Documents for VIGoR ver 1.1.0

Oct. 2021

Akio Onogi
onogiakio@gmail.com

Contents

1. Introduction

1-1. Overview

VIGoR (Variational Bayesian Inference for Genome-wide Regression) is an R package for fast genome-wide regression using variational Bayesian inference (VB). Although VIGoR is designed primarily for genome-wide regression, VIGoR also can accept any explanatory variables. VIGoR support eight learners (i.e., regression methods), Bayesian lasso (BL), extended Bayesian lasso (EBL), BayesA, BayesB, BayesC, Bayesian ridge regression (BRR), BLUP, and fixed effects (FIXED) (i.e., effects with non-informative priors). These learners can be included in a single model jointly.

VIGoR has a main R function *vigor* which has three main applications: 1) ***fitting***, 2) ***tuning***, and 3) ***cv*** (cross-validation). ***fitting*** conducts model fitting to data. ***tuning*** conducts model fitting after hyperparameter tuning using cross-validation. Multiple hyperparameter sets should be provided by users. ***cv*** evaluates the predictive ability of models using cross-validation (CV). When multiple hyperparameter sets are given, nested cv is conducted for hyperparameter tuning. VIGoR has two other R functions, *hyperpara* and *predict_vigor*. The former returns hyperparameter values based on several assumptions about the genetic architecture given by users. The latter returns predicted values based on the fitting result by vigor and new explanatory variables.

1-2. Installation

The R package includes:

- R functions (*vigor, hyperpara, predict_vigor*)
- Sample data (*X* and *Z*, explanatory variables; *Y*, response variables)
- R documentation

The R package is deposited at CRAN (http://cran.r-project.org/), and can be installed and imported to R in a standard manner. VIGoR requires installation of rrBLUP package which is used to calculate genomic relationship matrices.

1-3. Examples of usage

After installing the package *vigor*, load the package and sample data by typing

```
library(vigor)
data(sampledata)
```

To execute ***Model fitting*** using BayesC, type

```
ETA <- list(list(model = "BayesC", X = X))
Result <- vigor(Y, ETA)
```

Here, Y is a vector including phenotypic values (response variables) of an imaginary trait. X is a matrix of SNP genotypes (explanatory variables). **Note that the intercept is automatically added as the last**

learner when no fixed effect is added as FIXED by user.

vigor returns a list object (see **Subsection 4-3. Output lists**). The estimated SNP effects (β) and inclusion probability (ρ) can be visualized by

```
plot(Result$ETA[[1]]$Beta)
plot(Result$ETA[[1]]$Rho)
```

The intercept is added as the last (second in this case) learner.

```
Result$ETA[[2]]$Beta
```

Fixed effects can be incorporated by, for example, typing

```
Z.matrix <- model.matrix(~ Z)
ETA <- list(list(model = "FIXED", X = Z.matrix),
            list(model = "EBL", X = X))
Result <- vigor(Y, ETA)
```

Here Z is a vector consisting of characters “A”, “B”, and “C” which represent the levels of the fixed effect. **When FIXED is added by user, the intercept is not automatically added. Thus, variables in FIXED should contain the intercept.**

SNP effects estimated by EBL can be visualized by typing

```
plot(abs(Result$ETA[[2]]$Beta))
```

Alternatively, fixed effects can be incorporated using formula.

```
Data <- data.frame(Z = factor(Z))
ETA <- list(list(~ Z, model = "FIXED", data = Data),
            list(model = "BayesA", X = X))
Result <- vigor(Y, ETA)
```

Multiple learners (regression methods) can be incorporated in a single model. Some SNPs in X have dominance (non-additive) effects. Here additive and dominance effects are modeled using BayesC with different shrinkage levels.

```
X.d <- X
X.d[X == 2] <- 0
ETA <- list(list(~ Z, model = "FIXED", data = Data),
            list(model = "BayesC", X = X, H = c(5, 0.1, 0.01)),
            list(model = "BayesC", X = X.d, H = c(5, 0.1, 0.001)))
Result <- vigor(Y, ETA)
```

Here X.d is created from X to model dominance effects. The additive and dominance effects are modeled

with BayesC as the second and third learner. H denotes hyperparameters, and the three values of H correspond with v, S^2, κ (see **Subsection 2-1. Model descriptions** and Table 2-3). Inclusion probabilities of the additive and dominance effects can be confirmed by typing

```
plot(Result$ETA[[2]]$Rho)
plot(Result$ETA[[3]]$Rho)
```

Of course, different regression methods can be incorporated by, for example, typing

```
ETA <- list(list(~ Z, model = "FIXED", data = Data),
            list(model = "BL", X = X, H = c(1, 0.01)),
            list(model = "BayesC", X = X.d, H = c(5, 0.1, 0.001)))
Result <- vigor(Y, ETA)
```

To execute ***Model fitting after hyperparameter tuning***, specify “tuning” in the argument Function. To prepare candidate hyperparameter values, another function of this package, *hyperpara*, can be used.

```
ETA <- list(list(~ Z, model = "FIXED", data = Data),
            list(model = "BayesB", X = X,
                 H = hyperpara(X, 0.5, "BayesB", c(0.1, 0.01))))
Result <- vigor(Y, ETA, Function = "tuning")
```

hyperpara returns hyperparameter sets under two assumptions on genetic architecture (see **Section 3. Hyperparameters**). For example, type

```
hyperpara(X, 0.5, "BL", 0.01, Verbose = TRUE)
```

Here the second argument *Mvar* (0.5) indicates the proportion of phenotypic variance (i.e., variance of response variables) explainable by the SNPs (explanatory variables). The fourth argument *Kappa* (0.01) indicates **the proportion of SNPs with NON-ZERO EFFECTS.** Thus, this example assumes that 50% of the phenotypic variance is explained by 1% of the SNPs.

When vectors are given to *Mvar* and/or *Kappa*, a matrix of all combinations of hyperparameter values are returned. For example, if type

```
hyperpara(X, c(0.3, 0.5, 0.8), "BayesC", c(0.1, 0.01))
```

indicating that total 3 (*Mvar*) × 2 (*Kappa*) = 6 hyperparameter sets are returned as a matrix.

In ***Model fitting after hyperparameter tuning,*** when multiple learners are used, all combinations of hyperparameter sets are compared. For example, consider the following scripts.

```
ETA <- list(list(~ Z, model = "FIXED", data = Data),
            list(model = "BayesB", X = X,
                 H = hyperpara(X, 0.5, "BayesB", c(0.1,0.01))),
            list(model = "BayesC", X = X.d,
                 H = hyperpara(X, 0.5, "BayesC", c(0.1,0.01))))
Result <- vigor(Y, ETA, Function = "tuning")
```

Here BayesB and BayesC have two candidate sets, respectively. Thus, total 2 × 2 = 4 combinations (hyperparameter sets) are compared in CV.

The results tuned with CV can be confirmed by typing

```
Result$Metrics
Result$H
```

The first line above outputs the prediction errors under each hyperparameter set in CV. The second line outputs the selected hyperparameter sets after tuning.

The model is fitted to the full data with this best hyperparameter set. Inclusion probabilities of each method estimated with the tuned hyperparameters can be visualized by typing

```
plot(Result$ETA[[2]]$Rho)
plot(Result$ETA[[3]]$Rho)
```

To execute ***Cross-Validation***, specify “cv” in the argument Function as, for example,

```
ETA <- list(list(~ Z, model = "FIXED", data = Data),
            list(model = "BayesC", X = X,
                 H = hyperpara(X, 0.5, "BayesC", c(0.1,0.01))))
Result <- vigor(Y, ETA, Function="cv")
```

Because two hyperparameter sets are given to BayesC, nested CV is conducted at each fold to tune hyperparameters.

To confirm which set was selected at each fold, type

```
Result$Metrics
```

To compare predicted values with Y, type

```
plot(Y, Result$Prediction)
```

To conduct CV with another model using the same partition, use the argument Partition (see **Subsection 4-1. Mandatory arguments**). For example, type

```
ETA <- list(list(~ Z, model = "FIXED", data = Data),
            list(model = "EBL", X = X))
Result2 <- vigor(Y, ETA, Function="cv", Partition = Result$Partition)
plot(Y, Result2$Prediction)
```

To predict for new X, predict_vigor can be used. Train EBL using the first 80% of data and then predict the remaining 20% data.

```
ETA <- list(list(model = "EBL",X = X[1:400, ]))
Train <- vigor(Y[1:400], ETA)
newX <- list(X[401:500, ])
Predict <- predict_vigor(Train, newX)
plot(Y[401:500], Predict)
```

When the intercept is automatically added when training, the intercept is again automatically added to predicted values. **New X has to be provided as a list.**

In the next example, multiple regression methods are used in a single model.

```
ETA <- list(list(~ Z, model = "FIXED", data = Data[1:400, ,drop=FALSE]),
            list(model = "BayesB", X = X[1:400, ], H = c(5, 0.1, 0.01)),
            list(model = "BayesB", X = X.d[1:400, ], H = c(5, 0.1, 0.001)))
Train <- vigor(Y[1:400], ETA)
newX <- list(~ Z, data = Data[401:500, , drop=FALSE], X[401:500, ], X.d[401:500, ])
Predict <- predict_vigor(Train, newX)
plot(Y[401:500], Predict)
```

When multiple learners are included in ETA, explanatory variables in new X also should be arranged in the same order. When the fixed effects are specified as a matrix not as a formula, data in new X is unnecessary.

Contributions of each learner to prediction can be assessed by filling new X with NULL. For example, to see the contribution of the additive effects, type

```
newX <- list(NULL, X[401:500, ], NULL)
Predict <- predict_vigor(Train, newX)
plot(Y[401:500], Predict)
```

The first and third NULL in new X correspond with the fixed and dominance effects, and are ignored by predict_vigor. Thus, values predicted by only the additive effects are returned.

2- Regression methods and algorithms

2-1 Model descriptions

The linear regression model assumed in VIGoR ver. 1.1 is, for individual *i*,

$$y_i = \sum_{m=1}^{M} f_m(\mathbf{x}_{m,i}, \quad \boldsymbol{\theta}_m) + \varepsilon_i$$

where y_i is the observed value (e.g., phenotypic value), M is the number of learners (i.e., regression methods) included in the model, f_m indicates the mth learner, $\mathbf{x}_{m,i}$ is the explanatory variables (e.g., SNP genotypes) for learner m and sample i, $\boldsymbol{\theta}_m$ is the parameters of learner m, and ε_i is the residual. Likelihood is defined as,

$$N\left(\sum_{m=1}^{M} f_m(\mathbf{x}_{m,i}, \quad \boldsymbol{\theta}_m), \frac{1}{\tau_0^2}\right)$$

where N indicates a normal distribution, and τ_0^2 is the precision of the residuals. The prior distribution of τ_0^2 is $p(\tau_0^2) = \frac{1}{\tau_0^2}$. Note that the index of **x** for learners (m) is not explicitly shown in the equations in **Subsection 2-3. Update procedures** for the sake of simplicity.

VIGoR implements eight regression methods, Bayesian lasso (BL), extended Bayesian lasso (EBL), BayesA, BayesB, BayesC, Bayesian ridge regression (BRR), BLUP, and fixed effects (FIXED). Here BLUP indicates random effects of which covariance structure is defined with **x**, and fixed effects indicate the effects which are assumed to be proportional to constant values. **Table 2-1** presents the references of the regression methods and VB algorithms. BRR and FIXED are excluded from the table because these methods are fundamentally, and exact references are difficult to be defined. The prior distributions of these effects are presented in **Table 2-2**. Learners except for BLUP have regression coefficients (**β**) of **x** at the first hierarchy level, and BLUP have random effects (**u**). A list of hyperparameters is presented in **Table 2-3**.

In the development of VIGoR, we modified the VB algorithms for BL and EBL from those in Li and Sillanpaa (2012): in their parameterization, the SNP effect is independent from the residual variance, whereas in VIGoR, it is conditional on the residual variance. In our experience, this manipulation slightly accelerated convergence. Carbonetto and Stephens (2012) introduced a VB algorithm for linear regression with a spike and slab prior which is equivalent to BayesC. The difference between the algorithm for BayesC in VIGoR and that in Carbonetto and Stephens (2012) is that Carbonetto and Stephens used the importance sampling technique to infer the posterior distribution of σ^2, τ_0^2, and κ, whereas we inferred the factorized posteriors of σ^2 and τ_0^2 and used a fixed value for κ.

Table 2-1 References of regression methods and VB algorithms

	Regression methods	VB algorithm
BL	Park and Casella (2008)	Li and Sillanpaa (2012)
EBL	Mutshinda and Sillanpaa (2010)	Li and Sillanpaa (2012)
BayesA	Meuwissen et al. (2001)	Onogi and Iwata (2015)
BayesB	Meuwissen et al. (2001)	Onogi and Iwata (2015)
BayesC	Habier et al. (2011)	Carbonetto and Stephens (2012)
BLUP	Wang et al. (1993)	Arakawa et al. (2016)

Table 2-2 Prior distributions of regression methods[a]

Hierarchical level	1st Effect and indicator	2nd Effect variance and indicator	3rd Shrinkage magnitude
BL	$\beta_p \sim N\left(0, \frac{1}{\tau_0^2 \tau_p^2}\right)$	$\tau_p^2 \sim Inv-G\left(1, \frac{\lambda^2}{2}\right)$	$\lambda^2 \sim G(\varphi, \varpi)$
EBL	$\beta_p \sim N\left(0, \frac{1}{\tau_0^2 \tau_p^2}\right)$	$\tau_p^2 \sim Inv-G\left(1, \frac{\delta^2 \eta_p^2}{2}\right)$	$\delta^2 \sim G(\varphi, \varpi)$ $\eta_p^2 \sim G(\psi, \theta)$
BayesA	$\beta_p \sim N(0, \sigma_p^2)$	$\sigma_p^2 \sim \chi^{-2}(\nu, S^2)$	
BayesB	$\beta_p \sim N(0, \sigma_p^2)$ if $\rho_p = 1$ $\beta_p = 0$ if $\rho_p = 0$	$\sigma_p^2 \sim \chi^{-2}(\nu, S^2)$ $\rho_p \sim Bernoulli(\kappa)$	
BayesC	$\beta_p \sim N(0, \sigma^2)$ if $\rho_p = 1$ $\beta_p = 0$ if $\rho_p = 0$	$\sigma^2 \sim \chi^{-2}(\nu, S^2)$ $\rho_p \sim Bernoulli(\kappa)$	
BRR	$\beta_p \sim N(0, \sigma^2)$	$\sigma^2 \sim \chi^{-2}(\nu, S^2)$	
BLUP	$\mathbf{u} \sim N(\mathbf{0}, \mathbf{K}\sigma^2)$	$\sigma^2 \sim \chi^{-2}(\nu, S^2)$	
FIXED	$\beta_p \sim const.$		

[a]*p* is the index of explanatory variables (i.e., *p*th column of **x**).

N, normal distribution; *Inv-G*, inverse-gamma distribution; *G*, gamma distribution; *Bernoulli*, Bernoulli distribution; χ^{-2}, scaled inverse-chi-square distribution; *const*, constant value; **K**, varaince-covariance matrix.

Table 2-3 Hyperparameters

Methods	Hyperparameters
BL	φ, ω
EBL	φ, ω, ψ, θ
BayesA, BRR, BLUP	ν, S^2
BayesB, BayesC	ν, S^2, κ

2-2. Variational Bayesian inference

VB approximates the marginal log likelihood of data ($\boldsymbol{y}$) by maximizing the lower bound of the marginal log likelihood. The lower bound can be written by using Jensen's inequality as

$$\log p(\boldsymbol{y}) = \log \int q(\boldsymbol{\theta}) \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} \geq \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta},$$

(**Eq. 2-1**)

where q denotes any probability distribution of $\boldsymbol{\theta}$. In VB, factorized posterior distributions are used as q. That is,

$$q(\boldsymbol{\theta}) = \prod_{i=1}^{P} q_i(\boldsymbol{\theta}_i|\boldsymbol{y}),$$

where P is the number of parameters included in the model. Hereinafter, we denote the factorized posterior distributions as pseudo posterior distributions. The lower bound can be maximized with regard to $q_i(\boldsymbol{\theta}_i|\boldsymbol{y})$ by setting

$$q_i(\boldsymbol{\theta}_i|\boldsymbol{y}) \propto \exp\left(E_{q_j, j\neq i}[\log p(\boldsymbol{y}, \boldsymbol{\theta})]\right)$$

because

$$\begin{aligned}
&\int q(\boldsymbol{\theta}|\boldsymbol{y}) \log \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{y})} d\boldsymbol{\theta} \\
&\quad = \int q_i(\boldsymbol{\theta}_i|\boldsymbol{y}) \prod_{j\neq i} q_j(\boldsymbol{\theta}_j|\boldsymbol{y}) \left[\log p(\boldsymbol{y}, \boldsymbol{\theta}) - \log q_i(\boldsymbol{\theta}_i|\boldsymbol{y})\right] d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}|\boldsymbol{y}) \sum_{j\neq i} \log q_j(\boldsymbol{\theta}_j|\boldsymbol{y})\, d\boldsymbol{\theta} \\
&\quad = \int q_i(\boldsymbol{\theta}_i|\boldsymbol{y}) \left\{ \int \prod_{j\neq i} q_j(\boldsymbol{\theta}_j|\boldsymbol{y}) \log p(\boldsymbol{y}, \boldsymbol{\theta})\, d\,\boldsymbol{\theta}_{j\neq i} - \log q_i(\boldsymbol{\theta}_i|\boldsymbol{y}) \right\} d\boldsymbol{\theta}_i + Const. \\
&\quad = -KL\left\{ q_i(\boldsymbol{\theta}_i|\boldsymbol{y}) || \exp\left(E_{q_j, j\neq i}[\log p(\boldsymbol{y}, \boldsymbol{\theta})]\right) \right\} + Const.
\end{aligned}$$

where *KL* indicates the Kullback–Leibler divergence. Here, we use $\int q_j(\theta_j|y) d\theta_j = 1$, which holds for any j. The maximization of the lower bound is equivalent to the minimization of the *KL* divergence between the posterior and the pseudo posterior distributions of $\boldsymbol{\theta}$. That is,

$$\begin{aligned}
\log p(\boldsymbol{y}) = \int q(\boldsymbol{\theta}|\boldsymbol{y}) \log p(\boldsymbol{y})\, d\boldsymbol{\theta} &= \int q(\boldsymbol{\theta}|\boldsymbol{y}) \log \frac{p(\boldsymbol{y}) p(\boldsymbol{\theta}|\boldsymbol{y})}{p(\boldsymbol{\theta}|\boldsymbol{y})} d\boldsymbol{\theta} \\
&= \int q(\boldsymbol{\theta}|\boldsymbol{y}) \log \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{y})} d\boldsymbol{\theta} - \int q(\boldsymbol{\theta}|\boldsymbol{y}) \log \frac{p(\boldsymbol{\theta}|\boldsymbol{y})}{q(\boldsymbol{\theta}|\boldsymbol{y})} d\boldsymbol{\theta} \\
&= \int q(\boldsymbol{\theta}|\boldsymbol{y}) \log \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{y})} d\boldsymbol{\theta} + KL[q(\boldsymbol{\theta}|\boldsymbol{y}) || p(\boldsymbol{\theta}|\boldsymbol{y})]
\end{aligned}$$

In VB, as well as in Markov chain Monte Carlo, the posterior uncertainty of parameters can be inferred in addition to the posterior means. More detailed descriptions of VB can be found elsewhere, e.g. Bishop (2006) or Murphy (2012).

2-3. Update procedures

In VB, parameters are iteratively updated from initial values until convergence. The iterative update is stopped when $\frac{\|\boldsymbol{\theta}^{*}-\boldsymbol{\theta}\|^2}{\|\boldsymbol{\theta}^{*}\|^2} < Th$, where $\|\ \|$ is the Euclidean norm, $\boldsymbol{\theta}$ is the vector that contains all parameter values at the previous iteration, $\boldsymbol{\theta}^{*}$ is the vector consisting of newly updated parameter values at the iteration, and *Th* defines the criterion for convergence. Update procedures for each regression method are described in the subsequent sections.

2-3-1. Bayesian lasso (BL)

The joint log posterior distribution, $\log p(\boldsymbol{y}, \boldsymbol{\theta})$, is

$$\frac{N}{2}\log\tau_0^2 + \frac{\tau_0^2}{2}\sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m) - \sum_{p=1}^{P} x_{ip}\beta_p\right)^2$$

$$-\log\tau_0^2 + \frac{P}{2}\log\tau_0^2 + \frac{1}{2}\sum_{p=1}^{P}\log\tau_p^2 - \frac{\tau_0^2}{2}\sum_{p=1}^{P}\tau_p^2\beta_p^2 - 2\sum_{p=1}^{P}\log\tau_p^2 - \frac{1}{2}\sum_{p=1}^{P}\frac{\lambda^2}{\tau_p^2} + (\phi-1)\log\lambda^2 - \varpi\lambda^2 + Const.$$

where $\sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)$ is the summation of regression methods included in the model other than BL, and *Const.* indicates the constant term. In VB, the joint posterior distribution of parameters is factorized into pseudo posterior distributions for each parameter as described above. The pseudo posterior distribution of a parameter is obtained by taking expectations of the joint posterior log likelihood with regard to the pseudo posterior distributions of the remaining parameters.

The pseudo posterior distribution of β_p is a normal distribution with

$$E[\beta_p] = H_p E[\tau_0^2]\sum_{i=1}^{N} x_{ip}\left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k\neq p}^{P} E[\beta_k]x_{ik}\right)$$

and

$$V[\beta_p] = H_p$$

where

$$H_p^{-1} = E[\tau_0^2]\sum_{i=1}^{N} x_{ip}^2 + E[\tau_p^2]E[\tau_0^2]$$

and $E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)]$ is mentioned later. The pseudo posterior distribution of τ_p^2 is an inverse-Gaussian distribution with

$$E[\tau_p^2] = \mu_p$$

and

$$E\left[\frac{1}{\tau_p^2}\right] = \frac{1}{\mu_p} + \frac{1}{\xi_p}$$

where

$$\mu_p = \sqrt{\frac{E[\lambda^2]}{E[\beta_p^2]E[\tau_0^2]}}$$

and

$$\xi_p = E[\lambda^2].$$

Here we use

$$E[X^r] = \mu^r \sum_{s=0}^{r-1} \frac{(r-1+s)!}{s!\,(r-1-s)!} \left(\frac{2\xi}{\mu}\right)^{-s}$$

and

$$E[X^{-r}] = \frac{E[X^{r+1}]}{\mu^{2r+1}}$$

when X follows an inverse Gaussian distribution with parameters μ and ζ (Chhikara and Folks 1989). The pseudo posterior distribution of λ^2 is also a gamma distribution with

$$E[\lambda^2] = \frac{a_2}{b_2}$$

where

$$a_2 = P + \phi$$

and

$$b_2 = \frac{1}{2}\sum_{p=1}^{P} E\left[\frac{1}{\tau_p^2}\right] + \varpi.$$

2-3-2. Extended Bayesian lasso (EBL)

The joint log posterior distribution is

$$\frac{N}{2}\log\tau_0^2 + \frac{\tau_0^2}{2}\sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m) - \sum_{p=1}^{P} x_{ip}\beta_p\right)^2$$

$$-\log\tau_0^2 + \frac{P}{2}\log\tau_0^2 + \frac{1}{2}\sum_{p=1}^{P}\log\tau_p^2 - \frac{\tau_0^2}{2}\sum_{p=1}^{P}\tau_p^2\beta_p^2 - 2\sum_{p=1}^{P}\log\tau_p^2 - \frac{1}{2}\sum_{p=1}^{P}\frac{\delta^2\eta_p^2}{\tau_p^2}$$

$$+(\phi-1)\log\delta^2 - \varpi\delta^2 + (\psi-1)\sum_{p=1}^{P}\log\eta_p^2 - \theta\sum_{p=1}^{P}\eta_p^2 + Const.$$

As in BL, the prior distributions of the SNP effects are conditioned by the residual variance. The pseudo posterior distribution of β_p is a normal distribution with

$$E[\beta_p] = H_p E[\tau_0^2]\sum_{i=1}^{N} x_{ip}\left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k\neq p}^{P} E[\beta_k]x_{ik}\right)$$

and

$$V[\beta_p] = H_p$$

where

$$H_p^{-1} = E[\tau_0^2] \sum_{i=1}^{N} x_{ip}^2 + E[\tau_p^2] E[\tau_0^2].$$

The pseudo posterior distribution of τ_p^2 is an inverse-Gaussian distribution with

$$E[\tau_p^2] = \sqrt{\frac{E[\delta^2] E[\eta_p^2]}{E[\beta_p^2] E[\tau_0^2]}}$$

and

$$E\left[\frac{1}{\tau_p^2}\right] = \sqrt{\frac{E[\beta_p^2] E[\tau_0^2]}{E[\delta^2] E[\eta_p^2]}} + \frac{1}{E[\delta^2] E[\eta_p^2]}.$$

The pseudo posterior distributions of δ^2 and η_p^2 are gamma distributions; the expectations are

$$E[\delta^2] = \frac{a_2}{b_2}$$

where

$$a_2 = P + \phi$$

and

$$b_2 = \frac{1}{2} \sum_{p=1}^{P} E[\eta_p^2] E\left[\frac{1}{\tau_p^2}\right] + \varpi,$$

and

$$E[\eta_p^2] = \frac{a_3}{b_{3,p}},$$

where

$$a_3 = 1 + \psi$$

and

$$b_{3,p} = \frac{1}{2} E[\delta^2] E\left[\frac{1}{\tau_p^2}\right] + \theta.$$

2-3-3. BayesA

The joint log posterior distribution is

$$\frac{N}{2} \log \tau_0^2 - \frac{\tau_0^2}{2} \sum_{i=1}^{N} \left(y_i - \sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m) - \sum_{p=1}^{P} x_{ip} \beta_p \right)^2$$

$$- \log \tau_0^2 + \sum_{p=1}^{P} \left[-\frac{1}{2} \log \sigma_p^2 - \frac{\beta_p^2}{2\sigma_p^2} \right] + \left(-\frac{\nu}{2} - 1 \right) \sum_{p=1}^{P} \log \sigma_p^2 - \frac{\nu S^2}{2} \sum_{p=1}^{P} \frac{1}{\sigma_p^2} + Const.$$

The pseudo posterior distribution of β_p is a normal distribution with

$$E[\beta_p] = H_p E[\tau_0^2] \sum_{i=1}^{N} x_{ip} \left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k \neq p}^{P} E[\beta_k] x_{ik} \right)$$

and

$$V[\beta_p] = H_p,$$

where

$$H_p^{-1} = E[\tau_0^2] \sum_{i=1}^{N} x_{ip}^2 + \frac{1}{\tilde{S}_p^2}.$$

Herein,

$$\tilde{S}_p^2 = \frac{v S^2 + E[\beta_p^2]}{\tilde{v}_p^2},$$

where

$$\tilde{v}_p = v + 1,$$

The pseudo posterior distribution of σ^2 is a scaled inverse-chi-squared distribution with

$$E\left[\frac{1}{\sigma^2}\right] = \frac{1}{\tilde{S}_p^2}.$$

2-3-4. BayesB

The joint log posterior distribution is

$$\frac{N}{2} \log \tau_0^2 - \frac{\tau_0^2}{2} \sum_{i=1}^{N} \left(y_i - \sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m) - \sum_{p=1}^{P} x_{ip} \beta_p \right)^2$$

$$- \log \tau_0^2 + \sum_{p=1}^{P} \rho_p \left[-\frac{1}{2} \log \sigma_p^2 - \frac{\beta_p^2}{2\sigma_p^2} \right] + \left(-\frac{\nu}{2} - 1 \right) \sum_{p=1}^{P} \log \sigma_p^2 - \frac{\nu S^2}{2} \sum_{p=1}^{P} \frac{1}{\sigma_p^2}$$

$$+ \left[\sum_{p=1}^{P} \rho_p \right] \log \kappa + \left[P - \sum_{p=1}^{P} \rho_p \right] \log(1 - \kappa) + Const.$$

For β_p and ρ_p, we consider the joint posterior distribution to be,

$$q(\beta_p, \rho_p) \propto -\frac{E[\tau_0^2]}{2} \left[\beta_p^2 \sum_{i=1}^{N} x_{ip}^2 - 2\beta_p \sum_{i=1}^{N} x_{ip} \left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k \neq p}^{P} x_{ik} E[\beta_k] \right) \right]$$

$$+ \rho_p \left[\frac{1}{2} \Phi\left(\frac{\tilde{v}_p}{2} \right) - \frac{1}{2} log \frac{\tilde{v}_p \tilde{S}_p^2}{2} - \frac{\beta_p^2}{2\tilde{S}_p^2} \right] + \rho_p \log \kappa + (1 - \rho_p) \log(1 - \kappa)$$

(**Eq. 2-2**)

where

$$\tilde{v}_p = v + E[\rho_p],$$

$$\tilde{S}_p^2 = \frac{\nu S^2 + E\left[\beta_p^2\right]}{\tilde{\nu}_p^2},$$

and Φ indicates the digamma function, that is,

$$\Phi(x) = \frac{d\Gamma(x)}{dx}\Gamma(x)^{-1} = \Gamma'(x)\Gamma(x)^{-1}.$$

The digamma function stems from

$$\int q(x)\log x\,dx = -\Phi\left(\frac{\nu}{2}\right) + \log\frac{\nu S^2}{2} \qquad \textbf{(Eq. 2-3)}$$

where $q(x)$ indicates the density of $\chi^{-2}(\nu, S^2)$ and is written as

$$q(x) = \Gamma\left(\frac{\nu}{2}\right)^{-1}\left(\frac{\nu S^2}{2}\right)^{\frac{\nu}{2}} x^{-\frac{\nu}{2}-1}\exp\left(-\frac{\nu S^2}{2x}\right).$$

Eq. 2-3 can be derived as follows: first, differentiate a part of $q(x)$ with respect to ν:

$$\frac{d}{d\nu}\left[x^{-\frac{\nu}{2}-1}\exp\left(-\frac{\nu S^2}{2x}\right)\right] = x^{-\frac{\nu}{2}-1}\left(-\frac{S^2}{2x}\right)\exp\left(-\frac{\nu S^2}{2x}\right) - \frac{1}{2}x^{-\frac{\nu}{2}-1}\log x\exp\left(-\frac{\nu S^2}{2x}\right).$$

By multiplying both sides by $D = \Gamma\left(\frac{\nu}{2}\right)^{-1}\left(\frac{\nu S^2}{2}\right)^{\frac{\nu}{2}}$, and integrating both sides with respect to x, we obtain

$$D\frac{d}{d\nu}\int x^{-\frac{\nu}{2}-1}\exp\left(-\frac{\nu S^2}{2x}\right)dx = -\frac{S^2}{2}\int\frac{1}{x}Dx^{-\frac{\nu}{2}-1}\exp\left(-\frac{\nu S^2}{2x}\right)dx - \frac{1}{2}\int \log x D x^{-\frac{\nu}{2}-1}\exp\left(-\frac{\nu S^2}{2x}\right)dx$$

$$= -\frac{S^2}{2}\int q(x)\frac{1}{x}dx - \frac{1}{2}\int q(x)\log x dx.$$

Using $\int D^{-1}q(x)dx = D^{-1}$, we obtain

$$D\frac{dD^{-1}}{d\nu} = -\frac{S^2}{2}\int q(x)\frac{1}{x}dx - \frac{1}{2}\int q(x)\log x\,dx.$$

This can be rewritten as

$$\int q(x)\log x\,dx = -2D\frac{dD^{-1}}{d\nu} - S^2\int q(x)\frac{1}{x}dx = -2D\frac{dD^{-1}}{d\nu} - 1.$$

Herein, because

$$\frac{dD^{-1}}{d\nu} = \frac{d}{d\nu}\left[\Gamma\left(\frac{\nu}{2}\right)\left(\frac{\nu S^2}{2}\right)^{-\frac{\nu}{2}}\right] = \frac{1}{2}\Gamma'\left(\frac{\nu}{2}\right)\left(\frac{\nu S^2}{2}\right)^{-\frac{\nu}{2}} + \Gamma\left(\frac{\nu}{2}\right)\frac{d}{d\nu}\left(\frac{\nu S^2}{2}\right)^{-\frac{\nu}{2}}$$

$$= \frac{1}{2}\Gamma'\left(\frac{\nu}{2}\right)\left(\frac{\nu S^2}{2}\right)^{-\frac{\nu}{2}} - \frac{1}{2}\Gamma\left(\frac{\nu}{2}\right)\left(\log\frac{\nu S^2}{2} + 1\right)\left(\frac{\nu S^2}{2}\right)^{-\frac{\nu}{2}},$$

we obtain

$$\int q(x)\log x\,dx = -2D\frac{dD^{-1}}{d\nu} - 1$$

$$= -2\Gamma\left(\frac{\nu}{2}\right)^{-1}\left(\frac{\nu S^2}{2}\right)^{\frac{\nu}{2}}\left[\frac{1}{2}\Gamma'\left(\frac{\nu}{2}\right)\left(\frac{\nu S^2}{2}\right)^{-\frac{\nu}{2}} - \frac{1}{2}\Gamma\left(\frac{\nu}{2}\right)\left(\log\frac{\nu S^2}{2}+1\right)\left(\frac{\nu S^2}{2}\right)^{-\frac{\nu}{2}}\right] - 1$$

$$= -\Gamma\left(\frac{\nu}{2}\right)^{-1}\Gamma'\left(\frac{\nu}{2}\right) + \log\frac{\nu S^2}{2} = -\Phi\left(\frac{\nu}{2}\right) + \log\frac{\nu S^2}{2}.$$

The pseudo posterior distribution of ρ_p can be obtained by integrating out β_p in Eq. 2-2 as

$$q(\rho_p = 1) = \int q(\beta_p, \rho_p = 1)\,d\beta_p$$

$$\propto \frac{H_p}{2}\left[E[\tau_0^2]\sum_{i=1}^{N} x_{ip}\left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k\neq p}^{P} E[\beta_k]x_{ik}\right)\right]^2 + \frac{1}{2}\log H_p + \frac{1}{2}\Phi\left(\frac{\tilde{\nu}_p}{2}\right)$$

$$-\frac{1}{2}\log\frac{\tilde{\nu}_p\tilde{S}_p^2}{2} + \log\kappa \propto F_p + \log\kappa$$

where

$$H_p^{-1} = E[\tau_0^2]\sum_{i=1}^{N} x_{ip}^2 + \frac{1}{\tilde{S}_p^2}.$$

Similarly, we obtain $q(\rho_p = 0) \propto log(1-\kappa)$. Consequently,

$$E[\rho_p] = \frac{\kappa\, exp(F_p)}{\kappa\, exp(F_p) + (1-\kappa)}.$$

The pseudo posterior distribution of β_p can be obtained by integrating out ρ_p in Eq. 2-2 as

$$q(\beta_p) = q(\beta_p, \gamma_p = 1) + q(\beta_p, \gamma_p = 0) = q(\beta_p|\gamma_p = 1)q(\gamma_p = 1) + q(\beta_p|\gamma_p = 0)q(\gamma_p = 0).$$

Thus, $E[\beta_p] = E[\beta_p|\rho_p = 1]E[\rho_p]$ and $E[\beta_p^2] = E[\beta_p^2|\rho_p = 1]E[\rho_p]$. From Eq. 2-2, $q(\beta_p|\gamma_p = 1)$ turns out to be a normal distribution with

$$E[\beta_p|\rho_p = 1] = H_p E[\tau_0^2]\sum_{i=1}^{N} x_{ip}\left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k\neq p}^{P} E[\beta_k]x_{ik}\right)$$

and

$$V[\beta_p|\rho_p = 1] = H_p.$$

The second moment can be obtained as

$$E[\beta_p^2|\rho_p = 1] = V[\beta_p|\rho_p = 1] + E[\beta_p|\rho_p = 1]^2.$$

The pseudo posterior distribution of σ^2 is a scaled inverse-chi-squared distribution with

$$E\left[\frac{1}{\sigma^2}\right] = \frac{1}{\tilde{S}_p^2}.$$

2-3-5. BayesC

The joint log posterior distribution is

$$\frac{N}{2}\log\tau_0^2 - \frac{\tau_0^2}{2}\sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m) - \sum_{p=1}^{P} x_{ip}\beta_p\right)^2$$

$$-\log\tau_0^2 + \sum_{p=1}^{P}\rho_p\left[-\frac{1}{2}\log\sigma^2 - \frac{\beta_p^2}{2\sigma^2}\right] + \left(-\frac{\nu}{2} - 1\right)\log\sigma^2 - \frac{\nu S^2}{2\sigma^2}$$

$$+\left[\sum_{p=1}^{P}\rho_p\right]\log\kappa + \left[P - \sum_{p=1}^{P}\rho_p\right]\log(1-\kappa) + Const.$$

For β_p and ρ_p, as we do in BayesB, we consider the joint posterior distribution to be,

$$q(\beta_p, \rho_p) \propto -\frac{E[\tau_0^2]}{2}\left[\beta_p^2\sum_{i=1}^{N} x_{ip}^2 - 2\beta_p\sum_{i=1}^{N} x_{ip}\left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k\neq p}^{P} x_{ik}E[\beta_k]\right)\right]$$

$$+\rho_p\left[\frac{1}{2}\Phi\left(\frac{\tilde{v}}{2}\right) - \frac{1}{2}\log\frac{\tilde{v}\tilde{S}^2}{2} - \frac{\beta_p^2}{2\tilde{S}^2}\right] + \rho_p\log\kappa + (1-\rho_p)\log(1-\kappa),$$

(**Eq. 2-4**)

where

$$\tilde{v} = v + \sum_{j=1}^{P} E[\rho_j]$$

and

$$\tilde{S}^2 = \frac{vS^2 + \sum_{j=1}^{P} E[\beta_j^2]}{\tilde{v}}.$$

The pseudo posterior distribution of ρ_p can be obtained by integrating out β_p in Eq. 2-4 as

$$q(\rho_p = 1) = \int q(\beta_p, \rho_p = 1)\, d\beta_p$$

$$\propto \frac{H_p}{2}\left[E[\tau_0^2]\sum_{i=1}^{N} x_{ip}\left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k\neq p}^{P} E[\beta_k]x_{ik}\right)\right]^2 + \frac{1}{2}\log H_p + \frac{1}{2}\Phi\left(\frac{\tilde{v}}{2}\right)$$

$$-\frac{1}{2}\log\frac{\tilde{v}\tilde{S}^2}{2} + \log\kappa \propto F_p + \log\kappa,$$

where

$$H_p^{-1} = E[\tau_0^2]\sum_{i=1}^{N} x_{ip}^2 + \frac{1}{\tilde{S}}.$$

Similarly, we obtain $q(\rho_p = 0) \propto log(1-\kappa)$. Consequently,

$$E[\rho_p] = \frac{\kappa\, exp(F_p)}{\kappa\, exp(F_p) + (1-\kappa)}.$$

The pseudo posterior distribution of β_p can be obtained by integrating out ρ_p in Eq. 2-4 as done in BayesB. We obtain

$$E[\beta_p] = E[\beta_p|\rho_p = 1]E[\rho_p],$$

$$E\left[\beta_p^2\right] = E\left[\beta_p^2|\rho_p = 1\right]E\left[\rho_p\right],$$

$$E\left[\beta_p|\rho_p = 1\right] = H_p E[\tau_0^2]\sum_{i=1}^{N} x_{ip}\left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k\neq p}^{P} E[\beta_k]x_{ik}\right)$$

and

$$V\left[\beta_p|\rho_p = 1\right] = H_p.$$

The second moment can be obtained as

$$E\left[\beta_p^2|\rho_p = 1\right] = V\left[\beta_p|\rho_p = 1\right] + E\left[\beta_p|\rho_p = 1\right]^2.$$

The pseudo posterior distribution of σ^2 is a scaled inverse-chi-squared distribution with

$$E\left[\frac{1}{\sigma^2}\right] = \frac{1}{\tilde{S}^2}.$$

2-3-6. Bayesian ridge regression (BRR)

The joint log posterior distribution is

$$\frac{N}{2}\log\tau_0^2 - \frac{\tau_0^2}{2}\sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m) - \sum_{p=1}^{P} x_{ip}\beta_p\right)^2$$

$$-\log\tau_0^2 + \sum_{p=1}^{P}\left[-\frac{1}{2}\log\sigma^2 - \frac{\beta_p^2}{2\sigma^2}\right] + \left(-\frac{\nu}{2} - 1\right)\log\sigma^2 - \frac{\nu S^2}{2\sigma^2} + Const.$$

The pseudo posterior distribution of β_p is a normal distribution with

$$E\left[\beta_p\right] = H_p E[\tau_0^2]\sum_{i=1}^{N} x_{ip}\left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k\neq p}^{P} E[\beta_k]x_{ik}\right)$$

and

$$V\left[\beta_p\right] = H_p$$

where

$$H_p^{-1} = E[\tau_0^2]\sum_{i=1}^{N} x_{ip}^2 + \frac{1}{\tilde{S}}.$$

Herein,

$$\tilde{S}^2 = \frac{vS^2 + \sum_{j=1}^{P} E\left[\beta_j^2\right]}{\tilde{v}}$$

and

$$\tilde{v} = v + \sum_{j=1}^{P} E\left[\rho_j\right].$$

The pseudo posterior distribution of σ^2 is a scaled inverse-chi-squared distribution with

$$E\left[\frac{1}{\sigma^2}\right] = \frac{1}{\tilde{S}^2}.$$

2-3-7. BLUP

The joint log posterior distribution is

$$\frac{N}{2}\log\tau_0^2 - \frac{\tau_0^2}{2}\sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m) - u_i\right)^2$$

$$-\log\tau_0^2 - \frac{N}{2}\log\sigma^2 - \frac{\log|\mathbf{K}|}{2} - \frac{\mathbf{u}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{u}}{2\sigma^2} + \left(-\frac{\nu}{2} - 1\right)\log\sigma^2 - \frac{\nu S^2}{2\sigma^2} + Const.$$

The pseudo posterior distribution of **u** is a normal distribution with

$$\mathrm{E}[\mathbf{u}] = \mathbf{C}^{-1}\left(\mathbf{y} - \sum_{m=1}^{M-1} f_m(\mathbf{x}, \quad \boldsymbol{\theta}_m)\right)$$

and

$$\mathrm{V}[\mathbf{u}] = \mathbf{C}^{-1}\frac{1}{\mathrm{E}[\tau_0^2]}$$

where

$$\mathbf{C}^{-1} = \left(\mathbf{I} + \mathbf{K}^{-1}\frac{1}{\mathrm{E}[\tau_0^2]}\mathrm{E}\left[\frac{1}{\sigma^2}\right]\right)^{-1}.$$

The pseudo posterior distribution of σ^2 is a scaled inverse-chi-squared distribution with

$$E\left[\frac{1}{\sigma^2}\right] = \frac{a_1}{b_1}$$

where

$$a_1 = v + N$$

and

$$b_1 = \nu S^2 + \sum_{i=1}^{N}\sum_{j=1}^{N} K'_{ij}E[u_i]E[u_j] + \sum_{i=1}^{N}\sum_{j=1}^{N} K'_{ij}cov[u_i, u_j]$$

where K'_{ij} denotes the *ij* element in the inverse of **K** and $cov[u_i, u_j]$ denotes the *ij* element in $\mathrm{V}[\mathbf{u}]$.

2-3-8. Fixed effects (FIXED)

The joint log posterior distribution is

$$\frac{N}{2}\log\tau_0^2 - \frac{\tau_0^2}{2}\sum_{i=1}^{N}\left(y_i - \sum_{m=1}^{M-1} f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m) - \sum_{p=1}^{P} x_{ip}\beta_p\right)^2 - \log\tau_0^2 + Const.$$

The pseudo posterior distribution of β_p is a normal distribution with

$$E[\beta_p] = H_p E[\tau_0^2] \sum_{i=1}^{N} x_{ip} \left(y_i - \sum_{m=1}^{M-1} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] - \sum_{k \neq p}^{P} E[\beta_k] x_{ik} \right)$$

and

$$V[\beta_p] = H_p$$

where

$$H_p^{-1} = E[\tau_0^2] \sum_{i=1}^{N} x_{ip}^2.$$

2-3-9. Update of residual precision

The pseudo posterior of τ_0^2 is a gamma distribution with

$$E[\tau_0^2] = \frac{a_1}{b_1}$$

where

$$a_1 = \frac{N}{2}$$

and

$$b_1 = \frac{1}{2} \left\{ \sum_{i=1}^{N} \left(y_i - \sum_{m=1}^{M} E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] \right)^2 + \sum_{m=1}^{M} V[f_m(\mathbf{x}, \quad \boldsymbol{\theta}_m)] \right\}.$$

$E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)]$ and $V[f_m(\mathbf{x}, \quad \boldsymbol{\theta}_m)]$ vary depending on regression methods. When method *m* is BL or EBL,

$$E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] = \sum_{p=1}^{P} E[\beta_p] x_{ip}$$

and

$$V[f_m(\mathbf{x}, \quad \boldsymbol{\theta}_m)] = \sum_{p=1}^{P} V[\beta_p] \sum_{i=1}^{N} x_{ip}^2 + \sum_{p=1}^{P} E[\tau_p^2] E[\beta_p^2],$$

respectively. When method *m* is BayesA, BayesB, BayesC, BRR, or FIXED,

$$E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] = \sum_{p=1}^{P} E[\beta_p] x_{ip}$$

and

$$V[f_m(\mathbf{x}, \quad \boldsymbol{\theta}_m)] = \sum_{p=1}^{P} V[\beta_p] \sum_{i=1}^{N} x_{ip}^2,$$

respectively. Note that, when method *m* is BayesB and BayesC,

$$E[\beta_p] = E[\beta_p|\rho_p = 1]E[\rho_p]$$

and

$$V[\beta_p] = E[\beta_p^2] - E[\beta_p]^2.$$

When method m is BLUP,

$$E[f_m(\mathbf{x}_i, \quad \boldsymbol{\theta}_m)] = E[u_i]$$

and

$$V[f_m(\mathbf{x}, \quad \boldsymbol{\theta}_m)] = \sum_{i=1}^{N} V[u_i],$$

respectively.

2-4. Lower bound of marginal log likelihood

The moments of the pseudo posterior distributions are iteratively updated from the initial values until convergence. The iterative update monotonically increases the lower bound of the marginal log likelihood of data, p(***y***). From Eq. 2-1, the lower bound of p(***y***) can be written as

$$\int q(\theta) \log p\,(Y, \theta) d\theta - \int q(\theta) \log q\,(\theta) d\theta$$

The first term denotes the expectation of the joint log posterior distribution with regard to the pseudo posterior distributions. The second term denotes the expectation of the log pseudo posterior distributions with regard to the pseudo posterior distributions. *vigor* calculates the lower bound only at the last iteration to reduce computational time. **Note that the lower bound is calculated with standardized Y.**

2-5. Model fitting

2-5-1. Standardization

All of the regression methods standardize the response variables to a mean and standard deviation of 0 and 1, respectively. Two aspects of the standardization should be noted:

1) in the CV, **only the response values used for training are standardized at each fold**. Testing individuals are excluded from the standardization.
2) VIGoR outputs most of the estimated parameter values on the original scale. **However, the lower bound of the marginal log likelihood is returned in the standardized scale.**
3) **VIGoR does nothing to explanatory variables.**

2-5-2. Randomization

In the default setting, VIGoR starts analyses from the same initial values but randomizes the update order of SNP effects. **Thus, the results might change across runs**. The initial values can also be randomized.

2-5-3. Intercept

VIGoR automatically adds the intercept as the last learner when no fixed effect is added as FIXED by user. **When fixed effects are added as FIXED by user, the intercept should be incorporated in FIXED.**

3. Hyperparameters

3-1. Hyperparameters determined by *hyperpara*

Choosing the hyperparameter values is often problematic in Bayesian computation. Although VIGoR provides default hyperparameter values, users may want to specify the values themselves. The function *hyperpara* will be useful in such case. *hyperpara* determines hyperparameter values based on two assumptions about the genetic architecture (Habier et al. 2011). *hyperpara* determines the more influential hyperparameters from these assumptions and values of less influential hyperparameters (Table 3-1). **Note that hyperpara is not applicable to BLUP** because BLUP is relatively insensitive to hyperparameters.

Table 3-1 Hyperparameters determined by *hyperpara*

Methods	Less influential hyperparameters (values given as default)	Influential hyperparameters determined by *hyperpara*
BL	φ (1.0)	ω
EBL	φ (0.1), ω (0.1), ψ (1.0)	θ
BayesA	ν (5.0)	S^2
BayesB	ν (5.0)	S^2, κ
BayesC	ν (5.0)	S^2, κ
BRR	ν (5.0)	S^2

The required two assumptions are:

- κ : proportion of SNPs (explanatory variables) with **NON-ZERO EFFECTS**.
- *Mvar* : proportion of phenotypic variance (variance of response variables) explainable by SNPs (explanatory variables).

For example, when and *Mvar* are respectively set to 0.01 and 0.5, a half of the phenotypic variance is assumed to be explained by 1 % of SNPs. When multiple regression methods are included in a model, relative contributions of the methods would be controllable by assigning different values to *Mvar*.

3-2. Calculation by *hyperpara*

Suppose that the phenotypic variance is standardized such that the mean and variance is 0 and 1, respectively, as done by *vigor*. When the effect variance of SNP j is σ_j^2, the proportion of SNPs with non-zero effects is κ, and linkage equilibrium is assumed, *Mvar* can be represented as

$$Mvar = \kappa \sum_{j=1}^{P} \sigma_j^2 2(1+f) p_j (1-p_j)$$

(**Eq. 3-1**)

where *f* is the inbreeding coefficient, and p_j is the allele frequency for the SNP *j* (Habier *et al*. 2011). When the explanatory variables are not SNPs, $2(1+f)p_j(1-p_j)$ is replaced by the sample variance of the variable *j*. For BL, because σ_j^2 can be written as $\frac{1}{\tau_j^2}(1-Mvar)$ (Table 2-2), we obtain

$$Mvar = \kappa \sum_{j=1}^{P} \frac{1}{\tau_j^2}(1-Mvar)2(1+f)p_j(1-p_j)$$

(**Eq. 3-2**)

The expectations of SNP effect variance and λ^2 are $E\left[\frac{1}{\tau_j^2}\right] = \frac{2}{\lambda^2}$ and $\frac{\varphi}{\omega}$, respectively. By plugging these expectations into Eq. 3-2 and solving the equation with regard to ω, we obtain

$$\varpi = \frac{\varphi}{4\kappa(1+f)\sum_j^P p_j(1-p_j)\left(\frac{1}{Mvar}-1\right)}$$

Thus, we can determine ω if we give values for φ and κ. For EBL, with a similar approach, we obtain

$$\theta = \frac{\psi\varphi}{4\kappa\varpi(1+f)\sum_j^P p_j(1-p_j)\left(\frac{1}{Mvar}-1\right)}$$

For BayesB and BayesC, because the expectation of SNP effect variance is $\frac{\nu S^2}{(\nu-2)}$, by plugging this into Eq. 3-1, we obtain

$$S^2 = \frac{(\nu-2)Mvar}{\nu\kappa(1+f)\sum_{j=1}^P 2p_j(1-p_j)}$$

For BayesA and BRR, because *κ* is 1, we obtain

$$S^2 = \frac{(\nu-2)Mvar}{\nu(1+f)\sum_{j=1}^P 2p_j(1-p_j)}$$

4. R function *vigor*

The manual of the R function *vigor* is also provided as R documentation. To view this manual, type ?vigor on the R console. The usage of *vigor* is

```
vigor (Y,  ETA,  Function = c("fitting", "tuning", "cv"),
        Nfold = 5,  CVFoldTuning = 5,  Partition = NULL,
        Thresholdvalue = 1e-5,  Maxiteration = 1000,  RandomIni = TRUE,
        Metrics = c("rmse", "cor"),  Verbose = TRUE)
```

The first two arguments are mandatory; the remaining arguments are optional.

4-1. Mandatory arguments

The two mandatory arguments of the R function *vigor* are *Y* and *ETA*.

Arguments

Y

Y is a vector of phenotypic values (response variables). Its length is the number of samples (*N*). Missing values (coded as NA) are allowed.

ETA

A nested list to specify regression methods, explanatory variables, and hyperparameters. The length of ETA is the number of methods (learners) incorporated in a single model. Each element (list) of ETA consists of the following objects.

- model : One of strings representing regression methods, “BL”, “EBL”, “BayesA”, “BayesB”, “BayesC”, “BRR”, “BLUP”, or “FIXED”
- *X* : An explanatory variables (e.g., SNP genotypes) of (*N* x *P*) matrix, where *N* and *P* denote the number of samples and variables, respectively
- *K* : An *N* x *N* kernel matrix (e.g., genomic relationship matrix) used when BLUP is specified as model. When BLUP is specified but K is lacked, the linear kernel is created from X as scale(X)*t(scale(X))/ncol(X).
- *H* : A vector or matrix including hyperparameters
- *data* : A data frame containing fixed effects. Used to model FIXED using formula.

Specification of model is essential for all methods. For regression methods except for BLUP and FIXED, *X* is essential. For BLUP, either *X* or *K* is essential. For FIXED, either *X* or formula with data is essential. *H* is a vector when a single hyperparameter set is specified or a matrix when multiple hyperparameter sets are specified. In the vector or each row of the matrix, **hyperparameter values should be ordered as indicated in Table 2-3.**

4-2. Optional arguments

Arguments

Function

Specifies the functions of *vigor*.

- “fitting” : model fitting
- “tuning” : model fitting after hyperparameter tuning
- “cv” : cross-validation

The default function is “fitting”.

Nfold

Fold number of CV. This argument is used with the “cv” function.

- *n* (*n*>1) : *n*-fold cross-validation with randomly partitioned individuals
- −1 : leave-one-out CV

Nfold is ignored when *Partition* is specified. The default setting of *Nfold* is 5.

CVFoldTuning

The cross-validation for tuning the hyperparameters requires an integer fold number. The *CVFoldTuning* argument is used with the “tuning” and “cv” functions. The default is 5.

Partition

The *Partition* matrix defines the partitioning of individuals in CV and is used in the “cv” function. ***Partition* specifies the individuals to be predicted (i.e., the individuals that are not used for training) at each fold**.

Ex. 1). The following matrix is the *Partition* matrix of 19 individuals in a five-fold CV.

16	5	17	13	9
12	18	3	14	6
8	7	11	15	19
1	10	2	4	−9

The matrix columns specify the test individuals at each fold. The elements correspond to the row numbers of *X* and the vector indices of *Y*. In this example, individuals 16, 12, 8, and 1 are excluded from training and predicted at the first fold. Individuals 5, 18, 7, and 10 are excluded and predicted at the second fold. Spaces in the matrix are filled with “−9” (note the missing fourth individual at the fifth fold).

Ex. 2). *Partition* can be employed in random sampling, which may sample some individuals more than once. For example, consider 19 individuals and split five times, with four individuals tested at each split. The corresponding *Partition* matrix is

18	3	11	16	13
17	8	13	13	18
7	15	14	19	7
1	13	12	7	2

Herein, individuals 18, 13, and 7 are repeatedly selected for testing.

Threshold

This variable is the convergence threshold. Smaller values indicate stricter thresholds. The convergence criterion is given as *Th* in **2-3. Update procedures**. The default is 1e−5.

Maxiteration

Maximum number of iterations. The default is 1000.

RandomIni

If TRUE, the initial values are randomized. The default is FALSE.

Metrics

Metrics used in tuning.

- “rmse” : root mean squared error
- “cor” : Pearson correlation

Verbose

If TRUE (the default condition), the run information is printed to the console.

4-3. Output lists

Vigor has two output list formats one for ***Model fitting*** and ***Model fitting after hyperparameter tuning***, the other for ***cross-validation***.

4-3-1. The output list of “fitting” or “tuning”

LB

Lower bound of the marginal log likelihood of *Y*.

ResidualVar

Residual variances ($\frac{1}{\tau_0^2}$) at each iteration (from start to end).

H

Used hyperparameters.

Fittedvalue

Fitted values

Metrics

Metrics for hyperparameter tuning. Returned when Function = "tuning".

ETA

A list containing results for each regression method.

Beta : Posterior means of regression coefficients of X

Sd.beta : Posterior standard deviations of *Beta* (uncertainty of *Beta*)

Sigma2 : Posterior means of variance of *Beta* or *U*

Rho : Posterior means of model-inclusion probabilities

U : Posterior means of random effects of BLUP

Sd.u : Posterior standard deviations of *U* (uncertainty of *U*)

iK : Inverse of **K**

Beta and *Sd.beta* are returned for BL, EBL, BayesA, BayesB, BayesC, and FIXED, and *U*, *Sd.u*, and *iK* are returned for BLUP. *Rho* is returned for BayesB and BayesC. *Sigma2* is returned for methods except for BL and EBL.

AddIntercept

True when the intercept was added automatically.

4-3-2. The output list of “cv”

Prediction

A vector of predicted values

Metrics

Metrics of hyperparameter tuning. Chosen sets and corresponding metrics at each fold are returned.

Partition

A matrix representing the partition used in random partitioning. This matrix can be used as the argument *Partition* in subsequent analyses.

AddIntercept

True when the intercept was added automatically.

5. R function *hyperpara*

The function *hyperpara* calculates the values of the hyperparameters that influence on the inference, based on two assumptions of the genetic architecture. The calculation equations are given in **Subsection 3-2. Calculation by hyperpara**. The manual of the R function *hyperpara* is also provided as R documentation. To view this manual, type ?hyperpara on the R console. The usage of *hyperpara* is

```
hyperpara(X, Mvar, Model = c("BL", "EBL", "BayesA", "BayesB", "BayesC", "BRR"),
        Kappa = 0.01, Xtype = c("Geno", "Var"), f = 0, BL.Phi = 1, EBL.Phi = 0.1,
        EBL.Omega = 0.1, Psi = 1, Nu = 5, Verbose = FALSE)
```

The first three arguments are mandatory; the remaining arguments are optional.

5-1. Mandatory arguments

The mandatory arguments of the R function *hyperpara* are *X*, *Mvar*, and *Model*.

Arguments

X

An ($N \times P$) matrix, where *N* and *P* denote the number of samples and variables, respectively. When *X* is SNP genotypes as specified by *Xtype* = "Geno", SNP genotypes should be coded with values between 0 and 2. SNP genotypes are used to calculate sum (2**Q**(1-*Q*)*(1+*f*)) where *Q* is a vector of allele frequencies and *f* is the inbreeding coefficient. Missing values in *X* are not allowed. When *X* is "Var"(variables other than SNPs), variances of variables are used instead of sum (2**Q**(1-*Q*)*(1+*f*)).

Mvar

Mvar specifies the assumed proportion of the variance of *Y* that can be explained by *X*. In BL and EBL regression, $0 < Mvar < 1$; in the other methods, $0 < Mvar \leq 1$. *Mvar* can be a vector.

Model

Six regression methods are available.

- BL : Bayesian Lasso
- EBL : Extended Bayesian Lasso
- BayesA : BayesC
- BayesB : BayesB
- BayesC : BayesC
- BRR : Bayesian ridge regression

5-2. Optional arguments

Arguments

Kappa

The assumed proportion of **X with NON-ZERO EFFECTS**. 0 < *Kappa* ≤ 1. *Kappa* can be a vector. For BayesA and BRR, *Kappa* is fixed to 1.

Xtype

Specifies the type of predictor variables. This option is recommended when the predictor variables are not SNP genotypes.

- "Geno" : SNP genotypes
- "Var" : variables other than SNP genotypes

The default *Xtype* is "Geno".

f

Inbreeding coefficient of the genotyped population. When analyzing inbred species, *f* is set to 1. Its default value is 0.

BL.Phi

φ value of BL. Default is 1. *BL.Phi* can be a vector.

EBL.Phi

φ value of EBL. Default is 0.1. *EBL.Phi* can be a vector.

EBL.Omega

ω value of EBL. Default is 0.1. *EBL.Omega* can be a vector.

Psi

ψ value of EBL. Default is 1. *Psi* can be a vector.

Nu

v (Nu) value in the BayesA, BayesB, BayesC, and BRR models. Default is 5. *Nu* can be a vector.

Verbose

If TRUE, the run information is printed to the console. Default is FALSE.

5-3. Output objects

The function *hyperpara* outputs a vector when a single hyperparameter set (combination) is created and a matrix when multiple hyperparameter sets are created.

6. R function *predict_vigor*

The function *predict_vigor* predicts *Y* of new *X* using a training result with *vigor*. The manual of the R function *predict_vigor* is also provided as R documentation. To view this manual, type ?predict_vigor on the R console. The usage of *predict_vigor* is

```
predict_vigor(object, newX)
```

The two arguments are mandatory.

6-1. Mandatory arguments

Arguments

object

Training result of *vigor*. A list object ouput by *vigor*.

newX

A list of X. *newX* contains *X* for each learner. The length and order of learners should be same as those of ETA used for training. Each element of newX is a matrix. When BLUP is used, X is an $n_1 \times n_2$ relationship matrix where n_1 and n_2 are the numbers of samples in test and training data, respectively. When the other methods are used, *X* is an $n_1 \times p$ matrix where *p* is the number of explanatory variables. *p* should be the same as the training data. When the intercept is added automatically in training, the intercept needs not to be included in *newX*.

6-2. Output vector

This function predict *Y* of new data (*newX*). When multiple learners are included in the model, predicted values are calculated for each element of newX, and the summation of all predicted values is returned as a vector (predicted values of each learner are not returned). When contributions of each learner are of interest, add NULL to the elements of newX to be ignored.

7. References

Arakawa A, Taniguchi M, Hayashi T, and Mikawa S, 2016 Variational Bayesian method of estimating variance components. Animal Science Journal, 87: 863-872.

Bishop CM, 2006 Pattern recognition and machine learning. Springer, New York.

Carbonetto P and Stephens M, 2012 Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. Bayesian Analysis, 7: 73-108.

Chhikara RS and Folks JL, 1989 The inverse Gaussian distribution. Marcel Dekker, New York.

Habier D, Fernando RL, Kizilkaya K, and Garrick DJ, 2011 Extension of the Bayesian alphabet for genomic selection. BMC Bioinformatics, 12: 186.

Li Z and Sillanpaa MJ, 2012 Estimation of quantitative trait locus effects with epistasis by variational Bayes algorithms. Genetics, 190: 231-249.

Meuwissen TH, Hayes BJ, and Goddard ME, 2001 Prediction of total genetic value using genome-wide dense marker maps. Genetics, 157: 1819-1829.

Murphy KP, 2012 Machine learning: a probabilistic perspective. MIT press, London.

Mutshinda CM and Sillanpaa MJ, 2010 Extended Bayesian LASSO for multiple quantitative trait loci mapping and unobserved phenotype prediction. Genetics, 186: 1067-1075.

Onogi A and Iwata H, 2015 Documents for VIGoR, https://github.com/Onogi/VIGoR/tree/master/Archive.

Park T and Casella G, 2008 The Bayesian lasso. Journal of the American Statistical Association, 103: 681-686.

Wang CS, Rutledge JJ, Gianola D, 1993 Marginal Inferences About Variance Components in a Mixed Linear Model Using Gibbs Sampling. Genetics, Selection, Evolution, 25: 41–62