A General Framework for Variable Selection in Linear Mixed Models with Applications to Genetic Studies with Structured Populations

Sahir R Bhatnagar[1,2], Karim Oualkacha[3], Yi Yang[4], Marie Forest[2], and Celia MT Greenwood[1,2,5]

[1]Department of Epidemiology, Biostatistics and Occupational Health, McGill University

[2]Lady Davis Institute, Jewish General Hospital, Montréal, QC

[3]Département de Mathématiques, Université de Québec À Montréal

[4]Department of Mathematics and Statistics, McGill University

[5]Departments of Oncology and Human Genetics, McGill University

August 6, 2018

Abstract

Complex traits are known to be influenced by a combination of environmental factors and rare and common genetic variants. However, detection of such multivariate associations can be compromised by low statistical power and confounding by population structure. Linear mixed effect models (LMM) can account for correlations due to relatedness but have not been applicable in high-dimensional (HD) settings where the

number of fixed effect predictors greatly exceeds the number of samples. False positives can result from two-stage approaches, where the residuals estimated from a null model adjusted for the subjects' relationship structure are subsequently used as the response in a standard penalized regression model. To overcome these challenges, we develop a general penalized LMM framework called `ggmix` that simultaneously, in one step, selects variables and estimates their effects, while accounting for between individual correlations. Our method can accommodate several sparsity-inducing penalties such as the lasso, elastic net and group lasso, and also readily handles prior annotation information in the form of weights. We develop a blockwise coordinate descent algorithm which is highly scalable, computationally efficient and has theoretical guarantees of convergence. Through simulations, we show that `ggmix` leads to correct Type 1 error control and improved variance component estimation compared to the two-stage approach or principal component adjustment. `ggmix` is also robust to different kinship structures and heritability proportions. Our algorithms are available in an R package (https://github.com/greenwoodlab).

1 Introduction

Genome-wide association studies (GWAS) have become the standard method for analyzing genetic datasets owing to their success in identifying thousands of genetic variants associated with complex diseases (https://www.genome.gov/gwastudies/). Despite these impressive findings, the discovered markers have only been able to explain a small proportion of the phenotypic variance; this is known as the missing heritability problem [1]. One plausible explanation is that there are many causal variants that each explain a small amount of variation with small effect sizes [2]. Methods such GWAS, which test each variant or single nucleotide polymorphism (SNP) independently, may miss these true associations due to the stringent significance thresholds required to reduce the number of false positives [1]. Another major issue to overcome is that of confounding due to geographic population structure, family

and/or cryptic relatedness which can lead to spurious associations [3]. For example, there may be subpopulations within a study that differ with respect to their genotype frequencies at a particular locus due to geographical location or their ancestry. This heterogeneity in genotype frequency can cause correlations with other loci and consequently mimic the signal of association even though there is no biological association [4, 5]. Studies that separate their sample by ethnicity to address this confounding suffer from a loss in statistical power.

To address the first problem, multivariable regression methods have been proposed which simultaneously fit many SNPs in a single model [6, 7]. Indeed, the power to detect an association for a given SNP may be increased when other causal SNPs have been accounted for. Conversely, a stronger signal from a causal SNP may weaken false signals when modeled jointly [6].

Solutions for confounding by population structure have also received significant attention in the literature [8, 9, 10, 11]. There are two main approaches to account for the relatedness between subjects: 1) the principal component (PC) adjustment method and 2) the linear mixed model (LMM). The PC adjustment method includes the top PCs of genome-wide SNP genotypes as additional covariates in the model [12]. The LMM uses an estimated covariance matrix from the individuals' genotypes and includes this information in the form of a random effect [3].

While these problems have been addressed in isolation, there has been relatively little progress towards addressing them jointly at a large scale. Region-based tests of association have been developed where a linear combination of p variants is regressed on the response variable in a mixed model framework [13]. In case-control data, a stepwise logistic-regression procedure was used to evaluate the relative importance of variants within a small genetic region [14]. These methods however are not applicable in the high-dimensional setting, i.e., when the number of variables p is much larger than the sample size n, as is often the case in genetic studies where millions of variants are measured on thousands of individuals.

There has been recent interest in using penalized linear mixed models, which place a constraint on the magnitude of the effect sizes while controlling for confounding factors such as population structure. For example, the LMM-lasso [15] places a Laplace prior on all main effects while the adaptive mixed lasso [16] uses the L_1 penalty [17] with adaptively chosen weights [18] to allow for differential shrinkage amongst the variables in the model. Another method applied a combination of both the lasso and group lasso penalties in order to select variants within a gene most associated with the response [19]. However, these methods are normally performed in two steps. First, the variance components are estimated once from a LMM with a single random effect. These LMMs normally use the estimated covariance matrix from the individuals' genotypes to account for the relatedness but assumes no SNP main effects (i.e. a null model). The residuals from this null model with a single random effect can be treated as independent observations because the relatedness has been effectively removed from the original response. In the second step, these residuals are used as the response in any high-dimensional model that assumes uncorrelated errors. This approach has both computational and practical advantages since existing penalized regression software such as `glmnet` [20] and `gglasso` [21], which assume independent observations, can be applied directly to the residuals. However, recent work has shown that there can be a loss in power if a causal variant is included in the calculation of the covariance matrix as its effect will have been removed in the first step [13, 22].

In this paper we develop a general penalized LMM framework called `ggmix` that simultaneously selects variables and estimates their effects, accounting for between-individual correlations. Our method can accommodate several sparsity inducing penalties such as the lasso [17], elastic net [23] and group lasso [24]. `ggmix` also readily handles prior annotation information in the form of a penalty factor, which can be useful, for example, when dealing with rare variants. We develop a blockwise coordinate descent algorithm which is highly scalable and has theoretical guarantees of convergence to a stationary point. All of our algorithms are implemented in the `ggmix` R package hosted on GitHub with extensive

documentation (http://sahirbhatnagar.com/ggmix/). We provide a brief demonstration of the `ggmix` package in Appendix C.

The rest of the paper is organized as follows. Section 2 describes the `ggmix` model. Section 3 contains the optimization procedure and the algorithm used to fit the `ggmix` model. In Section 4, we compare the performance of our proposed approach and demonstrate the scenarios where it can be advantageous to use over existing methods through simulation studies. Section 5 discusses some limitations and future directions.

2 Penalized Linear Mixed Models

2.1 Model Set-up

Let $i = 1, \ldots, N$ be a grouping index, $j = 1, \ldots, n_i$ the observation index within a group and $N_T = \sum_{i=1}^{N} n_i$ the total number of observations. For each group let $\boldsymbol{y}_i = (y_1, \ldots, y_{n_i})$ be the observed vector of responses or phenotypes, $\mathbf{X}_i$ an $n_i \times (p+1)$ design matrix (with the column of 1s for the intercept), $\boldsymbol{b}_i$ a group-specific random effect vector of length n_i and $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \ldots, \varepsilon_{in_i})$ the individual error terms. Denote the stacked vectors $\mathbf{Y} = (\boldsymbol{y}_i, \ldots, \boldsymbol{y}_N)^T \in \mathbb{R}^{N_T \times 1}$, $\boldsymbol{b} = (\boldsymbol{b}_i, \ldots, \boldsymbol{b}_N)^T \in \mathbb{R}^{N_T \times 1}$, $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_i, \ldots, \boldsymbol{\varepsilon}_N)^T \in \mathbb{R}^{N_T \times 1}$, and the stacked matrix
$\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)^T \in \mathbb{R}^{N_T \times (p+1)}$. Furthermore, let $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T \in \mathbb{R}^{(p+1) \times 1}$ be a vector of fixed effects regression coefficients corresponding to $\mathbf{X}$. We consider the following linear mixed model with a single random effect [25]:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{b} + \boldsymbol{\varepsilon} \tag{1}$$

where the random effect $\boldsymbol{b}$ and the error variance $\boldsymbol{\varepsilon}$ are assigned the distributions

$$\boldsymbol{b} \sim \mathcal{N}(0, \eta\sigma^2\boldsymbol{\Phi}) \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, (1-\eta)\sigma^2\mathbf{I}) \tag{2}$$

Here, $\boldsymbol{\Phi}_{N_T \times N_T}$ is a known positive semi-definite and symmetric covariance or kinship matrix calculated from SNPs sampled across the genome, $\mathbf{I}_{N_T \times N_T}$ is the identity matrix and parameters σ^2 and $\eta \in [0, 1]$ determine how the variance is divided between $\boldsymbol{b}$ and $\boldsymbol{\varepsilon}$. Note that η is also the narrow-sense heritability (h^2), defined as the proportion of phenotypic variance attributable to the additive genetic factors [1]. The joint density of $\mathbf{Y}$ is therefore multivariate normal:

$$\mathbf{Y}|(\boldsymbol{\beta}, \eta, \sigma^2) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \eta\sigma^2\boldsymbol{\Phi} + (1-\eta)\sigma^2\mathbf{I}) \tag{3}$$

The LMM-Lasso method [15] considers an alternative but equivalent parameterization given by:

$$\mathbf{Y}|(\boldsymbol{\beta}, \delta, \sigma_g^2) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma_g^2(\boldsymbol{\Phi} + \delta\mathbf{I})) \tag{4}$$

where $\delta = \sigma_e^2/\sigma_g^2$, σ_g^2 is the genetic variance and σ_e^2 is the residual variance. We instead consider the parameterization in (3) since maximization is easier over the compact set $\eta \in [0, 1]$ than over the unbounded interval $\delta \in [0, \infty)$ [25]. We define the complete parameter vector as $\boldsymbol{\Theta} := (\boldsymbol{\beta}, \eta, \sigma^2)$. The negative log-likelihood for (3) is given by

$$-\ell(\boldsymbol{\Theta}) \propto \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\log\left(\det(\mathbf{V})\right) + \frac{1}{2\sigma^2}\left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right)^T \mathbf{V}^{-1}\left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right) \tag{5}$$

where $\mathbf{V} = \eta\boldsymbol{\Phi} + (1-\eta)\mathbf{I}$ and $\det(\mathbf{V})$ is the determinant of $\mathbf{V}$.

Let $\boldsymbol{\Phi} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ be the eigen (spectral) decomposition of the kinship matrix $\boldsymbol{\Phi}$, where $\mathbf{U}_{N_T \times N_T}$ is an orthonormal matrix of eigenvectors (i.e. $\mathbf{U}\mathbf{U}^T = \mathbf{I}$) and $\mathbf{D}_{N_T \times N_T}$ is a diagonal

matrix of eigenvalues Λ_i. $\mathbf{V}$ can then be further simplified [25]

$$\begin{aligned}
\mathbf{V} &= \eta\mathbf{\Phi} + (1-\eta)\mathbf{I} \\
&= \eta\mathbf{U}\mathbf{D}\mathbf{U}^T + (1-\eta)\mathbf{U}\mathbf{I}\mathbf{U}^T \\
&= \mathbf{U}\eta\mathbf{D}\mathbf{U}^T + \mathbf{U}(1-\eta)\mathbf{I}\mathbf{U}^T \\
&= \mathbf{U}\left(\eta\mathbf{D} + (1-\eta)\mathbf{I}\right)\mathbf{U}^T \\
&= \mathbf{U}\widetilde{\mathbf{D}}\mathbf{U}^T
\end{aligned} \tag{6}$$

where

$$\widetilde{\mathbf{D}} = \eta\mathbf{D} + (1-\eta)\mathbf{I} \tag{7}$$

$$\begin{aligned}
&= \eta \begin{bmatrix} \Lambda_1 & & & \\ & \Lambda_2 & & \\ & & \ddots & \\ & & & \Lambda_{N_T} \end{bmatrix} + (1-\eta) \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1+\eta(\Lambda_1-1) & & & \\ & 1+\eta(\Lambda_2-1) & & \\ & & \ddots & \\ & & & 1+\eta(\Lambda_{N_T}-1) \end{bmatrix} \\
&= \operatorname{diag}\left\{1+\eta(\Lambda_1-1), 1+\eta(\Lambda_2-1), \ldots, 1+\eta(\Lambda_{N_T}-1)\right\}
\end{aligned} \tag{8}$$

Since (7) is a diagonal matrix, its inverse is also a diagonal matrix:

$$\widetilde{\mathbf{D}}^{-1} = \operatorname{diag}\left\{\frac{1}{1+\eta(\Lambda_1-1)}, \frac{1}{1+\eta(\Lambda_2-1)}, \ldots, \frac{1}{1+\eta(\Lambda_{N_T}-1)}\right\} \tag{9}$$

From (6) and (8), $\log(\det(\mathbf{V}))$ simplifies to

$$\begin{aligned}\log(\det(\mathbf{V})) &= \log\left(\det(\mathbf{U})\det\left(\widetilde{\mathbf{D}}\right)\det(\mathbf{U}^T)\right)\\ &= \log\left\{\prod_{i=1}^{N_T}(1+\eta(\Lambda_i-1))\right\}\\ &= \sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) \qquad (10)\end{aligned}$$

since $\det(\mathbf{U}) = 1$. It also follows from (6) that

$$\begin{aligned}\mathbf{V}^{-1} &= \left(\mathbf{U}\widetilde{\mathbf{D}}\mathbf{U}^T\right)^{-1}\\ &= \left(\mathbf{U}^T\right)^{-1}\left(\widetilde{\mathbf{D}}\right)^{-1}\mathbf{U}^{-1}\\ &= \mathbf{U}\widetilde{\mathbf{D}}^{-1}\mathbf{U}^T \qquad (11)\end{aligned}$$

since for an orthonormal matrix $\mathbf{U}^{-1} = \mathbf{U}^T$. Substituting (9), (10) and (11) into (5) the negative log-likelihood becomes

$$-\ell(\boldsymbol{\Theta}) \propto \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) + \frac{1}{2\sigma^2}\left(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\right)^T\mathbf{U}\widetilde{\mathbf{D}}^{-1}\mathbf{U}^T\left(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\right) \qquad (12)$$

$$\begin{aligned}&= \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) + \frac{1}{2\sigma^2}\left(\mathbf{U}^T\mathbf{Y}-\mathbf{U}^T\mathbf{X}\boldsymbol{\beta}\right)^T\widetilde{\mathbf{D}}^{-1}\left(\mathbf{U}^T\mathbf{Y}-\mathbf{U}^T\mathbf{X}\boldsymbol{\beta}\right)\\ &= \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) + \frac{1}{2\sigma^2}\left(\widetilde{\mathbf{Y}}-\widetilde{\mathbf{X}}\boldsymbol{\beta}\right)^T\widetilde{\mathbf{D}}^{-1}\left(\widetilde{\mathbf{Y}}-\widetilde{\mathbf{X}}\boldsymbol{\beta}\right)\\ &= \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) + \frac{1}{2\sigma^2}\sum_{i=1}^{N_T}\frac{\left(\widetilde{Y}_i-\sum_{j=0}^{p}\widetilde{X}_{ij+1}\beta_j\right)^2}{1+\eta(\Lambda_i-1)} \qquad (13)\end{aligned}$$

where $\widetilde{\mathbf{Y}} = \mathbf{U}^T\mathbf{Y}$, $\widetilde{\mathbf{X}} = \mathbf{U}^T\mathbf{X}$, $\widetilde{Y}_i$ denotes the ith element of $\widetilde{\mathbf{Y}}$, $\widetilde{X}_{ij}$ is the i,j^{th} entry of $\widetilde{\mathbf{X}}$ and $\mathbf{1}$ is a column vector of N_T ones.

2.2 Penalized Maximum Likelihood Estimator

We define the $p+3$ length vector of parameters $\boldsymbol{\Theta} := (\Theta_0, \Theta_1, \ldots, \Theta_{p+1}, \Theta_{p+2}, \Theta_{p+3}) = (\boldsymbol{\beta}, \eta, \sigma^2)$ where $\boldsymbol{\beta} \in \mathbb{R}^{p+1}, \eta \in [0,1], \sigma^2 > 0$. In what follows, $p+2$ and $p+3$ are the indices in $\boldsymbol{\Theta}$ for η and σ^2, respectively. In light of our goals to select variables associated with the response in high-dimensional data, we propose to place a constraint on the magnitude of the regression coefficients. This can be achieved by adding a penalty term to the likelihood function (13). The penalty term is a necessary constraint because in our applications, the sample size is much smaller than the number of predictors. We define the following objective function:

$$Q_\lambda(\boldsymbol{\Theta}) = f(\boldsymbol{\Theta}) + \lambda \sum_{j \neq 0} v_j P_j(\beta_j) \tag{14}$$

where $f(\boldsymbol{\Theta}) := -\ell(\boldsymbol{\Theta})$ is defined in (13), $P_j(\cdot)$ is a penalty term on the fixed regression coefficients $\beta_1, \ldots, \beta_{p+1}$ (we do not penalize the intercept) controlled by the nonnegative regularization parameter λ, and v_j is the penalty factor for jth covariate. These penalty factors serve as a way of allowing parameters to be penalized differently. Note that we do not penalize η or σ^2. An estimate of the regression parameters $\widehat{\boldsymbol{\Theta}}_\lambda$ is obtained by

$$\widehat{\boldsymbol{\Theta}}_\lambda = \arg\min_{\boldsymbol{\Theta}} Q_\lambda(\boldsymbol{\Theta}) \tag{15}$$

This is the general set-up for our model. In Section 3 we provide more specific details on how we solve (15).

3 Computational Algorithm

We use a general purpose block coordinate gradient descent algorithm (CGD) [26] to solve (15). At each iteration, we cycle through the coordinates and minimize the objective function with respect to one coordinate only. For continuously differentiable $f(\cdot)$ and convex and block-

separable $P(\cdot)$ (i.e. $P(\boldsymbol{\beta}) = \sum_i P_i(\beta_i)$), Tseng and Yun [26] show that the solution generated by the CGD method is a stationary point of $Q_\lambda(\cdot)$ if the coordinates are updated in a Gauss-Seidel manner i.e. $Q_\lambda(\cdot)$ is minimized with respect to one parameter while holding all others fixed. The CGD algorithm has been successfully applied in fixed effects models (e.g. [27], [20]) and linear mixed models with an ℓ_1 penalty [28]. In the next section we provide some brief details about Algorithm 1. A more thorough treatment of the algorithm is given in Appendix A.

We emphasize here that previously developed methods such as the LMM-lasso [15] use a two-stage fitting procedure without any convergence details. From a practical point of view, there is currently no implementation that provides a principled way of determining the sequence of tuning parameters to fit, nor a procedure that automatically selects the optimal value of λ. To our knowledge, we are the first to develop a CGD algorithm in the specific context of fitting a penalized LMM for population structure correction with theoretical guarantees of convergence. Furthermore, we develop a principled method for automatic tuning parameter selection and provide an easy-to-use software implementation in order to promote wider uptake of these more complex methods by applied practitioners.

Algorithm 1: Block Coordinate Gradient Descent

Set the iteration counter $k \leftarrow 0$, initial values for the parameter vector $\boldsymbol{\Theta}^{(0)}$ and convergence threshold ϵ;

for $\lambda \in \{\lambda_{max}, \ldots, \lambda_{min}\}$ **do**

  repeat

$$\boldsymbol{\beta}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{\beta}} Q_\lambda\left(\boldsymbol{\beta}, \eta^{(k)}, \sigma^{2\ (k)}\right)$$

$$\eta^{(k+1)} \leftarrow \arg\min_{\eta} Q_\lambda\left(\boldsymbol{\beta}^{(k+1)}, \eta, \sigma^{2\ (k)}\right)$$

$$\sigma^{2\ (k+1)} \leftarrow \arg\min_{\sigma^2} Q_\lambda\left(\boldsymbol{\beta}^{(k+1)}, \eta^{(k+1)}, \sigma^2\right)$$

    $k \leftarrow k+1$

  until *convergence criterion is satisfied:* $\left\|\boldsymbol{\Theta}^{(k+1)} - \boldsymbol{\Theta}^{(k)}\right\|_2 < \epsilon$;

end

3.1 Updates for the β parameter

Recall that the part of the objective function that depends on $\boldsymbol{\beta}$ has the form

$$Q_\lambda(\boldsymbol{\Theta}) = \frac{1}{2}\sum_{i=1}^{N_T} w_i \left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} v_j|\beta_j| \tag{16}$$

where

$$w_i := \frac{1}{\sigma^2\left(1+\eta(\Lambda_i - 1)\right)} \tag{17}$$

Conditional on $\eta^{(k)}$ and $\sigma^{2\,(k)}$, it can be shown that the solution for β_j, $j = 1, \ldots, p$ is given by

$$\beta_j^{(k+1)} \leftarrow \frac{\mathcal{S}_\lambda\left(\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right)\right)}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2} \tag{18}$$

where $\mathcal{S}_\lambda(x)$ is the soft-thresholding operator

$$\mathcal{S}_\lambda(x) = \operatorname{sign}(x)(|x| - \lambda)_+$$

$\operatorname{sign}(x)$ is the signum function

$$\operatorname{sign}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

and $(x)_+ = \max(x, 0)$. We provide the full derivation in Appendix A.1.2.

3.2 Updates for the η paramter

Given $\boldsymbol{\beta}^{(k+1)}$ and $\sigma^{2\,(k)}$, solving for $\eta^{(k+1)}$ becomes a univariate optimization problem:

$$\eta^{(k+1)} \leftarrow \arg\min_{\eta} \frac{1}{2}\sum_{i=1}^{N_T} \log(1+\eta(\Lambda_i-1)) + \frac{1}{2\sigma^{2\,(k)}}\sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j^{(k+1)}\right)^2}{1+\eta(\Lambda_i-1)} \tag{19}$$

We use a bound constrained optimization algorithm [29] implemented in the `optim` function in `R` and set the lower and upper bounds to be 0.01 and 0.99, respectively.

3.3 Updates for the σ^2 parameter

Conditional on $\boldsymbol{\beta}^{(k+1)}$ and $\eta^{(k+1)}$, $\sigma^{2\,(k+1)}$ can be solved for using the following equation:

$$\sigma^{2\,(k+1)} \leftarrow \arg\min_{\sigma^2} \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j\right)^2}{1+\eta(\Lambda_i-1)} \tag{20}$$

There exists an analytic solution for (20) given by:

$$\sigma^{2\,(k+1)} \leftarrow \frac{1}{N_T}\sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j^{(k+1)}\right)^2}{1+\eta^{(k+1)}(\Lambda_i-1)} \tag{21}$$

3.4 Regularization path

In this section we describe how determine the sequence of tuning parameters λ at which to fit the model. Recall that our objective function has the form

$$Q_\lambda(\boldsymbol{\Theta}) = \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) + \frac{1}{2}\sum_{i=1}^{N_T} w_i\left(\widetilde{Y}_i - \sum_{j=0}^{p}\widetilde{X}_{ij+1}\beta_j\right)^2 + \lambda\sum_{j=1}^{p} v_j|\beta_j| \tag{22}$$

The Karush-Kuhn-Tucker (KKT) optimality conditions for (22) are given by:

$$\begin{aligned}
\frac{\partial}{\partial \beta_1, \ldots, \beta_p} Q_\lambda(\boldsymbol{\Theta}) &= \mathbf{0}_p \\
\frac{\partial}{\partial \beta_0} Q_\lambda(\boldsymbol{\Theta}) &= 0 \\
\frac{\partial}{\partial \eta} Q_\lambda(\boldsymbol{\Theta}) &= 0 \\
\frac{\partial}{\partial \sigma^2} Q_\lambda(\boldsymbol{\Theta}) &= 0
\end{aligned} \tag{23}$$

The equations in (23) are equivalent to

$$\begin{aligned}
\sum_{i=1}^{N_T} w_i \widetilde{X}_{i1} \left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1} \beta_j \right) &= 0 \\
\frac{1}{v_j} \sum_{i=1}^{N_T} w_i \widetilde{X}_{ij} \left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1} \beta_j \right) &= \lambda \gamma_j, \\
\gamma_j \in \begin{cases} \text{sign}(\hat{\beta}_j) & \text{if} \quad \hat{\beta}_j \neq 0 \\ [-1, 1] & \text{if} \quad \hat{\beta}_j = 0 \end{cases}, & \qquad \text{for } j = 1, \ldots, p \\
\frac{1}{2} \sum_{i=1}^{N_T} \frac{\Lambda_i - 1}{1 + \eta(\Lambda_i - 1)} \left(1 - \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1} \beta_j \right)^2}{\sigma^2 (1 + \eta(\Lambda_i - 1))} \right) &= 0 \\
\sigma^2 - \frac{1}{N_T} \sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1} \beta_j \right)^2}{1 + \eta(\Lambda_i - 1)} &= 0
\end{aligned} \tag{24}$$

where w_i is given by (17), $\widetilde{\mathbf{X}}_{-1}^T$ is $\widetilde{\mathbf{X}}^T$ with the first column removed, $\widetilde{\mathbf{X}}_1^T$ is the first column of $\widetilde{\mathbf{X}}^T$, and $\boldsymbol{\gamma} \in \mathbb{R}^p$ is the subgradient function of the ℓ_1 norm evaluated at $(\hat{\beta}_1, \ldots, \hat{\beta}_p)$. Therefore $\widehat{\boldsymbol{\Theta}}$ is a solution in (15) if and only if $\widehat{\boldsymbol{\Theta}}$ satisfies (24) for some γ. We can determine a decreasing sequence of tuning parameters by starting at a maximal value for $\lambda = \lambda_{max}$ for which $\hat{\beta}_j = 0$ for $j = 1, \ldots, p$. In this case, the KKT conditions in (24) are equivalent

to

$$
\begin{aligned}
\frac{1}{v_j}\sum_{i=1}^{N_T}\left|w_i\widetilde{X}_{ij}\left(\widetilde{Y}_i-\widetilde{X}_{i1}\beta_0\right)\right| &\leq \lambda, \quad \forall j=1,\ldots,p \\
\beta_0 &= \frac{\sum_{i=1}^{N_T} w_i\widetilde{X}_{i1}\widetilde{Y}_i}{\sum_{i=1}^{N_T} w_i\widetilde{X}_{i1}^2} \\
\frac{1}{2}\sum_{i=1}^{N_T}\frac{\Lambda_i-1}{1+\eta(\Lambda_i-1)}\left(1-\frac{\left(\widetilde{Y}_i-\widetilde{X}_{i1}\beta_0\right)^2}{\sigma^2(1+\eta(\Lambda_i-1))}\right) &= 0 \\
\sigma^2 &= \frac{1}{N_T}\sum_{i=1}^{N_T}\frac{\left(\widetilde{Y}_i-\widetilde{X}_{i1}\beta_0\right)^2}{1+\eta(\Lambda_i-1)}
\end{aligned}
\tag{25}
$$

We can solve the KKT system of equations in (25) (with a numerical solution for η) in order to have an explicit form of the stationary point $\widehat{\boldsymbol{\Theta}}_0=\left\{\hat{\beta}_0,\mathbf{0}_p,\hat{\eta},\widehat{\sigma}^2\right\}$. Once we have $\widehat{\boldsymbol{\Theta}}_0$, we can solve for the smallest value of λ such that the entire vector $(\hat{\beta}_1,\ldots,\hat{\beta}_p)$ is 0:

$$
\lambda_{max}=\max_j\left\{\left|\frac{1}{v_j}\sum_{i=1}^{N_T}\hat{w}_i\widetilde{X}_{ij}\left(\widetilde{Y}_i-\widetilde{X}_{i1}\hat{\beta}_0\right)\right|\right\}, \quad j=1,\ldots,p \tag{26}
$$

Following Friedman et al. [20], we choose $\tau\lambda_{max}$ to be the smallest value of tuning parameters λ_{min}, and construct a sequence of K values decreasing from λ_{max} to λ_{min} on the log scale. The defaults are set to $K=100$, $\tau=0.01$ if $n<p$ and $\tau=0.001$ if $n\geq p$.

3.5 Warm Starts

The way in which we have derived the sequence of tuning parameters using the KKT conditions, allows us to implement warm starts. That is, the solution $\widehat{\boldsymbol{\Theta}}$ for λ_k is used as the initial value $\boldsymbol{\Theta}^{(0)}$ for λ_{k+1}. This strategy leads to computational speedups and has been implemented in the `ggmix` R package.

3.6 Prediction of the random effects

We use an empirical Bayes approach (e.g. [30]) to predict the random effects $\boldsymbol{b}$. Let the maximum a posteriori (MAP) estimate be defined as

$$\widehat{\boldsymbol{b}} = \arg\max_{\boldsymbol{b}} f(\boldsymbol{b}|\mathbf{Y}, \boldsymbol{\beta}, \eta, \sigma^2) \tag{27}$$

where, by using Bayes rule, $f(\boldsymbol{b}|\mathbf{Y}, \boldsymbol{\beta}, \eta, \sigma^2)$ can be expressed as

$$\begin{aligned} f(\boldsymbol{b}|\mathbf{Y}, \boldsymbol{\beta}, \eta, \sigma^2) &= \frac{f(\mathbf{Y}|\boldsymbol{b}, \boldsymbol{\beta}, \eta, \sigma^2)\pi(\boldsymbol{b}|\eta, \sigma^2)}{f(\mathbf{Y}|\boldsymbol{\beta}, \eta, \sigma^2)} \\ &\propto f(\mathbf{Y}|\boldsymbol{b}, \boldsymbol{\beta}, \eta, \sigma^2)\pi(\boldsymbol{b}|\eta, \sigma^2) \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b})^T\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b}) - \frac{1}{2\eta\sigma^2}\boldsymbol{b}^T\boldsymbol{\Phi}^{-1}\boldsymbol{b}\right\} \\ &= \exp\left\{-\frac{1}{2\sigma^2}\left[(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b})^T\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b}) + \frac{1}{\eta}\boldsymbol{b}^T\boldsymbol{\Phi}^{-1}\boldsymbol{b}\right]\right\} \end{aligned} \tag{28}$$

Solving for (27) is equivalent to minimizing the exponent in (28):

$$\widehat{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}} \left\{(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b})^T\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b}) + \frac{1}{\eta}\boldsymbol{b}^T\boldsymbol{\Phi}^{-1}\boldsymbol{b}\right\} \tag{29}$$

Taking the derivative of (29) with respect to $\boldsymbol{b}$ and setting it to 0 we get:

$$
\begin{aligned}
0 &= -2\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b}) + \frac{2}{\eta}\boldsymbol{\Phi}^{-1}\boldsymbol{b} \\
&= -\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \left(\mathbf{V}^{-1} + \frac{1}{\eta}\boldsymbol{\Phi}^{-1}\right)\boldsymbol{b} \\
\widehat{\boldsymbol{b}} &= \left(\mathbf{V}^{-1} + \frac{1}{\widehat{\eta}}\boldsymbol{\Phi}^{-1}\right)^{-1}\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) \\
&= \left(\mathbf{U}\widetilde{\mathbf{D}}^{-1}\mathbf{U}^T + \frac{1}{\widehat{\eta}}\mathbf{U}\mathbf{D}^{-1}\mathbf{U}^T\right)^{-1}\mathbf{U}\widetilde{\mathbf{D}}^{-1}\mathbf{U}^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) \\
&= \left(\mathbf{U}\left[\widetilde{\mathbf{D}}^{-1} + \frac{1}{\widehat{\eta}}\mathbf{D}^{-1}\right]\mathbf{U}^T\right)^{-1}\mathbf{U}\widetilde{\mathbf{D}}^{-1}(\widetilde{\mathbf{Y}} - \widetilde{\mathbf{X}}\widehat{\boldsymbol{\beta}}) \\
&= \mathbf{U}\left[\widetilde{\mathbf{D}}^{-1} + \frac{1}{\widehat{\eta}}\mathbf{D}^{-1}\right]^{-1}\mathbf{U}^T\mathbf{U}\widetilde{\mathbf{D}}^{-1}(\widetilde{\mathbf{Y}} - \widetilde{\mathbf{X}}\widehat{\boldsymbol{\beta}})
\end{aligned}
$$

where $\mathbf{V}^{-1}$ is given by (11), and $(\widehat{\boldsymbol{\beta}}, \widehat{\eta})$ are the estimates obtained from Algorithm 1.

3.7 Choice of the optimal tuning parameter

In order to choose the optimal value of the tuning parameter λ, we use the generalized information criterion [31] (GIC):

$$
GIC_\lambda = -2\ell(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}^2, \widehat{\eta}) + a_n \cdot \widehat{df}_\lambda \tag{30}
$$

where $\widehat{df}_\lambda$ is the number of non-zero elements in $\widehat{\boldsymbol{\beta}}_\lambda$ [32] plus two (representing the variance parameters η and σ^2). Several authors have used this criterion for variable selection in mixed models with $a_n = \log N_T$ [28, 33], which corresponds to the BIC. We instead choose the high-dimensional BIC [34] given by $a_n = \log(\log(N_T)) * \log(p)$. This is the default choice in our `ggmix` R package, though the interface is flexible to allow the user to select their choice of a_n.

4 Simulation Study

To assess the performance of `ggmix`, we simulated random genotypes from the BN-PSD admixture model using the `bnpsd` package [35, 36]. We used a block diagonal kinship structure with 5 subpopulations. In Figure 1, we plot an estimated kinship matrix ($\mathbf{\Phi}$), based on a single simulated dataset, in the form of a heatmap. Each block represents a subpopulation, and a darker color indicates a closer genetic relationship.

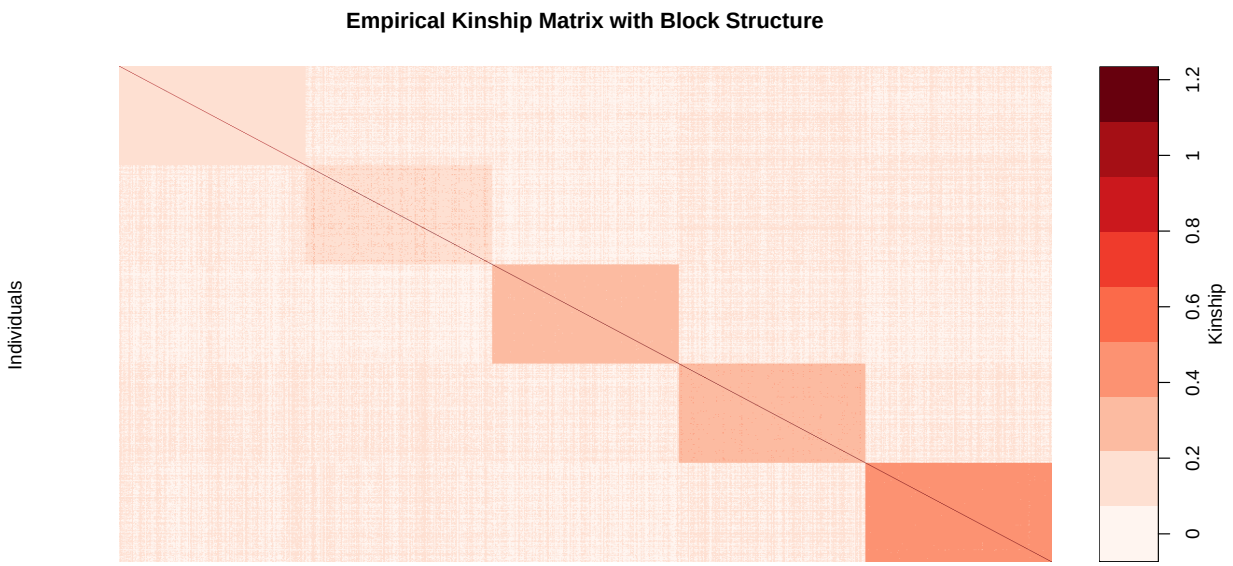


Figure 1: Empirical kinship matrix with block diagnoal structure used in simulation studies. Each block represents a subpopulation.

In Figure 2 we plot the first two principal component scores calculated from the block diagonal kinship matrix in Figure 1, and color each point by subpopulation membership. We can see that the PCs can identify the subpopulations which is why including them as additional covariates in a regression model has been considered a reasonable approach to control for confounding.

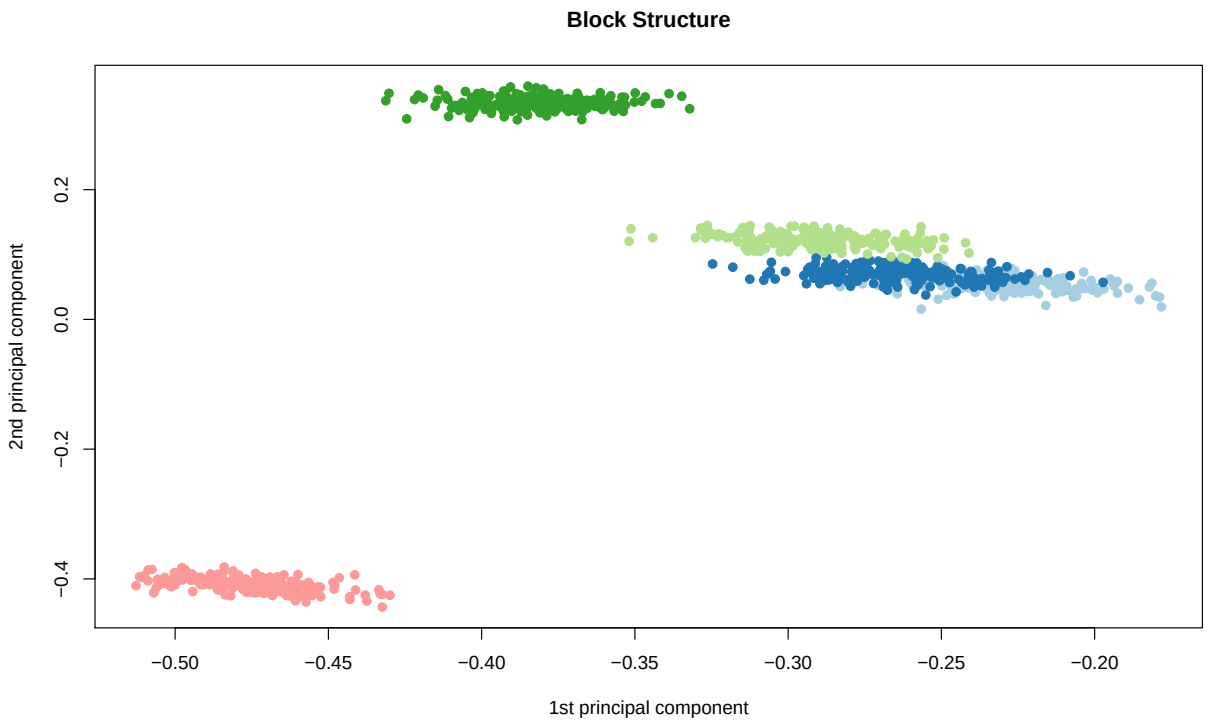


Figure 2: First two principal component scores of the block diagonal kinship matrix where each color represents one of the 5 simulated subpopulations.

For other parameters in our simulation study, we define the following quantities:

- c: percentage of causal SNPs
- $\mathbf{X}^{(fixed)}$: $n \times p_{fixed}$ matrix of SNPs that will be included as fixed effects in our model.
- $\mathbf{X}^{(causal)}$: $n \times (c * p_{fixed})$ matrix of SNPs that will be truly associated with the simulated phenotype, where $\mathbf{X}^{(causal)} \subseteq \mathbf{X}^{(fixed)}$
- $\mathbf{X}^{(other)}$: $n \times p_{other}$ matrix of SNPs that will be used in the construction of the kinship matrix. Some of these $\mathbf{X}^{(other)}$ SNPs, in conjunction with some of the SNPs in $\mathbf{X}^{(test)}$ will be used in construction of the kinship matrix. We will alter the balance between these two contributors and with the proportion of causal SNPs used to calculate kinship.
- $\mathbf{X}^{(kinship)}$: $n \times k$ matrix of SNPs used to construct the kinship matrix.

- β_j: effect size for the jth SNP, simulated from a $Uniform(0.3, 0.7)$ distribution for $j = 1, \ldots, (c * p_{fixed})$

We simulate data from the model

$$\mathbf{Y} = \mathbf{X}^{(fixed)}\boldsymbol{\beta} + \mathbf{P} + \boldsymbol{\varepsilon} \tag{31}$$

where $\mathbf{P} \sim \mathcal{N}(0, \eta\sigma^2\boldsymbol{\Phi})$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, (1-\eta)\sigma^2\mathbf{I})$. The values of the parameters that we used were as follows: narrow sense heritability $\eta = \{0.1, 0.5\}$, sample size $n = 1000$, number of fixed effects $p_{fixed} = 5000$, number of SNPs used to calculate the kinship matrix $k = 10000$, percentage of causal SNPs $c = \{0\%, 1\%\}$ and $\sigma^2 = 1$. In addition to these parameters, we also varied the amount of overlap between the causal SNPs and the SNPs used to generate the kinship matrix. We considered two main scenarios:

1. None of the causal SNPs are included in the calculation of the kinship matrix:

$$\mathbf{X}^{(kinship)} = \left[\mathbf{X}^{(other)}\right]$$

2. All the causal SNPs are included in the calculation of the kinship matrix:

$$\mathbf{X}^{(kinship)} = \left[\mathbf{X}^{(other)}; \mathbf{X}^{(causal)}\right].$$

Both kinship matrices are meant to contrast the model behavior when the causal SNPs are included in both the main effects and random effects versus when the causal SNPs are only included in the main effects. These scenarios are motivated by the current standard of practice in GWAS where the candidate marker is excluded from the calculation of the kinship matrix [8]. This approach becomes much more difficult to apply in large-scale multivariable models where there is likely to be overlap between the variables in the design matrix and kinship matrix.

We compare `ggmix` to the lasso and the twostep method. For the lasso, we include the first 10 principal components of the estimated kinship as unpenalized predictors in the design matrix. For the twostep method, we first fit an intercept only model with a single random effect using the average information restricted maximum likelihood (AIREML) algorithm [37] as implemented in the `gaston` R package [38]. The residuals from this model are then used as the response in a regular lasso model. Note that in the twostep method, we have removed the kinship effect in the first step and therefore do not need to make any further adjustments when fitting the penalized model. We fit the lasso using the default settings in the `glmnet` package [20] and select the optimal value of the regularization parameter using 10-fold cross-validation.

Let $\hat{\lambda}$ be the estimated value of the optimal regularization parameter selected via cross-validation or GIC, $\widehat{\boldsymbol{\beta}}_{\hat{\lambda}}$ the estimate of $\boldsymbol{\beta}$ at regularization parameter $\hat{\lambda}$, $S_0 = \{j; (\boldsymbol{\beta})_j \neq 0\}$ the index of the true active set, $\widehat{S}_{\hat{\lambda}} = \left\{j; (\widehat{\boldsymbol{\beta}}_{\hat{\lambda}})_j \neq 0\right\}$ the index of the set of non-zero estimated coefficients, and $|A|$ the cardinality of set A.

We evaluate the methods based on correct sparsity defined as $\frac{1}{p}\sum_{j=1}^{p} A_j$, where

$$A_j = \begin{cases} 1 & \text{if } (\widehat{\boldsymbol{\beta}}_{\hat{\lambda}})_j = (\boldsymbol{\beta})_j = 0 \\ 1 & \text{if } (\widehat{\boldsymbol{\beta}}_{\hat{\lambda}})_j \neq 0, (\boldsymbol{\beta})_j \neq 0 \\ 0 & \text{if else.} \end{cases}$$

We also compare the model error ($\|\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{\hat{\lambda}}\|_2$), true positive rate ($|\widehat{S}_{\hat{\lambda}} \in S_0|/|S_0|$), false positive rate ($|\widehat{S}_{\hat{\lambda}} \notin S_0|/|j \notin S_0|$), and the variance components for the random effect and error term. The following estimator is used for the error variance of the lasso [39]:

$$\frac{1}{n - \widehat{S}_{\hat{\lambda}}} \left\| \mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{\hat{\lambda}} \right\|_2^2 \tag{32}$$

4.1 Results

We first plot the correct sparsity results for the null model ($c = 0$) and the model with 1% causal SNPs ($c = 0.01$) in Figures 3 and 4, respectively. When the true model has no causal SNPs, we see that `ggmix` has perfect Type 1 error control across all 200 replications while both the twostep and lasso methods sometimes estimate a model with a large number of false positives. When the true model contains some causal SNPs, `ggmix` again outperforms the other two methods in terms of correct sparsity. The distribution of $\widehat{S}_{\hat{\lambda}}$ for each of the three methods is shown in Figure 9 for $c = 0$ and Figure 10 for $c = 0.01$ of Supplemental Section B.

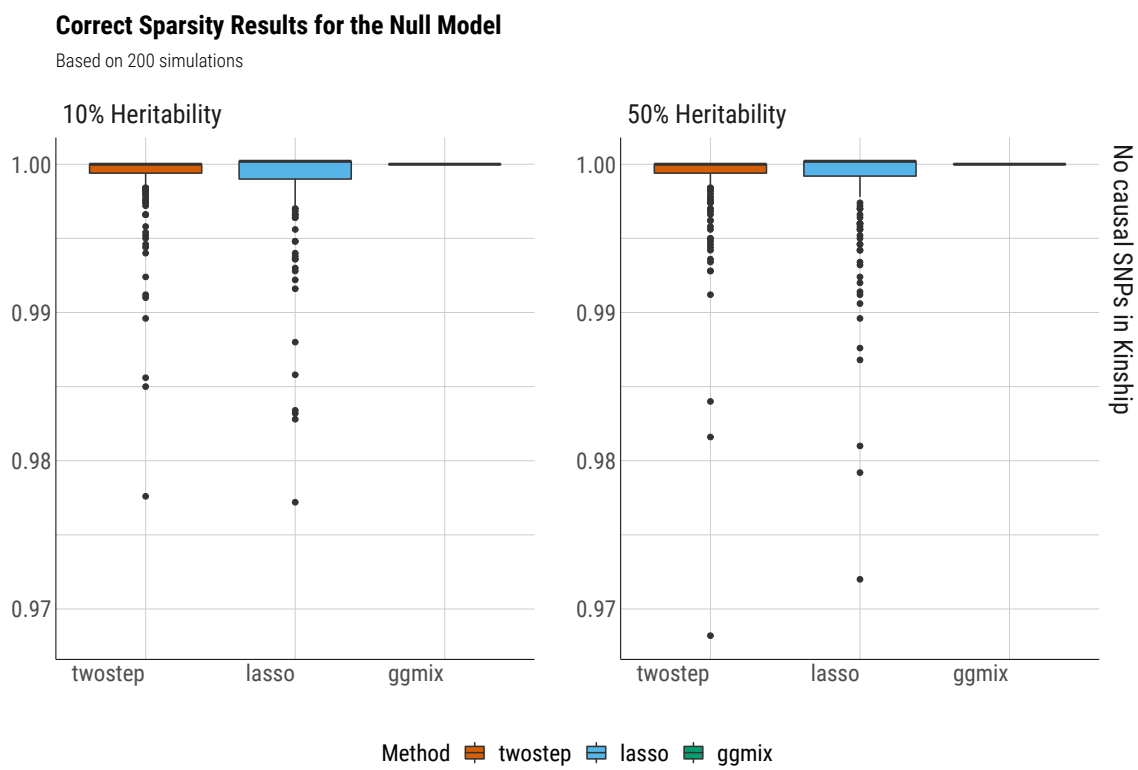


Figure 3: Boxplots of the correct sparsity from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ for the null model ($c = 0$).

The true positive vs. false positive rate for the model with 1% causal SNPs ($c = 0.01$) is

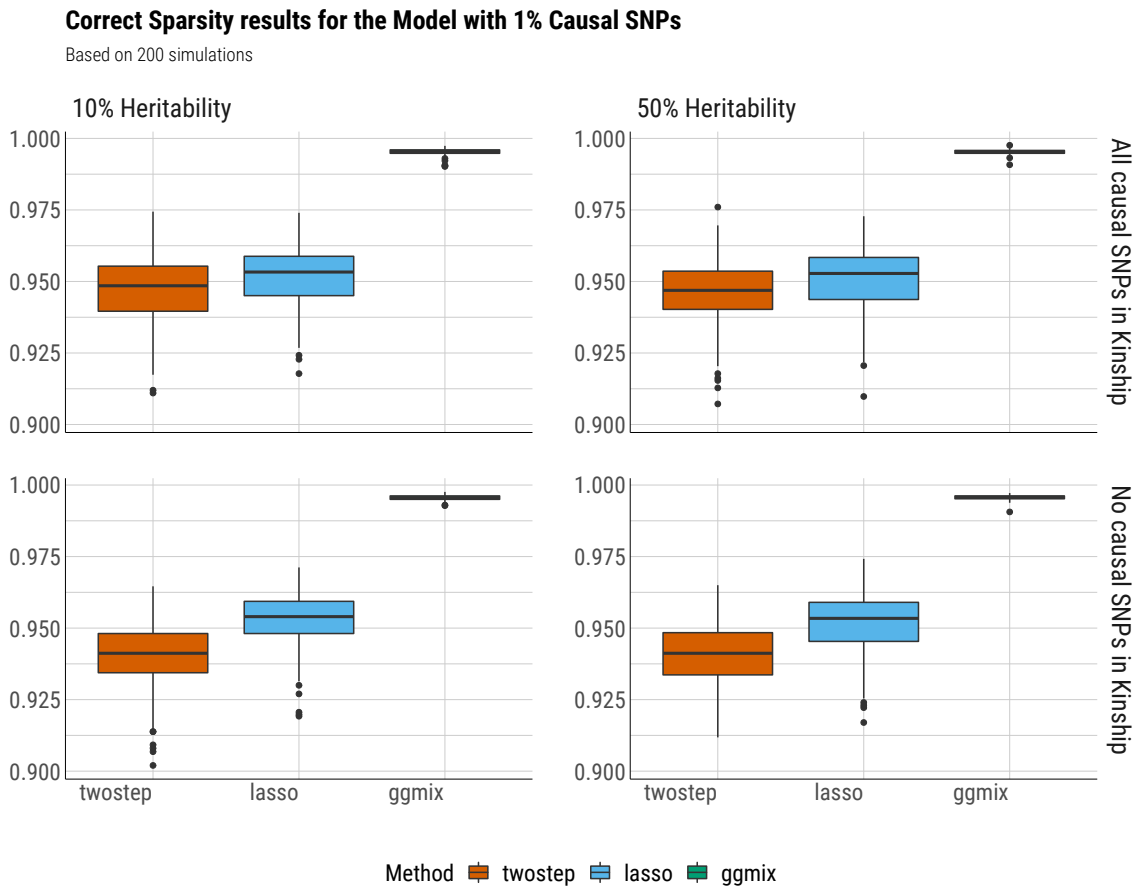


Figure 4: Boxplots of the correct sparsity from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ and number of causal SNPs that were included in the calculation of the kinship matrix for the model with 1% causal SNPs ($c = 0.01$).

shown in Figure 5. Both the lasso and twostep outperform `ggmix` in terms of identifying the true model. This accuracy however, comes at the cost of a very high false positive rate compared to `ggmix`.

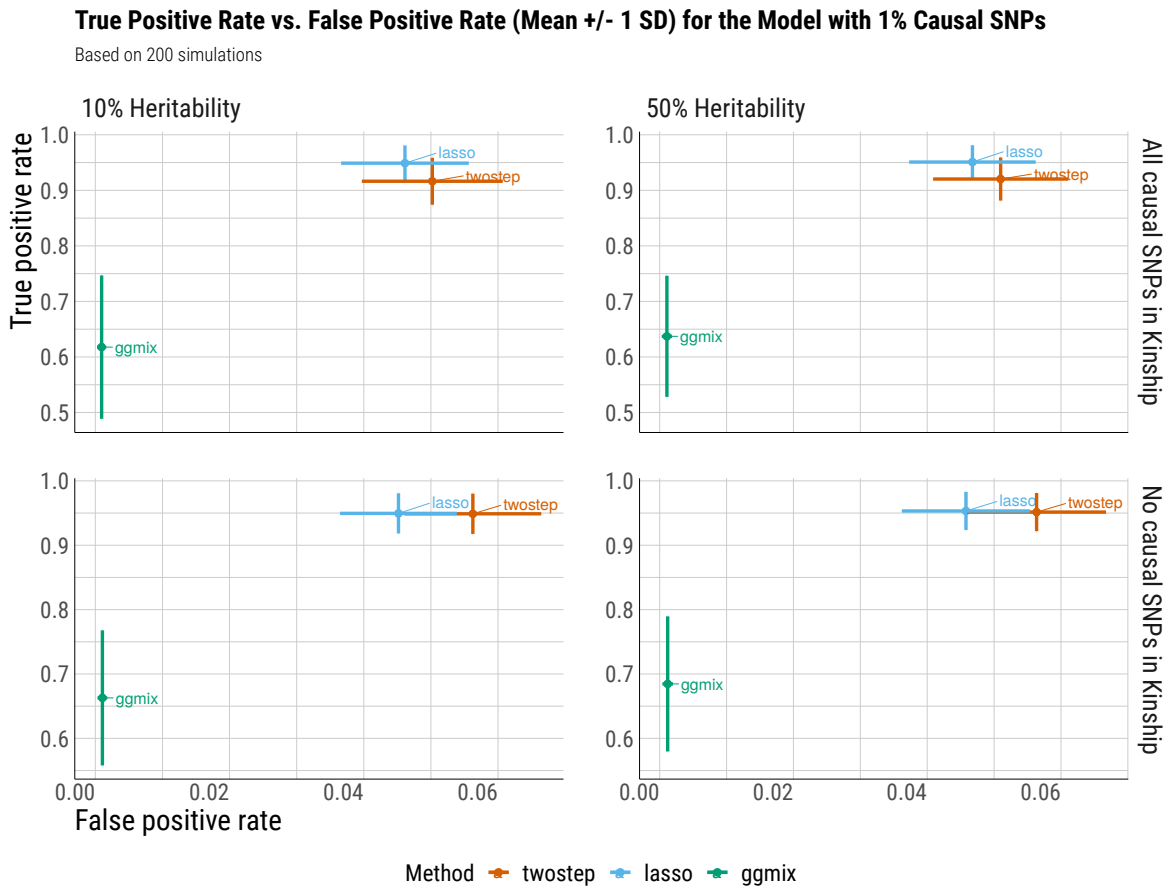


Figure 5: Means ± 1 standard deviation of true positive rate vs. false positive rate from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ and number of causal SNPs that were included in the calculation of the kinship matrix for the model with 1% causal SNPs ($c = 0.01$).

We plot the twostep and `ggmix` heritability estimates for $c = 0$ (Figure 11, Supplemental Section B) and $c = 0.01$ (Figure 6). We see that both methods correctly estimate the heritability in the null model. When all of the causal SNPs are in the kinship matrix, both methods overestimate η though `ggmix` is closer to the true value. When none of the causal SNPs are in the kinship, both methods tend to overestimate the truth when $\eta = 10\%$ and underestimate when $\eta = 50\%$.

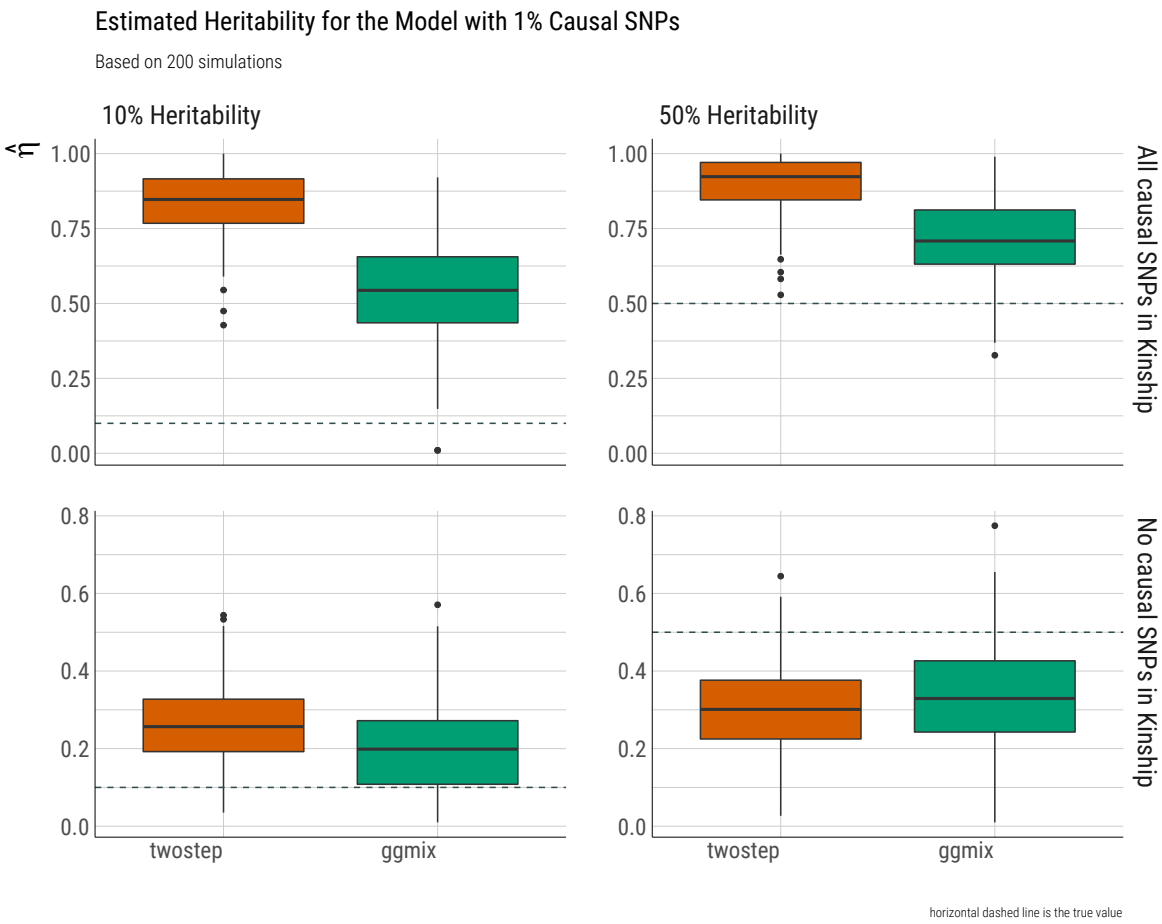


Figure 6: Boxplots of the heritability estimate $\hat{\eta}$ from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ and number of causal SNPs that were included in the calculation of the kinship matrix for the model with 1% causal SNPs ($c = 0.01$).

In Figures 12 (Supplemental Section B) and 7, we plot the error variance for $c = 0$ and $c = 0.01$, respectively. The twostep and `ggmix` methods correctly estimate the error variance while the lasso overestimates it for the null model and for when 1% of the causal SNPs are in the kinship matrix. We see an inflated estimated error variance across all three methods when $c = 0.01$ and none of the causal SNPs are in the kinship matrix with the lasso and `ggmix` performing similarly.

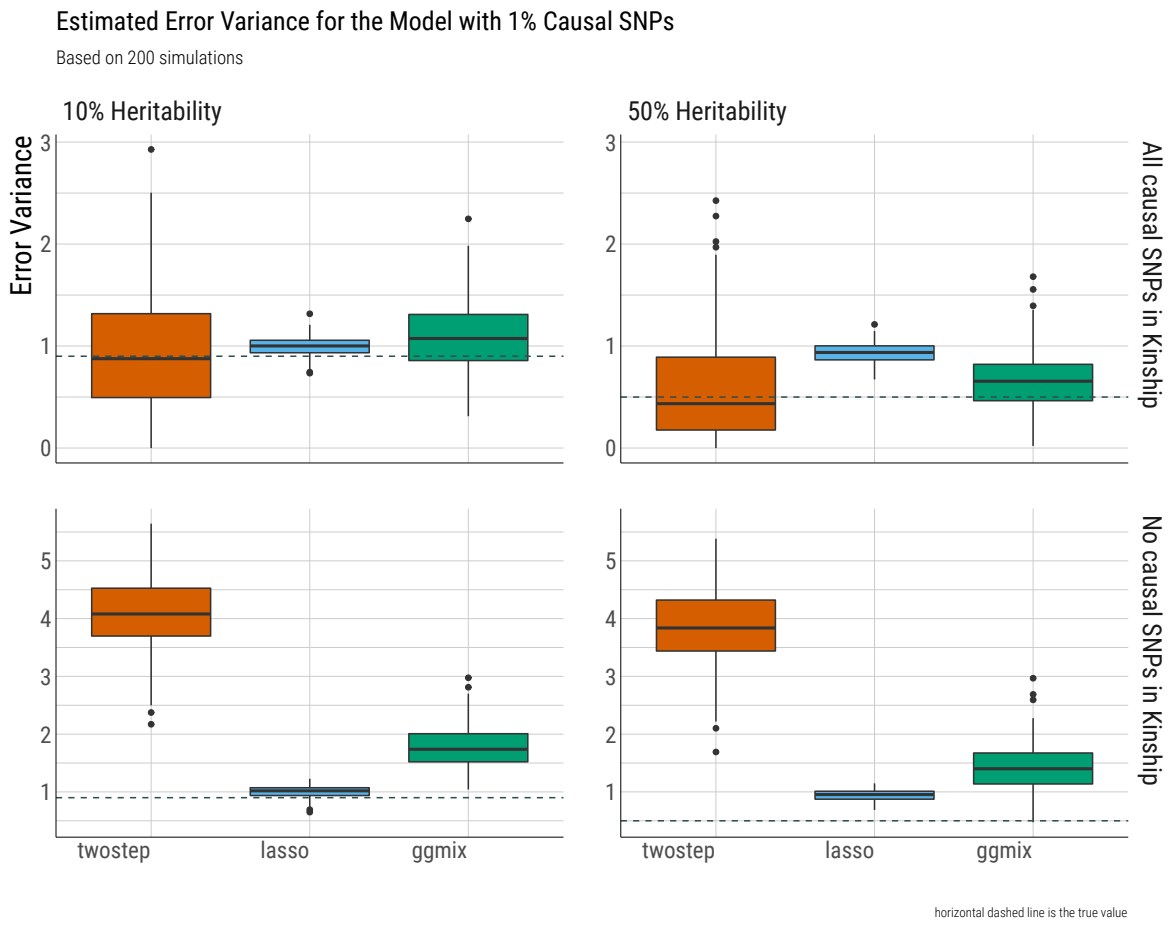


Figure 7: Boxplots of the estimated error variance from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ and number of causal SNPs that were included in the calculation of the kinship matrix for the model with 1% causal SNPs ($c = 0.01$).

We compare the model error as a function of $\widehat{S}_{\hat{\lambda}}$ in Figures 13 (Supplemental Section B) and 8 for $c = 0$ and $c = 0.01$, respectively. Lasso achieves the smallest model error across all scenarios (for $c = 0.01$), albeit with a large number of active variables. `ggmix` has a smaller model error compared to twostep when all causal SNPs are in the kinship matrix and similar performance when none of the causal SNPs are in the kinship matrix.

Figure 8: Means ± 1 standard deviation of the model error vs. the number of active variables by the true heritability $\eta = \{10\%, 50\%\}$ and number of causal SNPs that were included in the calculation of the kinship matrix for the model with 1% causal SNPs ($c = 0.01$).

Overall, we observe that variable selection results for `ggmix` are similar regardless of whether the causal SNPs are in the kinship matrix or not. This result is encouraging since in practice the kinship matrix is constructed from a random sample of SNPs across the genome, some of which are likely to be causal. `ggmix` has very good Type 1 error control, while both the lasso and twostep have a very high false positive rate. Inclusion of the causal SNPs in the kinship calculation has a strong impact on the variance component estimation with the heritabilty and error variance estimates working in opposite directions. That is, when all causal SNPs are in the kinship matrix, the heritability estimates are biased towards 1

while the error variance is correctly estimated. Conversely, when none of the causal SNPs are included in the kinship matrix, the estimated heritability is closer to the true value, while the error variance is inflated. Both the lasso and twostep methods have better signal recovery as compared to `ggmix`. However, this signal is being spread across many variables leading to many Type 1 errors.

5 Discussion

We develop a general penalized LMM framework for population structure correction that simultaneously selects and estimates variables, accounting for between individual correlations, in one step. Our CGD algorithm is computationally efficient and has theoretical guarantees of convergence. We provide an easy-to-use software implementation of our algorithm along with a principled method for automatic tuning parameter selection. Through simulation studies, we show that existing approaches such as a two-stage approach or the lasso with a principal component adjustment lead to a large number of false positives. Our proposed method has excellent Type 1 error control and is robust to the inclusion of causal SNPs in the kinship matrix. This feature is important since in practice the kinship matrix is constructed from a random sample of SNPs across the genome, some of which are likely to be causal.

Although we derive a CGD algorithm for the ℓ_1 penalty, our approach can also be easily extended to other penalties such as the elastic net and group lasso with the same guarantees of convergence.

A limitation of `ggmix` is that it first requires computing the covariance matrix with a computation time of $\mathcal{O}(n^2k)$ followed by a spectral decomposition of this matrix in $\mathcal{O}(n^3)$ time where k is the number of SNP genotypes used to construct the covariance matrix. This computation becomes prohibitive for large cohorts such as the UK Biobank [40] which have collected

genetic information on half a million individuals. When the matrix of genotypes used to construct the covariance matrix is low rank, there are additional computational speedups that can be implemented. While this has been developed for the univariate case [8], to our knowledge, this has not been explored in the multivariable case. We are currently developing a low rank version of the penalized LMM developed here, which reduces the time complexity from $\mathcal{O}(n^2k)$ to $\mathcal{O}(nk^2)$.

While the predominant motivation for our approach has been association testing, we believe that there are other applications in which it can be used as well. For example, in the most recent Genetic Analysis Workship 20 (GAW20), the causal modeling group investigated causal relationships between DNA methylation (exposure) within some genes and the change in high-density lipoproteins ΔHDL (outcome) using Mendelian randomization (MR) [41]. Penalized regression methods could be used to select SNPs strongly associated with the exposure in order to be used as an instrumental variable (IV). However, since GAW20 data consisted of families, two step methods were used which could have resulted in a large number of false positives. `ggmix` is an alternative approach that could be used for selecting the IV while accounting for the familial structure of the data. Our method is also suitable for fine mapping SNP association signals in genomic regions, where the goal is to pinpoint individual variants most likely to impact the undelying biological mechanisms of disease [42].

References

[1] Teri A Manolio, Francis S Collins, Nancy J Cox, David B Goldstein, Lucia A Hindorff, David J Hunter, Mark I McCarthy, Erin M Ramos, Lon R Cardon, Aravinda Chakravarti, et al. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747, 2009.

[2] Jian Yang, Beben Benyamin, Brian P McEvoy, Scott Gordon, Anjali K Henders, Dale R Nyholt, Pamela A Madden, Andrew C Heath, Nicholas G Martin, Grant W Montgomery, et al. Common snps explain a large proportion of the heritability for human height. *Nature genetics*, 42(7):565, 2010.

[3] William Astle, David J Balding, et al. Population structure and cryptic relatedness in genetic association studies. *Statistical Science*, 24(4):451–471, 2009.

[4] Minsun Song, Wei Hao, and John D Storey. Testing for genetic associations in arbitrarily structured populations. *Nature genetics*, 47(5):550–554, 2015.

[5] Jonathan Marchini, Lon R Cardon, Michael S Phillips, and Peter Donnelly. The effects of human population structure on large genetic association studies. *Nature genetics*, 36(5):512, 2004.

[6] Clive J Hoggart, John C Whittaker, Maria De Iorio, and David J Balding. Simultaneous analysis of all snps in genome-wide and re-sequencing association studies. *PLoS genetics*, 4(7):e1000130, 2008.

[7] Jiahan Li, Kiranmoy Das, Guifang Fu, Runze Li, and Rongling Wu. The bayesian lasso for genome-wide association studies. *Bioinformatics*, 27(4):516–523, 2010.

[8] Christoph Lippert, Jennifer Listgarten, Ying Liu, Carl M Kadie, Robert I Davidson, and David Heckerman. Fast linear mixed models for genome-wide association studies. *Nature methods*, 8(10):833–835, 2011.

[9] Hyun Min Kang, Jae Hoon Sul, Noah A Zaitlen, Sit-yee Kong, Nelson B Freimer, Chiara Sabatti, Eleazar Eskin, et al. Variance component model to account for sample structure in genome-wide association studies. *Nature genetics*, 42(4):348, 2010.

[10] Jianming Yu, Gael Pressoir, William H Briggs, Irie Vroh Bi, Masanori Yamasaki, John F Doebley, Michael D McMullen, Brandon S Gaut, Dahlia M Nielsen, James B Holland, et al. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature genetics*, 38(2):203, 2006.

[11] Jakris Eu-Ahsunthornwattana, E Nancy Miller, Michaela Fakiola, Selma MB Jeronimo, Jenefer M Blackwell, Heather J Cordell, Wellcome Trust Case Control Consortium 2, et al. Comparison of methods to account for relatedness in genome-wide association studies with family-based data. *PLoS Genet*, 10(7):e1004445, 2014.

[12] Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904, 2006.

[13] Karim Oualkacha, Zari Dastani, Rui Li, Pablo E Cingolani, Timothy D Spector, Christopher J Hammond, J Brent Richards, Antonio Ciampi, and Celia MT Greenwood. Adjusted sequence kernel association test for rare variants controlling for cryptic and family relatedness. *Genetic epidemiology*, 37(4):366–376, 2013.

[14] Heather J Cordell and David G Clayton. A unified stepwise regression procedure for evaluating the relative effects of polymorphisms within a gene using case/control or family data: application to hla in type 1 diabetes. *The American Journal of Human Genetics*, 70(1):124–141, 2002.

[15] Barbara Rakitsch, Christoph Lippert, Oliver Stegle, and Karsten Borgwardt. A lasso multi-marker mixed model for association mapping with population structure correction. *Bioinformatics*, 29(2):206–214, 2013.

[16] Dong Wang, Kent M Eskridge, and Jose Crossa. Identifying qtls and epistasis in structured plant populations using adaptive mixed lasso. *Journal of agricultural, biological, and environmental statistics*, 16(2):170–184, 2011.

[17] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[18] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

[19] Xiuhua Ding, Shaoyong Su, Kannabiran Nandakumar, Xiaoling Wang, and David W Fardo. A 2-step penalized regression method for family-based next-generation sequencing association studies. In *BMC proceedings*, volume 8, page S25. BioMed Central, 2014.

[20] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[21] Yi Yang and Hui Zou. A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing*, 25(6):1129–1141, 2015.

[22] Jian Yang, Noah A Zaitlen, Michael E Goddard, Peter M Visscher, and Alkes L Price. Advantages and pitfalls in the application of mixed-model association methods. *Nature genetics*, 46(2):100, 2014.

[23] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[24] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped

variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[25] Matti Pirinen, Peter Donnelly, Chris CA Spencer, et al. Efficient computation with a linear mixed model on large-scale data sets with applications to genetic studies. *The Annals of Applied Statistics*, 7(1):369–390, 2013.

[26] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009.

[27] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[28] Jürg Schelldorfer, Peter Bühlmann, GEER DE, and SARA VAN. Estimation for high-dimensional linear mixed-effects models using l1-penalization. *Scandinavian Journal of Statistics*, 38(2):197–214, 2011.

[29] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[30] Jon Wakefield. *Bayesian and frequentist regression methods*. Springer Science & Business Media, 2013.

[31] Ryuei Nishii. Asymptotic properties of criteria for selection of variables in multiple regression. *The Annals of Statistics*, pages 758–765, 1984.

[32] Hui Zou, Trevor Hastie, Robert Tibshirani, et al. On the degrees of freedom of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.

[33] Howard D Bondell, Arun Krishna, and Sujit K Ghosh. Joint variable selection for fixed and random effects in linear mixed-effects models. *Biometrics*, 66(4):1069–1077, 2010.

[34] Yingying Fan and Cheng Yong Tang. Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):531–552, 2013.

[35] Alejandro Ochoa and John D. Storey. FST and kinship for arbitrary population structures I: Generalized definitions. *bioRxiv*, 2016.

[36] Alejandro Ochoa and John D. Storey. FST and kinship for arbitrary population structures II: Method of moments estimators. *bioRxiv*, 2016.

[37] Arthur R Gilmour, Robin Thompson, and Brian R Cullis. Average information reml: an efficient algorithm for variance parameter estimation in linear mixed models. *Biometrics*, pages 1440–1450, 1995.

[38] Claire Dandine-Roulland. *gaston: Genetic Data Handling (QC, GRM, LD, PCA) and Linear Mixed Models*, 2018. R package version 1.5.3.

[39] Stephen Reid, Robert Tibshirani, and Jerome Friedman. A study of error variance estimation in lasso regression. *Statistica Sinica*, pages 35–67, 2016.

[40] Naomi Allen, Cathie Sudlow, Paul Downey, Tim Peakman, John Danesh, Paul Elliott, John Gallacher, Jane Green, Paul Matthews, Jill Pell, et al. Uk biobank: Current status and what it means for epidemiology. *Health Policy and Technology*, 1(3):123–126, 2012.

[41] George Davey Smith and Shah Ebrahim. mendelian randomization: can genetic epidemiology contribute to understanding environmental determinants of disease? *International journal of epidemiology*, 32(1):1–22, 2003.

[42] Sarah L Spain and Jeffrey C Barrett. Strategies for fine-mapping complex traits. *Human molecular genetics*, 24(R1):R111–R119, 2015.

[43] Yihui Xie. *Dynamic Documents with R and knitr*, volume 29. CRC Press, 2015.

A Block Coordinate Descent Algorithm

We use a general purpose block coordinate descent algorithm (CGD) [26] to solve (15). At each iteration, the algorithm approximates the negative log-likelihood $f(\cdot)$ in $Q_\lambda(\cdot)$ by a strictly convex quadratic function and then applies block coordinate decent to generate a decent direction followed by an inexact line search along this direction [26]. For continuously differentiable $f(\cdot)$ and convex and block-separable $P(\cdot)$ (i.e. $P(\boldsymbol{\beta}) = \sum_i P_i(\beta_i)$), [26] show that the solution generated by the CGD method is a stationary point of $Q_\lambda(\cdot)$ if the coordinates are updated in a Gauss-Seidel manner i.e. $Q_\lambda(\cdot)$ is minimized with respect to one parameter while holding all others fixed. The CGD algorithm can thus be run in parallel and therefore suited for large p settings. It has been successfully applied in fixed effects models (e.g. [27], [20]) and [28] for mixed models with an ℓ_1 penalty. Following Tseng and Yun [26], the CGD algorithm is given by Algorithm 2.

The Armijo rule is defined as follows [26]:

Choose $\alpha_{init}^{(k)} > 0$ and let $\alpha^{(k)}$ be the largest element of $\left\{\alpha_{init}^{k}\delta^r\right\}_{r=0,1,2,\ldots}$ satisfying

$$Q_\lambda(\Theta_j^{(k)} + \alpha^{(k)} d^{(k)}) \leq Q_\lambda(\Theta_j^{(k)}) + \alpha^{(k)} \varrho \Delta^{(k)} \tag{37}$$

where $0 < \delta < 1$, $0 < \varrho < 1$, $0 \leq \gamma < 1$ and

$$\Delta^{(k)} := \nabla f(\Theta_j^{(k)}) d^{(k)} + \gamma (d^{(k)})^2 H_{jj}^{(k)} + \lambda P(\Theta_j^{(k)} + d^{(k)}) - \lambda P(\Theta^{(k)}) \tag{38}$$

Common choices for the constants are $\delta = 0.1$, $\varrho = 0.001$, $\gamma = 0$, $\alpha_{init}^{(k)} = 1$ for all k [28].

Below we detail the specifics of Algorithm 2 for the ℓ_1 penalty.

Algorithm 2: Coordinate Gradient Descent Algorithm to solve (15)

Set the iteration counter $k \leftarrow 0$ and choose initial values for the parameter vector $\mathbf{\Theta}^{(0)}$;

repeat

Approximate the Hessian $\nabla^2 f(\mathbf{\Theta}^{(k)})$ by a symmetric matix $H^{(k)}$:

$$H^{(k)} = \text{diag}\left[\min\left\{\max\left\{\left[\nabla^2 f(\mathbf{\Theta}^{(k)})\right]_{jj}, c_{min}\right\} c_{max}\right\}\right]_{j=1,\ldots,p} \tag{33}$$

for $j = 1, \ldots, p$ **do**

Solve the descent direction $d^{(k)} := d_{H^{(k)}}(\Theta_j^{(k)})$;

if $\Theta_j^{(k)} \in \{\beta_1, \ldots, \beta_p\}$ **then**

$$d_{H^{(k)}}(\Theta_j^{(k)}) \leftarrow \arg\min_d \left\{\nabla f(\Theta_j^{(k)})d + \frac{1}{2}d^2 H_{jj}^{(k)} + \lambda P(\Theta_j^{(k)} + d)\right\} \tag{34}$$

end

end

Choose a stepsize;

$$\alpha_j^{(k)} \leftarrow \text{line search given by the Armijo rule}$$

Update;

$$\widehat{\Theta}_j^{(k+1)} \leftarrow \widehat{\Theta}_j^{(k)} + \alpha_j^{(k)} d^{(k)}$$

Update;

$$\widehat{\eta}^{(k+1)} \leftarrow \arg\min_\eta \frac{1}{2}\sum_{i=1}^{N_T} \log(1 + \eta(\Lambda_i - 1)) + \frac{1}{2\sigma^{2\,(k)}} \sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j^{(k+1)}\right)^2}{1 + \eta(\Lambda_i - 1)} \tag{35}$$

Update;

$$\widehat{\sigma^2}^{\,(k+1)} \leftarrow \frac{1}{N_T}\sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j^{(k+1)}\right)^2}{1 + \eta^{(k+1)}(\Lambda_i - 1)} \tag{36}$$

$k \leftarrow k + 1$

until *convergence criterion is satisfied*;

A.1 ℓ_1 penalty

The objective function is given by

$$Q_\lambda(\boldsymbol{\Theta}) = f(\boldsymbol{\Theta}) + \lambda|\boldsymbol{\beta}| \tag{39}$$

A.1.1 Descent Direction

For simplicity, we remove the iteration counter (k) from the derivation below.

For $\Theta_j^{(k)} \in \{\beta_1, \ldots, \beta_p\}$, let

$$d_H(\Theta_j) = \arg\min_d G(d) \tag{40}$$

where

$$G(d) = \nabla f(\Theta_j)d + \frac{1}{2}d^2 H_{jj} + \lambda|\Theta_j + d|$$

Since $G(d)$ is not differentiable at $-\Theta_j$, we calculate the subdifferential $\partial G(d)$ and search for d with $0 \in \partial G(d)$:

$$\partial G(d) = \nabla f(\Theta_j) + dH_{jj} + \lambda u \tag{41}$$

where

$$u = \begin{cases} 1 & \text{if} \quad d > -\Theta_j \\ -1 & \text{if} \quad d < -\Theta_j \\ [-1, 1] & \text{if} \quad d = \Theta_j \end{cases} \tag{42}$$

We consider each of the three cases in (41) below

1. $d > -\Theta_j$

$$\partial G(d) = \nabla f(\Theta_j) + dH_{jj} + \lambda = 0$$
$$d = \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}}$$

Since $\lambda > 0$ and $H_{jj} > 0$, we have

$$\frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}} > \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}} = d \overset{\text{def}}{>} -\Theta_j$$

The solution can be written compactly as

$$d = \text{mid}\left\{\frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}, -\Theta_j, \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}}\right\}$$

where mid $\{a, b, c\}$ denotes the median (mid-point) of a, b, c [26].

2. $d < -\Theta_j$

$$\partial G(d) = \nabla f(\Theta_j) + dH_{jj} - \lambda = 0$$
$$d = \frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}$$

Since $\lambda > 0$ and $H_{jj} > 0$, we have

$$\frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}} < \frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}} = d \overset{\text{def}}{<} -\Theta_j$$

Again, the solution can be written compactly as

$$d = \text{mid}\left\{\frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}, -\Theta_j, \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}}\right\}$$

3. $d_j = -\Theta_j$

There exists $u \in [-1, 1]$ such that

$$\partial G(d) = \nabla f(\Theta_j) + dH_{jj} + \lambda u = 0$$
$$d = \frac{-(\nabla f(\Theta_j) + \lambda u)}{H_{jj}}$$

For $-1 \leq u \leq 1$, $\lambda > 0$ and $H_{jj} > 0$ we have

$$\frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}} \leq d \stackrel{\text{def}}{=} -\Theta_j \leq \frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}$$

The solution can again be written compactly as

$$d = \text{mid}\left\{\frac{-(\nabla f(\Theta_j) - \lambda)}{H_{jj}}, -\Theta_j, \frac{-(\nabla f(\Theta_j) + \lambda)}{H_{jj}}\right\}$$

We see all three cases lead to the same solution for (40). Therefore the descent direction for $\Theta_j^{(k)} \in \{\beta_1, \ldots, \beta_p\}$ for the ℓ_1 penalty is given by

$$d = \text{mid}\left\{\frac{-(\nabla f(\beta_j) - \lambda)}{H_{jj}}, -\beta_j, \frac{-(\nabla f(\beta_j) + \lambda)}{H_{jj}}\right\} \tag{43}$$

A.1.2 Solution for the β parameter

If the Hessian $\nabla^2 f(\boldsymbol{\Theta}^{(k)}) > 0$ then $H^{(k)}$ defined in (33) is equal to $\nabla^2 f(\boldsymbol{\Theta}^{(k)})$. Using $\alpha_{init} = 1$, the largest element of $\left\{\alpha_{init}^{(k)}\delta^r\right\}_{r=0,1,2,\ldots}$ satisfying the Armijo Rule inequality is reached for $\alpha^{(k)} = \alpha_{init}^{(k)}\delta^0 = 1$. The Armijo rule update for the $\boldsymbol{\beta}$ parameter is then given by

$$\beta_j^{(k+1)} \leftarrow \beta_j^{(k)} + d^{(k)}, \qquad j = 1, \ldots, p \tag{44}$$

Substituting the descent direction given by (43) into (44) we get

$$\beta_j^{(k+1)} = \text{mid}\left\{\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) - \lambda)}{H_{jj}}, 0, \beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) + \lambda)}{H_{jj}}\right\} \tag{45}$$

We can further simplify this expression. Let

$$w_i := \frac{1}{\sigma^2\left(1 + \eta(\Lambda_i - 1)\right)} \tag{46}$$

.

Re-write the part depending on $\boldsymbol{\beta}$ of the negative log-likelihood in (13) as

$$g(\boldsymbol{\beta}^{(k)}) = \frac{1}{2}\sum_{i=1}^{N_T} w_i \left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)} - \widetilde{X}_{ij}\beta_j^{(k)}\right)^2 \tag{47}$$

The gradient and Hessian are given by

$$\nabla f(\beta_j^{(k)}) := \frac{\partial}{\partial\beta_j^{(k)}} g(\boldsymbol{\beta}^{(k)}) = -\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)} - \widetilde{X}_{ij}\beta_j^{(k)}\right) \tag{48}$$

$$H_{jj} := \frac{\partial^2}{\partial{\beta_j^{(k)}}^2} g(\boldsymbol{\beta}^{(k)}) = \sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}^2 \tag{49}$$

Substituting (48) and (49) into $\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)})-\lambda)}{H_{jj}}$

$$\begin{aligned}
&\beta_j^{(k)} + \frac{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)} - \widetilde{X}_{ij}\beta_j^{(k)}\right) + \lambda}{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}^2} \\
&= \beta_j^{(k)} + \frac{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) + \lambda}{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}^2} - \frac{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}^2\beta_j^{(k)}}{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}^2} \\
&= \frac{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) + \lambda}{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}^2}
\end{aligned} \tag{50}$$

Similarly, substituting (48) and (49) in $\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)})+\lambda)}{H_{jj}}$ we get

$$\frac{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) - \lambda}{\sum_{i=1}^{N_T} w_i\widetilde{X}_{ij}^2} \tag{51}$$

Finally, substituting (50) and (51) into (45) we get

$$\begin{aligned}\beta_j^{(k+1)} &= \text{mid}\left\{\frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) - \lambda}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2}, 0, \frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) + \lambda}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2}\right\} \\ &= \frac{\mathcal{S}_\lambda\left(\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right)\right)}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2} \qquad (52)\end{aligned}$$

Where $\mathcal{S}_\lambda(x)$ is the soft-thresholding operator

$$\mathcal{S}_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+$$

$\text{sign}(x)$ is the signum function

$$\text{sign}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

and $(x)_+ = \max(x, 0)$.

B Additional Simulation Results

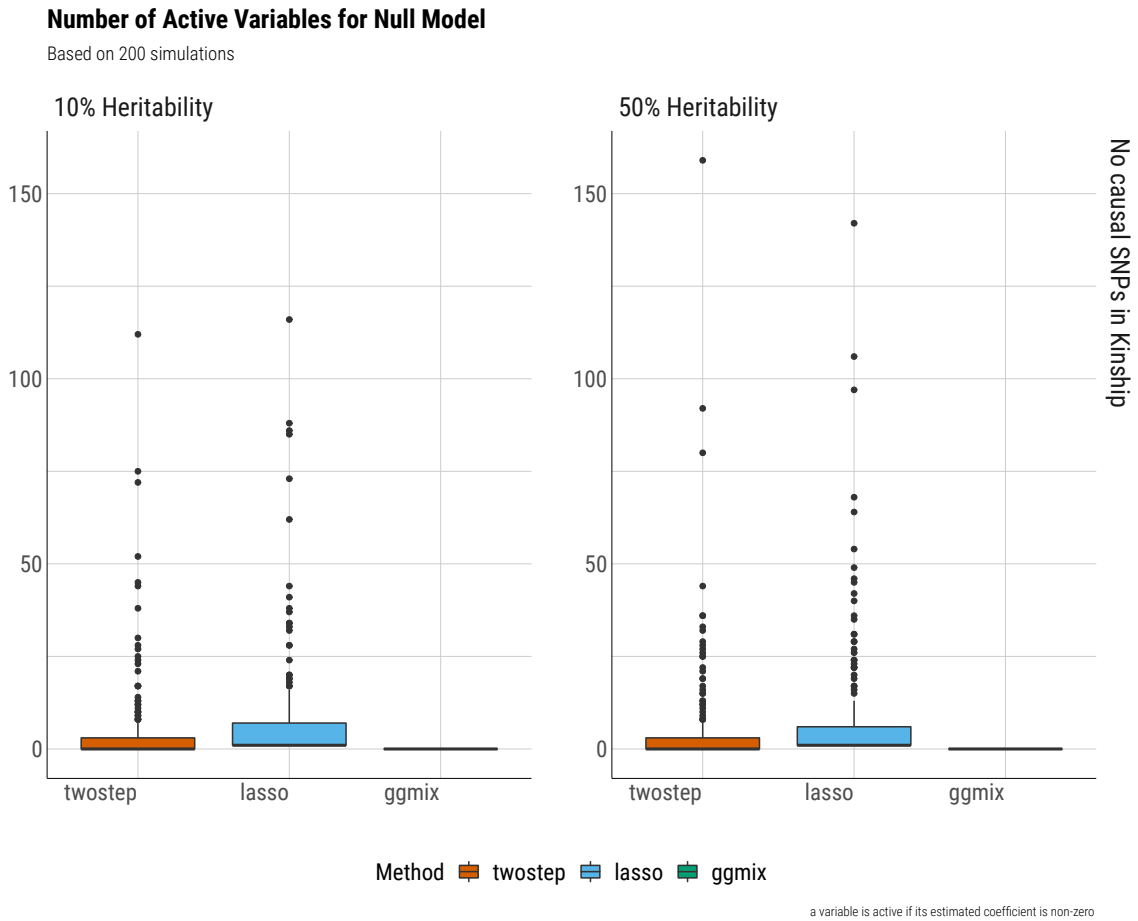


Figure 9: Boxplots of the number of active variables from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ for the null model ($c = 0$).

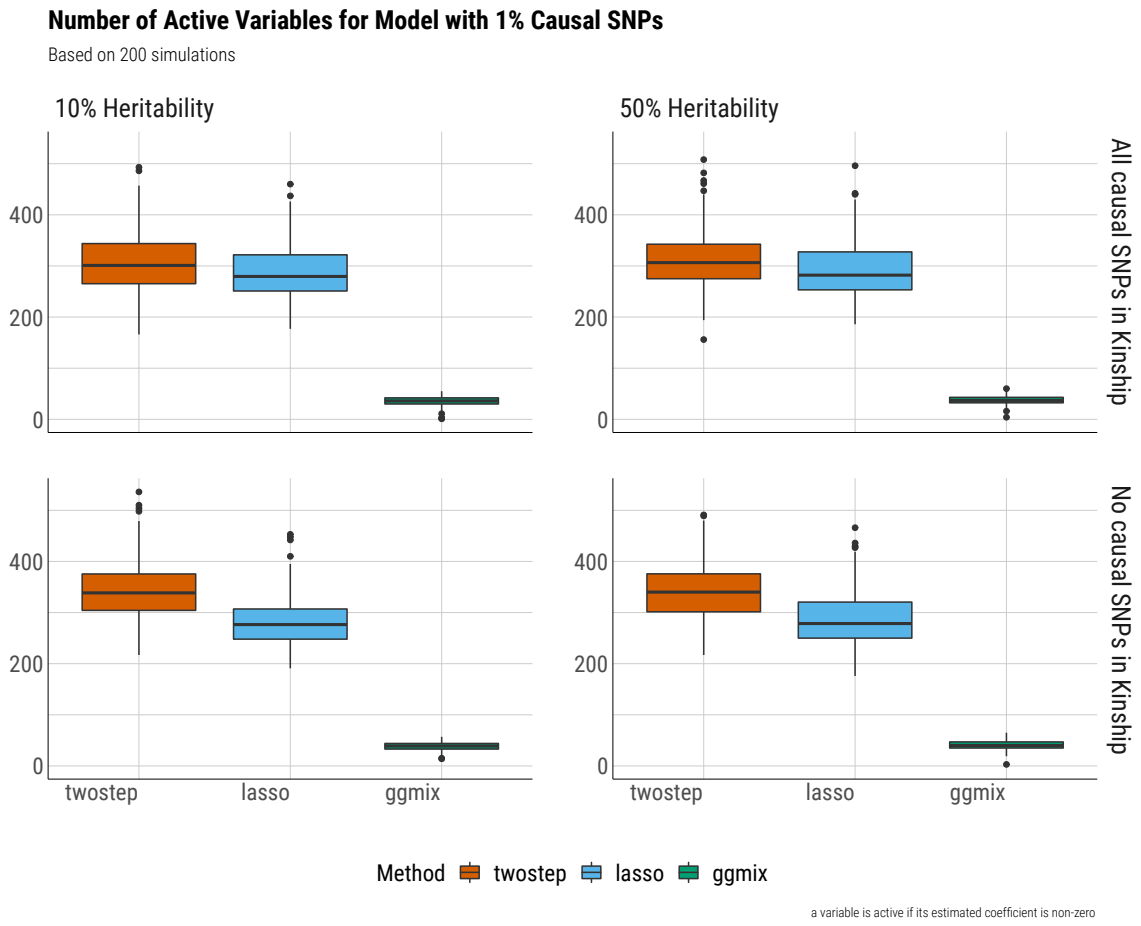


Figure 10: Boxplots of the number of active variables from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ and number of causal SNPs that were included in the calculation of the kinship matrix for the model with 1% causal SNPs ($c = 0.01$).

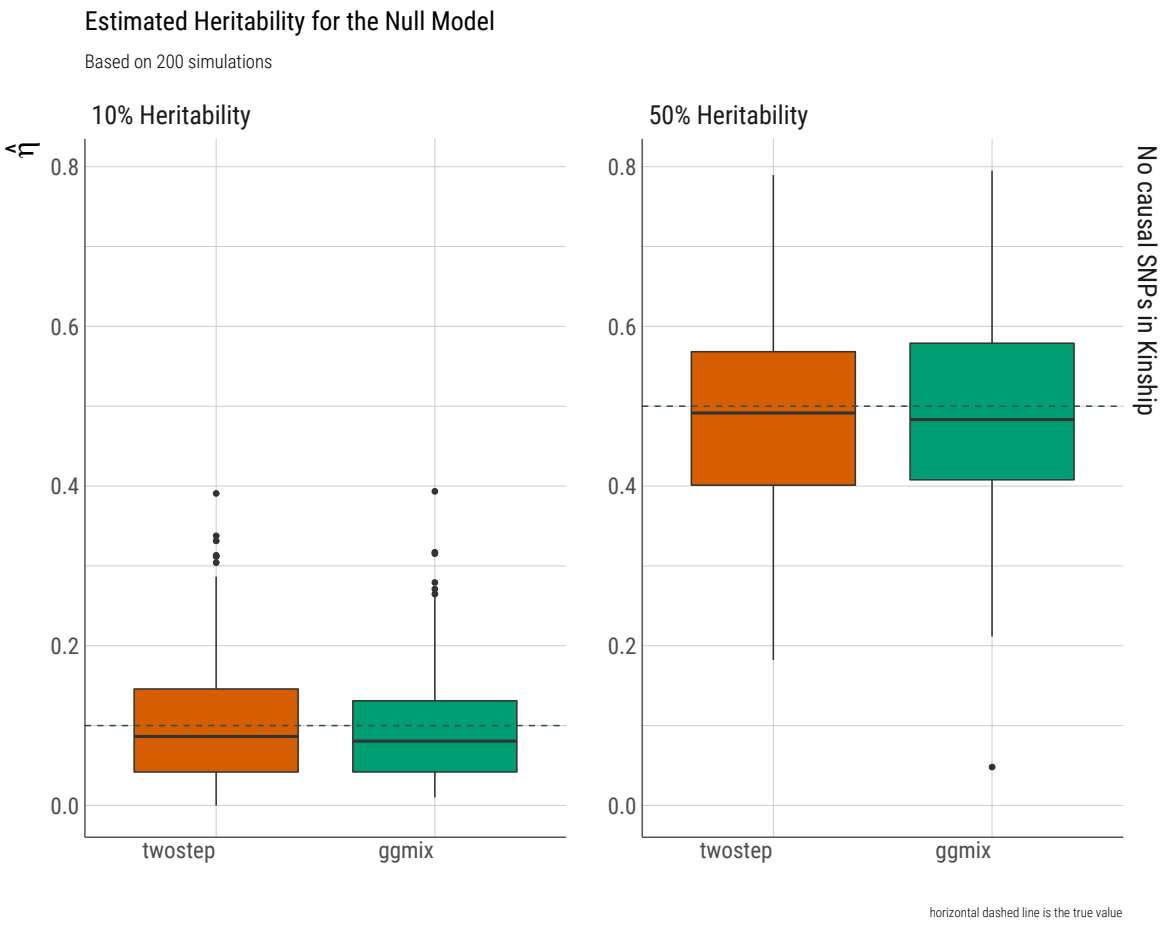


Figure 11: Boxplots of the heritability estimate $\hat{\eta}$ from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ for the null model ($c = 0$).

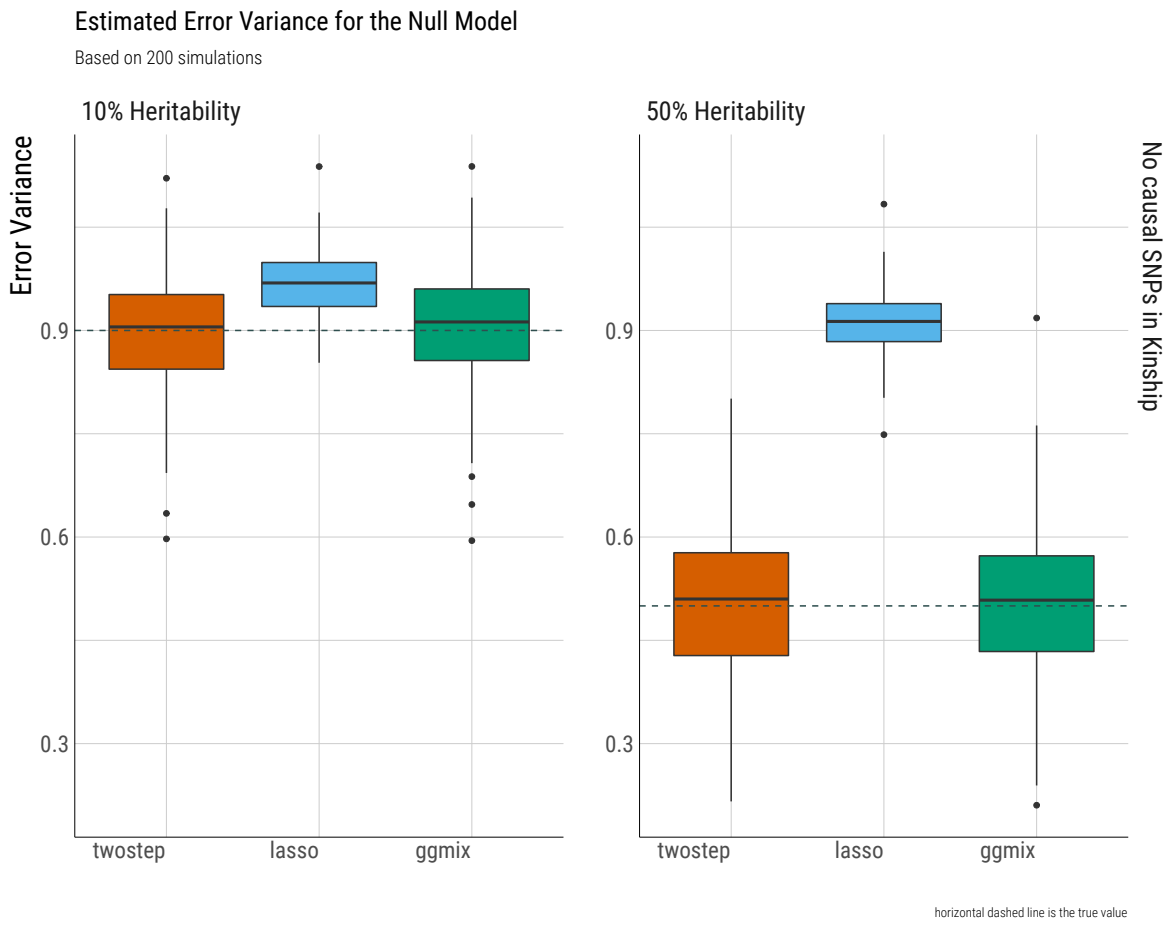


Figure 12: Boxplots of the estimated error variance from 200 simulations by the true heritability $\eta = \{10\%, 50\%\}$ for the null model ($c = 0$).

Figure 13: Means ± 1 standard deviation of the model error vs. the number of active variables by the true heritability $\eta = \{10\%, 50\%\}$ for the null model ($c = 0$).

C ggmix Package Showcase

In this section we briefly introduce the freely available and open source ggmix package in R. More comprehensive documentation is available at https://sahirbhatnagar.com/ggmix. Note that this entire section is reproducible; the code and text are combined in an .Rnw[1] file and compiled using knitr [43].

C.1 Installation

The package can be installed from GitHub via

```
install.packages("pacman")
pacman::p_load_gh('sahirbhatnagar/ggmix')
```

To showcase the main functions in ggmix, we will use the simulated data which ships with the package and can be loaded via:

```
library(ggmix)
data("admixed")
names(admixed)

##  [1] "y"               "x"               "causal"
##  [4] "beta"            "kin"             "Xkinship"
##  [7] "not_causal"      "causal_positive" "causal_negative"
## [10] "x_lasso"
```

For details on how this data was simulated, see help(admixed).

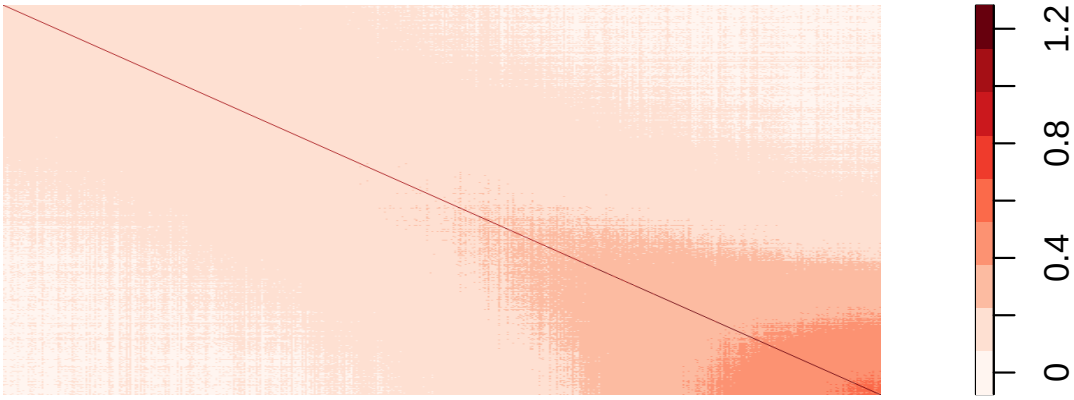
There are three basic inputs that ggmix needs:

1. Y: a continuous response variable
2. X: a matrix of covariates of dimension $N \times p$ where N is the sample size and p is the number of covariates
3. $\mathbf{\Phi}$: a kinship matrix

[1] scripts available at https://github.com/sahirbhatnagar/ggmix/tree/master/manuscript

We can visualize the kinship matrix in the `admixed` data using the `popkin` package:

```
# need to install the package if you don't have it
# pacman::p_load_gh('StoreyLab/popkin')
popkin::plotPopkin(admixed$kin)
```



C.2 Fit the linear mixed model with Lasso Penalty

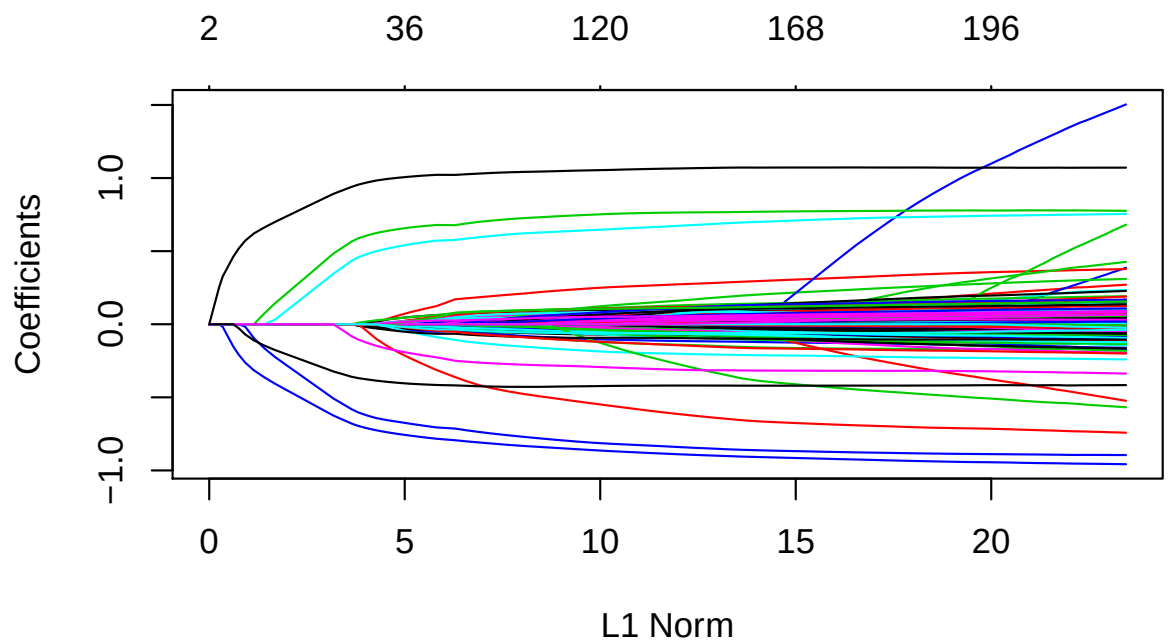
We will use the most basic call to the main function of this package, which is called `ggmix`. This function will by default fit a L_1 penalized linear mixed model (LMM) for 100 distinct values of the tuning parameter λ. It will choose its own sequence:

```
fit <- ggmix(x = admixed$x, y = admixed$y, kinship = admixed$kin)
names(fit)

##  [1] "result"       "ggmix_object" "n_design"     "p_design"
##  [5] "lambda"       "coef"         "b0"           "beta"
##  [9] "df"           "eta"          "sigma2"       "nlambda"
## [13] "cov_names"    "call"

class(fit)
```

```
## [1] "lassofullrank" "ggmix_fit"
```

We can see the solution path for each variable by calling the `plot` method for objects of class `ggmix_fit`:

```
plot(fit)
```



We can also get the coefficients for given value(s) of lambda using the `coef` method for objects of class `ggmix_fit`:

```
# only the first 5 coefficients printed here for brevity
coef(fit, s = c(0.1,0.02))[1:5, ]

## 5 x 2 Matrix of class "dgeMatrix"
##                      1            2
## (Intercept) -0.3824525 -0.030227753
## X62          0.0000000  0.000000000
## X185         0.0000000  0.001444670
## X371         0.0000000  0.009513604
## X420         0.0000000  0.000000000
```

Here, `s` specifies the value(s) of λ at which the extraction is made. The function uses linear

interpolation to make predictions for values of `s` that do not coincide with the lambda sequence used in the fitting algorithm.

We can also get predictions ($X\widehat{\boldsymbol{\beta}}$) using the `predict` method for objects of class `ggmix_fit`:

```
# need to provide x to the predict function
# predict for the first 5 subjects
predict(fit, s = c(0.1,0.02), newx = admixed$x[1:5,])

##               1          2
## id1 -1.19165061 -1.3123396
## id2 -0.02913052  0.3885921
## id3 -2.00084875 -2.6460045
## id4 -0.37255277 -0.9542455
## id5 -1.03967831 -2.1377274
```

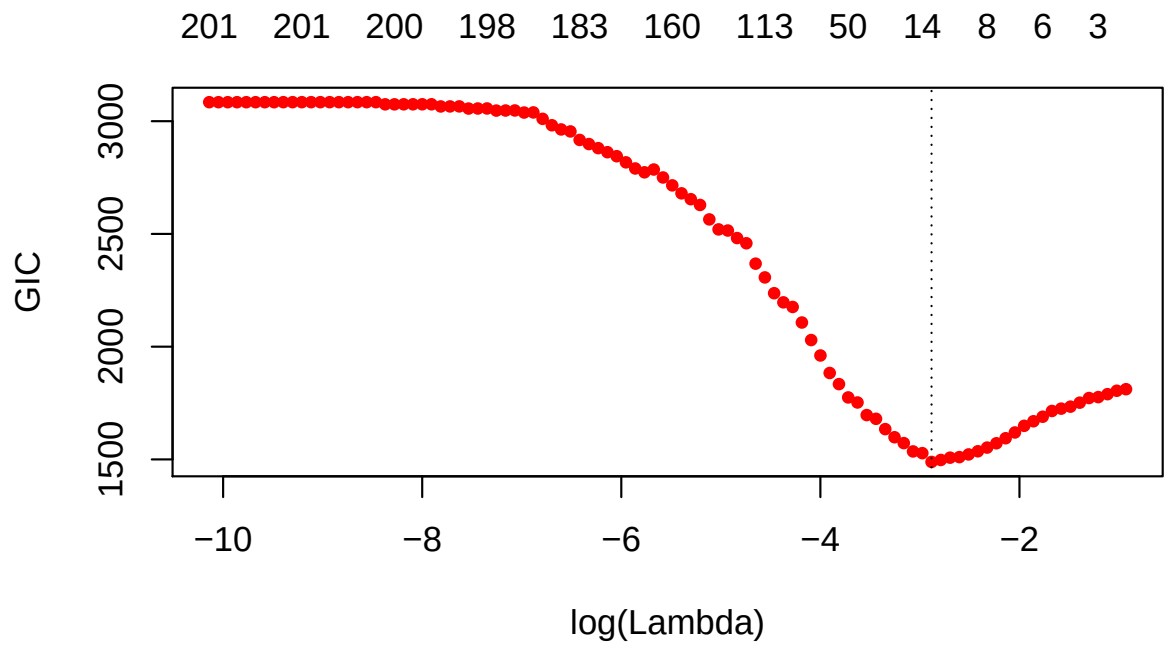
C.3 Find the Optimal Value of the Tuning Parameter

We use the Generalized Information Criterion (GIC) to select the optimal value for λ. The default is $a_n = log(log(n)) * log(p)$ which corresponds to a high-dimensional BIC (HD-BIC):

```
# pass the fitted object from ggmix to the gic function:
hdbic <- gic(fit)
class(hdbic)

## [1] "ggmix_gic"     "lassofullrank" "ggmix_fit"

# we can also fit the BIC by specifying the an argument
bicfit <- gic(fit, an = log(length(admixed$y)))
```

We can plot the HDBIC values against $\log(\lambda)$ using the `plot` method for objects of class `ggmix_gic`:

```
plot(hdbic)
```

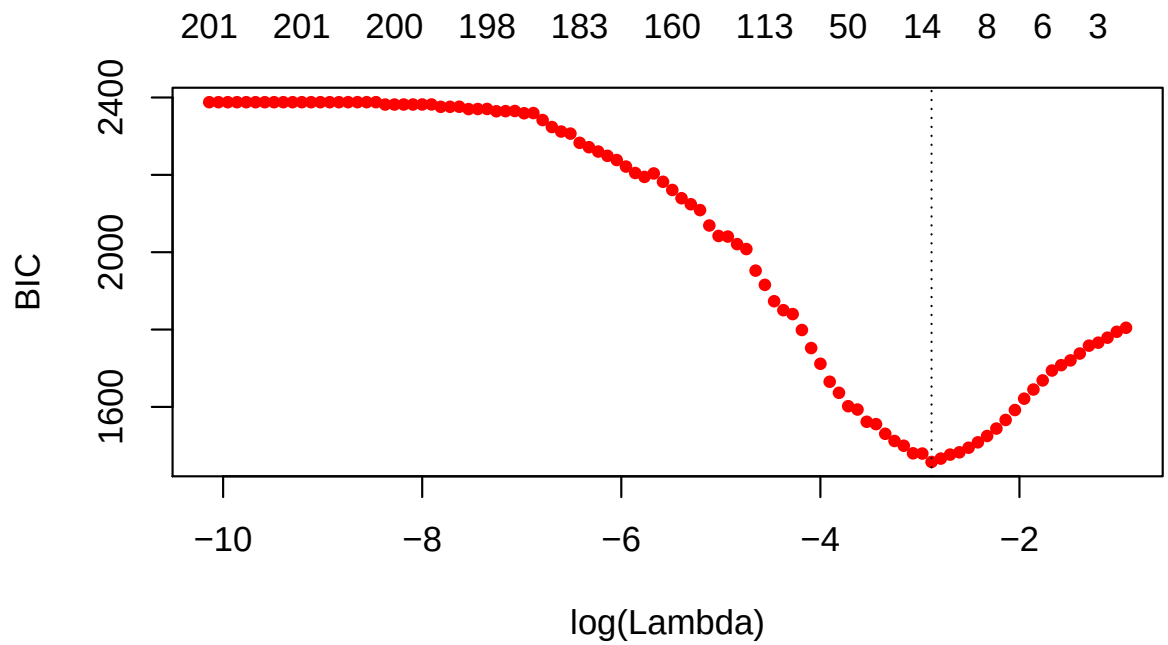


The optimal value for λ according to the HDBIC, i.e., the λ that leads to the minium HDBIC is:

```
hdbic[["lambda.min"]]

## [1] 0.05596623
```

We can also plot the BIC results:

```
plot(bicfit, ylab = "BIC")
```




```
bicfit[["lambda.min"]]

## [1] 0.05596623
```

C.4 Get Coefficients Corresponding to Optimal Model

We can use the object outputted by the `gic` function to extract the coefficients corresponding to the selected model using the `coef` method for objects of class `ggmix_gic`:

```
coef(hdbic)[1:5, , drop = FALSE]

## 5 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) -0.2668419
## X62          .
## X185         .
## X371         .
## X420         .
```

We can also extract just the nonzero coefficients which also provide the estimated variance components η and σ^2:

```
coef(hdbic, type = "nonzero")

##                       1
## (Intercept) -0.26684191
## X336        -0.67986393
## X7638        0.43403365
## X1536        0.93994982
## X1943        0.56600730
## X2849       -0.58157979
## X56         -0.08244685
## X4106       -0.35939830
## eta          0.26746240
## sigma2       0.98694300
```

We can also make predictions from the `hdbic` object, which by default will use the model corresponding to the optimal tuning parameter:

```
predict(hdbic, newx = admixed$x[1:5,])

##              1
## id1 -1.3061041
## id2  0.2991654
## id3 -2.3453664
## id4 -0.4486012
## id5 -1.3895793
```

C.5 Extracting Random Effects

The user can compute the random effects using the provided `ranef` method for objects of class `ggmix_gic`. This command will compute the estimated random effects for each subject using the parameters of the selected model:

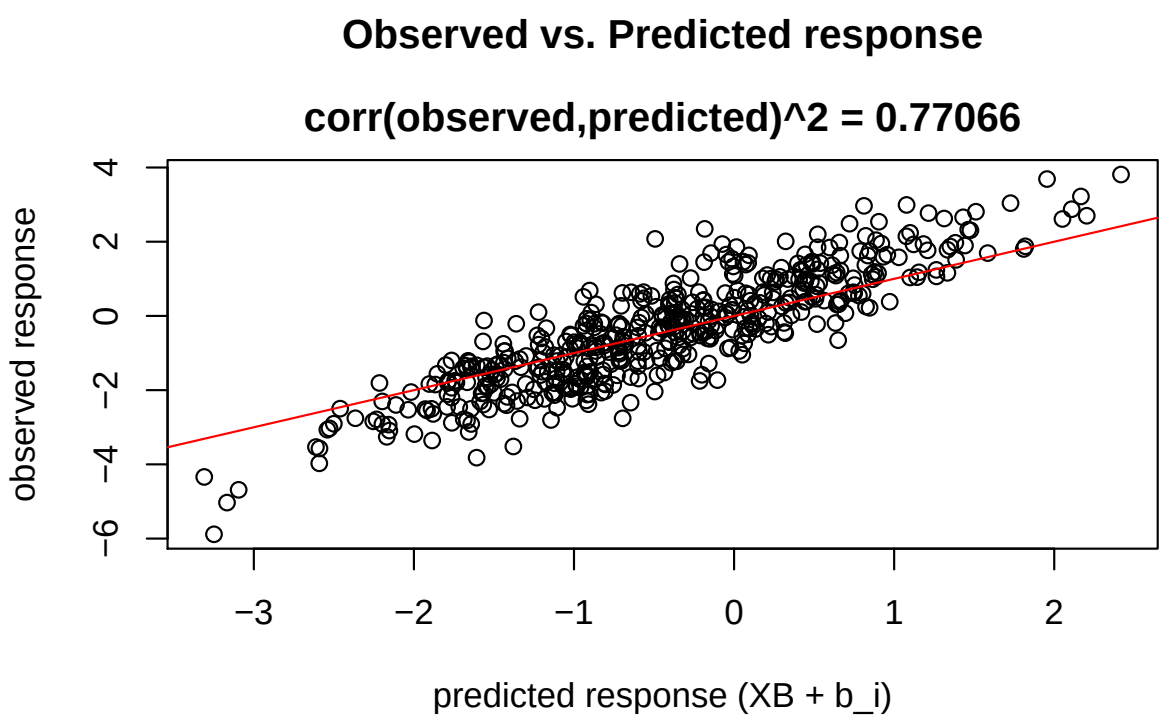
```
ranef(hdbic)[1:5]

## [1] -0.02548691 -0.10011680  0.13020240 -0.30650997  0.16045768
```

C.6 Diagnostic Plots

We can also plot some standard diagnotic plots such as the observed vs. predicted response, QQ-plots of the residuals and random effects and the Tukey-Anscombe plot. These can be plotted using the `plot` method on a `ggmix_gic` object as shown below.

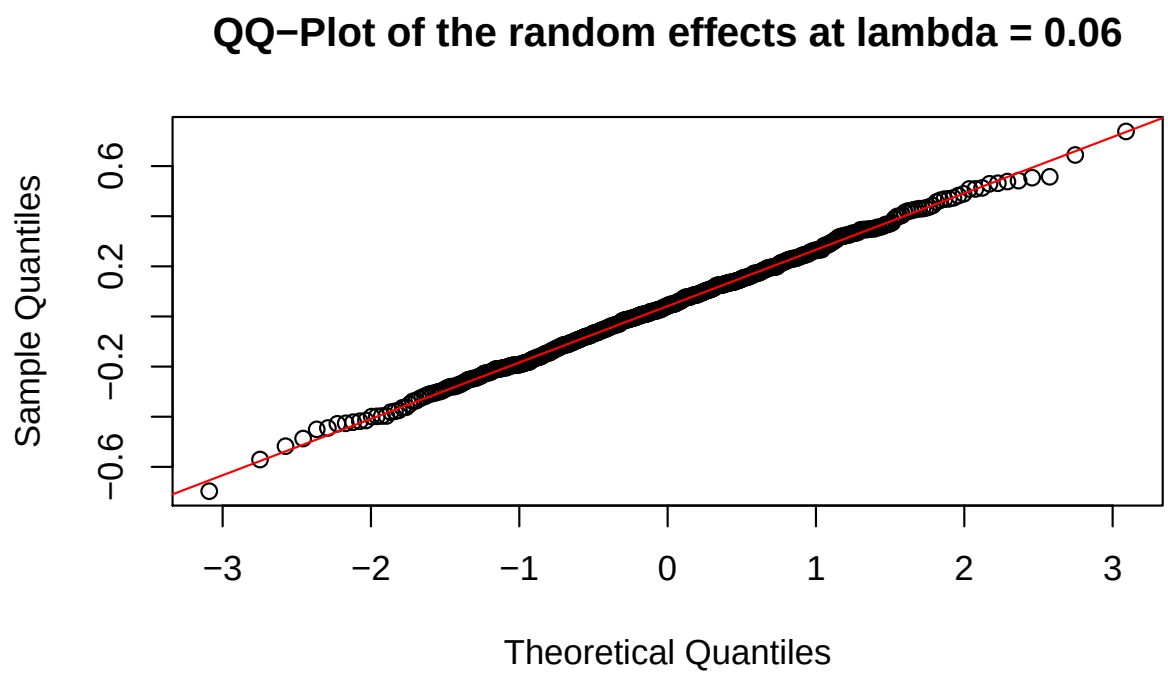
C.6.1 Observed vs. Predicted Response

```
plot(hdbic, type = "predicted", newx = admixed$x, newy = admixed$y)
```

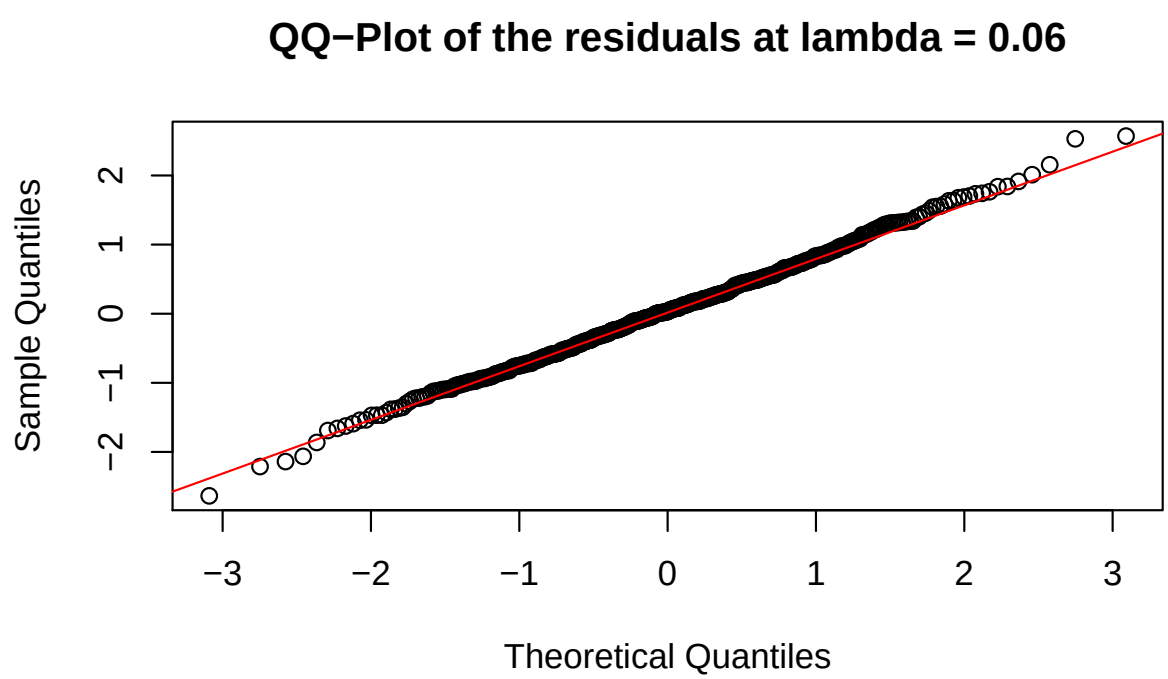


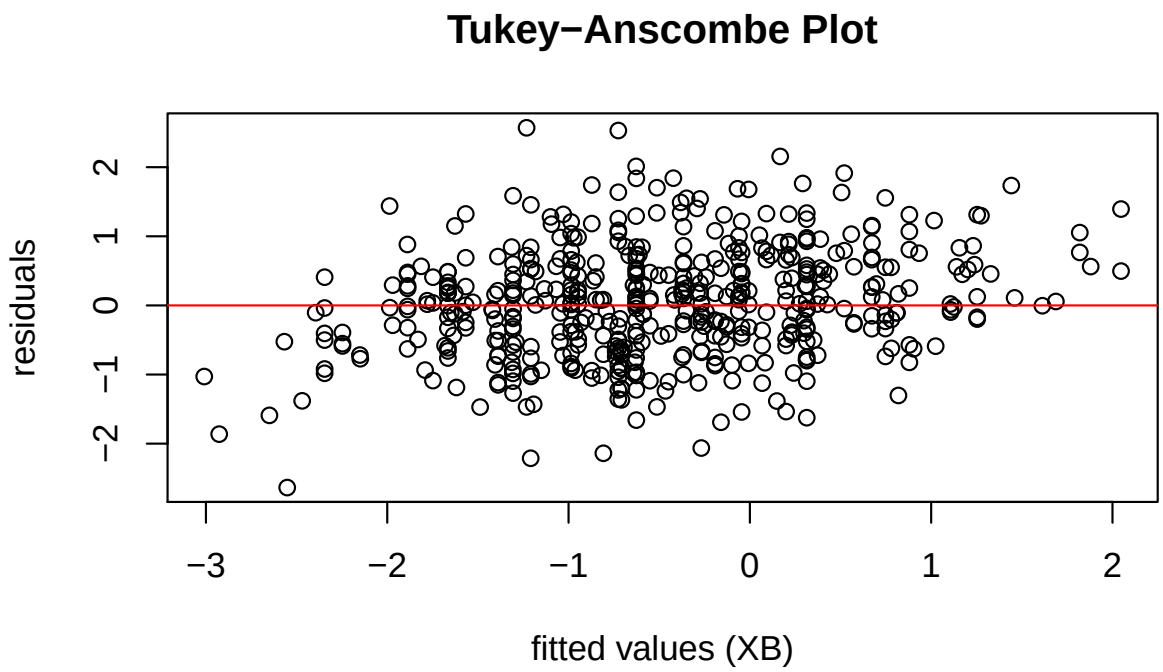
C.6.2 QQ-plots for Residuals and Random Effects

```
plot(hdbic, type = "QQranef", newx = admixed$x, newy = admixed$y)
```




```
plot(hdbic, type = "QQresid", newx = admixed$x, newy = admixed$y)
```



C.6.3 Tukey-Anscombe Plot

```
plot(hdbic, type = "Tukey", newx = admixed$x, newy = admixed$y)
```