). Then, the user would be able to compare the results at variable-level for DS and HS (Figure 3). Using this overall strategy, a user would have to run the HS method on a smaller number of groups and so avoid dealing with long computational time of the HS function for all groups at first place.

The posterior estimates $\hat{\beta}_{kj}$, $j = 1, 2, \cdots, m$, $k = 1, 2, \cdots, K$ can be used to detect a pleiotropic signal for the j^{th} variable. We consider that the j^{th} variable has a pleiotropic effect if and only if there are at least two non-zero signals among the K studies. Note that for this purpose, in addition to a 95% CI, the sparseness of the posterior median of the regression coefficient β_{kj}, $k = 1, \cdots, K$, $j = 1, 2, \cdots, m_g$

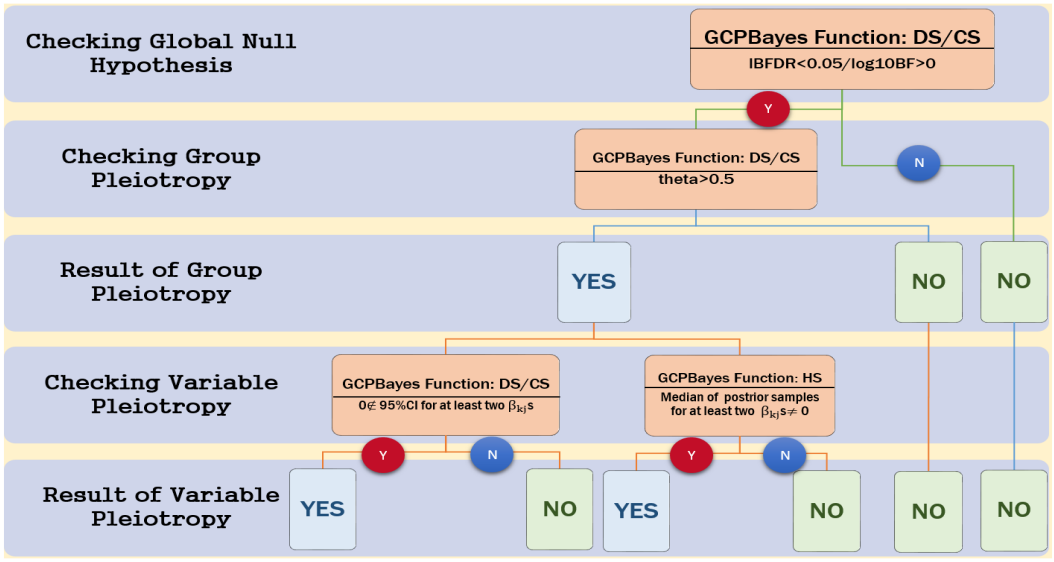

Figure 3: A diagram for detecting group and variable pleiotropic effects using GCPBayes package. For more information see the text.

can be used. We recommand to consider pleiotropy at variable-level based on the posterior median: e.g. the j^{th} variable can be considered as pleiotropic if the posterior median of a β_{kj} is different of 0 for at least two studies k. Finally, HS consider pleiotropy at group-level if at least one pleiotropic variable is detected within the group. Moreover, HS allows to detect pleiotropy at group-level in other situations. Note that the pleiotropy effect can wear different shapes. As presented in (Solovieff et al. 2013), biological pleiotropy at gene level can occur when two different signals within the same gene are associated with two different phenotypes. This can affect both phenotypes in a similar way. Therefore, we also consider pleiotropy at group-level when an association is detected (not necessarily detected as pleiotropic) for different variables within the group in at least two different studies.

Computational time for GCPBayes

In this section, the computational time of GCPBayes package is explored using reanalysis of the experimental data for nine groups (genes) described in (Baghfalaki et al. 2021). The analysis has been conducted using a PC with a Core(TM) i5-4690K CPU @ 3.50 GHz processor and 8 GB of memory. For this purpose, we consider nine genes with different numbers of variables (SNPs). The number of iterations of MCMC is considered to be 2000 and half of it is discarded as burn-in. Table 1 represents the results of computational times using the GCPBayes package for analysing the data. As expected, the computational time increases when the number of variables within a group increases. The computational time for the DS method is the shortest, and the more complex model HS has the longest computational time, as expected. For another exploring about computational time for the GCPBayes package based on a simulation studies could be found in (Baghfalaki et al. 2021).

We have also applied GCPBayes on whole genome data of breast and ovarian cancers (summary statistics data). The computational time for running GCPBayes over all coding genes (number of genes: 18,244 and number of SNPs: 1 to 9,595) was about 17 days (on a PC with Intel Core(TM) i7-1165G7 @ 2.80 GHz, 32 GB RAM). However, if a user applies parallelized loops with R for running GCPBayes on more than one gene at the same time (using "foreach" command), the computational time would be reduced. For instance, the computational time was about 40 hours using the same PC but running 6 genes in each loop (i.e. using 6 CPUs instead of 1).

```
knitr::include_graphics("tree.png")
```

Concluding remarks

We have developed an R package GCPBayes that implements several Bayesian meta-analysis methods for studying cross-phenotype genetic associations (pleiotropy), taking into account group structure

Table 1: Computational time (in mins) of GCPBayes.

Name of gene	Number of SNPs	CS	DS	HS
ADH1A	2	0.070	0.030	0.139
ACE	5	0.069	0.036	0.172
ADH1B	10	0.084	0.049	0.243
RFC3	14	0.105	0.066	0.347
AOX1	24	0.158	0.127	0.629
ABCC8	30	0.220	0.190	0.966
BCAT1	50	0.555	0.539	3.123
EBF2	60	0.779	0.775	5.156
EGFR	103	2.697	2.882	30.794

from prior biological knowledge. GCPBayes offers multivariate spike and slab priors for group-level pleiotropy using either DS or CS formulations, and for pleiotropy at both group and variable-level using the hierarchical spike prior (HS) formulation. These approaches take into account heterogeneity in the size and direction of the genetic effects across traits. Although DS and CS are not designed for variable selection, these methods allow to investigate the pleiotropy for each variable in the group by using 95% credible intervals and median threshold as criteria for variable selection in groups for which a pleiotropic effect has been detected. On the other hand, the posterior estimates can be used to detect a pleiotropic signal for each variable using HS. Thus, a variable is considered to have a pleiotropic effect if and only if there are at least two non-zero signals among the studies. For this purpose, in addition to a 95% credible intervals, the sparseness of the posterior median of the regression coefficients is reported. Also, it should be mentioned that GCPBayes package provides marginal trait-specific posterior probability of association of each study (PPA), direction of associations for each variable, Bayes factor and local Bayesian false discovery rate of the global hypothesis, a statistical criterion for detection of group pleiotropic signal (θ), credible interval and median of the variables, summary information of the variables, and the generated MCMC samples. Besides, as we mentioned, HS allows to consider the pleiotropy at group-level if at least one pleiotropic variable is detected within the group. But it also allows detection of pleiotropy at group-level without pleiotropy at variable-level if two or more different variables are significant in different studies (Baghfalaki et al. 2021). This kind of situation might be an interesting biological case while, for example, the same gene (group) may have a similar effect on several phenotypes through different genetic markers (variables) (Solovieff et al. 2013).

We provided 9 examples to demonstrate different scenarios of working with the package using both simulated and real data. In addition, three more examples are provided in the Appendix B to show the ability of the GCPBayes package to deal with various types of input data. As we mentioned earlier, while the input of the package is summary statistics data, it is possible to use the package for investigating pleiotropy in various kind of phenotypes. So, a user could explore for pleiotropic signals in different types of phenotypes including categorical, continuous, mixed of categorical and continuous, survival data, etc. without any limitation.

However, there are still some limitations which a user could consider while working with the package. For example, for more accurate detection of pleiotropic signals, large enough sample size for an input data should be provided. This means having larger sample size in the individual level data (which summary statistics data is calculated from) would lead to a better accuracy of the method for detecting pleiotropic signals. Besides, the GCPBayes package is designed for uncorrelated studies (no overlapping samples between studies). So, an improvement of the package for correlated studies could be considered for the future. Also, GCPBayes can deal with correlated samples in uncorrelated studies as it uses summary statistics. Though, a user should be careful that summary statistics has been calculated using methods taking into account for such correlations between samples. More user-friendly functions to prepare the data, pre-selection and assigning key variables into groups before applying the GCPBayes methods are currently in development. Furthermore, since the main focus of the package is on detection of pleiotropic effect using GWAS data, we are currently developing a guideline to optimize the package for usage in a real large scale data by considering for instance optimal iterations of the MCMC step for convergence and development of a general strategy for dealing with genes with large number of SNPs or large size.

Acknowledgements

The authors acknowledge the calculus center MCIA (Mésocentre de Calcul Intensif Aquitain) for providing its facilities. The 'Ligue contre le Cancer' is acknowledged as well for its support for "Cross Cancer Genomic Investigation of Pleiotropy project". The INSERM and Aviesan ITMO cancer are acknowledged as well for their support for "Advanced Machine Learning Algorithms for leveraging

Pleiotropy effect project". This project is also part of the INSERM cross-cutting project GOLD.

Agresti, Alan. 2018. *An Introduction to Categorical Data Analysis*. John Wiley & Sons.

Baghfalaki, Taban, Pierre-Emmanuel Sugier, Therese Truong, Anthony N Pettitt, Kerrie Mengersen, and Benoit Liquet. 2021. "Bayesian Meta-Analysis Models for Cross Cancer Genomic Investigation of Pleiotropic Effects Using Group Structure." *Statistics in Medicine* 40 (6): 1498–518.

Bhattacharjee, Samsiddhi, Preetha Rajaraman, Kevin B Jacobs, William A Wheeler, Beatrice S Melin, Patricia Hartge, Meredith Yeager, Charles C Chung, Stephen J Chanock, and Nilanjan Chatterjee. 2012. "A Subset-Based Approach Improves Power and Interpretation for the Combined Analysis of Genetic Association Studies of Heterogeneous Traits." *The American Journal of Human Genetics* 90 (5): 821–35.

Broc, Camilo, Therese Truong, and Benoit Liquet. 2021. "Penalized Partial Least Squares for Pleiotropy." *BMC Bioinformatics* 22 (1): 1–31.

Brooks, Stephen P, and Andrew Gelman. 1998. "General Methods for Monitoring Convergence of Iterative Simulations." *Journal of Computational and Graphical Statistics* 7 (4): 434–55.

Chung, Dongjun, Can Yang, Cong Li, Joel Gelernter, and Hongyu Zhao. 2014. "GPA: A Statistical Approach to Prioritizing GWAS Results by Integrating Pleiotropy and Annotation." *PLoS Genet* 10 (11): e1004787.

Dunn, Peter K, and Gordon K Smyth. 2018. *Generalized Linear Models with Examples in r*. Springer.

Efron, Bradley. 1992. "Bootstrap Methods: Another Look at the Jackknife." In *Breakthroughs in Statistics*, 569–93. Springer.

———. 2012. *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Vol. 1. Cambridge University Press.

Fox, John, and Sanford Weisberg. 2018. *An r Companion to Applied Regression*. Sage publications.

Gareth, James, Witten Daniela, Hastie Trevor, and Tibshirani Robert. 2013. *An Introduction to Statistical Learning: With Applications in r*. Spinger.

Gelman, Andrew, Donald B Rubin, et al. 1992. "Inference from Iterative Simulation Using Multiple Sequences." *Statistical Science* 7 (4): 457–72.

George, Edward I, and Robert E McCulloch. 1993. "Variable Selection via Gibbs Sampling." *Journal of the American Statistical Association* 88 (423): 881–89.

Hackinger, Sophie, and Eleftheria Zeggini. 2017. "Statistical Methods to Detect Pleiotropy in Human Complex Traits." *Open Biology* 7 (11): 170125.

Hilt, Donald E, and Donald W Seegrist. 1977. *Ridge, a Computer Program for Calculating Ridge Regression Estimates*. Vol. 236. Department of Agriculture, Forest Service, Northeastern Forest Experiment ....

Johnston, Ron, Kelvyn Jones, and David Manley. 2018. "Confounding and Collinearity in Regression Analysis: A Cautionary Tale and an Alternative Procedure, Illustrated by Studies of British Voting Behaviour." *Quality & Quantity* 52 (4): 1957–76.

Krapohl, Eva, Hamel Patel, Stephen Newhouse, Charles J Curtis, Sophie von Stumm, Philip S Dale, Delilah Zabaneh, Gerome Breen, Paul F OReilly, and Robert Plomin. 2018. "Multi-Polygenic Score Approach to Trait Prediction." *Molecular Psychiatry* 23 (5): 1368–74.

Li, Xihao, Zilin Li, Hufeng Zhou, Sheila M Gaynor, Yaowu Liu, Han Chen, Ryan Sun, et al. 2020. "Dynamic Incorporation of Multiple in Silico Functional Annotations Empowers Rare Variant Association Analysis of Large Whole-Genome Sequencing Studies at Scale." *Nature Genetics* 52 (9): 969–83.

Liu, Zhonghua, and Xihong Lin. 2018. "Multiple Phenotype Association Tests Using Summary Statistics in Genome-Wide Association Studies." *Biometrics* 74 (1): 165–75.

Lu, Qiongshi, Boyang Li, Derek Ou, Margret Erlendsdottir, Ryan L Powles, Tony Jiang, Yiming Hu, et al. 2017. "A Powerful Approach to Estimating Annotation-Stratified Genetic Covariance via GWAS Summary Statistics." *The American Journal of Human Genetics* 101 (6): 939–64.

Majumdar, Arunabha, Tanushree Haldar, Sourabh Bhattacharya, and John S Witte. 2018. "An Efficient Bayesian Meta-Analysis Approach for Studying Cross-Phenotype Genetic Associations." *PLoS Genetics* 14 (2): e1007139.

Malo, Nathalie, Ondrej Libiger, and Nicholas J Schork. 2008. "Accommodating Linkage Disequilibrium in Genetic-Association Analyses via Ridge Regression." *The American Journal of Human Genetics* 82 (2): 375–85.

McCullagh, Peter. 2019. *Generalized Linear Models*. Routledge.

Menard, Scott. 2002. *Applied Logistic Regression Analysis*. Vol. 106. Sage.

Mitchell, Toby J, and John J Beauchamp. 1988. "Bayesian Variable Selection in Linear Regression." *Journal of the American Statistical Association* 83 (404): 1023–32.

Ray, Debashree, and Nilanjan Chatterjee. 2020. "A Powerful Method for Pleiotropic Analysis Under Composite Null Hypothesis Identifies Novel Shared Loci Between Type 2 Diabetes and Prostate Cancer." *PLoS Genetics* 16 (12): e1009218.

Saleh, AK Md Ehsanes, Mohammad Arashi, and BM Golam Kibria. 2019. *Theory of Ridge Regression Estimation with Applications*. Vol. 285. John Wiley & Sons.

Solovieff, Nadia, Chris Cotsapas, Phil H Lee, Shaun M Purcell, and Jordan W Smoller. 2013. "Pleiotropy in Complex Traits: Challenges and Strategies." *Nature Reviews Genetics* 14 (7): 483–95.

Trochet, Holly, Matti Pirinen, Gavin Band, Luke Jostins, Gilean McVean, and Chris CA Spencer. 2019. "Bayesian Meta-Analysis Across Genome-Wide Association Studies of Diverse Phenotypes." *Genetic Epidemiology* 43 (5): 532–47.

Truong, Therese, Fabienne Lesueur, Pierre-Emmanuel Sugier, Julie Guibon, Constance Xhaard, Mojgan Karimi, Om Kulkarni, et al. 2021. "Multiethnic Genome-Wide Association Study of Differentiated Thyroid Cancer in the EPITHYR Consortium." *International Journal of Cancer* 148 (12): 2935–46.

Verma, Anurag, Shefali S Verma, Sarah A Pendergrass, Dana C Crawford, David R Crosslin, Helena Kuivaniemi, William S Bush, et al. 2016. "eMERGE Phenome-Wide Association Study (PheWAS) Identifies Clinical Associations and Pleiotropy for Stop-Gain Variants." *BMC Medical Genomics* 9 (1): 19–25.

Vittinghoff, Eric, David V Glidden, Stephen C Shiboski, and Charles E McCulloch. 2011. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. Springer Science & Business Media.

Watanabe, Kyoko, Sven Stringer, Oleksandr Frei, et al. 2019. "A Global Overview of Pleiotropy and Genetic Architecture in Complex Traits." *Nature Genetics* 51 (9): 1339–48.

Xu, Xiaofan, Malay Ghosh, et al. 2015. "Bayesian Variable Selection and Estimation for Group Lasso." *Bayesian Analysis* 10 (4): 909–36.

Yi, Nengjun, Zaixiang Tang, Xinyan Zhang, and Boyi Guo. 2019. "BhGLM: Bayesian Hierarchical GLMs and Survival Models, with Applications to Genomics and Epidemiology." *Bioinformatics* 35 (8): 1419–21.

Zhang, Haoyu, Thomas U Ahearn, Julie Lecarpentier, Daniel Barnes, Jonathan Beesley, Guanghao Qi, Xia Jiang, et al. 2020. "Genome-Wide Association Study Identifies 32 Novel Breast Cancer Susceptibility Loci from Overall and Subtype-Specific Analyses." *Nature Genetics* 52 (6): 572–81.

Bibliography

Taban Baghfalaki
Tarbiat Modares University
Faculty of Mathematical scinces,
Department of Statitics,
Tehran, Iran.
Université Paris-Saclay; UVSQ; INSERM, U1018; Gustave Roussy; CESP, Team Exposome and Heredity
Villejuif, France
ORCiD: 0000-0002-2100-4532
t.baghfalaki@modares.ac.ir

Pierre-Emmanuel Sugier
Laboratoire de Mathématiques et de leurs Applications de Pau
Université de Pau et des Pays de l'Adour,
UMR CNRS 5142, E2S-UPPA, France.
Université Paris-Saclay; UVSQ; INSERM, U1018; Gustave Roussy; CESP, Team Exposome and Heredity
Villejuif, France
ORCiD: 0000-0002-5846-1104
pe.sugier@univ-pau.fr

Yazdan Asgari
Université Paris-Saclay; UVSQ; INSERM, U1018; Gustave Roussy; CESP, Team Exposome and Heredity
Villejuif, France
ORCiD: 0000-0001-6993-6956
yazdan.asgari@inserm.fr

Thérèse Truong
Université Paris-Saclay; UVSQ; INSERM, U1018; Gustave Roussy; CESP, Team Exposome and Heredity
Villejuif, France
ORCiD: 0000-0002-2943-6786
therese.truong@inserm.fr

Benoit Liquet
Macquarie University
School of Mathematics and Physical Sciences
NSW, Australia
Universite de Pau et des Pays de l'Adour
Laboratoire de Mathématiques et de leurs Applications de Pau
UMR CNRS 5142, E2S-UPPA, France
ORCiD: 0000-0002-8136-2294
benoit.liquet-weiland@mq.edu.au, benoit.liquet@univ-pau.fr