Simultaneous SNP selection and adjustment for population structure in high dimensional prediction models

Sahir R Bhatnagar[1,2], Yi Yang[4], Tianyuan Lu[2], Erwin Schurr[6], JC Loredo-Osti[7], Marie Forest[2], Karim Oualkacha[3], 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

[6]Department of Medicine, McGill University

[7]Department of Mathematics and Statistics, Memorial University

December 6, 2019

Abstract

Complex traits are known to be influenced by a combination of environmental fac-

tors 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 effects 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 or false negatives 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` for simultaneous SNP selection and adjustment for population structure in high dimensional prediction models. 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 and two real data examples, we show that `ggmix` leads to better sensitivity and specificity compared to the two-stage approach or principal component adjustment with better prediction accuracy. `ggmix` can be used to construct polygenic risk scores and select instrumental variables in Mendelian randomization studies. Our algorithms are available in an R package (https://github.com/greenwoodlab/ggmix).

1 Author Summary

This work addresses a recurring challenge in the analysis and interpretation of genetic association studies: which genetic variants can best predict and are independently associated with a given phenotype in the presence of population structure ? Not controlling confounding due to geographic population structure, family and/or cryptic relatedness can lead to spurious associations. Much of the existing research has therefore focused on modeling the

association between a phenotype and a single genetic variant in a linear mixed model with a random effect. However, this univariate approach may miss true associations due to the stringent significance thresholds required to reduce the number of false positives and also ignores the correlations between markers. We propose an alternative method for fitting high-dimensional multivariable models, which selects SNPs that are independently associated with the phenotype while also accounting for population structure. We provide an efficient implementation of our algorithm and show through simulation studies and real data examples that our method outperforms existing methods in terms of prediction accuracy and controlling the false discovery rate.

2 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 [**?**]. One plausible reason is that there are many causal variants that each explain a small amount of variation with small effect sizes [**?**]. 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 [**?**]. 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 [**?**]. 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 [**?** **?**]. Studies that separate

their sample by ethnicity to address this confounding suffer from a loss in statistical power due to the drop in sample size.

To address the first problem, multivariable regression methods have been proposed which simultaneously fit many SNPs in a single model [**?** **?**]. 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 [**?**].

Solutions for confounding by population structure have also received significant attention in the literature [**?** **?** **?** **?**]. 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 [**?**]. The LMM uses an estimated covariance matrix from the individuals' genotypes and includes this information in the form of a random effect [**?**].

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 [**?**]. In case-control data, a stepwise logistic-regression procedure was used to evaluate the relative importance of variants within a small genetic region [**?**]. 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 [**?**] places a Laplace prior on all main effects while the adaptive mixed lasso [**?**] uses the L_1 penalty [**?**] with adaptively

chosen weights [**?**] 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 [**?**]. 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` [**?**] and `gglasso` [**?**], 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 [**? ?**].

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 [**?**], elastic net [**?**] and group lasso [**?**]. `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 (https://github.com/greenwoodlab/ggmix). We provide a brief demonstration of the `ggmix` package in Appendix C.

The rest of the paper is organized as follows. In Section 3, 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 and two real data analyses. This is followed by a discussion of our results, some limitations and future directions in Section 4. Section 5 describes the `ggmix` model, the optimization procedure and the algorithm used to fit it.

3 Results

In this section we demonstrate the performance of `ggmix` in a simulation study and two real data applications.

3.1 Simulation Study

We evaluated the performance of `ggmix` in a variety of simulated scenarios. For each simulation scenario we compared `ggmix` to the `lasso` and the `twostep` method. For the `lasso`, we included the top 10 principal components from the simulated genotypes used to calculate the kinship matrix as unpenalized predictors in the design matrix. For the `twostep` method, we first fitted an intercept only model with a single random effect using the average information restricted maximum likelihood (AIREML) algorithm [**?**] as implemented in the `gaston` R package [**?**]. The residuals from this model were then used as the response in a regular `lasso` model. Note that in the `twostep` method, we removed the kinship effect in the first step and therefore did not need to make any further adjustments when fitting the penalized model. We fitted the `lasso` using the default settings and `standardize=FALSE` in the `glmnet` package [**?**]. For other parameters in our simulation study, we defined the following quantities:

- n: sample size

- c: percentage of causal SNPs
- $\boldsymbol{\beta}$: true effect size vector of length p_{fixed}
- $S_0 = \{j; (\boldsymbol{\beta})_j \neq 0\}$ the index of the true active set with cardinality $|S_0| = c \times p_{fixed}$
- $\mathbf{X}^{(fixed)}$: $n \times p_{fixed}$ matrix of SNPs that were included as fixed effects in the model
- $\mathbf{X}^{(causal)}$: $n \times |S_0|$ matrix of SNPs that were 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 were used in the construction of the kinship matrix. Some of these $\mathbf{X}^{(other)}$ SNPs, in conjunction with some of the SNPs in $\mathbf{X}^{(fixed)}$ were used in construction of the kinship matrix. We altered 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

We simulated data from the model

$$\mathbf{Y} = \mathbf{X}^{(fixed)}\boldsymbol{\beta} + \mathbf{P} + \boldsymbol{\varepsilon} \tag{1}$$

where $\mathbf{P} \sim \mathcal{N}(0, \eta\sigma^2\boldsymbol{\Phi})$ is the polygenic effect and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, (1-\eta)\sigma^2\mathbf{I})$ is the error term. Here, $\boldsymbol{\Phi}_{n\times n}$ is the covariance matrix calculated from $\mathbf{X}^{(kinship)}$, $\mathbf{I}_{n\times n}$ is the identity matrix and parameters σ^2 and $\eta \in [0, 1]$ determine how the variance is divided between $\mathbf{P}$ and $\boldsymbol{\varepsilon}$. The values of the parameters that we used were as follows: narrow sense heritability $\eta = \{0.1, 0.3\}$, number of fixed effects $p_{fixed} = 5,000$, number of SNPs used to calculate the kinship matrix $k = 10,000$, 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 were 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 [**?**]. 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 simulated random genotypes from the BN-PSD admixture model with 1D geography and 10 subpopulations using the `bnpsd` package [**?** **?**]. In Figure **??**, we plot the estimated kinship matrix from a single simulated dataset in the form of a heatmap where a darker color indicates a closer genetic relationship.

In Figure **??** we plot the first two principal component scores calculated from the simulated genotypes used to calculate the kinship matrix in Figure **??**, 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.

Using this set-up, we randomly partitioned 1000 simulated observations into 80% for training and 20% for testing. The training set was used to fit the model and select the optimal tuning parameter only, and the resulting model was evaluated on the test set. Let $\hat{\lambda}$ be the estimated value of the optimal regularization parameter, $\widehat{\boldsymbol{\beta}}_{\hat{\lambda}}$ the estimate of $\boldsymbol{\beta}$ at regularization

parameter $\hat{\lambda}$, and $\widehat{S}_{\hat{\lambda}} = \left\{ j; (\widehat{\boldsymbol{\beta}}_{\hat{\lambda}})_j \neq 0 \right\}$ the index of the set of non-zero estimated coefficients. We evaluated the methods based on correct sparsity defined as $\frac{1}{p}\sum_{j=1}^{p} A_j$, where

$$A_j = \begin{cases} 1 & \text{if } (\widehat{\boldsymbol{\beta}}_{\hat{\lambda}})_j = (\boldsymbol{\beta})_j = 0 \\ 1 & \text{if } (\widehat{\boldsymbol{\beta}}_{\hat{\lambda}})_j \neq 0, (\boldsymbol{\beta})_j \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We also compared the test set prediction error based on the refitted unpenalized estimates for each selected model, the estimation error ($||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}||_2^2$), 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 (η, σ^2) for the polygenic random effect and error term.

In Figure **??**, we present the results for the scenario with 1% causal SNPs ($c = 0.01$) which were all used in the calculation of the kinship matrix and true heritability η =10%. The complete simulation results are shown in supplementary Section B. We see that `ggmix` outperformed both the `twostep` and `lasso` in terms of correct sparsity and estimation error (Figure **??** panels A and B). This was true regardless of true heritability and whether the causal SNPs were included in the calculation of the kinship matrix (Figures **??**, **??**, **??** and **??**). Across all simulation scenarios, `ggmix` had the smallest root mean squared prediction error (RMSE) on the test set while also producing the most parsimonious models (Figures **??** panel B, **??** and **??**). Both the `lasso` and `twostep` had on average, slightly higher true positive rate compared to `ggmix` but came at the cost of a higher false positive rate (Figures **??** panel D, **??** and **??**). Both the `twostep` and `ggmix` overestimated the heritability though `ggmix` was closer to the true value (Figure **??** panel E). When none of the causal SNPs were in the kinship, both methods tended to overestimate the truth when $\eta = 10\%$ and underestimate when $\eta = 30\%$ (Figure **??**). Across all simulation scenarios `ggmix` was able to (on average) correctly estimate the error variance (Figures **??** panel F, **??** and **??**). The `lasso` tended to overestimate σ^2 in the null model while the `twostep`

overestimated σ^2 when none of the causal SNPs were in the kinship matrix.

Overall, we observed that variable selection results and RMSE for `ggmix` were similar regardless of whether the causal SNPs were 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, particularly in polygenic traits. `ggmix` had very good Type 1 and II error control, while both the `lasso` and `twostep` had a very high false positive rate in all simulation scenarios. In particular, our simulation results show that the principal component adjustment method may not be the best approach to control for confounding by population structure, particularly when variable selection is of interest.

3.2 Real Data Applications

Two datasets with contrasting features are used to illustrate the potential advantages of `ggmix` over existing approaches such as PC adjustment in a `lasso` regression. In one dataset, family structure induces low levels of correlation and sparsity in signals. In the second, a dataset involving mouse crosses, correlations are extremely strong and can confound signals.

3.2.1 GAW20

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) [**?**]. Penalized regression methods were 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, `twostep` methods were used which could have resulted in a large number of false positives or false negatives. `ggmix` is an alternative approach that

Table 1: Mean (standard deviation) from 200 simulations stratified by the number of causal SNPs (null, 1%), the overlap between causal SNPs and kinship matrix (no overlap, all causal SNPs in kinship), and true heritability (10%, 30%). For all simulations, sample size is $n = 1000$, the number of fixed effects is $p_{fixed} = 5000$, and the number of SNPs used to estimate the kinship matrix is $k = 10000$. TPR at FPR=5% is the true positive rate at a fixed false positive rate of 5%. Model Size is the number of selected variables in the training set using the high-dimensional BIC for `ggmix` and 10-fold cross validation for `lasso` and `twostep`. RMSE is the root mean squared error on the test set. Estimation error is the squared distance between the estimated and true effect sizes. Error variance (σ^2) for `twostep` is estimated from an intercept only LMM with a single random effect and is modeled explicitly in `ggmix`. For the `lasso` we use $\frac{1}{n-|\widehat{S}_{\hat{\lambda}}|}||\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{\hat{\lambda}}||_2^2$ [**?**] as an estimator for σ^2. Heritability (η) for `twostep` is estimated as $\sigma_g^2/(\sigma_g^2 + \sigma_e^2)$ from an intercept only LMM with a single random effect where σ_g^2 and σ_e^2 are the variance components for the random effect and error term, respectively. η is explictly modeled in `ggmix`. There is no positive way to calculate η for the `lasso` since we are using a PC adjustment.

		Null model				1% Causal SNPs			
		No overlap		All causal SNPs in kinship		No overlap		All causal SNPs in kinship	
Metric	Method	10%	30%	10%	30%	10%	30%	10%	30%
TPR at FPR=5%	twostep	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.84 (0.05)	0.84 (0.05)	0.76 (0.09)	0.77 (0.08)
	lasso	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.86 (0.05)	0.85 (0.05)	0.86 (0.05)	0.86 (0.05)
	ggmix	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.86 (0.05)	0.86 (0.05)	0.85 (0.05)	0.86 (0.05)
Model Size	twostep	7 (15)	5 (12)	7 (15)	5 (12)	338 (71)	339 (68)	289 (62)	286 (55)
	lasso	7 (14)	7 (17)	7 (14)	7 (17)	282 (51)	281 (52)	286 (51)	284 (54)
	ggmix	0 (0)	0 (0)	0 (0)	0 (0)	44 (8)	44 (8)	44 (8)	43 (9)
RMSE	twostep	1.02 (0.07)	1.02 (0.06)	1.02 (0.07)	1.02 (0.06)	1.42 (0.10)	1.41 (0.10)	1.44 (0.33)	1.40 (0.22)
	lasso	1.02 (0.06)	1.02 (0.06)	1.02 (0.06)	1.02 (0.06)	1.39 (0.09)	1.38 (0.09)	1.40 (0.08)	1.38 (0.08)
	ggmix	1.00 (0.05)	1.00 (0.05)	1.00 (0.05)	1.00 (0.05)	1.22 (0.10)	1.20 (0.10)	1.23 (0.11)	1.23 (0.12)
Estimation Error	twostep	0.12 (0.22)	0.09 (0.19)	0.12 (0.22)	0.09 (0.19)	2.97 (0.60)	2.92 (0.60)	3.60 (5.41)	3.21 (3.46)
	lasso	0.13 (0.21)	0.12 (0.22)	0.13 (0.21)	0.12 (0.22)	2.76 (0.46)	2.69 (0.47)	2.82 (0.48)	2.75 (0.48)
	ggmix	0.00 (0.01)	0.01 (0.02)	0.00 (0.01)	0.01 (0.02)	2.11 (1.28)	2.04 (1.22)	2.21 (1.24)	2.28 (1.34)
Error Variance	twostep	0.87 (0.11)	0.69 (0.15)	0.87 (0.11)	0.69 (0.15)	14.23 (3.53)	14.13 (3.52)	1.42 (1.71)	1.28 (1.66)
	lasso	0.98 (0.05)	0.96 (0.05)	0.98 (0.05)	0.96 (0.05)	1.04 (0.13)	1.02 (0.13)	1.03 (0.14)	1.01 (0.14)
	ggmix	0.85 (0.18)	0.64 (0.20)	0.85 (0.18)	0.64 (0.20)	2.00 (0.49)	1.86 (0.51)	1.06 (0.46)	0.83 (0.45)
Heritability	twostep	0.13 (0.11)	0.31 (0.15)	0.13 (0.11)	0.31 (0.15)	0.26 (0.14)	0.26 (0.14)	0.92 (0.08)	0.93 (0.08)
	lasso	–	–	–	–	–	–	–	–
	ggmix	0.15 (0.18)	0.37 (0.21)	0.15 (0.18)	0.37 (0.21)	0.18 (0.16)	0.23 (0.17)	0.59 (0.20)	0.68 (0.19)

could be used for selecting the IV while accounting for the family structure of the data.

We applied `ggmix` to all 200 GAW20 simulation datasets, each of 679 observations, and compared its performance to the `twostep` and `lasso` methods. Using a FaST-LMM (Factored Spectrally Transformed Linear Mixed Model) [**?**], we validated the effect of rs9661059 on blood lipid trait to be significant (genome-wide $p = 6.29 \times 10^{-9}$). Though several other SNPs are also associated with the phenotype, these associations are probably mediated by CpG-SNP interaction pairs and do not reach statistical significance. Therefore, to avoid ambiguity, we only focused on chromosome 1 containing 51,104 SNPs where rs9661059 resides. Given that population admixture in the GAW20 data is likely, we estimated the population kinship using REAP [**?**] after decomposing population compositions using ADMIXTURE [**?**]. We supplied the estimated kinship matrix directly to `ggmix` . For both the `lasso` and `twostep` methods, we adopted the same strategies as described in our simulation study in section 3.1, supplying the same kinship matrix estimated by REAP.

On each simulated replicate, we calibrated the methods so that they could be easily compared by fixing the true positive rate to 1 and then minimizing the false positive rate. Hence, the selected SNP, rs9661059, is likely to be the true positive for each method, and non-causal SNPs are excluded to the greatest extent. All of the three mothods precisely choose the correct predictor without any false positives in more than half of the replicates, given the strong causal signal. When some false positives are selected, `ggmix` performs comparably to `twostep`, and the `lasso` tends to select more false positives (Figure **??**). In terms of phenotype prediction, we observed that `ggmix` outperforms the `twostep` method without requiring more SNPs, while it achieves roughly the same prediction accuracy as `lasso` but with fewer non-causal SNPs (Figure **??**).

3.2.2 Mouse Crosses and Sensitivity to Mycobacterial Infection

Mouse inbred strains of genetically identical individuals are extensively used in research. Crosses of different inbred strains are useful for various studies of heritability focusing on either observable phenotypes or molecular mechanisms, and in particular, recombinant congenic strains have been an extremely useful resource for many years [**?**]. However, ignoring complex genetic relationships in association studies can lead to inflated false positives in genetic association studies when different inbred strains and their crosses are investigated [**?** **?** **?**]. Therefore, a previous study developed and implemented a mixed model to find loci associated with mouse sensitivity to mycobacterial infection [**?**]. The random effects in the model captured complex correlations between the recombinant congenic mouse strains based on the proportion of the DNA shared identical by descent. Through a series of mixed model fits at each marker, new loci that impact growth of mycobacteria on chromosome 1 and chromosome 11 were identified.

Here we show that `ggmix` can identify these loci, as well as potentially others, in a single analysis. We reanalyzed the growth permissiveness in the spleen, as measured by colony forming units (CFUs), 6 weeks after infection from *Mycobacterium bovis* Bacille Calmette-Guerin (BCG) Russia strain as reported in [**?**].

By taking the consensus between the "main model" and the "conditional model" of the original study, we regarded markers D1Mit435 on chromosome 1 and D11Mit119 on chromosome 11 as two true positive loci. Similar to the strategy used when analyzing the GAW20 data, we optimized models by tuning the penalty factor such that these two loci are picked up, while the number of other active loci is minimized. To evaluate robustness of different models, we bootstrapped the 189-sample dataset and repeated the analysis 200 times. We directly estimated the kinship between mice using genotypes at 625 microsatellite markers. The estimated kinship entered directly into `ggmix` and `twostep`. For the `lasso`, we calculated and included the first 10 principal components of the estimated kinship. Significant

markers are defined as those captured in at least half of the bootstrap replicates, and in which the corresponding method successfully captures both pre-selected true positives with a penalty factor minimizing the number of active loci (Figure **??**).

We demonstrate that `ggmix` recognizes the true associations more robustly than `twostep` and `lasso`. In almost all (99%) bootstrap replicates, `ggmix` is able to capture both true positives, while `twostep` failed in 19% of the replicates and `lasso` failed in 56% of the replicates by missing of at least one of the two true positives (Figure **??**). We also identified several other loci that might also be associated with susceptibility to mycobacterial infection (Table 2). Among these new potentially-associated markers, D2Mit156 was found to play a role in control of parasite numbers of *Leishmania tropica* in lymph nodes [**?**]. This locus is considered significant by our definition for both `ggmix` and `lasso`. An earlier study identified a parent-of-origin effect at D17Mit221 on CD4M levels [**?**]. This effect was more visible in crosses than in parental strains. In addition, D14Mit131, selected only by `ggmix` , was found to have a 9% loss of heterozygosity in hybrids of two inbred mouse strains [**?**], indicating the potential presence of putative suppressor genes pertaining to immune surveillance and tumor progression [**?**]. This result might also suggest association with anti-bacterial responses yet to be discovered.

Table 2: Additional loci significantly associated with mouce susceptibility to myobacterial infection, after excluding two true positives. Loci needed to be identified in at least 50% of the successful bootstrap replicates that captured both true positive loci.

Method	Marker	Position in cM	Position in bp
`twostep`	N/A	N/A	N/A
`lasso`	D2Mit156	Chr2:31.66	Chr2:57081653-57081799
	D14Mit155	Chr14:31.52	Chr14:59828398-59828596
ggmix	D2Mit156	Chr2:31.66	Chr2:57081653-57081799
	D14Mit131	Chr14:63.59	Chr14:120006565-120006669
	D17Mit221	Chr17:59.77	Chr17:90087704-90087842

4 Discussion

We have developed a general penalized LMM framework called `ggmix` which simultaneously selects SNPs and adjusts for population structure in high dimensional prediction models. Through an extensive simulation study and two real data analyses, we show that the current approaches of PC adjustment and two-stage procedures are not necessarily sufficient to control for confounding by population structure leading to a high number of false positives or false negatives. Furthermore, `ggmix` showed improved prediction performance with a more parsimonious model compared to both the `lasso` and `twostep`. Our proposed method has excellent Type 1 error control and is robust to the inclusion of causal SNPs in the kinship matrix. Many methods for single-SNP analyses avoid this "proximal contamination" [**?**] by using a leave-one-chromosome-out scheme [**?**], i.e., construct the kinship matrix using all chromosomes except the one on which the marker being tested is located. However, this approach is not possible if we want to model many SNPs (across many chromosomes) jointly. We also demonstrated `ggmix` using two examples that mimic many experimental designs in

genetics. In the GAW20 example, we showed that while all methods were able to select the strongest causal SNP, `ggmix` did so with the least amount of false positives while also maintaining good predictive ability. In the mouse crosses example, we showed that `ggmix` is robust to perturbations in the data using a bootstrap analysis. Indeed, `ggmix` was able to consistently select the true positives across bootstrap replicates, while `twostep` failed in 19% of the replicates and `lasso` failed in 56% of the replicates by missing of at least one of the two true positives. Our re-analysis of the data also lead to some potentially new findings, not found by existing methods, that may warrant further study.

We emphasize here that previously developed methods such as the LMM-lasso [**?**] 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 the tuning parameter. To our knowledge, we are the first to develop a coordinate gradient descent (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.

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 [**?**] 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 [**?**], 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 $\mathcal{O}(nk^2)$.

There are other applications in which our method could be used as well. For example, there has been a renewed interest in polygenic risk scores (PRS) which aim to predict complex diseases from genotypes. `ggmix` could be used to build a PRS with the distinct advantage of modeling SNPs jointly, allowing for main effects as well as interactions to be accounted for. Based on our results, `ggmix` has the potential to produce more robust and parsimonious models than the `lasso` with better predictive accuracy. 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 [**?**].

5 Materials and Methods

5.1 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 [**?**]:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{b} + \boldsymbol{\varepsilon} \tag{3}$$

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

$$\boldsymbol{b} \sim \mathcal{N}(0, \eta\sigma^2\boldsymbol{\Phi}) \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, (1-\eta)\sigma^2\mathbf{I}) \tag{4}$$

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 [**?**]. The joint density of $\mathbf{Y}$ is therefore multivariate normal:

$$\mathbf{Y}|(\boldsymbol{\beta}, \eta, \sigma^2) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \eta\sigma^2\boldsymbol{\Phi} + (1-\eta)\sigma^2\mathbf{I}) \tag{5}$$

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

$$\mathbf{Y}|(\boldsymbol{\beta}, \delta, \sigma_g^2) \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma_g^2(\boldsymbol{\Phi} + \delta\mathbf{I})) \tag{6}$$

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 (5) since maximization is easier over the compact set $\eta \in [0, 1]$ than over the unbounded interval $\delta \in [0, \infty)$ [**?**]. We define the complete parameter vector as $\boldsymbol{\Theta} \coloneqq (\boldsymbol{\beta}, \eta, \sigma^2)$. The negative log-likelihood for (5) is given by

$$-\ell(\boldsymbol{\Theta}) \propto \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\log\left(\det(\mathbf{V})\right) + \frac{1}{2\sigma^2}\left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right)^T \mathbf{V}^{-1}\left(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\right) \tag{7}$$

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

Let $\mathbf{\Phi} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ be the eigen (spectral) decomposition of the kinship matrix $\mathbf{\Phi}$, 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 [**?**]

$$\begin{aligned}
\mathbf{V} &= \eta\mathbf{\Phi} + (1-\eta)\mathbf{I} \\
&= \eta\mathbf{U}\mathbf{D}\mathbf{U}^T + (1-\eta)\mathbf{U}\mathbf{I}\mathbf{U}^T \\
&= \mathbf{U}\eta\mathbf{D}\mathbf{U}^T + \mathbf{U}(1-\eta)\mathbf{I}\mathbf{U}^T \\
&= \mathbf{U}\left(\eta\mathbf{D} + (1-\eta)\mathbf{I}\right)\mathbf{U}^T \\
&= \mathbf{U}\widetilde{\mathbf{D}}\mathbf{U}^T
\end{aligned} \tag{8}$$

where

$$\widetilde{\mathbf{D}} = \eta\mathbf{D} + (1-\eta)\mathbf{I} \tag{9}$$

$$\begin{aligned}
&= \eta \begin{bmatrix} \Lambda_1 & & & \\ & \Lambda_2 & & \\ & & \ddots & \\ & & & \Lambda_{N_T} \end{bmatrix} + (1-\eta) \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1+\eta(\Lambda_1-1) & & & \\ & 1+\eta(\Lambda_2-1) & & \\ & & \ddots & \\ & & & 1+\eta(\Lambda_{N_T}-1) \end{bmatrix} \\
&= \operatorname{diag}\left\{1+\eta(\Lambda_1-1), 1+\eta(\Lambda_2-1), \ldots, 1+\eta(\Lambda_{N_T}-1)\right\}
\end{aligned} \tag{10}$$

Since (9) is a diagonal matrix, its inverse is also a diagonal matrix:

$$\widetilde{\mathbf{D}}^{-1} = \text{diag}\left\{\frac{1}{1+\eta(\Lambda_1-1)}, \frac{1}{1+\eta(\Lambda_2-1)}, \ldots, \frac{1}{1+\eta(\Lambda_{N_T}-1)}\right\} \tag{11}$$

From (8) and (10), $\log(\det(\mathbf{V}))$ simplifies to

$$\begin{aligned}\log(\det(\mathbf{V})) &= \log\left(\det(\mathbf{U})\det\left(\widetilde{\mathbf{D}}\right)\det(\mathbf{U}^T)\right) \\ &= \log\left\{\prod_{i=1}^{N_T}(1+\eta(\Lambda_i-1))\right\} \\ &= \sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1))\end{aligned} \tag{12}$$

since $\det(\mathbf{U}) = 1$. It also follows from (8) that

$$\begin{aligned}\mathbf{V}^{-1} &= \left(\mathbf{U}\widetilde{\mathbf{D}}\mathbf{U}^T\right)^{-1} \\ &= \left(\mathbf{U}^T\right)^{-1}\left(\widetilde{\mathbf{D}}\right)^{-1}\mathbf{U}^{-1} \\ &= \mathbf{U}\widetilde{\mathbf{D}}^{-1}\mathbf{U}^T\end{aligned} \tag{13}$$

since for an orthonormal matrix $\mathbf{U}^{-1} = \mathbf{U}^T$. Substituting (11), (12) and (13) into (7) the negative log-likelihood becomes

$$-\ell(\boldsymbol{\Theta}) \propto \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) + \frac{1}{2\sigma^2}(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta})^T\mathbf{U}\widetilde{\mathbf{D}}^{-1}\mathbf{U}^T(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}) \tag{14}$$

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

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

$$= \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) + \frac{1}{2\sigma^2}\sum_{i=1}^{N_T}\frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p}\widetilde{X}_{ij+1}\beta_j\right)^2}{1+\eta(\Lambda_i-1)} \tag{15}$$

where $\widetilde{\mathbf{Y}} = \mathbf{U}^T\mathbf{Y}$, $\widetilde{\mathbf{X}} = \mathbf{U}^T\mathbf{X}$, $\widetilde{Y}_i$ denotes the i^{th} 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.

5.2 Penalized Maximum Likelihood Estimator

We define the $p+3$ length vector of parameters $\boldsymbol{\Theta} \coloneqq (\Theta_0, \Theta_1, \ldots, \Theta_{p+1}, \Theta_{p+2}, \Theta_{p+3}) = (\boldsymbol{\beta}, \eta, \sigma^2)$ 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 (15). 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:

$$Q_\lambda(\boldsymbol{\Theta}) = f(\boldsymbol{\Theta}) + \lambda \sum_{j \neq 0} v_j P_j(\beta_j) \tag{16}$$

where $f(\boldsymbol{\Theta}) \coloneqq -\ell(\boldsymbol{\Theta})$ is defined in (15), $P_j(\cdot)$ 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

$$\widehat{\boldsymbol{\Theta}}_\lambda = \arg\min_{\boldsymbol{\Theta}} Q_\lambda(\boldsymbol{\Theta}) \tag{17}$$

This is the general set-up for our model. In Section 5.3 we provide more specific details on how we solve (17).

5.3 Computational Algorithm

We use a general purpose block coordinate gradient descent algorithm (CGD) [**?**] to solve (17). 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 [**?**] 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. [**?**], [**?**]) and linear mixed models with an ℓ_1 penalty [**?**]. In the next section we provide some brief details about Algorithm 1. A more thorough treatment of the algorithm is given in Appendix A.

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

5.3.1 Updates for the β parameter

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

$$Q_\lambda(\boldsymbol{\Theta}) = \frac{1}{2}\sum_{i=1}^{N_T} w_i \left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} v_j|\beta_j| \tag{18}$$

where

$$w_i \coloneqq \frac{1}{\sigma^2\,(1+\eta(\Lambda_i - 1))} \tag{19}$$

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

$$\beta_j^{(k+1)} \leftarrow \frac{\mathcal{S}_\lambda\left(\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell \neq j} \widetilde{X}_{i\ell}\beta_\ell^{(k)}\right)\right)}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2} \tag{20}$$

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

5.3.2 Updates for the η paramter

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

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

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

5.3.3 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:

$$\sigma^{2\,(k+1)} \leftarrow \arg\min_{\sigma^2} \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j\right)^2}{1+\eta(\Lambda_i-1)} \tag{22}$$

There exists an analytic solution for (22) given by:

$$\sigma^{2\,(k+1)} \leftarrow \frac{1}{N_T}\sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1}\beta_j^{(k+1)}\right)^2}{1+\eta^{(k+1)}(\Lambda_i-1)} \tag{23}$$

5.3.4 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

$$Q_\lambda(\boldsymbol{\Theta}) = \frac{N_T}{2}\log(\sigma^2) + \frac{1}{2}\sum_{i=1}^{N_T}\log(1+\eta(\Lambda_i-1)) + \frac{1}{2}\sum_{i=1}^{N_T} w_i \left(\widetilde{Y}_i - \sum_{j=0}^{p}\widetilde{X}_{ij+1}\beta_j\right)^2 + \lambda\sum_{j=1}^{p} v_j|\beta_j| \tag{24}$$

411 The Karush-Kuhn-Tucker (KKT) optimality conditions for (24) are given by:

$$\begin{aligned} \frac{\partial}{\partial \beta_1, \ldots, \beta_p} Q_\lambda(\boldsymbol{\Theta}) &= \mathbf{0}_p \\ \frac{\partial}{\partial \beta_0} Q_\lambda(\boldsymbol{\Theta}) &= 0 \\ \frac{\partial}{\partial \eta} Q_\lambda(\boldsymbol{\Theta}) &= 0 \\ \frac{\partial}{\partial \sigma^2} Q_\lambda(\boldsymbol{\Theta}) &= 0 \end{aligned} \tag{25}$$

412 The equations in (25) are equivalent to

$$\begin{aligned} \sum_{i=1}^{N_T} w_i \widetilde{X}_{i1} \left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1} \beta_j \right) &= 0 \\ \frac{1}{v_j} \sum_{i=1}^{N_T} w_i \widetilde{X}_{ij} \left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1} \beta_j \right) &= \lambda \gamma_j, \\ \gamma_j \in \begin{cases} \text{sign}(\hat{\beta}_j) & \text{if} \quad \hat{\beta}_j \neq 0 \\ [-1, 1] & \text{if} \quad \hat{\beta}_j = 0 \end{cases}, &\qquad \text{for } j = 1, \ldots, p \\ \frac{1}{2} \sum_{i=1}^{N_T} \frac{\Lambda_i - 1}{1 + \eta(\Lambda_i - 1)} \left(1 - \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1} \beta_j \right)^2}{\sigma^2 (1 + \eta(\Lambda_i - 1))} \right) &= 0 \\ \sigma^2 - \frac{1}{N_T} \sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^{p} \widetilde{X}_{ij+1} \beta_j \right)^2}{1 + \eta(\Lambda_i - 1)} &= 0 \end{aligned} \tag{26}$$

413 where w_i is given by (19), $\widetilde{\mathbf{X}}_{-1}^T$ is $\widetilde{\mathbf{X}}^T$ with the first column removed, $\widetilde{\mathbf{X}}_1^T$ is the first column
414 of $\widetilde{\mathbf{X}}^T$, and $\boldsymbol{\gamma} \in \mathbb{R}^p$ is the subgradient function of the ℓ_1 norm evaluated at $(\hat{\beta}_1, \ldots, \hat{\beta}_p)$.
415 Therefore $\widehat{\boldsymbol{\Theta}}$ is a solution in (17) if and only if $\widehat{\boldsymbol{\Theta}}$ satisfies (26) for some γ. We can determine
416 a decreasing sequence of tuning parameters by starting at a maximal value for $\lambda = \lambda_{max}$
417 for which $\hat{\beta}_j = 0$ for $j = 1, \ldots, p$. In this case, the KKT conditions in (26) are equivalent

418 to

$$\begin{aligned}
\frac{1}{v_j}\sum_{i=1}^{N_T}\left|w_i\widetilde{X}_{ij}\left(\widetilde{Y}_i-\widetilde{X}_{i1}\beta_0\right)\right| &\leq \lambda, \quad \forall j=1,\ldots,p \\
\beta_0 &= \frac{\sum_{i=1}^{N_T} w_i\widetilde{X}_{i1}\widetilde{Y}_i}{\sum_{i=1}^{N_T} w_i\widetilde{X}_{i1}^2} \\
\frac{1}{2}\sum_{i=1}^{N_T}\frac{\Lambda_i-1}{1+\eta(\Lambda_i-1)}&\left(1-\frac{\left(\widetilde{Y}_i-\widetilde{X}_{i1}\beta_0\right)^2}{\sigma^2(1+\eta(\Lambda_i-1))}\right)=0 \\
\sigma^2 &= \frac{1}{N_T}\sum_{i=1}^{N_T}\frac{\left(\widetilde{Y}_i-\widetilde{X}_{i1}\beta_0\right)^2}{1+\eta(\Lambda_i-1)}
\end{aligned} \tag{27}$$

419 We can solve the KKT system of equations in (27) (with a numerical solution for η) in order
420 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
421 can solve for the smallest value of λ such that the entire vector $(\hat{\beta}_1,\ldots,\hat{\beta}_p)$ is 0:

$$\lambda_{max}=\max_j\left\{\left|\frac{1}{v_j}\sum_{i=1}^{N_T}\hat{w}_i\widetilde{X}_{ij}\left(\widetilde{Y}_i-\widetilde{X}_{i1}\hat{\beta}_0\right)\right|\right\}, \quad j=1,\ldots,p \tag{28}$$

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

425 5.3.5 Warm Starts

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

5.3.6 Prediction of the random effects

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

$$\widehat{\boldsymbol{b}} = \arg\max_{\boldsymbol{b}} f(\boldsymbol{b}|\mathbf{Y}, \boldsymbol{\beta}, \eta, \sigma^2) \tag{29}$$

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

$$\begin{aligned} f(\boldsymbol{b}|\mathbf{Y}, \boldsymbol{\beta}, \eta, \sigma^2) &= \frac{f(\mathbf{Y}|\boldsymbol{b}, \boldsymbol{\beta}, \eta, \sigma^2)\pi(\boldsymbol{b}|\eta, \sigma^2)}{f(\mathbf{Y}|\boldsymbol{\beta}, \eta, \sigma^2)} \\ &\propto f(\mathbf{Y}|\boldsymbol{b}, \boldsymbol{\beta}, \eta, \sigma^2)\pi(\boldsymbol{b}|\eta, \sigma^2) \\ &\propto \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b})^T\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b}) - \frac{1}{2\eta\sigma^2}\boldsymbol{b}^T\boldsymbol{\Phi}^{-1}\boldsymbol{b}\right\} \\ &= \exp\left\{-\frac{1}{2\sigma^2}\left[(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b})^T\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b}) + \frac{1}{\eta}\boldsymbol{b}^T\boldsymbol{\Phi}^{-1}\boldsymbol{b}\right]\right\} \end{aligned} \tag{30}$$

Solving for (29) is equivalent to minimizing the exponent in (30):

$$\widehat{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}} \left\{(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b})^T\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{b}) + \frac{1}{\eta}\boldsymbol{b}^T\boldsymbol{\Phi}^{-1}\boldsymbol{b}\right\} \tag{31}$$

Taking the derivative of (31) 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 (13), and $(\widehat{\boldsymbol{\beta}}, \widehat{\eta})$ are the estimates obtained from Algorithm 1.

5.3.7 Phenotype prediction

Here we describe the method used for predicting the unobserved phenotype $\mathbf{Y}^\star$ in a set of individuals with predictor set $\mathbf{X}^\star$ that were not used in the model training e.g. a testing set. Let q denote the number of observations in the testing set and $N - q$ the number of observations in the training set. We assume that a `ggmix` model has been fit on a set of training individuals with observed phenotype $\mathbf{Y}$ and predictor set $\mathbf{X}$. We further assume that $\mathbf{Y}$ and $\mathbf{Y}^\star$ are jointly multivariate Normal:

$$
\begin{bmatrix} \mathbf{Y}^\star \\ \mathbf{Y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_{1_{(q\times 1)}} \\ \boldsymbol{\mu}_{2_{(N-q)\times 1}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11_{(q\times q)}} & \boldsymbol{\Sigma}_{12_{q\times(N-q)}} \\ \boldsymbol{\Sigma}_{21_{(N-q)\times q}} & \boldsymbol{\Sigma}_{22_{(N-q)\times(N-q)}} \end{bmatrix}\right) \tag{32}
$$

Then, from standard multivariate Normal theory, the conditional distribution $\mathbf{Y}^\star|\mathbf{Y}, \eta, \sigma^2, \boldsymbol{\beta}, \mathbf{X}, \mathbf{X}^\star$ is $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}^\star)$ where

$$\boldsymbol{\mu}^{\star} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{Y} - \mu_2) \tag{33}$$

$$\boldsymbol{\Sigma}^{\star} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \tag{34}$$

The phenotype prediction is thus given by:

$$\begin{aligned}
\boldsymbol{\mu}^{\star}_{q\times 1} &= \mathbf{X}^{\star}\boldsymbol{\beta} + \frac{1}{\sigma^2}\boldsymbol{\Sigma}_{12}\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) && (35)\\
&= \mathbf{X}^{\star}\boldsymbol{\beta} + \frac{1}{\sigma^2}\boldsymbol{\Sigma}_{12}\mathbf{U}\widetilde{\mathbf{D}}^{-1}\mathbf{U}^{T}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) && (36)\\
&= \mathbf{X}^{\star}\boldsymbol{\beta} + \frac{1}{\sigma^2}\boldsymbol{\Sigma}_{12}\mathbf{U}\widetilde{\mathbf{D}}^{-1}(\widetilde{\mathbf{Y}} - \widetilde{\mathbf{X}}\boldsymbol{\beta}) && (37)\\
&= \mathbf{X}^{\star}\boldsymbol{\beta} + \frac{1}{\sigma^2}\eta\sigma^2\boldsymbol{\Phi}^{\star}\mathbf{U}\widetilde{\mathbf{D}}^{-1}(\widetilde{\mathbf{Y}} - \widetilde{\mathbf{X}}\boldsymbol{\beta}) && (38)\\
&= \mathbf{X}^{\star}\boldsymbol{\beta} + \eta\boldsymbol{\Phi}^{\star}\mathbf{U}\widetilde{\mathbf{D}}^{-1}(\widetilde{\mathbf{Y}} - \widetilde{\mathbf{X}}\boldsymbol{\beta}) && (39)
\end{aligned}$$

where $\boldsymbol{\Phi}^{\star}$ is the $q \times (N - q)$ covariance matrix between the testing and training individuals.

5.3.8 Choice of the optimal tuning parameter

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

$$GIC_{\lambda} = -2\ell(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}^2, \widehat{\eta}) + a_n \cdot \widehat{df}_{\lambda} \tag{40}$$

where $\widehat{df}_{\lambda}$ is the number of non-zero elements in $\widehat{\boldsymbol{\beta}}_{\lambda}$ [**?**] plus two (representing the variance parameters η and σ^2). Several authors have used this criterion for variable selection in mixed models with $a_n = \log N_T$ [**? ?**], which corresponds to the BIC. We instead choose the high-

dimensional BIC [**?**] 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.

Availability of data and material

1. The GAW20 data is freely available upon request from https://www.gaworkshop.org/data-sets.
2. Mouse cross data is available from ES upon request.
3. The entire simulation study is reproducible. Source code available at https://github.com/sahirbhatnagar/ggmix/tree/pgen/simulation. This includes scripts for `ggmix`, `lasso` and `twostep` methods.
4. The R package `ggmix` is freely available from GitHub at https://github.com/greenwoodlab/ggmix.
5. A website describing how to use the package is available at https://sahirbhatnagar.com/ggmix/.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

SRB, KO, YY and CMTG conceived the idea. SRB developped the algorithms, software and simulation study. TL completed the real data analysis. ES and JCLO provided data and interpretations. SRB, TL and CMTG wrote a draft of the manuscript then all authors edited, read and approved the final manuscript.

Acknowledgements

SRB was supported by the Ludmer Centre for Neuroinformatics and Mental Health and the Canadian Institutes for Health Research PJT 148620. This research was enabled in part by support provided by Calcul Québec (www.calculquebec.ca) and Compute Canada (www.computecanada.ca). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Supporting Information

Contains the following sections:

A **Block Coordinate Descent Algorithm** - a detailed description of the algorithm used to fit our `ggmix` model

B **Additional Simulation Results** - complete simulation results

C **`ggmix` Package Showcase** - a vignette describing how to use our `ggmix` R package

A Block Coordinate Descent Algorithm

We use a general purpose block coordinate descent algorithm (CGD) [**?**] to solve (17). 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 [**?**]. For continuously differentiable $f(\cdot)$ and convex and block-separable $P(\cdot)$ (i.e. $P(\boldsymbol{\beta}) = \sum_i P_i(\beta_i)$), [**?**] 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. [**?**], [**?**]) and [**?**] for mixed models with an ℓ_1 penalty. Following Tseng and Yun [**?**], the CGD algorithm is given by Algorithm 2.

The Armijo rule is defined as follows [**?**]:

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

$$Q_\lambda(\Theta_j^{(k)} + \alpha^{(k)}d^{(k)}) \leq Q_\lambda(\Theta_j^{(k)}) + \alpha^{(k)}\varrho\Delta^{(k)} \tag{45}$$

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

$$\Delta^{(k)} := \nabla f(\Theta_j^{(k)})d^{(k)} + \gamma(d^{(k)})^2 H_{jj}^{(k)} + \lambda P(\Theta_j^{(k)} + d^{(k)}) - \lambda P(\Theta^{(k)}) \tag{46}$$

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

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

Algorithm 2: Coordinate Gradient Descent Algorithm to solve (17)

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

repeat

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

$$H^{(k)} = \text{diag}\left[\min\left\{\max\left\{\left[\nabla^2 f(\boldsymbol{\Theta}^{(k)})\right]_{jj}, c_{min}\right\} c_{max}\right\}\right]_{j=1,\ldots,p} \tag{41}$$

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**

$$d_{H^{(k)}}(\Theta_j^{(k)}) \leftarrow \arg\min_d \left\{\nabla f(\Theta_j^{(k)})d + \frac{1}{2}d^2 H_{jj}^{(k)} + \lambda P(\Theta_j^{(k)} + d)\right\} \tag{42}$$

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;

$$\widehat{\eta}^{(k+1)} \leftarrow \arg\min_\eta \frac{1}{2}\sum_{i=1}^{N_T} \log(1 + \eta(\Lambda_i - 1)) + \frac{1}{2\sigma^{2\,(k)}} \sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^p \widetilde{X}_{ij+1}\beta_j^{(k+1)}\right)^2}{1 + \eta(\Lambda_i - 1)} \tag{43}$$

Update;

$$\widehat{\sigma^2}^{\,(k+1)} \leftarrow \frac{1}{N_T}\sum_{i=1}^{N_T} \frac{\left(\widetilde{Y}_i - \sum_{j=0}^p \widetilde{X}_{ij+1}\beta_j^{(k+1)}\right)^2}{1 + \eta^{(k+1)}(\Lambda_i - 1)} \tag{44}$$

$k \leftarrow k + 1$

until *convergence criterion is satisfied*;

A.1 ℓ_1 penalty

The objective function is given by

$$Q_\lambda(\boldsymbol{\Theta}) = f(\boldsymbol{\Theta}) + \lambda|\boldsymbol{\beta}| \tag{47}$$

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

$$d_H(\Theta_j) = \arg\min_d G(d) \tag{48}$$

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 $0 \in \partial G(d)$:

$$\partial G(d) = \nabla f(\Theta_j) + dH_{jj} + \lambda u \tag{49}$$

where

$$u = \begin{cases} 1 & \text{if} \quad d > -\Theta_j \\ -1 & \text{if} \quad d < -\Theta_j \\ [-1, 1] & \text{if} \quad d = \Theta_j \end{cases} \tag{50}$$

We consider each of the three cases in (49) 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 [**?**].

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 (48). Therefore the descent direction for $\Theta_j^{(k)} \in \{\beta_1, \ldots, \beta_p\}$ for the ℓ_1 penalty is given by

$$d = \text{mid}\left\{\frac{-(\nabla f(\beta_j) - \lambda)}{H_{jj}}, -\beta_j, \frac{-(\nabla f(\beta_j) + \lambda)}{H_{jj}}\right\} \tag{51}$$

A.1.2 Solution for the β parameter

If the Hessian $\nabla^2 f(\boldsymbol{\Theta}^{(k)}) > 0$ then $H^{(k)}$ defined in (41) 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

$$\beta_j^{(k+1)} \leftarrow \beta_j^{(k)} + d^{(k)}, \qquad j = 1, \ldots, p \tag{52}$$

Substituting the descent direction given by (51) into (52) we get

$$\beta_j^{(k+1)} = \text{mid}\left\{\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) - \lambda)}{H_{jj}}, 0, \beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) + \lambda)}{H_{jj}}\right\} \tag{53}$$

We can further simplify this expression. Let

$$w_i \coloneqq \frac{1}{\sigma^2\left(1 + \eta(\Lambda_i - 1)\right)} \tag{54}$$

.

Re-write the part depending on $\boldsymbol{\beta}$ of the negative log-likelihood in (15) as

$$g(\boldsymbol{\beta}^{(k)}) = \frac{1}{2}\sum_{i=1}^{N_T} w_i \left(\widetilde{Y}_i - \sum_{\ell \neq j} \widetilde{X}_{i\ell}\beta_\ell^{(k)} - \widetilde{X}_{ij}\beta_j^{(k)}\right)^2 \tag{55}$$

The gradient and Hessian are given by

$$\nabla f(\beta_j^{(k)}) \coloneqq \frac{\partial}{\partial \beta_j^{(k)}} g(\boldsymbol{\beta}^{(k)}) = -\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell \neq j} \widetilde{X}_{i\ell}\beta_\ell^{(k)} - \widetilde{X}_{ij}\beta_j^{(k)}\right) \tag{56}$$

$$H_{jj} \coloneqq \frac{\partial^2}{\partial {\beta_j^{(k)}}^2} g(\boldsymbol{\beta}^{(k)}) = \sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2 \tag{57}$$

Substituting (56) and (57) into $\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) - \lambda)}{H_{jj}}$

$$\begin{aligned}
&\beta_j^{(k)} + \frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell \neq j} \widetilde{X}_{i\ell}\beta_\ell^{(k)} - \widetilde{X}_{ij}\beta_j^{(k)}\right) + \lambda}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2} \\
&= \beta_j^{(k)} + \frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell \neq j} \widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) + \lambda}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2} - \frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2 \beta_j^{(k)}}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2} \\
&= \frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell \neq j} \widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) + \lambda}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2}
\end{aligned} \tag{58}$$

Similarly, substituting (56) and (57) in $\beta_j^{(k)} + \frac{-(\nabla f(\beta_j^{(k)}) + \lambda)}{H_{jj}}$ we get

$$\frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell \neq j} \widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) - \lambda}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2} \tag{59}$$

Finally, substituting (58) and (59) into (53) we get

$$\begin{aligned}\beta_j^{(k+1)} &= \text{mid}\left\{\frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) - \lambda}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2}, 0, \frac{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right) + \lambda}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2}\right\} \\ &= \frac{\mathcal{S}_\lambda\left(\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}\left(\widetilde{Y}_i - \sum_{\ell\neq j}\widetilde{X}_{i\ell}\beta_\ell^{(k)}\right)\right)}{\sum_{i=1}^{N_T} w_i \widetilde{X}_{ij}^2} \end{aligned} \tag{60}$$

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

B.1 Null Model ($c = 0$)

B.2 1% of SNPs are Causal ($c = 0.01$)

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` [**?**].

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)
```

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

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
```

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

```
# 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)
class(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, ]
```

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,])
```

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 (HDBIC):

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


# 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"]]
```

We can also plot the BIC results:

```
plot(bicfit, ylab = "BIC")
bicfit[["lambda.min"]]
```

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]
```

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

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

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,])
```

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]
```

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)
```