Journal of Statistical Software

September 2025, Volume 114, Issue 6. *doi: 10.18637/jss.v114.i06*

hibayes: An R Package to Fit Individual-Level, Summary-Level and Single-Step Bayesian Regression Models for Genomic Prediction and Genome-Wide Association Studies

Lilin Yin
Huazhong Agricultural University

Haohao Zhang
Wuhan University of Technology

Xinyun Li
Huazhong Agricultural University

Shuhong Zhao
Huazhong Agricultural University

Xiaolei Liu
Huazhong Agricultural University

Abstract

With the rapid development of sequencing technology, the costs of individual genotyping have been reduced dramatically, leading to genomic prediction and genome-wide association studies being widely promoted and used to predict the unknown phenotypes and to locate candidate genes for animal and plant economic traits and, increasingly, for human diseases. Developing new advanced statistical models to improve prediction accuracy and location precision for the traits with various genetic architectures has always been a hot topic in those two research domains. The Bayesian regression model (BRM) has played a crucial role in the past decade, and it has been used widely in relevant genetic analyses owing to its flexible model assumptions on the unknown genetic architecture of complex traits. To fully utilize the available data from either a self-designed experimental population or a public database, statistical geneticists have constantly extended the fitting capacity of BRM, and a series of new methodologies have been proposed for different application scenarios. Here we introduce the R package **hibayes**, a software tool that can be used to fit individual-level, summary-level, and single-step Bayesian regression models. Including also the richest methods achieved thus far, it covers most of the functionalities involved in the field of genomic prediction and genome-wide association studies, potentially helping to address a wide range of research problems, while retaining an easy-to-learn and flexible-to-use experience. We believe that package **hibayes** will facilitate the academic research and practical application of statistical genetics for humans, plants, and animals.

Keywords: genomic prediction, genome-wide association studies, Bayesian regression, single-step model, summary statistics, **hibayes**, R.

1. Introduction

The phenotypes of agricultural traits and human diseases are the results of the combined influence of genetic and environmental factors. Since the genetic effects are invisible and cannot be measured directly, accurately estimating the genetic components from the recorded observations has always been a prominent and critical topic in statistical genetics. The linear mixed-effects model which fits the environmental factors as fixed or random effects has made great achievements in quantifying the environmental contributions to phenotypic observations, and the R (R Core Team 2025) packages **lme4** (Bates, Mächler, Bolker, and Walker 2015) and **brms** (Bürkner 2017) have became the most efficient and convenient tools to fit this kind of model. In the mid-20th century, the BLUP (best linear unbiased prediction) model (Henderson 1975), which can directly estimate the genetic value of each individual using the phenotypic observations, environmental records, and a relationship matrix derived from a historical pedigree, was proposed and adopted in livestock breeding data analysis. Since the BLUP used a pedigree-based relationship matrix, the model included genetic effects. However, the elements in the pedigree-based relationship matrix are values in theoretical expectations and the matrix could not capture the Mendelian sampling error fully, resulting in the same predictive performance for all sibling progeny. With the development of sequencing technology, high density genomic markers across the entire genome could be obtained and genomic prediction which essentially models the markers as independent variables and the phenotypic records as the dependent variables was proposed subsequently by Meuwissen, Hayes, and Goddard (2001). Compared with the traditional BLUP approach, using the genome-wide markers could capture Mendelian sampling error and genetic links through unknown common ancestors. Thus, genomic prediction is far more powerful at predicting almost all of the agricultural traits and human diseases. However, the number of markers m usually exceeds the number of individuals n, $m \gg n$, making the regression model an undetermined system. Much effort has been made by statistical geneticists worldwide to address this problem and to further improve the predictive performance on the traits and diseases (de los Campos, Hickey, Pong-Wong, Daetwyler, and Calus 2013; Wang, Tsuo, Kanai, Neale, and Martin 2022).

The most commonly used strategy to overcome this issue is to construct a genomic relationship matrix (GRM) using all the available markers, and then use this GRM to replace the pedigree-based relationship matrix in the BLUP model, known as genomic BLUP (GBLUP). This approach is simple, robust and has been implemented in some already existing software tools, e.g., **BLUPF90** (Misztal *et al.* 2002) and **HIBLUP** (Yin *et al.* 2023). The GBLUP model assumes that all markers have equal contributions to the population genetic variation as their effects are considered to come from the same distribution. Obviously, this rough assumption is not appropriate for some of the traits, especially for those that are controlled by several major genes. Another more reasonable strategy is to fit a Bayesian multiple regression model, known as individual level Bayesian model, which can assign flexible prior distributions to marker effects (i.e., regression coefficients). The crucial and difficult point is how to define this prior assumption appropriately, because the prediction accuracy of a trait or disease highly depends on how the prior assumptions approximate the practical genetic architectures. Statistical genetics researchers have constantly been devoted to optimizing the prior assumption of the marker effects distribution, and a series of Bayesian methods have been proposed, collectively being called "Bayesian alphabet" (Gianola 2013). However, none of the methods can consistently outperform the others across different traits or diseases, because the underlying genetic architecture is far more complex than the assumptions of a model (Wray, Wijmenga, Sullivan, Yang, and Visscher 2018). Package **BGLR** (Pérez and de los Campos 2014) is the most widely used tool to fit an individual level Bayesian model. Nevertheless, the methods implemented in **BGLR** are limited and its use needs to be complemented with the use of more advanced methods that are proposed lately to meet the needs of researchers.

In practice, it is always difficult to genotype all individuals for a very large population. To also utilize the phenotypic observations of non-genotyped individuals, Fernando, Dekkers, and Garrick (2014) proposed a single-step Bayesian regression model, which could simultaneously integrate pedigree, genotype, and phenotype data in a Bayesian linear model. The core idea of the single-step Bayesian regression model is to impute the genotype of non-genotyped individuals in pedigree from the information of genotyped individuals by using a pedigree-based additive relationship matrix. As more phenotypic observations are used in the model, its prediction accuracy for genotyped individuals is higher than that of the individual level Bayesian model, and for non-genotyped individuals, the prediction accuracy also increases significantly compared with the pedigree-based BLUP model owing to the inclusion of the imputed genotype. Nevertheless, there are only a few tools available to fit the single-step Bayesian model. Package **JWAS** developed by Cheng, Fernando, and Garrick (2018), written in Julia (Bezanson, Edelman, Karpinski, and Shah 2017), is currently the only choice. However, the number of users and developers worldwide who use Julia is not comparable with the number who work in R.

To fit the individual level or single-step Bayesian regression model, the individual level data including the genome-wide markers and one or several phenotypes measured on the same individuals necessarily need to be provided. However, the individual level data are sometimes not publicly accessible for some reasons of protection of personal privacy and legal or non-legal policies, especially in the field of human-related research. Therefore, there are now continuously increasing genome-wide association studies (GWAS) summary statistical datasets publicly available on hundreds of complex traits, each of which consists of the estimated marginal effect and variance at millions of markers. The restricted access to individual level data has motivated statistical geneticists to develop new methodological frameworks that only require publicly available summary level data. In recent years, Zhu and Stephens (2017) firstly introduced the individual level Bayesian regression model into summary level statistical analyses, and then a new advanced method named "SBayesR" was proposed (Lloyd-Jones *et al.* 2019). The results showed that SBayesR outperformed any of the other existing non-Bayesian methods in terms of prediction accuracy. The summary level Bayesian model successfully transforms the determinant of computational complexity from the number of individuals into the number of markers, making it also very promising for handing the livestock breeding data with large number of individuals that are genotyped by only few markers. As the summary level Bayesian model is still fresh to the public, currently the only tool to fit the summary level Bayesian model is **GCTB**, written in C++ by Zeng *et al.* (2018).

In addition to genomic prediction, Bayesian regression models can also be applied to genome-wide association studies (GWAS; Fernando and Garrick 2013), which test the statistical significance of regression coefficients for genomic markers. Since the first publication in 2002 (Ozaki *et al.* 2002), GWAS had great success in locating candidate genes for human diseases, as well as for plant and animal agricultural economic traits, bringing new insights into understanding of the genetic architecture for complex traits. Meantime, a series of advanced models have been developed for GWAS to overcome the population confounding problem which may cause false positive associations. Some of these models include the general linear model in

Plink (Purcell *et al.* 2007), the mixed linear model in **GCTA** (Yang, Lee, Goddard, and Visscher 2011), and the multiple locus "FarmCPU" model in **rMVP** (Yin *et al.* 2021). However, because all the models above are based on testing one marker or only small parts of markers jointly at a time, the significant associations explain only a small fraction of the genetic variance of traits. Additionally, as the number of tests for these models can be very large, controlling the genome-wise error rate results in very low power. In contrast, the Bayesian regression model simultaneously fits all markers jointly as random effects and could be able to account for more genetic variance. An additional advantage is that the power of detecting associations is not inversely related to the number of tested markers (Fernando, Toosi, Wolc, Garrick, and Dekkers 2017). Thus, the Bayesian regression model is an alternative option for GWAS analysis, but implementations in software tools are still not readily available.

As discussed above, the three types of Bayesian models can utilize different types of data and have unique application scenarios in a wide range of research topics. On the one hand, although different software tools have been developed for different types of Bayesian regression models, the provided models and methods are usually limited. For example, packages **lme4** and **brms** can fit the general mixed-effect model, but cannot handle the tremendous number of genomic markers; package **BGLR** can only fit the individual level Bayesian model, and it needs to be freshened with more advanced methods; package **JWAS** cannot implement the summary level Bayesian regression model; and the **GCTB** software cannot be used to fit the single-step model. On the other hand, the existing tools are written via different programming languages, such that their joint use requires users to be familiar with multiple programming languages (e.g., R, Julia, shell scripts), to distinguish the numerous functional parameters and to adapt to the different usage style of various tools. Moreover, the format of the input file for the same data (e.g., genotype) even varies across tools, and the returned results are hard to keep unified, making it expensive to get a comfortable user experience in terms of effort and time spent. The reasons described above motivated us to develop a comprehensive tool which enables the users to easily implement three types of Bayesian regression models with flexible parameter settings to conveniently switch to any of to desired methods.

Herein, we present package **hibayes** (Yin, Zhang, and Liu 2025), a feature-rich and user-friendly package developed on the open-source platform R. The package contains the richest methods achieved thus far, and it is the only tool that can simultaneously fit three types of Bayesian models using individual level, summary level, and individual plus pedigree level (single-step) data for both genomic prediction and genome-wide association studies. It was designed to estimate joint effects of markers, genetic and non-genetic parameters for a complex trait, including: (1) fixed effects and regression coefficients; (2) environmental random effects and their variances; (3) genetic variance; (4) residual variance; (5) heritability; (6) genomic estimated breeding values for both genotyped and non-genotyped individuals; (7) marker effects; (8) phenotype/genetic variance explained (PVE) by single or multiple markers; (9) window posterior probability of association (WPPA); and (10) posterior inclusive probability (PIP). The provided functionalities of package **hibayes** are not fixed, and we will keep enriching package **hibayes** with more features according to the feedback from users worldwide. In Table 1, we roughly compare the inputs, direct returns, available methods and models, and functionalities involved in most of the genetic analyses for the implementations in the packages **brms**, **BGLR**, **JWAS**, **GCTB**, and **hibayes**. It is obvious that package **hibayes** could be more attractive and competitive than others. Thus we believe that package **hibayes** will facilitate academic research and practical application of statistical genetics for humans, plants, and

Models	Features	**brms**	**BGLR**	**JWAS**	**GCTB**	**hibayes**
	Language	R	R, C++	Julia	C++	R, C++
	Version	2.19.0	1.0.9	1.0.0	2.04.3	3.0.0
Individual level Bayesian model	Covariates	✓	✓	✓	✓	✓
	Fixed effects	✓	✓	✓	×	✓
	Random effects	✓	×	✓	✓	✓
	Envir-interactions	✓	✓	✓	×	✓
	Variance components	✓	✓	✓	✓	✓
	PIP	×	×	✓	✓	✓
	WPPA	×	×	✓	×	✓
	Marker effects	×	✓	✓	✓	✓
	GEBVs	×	×	✓	×	✓
	Residuals	✓	×	×	×	✓
	Available methods	×	RR, A, B, Cpi, L	RR, A, B, Bpi, C, Cpi, L	RR, A, B, Bpi, C, Cpi, S, R	RR, A, B, Bpi, C, Cpi, L, BSLMM, R
Summary level Bayesian model	Variance components	×	×	×	✓	✓
	PIP	×	×	×	✓	✓
	WPPA	×	×	×	×	✓
	Marker effects	×	×	×	✓	✓
	Available methods	×	×	×	C, Cpi, R	CG, RR, A, B, Bpi, C, Cpi, L, R
Single-step Bayesian model	Covariates	×	×	✓	×	✓
	Fixed effects	×	×	✓	×	✓
	Random effects	×	×	✓	×	✓
	Envir-interactions	×	×	✓	×	✓
	Variance components	×	×	✓	×	✓
	PIP	×	×	✓	×	✓
	WPPA	×	×	✓	×	✓
	Marker effects	×	×	✓	×	✓
	GEBVs	×	×	✓	×	✓
	Residuals	×	×	×	×	✓
	Available methods	×	×	RR, A, B, Bpi, C, Cpi, L	×	RR, A, B, Bpi, C, Cpi, L, R

Table 1: Rough comparisons of inputs, direct returns, and available models and methods in the domain of statistical genetics for **brms**, **BGLR**, **JWAS**, **GCTB**, and **hibayes**.

animals. The **hibayes** package is free software, licensed under the Apache License 2.0, openly available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=hibayes, and the latest version in development could be installed from GitHub at https://github.com/YinLiLin/hibayes.

2. Description of models and methods

In Bayesian statistics, inferences of unknown parameters of a model are based on their posterior distributions. Let $\boldsymbol{\theta}$ denote all the unknown parameters in the model, then the density function of full conditional distributions can be expressed as

$$f(\theta_i|\boldsymbol{\theta}_{-i}, \boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta})f(\theta_i)f(\boldsymbol{\theta}_{-i})}{f(\boldsymbol{\theta}_{-i}, \boldsymbol{y})} \propto f(\boldsymbol{y}|\boldsymbol{\theta})f(\theta_i)f(\boldsymbol{\theta}_{-i}), \tag{1}$$

where θ_i and $\boldsymbol{\theta}_{-i}$ are the ith element and all other elements in $\boldsymbol{\theta}$, respectively. $\boldsymbol{y}$ is the vector of observations. $f(\boldsymbol{y}|\boldsymbol{\theta})$ is the density function of the conditional distribution of $\boldsymbol{y}$ given the values of the unknowns specified by $\boldsymbol{\theta}$. $f(\theta_i)$ and $f(\boldsymbol{\theta}_{-i})$ are the densities of the prior distributions of θ_i and $\boldsymbol{\theta}_{-i}$. Markov chain Monte Carlo (MCMC) sampling is commonly used to draw inferences from posterior distributions, and the most widely used method to construct such a Markov chain is the Gibbs sampler. By implementing a MCMC iterative process, all the elements in $\boldsymbol{\theta}$ can be inferred from a number of iterations.

2.1. Individual level Bayesian model

The individual level Bayesian model is essentially a version of a mixed-effect model, which describes the phenotypic observations as a function of some variables, including fixed effects, covariates, environmental random effects, and high dimensional genomic markers. The model could be mathematically formulated as

$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{X\beta} + \boldsymbol{Rr} + \boldsymbol{M\alpha} + \boldsymbol{e}, \tag{2}$$

where $\boldsymbol{y}$ is the vector of the phenotypic observations, $\boldsymbol{\mu}$ is the intercept, $\boldsymbol{X}$ represents the design matrix for the fixed effects and covariates, and $\boldsymbol{\beta}$ denotes its regression coefficients. $\boldsymbol{R}$ is the design matrix for environmental factors and $\boldsymbol{r}$ denotes the estimated effects. To simplify the model, it is common to assume that the random effects are independent with each other, thus the covariances among random effects are generally ignored. This implies that the elements of $\boldsymbol{r}$ are assumed to be independent and identically distributed as $\mathcal{N}(0, \sigma_r^2)$. $\boldsymbol{M}$ is the genotype covariate matrix (e.g., 0, 1, 2 for genotype AA, AB, BB) with a dimension of $n \times m$, where n and m are the number of individuals and genomic markers, and $\boldsymbol{\alpha}$ denotes the estimated marker effects, of which the elements are assumed to follow a mixture of normal distributions (i.e., zero mean and different variances $\boldsymbol{\sigma}_\alpha^2$) with the mixing proportions $\boldsymbol{\pi}$. $\boldsymbol{e}$ is the vector of residuals which independently follow the normal distribution $\mathcal{N}(0, \sigma_e^2)$. Let $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \sigma_r^2, \boldsymbol{\sigma}_\alpha^2, \sigma_e^2)$ denote all the unknown parameters in Model 2.

The full conditional distribution of a parameter in the Gibbs sampler is constructed by using the most recent estimates of the other parameters. Thus, it is required to adjust the phenotype observations $\boldsymbol{y}$ for all other effects in the model, which is considered to be time-consuming due to large matrix operations. In package **hibayes**, we implemented another commonly used strategy which employs the real-time model residuals $\boldsymbol{y}^*$ and the sampled value of the parameter at the previous iteration to construct its full conditional distribution. The detailed descriptions are given in Appendix A. To start up the MCMC iterations, all unknown parameters should be initialized to get $\boldsymbol{y}^*$ as follows:

$$\boldsymbol{y}^* = \boldsymbol{y} - \boldsymbol{\mu}^{[0]} - \boldsymbol{X\beta}^{[0]} - \boldsymbol{Rr}^{[0]} - \boldsymbol{M\alpha}^{[0]}, \tag{3}$$

where $\boldsymbol{\mu}^{[0]}$, $\boldsymbol{\beta}^{[0]}$, $\boldsymbol{r}^{[0]}$, $\boldsymbol{\alpha}^{[0]}$ are the start values for $\boldsymbol{\mu}$, $\boldsymbol{\beta}$, $\boldsymbol{r}$, $\boldsymbol{\alpha}$, respectively. In package **hibayes**, the intercept $\boldsymbol{\mu}^{[0]}$ is set to the average of the dependent variable $\boldsymbol{y}$, and $\boldsymbol{\beta}^{[0]}$, $\boldsymbol{r}^{[0]}$, and $\boldsymbol{\alpha}^{[0]}$ are set to zeros.

Fixed effects and environmental random effects

When including a recorded environmental factor in the model, it is necessary to determine whether to treat it as a fixed or random effect. If the levels of the records for this factor can fully cover all the possibilities of the entire population, it is usually treated as a fixed effect, otherwise it is in general more appropriate to be fitted as random term in the model.

The fixed effects and regression coefficients $\boldsymbol{\beta}$ are assumed to have flat priors as discussed by Sorensen and Gianola (2002), i.e., improper uniform priors. Thus the density function is a constant, denoted by $f(\boldsymbol{\beta}) \propto c$. By employing the ordinary least squares (OLS) method, the full conditional distribution of β_j can be easily constructed, as described in Appendix B.

The environmental random effects $\boldsymbol{r}$ are a priori assumed to follow the normal distribution $\mathcal{N}(0, \boldsymbol{I}\sigma_r^2)$, and can be either estimated by a single-site Gibbs sampler (Sorensen and Gianola 2002) or a block Gibbs sampler (García-Cortés and Sorensen 1996; Lund and Jensen 1999). The variance σ_r^2 is a priori assumed to have a scaled inverse chi-square distribution. The construction of the full conditional distribution and the detailed sampler schedule for the random effects $\boldsymbol{r}$ and the variance σ_r^2 can be found in Appendix C.

Genomic marker effects

How many markers potentially affect the phenotype ($\boldsymbol{\pi}$) and which distribution ($\boldsymbol{\sigma}_\alpha^2$) their effects come from are the key and difficult points in estimation of the marker effects. The prediction accuracy of a trait or disease depends on how well the prior assumptions of marker effects align with the actual genetic architecture. The closer these assumptions reflect the true genetic architecture, the higher the prediction accuracy will be. However, the true genetic architecture of a trait or disease is cryptic and complex, which makes the prior assumption of marker effects very challenging. At present, the prior assumption of marker effects can be essentially classified into two categories. The first category assumes the effect of each marker to follow an independent normal distribution with a zero mean and unique variance,

$$(\alpha_j \mid \pi_0, \sigma_{\alpha_j}^2) \sim \begin{cases} 0 & \text{with probability } \pi_0, \\ \mathcal{N}\left(0, \sigma_{\alpha_j}^2\right) & \text{with probability } 1 - \pi_0, \end{cases} \tag{4}$$

where π_0 is the probability of being assigned to the zero effect component for the jth marker (or generally known as the proportion of markers with zero effect). The methods BayesA, BayesB, BayesBpi, BayesLASSO are based on the above prior category. The second category of priors also assumes independent normality, but it allocates the genomic markers into different groups, where the markers in the same group share the same variance and the variances of different groups are varied,

$$(\alpha_j \mid \boldsymbol{\pi}, \sigma_{\alpha_j}^2) \sim \begin{cases} 0 & \mathbb{P}(\sigma_{\alpha_j}^2 = 0) = \pi_0 \\ \mathcal{N}\left(0, \sigma_{\alpha_1}^2\right) & \mathbb{P}(\sigma_{\alpha_j}^2 = \sigma_{\alpha_1}^2) = \pi_1 \\ \vdots & \\ \mathcal{N}\left(0, \sigma_{\alpha_k}^2\right) & \mathbb{P}(\sigma_{\alpha_j}^2 = \sigma_{\alpha_k}^2) = \pi_k \end{cases} \tag{5}$$

Method	Prior distribution	Brief description	Reference
BayesRR	$\alpha_j \sim \mathcal{N}(0, \sigma^2_\alpha)$	all markers have effects with same variance	Meuwissen *et al.* (2001)
BayesA	$\alpha_j \sim \mathcal{N}(0, \sigma^2_{\alpha_j})$ $\sigma^2_{\alpha_j} \sim \chi^{-2}(\nu, S)$	all markers have effects with unique variances	Meuwissen *et al.* (2001)
BayesB	$\alpha_j \sim 0.95\delta_0 + 0.05\mathcal{N}(0, \sigma^2_{\alpha_j})$ $\sigma^2_{\alpha_j} \sim \chi^{-2}(\nu, S)$	5% markers have effects with unique variances	Meuwissen *et al.* (2001)
BayesBpi	$\alpha_j \sim \pi_0\delta_0 + (1-\pi_0)\mathcal{N}(0, \sigma^2_{\alpha_j})$ $\sigma^2_{\alpha_j} \sim \chi^{-2}(\nu, S)$	$(1-\pi_0)$ markers have effects with unique variances, π_0 is estimated	Meuwissen *et al.* (2001)
BayesC	$\alpha_j \sim 0.95\delta_0 + 0.05\mathcal{N}(0, \sigma^2_\alpha)$	5% markers have effects with same variance	Habier, Fernando, Kizilkaya, and Garrick (2011)
BayesCpi	$\alpha_j \sim \pi_0\delta_0 + (1-\pi_0)\mathcal{N}(0, \sigma^2_\alpha)$	$(1-\pi_0)$ markers have effects with same variance, π_0 is estimated	Habier *et al.* (2011)
BayesL	$\alpha_j \sim \mathcal{N}(0, \sigma^2_{\alpha_j})$ $\sigma^2_{\alpha_j} \sim \mathit{Expon}(\lambda^2/2)$	all markers have effects with unique variances	Yi and Xu (2008)
BSLMM	$\alpha_j \sim (1-\pi)\mathcal{N}(0, \sigma^2_{\alpha_1})$ $+ \pi\mathcal{N}(0, \sigma^2_{\alpha_1} + \sigma^2_{\alpha_2})$	all markers have effects and are assigned into 2 groups with unique variances, π is estimated	Zhou, Carbonetto, and Stephens (2013)
BayesR	$\alpha_j \sim \pi_0\delta_0 + \pi_1\mathcal{N}(0, 10^{-4}\sigma^2_\alpha)$ $+\pi_2\mathcal{N}(0, 10^{-3}\sigma^2_\alpha) + \pi_3\mathcal{N}(0, 10^{-2}\sigma^2_\alpha)$	$(1-\pi_0)$ markers have effects and are assigned into 3 groups with graded variances, $\boldsymbol{\pi}$ is estimated	Moser, Lee, Hayes, Goddard, Wray, and Visscher (2015)

Table 2: Prior assumptions of marker effect distributions for the methods implemented in package **hibayes**. δ_0 represents the size of the marker effect equal to zero, other mathematical symbols are completely consistent with that in the main text.

where π_0 and π_l, $l = 1, \ldots, k$ are the sub-elements of the vector $\boldsymbol{\pi}$. π_l is the probability of being assigned to the lth normal distribution (i.e., the $l+1$th value in the vector of $\boldsymbol{\pi}$ where π_0 is the first element), of which the variance is $\sigma^2_{\alpha_l}$. Futhermore, $\pi_0 + \pi_1 + \ldots + \pi_k = 1$, and $\mathbb{P}$ is the probability. The methods BayesRR, BayesC, BayesCpi, BSLMM, BayesR rely on the second type of priors. Since different methods have various prior assumptions, none of the methods can always outperform the others across different traits (Yin *et al.* 2020; Meher, Rustgi, and Kumar 2022). It is thus valuable and necessary to integrate more methods in a software tool to accommodate the wide complexity of genetic architecture of traits and diseases. The different prior assumptions on the marker effects and the variance distributions for the methods in package **hibayes** are summarized in Table 2. When marker effect α_j is non-zero, its full conditional distribution can be mathematically formulated as follows:

$$(\alpha_j^{[i]} \mid \boldsymbol{y}^*, \boldsymbol{M}, \alpha_j^{[i-1]}, \sigma^2_{\alpha_j}, \sigma^2_e) \sim \mathcal{N}\left(\frac{\boldsymbol{M}_j^\top \boldsymbol{y}^* + \boldsymbol{M}_j^\top \boldsymbol{M}_j \alpha_j^{[i-1]}}{\boldsymbol{M}_j^\top \boldsymbol{M}_j + \sigma^2_e/\sigma^2_{\alpha_j}}, \frac{\sigma^2_e}{\boldsymbol{M}_j^\top \boldsymbol{M}_j + \sigma^2_e/\sigma^2_{\alpha_j}}\right), \tag{6}$$

where $\boldsymbol{y}^*$ is the real-time model residuals which can be obtained on the conditions of $\boldsymbol{y}$ and the sampled $\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\alpha}$. $\alpha_j^{[i]}$ represents the jth element of the marker effect $\boldsymbol{\alpha}$ at the ith Markov chain iteration. $\boldsymbol{M}_j$ represents the coded genotype vector for the jth markers (column j of matrix $\boldsymbol{M}$). For the methods where the prior assumptions include zero effect markers (i.e., $\pi_0 \neq 0$) or have several groups of normality, it must be known to which distribution the jth genomic marker belongs for the current iteration before sampling α_j from its full conditional distribution. Therefore, it is required to calculate all likelihoods assuming the considered genomic marker j being in one of the n_π (the number of elements of $\boldsymbol{\pi}$) distributions at a time with the respective probability $\boldsymbol{\pi}$. The log likelihood that the jth genomic marker is in distribution k can be expressed as

$$L_{\pi_k} = -\frac{1}{2}\left[\log\left(\frac{\boldsymbol{M}_j^\top \boldsymbol{M}_j \sigma_{\alpha_k}^2}{\sigma_e^2} + 1\right) - \frac{\left(\boldsymbol{M}_j^\top \boldsymbol{y}^* + \boldsymbol{M}_j^\top \boldsymbol{M}_j \alpha_j\right)^2}{(\boldsymbol{M}_j^\top \boldsymbol{M}_j + \sigma_e^2/\sigma_{\alpha_k}^2)\sigma_e^2}\right] + \log(\pi_k). \tag{7}$$

The detailed mathematical derivations for L_{π_k} can be found in Lloyd-Jones *et al.* (2019). Then, as described by Erbe *et al.* (2012), the probability that marker j is in distribution k is

$$\mathbb{P}(\sigma_{\alpha_j}^2 = \sigma_{\alpha_k}^2) = \frac{1}{\sum\limits_{i=1}^{n_\pi} \exp(L_{\pi_i} - L_{\pi_k})}. \tag{8}$$

Based on these probabilities, we can select the corresponding distribution to sample the marker effect by using a uniform random variate u from $U(0,1)$ and the probabilities of the marker being in each of the normal distributions, that is the lowest k where $\pi_k \geq u$. Once the distribution for marker j has been confirmed, the marker effect α_j can be sampled according to Equation 6, and its corresponding model residuals $\boldsymbol{y}^*$ can be updated subsequently as

$$\boldsymbol{y}^* = \boldsymbol{y}^* + \boldsymbol{M}_j(\alpha_j^{[i-1]} - \alpha_j^{[i]}). \tag{9}$$

By repeating the same steps above for markers one by one, the effects $\boldsymbol{\alpha}$ can be obtained for the current Markov chain iteration. Meanwhile, we can record the number of markers allocated into different distributions for the current iteration. Let $\boldsymbol{\eta} = (m_1, m_2, \ldots, m_k, \ldots, m_{n_\pi})$, where $m_1 + m_2 + \ldots + m_k + \ldots + m_{n_\pi} = m$, and m is the total number of markers. If $\boldsymbol{\pi}$ needs to be estimated, it can be sampled from a beta distribution when there are only 2 distributions, e.g., $\boldsymbol{\eta} = (m_{\pi_0}, m - m_{\pi_0})$ with m_{π_0} the number of genomic markers in zero effect, then we have

$$(\pi_0 \mid m_{\pi_0}) \sim \mathit{Beta}(m_{\pi_0} + 1, m - m_{\pi_0} + 1). \tag{10}$$

When the number of distributions is bigger than 2 ($n_\pi > 2$), only the Dirichlet distribution is adaptable for sampling (e.g., for the BayesR method), that is $(\boldsymbol{\pi} \mid \boldsymbol{\eta}) \sim \mathit{Dir}(n_\pi, \boldsymbol{\eta} + \mathbf{1})$.

The prior distribution for the variances of the marker effects is a scaled inverse chi-square distribution. For the methods where each marker has a unique variance as shown in Equation 4, the sampling of the variance should be implemented immediately after the effect of the single marker is updated, and the full conditional distribution is

$$(\sigma_{\alpha_j}^2 \mid \alpha_j) \sim \left(\alpha_j \cdot \alpha_j + \nu_\alpha S_\alpha^2\right) \chi_{\nu_\alpha+1}^{-2}, \tag{11}$$

where ν_α and S_α^2 are the degrees of freedom and the scale parameter, respectively. For the methods where markers are allocated into different normal distributions as shown in

Equation 5, the sampling of the variance is implemented after sampling of the effect for the last genomic marker, and the full conditional distribution can be referred as

$$(\sigma_\alpha^2 \mid \boldsymbol{\alpha}, m_{\pi_0}) \sim \left(\boldsymbol{\alpha}^\top \boldsymbol{\alpha} + \nu_\alpha S_\alpha^2\right) \chi^{-2}_{\nu_\alpha + m - m_{\pi_0}}. \tag{12}$$

In the BayesR method, the variances of the normal distributions of the different groups of markers are scaled relative to a temporary base variance, $\sigma^2_{\alpha_\gamma}$, using predefined scale factors $\boldsymbol{\gamma} = (0, 0.0001, 0.001, 0.01)$. It is not necessary to sample the variance for each group individually; instead, only the base variance $\sigma^2_{\alpha_\gamma}$ is sampled. Details of the sampling procedure for $\sigma^2_{\alpha_\gamma}$ are provided in Appendix D. The BSLMM method has an additional multivariate normal term in the model compared to the BayesCpi method. We thus implemented an efficient eigen decomposition based block Gibbs sampler under the framework of the BayesCpi method; more details are given in Appendix E.

Residual effects

The vector of model residuals $\boldsymbol{y}^*$ is updated after any of the model parameters $(\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\alpha})$ is sampled from its corresponding full conditional distribution. The prior assumption on the variance of the residuals σ_e^2 is that it follows a scaled inverse chi-square distribution, which can be updated at the end of every single Markov chain iteration by the following full conditional distribution,

$$(\sigma_e^2 \mid \boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\alpha}, \boldsymbol{y}) \sim \left(\boldsymbol{y}^{*\top} \boldsymbol{y}^* + \nu_e S_e^2\right) \chi^{-2}_{\nu_e + n}, \tag{13}$$

where $\nu_e = -2$, and $S_e^2 = 0$ by default in package **hibayes**.

2.2. Summary level Bayesian model

Restricted access to individual level data has motivated methodological frameworks that only require publicly available summary level data, one of which is the Bayesian summary statistics proposed in recent years (Zhu and Stephens 2017; Lloyd-Jones *et al.* 2019). The summary level Bayesian model can be written as

$$\boldsymbol{b} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{B} \boldsymbol{D}^{\frac{1}{2}} \boldsymbol{\alpha} + \boldsymbol{e}^*, \tag{14}$$

which can be inferred from the individual level Bayesian model as detailed in Appendix F. $\boldsymbol{b}$ is the vector of the marginal effects (also known as regression coefficients), which can be obtained from summary data directly. $\boldsymbol{D}$ is a diagonal matrix with diagonal elements given by $\{\boldsymbol{M}_j^\top \boldsymbol{M}_j\}_{j \in \{1,\ldots,m\}}$. These are equal to $\{2 n_j p_j q_j\}_{j \in \{1,\ldots,m\}}$ if the markers are in Hardy-Weinberg equilibrium, where $\boldsymbol{n}$, $\boldsymbol{p}$, and $\boldsymbol{q}$ are the sample size, the frequency of the reference allele and the frequency of the alternate allele for all markers. All these quantities can also be obtained from summary data. $\boldsymbol{B}$ is the correlation matrix of all genomic markers, it can be derived from the publicly available reference genotype panel (e.g., the 1000 Genomes Project, The 1000 Genomes Project Consortium 2015). The unknown parameters of Model 14 are $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{\sigma}_\alpha^2, \sigma_e^2)$.

As described in Section 2.1, to sample the marker effect α_j using Equation 6 and to calculate the log likelihoods for all distributions using Equation 7 for marker j, it requires the key components including $\boldsymbol{M}_j^\top \boldsymbol{M}_j$ and $\boldsymbol{M}_j^\top \boldsymbol{y}^*$. $\boldsymbol{M}_j^\top \boldsymbol{M}_j$ is the diagonal element of $\boldsymbol{D}$, which can be easily accessed by D_{jj}. The main challenge is to get $\boldsymbol{M}_j^\top \boldsymbol{y}^*$. Since $\boldsymbol{y}^*$ is the vector of model residuals and the phenotype observations $\boldsymbol{y}$ are not available in summary data, $\boldsymbol{y}^*$ cannot be accessed and updated directly in Markov chains. Instead of computing and storing $\boldsymbol{y}^*$ as in the individual level model, this problem can be addressed by computing and storing $\boldsymbol{M}^\top \boldsymbol{y}^*$ for the summary level model. Multiplying Equation 9 by $\boldsymbol{M}^\top$ one arrives at

$$\boldsymbol{M}^\top \boldsymbol{y}^* = \boldsymbol{M}^\top \boldsymbol{y}^* + \boldsymbol{M}^\top \boldsymbol{M}_j (\alpha_j^{[i-1]} - \alpha_j^{[i]}), \tag{15}$$

since the variance-covariance matrix $\boldsymbol{V} = \boldsymbol{M}^\top \boldsymbol{M}/n = \boldsymbol{D}^{\frac{1}{2}} \boldsymbol{B} \boldsymbol{D}^{\frac{1}{2}}$, the component $\boldsymbol{M}^\top \boldsymbol{M}_j$ can be written as $\boldsymbol{M}^\top \boldsymbol{M}_j = n_j \boldsymbol{V}_j$, where $\boldsymbol{V}_j$ is a vector of the jth column of $\boldsymbol{V}$, which can be obtained on the basis of $\boldsymbol{D}$ and $\boldsymbol{B}$. Let $\boldsymbol{\phi} = \boldsymbol{M}^\top \boldsymbol{y}^*$, we can rewrite the equation above as

$$\boldsymbol{\phi} = \boldsymbol{\phi} + n_j \boldsymbol{V}_j (\alpha_j^{[i-1]} - \alpha_j^{[i]}). \tag{16}$$

At the beginning of the MCMC iterations, the start value is $\boldsymbol{\phi}^{[0]} = \boldsymbol{D}\boldsymbol{b}$ and we can refresh $\boldsymbol{\phi}$ for any update of $\boldsymbol{\alpha}$ to implement a Gibbs sampler. Following Equation 6, the full conditional distribution of α_j can be written as

$$(\alpha_j^{[i]} \mid \boldsymbol{\phi}, \boldsymbol{D}, \alpha_j^{[i-1]}, \sigma_{\alpha_j}^2, \sigma_e^2) \sim \mathcal{N}\left(\frac{\boldsymbol{\phi}_j + D_{jj}\alpha_j^{[i-1]}}{D_{jj} + \sigma_e^2/\sigma_{\alpha_j}^2}, \frac{\sigma_e^2}{D_{jj} + \sigma_e^2/\sigma_{\alpha_j}^2} \right) \tag{17}$$

The log likelihoods for different normal distributions in Equation 7 can also be easily obtained accordingly following the same transformations as above. And the other elements of the Gibbs sampling routine are the same as for the individual level data model, except for the sampling of σ_e^2, which is outlined in Appendix F.

The available methods for the summary level Bayesian model in package **hibayes** are nearly the same as for the individual level Bayesian model, except for method BSLMM. Method BSLMM is not included because the individual genotype is inaccessible to construct the GRM. Looking through the theoretic derivations of the summary level Bayesian model above, we can find that a big problem is to compute the correlation matrix $\boldsymbol{B}$ fast and store it. The dimension of $\boldsymbol{B}$ equals the number of genomic markers in the analysis. Although it is computationally acceptable for chip array data at a level of several tens of thousands of markers, the computational burden caused by high density markers from sequencing ($> 100k$) would be a big challenge to face in the summary level Bayesian model. However, typically this problem can be addressed by using a fixed 1–10Mb window approach, which sets correlation values outside this window to zeros, or uses a shrunk linkage disequilibrium (LD) matrix proposed by Zhu and Stephens (2017). In package **hibayes**, we use a chi-square threshold $\hat{x}^2$ to make the $\boldsymbol{B}$ matrix sparse. If the condition $r^2/n < \hat{x}^2$ (r is the correlation of two markers) is met, then the correlation for these two markers will be set to zero, which can significantly reduce the memory consumption in analysis. However, as a consequence of the sparsification processing, certain elements of $\boldsymbol{\phi}$ in Equation 16 remain unaltered, leading to a biased mean in the full conditional distribution of Equation 17 for the markers associated with those elements. In certain situations, this bias may cause the MCMC iterations to encounter "blow up" issues. In this case, we recommend adjusting the chi-square threshold $\hat{x}^2$ to address this problem.

2.3. Single-step Bayesian model

The single-step GBLUP (SSGBLUP) model makes it possible to connect all phenotypic observations for both genotyped and non-genotyped individuals (Christensen and Lund 2010; Aguilar, Misztal, Johnson, Legarra, Tsuruta, and Lawlor 2010). However, on the one hand, SSGBLUP model requires the GRM and its inverse of all genotype individuals, which is an inefficient process; on the other hand, it assumes that all genomic markers have equal contributions to the phenotype, which is inconsistent with the real genetic architecture of traits. To overcome these issues, Fernando *et al.* (2014) and Fernando, Cheng, Golden, and Garrick (2016) proposed the single-step Bayesian regression (SSBR) model, which can be formulated as

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} = \boldsymbol{\mu} + \begin{bmatrix} \boldsymbol{X}, \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{J}_2 \\ \boldsymbol{X}, \boldsymbol{J}_2 \end{bmatrix} \boldsymbol{\beta} + \boldsymbol{R}\boldsymbol{r} + \begin{bmatrix} \boldsymbol{Z}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{M}_2\boldsymbol{\alpha} + \boldsymbol{\epsilon} \\ \boldsymbol{M}_2\boldsymbol{\alpha} \end{bmatrix} + \boldsymbol{e}, \tag{18}$$

where $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ are the phenotypic records for non-genotyped and genotyped individuals, respectively, $\boldsymbol{J}_2 = -\boldsymbol{1}$, $\boldsymbol{A}_{12}$ is the pedigree based additive relationship matrix between non-genotyped and genotyped individuals, $\boldsymbol{A}_{22}^{-1}$ is the inverse of the pedigree based additive relationship matrix between genotyped individuals, $\boldsymbol{M}_2$ is the genotype covariate matrix for genotyped individuals, $\boldsymbol{\epsilon}$ is the vector of imputation residuals for non-genotyped individuals, following a multivariate normal distribution, and the other symbols are the same as in Model 2. The unknown parameters for Model 18 are $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\epsilon}, \boldsymbol{\alpha}, \boldsymbol{\pi}, \sigma_r^2, \sigma_\epsilon^2, \boldsymbol{\sigma}_\alpha^2, \sigma_e^2)$.

The core idea of the SSBR model is to impute the genotype of the non-genotyped individuals in pedigree conditional on the genotyped individuals by using the pedigree based additive relationship matrix $\boldsymbol{A}$. However, constructing $\boldsymbol{A}_{12}$ and computing the inverse of $\boldsymbol{A}_{22}$ in Model 18 are not very efficient with increasing size of pedigree. Fortunately, Fernando *et al.* (2014) proved that the imputed markers can be obtained efficiently, using partitioned inverse results, by solving the easily formed very sparse system $\boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1} = (\boldsymbol{A}^{11})^{-1}(-\boldsymbol{A}^{12})$, where $\boldsymbol{A}^{11}$ is the partition of $\boldsymbol{A}^{-1}$ for non-genotyped individuals, $\boldsymbol{A}^{12}$ is the partition of $\boldsymbol{A}^{-1}$ between non-genotyped and genotyped individuals. As $\boldsymbol{A}^{11}$ is very sparse, it will be quite fast to get the solution on the left hand side by solving the sparse linear system on the right hand side. Moreover, Henderson (1976) presented a simple procedure to derive $\boldsymbol{A}^{-1}$ from pedigree directly without inverting matrix $\boldsymbol{A}$. Once the genotype imputation is done successfully, it is easy to implement the single-step Bayesian model, which is almost the same as the individual level Bayesian model, except for the imputation residuals $\boldsymbol{\epsilon}$, which are assumed to follow a multivariate normal distribution with zero mean and a variance of $(\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})\sigma_\epsilon^2$. By employing the block-partitioned matrix inversion lemma, we have $(\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})^{-1} = \boldsymbol{A}^{11}$. Then the conditional solution of $\boldsymbol{\epsilon}$ is given by the following system:

$$\left(\boldsymbol{Z}_1^\top \boldsymbol{Z}_1 + \boldsymbol{A}^{11}\frac{\sigma_e^2}{\sigma_\epsilon^2}\right)\boldsymbol{\epsilon}^{[i]} = \boldsymbol{Z}_1^\top(\boldsymbol{y}_1^* + \boldsymbol{Z}_1\boldsymbol{\epsilon}^{[i-1]}), \tag{19}$$

where $\boldsymbol{y}_1^*$ is the sub-vector of $\boldsymbol{y}^*$ (i.e., the model residuals) for non-genotyped individuals. In package **hibayes**, we draw $\boldsymbol{\epsilon}^{[i]}$ using the single-site Gibbs sampler as described in Appendix C. σ_ϵ^2 is the variance of the imputation residuals $\boldsymbol{\epsilon}$, which is a priori assumed to follow a scaled inverse chi-square distribution, and its full conditional distribution is

$$(\sigma_\epsilon^2 \mid \boldsymbol{A}, \boldsymbol{\epsilon}) \sim \left(\boldsymbol{\epsilon}^\top \boldsymbol{A}^{11}\boldsymbol{\epsilon} + \nu_\epsilon S_\epsilon^2\right)\chi^{-2}_{\nu_\epsilon + n_\epsilon}, \tag{20}$$

where n_ϵ is the number of elements in $\boldsymbol{\epsilon}$, ν_ϵ and S_ϵ^2 are the pre-defined degrees of freedom and the scale parameter, respectively. Once all the values in vector $\boldsymbol{\epsilon}$ are sampled, then the model residuals of non-genotyped individuals ($\boldsymbol{y}_1^*$) are updated as follows:

$$\boldsymbol{y}_1^* = \boldsymbol{y}_1^* + \boldsymbol{Z}_1(\boldsymbol{\epsilon}^{[i-1]} - \boldsymbol{\epsilon}^{[i]}) \tag{21}$$

and the residuals of genotyped individuals ($\boldsymbol{y}_2^*$) are kept unchanged. The sampling routine for other unknown parameters in $\boldsymbol{\theta}$ for the single-step Bayesian model is the same as in the individual level Bayesian model and is outlined Section 2.1.

The available methods for the single-step Bayesian model in package **hibayes** are nearly the same than for the individual level Bayesian model as described in Table 2, except for method BSLMM. This method is not included because the BSLMM method requires the GRM of all individuals, but the imputed genotype for non-genotyped individuals cannot be used directly to construct the GRM, because there are no specific imputation residuals for each single marker available, but only for individuals.

2.4. Genomic prediction and genome-wide association studies

Genomic prediction

For genomic selection, the main purpose is to obtain the individual's GEBV (genomic estimated breeding values) of agricultural traits or the PRS (polygenic risk score) of diseases, which measure the genetic merit of individuals. As discussed in the section above, Bayesian regression models estimate the effect for all markers. Thus to obtain the GEBV or PRS, the individual level genotype is required. If the individual genotype is available, the GEBVs can be computed by

$$\boldsymbol{g} = \boldsymbol{M}\boldsymbol{\alpha}, \tag{22}$$

which could be efficiently accomplished with the toolset **PLINK** using the function "`--score`". For the single-step Bayesian model, as described by Fernando *et al.* (2014), the GEBV includes three parts as follows:

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{J}_2 \\ \boldsymbol{J}_2 \end{bmatrix} \boldsymbol{\beta}_J + \begin{bmatrix} \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{M}_2 \\ \boldsymbol{M}_2 \end{bmatrix} \boldsymbol{\alpha} + \begin{bmatrix} \boldsymbol{Z}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{\epsilon}, \tag{23}$$

where the symbols correspond to those in Model 18. The first part comes from the estimated coefficient $\boldsymbol{\beta}_J$, the second part is derived from the genotype, and the third part includes the imputation residuals for non-genotyped individuals. There is no other software or pipeline available that can be used directly to accomplish the above calculation, and it is not that easy for users to implement manually. Therefore, in package **hibayes**, we have reported GEBVs directly for all individuals in the final returned lists.

Genome-wide association studies

The Bayesian regression model cannot only be used for genomic prediction, but also could be applied to genome-wide association studies to locate candidate genes of a trait (Fernando and Garrick 2013). Given such a model where the proportion of markers with zero effects (π_0) is close to one, the posterior probability that α_j is non-zero for at least one marker j in a window or segment can be used to make inferences on the presence of a QTL (quantitative trait locus) in that segment. We refer to this probability as the window posterior probability of association (WPPA). The underlying assumption here is that if a genomic window contains a QTL, one or more markers in that window will have a non-zero α_j. Thus, WPPA, which is estimated by counting the number of MCMC samples in which α_j is non-zero for at least one marker j in the window, can be used as a proxy for the posterior probability that the genomic window contains a QTL. WPPA can be formulated as

$$WPPA = \frac{N_c}{N_t - N_b}, \tag{24}$$

where N_t is the total number of iterations of the Markov chains, N_b is the number of discarded iterations as burn-in of the Markov chains, and N_c is the counted number of times that α_j is non-zero for at least one marker α_j in the window after the discarded iterations. Also, package **hibayes** provides the posterior probability that α_j is non-zero for each single marker, which is known as posterior inclusive probability (PIP) and can be used as a complementary result for more detailed location of causal markers.

3. The R package hibayes

As R has become one of the most widely used languages for statistical computing and graphics, with a large number of users worldwide, we developed package **hibayes** on the R platform. However, as the description of the details of Bayesian regression models in Section 2 shows, there are a huge number of iterations required for parameter estimation and thus it is not a good decision to write Bayesian regression models in pure R. Therefore, we accomplished that the core parts which take up most of the computation time of Bayesian regression models are implemented in the C++ language by the aid of the packages **Rcpp** (Eddelbuettel and François 2011) and **RcppArmadillo** (Eddelbuettel and Sanderson 2014). All parallelizable parts were sped up by **OpenMP** (Dagum and Menon 1998), some basic vector operations were enhanced by calling corresponding functions in **LAPACK** (Anderson *et al.* 1999). All of the above can be sped up automatically by using Intel **MKL** (Math Kernel Library) if it was linked with R by the user. Fast operations for dense and sparse matrices were implemented with the help of the **Matrix** package (Bates, Maechler, and Jagan 2025), and only the main functions used for data and parameter input were written in pure R, granting package **hibayes** with a pretty high computing efficiency.

Regarding the genotype information, it is expensive to code it into a numeric covariate matrix and read it into memory for each analysis. We thus provide an additional function to convert the genotype information into numeric memory-mapping files locally using the package **bigmemory** (Kane, Emerson, and Weston 2013). This only needs to be done at the first time, and no matter how big the number of individuals or markers in the genotype is, the memory-mapping files could be attached into memory on-the-fly within several seconds or minutes, making package **hibayes** very promising in handling big genomic data.

Package **hibayes** is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=hibayes`. The latest development version can be installed from GitHub at `https://github.com/YinLiLin/hibayes`. This article refers to version 3.1.0. The main available R functions provided by package **hibayes** are listed in Table 3. We do not list the arguments for the functions. Users can get the information on all the available

Function	Description
`read_plink`	Converts genotype file in **PLINK** binary format into memory-mapping file format.
`ldmat`	Constructs variance-covariance LD matrix ($\boldsymbol{V}$) of markers across the whole genome to fit a summary level Bayesian model.
`ibrm`	Fits an *i*ndividual level *B*ayesian *r*egression *m*odel.
`sbrm`	Fits a *s*ummary level *B*ayesian *r*egression *m*odel.
`ssbrm`	Fits a *s*ingle-*s*tep *B*ayesian *r*egression *m*odel.
`summary`	Summarizes the results and computes the standard deviation of model parameters.

Table 3: Main functions provided by package **hibayes**.

arguments and their detailed descriptions using `help()` or typing "`?`" in front of the function name in R, for example, `help("read_plink")` or `?read_plink`. Some important arguments are also discussed in the next section.

4. Quick start with simple examples

In the following, we display some examples to run different Bayesian regression models using the tutorial data attached in package **hibayes**, including the input file format, settings of main parameters, summary of the returned results, as well as relevant visualizations of some important genetic parameters. We start by installing and loading package **hibayes**:

```
R> install.packages("hibayes")
R> library("hibayes")
```

4.1. Examples for the individual level Bayesian model

To fit the individual level Bayesian model, the phenotypic observations, environmental records, and the genotype data should be provided. In package **hibayes**, a dataset which contains 500 phenotypic observations and 600 genotyped individuals is included to facilitate the tutorial. The phenotype data can be loaded and inspected using:

```
R> pheno_file_path <- system.file("extdata", "demo.phe",
+    package = "hibayes")
R> pheno <- read.table(pheno_file_path, header = TRUE)
R> dim(pheno)

[1] 500   8

R> head(pheno, 4)

       id  sex season day bwt loc     dam      T1
1 IND1001 Male Winter  92 1.2 132 IND0921  4.7658
```

```
2 IND1002 Male Spring  88 2.7 136 IND0921 12.4098
3 IND1003 Male Spring  91 1.0 117 IND0968  4.8545
4 IND1004 Male Autumn  93 1.0 137 IND0968 33.2217
```

The data contains one trait named T1 and some environmental records including sex, season, day from birth, body weight, and location. Missing values should be marked as 'NA', and the first column of the phenotype data must include the names of the individuals.

The tutorial genotype data was stored in **PLINK** binary files (see details at https://zzz.bwh.harvard.edu/plink/data.shtml#bed). It can be loaded using function read_plink():

```
R> bfile_path <- system.file("extdata", "demo", package = "hibayes")
R> bin <- read_plink(bfile = bfile_path, mode = "A", threads = 4)
R> fam <- bin[["fam"]]
R> geno <- bin[["geno"]]
R> map <- bin[["map"]]
```

The argument bfile indicates the prefix of the binary files, mode can be set to "A" or "D" for additive and dominant genetic effect, respectively. In this function, a missing genotype will be replaced by the major genotype of each allele. For additive mode, the genotype A_1A_1, A_1A_2, A_2A_2 will be coded as 2, 1, 0, respectively. A_1 is the first allele of each marker in map, and thus the estimated effects are on the A_1 allele for all markers. A quick inspection of the loaded genotype data shows:

```
R> dim(geno)
```

```
[1]  600 1000
```

```
R> geno[1:4, 1:5]
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    1    1    1    0
[2,]    1    0    1    1    0
[3,]    0    2    0    0    0
[4,]    1    1    1    1    0
```

The first dimension of geno is the number of genotyped individuals, while the second is the number of genomic markers. By default, the function read_plink() constructs memory-mapped files for genotype data, which are directed into the R temporary folder. Users could redirect to a new directory by the argument out, e.g., bin <- read_plink(bfile = bfile_path, out = "./demo"). The genotype data conversion only needs to be done once, and at the next time of use, no matter how big the number of individuals or markers in the genotype are, the memory-mapping files could be attached into memory on-the-fly within several seconds or minutes using geno <- attach.big.matrix("./demo.desc").

Object fam contains the names of all genotyped individuals, which are used to match the order of individuals between phenotype and genotype data:

```
R> geno.id <- fam[, 2]
R> head(geno.id)
```

```
[1] "IND0701" "IND0702" "IND0703" "IND0704" "IND0705" "IND0706"
```

Object `map` is a data frame which contains the detailed genomic information of markers. Its columns are the marker names, chromosome, physical position, the first allele, and the second allele, respectively. It is only required when implementing GWAS analysis, the format is as follows:

```
R> head(map, 4)

  SNP CHROM     POS A1 A2
1  M1     1 4825340  G  T
2  M2     1 6371512  A  G
3  M3     1 7946983  G  A
4  M4     1 8945290  C  G
```

All the physical positions located at the third column should be in digits because this variable will be used to cut the genome into smaller windows.

Now, we can fit the individual level Bayesian model as follows:

```
R> fitCpi <- ibrm(T1 ~ season + bwt + (1 | loc) + (1 | dam), data = pheno,
+    M = geno, M.id = geno.id, method = "BayesCpi", printfreq = 100,
+    Pi = c(0.98, 0.02), niter = 50000, nburn = 40000, thin = 1,
+    seed = 666666, map = map, windsize = 1e6, verbose = TRUE)
```

The first argument is the model formula of phenotype, fixed effects, and environmental random effects. The environmental random effects are distinguished by vertical bars `(1|·)` separating expressions as implemented in package **lme4**. The fixed effects should be provided as factors and fixed covariates as numeric values. Users can convert the columns of the data into the corresponding format during pre-processing, or convert it in the model formula, e.g., `T1 ~ as.factor(season) + as.numeric(bwt)`. The arguments `M` and `M.id` must be specified for genotype data, because the function `ibrm` can automatically take the intersection and adjust the order of individuals between phenotype and genotype data. Thus there is no need for users to adjust it in advance. Users can choose one of the methods in Table 2 by the argument `method`, change the total number of iterations and discarded number of iterations by the arguments `niter` and `nburn`, respectively. The printed log message records the descriptive information for the input data, the sampled details of the unknown parameters during the Markov chains, the time that remains for running, and summary statistics for some of the main genetic parameters. Also, users can turn off the log message by calling `ibrm` using `ibrm(..., verbose = FALSE)`.

The returned list is an object of class '`blrMod`', which stores all the estimated unknown parameters (e.g., use `str(fitCpi)` to get the details). For genomic prediction, the most important results are the estimated effects of genomic markers and the genomic estimated breeding values (or polygenic risk scores) of individuals. Users can easily access them from the returned object using:

```
R> SNPeffect <- fitCpi$alpha
R> gebv <- fitCpi$g
```

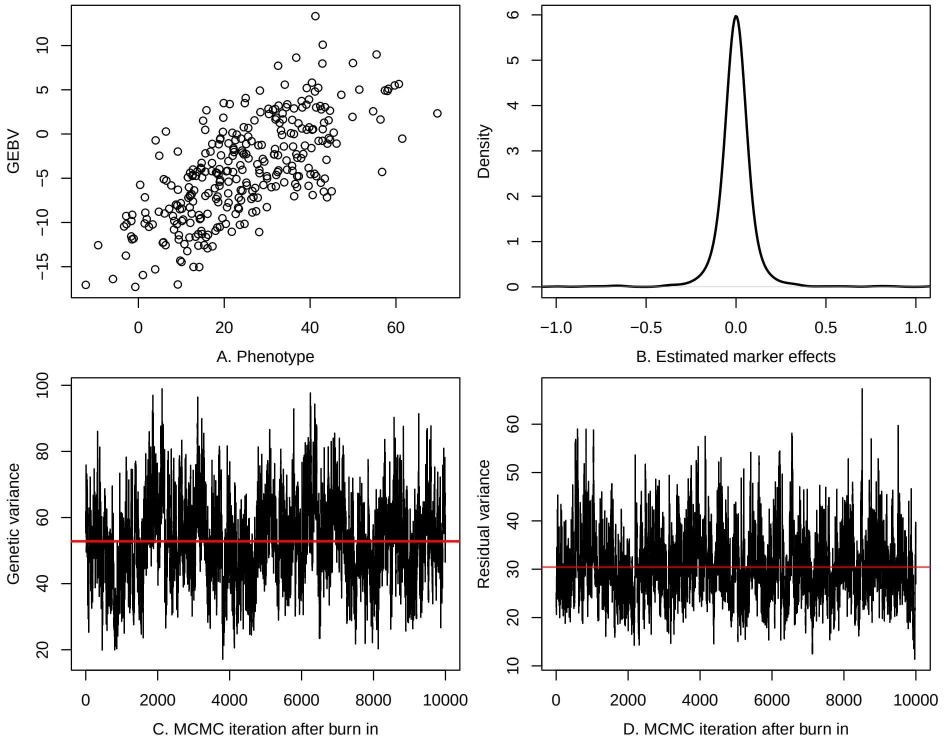

Figure 1: A. the scatter plot between phenotype and estimated GEBV; B. the posterior density distribution of marker effects; C. trace plot of genetic variance; D. trace plot of residual variance. The red line is the ultimately estimated value for the parameter.

`fitCpi$alpha` is the vector of estimated marker effects listed in the same order than the `map`, and the data frame `fitCpi$g` contains the estimated breeding values of all individuals whose names are listed in the first column.

As illustrated in Figure 1A, even though the correlation between phenotype and the estimated GEBVs is high, we cannot select candidate individuals for breeding merely via the original phenotype. The GEBVs should be the more accurate target reflecting the genetic difference among individuals. The accuracy of GEBVs depends on the precise estimation of marker effects. The posterior density of marker effects for the method BayesCpi is given in Figure 1B, where we see a very sharp peak on the distribution, which is fully consistent with its prior assumption that only a small proportion of markers have non-zero effects. As described in Table 2, different Bayesian methods have different prior assumptions on the distribution of marker effects and their variances, resulting in varied prediction performance of a trait or disease. We provide the code for running different Bayesian methods in the replication code file and visualize the posterior distribution of marker effects. The results show very different shapes of density plots for different methods. To quantify the prediction performance of different methods on different traits or diseases, cross-validation is the gold standard in real data analysis.

The return object `fitCpi` contains a list named `MCMCsamples` which records the sampled posterior values for all unknown parameters. Users can visualize them to check if the Markov chains have converged successfully for a parameter of interest. As shown in Figure 1C and

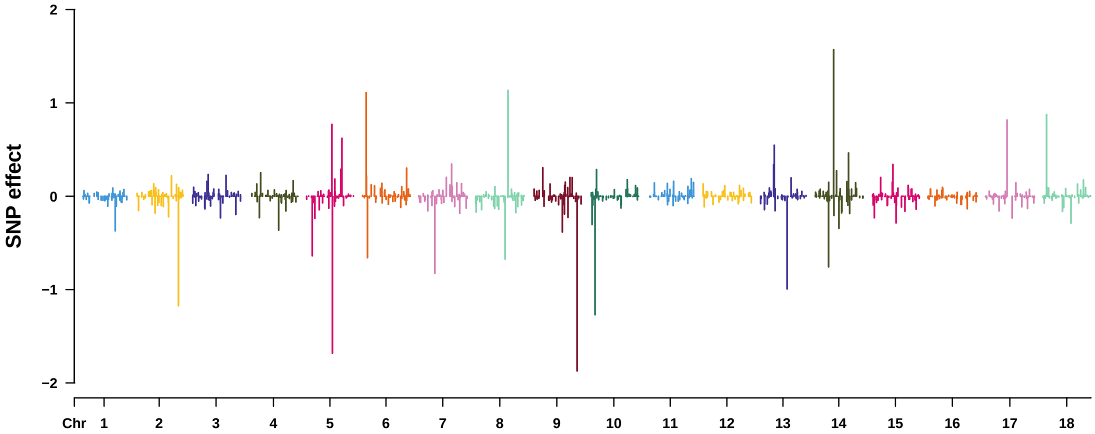

Figure 2: Estimated effects of the genomic markers across the entire genome. x-axis are the chromosomes, y-axis are the estimated marker effects, each vertical line represents a genomic marker.

Figure 1D (see the detailed commands in the replication code file), we can see the sampled values for both genetic variance and residual variance move around the red line, indicating a reasonable convergence of Markov chains. Users can change the total number of iterations and the number of burn-in iterations to obtain a converged chain according to the trace plots of estimated parameters.

The size of the marker effects can reflect the marker's contribution on the phenotype to a certain extent. We can visualize the marker effects using the **CMplot** package (Yin 2024) which has been included as dependency of package **hibayes**, as shown in the Manhattan plot in Figure 2:

```
R> CMplot(cbind(map[, 1:3], SNPeffect), type = "h", plot.type = "m",
+    LOG10 = FALSE, ylab = "SNP effect")
```

The bigger absolute value of a marker effect represents a higher contribution to the phenotype, suggesting the presence of a causal gene in its vicinity, and its positive or negative sign reflects an increase or decrease in the trait value.

The object of class 'blrMod' can be summarized by the `summary` function as follows:

```
R> sumfit <- summary(fitCpi)
R> sumfit

Individual level Bayesian model fit by [BayesCpi]
Formula: T1 ~ season + bwt + (1 | loc) + (1 | dam) + M

Residuals ($e):
    Min.  1st Qu.   Median  3rd Qu.     Max.
-9.10320 -2.19370  0.12939  2.19370  9.53260

Fixed effects ($beta):
```

```
             Estimate    SD
(Intercept)    35.163 7.242
seasonSpring  -21.911 1.503
seasonSummer  -11.544 1.465
seasonWinter  -11.433 1.567
bwt             2.378 0.814

Environmental random effects ($VER, $r):
         Variance    SD
loc         8.178 4.156
dam        54.414 9.779
Residual   30.447 6.674
Number of obs: 300, group: loc, 50; dam, 150

Genetic random effects ($VGR, $g):
   Estimate     SD
Vg  52.79860 12.335
h2   0.36061  0.073
pi1  0.92018  0.037
pi2  0.07982  0.037
Number of markers: 1000 , predicted individuals: 600

Marker effects ($alpha):
      Min.    1st Qu.     Median    3rd Qu.       Max.
-1.8716700 -0.0276056  0.0000000  0.0247178  1.5696500
```

As shown above, the summarized results describe the model formula and the method used in the analysis. The summary also contains the posterior estimates of some of the main model parameters, as well as their standard deviations. We classified the results into the following categories:

```
R> names(sumfit)
```

```
[1] "call"  "beta"  "VER"   "r"     "VGR"   "g"     "alpha" "e"
```

The list element `beta` is a data frame which contains the estimated fixed effects and regression coefficients. The list element `r` is a data frame of estimated environmental random effects, their variance components are stored in the list element `VER` (i.e., variance of environmental random effect); the list element `VGR` contains the variances of genetic random effects (`Vg`), the heritability of traits (`h2`), and the proportion of markers in different distributions (`pi`); the list element `g` is the data frame of genomic estimated breeding values (GEBVs) of both phenotypic and non-phenotypic individuals; `alpha` is the data frame of the estimated marker effects; and `e` is the data frame of model residuals. The names of the list returned by the `summary()` method are fully consistent with the mathematical symbols presented in Section 2. The detailed data structure of the summarized results can be viewed by `str(sumfit)`.

For GWAS analysis, package **hibayes** reports the WPPA and PIP for all markers as described in Section 2.4. If the arguments `windsize` or `windnum` are detected in the input commands, the list element named `gwas` in the returned object can be extracted as follows:

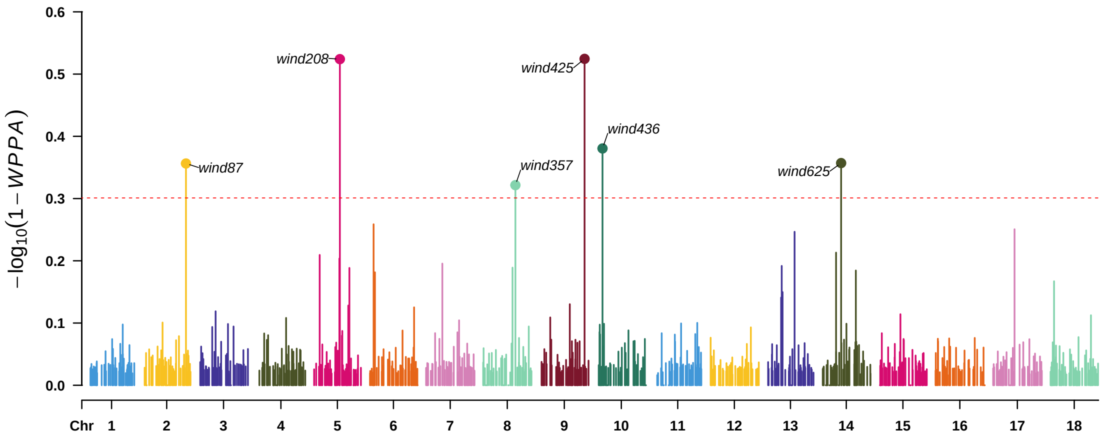

Figure 3: The derived window posterior probability of associations from MCMC iterations for all genomic markers. The x-axis are the chromosomes, the y-axis can be considered as the association significance, the red line is the significance level, and the labels around the points are the names of significantly associated markers at the given threshold.

```
R> gwas <- fitCpi[["gwas"]]
R> head(gwas, 4)
```

```
   Wind Chr N   Start     End   WPPA
1 wind1   1 1 4825340 4825340 0.0610
2 wind2   1 1 6371512 6371512 0.0647
3 wind3   1 1 7946983 7946983 0.0759
4 wind4   1 1 8945290 8945290 0.0510
```

The first column lists the names of all windows for the specified size, the second column is a vector of chromosomes, the third column reports the number of genomic markers included in each of the windows, the 4th and 5th columns are the physical positions of the first and last genomic marker of the windows, respectively. The last column is the computed WPPA, which can also be visualized in a Manhattan plot (Figure 3).

```
R> highlight <- gwas[(1 - gwas[, "WPPA"]) < 0.5, 1]
R> CMplot(data.frame(gwas[, c(1, 2, 4)], wppa = 1 - gwas[, "WPPA"]),
+    type = "h", plot.type = "m", LOG10 = TRUE, threshold = 0.5,
+    ylim = c(0, 0.6), ylab = expression(-log[10](1 - italic(WPPA))),
+    highlight = highlight, highlight.col = NULL,
+    highlight.text = highlight)
```

The bigger the y-axis in Figure 3, the stronger is the association of the window with the trait or disease, and thus causal genes are more likely within the window. However, it is still difficult to know which genomic markers in the windows of interest are the causal ones. So we need to further explore the association significance for the markers. In package **hibayes**, we report the posterior inclusive probability for every single marker, which could be used for a reference of importance to the trait and can be obtained by `fitCpi[["pip"]]`.

4.2. Examples for the summary level Bayesian model

To fit a summary level data based Bayesian model, the summary data and the variance-covariance matrix calculated from the reference panel should be provided. We have attached an example dataset in package **hibayes**. The summary data can be loaded using:

```
R> sumstat_path <- system.file("extdata", "demo.ma", package = "hibayes")
R> sumstat <- read.table(sumstat_path, header = TRUE)
R> head(sumstat, 4)

  SNP A1 A2    MAF    BETA    SE       P NMISS
1  M1  G  T 0.5267  0.1316 1.264 0.91710   300
2  M2  A  G 0.1458 -1.7920 1.685 0.28830   300
3  M3  G  A 0.3150  4.4080 1.828 0.01647   300
4  M4  C  G 0.5225 -1.1040 1.175 0.34840   300
```

As shown above, each row contains all the summary statistics for a marker. The first column is a column of marker names, the second and third columns are the reference and alternate allele, the remaining columns are the minor allele frequency (used to derive $\boldsymbol{D}$ in Model 14), the marginal effect (i.e., $\boldsymbol{b}$ in Model 14), the standard error, the p value of GWAS, and the effective sample size (i.e., n_j in Equation 16), respectively.

The LD variance-covariance matrix (i.e., $\boldsymbol{V}$ in Equation 16) can be calculated by package **hibayes** using either a public reference genotype panel or a personal genotype at hand. As discussed in Section 2.2, the $\boldsymbol{V}$ matrix is extremely huge if there are millions of genomic markers. To overcome this issue, we enable users to construct 4 types of different LD variance-covariance matrices including a genome-wide full dense matrix, a genome-wide sparse matrix, a chromosome-wide full dense matrix, and a chromosome-wide sparse matrix. These matrices have a different sparsity structure, and users can choose one of them according to the scale of data and the size of computational resources. Taking the tutorial data attached in package **hibayes**, one can proceed for example as follows:

```
R> bfile_path <- system.file("extdata", "demo", package = "hibayes")
R> bin <- read_plink(bfile_path)
R> geno <- bin[["geno"]]
R> map <- bin[["map"]]
R> ldm1 <- ldmat(geno, threads = 4)
R> ldm2 <- ldmat(geno, chisq = 5, threads = 4)
R> ldm3 <- ldmat(geno, map, ldchr = FALSE, threads = 4)
R> ldm4 <- ldmat(geno, map, ldchr = FALSE, chisq = 5, threads = 4)
```

The argument `chisq` is the chi-square threshold used for making the sparse matrix. Bigger `chisq` values generate a sparser LD matrix, but this may cause the Markov chain to “blow up” in certain situations as we discussed in Section 2.2. The argument `ldchr` is used to control whether to compute a genome-wide or chromosome-wide LD matrix. It should be noted that setting `ldmat(...., ldchr = TRUE)` could take an excessively long time and very big memory consumption to run if applied to a large dataset with many SNPs. We recommend turning it off for the majority of species because the interactions among chromosomes are generally negligible. The sparsity structure of different LD variance-covariance matrices can be visualized by the function `image()` in package **Matrix**, as demonstrated in the replication code file.

Before fitting the model, the prior adjustment regarding the order of genomic markers between summary data and LD matrix is required:

```
R> sumstat <- sumstat[match(map[, 1], sumstat[, 1]), ]
```

Then we can fit the summary level Bayesian model as follows:

```
R> fitCpi <- sbrm(sumstat = sumstat, ldm = ldm1, method = "BayesCpi",
+    Pi = c(0.95, 0.05), niter = 20000, nburn = 12000, seed = 666666,
+    map = map, windsize = 1e6)
```

Similar to the individual level Bayesian model, the detailed structure of the returned '`blrMod`' object can be viewed by `str(fitCpi)`, and it can also be summarized by `summary(fitCpi)`. Since the summary level Bayesian model does not require the individual level phenotype and genotype data, there are no lists regarding the fixed effects, model residuals, and the GEBVs in the summarized results. Only the genetic variance, residual variance, and the estimated marker effects are available:

```
R> names(summary(fitCpi))
```

```
[1] "call"   "VER" "VGR"  "alpha"
```

The above lists have the unified data structure across different types of Bayesian models, users can extract and visualize the parameters of interest for either genomic prediction or genome-wide association studies in the same way as shown in Section 4.1. Therefore we do not describe and illustrate more details about the results in this section.

4.3. Examples for single-step Bayesian model

To fit the single-step Bayesian model, at least the phenotype, the genotype, and the pedigree information should be provided. The formats of phenotype and genotype data used in the single-step Bayesian model are completely the same than for the individual level Bayesian model. We first load the phenotype and genotype data attached in package **hibayes** as follows:

```
R> pheno_file_path <- system.file("extdata", "demo.phe",
+    package = "hibayes")
R> pheno <- read.table(pheno_file_path, header = TRUE)
R> bfile_path <- system.file("extdata", "demo", package = "hibayes")
R> bin <- read_plink(bfile = bfile_path, mode = "A", threads = 4)
R> fam <- bin[["fam"]]
R> geno.id <- fam[, 2]
R> geno <- bin[["geno"]]
R> map <- bin[["map"]]
```

The data above has already been demonstrated and described in Section 4.1. We thus do not explain it again here. The only difference of the single-step Bayesian model compared to the individual level Bayesian model is the requirement of pedigree data, which is used to construct the additive genetic relationship matrix of all individuals (i.e., the A matrix in Model 18). The pedigree is a diagram that depicts the biological relationships between an organism and its ancestors. This data is typically stored in a text file, and we take the tutorial pedigree data attached in package **hibayes** as an example:

```
R> ped_file_path <- system.file("extdata", "demo.ped", package = "hibayes")
R> ped <- read.table(ped_file_path, header = TRUE)
R> head(ped, 4)
```

```
       id sir dam
1 IND0001   0   0
2 IND0002   0   0
3 IND0003   0   0
4 IND0004   0   0
```

As shown above, the first column contains the names of the individuals, and the remaining two columns show the names of their father and mother, respectively. Missing values in pedigree should be marked as "0" or "NA", and the columns must exactly follow the order of "id", "sir", and "dam".

After reading in all the data above successfully, we can now fit the single-step Bayesian model as follows:

```
R> fitR <- ssbrm(T1 ~ sex + bwt + (1 | dam), data = pheno, M = geno,
+    M.id = geno.id, pedigree = ped, method = "BayesR", niter = 20000,
+    nburn = 12000, printfreq = 100, Pi = c(0.95, 0.02, 0.02, 0.01),
+    fold = c(0, 0.0001, 0.001, 0.01), seed = 666666, map = map,
+    windsize = 1e6)
```

We can use `str(fitR)` to see the structure of the returned '`blrMod`' object and view the summarized results by `summary(fitR)`. All the genotyped and non-genotyped individuals in the pedigree will be predicted, therefore the total number of predicted individuals depends on the number of unique individuals in pedigree. The predicted GEBVs and the estimated marker effects could be accessed by `fitR$g` and `fitR$alpha`, respectively. The summarized information is closely similar to the individual level Bayesian model, except for the imputation regression coefficient term `J` and the imputation error variance `Veps`, see Model 18. Other returned parameters can be extracted and visualized in the same way as shown in Section 4.1.

5. Conclusion

The present paper is meant to provide a general overview on package **hibayes**, the only R package that can implement three types of Bayesian regression models with the richest methods achieved thus far. It is designed not only for genomic prediction, but also for genome-wide association studies. The package covers most of the functionalities involved in genetic evaluation, including estimation of fixed effects and coefficients of covariates, environmental random effects and the corresponding variance, genetic and residual variance, heritability of traits, and effects for all markers; computation of genomic estimated breeding values for both genotyped and non-genotyped individuals, phenotype/genetic variance explained for single or multiple markers; and the derivation of the posterior probability of association of the genomic window and posterior inclusive probability of markers. As shown in Table 1, package **hibayes** is more comprehensive compared to other tools for genomic relevant analyses.

The arguments of functions and the alias of returns in package **hibayes** are highly consistent with the mathematical equations presented in the main text. The functional style and idiomatic implementation in R make the package easy to use, flexible to extend, and transparent to validate. Although only a small selection of the modeling options available in package **hibayes** are discussed in detail, we hope that this article can serve as a good starting point to further explore the capabilities of the package. For the future, we have several plans on how to improve the functionality of package **hibayes**. We will keep on updating package **hibayes** with the latest advanced Bayesian models and methods of broad interest in the domain of genomic prediction or genome-wide association studies, ensuring that package **hibayes** always provides fresh functionality to the users or academic researchers. In addition to the MCMC sampling approach, another approximation based approach named "variational inference (VI)" is also commonly used to tackle Bayesian inference. The sampling process of MCMC is computationally pretty heavy but has no bias and, so, it is preferred when accurate results are expected, disregarding the time it takes for estimation. Conversely, although the choice of VI methods can clearly introduce a bias, this approach comes along with a reasonable optimization process that makes it particularly adapted to very large scale data requiring fast computations. Therefore, we will investigate the possibility of applying VI in statistical genetics and introduce it into package **hibayes** for efficient inference of unknown parameters using very big genomic data. Also, we will implement multiple traits Bayesian regression models in package **hibayes**, which can be used to estimate genetic correlation among traits and to further improve the prediction accuracy of traits and diseases.

Acknowledgments

Lilin Yin and Haohao Zhang contributed equally to this work.

First of all, we would like to thank Prof. Jian Zeng from the University of Queensland and Prof. Hao Cheng from the University of California, Davis, for their kind and generous assistance during our consultation regarding the summary-level and single-step Bayesian models in package **hibayes**. We thank Prof. Guosheng Su from Aarhus University for his professional corrections on the Bayesian theories in this paper. We thank Prof. Thomas A. Gavin from Cornell University for editing this paper. Furthermore, we also sincerely thank the many users who reported bugs or valuable ideas for new features, thus helping to continuously improvement for package **hibayes**. This work was supported by the National Natural Science Foundation of China [32102516, 32221005, 32072725]; Science Fund for Creative Research Groups of the Natural Science Foundation of Hubei Province [2020CFA006]; and the earmarked fund for CARS-35.

References

Aguilar I, Misztal I, Johnson DL, Legarra A, Tsuruta S, Lawlor TJ (2010). "Hot Topic: A Unified Approach to Utilize Phenotypic, Full Pedigree, and Genomic Information for Genetic Evaluation of Holstein Final Score." *Journal of Dairy Science*, **93**(2), 743–752. doi:10.3168/jds.2009-2730.

Aliloo H, Pryce JE, González-Recio O, Cocks BG, Goddard ME, Hayes BJ (2017). "Including Nonadditive Genetic Effects in Mating Programs to Maximize Dairy Farm Profitability." *Journal of Dairy Science*, **100**(2), 1203–1222. doi:10.3168/jds.2016-11261.

Anderson E, Bai Z, Bischof C, Blackford LS, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Sorensen D (1999). ***LAPACK** Users' Guide*. SIAM. doi:10.1137/1.9780898719604.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using **lme4**." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Bates D, Maechler M, Jagan M (2025). ***Matrix**: Sparse and Dense Matrix Classes and Methods*. doi:10.32614/CRAN.package.Matrix. R package version 1.7-3.

Bezanson J, Edelman A, Karpinski S, Shah VB (2017). "Julia: A Fresh Approach to Numerical Computing." *SIAM Review*, **59**(1), 65–98. doi:10.1137/141000671.

Bürkner PC (2017). "**brms**: An R Package for Bayesian Multilevel Models Using Stan." *Journal of Statistical Software*, **80**(1), 1–28. doi:10.18637/jss.v080.i01.

Cheng H, Fernando R, Garrick D (2018). "**JWAS**: Julia Implementation of Whole-Genome Analysis Software." In *Proceedings of the World Congress on Genetics Applied to Livestock Production*, volume 11, p. 859. URL https://reworkhow.github.io/JWAS.jl/latest/.

Christensen OF, Lund MS (2010). "Genomic Prediction When Some Animals Are Not Genotyped." *Genetics Selection Evolution*, **42**(1), 2. doi:10.1186/1297-9686-42-2.

Dagum L, Menon R (1998). "**OpenMP**: An Industry Standard API for Shared-Memory Programming." *IEEE Computational Science and Engineering*, **5**(1), 46–55. doi:10.1109/99.660313.

de los Campos G, Hickey JM, Pong-Wong R, Daetwyler HD, Calus MPL (2013). "Whole-Genome Regression and Prediction Methods Applied to Plant and Animal Breeding." *Genetics*, **193**(2), 327–345. doi:10.1534/genetics.112.143313.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

Eddelbuettel D, Sanderson C (2014). "**RcppArmadillo**: Accelerating R with High-Performance C++ Linear Algebra." *Computational Statistics & Data Analysis*, **71**, 1054–1063. doi:10.1016/j.csda.2013.02.005.

Erbe M, Hayes BJ, Matukumalli LK, Goswami S, Bowman PJ, Reich CM, Mason BA, Goddard ME (2012). "Improving Accuracy of Genomic Predictions Within and Between Dairy Cattle Breeds with Imputed High-Density Single Nucleotide Polymorphism Panels." *Journal of Dairy Science*, **95**(7), 4114–4129. doi:10.3168/jds.2011-5019.

Fernando R, Toosi A, Wolc A, Garrick D, Dekkers J (2017). "Application of Whole-Genome Prediction Methods for Genome-Wide Association Studies: A Bayesian Approach." *Journal of Agricultural, Biological and Environmental Statistics*, **22**(2), 172–193. doi:10.1007/s13253-017-0277-6.

Fernando RL, Cheng H, Golden BL, Garrick DJ (2016). "Computational Strategies for Alternative Single-Step Bayesian Regression Models with Large Numbers of Genotyped and Non-Genotyped Animals." *Genetics Selection Evolution*, **48**(1), 96. doi:10.1186/s12711-016-0273-2.

Fernando RL, Dekkers JCM, Garrick DJ (2014). "A Class of Bayesian Methods to Combine Large Numbers of Genotyped and Non-Genotyped Animals for Whole-Genome Analyses." *Genetics Selection Evolution*, **46**(1), 50. doi:10.1186/1297-9686-46-50.

Fernando RL, Garrick D (2013). "Bayesian Methods Applied to GWAS." In C Gondro, J Van der Werf, B Hayes (eds.), *Genome-Wide Association Studies and Genomic Prediction*, pp. 237–274. Humana Press. doi:10.1007/978-1-62703-447-0_10.

García-Cortés LA, Sorensen D (1996). "On a Multivariate Implementation of the Gibbs Sampler." *Genetics Selection Evolution*, **28**(1), 121. doi:10.1186/1297-9686-28-1-121.

Gianola D (2013). "Priors in Whole-Genome Regression: The Bayesian Alphabet Returns." *Genetics*, **194**(3), 573–596. doi:10.1534/genetics.113.151753.

Habier D, Fernando RL, Kizilkaya K, Garrick DJ (2011). "Extension of the Bayesian Alphabet for Genomic Selection." *BMC Bioinformatics*, **12**(1), 186. doi:10.1186/1471-2105-12-186.

Henderson CR (1975). "Best Linear Unbiased Estimation and Prediction under a Selection Model." *Biometrics*, **31**(2), 423–447. doi:10.2307/2529430.

Henderson CR (1976). "A Simple Method for Computing the Inverse of a Numerator Relationship Matrix Used in Prediction of Breeding Values." *Biometrics*, **32**(1), 69–83. doi:10.2307/2529339.

Kane MJ, Emerson J, Weston S (2013). "Scalable Strategies for Computing with Massive Data." *Journal of Statistical Software*, **55**(14), 1–19. doi:10.18637/jss.v055.i14.

Lloyd-Jones LR, Zeng J, Sidorenko J, Yengo L, Moser G, Kemper KE, Wang H, Zheng Z, Magi R, Esko T, Metspalu A, Wray NR, Goddard ME, Yang J, Visscher PM (2019). "Improved Polygenic Prediction by Bayesian Multiple Regression on Summary Statistics." *Nature Communications*, **10**(1), 5086. doi:10.1038/s41467-019-12653-0.

Lund MS, Jensen CS (1999). "Blocking Gibbs Sampling in the Mixed Inheritance Model Using Graph Theory." *Genetics Selection Evolution*, **31**(1), 3–24. doi:10.1186/1297-9686-31-1-3.

Meher PK, Rustgi S, Kumar A (2022). "Performance of Bayesian and BLUP Alphabets for Genomic Prediction: Analysis, Comparison and Results." *Heredity*, **128**(6), 519–530. doi:10.1038/s41437-022-00539-9.

Meuwissen THE, Hayes BJ, Goddard ME (2001). "Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps." *Genetics*, **157**(4), 1819–1829. doi:10.1093/genetics/157.4.1819.

Misztal I, Tsuruta S, Strabel T, Auvray B, Druet T, Lee D, *et al.* (2002). "**BLUPF90** and Related Programs (**BGF90**)." In *Proceedings of the 7th World Congress on Genetics Applied to Livestock Production*, volume 28. Montpellier.

Moser G, Lee SH, Hayes BJ, Goddard ME, Wray NR, Visscher PM (2015). "Simultaneous Discovery, Estimation and Prediction Analysis of Complex Traits Using a Bayesian Mixture Model." *PLOS Genetics*, **11**(4), e1004969. doi:10.1371/journal.pgen.1004969.

Ozaki K, Ohnishi Y, Iida A, Sekine A, Yamada R, Tsunoda T, Sato H, Sato H, Hori M, Nakamura Y, Tanaka T (2002). "Functional SNPs in the Lymphotoxin-α Gene That Are Associated with Susceptibility to Myocardial Infarction." *Nature Genetics*, **32**(4), 650–654. doi:10.1038/ng1047.

Pérez P, de los Campos G (2014). "Genome-Wide Regression and Prediction with the **BGLR** Statistical Package." *Genetics*, **198**(2), 483–495. doi:10.1534/genetics.114.164442.

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, De Bakker PIW, Daly MJ, Sham PC (2007). "**PLINK**: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses." *The American Journal of Human Genetics*, **81**(3), 559–575. doi:10.1086/519795.

R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. doi:10.32614/R.manuals. URL https://www.R-project.org/.

Sorensen D, Gianola D (2002). *Likelihood, Bayesian, and MCMC Methods in Quantitative Genetics*. Springer-Verlag. doi:10.1007/b98952.

The 1000 Genomes Project Consortium (2015). "A Global Reference for Human Genetic Variation." *Nature*, **526**(7571), 68–74. doi:10.1038/nature15393.

VanRaden PM (2008). "Efficient Methods to Compute Genomic Predictions." *Journal of Dairy Science*, **91**(11), 4414–4423. doi:10.3168/jds.2007-0980.

Wang Y, Tsuo K, Kanai M, Neale BM, Martin AR (2022). "Challenges and Opportunities for Developing More Generalizable Polygenic Risk Scores." *Annual Review of Biomedical Data Science*, **5**(1), 293–320. doi:10.1146/annurev-biodatasci-111721-074830.

Wray NR, Wijmenga C, Sullivan PF, Yang J, Visscher PM (2018). "Common Disease Is More Complex Than Implied by the Core Gene Omnigenic Model." *Cell*, **173**(7), 1573–1580. doi:10.1016/j.cell.2018.05.051.

Yang J, Ferreira T, Morris AP, Medland SE, Madden PAF, Heath AC, Martin NG, Montgomery GW, Weedon MN, Loos RJ, Frayling TM, McCarthy MI, Hirschhorn JN, Goddard ME, Visscher PM (2012). "Conditional and Joint Multiple-SNP Analysis of GWAS Summary Statistics Identifies Additional Variants Influencing Complex Traits." *Nature Genetics*, **44**(4), 369–375. doi:10.1038/ng.2213.

Yang J, Lee SH, Goddard ME, Visscher PM (2011). "**GCTA**: A Tool for Genome-Wide Complex Trait Analysis." *The American Journal of Human Genetics*, **88**(1), 76–82. doi:10.1016/j.ajhg.2010.11.011.

Yi N, Xu S (2008). "Bayesian LASSO for Quantitative Trait Loci Mapping." *Genetics*, **179**(2), 1045–1055. doi:10.1534/genetics.107.085589.

Yin L (2024). ***CMplot**: Circle Manhattan Plot*. doi:10.32614/CRAN.package.CMplot. R package version 4.5.1.

Yin L, Zhang H, Liu X (2025). ***hibayes**: Individual-Level, Summary-Level and Single-Step Bayesian Regression Model*. doi:10.32614/CRAN.package.hibayes. R package version 3.1.0.

Yin L, Zhang H, Tang Z, Xu J, Yin D, Zhang Z, Yuan X, Zhu M, Zhao S, Li X, Liu X (2021). "**rMVP**: A Memory-Efficient, Visualization-Enhanced, and Parallel-Accelerated Tool for Genome-Wide Association Study." *Genomics, Proteomics & Bioinformatics*, **19**(4), 619–628. doi:10.1016/j.gpb.2020.10.007.

Yin L, Zhang H, Tang Z, Yin D, Fu Y, Yuan X, Li X, Liu X, Zhao S (2023). "**HIBLUP**: An Integration of Statistical Models on The BLUP Framework for Efficient Genetic Evaluation Using Big Genomic Data." *Nucleic Acids Research*, **51**(8), 3501–3512. doi:10.1093/nar/gkad074.

Yin L, Zhang H, Zhou X, Yuan X, Zhao S, Li X, Liu X (2020). "**KAML**: Improving Genomic Prediction Accuracy of Complex Traits Using Machine Learning Determined Parameters." *Genome Biology*, **21**(1), 146. doi:10.1186/s13059-020-02052-w.

Zeng J, De Vlaming R, Wu Y, Robinson MR, Lloyd-Jones LR, Yengo L, Yap CX, Xue A, Sidorenko J, McRae AF, Powell JE, Montgomery GW, Metspalu A, Esko T, Gibson G, Wray NR, Visscher PM, Yang J (2018). "Signatures of Negative Selection in the Genetic Architecture of Human Complex Traits." *Nature Genetics*, **50**(5), 746–753. doi:10.1038/s41588-018-0101-4.

Zhou X, Carbonetto P, Stephens M (2013). "Polygenic Modeling with Bayesian Sparse Linear Mixed Models." *PLoS Genetics*, **9**(2), e1003264. doi:10.1371/journal.pgen.1003264.

Zhu X, Stephens M (2017). "Bayesian Large-Scale Multiple Regression with Summary Statistics from Genome-Wide Association Studies." *The Annals of Applied Statistics*, **11**(3), 1561–1592. doi:10.1214/17-aoas1046.

A. The implementation of an efficient sampling strategy

Given the model presented in the main text:

$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{X\beta} + \boldsymbol{Rr} + \boldsymbol{M\alpha} + \boldsymbol{e} \tag{25}$$

the full conditional distribution of a parameter used in the Gibbs sampler is constructed by using the most recent estimates of the other parameters. For example, when sampling the jth marker effect at the ith iteration ($\alpha_j^{[i]}$), it is required to adjust the phenotype observations $\boldsymbol{y}$ for all other effects in the model. Let $\widetilde{\boldsymbol{y}}$ be the vector of adjusted phenotype values, given by

$$\widetilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{\mu}^{[i]} - \boldsymbol{X}\boldsymbol{\beta}^{[i]} - \boldsymbol{R}\boldsymbol{r}^{[i]} - \sum_{k \neq j}^{m} \boldsymbol{M}_k \alpha_k, \tag{26}$$

where $\boldsymbol{\mu}^{[i]}$, $\boldsymbol{\beta}^{[i]}$, and $\boldsymbol{r}^{[i]}$ are the most recent estimates at the ith iteration, m is the total number of markers (i.e., the second dimension of $\boldsymbol{M}$), and $\boldsymbol{M}_k$ is the kth column of $\boldsymbol{M}$.

Since the matrix $\boldsymbol{M}$ is large, it would be extremely time-expensive to repetitively compute $\widetilde{\boldsymbol{y}}$ for each unknown parameter in every Markov chain using Equation 26. Another ingenious and efficient strategy is to obtain it based on the current model residuals ($\boldsymbol{y}^*$) and the sampled value of the parameter at the previous iteration ($\alpha_j^{[i-1]}$), i.e.,

$$\widetilde{\boldsymbol{y}} = \boldsymbol{y}^* + \boldsymbol{M}_j \alpha_j^{[i-1]}. \tag{27}$$

$\boldsymbol{y}^*$ is computed and stored at the beginning of the MCMC sampling, but updated immediately once any of the model effects ($\boldsymbol{\mu}$, $\boldsymbol{\beta}$, $\boldsymbol{r}$, $\boldsymbol{\alpha}$) is sampled. E.g., when we obtain the new marker effect $\alpha_j^{[i]}$ from its full conditional distribution, the corresponding model residual $\boldsymbol{y}^*$ can be updated by

$$\boldsymbol{y}^* = \boldsymbol{y}^* + \boldsymbol{M}_j(\alpha_j^{[i-1]} - \alpha_j^{[i]}). \tag{28}$$

The real-time $\boldsymbol{y}^*$ can link the full conditional distribution of one parameter with the latest estimates of other parameters without the needs of big matrix operations. Thus this approach should be more computationally beneficial for the Gibbs sampler.

B. Full conditional distribution of fixed effects

The fixed effects and regression coefficients $\boldsymbol{\beta}$ are assumed to have flat priors (Sorensen and Gianola 2002), i.e., improper uniform priors such that the density function is a constant, denoted by $f(\boldsymbol{\beta}) \propto c$. In contrast to Appendix A, here, we do not use in a similar way Equation 26 to obtain the adjusted phenotype for β_j, but instead we use Equation 27 to construct the full conditional distribution by means of ordinary least squares (OLS) as follows:

$$(\beta_j^{[i]} \mid \boldsymbol{y}^*, \boldsymbol{X}, \beta_j^{[i-1]}, \sigma_e^2) \sim \mathcal{N}\left(\frac{\boldsymbol{X}_j^\top \boldsymbol{y}^* + \boldsymbol{X}_j^\top \boldsymbol{X}_j \beta_j^{[i-1]}}{\boldsymbol{X}_j^\top \boldsymbol{X}_j}, \frac{\sigma_e^2}{\boldsymbol{X}_j^\top \boldsymbol{X}_j}\right), \tag{29}$$

where $\boldsymbol{y}^*$ is the real-time model residuals which can be obtained on the conditions of $\boldsymbol{y}$ and the sampled $\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\alpha}$. $\beta_j^{[i]}$ is the sampled value of the jth variable in $\boldsymbol{\beta}$ at the ith iteration, and $\beta_j^{[i-1]}$ is the sampled value of the previous iteration. $\boldsymbol{X}_j$ represents the jth column of the model matrix $\boldsymbol{X}$.

The corresponding model residuals for $\beta_j^{[i]}$ are updated in a similar way than in Equation 28 using

$$\boldsymbol{y}^* = \boldsymbol{y}^* + \boldsymbol{X}_j(\beta_j^{[i-1]} - \beta_j^{[i]}). \tag{30}$$

By repeating Equation 29 and updating the model residuals $\boldsymbol{y}^*$ subsequently, we can obtain the new estimates for all the elements in $\boldsymbol{\beta}$ at the current iteration.

C. Full cond. distribution of environmental random effects

The environmental random effects are a priori assumed to follow a normal distribution $\mathcal{N}(0, \boldsymbol{I}\sigma_r^2)$. The conditional solution for the environmental random effects $\boldsymbol{r}$ can be written as

$$\left(\boldsymbol{R}^\top\boldsymbol{R} + \boldsymbol{I}\frac{\sigma_e^2}{\sigma_r^2}\right)\boldsymbol{r}^{[i]} = \boldsymbol{R}^\top(\boldsymbol{y}^* + \boldsymbol{R}\boldsymbol{r}^{[i-1]}), \tag{31}$$

where $\boldsymbol{r}^{[i]}$ and $\boldsymbol{r}^{[i-1]}$ represent the sampled $\boldsymbol{r}$ at the current and previous iteration, respectively. To get the conditional estimates $\boldsymbol{r}^{[i]}$, two types of sampling algorithms can be taken into consideration: the first is the single-site Gibbs sampler (Sorensen and Gianola 2002), which draws the parameter from its full conditional distribution with density $\boldsymbol{p}(r_k|\boldsymbol{r}_{-k}), k = 1, \ldots, n_r$, where n_r is the number of elements in the environmental random effect $\boldsymbol{r}$. Setting $\boldsymbol{C} = \left(\boldsymbol{R}^\top\boldsymbol{R} + \boldsymbol{I}\frac{\sigma_e^2}{\sigma_r^2}\right)$ and $\boldsymbol{B} = \boldsymbol{R}^\top(\boldsymbol{y}^* + \boldsymbol{R}\boldsymbol{r}^{[i-1]})$, we simplify Equation 31 to $\boldsymbol{C}\boldsymbol{r}^{[i]} = \boldsymbol{B}$. The single-site Gibbs sampler strategy can then be carried out as follows:

$$(r_j^{[i]} \mid \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{r}_{-j}, \sigma_e^2) \sim \mathcal{N}\left(\frac{B_j - \sum\limits_{k\neq j}^{n_r} C_{kj}r_k}{C_{jj}}, \frac{\sigma_e^2}{C_{jj}}\right), \tag{32}$$

where B_j is the jth element of vector $\boldsymbol{B}$, and C_{jj} is the jth diagonal element of matrix $\boldsymbol{C}$. Obviously this cannot be processed in parallel, but the big advantage is that there is no need to compute the inverse of matrix $\boldsymbol{C}$.

The second algorithm is the block Gibbs sampler (García-Cortés and Sorensen 1996; Lund and Jensen 1999), which draws unknown parameters jointly from a full conditional multivariate distribution formulated as

$$(\boldsymbol{r}^{[i]} \mid \boldsymbol{B}, \boldsymbol{C}, \sigma_e^2) \sim \mathcal{N}\left(\boldsymbol{C}^{-1}\boldsymbol{B}, \boldsymbol{C}^{-1}\sigma_e^2\right). \tag{33}$$

The above implementation strategy is straightforward, but it requires to compute the inverse of matrix $\boldsymbol{C}$, therefore, is typically used to handle sparse systems for fast computing. After the environmental random effects are sampled successfully, then the corresponding model residuals could be updated similarly as in Equation 28 by

$$\boldsymbol{y}^* = \boldsymbol{y}^* + \boldsymbol{R}(\boldsymbol{r}^{[i-1]} - \boldsymbol{r}^{[i]}). \tag{34}$$

The variance σ_r^2 is a priori assumed to have a scaled inverse chi-square distribution, which can be sampled from the following full conditional distribution,

$$(\sigma_r^2 \mid \boldsymbol{r}) \sim \left(\boldsymbol{r}^\top\boldsymbol{r} + \nu_r S_r^2\right)\chi^{-2}_{\nu_r+n_r}, \tag{35}$$

where ν_r and S_r^2 are the degrees of freedom and the scale factor of the scaled inverse chi-square distribution, respectively. In package **hibayes**, we no longer update ν_r and S_r^2 in MCMC, but use pre-defined constants, which are $\nu_r = -1$ and $S_r^2 = 0$ by default.

D. The BayesR method

The prior assumption on the marker effects for the BayesR method is a mixture of four normal distributions (Moser *et al.* 2015), of which the variances are assigned with a user defined scale on a temporary base variance $\sigma^2_{\alpha_\gamma}$. The scale factor is $\boldsymbol{\gamma} = (0, 0.0001, 0.001, 0.01)$. Thus it is not required to draw the variances for different normal distributions, but only sample the variance $\sigma^2_{\alpha_\gamma}$ once at each MCMC iteration. Similar to the Equation 6 in main text, the full conditional distribution of non-zero marker effects of the BayesR method can be formulated as

$$(\alpha_j^{[i]} \mid \boldsymbol{y}^*, \boldsymbol{M}, \boldsymbol{\gamma}, \alpha_j^{[i-1]}, \sigma^2_\gamma, \sigma^2_e) \sim \mathcal{N}\left(\frac{\boldsymbol{M}_j^\top \boldsymbol{y}^* + \boldsymbol{M}_j^\top \boldsymbol{M}_j \alpha_j^{[i-1]}}{\boldsymbol{M}_j^\top \boldsymbol{M}_j + \sigma^2_e/(\gamma_j \sigma^2_{\alpha_\gamma})}, \frac{\sigma^2_e}{\boldsymbol{M}_j^\top \boldsymbol{M}_j + \sigma^2_e/(\gamma_j \sigma^2_{\alpha_\gamma})}\right) \tag{36}$$

where $\boldsymbol{y}^*$ is the real-time model residuals which can be obtained on the conditions of $\boldsymbol{y}$ and the sampled $\boldsymbol{\mu}, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\alpha}$. γ_j is the corresponding scaled value for the jth genomic marker. The log likelihood calculation of different distributions for BayesR is similar to Equation 7 in the main text and can be written as

$$L_{\pi_k} = -\frac{1}{2}\left[\log\left(\frac{\boldsymbol{M}_j^\top \boldsymbol{M}_j \gamma_k \sigma^2_{\alpha_\gamma}}{\sigma^2_e} + 1\right) - \frac{\left(\boldsymbol{M}_j^\top \boldsymbol{y}^* + \boldsymbol{M}_j^\top \boldsymbol{M}_j \alpha_j\right)^2}{(\boldsymbol{M}_j^\top \boldsymbol{M}_j + \sigma^2_e/(\gamma_k \sigma^2_{\alpha_\gamma}))\sigma^2_e}\right] + \log(\pi_k), \tag{37}$$

where γ_k is the kth element of $\boldsymbol{\gamma}$ with $k \in \{2, \ldots, n_\gamma\}$. Using the same derivation as in Equation 8 in the main text, we can calculate the probability that marker j is in the kth distribution.

The variance $\sigma^2_{\alpha_\gamma}$ is a priori assumed to follow a scaled inverse chi-square distribution. Its full conditional distribution is

$$(\sigma^2_{\alpha_\gamma} \mid \boldsymbol{\alpha}, \boldsymbol{\gamma}, m_{\pi_0}) \sim \left(\sum_{j=1}^{m_\gamma} \frac{(\alpha_j)^2}{\gamma_j} + \nu_\alpha S^2_\alpha\right) \chi^{-2}_{\nu_\alpha + m_\gamma}, \tag{38}$$

where $m_\gamma = m - m_{\pi_0}$, and γ_j is the corresponding scaled value for the jth genomic marker, ν_α and S^2_α are the pre-defined degrees of freedom and the scale factor of the scaled inverse chi-square distribution.

E. The BSLMM method

The model formula of the BSLMM method (Zhou *et al.* 2013) can be expressed as

$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{X\beta} + \boldsymbol{Rr} + \boldsymbol{Zg} + \boldsymbol{M\alpha} + \boldsymbol{e}. \tag{39}$$

Compared to Model 2 in the main text, the model of the BSLMM method has an additional term $\boldsymbol{Zg}$, where $\boldsymbol{Z}$ is the design matrix and $\boldsymbol{g}$ is a vector of polygenic genetic random effects, which follows a multivariate normal distribution $\mathcal{N}(0, \boldsymbol{K}\sigma_k^2)$. $\boldsymbol{K}$ is the additive genomic relationship matrix (GRM) that is derived from genotype information and $\boldsymbol{\alpha}$ is the vector of additional genetic effects which cannot be captured by $\boldsymbol{g}$ for a small proportion of genomic markers. It has the same prior distribution than BayesCpi, i.e., the markers with effects share the same variance of the normal distribution. There have been several algorithms proposed for GRM construction. We implemented the most commonly used one in package **hibayes** (VanRaden 2008). The corresponding mathematical formula is

$$\boldsymbol{K} = \frac{\boldsymbol{M}\boldsymbol{M}^\top}{\sum\limits_{j=1}^{m} 2p_j\left(1 - p_j\right)}, \tag{40}$$

where m is the number of genomic markers, p_k is the frequency of allele A_1 at the jth genomic marker, $\boldsymbol{M}$ is the centered additive marker covariate matrix with elements $2 - 2p_j$, $1 - 2p_j$, and $-2p_j$ for genotype A_1A_1, A_1A_2, and A_2A_2, respectively.

The BSLMM method can be considered as an extension of the BayesCpi method with an additional multivariate normal random term $\boldsymbol{g}$. Similar to Equation 31 in Appendix C, the conditional solution of $\boldsymbol{g}$ is given by the following system:

$$\left(\boldsymbol{Z}^\top\boldsymbol{Z} + \boldsymbol{K}^{-1}\frac{\sigma_e^2}{\sigma_k^2}\right)\boldsymbol{g}^{[i]} = \boldsymbol{Z}^\top(\boldsymbol{y}^* + \boldsymbol{Z}\boldsymbol{g}^{[i-1]}). \tag{41}$$

where $\boldsymbol{y}^*$ is the real-time model residuals. To construct the equation above, it is required to compute the inverse of matrix $\boldsymbol{K}$. However, the GRM matrix $\boldsymbol{K}$ is not always invertible for some reasons (e.g., not positive definite). Although it can be addressed by various types of algorithms, e.g., the Cholesky decomposition, lower-upper (LU) decomposition, and ridge regression, computing the inverse of a big matrix is time-expensive. To obtain the solution $\boldsymbol{g}^{[i]}$, another challenge is to calculate the inverse of the left-hand-side (LHS) of Equation 41 for every single MCMC iteration. Although the single-site Gibbs sampler described in Equation 32 can overcome this issue, it is still time-consuming for a very large sample size. In package **hibayes**, we implemented an efficient eigen decomposition based block Gibbs sampler to get the solution $\boldsymbol{g}^{[i]}$ without computing the inverse of either $\boldsymbol{K}$ or the LHS matrix.

The inverse of $\boldsymbol{K}$ can be calculated by eigen decomposition as follows:

$$\boldsymbol{K}^{-1} = (\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{-1})^{-1} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^{-1}. \tag{42}$$

Since $\boldsymbol{K}$ is symmetric, the eigen vectors $\boldsymbol{U}$ are guaranteed to be an orthogonal matrix. Therefore $\boldsymbol{U}^{-1} = \boldsymbol{U}^\top$. Furthermore, because $\boldsymbol{D}$ is a diagonal matrix, its inverse is easy to calculate. Then we have

$$\boldsymbol{K}^{-1} = \boldsymbol{U}\begin{bmatrix} \frac{1}{d_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{d_n} \end{bmatrix}\boldsymbol{U}^\top. \tag{43}$$

Since we only fit effective records in the model, $\boldsymbol{Z}$ is equal to identity matrix, then $\boldsymbol{Z}^\top\boldsymbol{Z} =$

$\boldsymbol{I} = \boldsymbol{U}\boldsymbol{U}^\top$. Let $\lambda = \frac{\sigma_e^2}{\sigma_k^2}$. The inverse of the LHS in Equation 41 can then be written as

$$\begin{aligned}\left(\boldsymbol{Z}^\top\boldsymbol{Z} + \boldsymbol{K}^{-1}\lambda\right)^{-1} &= \left(\boldsymbol{U}\boldsymbol{U}^\top + \boldsymbol{U}\frac{\boldsymbol{I}\lambda}{\boldsymbol{D}}\boldsymbol{U}^\top\right)^{-1} \\ &= \left(\boldsymbol{U}\frac{\boldsymbol{D}+\boldsymbol{I}\lambda}{\boldsymbol{D}}\boldsymbol{U}^\top\right)^{-1} \\ &= \boldsymbol{U}\frac{\boldsymbol{D}}{\boldsymbol{D}+\boldsymbol{I}\lambda}\boldsymbol{U}^\top.\end{aligned}$$

As shown above, by exploiting the results of the eigen decomposition of the $\boldsymbol{K}$ matrix, it is computationally efficient to construct the full conditional multivariate distribution of $\boldsymbol{g}^{[i]}$ without computing the inverse of any big matrix. To implement the Gibbs sampler for multivariate distributions formulated similarly as in Equation 33, both the Cholesky decomposition and the eigen decomposition can be employed for sampling. Given the multivariate distribution $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, in which $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and variance-covariance matrix, respectively, we can generate new samples based on the Cholesky decomposition as

$$\boldsymbol{X} \sim \boldsymbol{\mu} + \boldsymbol{L}\mathcal{N}(0, \boldsymbol{I}), \tag{44}$$

where $\boldsymbol{L}$ is the lower triangular matrix from the Cholesky decomposition of $\boldsymbol{\Sigma}$, and $\boldsymbol{I}$ is an identity matrix; Alternatively the eigen decomposition can be used resulting in

$$\boldsymbol{X} \sim \boldsymbol{\mu} + \boldsymbol{U}\sqrt{\boldsymbol{D}}\mathcal{N}(0, \boldsymbol{I}), \tag{45}$$

where $\boldsymbol{U}$ and $\boldsymbol{D}$ are the eigen vectors and eigen values of $\boldsymbol{\Sigma}$, respectively. Furthermore, the variance-covariance matrix of $\boldsymbol{g}^{[i]}$ (i.e., $\left(\boldsymbol{Z}^\top\boldsymbol{Z} + \boldsymbol{K}^{-1}\lambda\right)^{-1}\sigma_e^2$) and the GRM matrix $\boldsymbol{K}$ share the same eigen vectors; thus, it is not required to do eigen decomposition on the variance-covariance matrix. Then the full conditional multivariate normal distribution of $\boldsymbol{g}^{[i]}$ can be expressed as

$$(\boldsymbol{g}^{[i]} \mid \boldsymbol{y}^*, \boldsymbol{U}, \boldsymbol{D}, \boldsymbol{g}^{[i-1]}, \lambda, \sigma_e^2) \sim \boldsymbol{U}\frac{\boldsymbol{D}}{\boldsymbol{D}+\boldsymbol{I}\lambda}\boldsymbol{U}^\top(\boldsymbol{y}^* + \boldsymbol{g}^{[i-1]}) + \boldsymbol{U}\sqrt{\frac{\boldsymbol{D}\sigma_e^2}{\boldsymbol{D}+\boldsymbol{I}\lambda}}\mathcal{N}(0, \boldsymbol{I}). \tag{46}$$

where $\boldsymbol{y}^*$ is the real-time model residuals. As shown in the equation above, it only requires to do eigen decomposition on the $\boldsymbol{K}$ matrix once. No inverse computation of a big matrix is involved in the whole block Gibbs sampler, making it very efficient in MCMC iterations.

The variance σ_k^2 follows a scaled inverse chi-square distribution, with degrees of freedom and scale parameter in the full conditional distribution:

$$(\sigma_k^2 \mid \boldsymbol{g}, \boldsymbol{U}, \boldsymbol{D}) \sim \left(\boldsymbol{g}^\top\boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^\top\boldsymbol{g} + \nu_k S_k^2\right)\chi^{-2}_{\nu_k+n_k}, \tag{47}$$

where ν_k and S_k^2 are the pre-defined degrees of freedom and the scale factor of the scaled inverse chi-square distribution.

It should be pointed out that the ultimate marker effect of the BSLMM method includes two parts

$$\boldsymbol{\alpha}^* = \frac{\boldsymbol{M}^\top\boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^\top\boldsymbol{g}}{\sum\limits_{j=1}^{m} 2p_j\left(1-p_j\right)} + \boldsymbol{\alpha},$$

where the first part transforms the genetic random effect $\boldsymbol{g}$ into marker effects (Aliloo, Pryce, González-Recio, Cocks, Goddard, and Hayes 2017). To multiply the two big matrices (i.e., $\boldsymbol{M}$ and $\boldsymbol{U}$) in the above equation is extremely time-consuming. But this step is implemented only once at the end of the MCMC iteration and, therefore, does not constitute a big problem.

F. Inference of the summary level Bayesian model

The summary level Bayesian model is inferred from the individual level model. Let

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{e}. \tag{48}$$

Different from Model 2 in the main text, here $\boldsymbol{y}$ is the prior-adjusted phenotype, generally defined as the residuals extracted from a linear model that includes recorded environmental fixed effects, covariates, and random effects. The reason for the adjustment is that the public summary data only includes summary information for genomic markers, and we cannot directly access the recorded environmental factors.

Let $\boldsymbol{D} = \operatorname{diag}\left(\boldsymbol{M}_1^\top \boldsymbol{M}_1, \dots, \boldsymbol{M}_m^\top \boldsymbol{M}_m\right)$. By multiplying Equation 48 with $\boldsymbol{D}^{-1}\boldsymbol{M}^\top$ we arrive at

$$\boldsymbol{D}^{-1}\boldsymbol{M}^\top\boldsymbol{y} = \boldsymbol{D}^{-1}\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{D}^{-1}\boldsymbol{M}^\top\boldsymbol{e}. \tag{49}$$

The left hand side $\boldsymbol{D}^{-1}\boldsymbol{M}^\top\boldsymbol{y}$ corresponds to the estimated coefficients of the single marker regression, that is the marginal effect $\boldsymbol{b}$ in the summary data. As is well known, the correlation matrix $\boldsymbol{B}$ of markers can be expressed as follows:

$$\boldsymbol{B} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{D}^{-\frac{1}{2}}. \tag{50}$$

Let $\boldsymbol{e}^* = \boldsymbol{D}^{-1}\boldsymbol{M}^\top\boldsymbol{e}$. Then we can rewrite Equation 49 as

$$\boldsymbol{b} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{\alpha} + \boldsymbol{e}^*. \tag{51}$$

The sampling details of $\boldsymbol{\alpha}$ and its variance have been given in the main text. Since the model residuals $\boldsymbol{y^*}$ are not available, we thus cannot construct the full condition of σ_e^2 by $\boldsymbol{y^*}$. The residual variance σ_e^2 can be expressed as

$$\begin{aligned} \sigma_e^2 &= \frac{\boldsymbol{y}^{*\top}\boldsymbol{y}^*}{n} \\ &= \frac{(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\alpha})^\top(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\alpha})}{n} \\ &= \frac{\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{\alpha}^\top\boldsymbol{M}^\top\boldsymbol{y} + \boldsymbol{\alpha}^\top\boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{\alpha}}{n} \end{aligned} \tag{52}$$

and the phenotypic variance of the trait by $\sigma_y^2 = \boldsymbol{y}^\top\boldsymbol{y}/n$. Multiplying Model 48 with $\boldsymbol{M}^\top$, we have

$$\begin{aligned} \boldsymbol{M}^\top\boldsymbol{M}\boldsymbol{\alpha} &= \boldsymbol{M}^\top\boldsymbol{y} - \boldsymbol{M}^\top\boldsymbol{y}^* \\ &= \boldsymbol{D}\boldsymbol{b} - \boldsymbol{\phi} \end{aligned} \tag{53}$$

where $\boldsymbol{\phi}$ has been defined in main text. Now the residual variance σ_e^2 can be written as

$$\begin{aligned} \sigma_e^2 &= \frac{\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{\alpha}^\top\boldsymbol{D}\boldsymbol{b} + \boldsymbol{\alpha}^\top(\boldsymbol{D}\boldsymbol{b} - \boldsymbol{\phi})}{n} \\ &= \sigma_y^2 - \frac{\boldsymbol{\alpha}^\top(\boldsymbol{D}\boldsymbol{b} + \boldsymbol{\phi})}{n} \end{aligned} \tag{54}$$

and the phenotype variance σ_y^2 can be derived from summary data as follows:

$$(\sigma_y^2)_j = \frac{(\boldsymbol{y}^\top \boldsymbol{y})_j}{n_j} = \sigma_{b_j}^2 D_{jj}(n_j - 2) + b_j^2 D_{jj}, \tag{55}$$

where $\sigma_{b_j}^2$ is the square of the standard error of b_j. This can also be obtained directly from summary data. Then taking the median over the set of $(\sigma_y^2)_j$ results in a reliable estimate of σ_y^2 (Yang *et al.* 2012). Then the full conditional distribution of σ_e^2 is

$$(\sigma_e^2 \mid \boldsymbol{\alpha}, \boldsymbol{\phi}, \boldsymbol{D}, \boldsymbol{b}, \sigma_y^2) \sim \left(n\sigma_y^2 - \boldsymbol{\alpha}^\top \boldsymbol{D}\boldsymbol{b} - \boldsymbol{\alpha}^\top \boldsymbol{\phi} + \nu_e S_e^2 \right) \chi_{\nu_e+n}^{-2}. \tag{56}$$

Using that the genetic values (GEBVs) of individuals are $\boldsymbol{g} = \boldsymbol{M}\boldsymbol{\alpha}$, then the genetic variance σ_g^2 can be formulated as

$$\sigma_g^2 = \frac{\boldsymbol{\alpha}^\top \boldsymbol{M}^\top \boldsymbol{M}\boldsymbol{\alpha}}{n} = \frac{\boldsymbol{\alpha}^\top (\boldsymbol{D}\boldsymbol{b} - \boldsymbol{\phi})}{n}.$$

Since σ_g^2 is also priori assumed to follow a scaled inverse chi-square distribution, its full condition distribution is

$$(\sigma_g^2 \mid \boldsymbol{\alpha}, \boldsymbol{\phi}, \boldsymbol{D}, \boldsymbol{b}) \sim \left(\boldsymbol{\alpha}^\top \boldsymbol{D}\boldsymbol{b} - \boldsymbol{\alpha}^\top \boldsymbol{\phi} + \nu_g S_g^2 \right) \chi_{\nu_g+n}^{-2},$$

where ν_g and S_g^2 are the pre-defined degrees of freedom and the scale factor of the scaled inverse chi-square distribution.

Affiliation:

Lilin Yin
Key Laboratory of Agricultural Animal Genetics, Breeding and Reproduction, Ministry of Education
College of Animal Science and Technology
Huazhong Agricultural University
Wuhan, Hubei, 430070, PR China
E-mail: ylilin@mail.hzau.edu.cn

Haohao Zhang
School of Computer Science and Artificial Intelligence
Wuhan University of Technology
Wuhan, Hubei, 430070, PR China
E-mail: haohaozhang@whut.edu.cn

Xinyun Li
Key Laboratory of Agricultural Animal Genetics, Breeding and Reproduction, Ministry of Education
College of Animal Science and Technology
Huazhong Agricultural University
Wuhan, Hubei, 430070, PR China
E-mail: xyli@mail.hzau.edu.cn

Shuhong Zhao
Key Laboratory of Agricultural Animal Genetics, Breeding and Reproduction, Ministry of Education
Key Lab of Swine Genetics and Breeding of Ministry of Agriculture and Rural Affairs
The Cooperative Innovation Center for Sustainable Pig Production
College of Animal Science and Technology
Huazhong Agricultural University
Hubei Hongshan Laboratory
Wuhan, Hubei, 430070, PR China
E-mail: shzhao@mail.hzau.edu.cn

Xiaolei Liu
Key Laboratory of Agricultural Animal Genetics, Breeding and Reproduction, Ministry of Education
College of Animal Science and Technology
Shenzhen Institute of Nutrition and Health
Huazhong Agricultural University
Hubei Hongshan Laboratory
Wuhan, Hubei, 430070, PR China
E-mail: xiaoleiliu@mail.hzau.edu.cn

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
September 2025, Volume 114, Issue 6 *Submitted:* 2022-02-13
doi:10.18637/jss.v114.i06 *Accepted:* 2023-10-10