Multivariate Genomewide Association Analysis with IHT

Benjamin B. Chu[1], Seyoon Ko[2], Jin J. Zhou[3], Hua Zhou[2], Janet S. Sinsheimer[1,2,4], Kenneth Lange[1,4,5]*

[1]Department of Computational Medicine, David Geffen School of Medicine at UCLA, Los Angeles, USA
[2]Department of Biostatistics, Fielding School of Public Health at UCLA, Los Angeles, USA
[3]Division of Epidemiology and Biostatistics, University of Arizona, USA
[4]Department of Human Genetics, David Geffen School of Medicine at UCLA, Los Angeles, USA
[5]Department of Statistics at UCLA, Los Angeles, USA

keywords: sparsity; Gaussian; iterative hard thresholding; Julia

1 Abstract

In genome-wide association studies (GWAS), analyzing multiple correlated traits is potentially superior to conducting multiple univariate analyses. Standard methods for multivariate GWAS operate marker-by-marker and are computationally intensive. We present a penalized regression algorithm for multivariate GWAS based on iterative hard thresholding (IHT) and implement it in a convenient Julia package `MendelIHT.jl` (https://github.com/OpenMendel/MendelIHT.jl). In simulation studies with up to 100 traits, IHT exhibits similar true positive rates, smaller false positive rates, and faster execution times than `GEMMA`'s linear mixed models and `mv-PLINK`'s canonical correlation analysis. As evidence of its scalability, our IHT code completed a trivariate trait analysis on the UK Biobank with 200,000 samples and 500,000 SNPs in 20 hours on a single machine.

*Corresponding author. Email: klange@ucla.edu

2 Introduction

Current statistical methods for genome-wide association studies (GWAS) can be broadly categorized as single variant or multi-variant in their genomic predictors. Multi-variant sparse models ignore polygenic background and assume that only a small number of single-nucleotide polymorphisms (SNPs) are truly causal for a given phenotype. Model fitting is typically accomplished via regression with penalties such as the least absolute shrinkage and selection operator (LASSO) [2, 31, 38, 44, 42], minimax concave penalty (MCP) [6, 41], iterative hard thresholding (IHT) [7, 16], or Bayesian analogues [13]. Linear mixed models (LMM) dominate the single variant space. LMMs control for polygenic background while focusing on the effect of a single SNP. LMMs are implemented in the contemporary programs GEMMA [47], BOLT [22], GCTA [15, 39], and SAIGE [46]. The virtues of the various methods vary depending on the genetic architecture of a trait, and no method is judged uniformly superior [12].

Although there is no consensus on the best modeling framework for single-trait GWAS, there is considerable support for analyzing multiple correlated phenotypes jointly rather than separately [12, 28, 37]. When practical, joint analysis (a) incorporates extra information on cross-trait covariances, (b) distinguishes between pleiotropic and independent SNPs, (c) reduces the burden of multiple testing, and (d) ultimately increases statistical power. Surprisingly, simulation studies suggest these advantages hold even if only one of multiple traits is associated with a SNP or if the correlation among traits is weak [12]. These advantages motivate the current paper and our search for an efficient method for analyzing multivariate traits.

Existing methods for multivariate-trait GWAS build on the polygenic model or treat SNPs one by one. For instance, GEMMA [48] implements multivariate linear mixed models (mvLMM), mv-PLINK [10] implements canonical correlation analysis, and MultiPhen [27] and Scopa [24] invert regression so that the genotypes at a single SNP become the trait and observed phenotypes become predictors.

To our knowledge, there are no sparse regression methods for multivariate-trait GWAS. In this paper, we extend IHT [5] to the multivariate setting and implement it in the Julia [4] package `MendelIHT.jl`, part of the larger `OpenMendel` statistical genetics ecosystem [45]. We have previously demonstrated the virtues of IHT compared to LASSO regression and single-SNP analysis for univariate GWAS [7, 16]. Since IHT assumes sparsity and focuses on mean effects, it is ill suited to capture polygenic background as represented in classic variance components models. In the sequel we first describe our generalization of IHT. Then we study the performance of IHT on simulated traits given real genotypes. These simulations explore the impact of varying the sparsity level k and the number of traits r. To demonstrate the potential of IHT on real large-scale genomic data, we also apply it to three hypertension related traits from the UK Biobank. These studies showcase IHT's speed, low false positive rate, and scalability to large numbers of traits. Our concluding discussion summarizes our main findings, limitations of IHT, and question worthy of future research.

3 Materials and Methods

3.1 Model Development

Consider multivariate linear regression with r traits under a Gaussian model. Up to a constant, the loglikelihood $\mathcal{L}(\mathbf{B},\mathbf{\Gamma})$ for n independent samples is

$$\mathcal{L}(\mathbf{B},\mathbf{\Gamma}) \quad = \quad \frac{n}{2}\log(\det\mathbf{\Gamma}) - \frac{1}{2}\operatorname{tr}\left[\mathbf{\Gamma}(\mathbf{Y}-\mathbf{B}\mathbf{X})(\mathbf{Y}-\mathbf{B}\mathbf{X})^T\right]. \tag{3.1}$$

The loglikelihood $\mathcal{L}(\mathbf{B},\mathbf{\Gamma})$ is a function of the $r\times p$ regression coefficients matrix $\mathbf{B}$ and the $r\times r$ unstructured precision (inverse covariance) matrix $\mathbf{\Gamma}$. Furthermore, $\mathbf{Y}$ is the $r\times n$ matrix of traits (responses), and $\mathbf{X}$ is the $p\times n$ design matrix (genotypes plus non-genetic predictors). All predictors are treated as fixed effects.

IHT maximizes $\mathcal{L}(\mathbf{B},\mathbf{\Gamma})$ subject to the constraints that k or fewer entries of $\mathbf{B}$ are non-zero and that $\mathbf{\Gamma}$ is symmetric and positive definite. Optimizing $\mathcal{L}(\mathbf{B},\mathbf{\Gamma})$ with respect to $\mathbf{B}$ for $\mathbf{\Gamma}$ fixed relies on three core ideas. The first is gradient ascent. Elementary calculus tells us that the gradient $\nabla_{\mathbf{B}}\mathcal{L}(\mathbf{B},\mathbf{\Gamma})$ is the direction of steepest ascent of $\mathcal{L}(\mathbf{B},\mathbf{\Gamma})$ at $\mathbf{B}$ for $\mathbf{\Gamma}$ fixed. IHT updates $\mathbf{B}$ in the steepest ascent direction by the formula $\mathbf{B}_{m+1} = \mathbf{B}_m + t_m\nabla_{\mathbf{B}}\mathcal{L}(\mathbf{B}_m,\mathbf{\Gamma}_m)$, where $t_m > 0$ is an optimally chosen step length and $(\mathbf{B}_m,\mathbf{\Gamma}_m)$ is the current value of the pair $(\mathbf{B},\mathbf{\Gamma})$. The gradient is derived in the appendix as the matrix

$$\nabla_{\mathbf{B}}\mathcal{L}(\mathbf{B},\mathbf{\Gamma}) \quad = \quad \mathbf{\Gamma}(\mathbf{Y}-\mathbf{B}\mathbf{X})\mathbf{X}^T. \tag{3.2}$$

The second core idea dictates how to choose the step length t_m. This is accomplished by expanding the function $t \mapsto \mathcal{L}[\mathbf{B}_m + t_m\nabla_{\mathbf{B}}\mathcal{L}(\mathbf{B}_m,\mathbf{\Gamma}_m)]$ in a second-order Taylor series around $(\mathbf{B}_m,\mathbf{\Gamma}_m)$. Our appendix shows that the optimal t_m for this quadratic approximant is

$$t_m = \frac{\|\mathbf{C}_m\|_F^2}{\operatorname{tr}(\mathbf{X}^T\mathbf{C}_m^T\mathbf{\Gamma}_m\mathbf{C}_m\mathbf{X})} \tag{3.3}$$

given the abbreviation $\mathbf{C}_m = \nabla_{\mathbf{B}}\mathcal{L}(\mathbf{B}_m,\mathbf{\Gamma}_m)$. The third core idea of IHT involves projecting the steepest ascent update $\mathbf{B}_{m+1} = \mathbf{B}_m + t_m\nabla_{\mathbf{B}}\mathcal{L}(\mathbf{B}_m,\mathbf{\Gamma}_m)$ to the sparsity set $S_k = \{\mathbf{B} : \|\mathbf{B}\|_0 \le k\}$. The projection operator $P_{S_k}(\mathbf{B})$ sets to zero all but the largest k entries in magnitude of $\mathbf{B}$. This goal can be achieved efficiently by a partial sort on the vectorized version $\operatorname{vec}(\mathbf{B}_{m+1})$ of $\mathbf{B}_{m+1}$. For all predictors to be treated symmetrically in projection, they should be standardized to have mean 0 and variance 1. Likewise, in cross-validation of k with mean square error prediction, it is a good idea to standardize all traits.

To update the precision matrix $\mathbf{\Gamma}$ for $\mathbf{B}$ fixed, we take advantage of the gradient

$$\nabla_{\mathbf{\Gamma}}\mathcal{L}(\mathbf{B},\mathbf{\Gamma}) \quad = \quad \frac{n}{2}\mathbf{\Gamma}^{-1} - \frac{1}{2}(\mathbf{Y}-\mathbf{B}\mathbf{X})(\mathbf{Y}-\mathbf{B}\mathbf{X})^T \tag{3.4}$$

spelt out in the appendix. At a stationary point where $\nabla_{\mathbf{\Gamma}}\mathcal{L}(\mathbf{B},\mathbf{\Gamma}) = \mathbf{0}_{r\times r}$, the optimal $\mathbf{\Gamma}$ is

$$\mathbf{\Gamma}_{m+1} = \left[\frac{1}{n}(\mathbf{Y}-\mathbf{B}_m\mathbf{X})(\mathbf{Y}-\mathbf{B}_m\mathbf{X})^T\right]^{-1}. \tag{3.5}$$

Equation (3.5) preserves the symmetry and positive semidefiniteness of $\mathbf{\Gamma}_m$. The required matrix inversion is straightforward unless the number of traits r is exceptionally large. Our experiments suggest solving for $\mathbf{\Gamma}_{m+1}$ exactly is superior to running full IHT jointly on both $\mathbf{B}$ and $\mathbf{\Gamma}$. Figure 1 displays our block ascent algorithm.

Algorithm 1: Block Ascent Multivariate Iterative Hard-Thresholding

Input : Genotypes $\mathbf{X}_{p\times n}$, traits $\mathbf{Y}_{r\times n}$, sparsity parameter k.
Initialize: $\mathbf{B}$ to univariate regression values, $\mathbf{\Gamma}_{r\times r}$ to identity matrix.
while not converged do
  Calculate gradient: $\mathbf{C} = \mathbf{\Gamma}(\mathbf{Y} - \mathbf{B}\mathbf{X})\mathbf{X}^T$
  Calculate step size: $t = \frac{\|\mathbf{C}\|_F^2}{\mathrm{tr}(\mathbf{X}^T\mathbf{C}^T\mathbf{\Gamma}\mathbf{C}\mathbf{X})}$
  Project to Sparsity: $\mathbf{B}_{new} = P_{S_k}(\mathbf{B} + t\mathbf{C})$
  Update: $\mathbf{\Gamma}_{new} = \left[\frac{1}{n}(\mathbf{Y} - \mathbf{B}_{new}\mathbf{X})(\mathbf{Y} - \mathbf{B}_{new}\mathbf{X})^T\right]^{-1}$
  while $L(\mathbf{B}, \mathbf{\Gamma}) > L(\mathbf{B}_{new}, \mathbf{\Gamma}_{new})$ **and backtrack steps** ≤ 3 **do**
    $t = t/2$
    $\mathbf{B}_{new} = P_{S_k}(\mathbf{B} + t\mathbf{C})$
    $\mathbf{\Gamma}_{new} = \left[\frac{1}{n}(\mathbf{Y} - \mathbf{B}_{new}\mathbf{X})(\mathbf{Y} - \mathbf{B}_{new}\mathbf{X})^T\right]^{-1}$
  end
end
Output: $\mathbf{B}$ with k non-zero entries and a symmetric and positive definite $\mathbf{\Gamma}$

3.2 Linear Algebra with Compressed Genotype Matrices

We previously described how to manipulate PLINK files using the `OpenMendel` module `SnpArrays.jl` [45], which supports linear algebra on compressed genotype matrices [7]. We now outline several enhancements to our compressed linear algebra routines.

Compact genotype storage and fast reading. A binary PLINK genotype [30] stores each SNP genotype in two bits. Thus, an $n \times p$ genotype matrix requires $2np$ bits of memory. For bit-level storage `Julia` [4] supports the 8-bit unsigned integer type (`UInt8`) that can represent four sample genotypes simultaneously in a single 8-bit integer. Extracting sample genotypes can be achieved via bitshift and bitwise `and` operations. Genotypes are stored in little endian fashion, with 0, 1, 2, and missing genotypes mapped to the bit patterns 00, 01, 11, and 10, respectively. For instance, if a locus has four sample genotypes 1, 0, 2, and missing, then the corresponding `UInt8` integer is 10110001 in binary representation. Finally, because the genotype matrix is memory-mapped, opening a genotype file and accessing data is fast even for very large files.

SIMD-vectorized and tiled linear algebra. In IHT the most computationally intensive operations are the matrix-vector and matrix-matrix multiplications required in computing gradients. To speed up these operations, we employ SIMD (single instruction, multiple data) vectorization and tiling. On machines with SIMD support such as AVX (Advanced Vector Extensions), our linear algebra routine on compressed genotypes is

usually twice as fast as BLAS 2 (Basic Linear Algebra Subroutines) [21] calls with an uncompressed numeric matrix and comparable in speed to BLAS 3 calls if $\mathbf{B}$ is tall and thin.

Computation of the matrix product $\mathbf{C} = \mathbf{AB}$ requires special care when $\mathbf{A}$ is the binary PLINK-formatted genotype matrix and $\mathbf{B}$ and $\mathbf{C}$ are numeric matrices. The idea is to partition these three matrices into small blocks and exploit the representation $\mathbf{C}_{ij} = \sum_k \mathbf{A}_{ik}\mathbf{B}_{kj}$ by computing each tiled product $\mathbf{A}_{ik}\mathbf{B}_{kj}$ in parallel. Because entries of a small matrix block are closer together in memory, this strategy improves cache efficiency. The triple `for` loops needed for computing products $\mathbf{A}_{ik}\mathbf{B}_{kj}$ are accelerated by invoking Julia's `LoopVectorization.jl` package, which performs automatic vectorization on machines with SIMD support. Furthermore, this routine can be parallelized because individual blocks can be multiplied and added independently. Because multi-threading in Julia is composable, these parallel operations can be safely nested inside other multi-threading Julia functions such as IHT's cross-validation routine.

3.3 Simulated Data Experiments

Our simulation studies are based on the chromosome 1 genotype data of the Northern Finland Birth Cohort (NFBC) [32]. The original NFBC1966 data contain 5402 samples and 364,590 SNPs; 26,906 of the SNPs reside on chromosome 1. After filtering for samples with at least 98% genotype success rate and SNPs with missing data less than 2%, we ended with 5340 samples and 24,523 SNPs on chromosome 1. For r traits, phenotypes are simulated according to the matrix normal distribution [8, 11, 40] as

$$\mathbf{Y}_{r\times n} \quad \sim \quad \text{MatrixNormal}(\mathbf{B}_{r\times p}\mathbf{X}_{p\times n},\ \ \mathbf{\Sigma}_{r\times r},\ \ \sigma_g^2\Phi_{n\times n} + \sigma_e^2\mathbf{I}_{n\times n})$$

using the `OpenMendel` module `TraitSimulation.jl` [14]. Here $\mathbf{X}$ is the chromosome 1 NFBC $p \times n$ genotype matrix with n samples aligned along its columns. The matrix $\mathbf{B}$ contains the true regression coefficients b_{ij} uniformly drawn from $\{0.05, 0.1, ..., 0.5\}$ and randomly set to 0 so that k_{true} entries b_{ij} survive. In standard mathematical notation, $\|\mathbf{B}\|_0 = k_{true}$. Note the effect-size set $\{0.05, 0.1, ..., 0.5\}$ is comparable to previous studies [7]. To capture pleiotropic effects, k_{plei} SNPs are randomly chosen to impact 2 traits. The remaining k_{indep} causal SNPs affect only one phenotype. Thus, $k_{true} = 2k_{plei} + k_{indep}$. Note it is possible for 2 traits to share 0 pleiotropic SNPs. The row (trait) covariance matrix $\mathbf{\Sigma}$ is simulated so that its maximum condition number does not exceed 10. The column (sample) covariance matrix equals $\sigma_g^2\Phi + \sigma_e^2\mathbf{I}$, where Φ is the centered genetic relationship matrix (GRM) estimated by GEMMA [48]. We let $\sigma_g^2 = 0.1$ and $\sigma_e^2 = 0.9$, so polygenic heritability is 10%. Different combinations of r, k_{true}, k_{indep}, and k_{plei} are summarized in Table 1. Each combination is replicated 100 times. It is worth emphasizing that this generative model should favor LMM analysis.

3.4 Method Comparisons

In our simulation experiments, we compared `MendelIHT.jl` to `mv-PLINK` [10] and `GEMMA` [48]. The linear mixed model software `GEMMA` enjoys wide usage in genetic epidemiology. The software `mv-PLINK` is chosen for its speed. A recent review [12] rates `mv-PLINK` as the second fastest of the competing programs. The fastest method, MV-BIMBAM [34], is an older method published by the authors of `GEMMA`, so it is not featured in this study.

In simulated data experiments, all programs were run within 16 cores of an Intel Xeon Gold 6140 2.30GHz CPU with access to 32 GB of RAM. All experiments relied on version 1.4.2 of `MendelIHT` and Julia v1.5.4. IHT's sparsity level k is tuned by cross-validation. The number of cross-validation paths is an important determinant of both computation time and accuracy. Thus, for simulated data, we employed an initial grid search involving 5-fold cross validation over the sparsity levels $k \in \{5, 10, ..., 50\}$. This was followed by 5-fold cross-validation for $k \in \{k_{best} - 4, ..., k_{best} + 4\}$. This strategy first searches the space of potential values broadly, then zooms in on the most promising candidate sparsity level. `GEMMA` and `mv-PLINK` were run under their default settings. For both programs, we declared SNPs significant whose p-values were lower than $0.05/24523$. For `GEMMA`, we used the Wald test statistic.

3.5 Quality Control for UK Biobank

We ran `MendelIHT.jl` on a subset of the data from the second release of the UK Biobank [35], which contains $\sim 500,000$ samples and $\sim 800,000$ SNPs. Specifically, we jointly analyzed three potentially correlated traits: average systolic blood pressure (SBP), average diastolic blood pressure (DBP), and body mass index (BMI). These phenotypes were first log-transformed to minimize the impact of outliers. Then each phenotype was standardized to mean 0 and variance 1, so that the three traits were treated similarly in mean-squared error (MSE) cross-validation. Following [7], we first filtered samples exhibiting sex discordance, high heterozygosity, or high SNP missingness. We then excluded samples of non-European ancestry and first and second-degree relatives based on empirical kinship coefficients. We also excluded samples who were on hypertension medicine at baseline. Finally, we excluded samples with $< 98\%$ genotyping success rate and SNPs with $< 99\%$ genotyping success rate and imputed the remaining missing genotypes by the corresponding sample-mean genotypes. Note that imputation occurs in IHT on-the-fly. The final dataset contains $185,656$ samples and $470,228$ SNPs. Given these reduced data and ignoring the Biobank's precomputed prinicipal components, we computed afresh the top 10 principal components via `FlashPCA2` [1] and included these as predictors to adjust for hidden ancestry. We also designated sex, age, and age^2 as non-genetic predictors.

4 Results

4.1 Simulation Experiments

Table 1 summarizes the various experiments conducted on the simulated data. For IHT cross-validation times are included. Multivariate IHT is the fastest method across the board, and is the only method that can analyze more than 50 traits. Multivariate IHT's runtime increases roughly linearly with the number of traits. All methods perform similarly in recovering the pleiotropic and independent SNPs. Univariate IHT exhibit slightly worse true positive rate compared to multivariate methods. Given the identically distributed effect sizes in our simulations, all methods are better at finding pleiotropic SNPs than independent SNPs.

Notably, the false positive rates for both univariate and multivariate IHT are much lower than competing methods, often by more than an order of magnitude. Presumably, many of the false positives from mvLMM and CCA represent SNPs in significant LD with the causal SNP. IHT is better at distilling the true signal within these LD blocks because it considers the effect of all SNPs jointly. `GEMMA`'s mvLMM is better at controlling false positives than `mv-PLINK`'s CCA, but model fitting for mvLMM is slower, especially for large numbers of traits. In summary, IHT offers better model selection than these competitors with better computational speed.

4.2 UK Biobank Analysis

The UK Biobank analysis completed in 20 hours and 8 minutes on 36 cores of an Intel Xeon Gold 6140 2.30GHz CPU with access to 180 GB of RAM. As described in methods section, the featured phenotypes are body mass index (BMI), average systolic blood pressure (SBP), and average diastolic blood pressure (DBP). A first pass with 3-fold cross-validation across model sizes $k \in \{100, 200, ..., 1000\}$ showed that $k = 200$ minimizes the MSE (mean squared error). A second pass with 3 fold cross-validation across model sizes $k \in \{110, 120, ..., 290\}$ showed that $k = 190$ minimizes the MSE. A third 3-fold cross-validation pass across $k \in \{191, 192, ..., 209\}$ identified $k = 197$ as the best sparsity level. Given $k = 197$, we ran multivariate IHT on the full data to estimate effect sizes, correlation among traits, and proportion of phenotypic variance explained by the genotypes.

IHT selected 13 pleiotropic SNPs and 171 independent SNPs. Selected SNPs and non-genetic predictors appear in the supplement as Tables 2-6. To compare against previous studies, we used the `R` package `gwasrapidd` [25] to search the NHGRI-EBI GWAS catalog [23] for previously associated SNPs within 1 Mb of each IHT discovered SNP. After matching, all 13 pleiotropic SNPs and 158 independent SNPs are either previously associated or are within 1Mb of a previously associated SNP. We discovered 3 new associations with SBP and 10 new associations associated with DBP. Seven SNPs, rs2307111, rs6902725, rs11977526, rs2071518, rs11222084, rs365990, and rs77870048, are associated with two traits in opposite directions.

One can estimate the genotypic variance explained by the sparse model as $\text{Var}(\hat{\boldsymbol{\beta}}_i \mathbf{X}) / \text{Var}(\mathbf{y}_i)$ for each trait

	Time (sec)	Plei TP	Indep TP	FP
Set 1: (2 traits, $k_{true} = 10$, $k_{indep} = 4$, $k_{plei} = 3$)				
mIHT	164.6 ± 69.3	0.92 ± 0.16	0.76 ± 0.2	3.7 ± 6.4
uIHT	114.9 ± 48.6	0.93 ± 0.16	0.72 ± 0.2	1.4 ± 3.7
CCA	152.6 ± 57.3	0.96 ± 0.14	0.78 ± 0.2	77.8 ± 40.6
mvLMM	307.7 ± 121.4	0.95 ± 0.15	0.76 ± 0.2	42.8 ± 18.5
Set 2: (3 traits, $k_{true} = 20$, $k_{indep} = 10$, $k_{plei} = 5$)				
mIHT	214.4 ± 100.1	0.91 ± 0.12	0.75 ± 0.14	5.7 ± 6.0
uIHT	169.6 ± 81.9	0.86 ± 0.16	0.72 ± 0.16	2.4 ± 2.5
CCA	226.8 ± 101.9	0.95 ± 0.09	0.79 ± 0.15	125.3 ± 55.3
mvLMM	449.9 ± 221.7	0.93 ± 0.1	0.75 ± 0.16	66.1 ± 22.8
Set 3: (5 traits, $k_{true} = 30$, $k_{indep} = 16$, $k_{plei} = 7$)				
mIHT	227.9 ± 41.1	0.93 ± 0.09	0.73 ± 0.12	5.9 ± 4.6
uIHT	213.8 ± 45.7	0.9 ± 0.11	0.69 ± 0.12	3.2 ± 3.8
CCA	371.5 ± 34.0	0.96 ± 0.07	0.75 ± 0.11	173.2 ± 54.9
mvLMM	1135.3 ± 125.5	0.94 ± 0.09	0.71 ± 0.11	93.6 ± 22.7
Set 4: (10 traits, $k_{true} = 10$, $k_{indep} = 4$, $k_{plei} = 3$)				
mIHT	278.8 ± 53.0	0.97 ± 0.09	0.74 ± 0.2	2.2 ± 2.0
uIHT	245.8 ± 34.8	0.96 ± 0.11	0.7 ± 0.23	3.1 ± 6.4
CCA	985.1 ± 97.5	0.99 ± 0.06	0.78 ± 0.2	64.6 ± 30.5
mvLMM	8067.4 ± 3900.8	0.99 ± 0.06	0.74 ± 0.18	41.8 ± 16.4
Set 5: (50 traits, $k_{true} = 20$, $k_{indep} = 10$, $k_{plei} = 5$)				
mIHT	1892.2 ± 419.0	0.93 ± 0.12	0.75 ± 0.14	2.9 ± 2.5
uIHT	1336.9 ± 310.2	0.92 ± 0.11	0.72 ± 0.12	7.6 ± 5.9
CCA	26589.1 ± 907.7(*)	NA	NA	NA
mvLMM	NA	NA	NA	NA
Set 6: (100 traits, $k_{true} = 30$, $k_{indep} = 16$, $k_{plei} = 7$)				
mIHT	3699.3 ± 410.4	0.91 ± 0.11	0.71 ± 0.11	2.8 ± 2.1
uIHT	2353.8 ± 212.3	0.92 ± 0.11	0.7 ± 0.10	10.7 ± 4.3
CCA	NA	NA	NA	NA
mvLMM	NA	NA	NA	NA

Table 1: Comparison of multivariate IHT (mIHT) and multiple univariate IHT (uIHT) implemented in MendelIHT, canonical correlation analysis (CAA) implemented in mv-PLINK, and multivariate linear mixed models (mvLMM) implemented in GEMMA on chromosome 1 of the NFBC1966 data with simulated traits. Plei TP is the proportion of true positives for pleiotopic SNPs, Indep TP is proportion of true positives for independent SNPs, and FP is the total number of false positive. $\pm$ indicates standard deviations. k_{true} is the total number of non-zero entries in $\mathbf{B}$, k_{indep} is the number of independent SNPs affecting only one trait, and k_{plei} is the number of pleiotropic SNPs affecting two traits. These numbers satisfy $k_{true} = 2 \times k_{plei} + k_{indep}$. Each simulation relied on 100 replicates. NA: not available due to excessive run time. (*) Only two replicates contribute to timing.

$\mathbf{y}_i$ where $\hat{\boldsymbol{\beta}}_i \in \mathbb{R}^{1\times p}$ is the ith row of $\mathbf{B}$. `MendelIHT.jl` output the values $\sigma^2_{BMI} = 0.033$, $\sigma^2_{SBP} = 0.143$, and $\sigma^2_{DBP} = 0.048$. Note these estimates do not include contributions from the intercept or non-genetic predictors. The estimated correlations among traits are $r_{BMI,SBP} = 0.197$, $r_{BMI,DBP} = 0.286$, and $r_{SBP,DBP} = 0.738$. As expected, all traits are positively correlated, with a strong correlation between SBP and DBP and a weak correlation between BMI and both SBP and DBP.

5 Discussion

This paper presents multivariate IHT for analyzing multiple correlated phenotypes. In simulation studies, multivariate IHT exhibits similar true positive rates, significantly lower false positive rates, and better overall speed than linear mixed models and canonical correlation analysis. Computational time for multivariate IHT increases roughly linearly with the number of traits. Since IHT is a penalized regression method, the estimated effect size for each SNP is explicitly conditioned on other SNPs and non-genetic predictors. Analyzing three correlated UK Biobank traits with $\sim 200,000$ samples and $\sim 500,000$ SNPs took 20 hours on a single machine. IHT can output the correlation matrix and proportion of variance explained for component traits. `MendelIHT.jl` also automatically handles various input formats (binary PLINK, BGEN, and VCF files) by calling the relevant `OpenMendel` packages. If binary PLINK files are used, `MendelIHT.jl` avoids decompressing genotypes to full numeric matrices.

IHT's statistical and computational advantages come with limitations. For instance, it ignores hidden and explicit relatedness. IHT can exploit principal components to adjust for ancestry, but PCA alone is insufficient to account for small-scale family structure [29]. To overcome this limitation, close relatives can be excluded from a study. Although cross-validation may implicitly adjust for family relatedness, a more comprehensive analysis is required before we can recommend including related samples. Although our simulation studies suggest the contrary, there is also the possibility that strong linkage disequilibrium may confuse IHT. Finally, it is unclear how IHT will respond to wrongly imputed markers and the rare variants generated by sequencing. In spite of these qualms, the evidence presented here is very persuasive about IHT's potential for multivariate GWAS.

We will continue to explore improvements to IHT. Extension to non-Gaussian traits is hindered by the lack of flexible multivariate distributions with non-Gaussian margins. Cross-validation remains computationally intensive in tuning the sparsity level k. Although our vectorized linear algebra routine partially overcomes many of the computational barriers, we feel that further gains are possible through GPU computing [18, 17, 19, 43] and trading cross-validation for knockoff strategies in model selection [3, 33]. Given IHT's advantages, we recommend it for general use with the understanding that epidemiologists respect its limitations and complement its application with standard univariate statistical analysis.

SNP	Chrom	BMI	SBP	DBP	# prior report
rs1801131	1	0.0	0.009	0.008	41
rs17367504	1	0.0	0.02	0.019	41
rs16998073	4	0.0	-0.023	-0.023	24
rs1173727	5	0.0	0.016	0.017	19
rs2307111	5	0.013	0.0	-0.01	34
rs6902725	6	0.0	-0.009	0.01	4
rs11977526	7	0.0	0.013	-0.011	6
rs2071518	8	0.0	-0.008	0.009	6
rs11222084	11	0.0	-0.011	0.009	9
rs3184504	12	0.0	-0.011	-0.018	41
rs365990	14	0.0	0.008	-0.011	6
rs7497304	15	0.0	-0.019	-0.018	14
rs77870048	16	0.0	-0.011	0.01	20

Table 2: 13 pleiotropic SNPs selected by IHT listed with their effect sizes. An effect size of 0 means the particular predictor was not selected. The field *prior reports* records the number of SNPs previously associated with BMI, SBP, or DBP (p value $< 10^{-8}$) that are within 1Mb of the given SNP. BMI = body mass index; SBP = systolic blood pressure; DBP = diastolic blood pressure.

6 Appendix

6.1 Significant SNPs from the UKB analysis

Tables 2-6 list the SNPs discovered by our UK Biobank analysis. The reported effect sizes correspond to the predictors of the log-transformed and standardized phenotypes. To compare against previous studies, we searched the NHGRI-EBI GWAS catalog [23] using the `R` package `gwasrapidd` [25]. For each SNP discovered by IHT, we queried a 1Mb radius for other SNPs that have been previously associated with the given trait with p value $< 10^{-8}$. Each known association is defined as the most significant SNP in a locus identified to be associated with the trait. All 13 pleiotropic SNPs were previously known and 158 out of 171 independent SNPs were previously known. Among the 13 newly discovered associations, 3 were with SBP and 10 were with DBP.

covariate	BMI	SBP	DBP
intercept	0.0002	-0.0	0.0003
sex	0.1165	0.1639	0.1808
age	0.1073	-0.1395	0.6484
age^2	-0.0505	0.4581	-0.6131
PC1	-0.0287	-0.0066	-0.0074
PC2	0.0052	-0.0077	-0.0011
PC3	0.0174	0.0043	-0.0091
PC4	-0.0019	-0.0055	0.0015
PC5	0.0022	0.0137	0.0167
PC6	-0.0165	-0.0067	-0.0056
PC7	-0.003	-0.0038	-0.0078
PC8	-0.0014	-0.0015	-0.0014
PC9	0.0039	-0.0004	0.0017
PC10	0.0068	0.0005	0.0001

Table 3: Non-genetic covariates estimated by IHT listed with their effect sizes. PC is short for principal component. BMI = body mass index, SBP = systolic blood pressure, and DBP = diastolic blood pressure.

SNP	Chrom	β	# Prior reports	SNP	Chrom	β	# Prior reports
rs2815757	1	0.019	14	rs17207196	7	0.016	9
rs543874	1	-0.028	22	rs925946	11	-0.007	24
rs2820312	1	-0.014	9	rs6265	11	0.012	24
rs62106258	2	0.027	51	rs2049045	11	0.007	24
rs11127485	2	0.017	51	rs10835211	11	-0.011	24
rs13393304	2	0.008	51	rs7138803	12	-0.007	9
rs713586	2	-0.026	26	rs7132908	12	-0.014	9
rs9821675	3	0.007	20	rs4776970	15	0.01	17
rs1062633	3	0.012	20	rs16951304	15	0.008	17
rs957919	3	-0.015	17	rs2531995	16	0.016	17
rs61587156	3	0.017	9	rs72793809	16	-0.015	17
rs34811474	4	0.015	3	rs4788190	16	0.019	16
rs10938397	4	-0.022	17	rs4889490	16	0.014	20
rs13107325	4	-0.026	7	rs1421085	16	-0.05	56
rs2112347	5	0.009	22	rs17782313	18	-0.011	47
rs1422192	5	-0.015	20	rs10871777	18	-0.015	47
rs2744962	6	-0.017	33	rs17773430	18	-0.014	45
rs987237	6	-0.012	37	rs1800437	19	0.017	20
rs9473932	6	-0.009	37	rs3810291	19	0.022	12

Table 4: 38 SNPs associated with BMI independently of SBP and DBP listed with their effect sizes. The field *prior reports* records the number of GWAS Catalog associations with BMI (p value $< 10^{-8}$) that are within 1Mb of the given SNP.

SNP	Chrom	β	# Prior reports	SNP	Chrom	β	# Prior reports
rs3936009	1	0.009	7	rs12673516	7	0.01	3
rs1757915	1	-0.009	4	rs2282978	7	0.016	4
rs6684353	1	0.011	2	rs2392929	7	-0.027	5
rs12069946	1	-0.009	2	rs2978456	8	-0.01	1
rs2820441	1	0.009	1	Affx-32837790	8	-0.009	3
rs1522484	2	0.016	3	rs35758124	8	-0.01	2
rs9306894	2	0.009	4	rs10757278	9	-0.01	3
rs55654088	2	-0.01	4	rs10986626	9	-0.009	5
rs13002573	2	0.012	13	rs12258967	10	0.009	10
rs560887	2	0.011	1	rs1908339	10	0.01	4
rs10497529	2	0.011	3	rs11191064	10	0.009	6
rs1052501	3	0.02	6	rs11598702	10	-0.011	17
rs2498323	4	-0.008	3	rs4980389	11	-0.009	14
rs776590	4	0.008	2	rs573455	11	-0.014	6
rs17084051	4	-0.01	3	rs10750441	11	0.01	3
rs1229984	4	0.009	1	rs10770612	12	0.012	9
rs6842241	4	-0.008	2	rs2681492	12	0.011	21
rs4690974	4	-0.01	9	rs12882307	14	0.009	1
rs13116200	4	-0.01	2	rs4903064	14	-0.01	5
rs7715779	5	0.007	12	rs956006	15	0.009	1
rs1173771	5	0.005	12	rs34862454	15	-0.012	14
rs1982192	5	0.009	Novel	rs3803716	16	0.008	3
rs2303720	5	0.011	6	rs4888372	16	0.01	6
rs11954193	5	0.01	13	rs60675007	16	0.008	2
rs12198986	6	-0.008	1	rs9889363	17	-0.011	6
rs9349379	6	0.012	5	rs185478092	17	-0.01	Novel
rs385306	6	0.011	7	rs3744760	17	-0.01	9
rs12191865	6	-0.014	3	rs17608766	17	-0.013	3
rs1012257	6	-0.009	3	rs9909933	17	0.009	7
rs9689048	6	0.009	Novel	rs35688424	17	-0.011	8
rs2221389	6	-0.012	1	rs67882421	18	-0.012	4
rs9505897	6	-0.01	1	rs12459507	19	-0.009	6
rs57301765	7	-0.014	2	rs34328549	19	0.007	8

Table 5: 66 SNPs associated with SBP independently of BMI and DBP listed with their effect sizes. The field *prior reports* records the number of GWAS Catalog associations with SBP (p value $< 10^{-8}$) that are within 1Mb of the given SNP. A novel SNP is not within 1Mb of any GWAS Catalog associations.

SNP	Chrom	β	# Prior reports	SNP	Chrom	β	# Prior reports
rs61776719	1	-0.015	Novel	rs4754834	11	-0.009	1
rs12739904	1	-0.009	1	rs59317921	11	0.009	2
rs3766090	1	-0.01	Novel	rs7975252	12	-0.009	1
rs61822997	1	-0.009	Novel	rs7973748	12	0.011	10
rs665834	1	0.011	1	rs12581906	12	0.009	10
rs2275155	1	-0.013	6	rs17287293	12	0.01	Novel
rs11690961	2	-0.012	1	rs74340001	12	-0.01	1
rs10199082	2	-0.01	3	rs653178	12	-0.007	16
rs2692893	2	-0.012	5	rs12875271	13	0.011	3
rs17362588	2	-0.014	3	rs36033161	14	-0.009	Novel
rs12996836	2	-0.011	3	rs686861	15	-0.013	1
rs1863703	2	0.009	3	rs11853359	15	0.011	2
rs2624847	3	-0.009	3	rs1378942	15	-0.012	10
rs3617	3	0.008	5	rs4886615	15	-0.008	10
rs9850919	3	0.009	5	rs2277547	15	0.01	1
rs871606	4	-0.018	1	rs7174546	15	-0.009	2
rs1826909	4	-0.01	Novel	rs67456613	16	-0.01	2
rs1047440	5	-0.014	3	rs72790195	16	-0.008	1
rs11949055	5	0.009	5	rs11078485	17	-0.009	2
rs1233708	6	0.013	2	rs72824497	17	0.009	1
rs11154022	6	0.01	3	rs768168	19	-0.011	Novel
rs12110693	6	-0.012	3	rs997669	19	-0.01	2
rs9376740	6	-0.009	Novel	rs755690	19	-0.01	1
rs58023137	6	-0.011	1	rs2876201	20	-0.009	6
rs194524	7	-0.012	Novel	rs652661	20	0.015	6
rs13226502	7	-0.014	2	rs78309244	20	0.014	6
rs2469997	8	0.01	3	rs6046144	20	0.015	2
rs507666	9	0.012	2	rs76701589	20	0.009	3
rs3812595	9	-0.009	1	rs2831969	21	0.009	Novel
rs183348357	10	-0.009	7	rs9982601	21	-0.009	1
rs3739998	10	0.005	1	rs2298336	21	0.01	1
rs2505083	10	0.007	1	rs112005532	21	0.012	1
rs7125196	11	0.01	5	rs73167017	22	0.011	1
rs2304500	11	-0.012	1				

Table 6: 67 SNPs associated with DBP independently of BMI and SBP listed with their effect sizes. The field *prior reports* records the number of GWAS Catalog associations with DBP (p value $< 10^{-8}$) that are within 1Mb of the given SNP. A novel SNP is not within 1Mb of any GWAS Catalog associations.

6.2 Loglikelihood

Consider multivariate linear regression with r traits under a Gaussian model. Up to a constant, the loglikelihood for the response vector $\mathbf{y}_i$ of subject i can be written

$$\begin{aligned}
\mathcal{L}_i\begin{pmatrix}\mathbf{B}\\ \mathbf{\Gamma}\end{pmatrix} &= \frac{1}{2}\log(\det\mathbf{\Gamma}) - \frac{1}{2}(\mathbf{y}_i-\mathbf{B}\mathbf{x}_i)^T\mathbf{\Gamma}(\mathbf{y}_i-\mathbf{B}\mathbf{x}_i)\\
&= \frac{1}{2}\log(\det\mathbf{\Gamma}) - \frac{1}{2}\operatorname{tr}[\mathbf{\Gamma}(\mathbf{y}_i-\mathbf{B}\mathbf{x}_i)(\mathbf{y}_i-\mathbf{B}\mathbf{x}_i)^T],
\end{aligned}$$

where $\mathbf{B}$ is the $r\times p$ matrix of regression coefficients, $\mathbf{x}_i$ is the $p\times 1$ vector of predictors, and $\mathbf{\Gamma}$ is the $r\times r$ unstructured precision (inverse covariance) matrix. For n independent samples, let $\mathbf{Y}$ be the $r\times n$ matrix with ith column $\mathbf{y}_i$ and let $\mathbf{X}$ be the $p\times n$ design matrix with ith column $\mathbf{x}_i$. Then the loglikelihood for all samples is

$$\begin{aligned}
\mathcal{L}\begin{pmatrix}\mathbf{B}\\ \mathbf{\Gamma}\end{pmatrix} &= \sum_{i=1}^n\left\{\frac{1}{2}\log(\det\mathbf{\Gamma}) - \frac{1}{2}\operatorname{tr}[\mathbf{\Gamma}(\mathbf{y}_i-\mathbf{B}\mathbf{x}_i)(\mathbf{y}_i-\mathbf{B}\mathbf{x}_i)^T]\right\}\\
&= \frac{n}{2}\log(\det\mathbf{\Gamma}) - \frac{1}{2}\operatorname{tr}\left[\mathbf{\Gamma}\sum_{i=1}^n(\mathbf{y}_i-\mathbf{B}\mathbf{x}_i)(\mathbf{y}_i-\mathbf{B}\mathbf{x}_i)^T\right]\\
&= \frac{n}{2}\log(\det\mathbf{\Gamma}) - \frac{1}{2}\operatorname{tr}[(\mathbf{\Gamma}(\mathbf{Y}-\mathbf{B}\mathbf{X})(\mathbf{Y}-\mathbf{B}\mathbf{X})^T].
\end{aligned}$$

In subsequent sections we will present both full and block ascent IHT. The former updates $\mathbf{B}$ and $\mathbf{\Gamma}$ simultaneously. The latter alternates updates of $\mathbf{B}$ and $\mathbf{\Gamma}$, holding the other parameter block fixed.

6.3 First Directional Derivative

Recall that the Hadamard's semi-directional derivative [9, 20] of a function $f(\mathbf{x})$ in the direction $\mathbf{v}$ is defined as the limit

$$d_{\mathbf{v}}f(\mathbf{x}) = \lim_{\substack{h\to 0\\ \mathbf{w}\to\mathbf{v}}}\frac{f(\mathbf{x}+h\mathbf{w})-f(\mathbf{x})}{h}.$$

To calculate the directional derivative of the loglikelihood (3.1), we perturb $\mathbf{B}$ in the direction $\mathbf{U}$ and $\mathbf{\Gamma}$ in the symmetric direction $\mathbf{V}$. The sum and product rules then give

$$\begin{aligned}
&d_{\binom{\mathbf{U}}{\mathbf{V}}}\operatorname{tr}[\mathbf{\Gamma}(\mathbf{Y}-\mathbf{B}\mathbf{X})(\mathbf{Y}-\mathbf{B}\mathbf{X})^T]\\
=\ &\operatorname{tr}[\mathbf{V}(\mathbf{Y}-\mathbf{B}\mathbf{X})(\mathbf{Y}-\mathbf{B}\mathbf{X})^T] - \operatorname{tr}[\mathbf{\Gamma}(\mathbf{Y}-\mathbf{B}\mathbf{X})\mathbf{X}^T\mathbf{U}^T - \mathbf{\Gamma}\mathbf{U}\mathbf{X}(\mathbf{Y}-\mathbf{B}\mathbf{X})^T].
\end{aligned}$$

The directional derivative $d_{\mathbf{V}} \ln\det(\boldsymbol{\Gamma}) = \operatorname{tr}(\boldsymbol{\Gamma}^{-1}\mathbf{V})$ is derived in Example 3.2.6 of [20]. The trace properties $\operatorname{tr}(\mathbf{CD}) = \operatorname{tr}(\mathbf{DC})$ and $\operatorname{tr}(\mathbf{C}^T) = \operatorname{tr}(\mathbf{C})$ consequently imply

$$d_{\binom{\mathbf{U}}{\mathbf{V}}} \mathcal{L}(\mathbf{B},\boldsymbol{\Gamma}) = \frac{n}{2}\operatorname{tr}(\boldsymbol{\Gamma}^{-1}\mathbf{V}) - \frac{1}{2}\operatorname{tr}[(\mathbf{Y}-\mathbf{BX})(\mathbf{Y}-\mathbf{BX})^T\mathbf{V}] + \operatorname{tr}[\mathbf{X}(\mathbf{Y}-\mathbf{BX})^T\boldsymbol{\Gamma}\mathbf{U}]. \tag{6.1}$$

Because this last expression is linear in $(\mathbf{U},\mathbf{V})$, the loglikelihood is continuously differentiable.

6.4 Second Directional Derivative

Now we take the directional derivative of the directional derivative (6.1) in the new directions $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$. This action requires the inverse rule $d_{\tilde{\mathbf{V}}}\boldsymbol{\Gamma}^{-1} = -\boldsymbol{\Gamma}^{-1}\tilde{\mathbf{V}}\boldsymbol{\Gamma}^{-1}$ proved in Example 3.2.7 of [20]. Accordingly, we find

$$d_{\binom{\tilde{\mathbf{U}}}{\tilde{\mathbf{V}}}} \frac{n}{2}\operatorname{tr}(\boldsymbol{\Gamma}^{-1}\mathbf{V}) = -\frac{n}{2}\operatorname{tr}\left(\boldsymbol{\Gamma}^{-1}\tilde{\mathbf{V}}\boldsymbol{\Gamma}^{-1}\mathbf{V}\right).$$

We also calculate

$$d_{\binom{\tilde{\mathbf{U}}}{\tilde{\mathbf{V}}}} -\frac{1}{2}\operatorname{tr}[(\mathbf{Y}-\mathbf{BX})(\mathbf{Y}-\mathbf{BX})^T\mathbf{V}] = \frac{1}{2}\operatorname{tr}[(\mathbf{Y}-\mathbf{BX})\mathbf{X}^T\tilde{\mathbf{U}}^T\mathbf{V}] + \frac{1}{2}\operatorname{tr}[\tilde{\mathbf{U}}\mathbf{X}(\mathbf{Y}-\mathbf{BX})^T\mathbf{V}]$$

and

$$d_{\binom{\tilde{\mathbf{U}}}{\tilde{\mathbf{V}}}} \operatorname{tr}[\mathbf{X}(\mathbf{Y}-\mathbf{BX})^T\boldsymbol{\Gamma}\mathbf{U}] = \operatorname{tr}[\mathbf{X}(\mathbf{Y}-\mathbf{BX})^T\tilde{\mathbf{V}}\mathbf{U}] - \operatorname{tr}(\mathbf{X}^T\tilde{\mathbf{U}}^T\boldsymbol{\Gamma}\mathbf{U}\mathbf{X}).$$

Finally, setting the two directions equal so that $\tilde{\mathbf{V}} = \mathbf{V}$ and $\tilde{\mathbf{U}} = \mathbf{U}$ produces the quadratic form

$$\begin{aligned} Q(\mathbf{U},\mathbf{V}) &= -\frac{n}{2}\operatorname{tr}[\boldsymbol{\Gamma}^{-1}\mathbf{V}\boldsymbol{\Gamma}^{-1}\mathbf{V}] + \frac{1}{2}\operatorname{tr}[(\mathbf{Y}-\mathbf{BX})\mathbf{X}^T\mathbf{U}^T\mathbf{V}] + \frac{1}{2}\operatorname{tr}[\mathbf{U}\mathbf{X}(\mathbf{Y}-\mathbf{BX})^T\mathbf{V}] \\ &\quad + \operatorname{tr}[\mathbf{X}(\mathbf{Y}-\mathbf{BX})^T\mathbf{V}\mathbf{U}] - \operatorname{tr}(\mathbf{X}^T\mathbf{U}^T\boldsymbol{\Gamma}\mathbf{U}\mathbf{X}). \end{aligned} \tag{6.2}$$

generated by the second differential.

6.5 Extraction of the Gradient and Expected Information

To extract the gradient from a directional derivative, we recall the identity $d_{\mathbf{v}} f(\mathbf{x}) = \nabla f(\mathbf{x})^T\mathbf{v}$ for vectors $\mathbf{v}$ and $\mathbf{x}$ and the identity $\operatorname{tr}(\mathbf{A}^T\mathbf{B}) = \operatorname{vec}(\mathbf{A})^T\operatorname{vec}(\mathbf{B})$ for matrices $\mathbf{A}$ and $\mathbf{B}$ [26]. The first identity shows that the directional derivative is the inner product of the gradient with respect to the direction $\mathbf{v}$. The second displays the trace function as an inner product on dimensionally identical matrices. Thus, the matrix directional derivative is

$$d_{\mathbf{V}} f(\mathbf{X}) = \operatorname{vec}[\nabla f(\mathbf{X})]^T\operatorname{vec}(\mathbf{V}) = \operatorname{tr}[\nabla f(\mathbf{X})^T\mathbf{V}]. \tag{6.3}$$

Inspection of the directional derivative (6.1) now leads to the gradient with blocks

$$\nabla_{\mathbf{B}}\mathcal{L}\begin{pmatrix}\mathbf{B}\\ \mathbf{\Gamma}\end{pmatrix} = [\mathbf{X}(\mathbf{Y}-\mathbf{B}\mathbf{X})^T\mathbf{\Gamma}]^T = \mathbf{\Gamma}(\mathbf{Y}-\mathbf{B}\mathbf{X})\mathbf{X}^T \tag{6.4}$$

$$\nabla_{\mathbf{\Gamma}}\mathcal{L}\begin{pmatrix}\mathbf{B}\\ \mathbf{\Gamma}\end{pmatrix} = \frac{n}{2}\mathbf{\Gamma}^{-1} - \frac{1}{2}(\mathbf{Y}-\mathbf{B}\mathbf{X})(\mathbf{Y}-\mathbf{B}\mathbf{X})^T. \tag{6.5}$$

Analogously, the quadratic form (6.2) implicitly defines the Hessian $\mathbf{H}$ through the identity

$$\begin{aligned} Q(\mathbf{U},\mathbf{V}) &= \operatorname{tr}\left\{\begin{bmatrix}\operatorname{vec}(\mathbf{U})^T & \operatorname{vec}(\mathbf{V})^T\end{bmatrix}\begin{pmatrix}\mathbf{H}_{\mathbf{B}\mathbf{B}} & \mathbf{H}_{\mathbf{B}\mathbf{\Gamma}}\\ \mathbf{H}_{\mathbf{\Gamma}\mathbf{B}} & \mathbf{H}_{\mathbf{\Gamma}\mathbf{\Gamma}}\end{pmatrix}\begin{bmatrix}\operatorname{vec}(\mathbf{U})\\ \operatorname{vec}(\mathbf{V})\end{bmatrix}\right\} \\ &\equiv -\frac{n}{2}\operatorname{tr}\left(\mathbf{\Gamma}^{-1}\mathbf{V}\mathbf{\Gamma}^{-1}\mathbf{V}\right) + \frac{1}{2}\operatorname{tr}\left((\mathbf{Y}-\mathbf{B}\mathbf{X})\mathbf{X}^T\mathbf{U}^T\mathbf{V}\right) \\ &\quad + \frac{1}{2}\operatorname{tr}\left(\mathbf{U}\mathbf{X}(\mathbf{Y}-\mathbf{B}\mathbf{X})^T\mathbf{V}\right) + \operatorname{tr}\left(\mathbf{X}(\mathbf{Y}-\mathbf{B}\mathbf{X})^T\mathbf{V}\mathbf{U}\right) - \operatorname{tr}(\mathbf{X}^T\mathbf{U}^T\mathbf{\Gamma}\mathbf{U}\mathbf{X}). \end{aligned}$$

Because $\mathrm{E}(\mathbf{Y}) = \mathbf{B}\mathbf{X}$, the expected information $\mathbf{J} = \mathrm{E}(-\mathbf{H})$ has the off-diagonal blocks $\mathbf{J}_{\mathbf{B},\mathbf{\Gamma}} = \mathbf{0}_{pr\times r^2}$ and $\mathbf{J}_{\mathbf{\Gamma},\mathbf{B}} = \mathbf{0}_{r^2\times pr}$. Now the Kronecker product identity $\operatorname{vec}(\mathbf{A}\mathbf{B}\mathbf{C}) = (\mathbf{C}^T\otimes\mathbf{A})\operatorname{vec}(\mathbf{B})$ implies

$$\begin{aligned} \operatorname{tr}(\mathbf{X}^T\mathbf{U}^T\mathbf{\Gamma}\mathbf{U}\mathbf{X}) &= \operatorname{tr}(\mathbf{X}\mathbf{X}^T\mathbf{U}^T\mathbf{\Gamma}\mathbf{U}) \\ &= \operatorname{tr}[(\mathbf{\Gamma}\mathbf{U}\mathbf{X}\mathbf{X}^T)^T\mathbf{U}] \\ &= \operatorname{vec}(\mathbf{\Gamma}\mathbf{U}\mathbf{X}\mathbf{X}^T)^T\operatorname{vec}(\mathbf{U}) \\ &= \left[(\mathbf{X}\mathbf{X}^T\otimes\mathbf{\Gamma})\operatorname{vec}(\mathbf{U})\right]^T\operatorname{vec}(\mathbf{U}) \\ &= \operatorname{vec}(\mathbf{U})^T(\mathbf{X}\mathbf{X}^T\otimes\mathbf{\Gamma})\operatorname{vec}(\mathbf{U}). \end{aligned}$$

It follows that $\mathbf{J}_{\mathbf{B}\mathbf{B}} = \mathbf{X}\mathbf{X}^T\otimes\mathbf{\Gamma}$. Similarly,

$$\begin{aligned} \operatorname{tr}(\mathbf{\Gamma}^{-1}\mathbf{V}\mathbf{\Gamma}^{-1}\mathbf{V}) &= \left[(\mathbf{\Gamma}^{-1}\otimes\mathbf{\Gamma}^{-1})\operatorname{vec}(\mathbf{V})\right]^T\operatorname{vec}(\mathbf{V}) \\ &= \operatorname{vec}(\mathbf{V})^T[\mathbf{\Gamma}^{-1}\otimes\mathbf{\Gamma}^{-1}]\operatorname{vec}(\mathbf{V}), \end{aligned}$$

so that $\mathbf{J}_{\mathbf{\Gamma}\mathbf{\Gamma}} = \mathbf{\Gamma}^{-1}\otimes\mathbf{\Gamma}^{-1}$. In summary, the expected information matrix takes the block diagonal form

$$\mathbf{J} = \begin{pmatrix}(\mathbf{X}\mathbf{X}^T)\otimes\mathbf{\Gamma} & \mathbf{0}\\ \mathbf{0} & \mathbf{\Gamma}^{-1}\otimes\mathbf{\Gamma}^{-1}\end{pmatrix}. \tag{6.6}$$

In our projected steepest ascent algorithm, the expected information matrix is never explicitly formed. It is implicitly accessed in the step-size calculation through the associated quadratic form $Q(\mathbf{B},\mathbf{\Gamma})$.

6.6 Full IHT Step Size

The next iterate in full IHT is the projection of the point

$$\boldsymbol{\Delta}_{m+1} = \begin{pmatrix} \mathbf{B}_m \\ \boldsymbol{\Gamma}_m \end{pmatrix} + t_m \nabla \mathcal{L} \begin{pmatrix} \mathbf{B}_m \\ \boldsymbol{\Gamma}_m \end{pmatrix} = \begin{pmatrix} \mathbf{B}_m \\ \boldsymbol{\Gamma}_m \end{pmatrix} + t_m \begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix}, \tag{6.7}$$

where $\mathbf{C}_m = \nabla_{\mathbf{B}} \mathcal{L}$ and $\mathbf{W}_m = \nabla_{\boldsymbol{\Gamma}} \mathcal{L}$ evaluated at $(\mathbf{B}_m, \boldsymbol{\Gamma}_m)$. The loglikelihood along the ascent direction is a function of the scalar t_m and can be approximated by the second-order expansion

$$\mathcal{L}(\boldsymbol{\Delta}_{m+1}) \approx \mathcal{L} \begin{pmatrix} \mathbf{B}_m \\ \boldsymbol{\Gamma}_m \end{pmatrix} + t_m \operatorname{tr} \left[\begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix}^T \begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix} \right] - \frac{t_m^2}{2} \operatorname{tr} \left[\begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix}^T \mathbf{J} \begin{pmatrix} \mathbf{B}_m \\ \boldsymbol{\Gamma}_m \end{pmatrix} \begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix} \right].$$

The choice

$$t_m = \frac{\operatorname{tr} \left[\begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix}^T \begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix} \right]}{\operatorname{tr} \left[\begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix}^T J \begin{pmatrix} \mathbf{B}_m \\ \boldsymbol{\Gamma}_m \end{pmatrix} \begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix} \right]} = \frac{\left\| \begin{pmatrix} \mathbf{C}_m \\ \mathbf{W}_m \end{pmatrix} \right\|_F^2}{\operatorname{tr}(\mathbf{X}^T \mathbf{C}_m^T \boldsymbol{\Gamma}_m \mathbf{C}_m \mathbf{X}) + \frac{m}{2} \operatorname{tr}(\boldsymbol{\Gamma}_m^{-1} \mathbf{W}_m \boldsymbol{\Gamma}_m^{-1} \mathbf{W}_m)}$$

maximizes the approximation. If the support of the matrix $(\mathbf{B}, \boldsymbol{\Gamma})$ does not change under projection, then this IHT update is particularly apt.

6.7 IHT Projection

Recall that full IHT iterates according to

$$\begin{pmatrix} \mathbf{B}_{m+1} \\ \boldsymbol{\Gamma}_{m+1} \end{pmatrix} = P_{S_k}(\boldsymbol{\Delta}_{m+1}),$$

where $\boldsymbol{\Delta}_{m+1}$ is derived in equation (6.7). Here k is a positive integer representing the sparsity level, which is assumed known. In practice k is found through cross-validation. The projection $P_{S_k}(\boldsymbol{\Delta})$ splits into separate projections for $\mathbf{B}$ and $\boldsymbol{\Gamma}$. One can independently project each row of $\mathbf{B}$ to sparsity. Alternatively, one can require each row of $\mathbf{B}$ to have the same sparsity pattern if the same set of predictors plausibly contribute to all r traits. The $\boldsymbol{\Gamma}$ projection must preserve symmetry and positive semidefiniteness. Symmetry is automatic because the gradient of $\boldsymbol{\Gamma}$ is already symmetric. To project to positive semidefiniteness, one takes the SVD of $\boldsymbol{\Gamma}$ and project its eigenvalues $\boldsymbol{\lambda}$ to nonnegativity. One can even project $\boldsymbol{\Gamma}$ to the closest positive definite matrix with an acceptable condition number [36].

6.8 The Block Ascent IHT

In block ascent we alternate updates of $\mathbf{B}$ and $\boldsymbol{\Gamma}$. The exact update (3.4) of $\boldsymbol{\Gamma}$ is particularly convenient, and we take advantage of it. Symmetry and positive semidefiniteness are automatically preserved. Inversion can be carried out via Cholesky factorization of $\boldsymbol{\Gamma}$. This choice of $\boldsymbol{\Gamma}$ simplifies the step length

$$t_m = \frac{\|\mathbf{C}_m\|_F^2}{\operatorname{tr}(\mathbf{X}^T\mathbf{C}_m^T\boldsymbol{\Gamma}\mathbf{C}_m\mathbf{X})}.$$

Note that the denominator of the step size does not require formation of the $n \times n$ matrix $\mathbf{X}^T\mathbf{C}_m^T\boldsymbol{\Gamma}\mathbf{C}_m\mathbf{X}$. One can write $\operatorname{tr}(\mathbf{X}^T\mathbf{C}_m^T\boldsymbol{\Gamma}\mathbf{C}_m\mathbf{X}) = \operatorname{tr}(\mathbf{X}^T\mathbf{C}_m^T\mathbf{L}\mathbf{L}^T\mathbf{C}_m\mathbf{X}) = \|\mathbf{L}^T\mathbf{C}_m\mathbf{X}\|_F^2$, where $\mathbf{L}$ is the Cholesky factor of $\boldsymbol{\Gamma}$. The matrix $\mathbf{L}^T\mathbf{C}_m\mathbf{X}$ is fortunately only $r \times n$.

6.9 UK Biobank Runtime Script

Here is the script used to perform our UK Biobank analysis

```
#
# Parameter explanations
# MvNormal: Distribution of traits is multivariate normal
# q: number of cross-validation folds
# init_beta: get good initial estimates for genetic and nongenetic regression coefficients
# min_iter: iterate at least 10 times before checking for convergence
#

using MendelIHT, Random
Random.seed!(2021)                        # seed for reproducibility
plinkfile = "ukb.plink.filtered"          # plink files without .bed/bim/fam
phenotypes = "normalized.bmi.sbp.dbp.txt" # comma-seprated phenotype file (no header line)
covariates = "iht.covariates.txt"         # comma-separated covariates file (no header line)

# cross-validate k = 100, 200, ..., 1000
path = 100:100:1000
@time mses = cross_validate(plinkfile, MvNormal, path=path, q=3,
    covariates=covariates, phenotypes=phenotypes, init_beta=true, min_iter=10,
    cv_summaryfile="cviht.summary.roughpath1.txt")

# cross-validate k = 110, 120, ..., 290 (assuming best k was 200 in previous step)
k_rough_guess = path[argmin(mses)]
```

```
path = (k_rough_guess - 9):10:(k_rough_guess + 9)
@time mses = cross_validate(plinkfile, MvNormal, path=path, q=3,
    covariates=covariates, phenotypes=phenotypes, init_beta=true, min_iter=10,
    cv_summaryfile="cviht.summary.roughpath2.txt")

# cross-validate k = 181, 182, ..., 209 (assuming best k was 190 in previous step)
k_rough_guess = path[argmin(mses)]
dense_path = (k_rough_guess - 9):(k_rough_guess + 9)
@time mses_new = cross_validate(plinkfile, MvNormal, path=dense_path,
    covariates=covariates, q=3, phenotypes=phenotypes, min_iter=10, init_beta=true)

# now run IHT on best k
@time iht_result = iht(plinkfile, dense_path[argmin(mses_new)], MvNormal,
    covariates=covariates, phenotypes=phenotypes, min_iter=10, init_beta=true)
```

7 Acknowledgements

This work is partially supported by National Institutes of Health grants T32-HG02536 (BC), R01-HG006139 (BC, KL, HZ), R35 GM141798 (KL, JS, HZ), R01-HG009120 (JS), K01DK106116 (JJZ), and R21HL150374 (JJZ); National Science Foundation grants DMS-1264153 (JS) and DMS-2054253 (HZ, JZ); and a National Research Foundation of Korea (NRF) grant 2020R1A6A3A03037675 (SK) from the Korean government (MSIT). The UCLA Institute for Digital Research and Education's Research Technology Group supplied computational and storage services through its Hoffman2 Shared Cluster.

8 Author Contributions

KL conceived the project. BC, JS, and KL devised the methods and simulations. BC, SK, and HZ wrote the software. JZ accessed and analyzed the UK Biobank data. BC wrote the initial draft of the paper. All authors reviewed and edited the draft.

9 Competing interests

The authors declare no competing interests.

10 Web Resources

Project name: MendelIHT.jl

Project home page: https://github.com/OpenMendel/MendelIHT.jl

Supported operating systems: Mac OS, Linux, Windows

Programming language: Julia 1.6

License: MIT

All commands needed to reproduce the following results are available at the `MendelIHT` site in the `manuscript` sub-folder. `SnpArrays.jl` is available at https://github.com/OpenMendel/SnpArrays.jl. `VCFTools.jl` is available at https://github.com/OpenMendel/VCFTools.jl. `BGEN.jl` is available at https://github.com/OpenMendel/BGEN.jl

References

[1] G. Abraham, Y. Qiu, and M. Inouye. FlashPCA2: principal component analysis of Biobank-scale genotype datasets. *Bioinformatics*, 2017.

[2] D. H. Alexander and K. Lange. Stability selection for genome-wide association. *Genetic Epidemiology*, 35(7):722–728, 2011.

[3] R. F. Barber and E. J. Candès. Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085, 2015.

[4] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

[5] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and computational Harmonic Analysis*, 27(3):265–274, 2009.

[6] P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1):232, 2011.

[7] B. B. Chu, K. L. Keys, C. A. German, H. Zhou, J. J. Zhou, E. M. Sobel, J. S. Sinsheimer, and K. Lange. Iterative hard thresholding in genome-wide association studies: Generalized linear models, prior weights, and double sparsity. *GigaScience*, 9(6):giaa044, 2020.

[8] A. P. Dawid. Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274, 1981.

[9] M. C. Delfour. *Introduction to Optimization and Semidifferential Calculus*. SIAM, 2012.

[10] M. A. Ferreira and S. M. Purcell. A multivariate test of association. *Bioinformatics*, 25(1):132–133, 2009.

[11] N. A. Furlotte and E. Eskin. Efficient multiple-trait association and estimation of genetic correlation using the matrix-variate linear mixed model. *Genetics*, 200(1):59–68, 2015.

[12] T. E. Galesloot, K. Van Steen, L. A. Kiemeney, L. L. Janss, and S. H. Vermeulen. A comparison of multivariate genome-wide association methods. *PloS One*, 9(4):e95923, 2014.

[13] Y. Guan and M. Stephens. Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *The Annals of Applied Statistics*, pages 1780–1815, 2011.

[14] S. S. Ji, C. A. German, K. Lange, J. S. Sinsheimer, H. Zhou, J. Zhou, and E. M. Sobel. Modern simulation utilities for genetic analysis. *BMC bioinformatics*, 22(1):1–13, 2021.

[15] L. Jiang, Z. Zheng, T. Qi, K. E. Kemper, N. R. Wray, P. M. Visscher, and J. Yang. A resource-efficient tool for mixed model association analysis of large-scale data. *Nature Genetics*, 51(12):1749–1755, 2019.

[16] K. L. Keys, G. K. Chen, and K. Lange. Iterative hard thresholding for model selection in genome-wide association studies. *Genetic Epidemiology*, 41(8):756–768, 2017.

[17] S. Ko, G. X. Li, H. Choi, and J.-H. Won. Computationally scalable regression modeling for ultrahigh-dimensional omics data with ParProx. *Briefings in Bioinformatics*, in press, 2021.

[18] S. Ko, H. Zhou, J. Zhou, and J.-H. Won. DistStat.jl: Towards unified programming for high-performance statistical computing environments in Julia. *arXiv preprint arXiv:2010.16114*, 2020.

[19] S. Ko, H. Zhou, J. J. Zhou, and J.-H. Won. High-performance statistical computing in the computing environments of the 2020s. *Statistical Science*, in press, 2021.

[20] K. Lange. *MM Optimization Algorithms*, volume 147. SIAM, 2016.

[21] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic Linear Algebra Subprograms for Fortran usage. *ACM Transactions on Mathematical Software*, 5(3):308–323, 1979.

[22] P.-R. Loh, G. Kichaev, S. Gazal, A. P. Schoech, and A. L. Price. Mixed-model association for biobank-scale datasets. *Nature Genetics*, 50(7):906–908, 2018.

[23] J. MacArthur, E. Bowler, M. Cerezo, L. Gil, P. Hall, E. Hastings, H. Junkins, A. McMahon, A. Milano, J. Morales, et al. The new NHGRI-EBI catalog of published genome-wide association studies (GWAS Catalog). *Nucleic Acids Research*, 45(D1):D896–D901, 2017.

[24] R. Mägi, Y. V. Suleimanov, G. M. Clarke, M. Kaakinen, K. Fischer, I. Prokopenko, and A. P. Morris. Scopa and META-SCOPA: software for the analysis and aggregation of genome-wide association studies of multiple correlated phenotypes. *BMC Bioinformatics*, 18(1):1–8, 2017.

[25] R. Magno and A.-T. Maia. gwasrapidd: an R package to query, download and wrangle GWAS Catalog data. *Bioinformatics*, pages 1–2, 2019.

[26] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics , 2nd edition*. John Wiley & Sons, 2019.

[27] P. F. O'Reilly, C. J. Hoggart, Y. Pomyen, F. C. Calboli, P. Elliott, M.-R. Jarvelin, and L. J. Coin. Multiphen: joint model of multiple phenotypes can increase discovery in GWAS. *PloS One*, 7(5):e34861, 2012.

[28] H. F. Porter and P. F. O'Reilly. Multivariate simulation framework reveals performance of multi-trait GWAS methods. *Scientific Reports*, 7(1):1–12, 2017.

[29] A. L. Price, N. A. Zaitlen, D. Reich, and N. Patterson. New approaches to population stratification in genome-wide association studies. *Nature Reviews Genetics*, 11(7):459–463, 2010.

[30] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. De Bakker, M. J. Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007.

[31] J. Qian, Y. Tanigawa, W. Du, M. Aguirre, C. Chang, R. Tibshirani, M. A. Rivas, and T. Hastie. A fast and scalable framework for large-scale and ultrahigh-dimensional sparse regression with application to the uk biobank. *PLoS Genetics*, 16(10):e1009141, 2020.

[32] C. Sabatti, S. K. Service, A.-L. Hartikainen, A. Pouta, S. Ripatti, J. Brodsky, C. G. Jones, N. A. Zaitlen, T. Varilo, M. Kaakinen, et al. Genome-wide association analysis of metabolic traits in a birth cohort from a founder population. *Nature Genetics*, 41(1):35, 2009.

[33] M. Sesia, S. Bates, E. Candès, J. Marchini, and C. Sabatti. FDR control in GWAS with population structure. *bioRxiv*, pages 2020–08, 2021.

[34] M. Stephens. A unified framework for association analysis with multiple related phenotypes. *PloS One*, 8(7):e65245, 2013.

[35] C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray, B. Liu, P. Matthews, G. Ong, J. Pell, A. Silman, A. Young, T. Sprosen, T. Peakman, and R. Collins. UK BioBank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Medicine*, 12:e1001779, 2015.

[36] M. Tanaka and K. Nakata. Positive definite matrix approximation with condition number constraint. *Optimization Letters*, 8(3):939–947, 2014.

[37] M. C. Turchin and M. Stephens. Bayesian multivariate reanalysis of large genetic studies identifies many new associations. *PLoS Genetics*, 15(10):e1008431, 2019.

[38] T. T. Wu, Y. F. Chen, T. Hastie, E. Sobel, and K. Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009.

[39] J. Yang, S. H. Lee, M. E. Goddard, and P. M. Visscher. GCTA: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82, 2011.

[40] J. Yin and H. Li. Model selection and estimation in the matrix normal graphical model. *Journal of Multivariate Analysis*, 107:119–140, 2012.

[41] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

[42] H. Zhou, D. Alexander, M. Sehl, J. Sinsheimer, E. Sobel, and K. Lange. Penalized regression for genome-wide association screening of sequence data. *Pacific Symposium on Biocomputing*, 2011:106–117, 2011.

[43] H. Zhou, K. Lange, and M. Suchard. Graphical processing units and high-dimensional optimization. *Statistical Science*, 25:311–324, 2010.

[44] H. Zhou, M. Sehl, J. Sinsheimer, and K. Lange. Association screening of common and rare genetic variants by penalized regression. *Bioinformatics*, 26:2357–2382, 2010.

[45] H. Zhou, J. S. Sinsheimer, D. M. Bates, B. B. Chu, C. A. German, S. S. Ji, K. L. Keys, J. Kim, S. Ko, G. D. Mosher, et al. OpenMendel: a cooperative programming project for statistical genetics. *Human Genetics*, 139(1):61–71, 2020.

[46] W. Zhou, J. B. Nielsen, L. G. Fritsche, R. Dey, M. E. Gabrielsen, B. N. Wolford, J. LeFaive, P. VandeHaar, S. A. Gagliano, A. Gifford, et al. Efficiently controlling for case-control imbalance and sample relatedness in large-scale genetic association studies. *Nature Genetics*, 50(9):1335–1341, 2018.

[47] X. Zhou and M. Stephens. Genome-wide efficient mixed-model analysis for association studies. *Nature Genetics*, 44(7):821, 2012.

[48] X. Zhou and M. Stephens. Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nature Methods*, 11(4):407–409, 2014.