Secure and federated linear mixed model association tests

Jeffrey Chen[1,2], Manaswitha Edupalli[1], Bonnie Berger[1,2,†], and Hyunghoon Cho[1,†]

[1]Broad Institute of MIT and Harvard, Cambridge, MA, 02142, USA
[2]MIT, Cambridge, MA, 02142, USA
[†]bab@mit.edu, hhcho@broadinstitute.org

Abstract

Privacy-preserving algorithms for genome-wide association studies (GWAS) promise to facilitate data sharing across silos to accelerate new discoveries. However, existing approaches do not support an important, prevalent class of methods known as linear mixed model (LMM) association tests or would provide limited privacy protection, due to the high computational burden of LMMs under existing secure computation frameworks. Here we introduce SafeGENIE, an efficient and provably secure algorithm for LMM-based association studies, which allows multiple entities to securely share their data to jointly compute association statistics without leaking any intermediary results. We overcome the computational burden of LMMs by leveraging recent advances in LMMs and secure computation, as well as a novel scalable dimensionality reduction technique. Our results show that SafeGENIE obtains accurate association test results comparable to a state-of-the-art centralized algorithm (REGENIE), and achieves practical runtimes even for large datasets of up to 100K individuals. Our work unlocks the promise of secure and distributed algorithms for collaborative genomic studies.

1 Introduction

Genome-wide association studies (GWAS) have been a major driving force behind genomics research [1, 2]. Ongoing efforts to amass large and diverse collections of genomic data, such as the All of Us Program [3] and the UK Biobank [4], will continue to bring new discoveries of genetic variants of biological and clinical importance. However, many of the existing genomic datasets are held in isolated repositories within national or organizational boundaries under strict data sharing limitations, presenting a key hurdle for collaborative research [4–6]. Achieving sufficient statistical power for rare diseases and underrepresented populations (e.g. admixed individuals) requires new strategies to facilitate sharing of genomic data across multiple data repositories.

To this end, recent studies have proposed a range of privacy-preserving solutions for GWAS. These works aim to allow analyses to be jointly performed on multiple parties' datasets without sharing the raw individual-level data, thus providing a path to circumvent regulatory restrictions. In particular, cryptographic approaches based on secure computation frameworks [7–10], such as homomorphic encryption (HE) [11, 12] and secure multiparty computation (MPC) [13], offer the strongest notions of data privacy, which state that any (encrypted) data that is externally shared by each party is statistically indistinguishable from random (under certain security assumptions, such as the parties do not collude in the case of MPC) [14]. Alternative approaches based on trusted execution environment (TEE) technology [15, 16] or distributed/federated algorithms [17, 18] have also been proposed, albeit providing weaker forms of privacy protection.

A key limitation shared by most existing solutions for privacy-preserving GWAS is that they consider simplified analysis workflows that do not reflect the current standard practice in genomics. Specifically, population stratification correction—a crucial step in GWAS for mitigating confounding effects arising from the implicit association between population structure in the study cohort and the target phenotype [19]—has long been unaddressed by solutions for privacy-preserving GWAS; the one exception is recent work that incorporated an approach based on principal components analysis (PCA) [13]. The gap between current solutions and practical workflows is in part due to the fact that existing techniques for privacy protection typically incur a computational overhead that grows with the complexity of the algorithm being implemented; this renders the task of making these sophisticated algorithms secure for modern GWAS highly challenging. Overcoming this barrier is a necessary step to realize the full potential of emerging privacy techniques.

Linear mixed models (LMMs) are one of two main classes of methods in the literature for addressing population stratification in GWAS. The other is a more traditional PCA-based approach [20], where the top principal components of the genotype matrix are included as fixed-effect covariates in the statistical models of genetic association to control for global population structure. In contrast, LMMs view the ancestry-related effect on the phenotype as a *random* effect, whose covariance is determined by the genetic relatedness patterns in the study cohort, also estimated from the data. LMMs are known to be more effective at capturing cryptic relatedness and fine-grain population structure within the study cohort [21]. Further enabled by recent algorithmic advances on reducing the computational burden of LMM computation [21–23], LMMs are increasingly becoming the preferred approach for GWAS [24]. However, little work exists in the literature on *privacy-preserving* LMM-based GWAS over decentralized, or distributed, datasets, which we ascribe to the complexity of existing LMM algorithms and their high computational cost even with full access to the data. This severe bottleneck hinders the use of privacy-preserving techniques for a full GWAS pipeline in practice.

In this work, we present SafeGENIE (secure and federated REGENIE), a privacy-preserving algorithm for performing LMM-based association tests on distributed genomic datasets. We develop a scalable algorithm for privacy-preserving LMM computation by synthesizing recent advances in cryptography, distributed algorithms, and population genetics, including: multiparty homomorphic encryption [25], distributed linear regression models [26], an efficient stacked ridge regression approach for LMMs (called REGENIE [27]). SafeGENIE provides formal security guarantees offered by the underlying cryptographic frameworks, including protection for all intermediate results, while maintaining efficiency by maximally leveraging local computation using plaintext (raw) data. Our work also incorporates new algorithmic strategies (e.g. linear algebra techniques for low-rank matrix update) to overcome the unique computational bottlenecks that arise when jointly utilizing the aforementioned techniques in SafeGENIE. To our knowledge, SafeGENIE is the first privacy-preserving algorithm to fully implement the LMM-based GWAS pipeline.

Our experimental results show that SafeGENIE produces LMM association statistics closely matching those of the plaintext algorithm REGENIE, demonstrating its utility for real-world studies. SafeGENIE also maintains runtimes on the order of days for datasets including up to a hundred thousand individuals and is efficient in memory usage. Although this runtime reflects the heavy computational burden of the underlying LMM computation, we demonstrate that our method is a significant improvement over the state-of-the-art baseline approaches and can be readily applied to many existing datasets. Our work represents an important step towards enabling a wide range of genomic analyses to be performed while protecting the private data.

2 Problem Definitions and Review of Existing Methods

2.1 Linear Mixed Model (LMM) Association Studies

We start by formally describing the LMMs used for genome-wide association tests. LMMs model the target phenotype vector $\mathbf{y}$ of length N individuals using the following linear model

$$\mathbf{y} = \beta_{\text{test}}\mathbf{x}_{\text{test}} + \mathbf{Z}\boldsymbol{\alpha} + \mathbf{g} + \mathbf{e},$$

where $\mathbf{x}_{\text{test}}$ is a vector of allele dosages of the variant being tested across N individuals, $\mathbf{Z}$ is a N-by-C matrix of observed covariates where C denotes the number of covariates, $\mathbf{g}$ represents the ambient genetic effect, and $\mathbf{e}$ represents the environmental effect. Both $\mathbf{x}_{\text{test}}$ and $\mathbf{y}$ are standardized to have zero mean and unit variance. We let N and M be the number of individuals and the number of variants in the dataset, respectively.

In this model, β_{test} and $\boldsymbol{\alpha}$ describe the fixed effect sizes associated with the tested variant and the covariates, respectively, whereas $\mathbf{g}$ and $\mathbf{e}$ are modeled as random effects (hence the term "mixed" model). Under the standard infinitesimal model, which posits that the genetic effect on phenotype consists of many small effect-size variants, we can express

$$\mathbf{g} = \mathbf{X}_{\text{LOCO}}\boldsymbol{\beta}$$

where $\mathbf{X}_{\text{LOCO}}$ is a N-by-M_{LOCO} matrix consisting of the standardized genotypes of M_{LOCO} variants used to model the genetic effect based on the standard leave-one-chromosome-out (LOCO) scheme, which excludes all variants in the same chromosome as the tested variant in order to avoid the effects of linkage disequilibrium. These ambient variants are associated with effect sizes $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, (\sigma_g^2/M_{\text{LOCO}})\mathbf{I}_{M_{\text{LOCO}}})$, inducing a distribution over the genetic effect as $\mathbf{g} \sim \mathcal{N}(0, \sigma_g^2\mathbf{K})$. $\mathbf{K} = \mathbf{X}_{\text{LOCO}}\mathbf{X}_{\text{LOCO}}^T/M_{\text{LOCO}}$ is referred to as the genetic relatedness (or empirical kinship) matrix. The environmental effect is modeled as $\mathbf{e} \sim \mathcal{N}(0, \sigma_e^2\mathbf{I}_N)$. Note that σ_g and σ_e represent the variances of the polygenic and environmental components. The goal of the association test is to test the null hypothesis $H_0 : \beta_{\text{test}} = 0$.

A standard technique [21,27] is to project the covariates out of the phenotypes and genotypes to simplify the computation. This results in a modified model

$$\tilde{\mathbf{y}} = \beta_{\text{test}}\tilde{\mathbf{x}}_{\text{test}} + \tilde{\mathbf{X}}_{\text{LOCO}}\boldsymbol{\beta} + \mathbf{e}, \tag{1}$$

where $\tilde{\mathbf{y}} = \mathbf{P}\mathbf{y}$, $\tilde{\mathbf{x}} = \mathbf{P}\mathbf{x}$, and $\tilde{\mathbf{X}}_{\text{LOCO}} = \mathbf{P}\mathbf{X}_{\text{LOCO}}$ with $\mathbf{P} = \mathbf{I}_N - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$. Note that $\mathbf{P}$ projects a vector onto the null space of $\mathbf{Z}$.

The LMM-based χ^2 test statistic is given by

$$\chi^2 = \frac{(\tilde{\mathbf{x}}_{\text{test}}^T\mathbf{V}_{\text{LOCO}}^{-1}\tilde{\mathbf{y}})^2}{\tilde{\mathbf{x}}_{\text{test}}^T\mathbf{V}_{\text{LOCO}}^{-1}\tilde{\mathbf{x}}_{\text{test}}}, \tag{2}$$

where $\mathbf{V}_{\text{LOCO}} = \hat{\sigma}_g^2\mathbf{K} + \hat{\sigma}_e^2\mathbf{I}_N$ given the estimates $\hat{\sigma}_g$ and $\hat{\sigma}_e$ of the variance parameters.

2.2 Review of REGENIE: Efficient Stacked Regression for LMMs

Computing the LMM association statistics is a computationally expensive task, in part because the maximum likelihood estimation of the variance parameter σ_g involves costly matrix operations involving the N-by-N genetic relatedness matrix (GRM), which becomes prohibitively large for large-scale datasets. Much of the prior algorithmic development efforts have focused on speeding up the use of GRM, e.g. by exploiting a factorization of the matrix [23]. A recent algorithm called REGENIE [27] introduced a different strategy based on *stacked regression*, resulting in significant scalability improvements for LMM-based association tests, while achieving comparable accuracy to existing state-of-the-art tools such as BOLT-LMM [21], fastGWA [22], and SAIGE [28]. Indeed, we observed similar performance between REGENIE and BOLT-LMM on the benchmark dataset used in our study (Supplementary Figure 1). Moreover, we discovered that REGENIE's approach is more amenable to efficient computation over distributed datasets, thereby forming an important basis for our SafeGENIE method.

In REGENIE [27], the whole-genome regression model in Equation 1 is estimated in two phases by first regressing out the contribution of $\tilde{\mathbf{X}}_{\text{LOCO}}$ from $\tilde{\mathbf{y}}$, then fitting β_{test} on the residuals to test for association. To further reduce the cost of regression over the large genome-wide matrix $\tilde{\mathbf{X}}_{\text{LOCO}}$, REGENIE employs a stacked regression approach in the following two steps, referred to as Levels 0 and 1.

Level 0. The genotype matrix is first split into B contiguous blocks of T variants:

$$\tilde{\mathbf{X}}_{\text{LOCO}} = (\tilde{\mathbf{X}}^{(1)}_{\text{LOCO}}, \tilde{\mathbf{X}}^{(2)}_{\text{LOCO}}, \ldots, \tilde{\mathbf{X}}^{(B)}_{\text{LOCO}}).$$

Then for each block $b \in [B]$ and different choices of the regularization parameter $\lambda_r \in \{\lambda_1, \ldots, \lambda_R\}$, where R represents the number of regularization parameters being tested, the following solution to the ridge regression problem $\tilde{\mathbf{y}} \approx \tilde{\mathbf{X}}^{(b)}_{\text{LOCO}}\boldsymbol{\beta}$ is computed.

$$\hat{\boldsymbol{\beta}}^{(b)}_{\lambda_r} := ((\tilde{\mathbf{X}}^{(b)}_{\text{LOCO}})^T \tilde{\mathbf{X}}^{(b)}_{\text{LOCO}} + \lambda_r \mathbf{I}_N)^{-1} (\tilde{\mathbf{X}}^{(b)}_{\text{LOCO}})^T \tilde{\mathbf{y}}, \tag{3}$$

$$\hat{\mathbf{y}}^{(b,r)}_{\text{LOCO}} := \tilde{\mathbf{X}}^{(b)}_{\text{LOCO}} \hat{\boldsymbol{\beta}}^{(b)}_{\lambda_r}. \tag{4}$$

The $\hat{\mathbf{y}}^{(b,r)}_{\text{LOCO}}$ is referred to as the *predictors*, representing the best polygenic prediction of the phenotype within a given genomic region, accounting for genotype correlations.

Level 1. The local predictors from Level 0 are aggregated to form a N-by-BR global feature matrix

$$\mathbf{W} := (\hat{\mathbf{y}}^{(1,1)}_{\text{LOCO}}, \ldots, \hat{\mathbf{y}}^{(B,R)}_{\text{LOCO}}). \tag{5}$$

Then another around of ridge regression is performed (with K-fold cross validation to choose the optimal regularization parameter η) to obtain the following genome-wide phenotype predictions:

$$\hat{\mathbf{y}}_{\text{LOCO}} := \mathbf{W}(\mathbf{W}^T\mathbf{W} + \eta \mathbf{I}_N)^{-1}\mathbf{W}\tilde{\mathbf{y}}. \tag{6}$$

Given this global predictor as a proxy for ambient genetic effect, the χ^2 statistic (with one degree of freedom) for the variant being tested in Equation 2 can now be formulated as

$$\chi^2 = \frac{(\tilde{\mathbf{x}}^T_{\text{test}}(\tilde{\mathbf{y}} - \hat{\mathbf{y}}_{\text{LOCO}}))^2}{\hat{\sigma}^2_e \cdot (\tilde{\mathbf{x}}^T_{\text{test}}\tilde{\mathbf{x}}_{\text{test}})}, \tag{7}$$

where $\hat{\sigma}^2_e = \|\tilde{\mathbf{y}} - \hat{\mathbf{y}}_{\text{LOCO}}\|^2_2/(N - C)$ is the estimated variance of the environmental effect.

The above approach substantially improves the speed of LMM computation by decomposing the problem into B separate local ridge regression problems, which can be performed in parallel in high performance computing environments. Importantly, by formulating the problem as a series of ridge regression tasks, the problem becomes more tractable for secure distributed computation, an aspect we exploit in our design of SafeGENIE to achieve efficiency.

2.3 Review of Privacy-Preserving GWAS Algorithms and Their Limitations

Prior works have developed privacy-aware algorithms for conducting GWAS over private datasets to facilitate genomic data sharing and collaboration. These methods leverage a range of different computational techniques with different strengths and weaknesses. Methods based on cryptographic frameworks such as secure multiparty computation (MPC) [7,29] and homomorphic encryption (HE) [9,30] aim to directly allow computation over encrypted data. MPC relies on interactive protocols among multiple parties to carry out the joint computation without revealing the private data, whereas HE enables non-interactive computation over the ciphertexts, albeit incurring higher computational overhead. GWAS algorithms based on both approaches have been developed [11–13], but they do not support LMM-based analyses.

Another branch of methods follow a distributed or federated algorithm design [17, 18], leveraging local computation performed by each party on their respective dataset combined with aggregation steps that exchange information among the parties to move towards a global solution. Although these methods enjoy higher computational flexibility and efficiency compared to cryptographic approaches, there is a general lack of understanding of the extent to which private information is leaked in the intermediate results shared among the parties, thus providing limited privacy protection. Notably, a recent work in this line presented a promising solution for distributed training of generalized LMMs [31], which is closely related to our work. However, we note that the prior method did not address association testing and supports only a single covariate, thus the LMM-based GWAS remains an open problem. Approaches based on trusted execution environments, such as the Intel SGX enclave, have also been proposed [15,16]. They perform computation on private data securely within a secure hardware component that is isolated from the rest of the host operating system. Similar to the plaintext distributed approach, these methods are efficient but offer limited privacy protection due to various security vulnerabilities that security researchers continue to discover, necessitating careful mitigation strategies (e.g. [32]). In this work, we adopt the cryptographic paradigm for secure computation, which offers the strongest notion of privacy, and focus on efficient algorithm design to address the associated computational burden.

3 Our Method: SafeGENIE

3.1 Overview of SafeGENIE

SafeGENIE is a privacy-preserving algorithm for LMM association analysis, which jointly analyzes private datasets held by different entities without leaking individual-level information. The results of SafeGENIE, by design, closely match those of running REGENIE on a pooled dataset with minimal loss in precision, while additionally protecting data privacy. The core computational framework of SafeGENIE is a hybrid of secure multiparty computation (MPC) and homomorphic encryption (HE), building upon the recently introduced framework of multiparty homomorphic encryption (MHE) [25,33–35]. Combining the ability of HE to perform non-interactive secure computation at each site with efficient MPC protocols for high-complexity operations such as comparison and inverse square root [13], our hybrid approach enables SafeGENIE to leverage efficient local plaintext computation while maintaining flexibility and accuracy in computational capabilities. In addition, our framework provides formal privacy guarantees by only exchanging *encrypted* forms of all intermediate results in accordance with established HE and MPC techniques throughout the protocol to facilitate the distributed computation (in contrast to other federated approaches that leak intermediate results and therefore sensitive data). Using these cryptographic tools, SafeGENIE implements our distributed algorithm for stacked regression for LMMs, inspired by the success of the centralized plaintext algorithm introduced by REGENIE. We introduce new algorithmic techniques for maximizing local computation that allow SafeGENIE to achieve near-constant scalability to larger datasets as we show in our results. We provide a graphical illustration of SafeGENIE in Figure 1.

3.2 Our Secure Computation Model

In our application setting, we consider P independent parties, each with a local GWAS dataset, who wish to jointly perform LMM association analysis on the pooled data without revealing private data. Our core approach is to keep these local datasets in plaintext and to encrypt, using MHE, only the intermediate results that need to be aggregated across the parties. MHE [25] is an extension of homomorphic encryption (HE) schemes where the decryption key is split among multiple parties using secret sharing, which ensures that encryption and homomorphic operations (performing computation over the secrets without decrypting the data) can be performed locally, while decryption requires all parties to cooperate, thus giving each party control over which pieces of results (such as the final association statistics) are revealed to other parties.

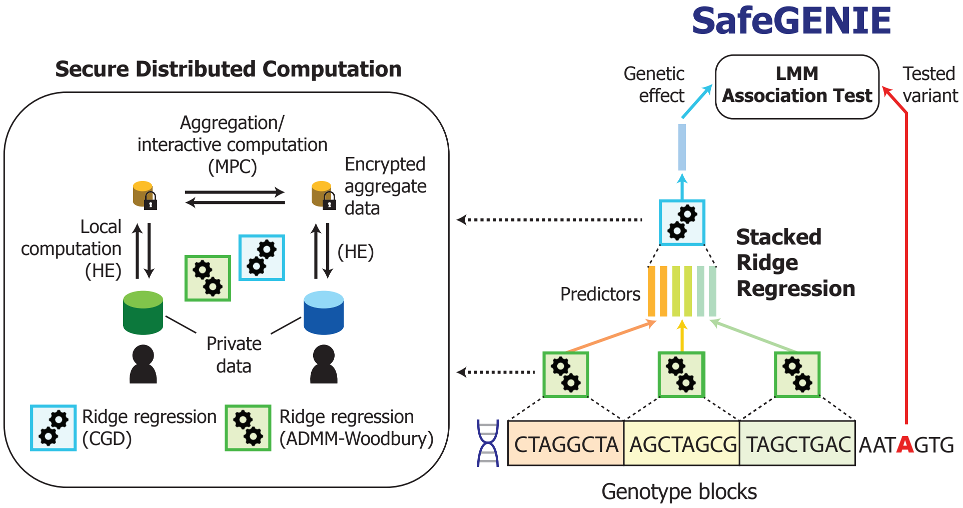


Figure 1: **Overview of SafeGENIE.** SafeGENIE implements a stacked ridge regression approach [27] to LMM association testing, whereby regression models trained on local genomic blocks are combined in another round of regression to produce estimated background genetic effects for the association tests. **(Left)** Both levels of ridge regression are performed using secure distributed algorithms we developed (CGD and ADMM-Woodbury), which leverage both homomorphic encryption and secure multiparty computation frameworks to provide privacy protection.

Importantly, MHE enables our distributed approach to secure computation, where we iterate between (1) a local computation phase, where each party computes local results leveraging both local plaintext data and encrypted shared data, and (2) an aggregation phase, where the parties interact to update the encrypted shared data, to perform the desired analysis. We additionally incorporate secret sharing-based MPC routines into our protocol by securely switching data representation between HE and secret shares. Secret shared data allow us to leverage efficient interactive protocols for more complex operations at the expense of higher communication. In our protocols, we securely perform comparison, inverse square root, and eigendecomposition over a small matrix using interactive MPC subroutines, while relying on HE for the rest. Note that, following the prior work on MPC-based GWAS [13], we adopt an efficient server-aided model for MPC, in which a coordinating party facilitates the computation by supplying the main parties with random numbers satisfying a certain structure. The coordinating party does not receive any portion of the private data other than the knowledge of data dimensions.

For the purposes of describing our distributed LMM algorithm implemented in SafeGENIE, we view our secure computation framework as a collection of building block protocols, each implementing a simple operation such as matrix multiplication or square root, which we compose to implement an end-to-end secure protocol that implements the LMM computation. As the focus of this work is on distributed algorithm design, in particular on optimizing the use of secure subroutines to efficiently perform LMM computation, we refer to relevant prior works on MPC and MHE for detailed descriptions of the subroutines used in SafeGENIE [12, 13].

3.3 Key Challenges of Distributed LMM Computation

To motivate our novel algorithmic techniques, we first describe the computational challenges in distributing the LMM computation. Recall that state-of-the-art LMM-based GWAS algorithms (e.g. BOLT-LMM [21]) account for population stratification by using the N-by-N genetic relatedness matrix (GRM) $\mathbf{K}$, which intuitively captures how individuals within a dataset are related to one another. A core computational step in the estimation of variance parameters or the calculation of association statistics is calculating a quantity of the form

$$(\mathbf{K} + \lambda \mathbf{I}_N)^{-1}\mathbf{v} = (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T/M + \lambda \mathbf{I}_N)^{-1}\mathbf{v}, \tag{8}$$

for some length-N vector $\mathbf{v}$. Because the size of $\mathbf{K}$ scales with the number of individuals in the dataset, for large-scale GWAS, this computation inherently incurs an overwhelming computational cost. As such, addressing this challenge has been the focus of recent algorithmic development efforts for LMM.

In our setting, the fact that the off-diagonal blocks of $\mathbf{K}$ describe relatedness between individuals in *different* collaborating sites introduces a unique difficulty in distributing the computation. When naïvely implemented, those interactive terms in $\mathbf{K}$ are bound to require heavy communication among parties to account for their contributions. Even recently proposed iterative approaches for efficiently solving this linear system of equations without the inverse (e.g. conjugate gradient descent used by BOLT-LMM [21]) presents a similar challenge, as it involves repeated multiplications of $\mathbf{K}$ with a candidate solution vector. Moreover, we note that $\mathbf{K}$ is typically defined over covariate-corrected genotypes $\tilde{\mathbf{X}} = \mathbf{PX}$, where $\mathbf{P}$ denotes a projection matrix for removing the covariate effect, which introduces another layer of entanglement between the private datasets at different sites, making distributed computation further challenging.

3.4 Our Approach: Secure Distributed Ridge Regression with Covariates

SafeGENIE approaches this problem differently. Following the methodology of REGENIE [27], instead of using the notion of a GRM, we train ridge regression models for local genomic windows to use as a proxy for ambient genetic effect on phenotype. Correcting for these polygenic predictions by testing for a variant's association with the phenotype residuals, our approach implicitly accounts for population stratification under the LMM formulation.

Our work identifies this alternative approach for LMMs as a key enabling factor for distributing the computation. Let $\tilde{\mathbf{X}}^{(b)}$ be a subset of columns from a genotype matrix corresponding to a genomic block with M_b variants. The ridge regression problem for LMM reduces to computing the expression

$$((\tilde{\mathbf{X}}^{(b)})^T\tilde{\mathbf{X}}^{(b)} + \lambda \mathbf{I}_{M_b})^{-1}(\tilde{\mathbf{X}}^{(b)})^T\tilde{\mathbf{y}}. \tag{9}$$

In contrast to Equation 8, we see that the inverse operation is for a matrix of size M_b-by-M_b, which is significantly smaller than $\mathbf{K}$ (note M_b is typically 1000). Furthermore, for horizontally distributed $\tilde{\mathbf{X}}^{(b)}$, where each party p holds a subset of rows in this matrix denoted $\tilde{\mathbf{X}}_p^{(b)}$, we have the following decomposition

$$(\tilde{\mathbf{X}}^{(b)})^T\tilde{\mathbf{X}}^{(b)} = \sum_{p=1}^{P}(\tilde{\mathbf{X}}_p^{(b)})^T\tilde{\mathbf{X}}_p^{(b)}. \tag{10}$$

This property allows SafeGENIE to distribute the computation more efficiently across the parties and maximally leverage plaintext data that is available locally. In the following section, we describe how we exploit this insight to design secure and distributed algorithms for conjugate gradient descent (CGD) and alternating direction method of multipliers (ADMM) approaches for ridge regression, which are used by SafeGENIE to carry out the Level 1 and Level 0 steps of REGENIE, respectively.

3.5 SafeGENIE Algorithm Details

Recall from Section 2.2 that stacked regression approach to LMM involves solving the following two ridge regression problems in Levels 0 and 1.

$$\hat{\boldsymbol{\beta}}_{\lambda_r}^{(b)} := ((\tilde{\mathbf{X}}_{\text{LOCO}}^{(b)})^T\tilde{\mathbf{X}}_{\text{LOCO}}^{(b)} + \lambda_r \mathbf{I}_N)^{-1}(\tilde{\mathbf{X}}_{\text{LOCO}}^{(b)})^T\tilde{\mathbf{y}}, \qquad (\textbf{Level 0}) \tag{11}$$

$$\hat{\mathbf{y}}_{\text{LOCO}} := \mathbf{W}(\mathbf{W}^T\mathbf{W} + \eta \mathbf{I}_N)^{-1}\mathbf{W}\tilde{\mathbf{y}}. \qquad (\textbf{Level 1}) \tag{12}$$

The first is solved KBR times for each pair of a block $b \in [B]$ and a regularization parameter λ_r for $r \in [R]$ using K-fold cross validation, and the second is solved KR times for each of R values of η using the same K-fold cross validation. The challenging step in both is the multiplication by the inverse matrix, which is infeasible to solve directly when the matrix is only available in encrypted form. This is unlike REGENIE, which explicitly solves for the inverse using eigenfactorization. Explicitly computing the inverse for a large, homomorphically encrypted matrix imposes a considerable computational burden, which we aim to avoid. To address this, we develop two algorithms described below.

3.5.1 Secure Distributed Conjugate Gradient Descent (CGD)

Conjugate Gradient Descent (CGD) [36] is a well-known iterative algorithm for solving a system of linear equations without explicitly constructing the inverse of the design matrix. Notably, BOLT-LMM heavily utilizes the CGD algorithm to avoid working with the inverse of GRM. A requirement of CGD is that the design matrix be positive definite and well conditioned; in our setting, the regularization term in ridge regression ensures this property [37]. Hence, CGD can be applied to any of the ridge regression problems in our task. The central step in CGD is a multiplication of a candidate solution vector with the design matrix (not the inverse), which lends itself to efficient distributed computation. We outline our distributed CGD algorithm in Algorithm 1.

Algorithm 1 Distributed Conjugate Gradient Descent for Ridge Regression

```
Input: Number of parties $P$, horizontally distributed input matrix $\mathbf{A} = [\mathbf{A}_1^T, \ldots, \mathbf{A}_P^T]^T$, target vector $\mathbf{b}$, regularization parameter $\lambda$, number of iterations $\tau$.
Output: A vector $\mathbf{x}$ that satisfies $(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})\mathbf{x} \approx \mathbf{b}$.
  Initialize: $\mathbf{x}_0 \leftarrow \mathbf{0}$, $\mathbf{y}_0 \leftarrow \mathbf{b}$, $\mathbf{r}_0 \leftarrow \mathbf{b}$
  for $k \in \{0, \ldots, \tau - 1\}$ do
      Each party $p$ locally computes $\mathbf{z}_p \leftarrow \mathbf{A}_p^T\mathbf{A}_p\mathbf{y}_k$
      $\mathbf{z} \leftarrow \mathsf{SumAggregate}(\{\mathbf{z}_1, \ldots, \mathbf{z}_P\})$
      $\mathbf{u} \leftarrow \mathbf{z} + \lambda\mathbf{y}_k$                    ▷ $\mathbf{u} = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})\mathbf{y}_k$
      $\alpha \leftarrow (\mathbf{r}_k^T\mathbf{r}_k)/(\mathbf{y}_k^T\mathbf{z})$
      $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha\mathbf{y}_k$
      $\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k - \alpha\mathbf{z}$
      $\beta \leftarrow (\mathbf{r}_{k+1}^T\mathbf{r}_{k+1})/(\mathbf{r}_k^T\mathbf{r}_k)$
      $\mathbf{y}_{k+1} \leftarrow \mathbf{r}_k + \beta\mathbf{y}_k$
  end for
  return $\mathbf{x}_\tau$
```

We highlighted in blue the modified step in the distributed approach. As explained in Section 3.4, the fact that the design matrix $\mathbf{A}^T\mathbf{A}$ in our setting decomposes as a sum of local design matrices $\mathbf{A}_p^T\mathbf{A}_p$ allows this step to be performed independently, then aggregated after both multiplications (the SumAggregate step). Thus, the required communication scales with the number of predictive features (variants), not the number of samples held by each party, thereby offering better scaling to large datasets. To apply this algorithm to encrypted datasets, each of the calculations, namely matrix-vector multiplication, inner products, and addition/subtraction, are implemented using HE routines, with the exception of division, for which we switch to a secret sharing-based MPC routine. Note that the SumAggregate step involves each party broadcasting their share to others and adding up all shares (homomorphically), which does not involve any decryption. For efficiency, this procedure is implemented over a star network where a central coordinator aggregates all data and in turn relays the result to all parties.

3.5.2 Secure ADMM for Ridge Regression with Covariates

Although CGD offers a natural distributed solution for ridge regression, it is still computationally burdensome for large input matrices. For instance, consider applying CGD to Level 0 of REGENIE, where the input is given as

$$\mathbf{A} = \tilde{\mathbf{X}}^{(b)} = [(\tilde{\mathbf{X}}_1^{(b)})^T, \ldots, (\tilde{\mathbf{X}}_P^{(b)})^T]^T. \tag{13}$$

When each party performs the following local computation in Algorithm 1

$$\mathbf{z}_p \leftarrow (\tilde{\mathbf{X}}_p^{(b)})^T\tilde{\mathbf{X}}_p^{(b)}\mathbf{y}_k, \tag{14}$$

they first need to multiply $\mathbf{y}_k$ with $\tilde{\mathbf{X}}_p^{(b)}$, which has an output dimension of N_p (number of individuals in party p's dataset), followed by another multiplication with $(\tilde{\mathbf{X}}_p^{(b)})^T$, finally resulting in a vector of length M_b. This is due to the fact that $(\tilde{\mathbf{X}}_p^{(b)})^T\tilde{\mathbf{X}}_p^{(b)}$ cannot be precomputed in plaintext, since $\tilde{\mathbf{X}}_p^{(b)}$ requires covariate correction involving all parties covariate data. Note that M_b, the block size, is a user parameter typically set to a small value (e.g. 1000) whereas N_p can grow much larger for large-scale datasets. Therefore, CGD does not benefit from any dimension reduction (to M_b) that is otherwise exhibited in the plaintext formulation.

SafeGENIE overcomes this challenge by leveraging the alternating direction method of multipliers (ADMM) technique [38], which is a powerful method for transforming convex optimization problems into distributed optimization problems that can be more efficiently solved. Intuitively, ADMM relaxes the global objective by decoupling the terms involving each individual dataset, which in turn can be jointly optimized using local update equations that, in our case, involve plaintext matrices of size M_b as desired. Our techniques draw inspiration from a recent work in security literature, which introduced a secure multiparty ADMM algorithm for distributed linear regression [26]. Our work extends this work to the setting where the design matrix must be covariate-corrected, which introduces additional challenges as we describe below.

Here we describe how we apply ADMM to the ridge regression problem in Level 0 of REGENIE. Recall that the ridge regression of $\tilde{\mathbf{y}}$ onto $\tilde{\mathbf{X}}^{(b)}$ with regularization parameter λ can be equivalently formulated as the following optimization problem:

$$\text{minimize}_{\mathbf{w}} \quad \frac{1}{2}\|\tilde{\mathbf{X}}^{(b)}\mathbf{w} - \tilde{\mathbf{y}}\|_2^2 + \frac{1}{2}\lambda\|\mathbf{w}\|_2^2. \tag{15}$$

To apply ADMM, we first decouple the two terms using a slack variable $\mathbf{z}$ with an equality constraint as follows.

$$\begin{aligned} \text{minimize}_{\mathbf{w},\mathbf{z}} \quad & \frac{1}{2}\|\tilde{\mathbf{X}}^{(b)}\mathbf{w} - \tilde{\mathbf{y}}\|_2^2 + \frac{1}{2}\lambda\|\mathbf{z}\|_2^2, & (16) \\ \text{s.t.} \quad & \mathbf{w} - \mathbf{z} = 0. & (17) \end{aligned}$$

Next, we note that the first objective term can be written as a sum of squared loss computed over each party's dataset as

$$\|\tilde{\mathbf{X}}^{(b)}\mathbf{w} - \tilde{\mathbf{y}}\|_2^2 = \sum_{p=1}^{P} \|\tilde{\mathbf{X}}_p^{(b)}\mathbf{w} - \tilde{\mathbf{y}}_p\|_2^2, \tag{18}$$

where we partition $\tilde{\mathbf{y}} = [\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_P]$ in the same manner as $\tilde{\mathbf{X}}^{(b)}$. Finally, further decoupling the $\mathbf{w}$ parameters across parties for distributed optimization, we obtain

$$\begin{aligned} \text{minimize}_{\mathbf{w}_p,\mathbf{z}} \quad & \frac{1}{2}\sum_{p=1}^{P} \|\tilde{\mathbf{X}}_p^{(b)}\mathbf{w}_p - \tilde{\mathbf{y}}_p\|_2^2 + \frac{1}{2}\lambda\|\mathbf{z}\|_2^2, & (19) \\ \text{s.t.} \quad & \mathbf{w}_p - \mathbf{z} = 0, \forall p. & (20) \end{aligned}$$

The resulting iterative optimization procedure based on the standard ADMM derivation, for general local matrices $\mathbf{A}_1$,. . . ,$\mathbf{A}_P$, is shown in Algorithm 2.

Algorithm 2 Standard ADMM Algorithm for Ridge Regression (adapted from [38])

```
Input: Number of parties $P$, horizontally distributed input matrix $\mathbf{A} = [\mathbf{A}_1^T, \ldots, \mathbf{A}_P^T]^T$, target vector $\mathbf{b} = [\mathbf{b}_1^T, \ldots, \mathbf{b}_P^T]^T$, regularization parameter $\lambda$, learning rate $\rho$, number of iterations $\tau$.
Output: A vector $\mathbf{z}$ that satisfies $(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})\mathbf{z} \approx \mathbf{b}$.
  Each party $p$:
      Locally computes $\mathbf{R}_p^{-1} \leftarrow (\mathbf{A}_p^T\mathbf{A}_p + \rho I)^{-1}$
      Initializes $\mathbf{w}_p^0 \leftarrow 0, \mathbf{u}_p^0 \leftarrow 0$ and a global vector $\mathbf{z}^0 \leftarrow 0$
  for $k \in \{0, ..., \tau - 1\}$ do
      Each party $p$ locally computes $\mathbf{w}_p^{k+1} \leftarrow \mathbf{R}_p^{-1}(\mathbf{b} + \rho\mathbf{z}^k - \mathbf{u}_p^k)$
      $\bar{\mathbf{w}}^{k+1} \leftarrow \mathsf{SumAggregate}(\{\mathbf{w}_1^{k+1}, \ldots, \mathbf{w}_P^{k+1}\})/P$
      $\bar{\mathbf{u}}^k \leftarrow \mathsf{SumAggregate}(\{\mathbf{u}_1^k, \ldots, \mathbf{u}_P^k\})/P$
      Each party $p$ locally computes:
          $\mathbf{z}^{k+1} \leftarrow (\rho\bar{\mathbf{w}}^{k+1} + \bar{\mathbf{u}}^k)/(\lambda/P + \rho)$
          $\mathbf{u}_p^{k+1} \leftarrow \mathbf{u}_p^k + \rho(\mathbf{w}_p^{k+1} - \mathbf{z}^{k+1})$
  end for
  return $\mathbf{z}^\tau$
```

However, we note that our given $\mathbf{A}_p = \tilde{\mathbf{X}}_p^{(b)}$ is not available in plaintext, as it is meant to be standardized and covariate-corrected based on the global matrix $\tilde{\mathbf{X}}^{(b)}$. Therefore, although each party has access to their own raw genotype martix $\mathbf{X}_p^{(b)}$, they are not able to precompute the following matrix shown in Algorithm 2 in plaintext:

$$\mathbf{R}_p^{-1} = ((\tilde{\mathbf{X}}_p^{(b)})^T \tilde{\mathbf{X}}_p^{(b)} + \rho \mathbf{I}_{M_b})^{-1}. \tag{21}$$

In SafeGENIE, we introduce a technique to resolve this issue by using the Woodbury matrix identity [39] to perform covariate correction in the computation of $\mathbf{R}_p^{-1}$ on the fly as follows. First, recall that

$$\tilde{\mathbf{X}}^{(b)} = (\mathbf{I}_N - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T)\mathbf{X}^{(b)}\mathbf{S}^{(b)}, \tag{22}$$

where $\mathbf{Z}$ is an N-by-C covariate matrix, and $\mathbf{S}$ represents a diagonal matrix with inverse standard deviations for each column of $\mathbf{X}^{(b)}$. We include an all-ones vector as a covariate in $\mathbf{Z}$, which implicitly accounts for mean centering of $\mathbf{X}^{(b)}$. With one round of aggregation, we precompute a small C-by-M_b matrix

$$\mathbf{H}^{(b)} = \mathbf{Z}^T\mathbf{X}^{(b)} = \sum_{p=1}^{P} \mathbf{Z}_p^T \mathbf{X}_p^{(b)}, \tag{23}$$

where each summand is computed locally using plaintext matrices then aggregated in an encrypted form. Next, noting that

$$\tilde{\mathbf{X}}^{(b)} = (\mathbf{X}^{(b)} - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)})\mathbf{S}^{(b)}, \tag{24}$$

we are able to express $\mathbf{R}_p^{-1}$ (Equation 21) as

$$\mathbf{R}_p^{-1} = \mathbf{S}^{-1}(([(\mathbf{X}_p^{(b)})^T\mathbf{X}_p^{(b)} + \rho\mathbf{I}_{M_b}] + \mathbf{U}_p\mathbf{E}_p\mathbf{V}_p)^{-1}\mathbf{S}^{-1}, \tag{25}$$

for some matrices $\mathbf{U}_p, \mathbf{V}_p^T \in \mathbb{R}^{M_b \times 2C}$ and $\mathbf{E}_p \in \mathbb{R}^{2C \times 2C}$ (see Appendix for full derivation; note the inner dimension of $2C$). Finally, using the Woodbury identity and letting $\mathbf{Q} := (\mathbf{X}_p^{(b)})^T\mathbf{X}_p^{(b)} + \rho\mathbf{I}_{M_b}$ to simplify the notation, we can expand the inverse as

$$(\mathbf{Q}_p + \mathbf{U}_p\mathbf{E}_p\mathbf{V}_p)^{-1} = \mathbf{Q}_p^{-1} - \mathbf{Q}_p^{-1}\mathbf{U}_p(\mathbf{E}_p^{-1} + \mathbf{V}_p\mathbf{Q}_p^{-1}\mathbf{U}_p)^{-1}\mathbf{V}_p\mathbf{Q}_p^{-1}. \tag{26}$$

We have successfully reformulated the computation of $\mathbf{R}_p^{-1}$ as one involving an *inverse of a plaintext matrix* $\mathbf{Q}_p$ and several matrix multiplications with a small inner dimension of $2C$. Note that the new composite inverse matrix

$$\mathbf{L}_p := (\mathbf{E}_p^{-1} + \mathbf{V}_p\mathbf{Q}_p^{-1}\mathbf{U}_p)^{-1} \tag{27}$$

can be efficiently computed using secure MPC protocols, using the eigenfactorization routine introduced in prior work [13].

As a result, we can compute a key matrix $\mathbf{Q}_p^{-1}$ completely locally in plaintext, then use Equations 7 and 2 to compute the multiplication with $\mathbf{R}_p^{-1}$ on the fly using the plaintext $\mathbf{Q}_p^{-1}$. Note that this step is the only expensive matrix multiplication in the ADMM algorithm, and as such our reformulated ADMM offers significant reduction in computational cost. Moreover, we emphasize that, aside from the precomputation of $\mathbf{H}^{(b)}$ and $\mathbf{Q}_p^{-1}$, none of the matrix operations in our ADMM algorithm scales with the number of individuals in the dataset, and thus scales very efficiently to datasets with many samples, as our results show. Our final ADMM algorithm for ridge regression with covariates, leveraging the Woodbury identity technique, is presented in Algorithm 3 (changes with respect to the standard formulation are shown in blue).

3.6 Computational Complexity of SafeGENIE

SafeGENIE greatly reduces the runtime of a direct implementation of REGENIE in a distributed setting. By using our improved ADMM algorithm, we delegate large matrix inverse operations to be performed locally in plaintext. Since in practice the overhead of cryptographic operations greatly overshadows that of plaintext computation, in our complexity analysis we only consider homomorphic operations over the encrypted data.

The implementation of ADMM with our Woodbury optimization is separated into two components, the precomputation of a small matrix inverse in the Woodbury identity and the main ADMM iterations. For

Algorithm 3 Our Improved ADMM Algorithm for Ridge Regression with Covariates

```
Input: Number of parties $P$, horizontally distributed input matrix $\mathbf{X} = [\mathbf{X}_1^T, \ldots, \mathbf{X}_P^T]^T$ and a covariate matrix $\mathbf{Z} = [\mathbf{Z}_1^T, \ldots, \mathbf{Z}_P^T]^T$, a diagonal matrix $\mathbf{S}$ with inverse standard deviations of columns of $\mathbf{X}$, target vector $\mathbf{b} = [\mathbf{b}_1^T, \ldots, \mathbf{b}_P^T]^T$, regularization parameter $\lambda$, learning rate $\rho$, number of iterations $\tau$.
Output: A vector $\mathbf{z}$ that satisfies $(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})\mathbf{z} \approx \mathbf{b}$, where $\mathbf{A} := (\mathbf{I} - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T)\mathbf{X}\mathbf{S}$
  Each party $p$ locally computes $\mathbf{H}_p \leftarrow \mathbf{Z}_p^T\mathbf{X}_p$
  $\mathbf{H} \leftarrow \mathsf{SumAggregate}(\{\mathbf{H}_1, \ldots, \mathbf{H}_P\})$
  Each party $p$:
      Initializes $\mathbf{w}_p^0 \leftarrow 0, \mathbf{u}_p^0 \leftarrow 0$ and a global vector $\mathbf{z}^0 \leftarrow 0$
      Locally computes $\mathbf{Q}_p^{-1} \leftarrow (\mathbf{X}_p^T\mathbf{X}_p + \rho\mathbf{I}_{M_b})^{-1}$
  Precompute $\mathbf{L}_p := (\mathbf{E}_p^{-1} + \mathbf{V}_p\mathbf{Q}_p^{-1}\mathbf{U}_p)^{-1}$ in Equation 27 for all parties
  for $k \in \{0, ..., \tau - 1\}$ do
      Each party $p$:
          Locally computes $\mathbf{h} \leftarrow \mathbf{Q}_p^{-1}\mathbf{S}^{-1}(\mathbf{b} + \rho\mathbf{z}^k - \mathbf{u}_p^k)$
          Locally computes $\mathbf{w}_p^{k+1} \leftarrow \mathbf{S}^{-1}(\mathbf{h} - \mathbf{Q}_p^{-1}\mathbf{U}_p\mathbf{L}_p\mathbf{V}_p\mathbf{h})$
      $\bar{\mathbf{w}}^{k+1} \leftarrow \mathsf{SumAggregate}(\{\mathbf{w}_1^{k+1}, \ldots, \mathbf{w}_P^{k+1}\})/P$
      $\bar{\mathbf{u}}^k \leftarrow \mathsf{SumAggregate}(\{\mathbf{u}_1^k, \ldots, \mathbf{u}_P^k\})/P$
      Each party $p$ locally computes:
          $\mathbf{z}^{k+1} \leftarrow (\rho\bar{\mathbf{w}}^{k+1} + \bar{\mathbf{u}}^k)/(\lambda/P + \rho)$
          $\mathbf{u}_p^{k+1} \leftarrow \mathbf{u}_p^k + \rho(\mathbf{w}_p^{k+1} - \mathbf{z}^{k+1})$
  end for
  return $\mathbf{z}^\tau$
```

the precomputation of the $\mathbf{L}_p$ matrix (Equation 27), each party performs a M_b-by-M_b matrix multiplication around $2C$ times where C is the number of covariates, where M_b is the blocksize. Each main iteration is dominated by the work of multiplying a M_b-by-M_b plaintext matrix $\mathbf{Q}_p^{-1}$ twice with a ciphervector, combined with cipher-cipher multiplications with precomputed matrices $\mathbf{U}_p$, $\mathbf{L}_p$, and $\mathbf{V}_p$. Like in the plaintext setting, matrix vector multiplication in HE scales linearly with the size of the matrix. Since ridge regression in Level 0 is computed for each block (B), cross-validation fold (K), and regularization parameter (R), the complexity of Level 0 is $O(KBRT^2\tau)$ where τ is the number of iterations in ADMM. Since $M = BM_b$ by definition, this can be expressed as $O(M_bKRM\tau)$. We note that this is a much better asymptotic runtime than naively using CGD for Level 0. Each iteration of CGD scales linearly with the size of the matrix being multiplied, which is N-by-M_b in our case. Therefore, the total runtime of Level 0 with CGD is $O(M_bKBRN\tau)$, or equivalently $O(KRMN\tau)$, which is a factor of N/M_b larger than our ADMM method. For a dataset of $N = 10^5$ and $M_b = 10^3$, this amounts to a factor of 100 improvement using our ADMM solution, asymptotically.

For Level 1, since we evaluate the CGD subroutine lazily without explicitly constructing the design matrix, we distribute the work in a way where each party multiplies their respective plaintext genotype matrix with a vector. Since the size of the matrix is N-by-BR, and there are KR different ridge regressions that must be performed, the runtime complexity of Level 1 is $O(KR^2NB\tau)$, where τ is the number of iterations in CGD (typically 30), and K and R are small numbers (default values of 5 in REGENIE).

Lastly, to compute association statistics, much of the work can be done in plaintext. The chromosome-specific LOCO residual vectors (observed phenotypes minus the LOCO predictors of genetic effect) must be covariate corrected and then multiplied by the genotype matrix for the corresponding chromosome. This is equivalent to one matrix-vector multiplication between the full genotype matrix (plaintext) and a residual vector (ciphertext). While this leads to an asymptotic complexity of $O(NM)$, in practice since only one such multiplication is needed and because cipher-plain multiplications are considerably more efficient than cipher-cipher multiplications, computing association statistics is the quickest of the three levels of computation.

3.7 Numerical Stability of SafeGENIE

There are several hyperparameters and statistics to take note of with SafeGENIE's use of iterative methods for ridge regression that factor into the numerical stability and convergence time of SafeGENIE. Our ADMM algorithm for Level 0 depends on the hyperparameter ρ, which represents the step size used in the ADMM iterations. We have found that in practice, an effective ρ can be determined as a function of the number of individuals, since the genomic block size is typically chosen within a fixed range of 1K to 5K variants for

REGENIE. In our experiments, we set ρ to the number of individuals in the local dataset divided by the number of cross-validation folds.

Another hyperparameter of interest for both CGD and ADMM is the number of iterations. The convergence of both methods are generally dependent on the condition number of the input matrix [37, 40] and thus rely heavily on the magnitude of the regularization parameter relative to the input matrix. Therefore, we implemented an adaptive approach that checks for convergence at regular intervals and terminates when a suitable solution has been obtained. We also employed a warm start strategy whereby solutions from a higher value of the regularization parameter is used to initialize the run with a lower value, which considerably improved convergence overall.

We also note that precision is challenging to maintain in general in secure computation protocols given their dependence on fixed-point representation of continuous numbers and the possibilities of numerical under/overflow. Throughout our algorithm, we carefully managed the range of data values by scaling intermediate results accordingly (e.g. dividing by $\sqrt{n}$ when computing the sum of n values for large n). As our results show, the combination of these strategies allow SafeGENIE to obtain accurate association analysis results.

3.8 Implementation Details

SafeGENIE is implemented using the distributed CKKS framework in Lattigo [33], which is an open source library in Golang for homomorphic encryption schemes. We extended the library by implementing secret sharing-based MPC functionalities based on our prior work [13]. In addition, SafeGENIE was implemented using Golang's built in multi-threading framework and networking protocols, which allow interactive computation over encrypted data to be performed in a parallel manner across multiple machines. Our implementation also features efficient streaming pipelines for accessing blocks of the genotype matrix, which helps ensures that the memory usage of the program stays low regardless of the size of the dataset.

4 Experimental Results

4.1 SafeGENIE obtains accurate association statistics

To demonstrate the performance of SafeGENIE on a real GWAS dataset, we obtained a dataset of 9,178 East Asian lung cancer patients (5,054) and control individuals (4,053) from the dbGaP repository (accesion phs000716.v1.p1). After a quality control filter excluding individuals with missingness rate higher than 10%, we retained 9,098 individuals. We included in the analysis 378,482 single nucleotide polymorphisms (SNPs) that passed minor allele frequency (>0.1) and Hardy-Weinberg equilibirum ($\chi^2 < 28.374$) filters. We used a block size of 3000 SNPs for local ridge regressions in Level 0. For the association tests, we applied the standard leave-one-chromosome-out (LOCO) scheme, leaving out one chromosome at a time where the tested variant resides to correct for the background genetic effect.

For experimental setup, we created a set of three VM instances in the Google Cloud Platform, each with 128 RAM and 16 virtual CPUs, located in the same geographic zone. One VM served the role of a coordinating party for server-aided MPC routines, with the other two as main data holders participating in the collaborative LMM analysis. We split the GWAS data into two sets of individuals (4550 and 4548 individuals, respectively) and individually uploaded the corresponding genotype, phenotype, and covariate data to the two main parties' VMs. We then executed the SafeGENIE program with point-to-point communication channels between pairs of parties for interactive steps of the protocol. For Level 0, we used additional sets of VMs to further parallelize the computation over the blocks, the results of which were later combined (and the reported runtime appropriately aggregated) before proceeding with the rest of the protocol. We used 12 threads on each machine to leverage parallelism in each step of the pipeline.

Figure 2 shows the resulting LMM-based association statistics from SafeGENIE compared against the output from running REGENIE (obtained from `https://rgcgithub.github.io/regenie/`) on a pooled dataset. The first two Manhattan plots show the genome-wide association signals are nearly identical between the two approaches, suggesting that SafeGENIE successfully replicates the analysis performed by REGENIE in a centralized setting. Note that SafeGENIE never has access to the whole data in one site; it only jointly analyzes distributed datasets in a secure manner using our cryptographic protocols. We observe that in this dataset a particularly strong association is identified for SNP rs2736100, which is associated with the *TERT* gene, a known cancer gene whose expression is associated with general increased cancer risk and is hypothesized to impair telomere maintenance [41]. Quantitatively measuring the agreement between the two outputs resulted in a correlation coefficient of 0.9845 (Figure 2C). We also observed a strong agreement in the genome-wide polygenic predictions of the phenotype (Figure 2D), which represent a key intermediate result in the LMM analysis.

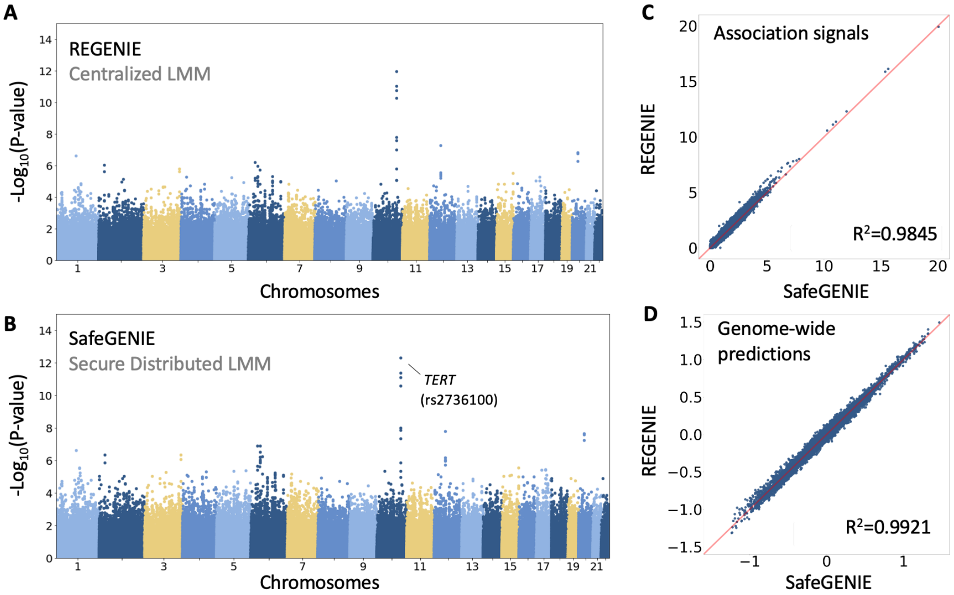


Figure 2: **SafeGENIE closely reproduces REGENIE association statistics while securely analyzing distributed GWAS datasets.** We evaluated SafeGENIE on a real lung cancer GWAS dataset including 9,098 individuals and 378,482 SNPs split between two parties. The Manhattan plot for SafeGENIE mirrors the Manhattan plot obtained from running the centralized REGENIE on the same lung cancer dataset (**A**, **B**). (**C**) Comparison of the negative log p-values for all variants in the dataset generated by REGENIE and by SafeGENIE. (**D**) Plot of the genome-wide phenotype prediction vectors obtained by both methods, which are provided as input to the association testing pipeline. Both show that the results from SafeGENIE are highly correlated to those of REGENIE with an R^2 value over 0.98 in both plots.

4.2 SafeGENIE efficiently scales to large datasets

The total runtime of SafeGENIE on the lung cancer dataset was 68.3 hours with a communication of 11 TB using 12 threads. Most of the runtime as well as communication can be attributed to Level 0 (61.4 hours and 11 TB; compared to 6.33 hours and 44.3 GB of Level 1), which fits many local ridge regression models across genomic blocks. We note that this step is embarrassingly parallel and thus can be sped up with more cores. In contrast, a baseline CGD solution for Level 0 without our ADMM-Woodbury algorithm, is estimated to take 216.75 hours of runtime (3.5 times slower) on 12 threads and 7.5 TB of communication (0.7 times less). Our runtime improvement is expected to be even greater at larger scales given our reduced dependence on N as described in Section 3.6.

We next evaluated the scalability of SafeGENIE on upsampled lung cancer datasets up to ten times its original size (91K individuals for the largest dataset) (Figure 2). We observed that the runtime of SafeGENIE is dominated by Level 0, whose runtime remained near-constant over the range of data sizes we tested. This incredible scalability is a result of the dimensionality reduction that is evident in our ADMM-Woodbury algorithm. Recall that to perform ridge regression over a genomic block, our algorithm calculates the inverse of a M_b-by-M_b matrix in plaintext (where M_b is the block size) and perform all subsequent operations using this matrix, effectively removing dependence of runtime on the dataset size N. In contrast, the baseline conjugate gradient descent (CGD) solution for Level 0 ridge regressions (see Algorithm 1) grows linearly

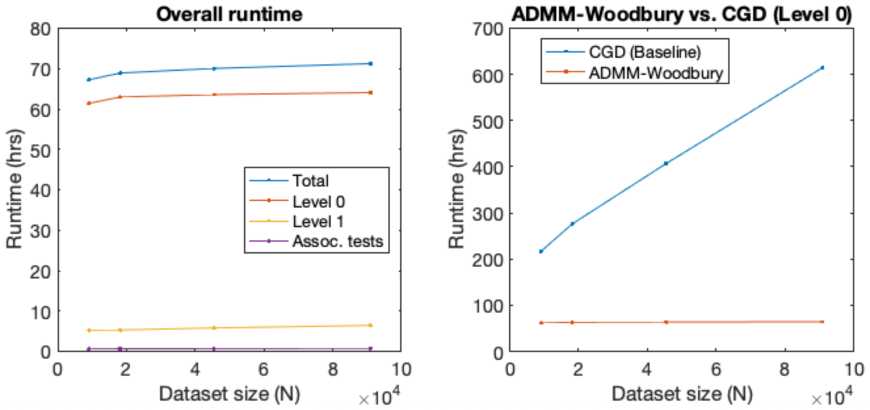


Figure 3: **SafeGENIE achieves near-constant scalability with respect to runtime to larger datasets.** We show the runtime of SafeGENIE on upsampled lung cancer datasets including up to 91K thousand individuals (ten times the original dataset). Runtimes shown are estimated based on a smaller number of blocks or iterations across which the computational load is expected to be identical. Measurements are based on a network of three co-located machines on Google Cloud with 12 cores each for parallelization. SafeGENIE's runtime remains near-constant as the size of the dataset grows, enabled by the efficient distributed algorithms introduced in SafeGENIE, which maximize the use of local plaintext computation. Right subfigure shows the comparison of our ADMM-Woodbury algorithm for Level 0 with the baseline CGD algorithm (which we also newly developed for the secure distributed setting), demonstrating the improved asymptotic complexity of our approach. For the largest dataset, our approach achieves a 9.6-fold speedup over the baseline.

with N, resulting in a wider gap in performance compared to our algorithm for larger datasets (Figure 2); on the largest dataset, our ADMM-Woodbury solution achieves a factor of 9.6 speedup over CGD. Note that, while plaintext CGD is a widely used technique for ridge regression (also used in BOLT-LMM [21]), here we compared to our novel *secure distributed* implementation of CGD, which we used in Level 1. Since the majority of work in Level 0 consists of separate invocations of ridge regression on genomic blocks, this step can also be parallelized across multiple CPUs and machines to further reduce the runtime in practice. These results demonstrate the practical feasibility of SafeGENIE for large GWAS datasets including hundreds of thousands of individuals.

5 Discussion

We introduced SafeGENIE, a privacy-preserving and distributed approach to LMM association studies. Leveraging the insight that a recent stacked regression approach to LMM presents a path for efficient distributed computation, we developed efficient distributed algorithms for ridge regression with covariates for use as core routines in SafeGENIE. Our results show that SafeGENIE produces nearly identical association results compared to REGENIE, a centralized LMM algorithm, while demonstrating efficient runtime performance (in the order of days) with near-constant scalability to larger datasets. Directions for future work include evaluation on biobank-scale cohorts; increasing the robustness of SafeGENIE to a wide range of parameter settings (e.g. including extreme values of variance estimates); and further developing methods to support other types of association tests based on LMM beyond the quantitative trait model addressed in this work. We expect the insights introduced in our work on how to design efficient distributed algorithms for secure computation to be of broad interest to enhancing privacy in other genomic analysis workflows.

References

[1] Ruth J.F Loos. 15 years of genome-wide association studies andno signs of slowing down. *Nature Communications*, 11(5900), 2020.

[2] Peter M Visscher, Naomi R Wray, Qian Zhang, Pamela Sklar, Mark I McCarthy, Matthew A Brown, and Jian Yang. 10 Years of GWAS Discovery: Biology, Function, and Translation. *AJHG*, 101(1):5–22, 2017.

[3] The All of US Research Program. *New EnglandJournal of Medicine*, 381(7):668–676, 2019.

[4] Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, Bette Liu, Paul Matthews, Giok Ong, Jill Pell, Alan Silman, Alan Young, Tim Sprosen, Tim Peakman, and Rory 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(3):1–10, 03 2015.

[5] John Micheal Gaziano, John Concato, Mary Brophy, Louis Fiore, Saiju Pyarajan, James Breeling, Stacey Whitbourne, Jennifer Deen, Coleen Shannon, Donald Humphries, Peter Guarino, Mihaela Aslan, Daniel Anderson, Rene LaFleur, Timothy Hammond, Kendra Schaa, Jennifer Moser, Grant Huang, Sumitra Muralidhar, Ronald Przygodzki, and Timothy J O'Leary. Million Veteran Program: a mega-biobank to study genetic influences on health and disease. *J. Clin. Epidemiol*, 70:214–223, 2016.

[6] Zhengming Chen, Junshi Chen, Rory Collins, Yu Guo, Richard Peto, Fan Wu, and Liming Li. China Kadoorie Biobank of 0.5 million people: survey methods, baseline characteristics and long-term follow-up. *Int. J. Epidemiol*, 40:1652–1666, 2011.

[7] Andrew C. Yao. Protocols for secure computations. *IEEE Annual Symposium on Foundations of Computer Science*, 1982.

[8] Ronald Cramer and Ivan Bjerre Damgård. *Secure Multiparty Computation.* Cambridge University Press, 2015.

[9] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. *Cryptology ePrint Archive, Report 2016/421*, 2016.

[10] Junfeng Fan and Frederik Vercauteren. Somewhat Practical Fully Homomorphic Encryption. 2012.

[11] Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, and Shafi Goldwasser. Secure large-scale genome-wide association studies using homomorphic encryption. *Proceedings of the National Academy of Sciences*, 117(21):11608–11613, 2020.

[12] David Froelicher, Juan R Troncoso-Pastoriza, Jean Louis Raisaro, Michel A Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature Communications*, 12(1), 2021.

[13] Hyunghoon Cho, David J Wu, and Bonnie Berger. Secure genome-wide association analysis using multiparty computation. *Nature Biotechnology*, 36(6):547–551, 2018.

[14] Bonnie Berger and Hyunghoon Cho. Emerging technologies towards enhancing privacy in genomic data sharing. *Genome Biology*, 20(128), 2019.

[15] Antoine Boutet, Túlio Pascoal, Jérémie Decouchant, and Paulo Esteves-Verissimo. DyPS: Dynamic, Private and Secure GWAS. pages 1–19, 2021.

[16] Can Kockan, Kaiyuan Zhu, Natnatee Dokmai, Nikolai Karpov, M.Oguzhan Kulekci, David P. Woodruff, and S. Cenk Sahinalp. Sketching algorithms for genomic data analysis and querying in a secure enclave. *Nature Methods*, 2020.

[17] Reihaneh Nasirigerdeh, Reza andTorkzadehmahani, Julian Matschinske, Tobias Frisch, Markus List, Julian Späth, Uwe Weiß, Stefan andVölker, Dominik Heider, Nina Kerstin Wenke, Tim Kacprowski, and Jan Baumbach. splink: A federated, privacy-preserving tool as a robust alternative to meta-analysis in genome-wide association studies. *bioRxiv*, 2020.

[18] Md Nazmus Sadat, Md Momin Al Aziz, Noman Mohammed, Feng Chen, Xiaoqian Jiang, and Shuang Wang. SAFETY: Secure gwAs in Federated Environment through a hYbrid Solution. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(1):93–102, 2019.

[19] Matthew L Freedman, David Reich, Kathryn L Penney, Gavin J McDonald, Andre A Mignault, Nick Patterson, Stacey B Gabriel, Eric J Topol, Jordan W Smoller, Carlos N Pato, et al. Assessing the impact of population stratification on genetic association studies. *Nature genetics*, 36(4):388–393, 2004.

[20] Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904–909, 2006.

[21] Po-Ru Loh, George Tucker, Brendan K Bulik-Sullivan, Bjarni J Vilhjalmsson, Hilary K Finucane, Rany M Salem, Daniel I Chasman, Paul M Ridker, Benjamin M Neale, Bonnie Berger, et al. Efficient bayesian mixed-model analysis increases association power in large cohorts. *Nature genetics*, 47(3):284–290, 2015.

[22] Longda Jiang, Zhili Zheng, Ting Qi, Kathryn E Kemper, Naomi R Wray, Peter M Visscher, and Jian Yang. A resource-efficient tool for mixed model association analysis of large-scale data. *Nature genetics*, 51(12):1749–1755, 2019.

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

[24] Po-Ru Loh, Gleb Kichaev, Steven Gazal, Armin P Schoech, and Alkes L Price. Mixed-model association for biobank-scale datasets. *Nature genetics*, 50(7):906–908, 2018.

[25] Christian Mouchet, Juan Troncoso-Pastoriza, Jean-Philippe Bossuat, and Jean-Pierre Hubaux. Multiparty Homomorphic Encryption from Ring-Learning-with-Errors. *Proceedings on Privacy Enhancing Technologies*, 2021(4):291–311, 2021.

[26] Wenting Zheng, Raluca Ada Popa, Joseph E Gonzalez, and Ion Stoica. Helen: Maliciously secure coopetitive learning for linear models. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 724–738. IEEE, 2019.

[27] J. Mbatchou, L. Barnard, and J. et al. Backman. Computationally efficient whole-genome regression for quantitative and binary traits. *Nat Genet*, 53:1097–1103, 2021.

[28] Wei Zhou, Jonas B Nielsen, Lars G Fritsche, Rounak Dey, Maiken E Gabrielsen, Brooke N Wolford, Jonathon LeFaive, Peter VandeHaar, Sarah A Gagliano, Aliya 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.

[29] Ronald Cramer, Ivan Damgård, and Ueli Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 316–334. Springer, 2000.

[30] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

[31] Rui Zhu, Chao Jiang, Xiaofeng Wang, Shuang Wang, Hao Zheng, and Haixu Tang. Privacy-preserving construction of generalized linear mixed model for biomedical computation. *Bioinformatics*, 36(Supplement_1):i128–i135, 07 2020.

[32] Natnatee Dokmai, Can Kockan, Kaiyuan Zhu, XiaoFeng Wang, S Cenk Sahinalp, and Hyunghoon Cho. Privacy-preserving genotype imputation in a trusted execution environment. *Cell Systems*, 12(01):983–993, 2021.

[33] Christian Mouchet, Jean-Philippe Bossuat, Juan Troncoso-Pastoriza, and Jean-Pierre Hubaux. Lattigo: A multiparty homomorphic encryption library in GO. *Workshop on Encrypted Computing and Applied Homomorphic Cryptography*, 2021.

[34] David Froelicher, Juan R. Troncoso-Pastoriza, Apostolos Pyrgelis, Sinem Sav, Joao Sa Sousa, Jean-Philippe Bossuat, and Jean-Pierre Hubaux. Scalable Privacy-Preserving Distributed Learning. *Privacy Enhancing Technologies Symposium*, 2021.

[35] Sinem Sav, Apostolos Pyrgelis, Juan R. Troncoso-Pastoriza, David Froelicher, Jean-Philippe Bossuat, Joao Sa Sousa, and Jean-Pierre Hubaux. POSEIDON: Privacy-Preserving Federated Neural Network Learning. *Network and Distributed Systems Security Symposium*, 2021.

[36] William W Hager and Hongchao Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, 16(1):170–192, 2005.

[37] *Iterative Solution of Large Sparse Systems of Equations*. Springer, Cham, Switzerland, 2016.

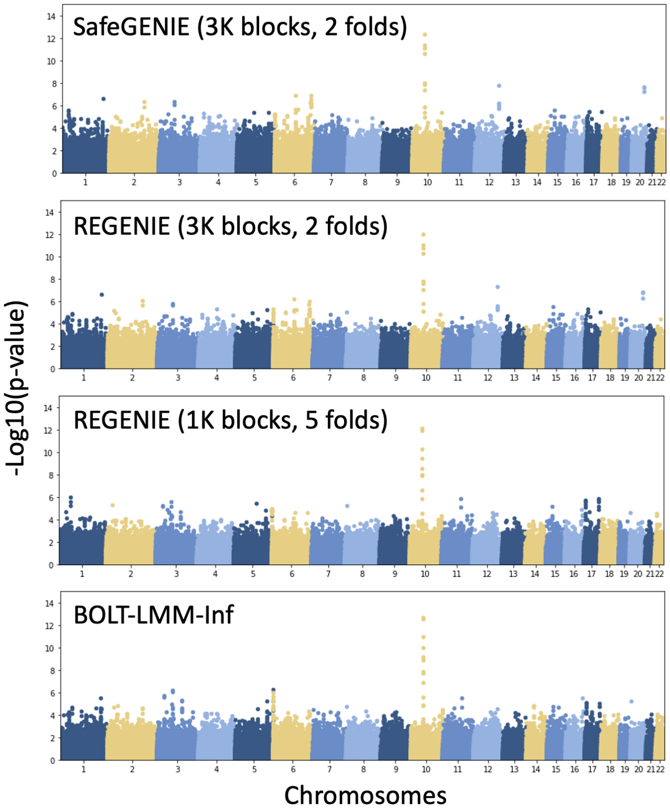
[38] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[39] William W Hager. Updating the inverse of a matrix. *SIAM review*, 31(2):221–239, 1989.

[40] Robert Nishihara, Laurent Lessard, Benjamin Recht, Andrew Packard, and Michael I. Jordan. A General Analysis of the Convergence of ADMM. *International Conference on Machine Learning*, 32, 02 2015.

[41] H. Li, Y. Xu, H. Mei, L. Peng, X. Li, and J. Tang. The TERT rs2736100 polymorphism increases cancer risk: A meta-analysis. *Oncotarget*, 8(24):38693–38705, Jun 2017.

Appendix



Supplementary Figure 1: **Comparable performance of REGENIE and BOLT-LMM-Inf on the lung cancer dataset.** We compared the association results of REGENIE on the lung cancer dataset with BOLT-LMM, a popular method for LMM analysis. Top two figures reproduce our results in Figure 2, which are based on a parameter setting with 3K variants per block and 2 folds. These results demonstrate that SafeGENIE closely matches the performance of REGENIE. Bottom two figures show similar results between BOLT-LMM-Inf (based on the infinitesimal model analogous to REGENIE) and REGENIE with 1K variants per block and 5 folds for cross-validation. Because our dataset includes a moderate number of individuals (9098) and SNPs (378K), we expect smaller block sizes and more folds to lead to better approximation performance in general for REGENIE, which explains the increased discrepancy between BOLT-LMM and REGENIE in our parameter setting. Note that SafeGENIE can run with any suitable choice of parameters for a given dataset to emulate the centralized REGENIE analysis in a distributed setting.

Derivation of Improved ADMM based on the Woodbury Identity

To better exploit the power of ADMM, we aim to expand the following inverse which denotes the inverse matrix that is computed locally by each party.

$$\mathbf{R}_p^{-1} = ((\tilde{\mathbf{X}}_p^{(b)})^T \tilde{\mathbf{X}}_p^{(b)} + \rho \mathbf{I}_{M_b})^{-1}. \tag{1}$$

The Woodbury Matrix identity allows for the expansion of inverses into a new inverse with a small rank one update if the initial matrix is of the following form

$$(\mathbf{Q} + \mathbf{UEV})^{-1} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{U}(\mathbf{E}^{-1} + \mathbf{VQ}^{-1}\mathbf{U})^{-1}\mathbf{VQ}^{-1}. \tag{2}$$

We can begin by expanding $\tilde{\mathbf{X}}_p^{(b)}$ into its components. Namely, we know that

$$\tilde{\mathbf{X}}^{(b)} = (\mathbf{X}^{(b)} - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)})\mathbf{S}^{(b)}. \tag{3}$$

Therefore,

$$\mathbf{R}_p^{-1} = (((\mathbf{X}^{(b)} - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)})\mathbf{S}^{(b)})^T(\mathbf{X}^{(b)} - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)})\mathbf{S}^{(b)} + \rho \mathbf{I}_{M_b})^{-1}, \tag{4}$$

After flipping the inner transpose, we get the following

$$\mathbf{R}_p^{-1} = (\mathbf{S}^{(b)}((\mathbf{X}^{(b)})^T - (\mathbf{H}^{(b)})^T(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T)(\mathbf{X}^{(b)} - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)})\mathbf{S}^{(b)} + \rho \mathbf{I}_{M_b})^{-1}, \tag{5}$$

After distribution the expression becomes

$$\mathbf{R}_p^{-1} = ((\mathbf{X}^{(b)})^T\mathbf{X}^{(b)} + \rho \mathbf{I}_{M_b} - (\mathbf{H}^{(b)})^T(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{X}^{(b)} - (\mathbf{X}^{(b)})^T(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)} + (\mathbf{H}^{(b)})^T(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)})^{-1},$$

We then consolidate two terms together.

$$\mathbf{R}_p^{-1} = ((\mathbf{X}^{(b)})^T\mathbf{X}^{(b)} + \rho \mathbf{I}_{M_b} - (\mathbf{X}^{(b)})^T(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)} - (\mathbf{H}^{(b)})^T(\mathbf{Z}^T\mathbf{Z})^{-1}(\mathbf{Z}^T\mathbf{X}^{(b)} - \mathbf{Z}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)}))^{-1},$$

For simplicity of notation let $\mathbf{J} = \mathbf{Z}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)}$. We can then simply the expression to

$$\mathbf{R}_p^{-1} = ((\mathbf{X}^{(b)})^T\mathbf{X}^{(b)} + \rho \mathbf{I}_{M_b} - (\mathbf{X}^{(b)})^T(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{H}^{(b)} - (\mathbf{H}^{(b)})^T(\mathbf{Z}^T\mathbf{Z})^{-1}(\mathbf{Z}^T\mathbf{X}^{(b)} - \mathbf{J}))^{-1},$$

Let M_b denote blocksize and C denote the number of covariates. Now let

$$\mathbf{U} = \begin{bmatrix} \mathbf{H}^{(b)} & (\mathbf{X}^{(b)})^T\mathbf{Z} \end{bmatrix},$$

which is a $M_b \times 2C$ matrix

$$\mathbf{E} = \begin{bmatrix} \mathbf{Z}^T\mathbf{Z} & 0 \\ 0 & \mathbf{Z}^T\mathbf{Z} \end{bmatrix}^{-1},$$

which is a $2C \times 2C$ matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{Z}^T\mathbf{X}^{(b)} - \mathbf{J} \\ \mathbf{H}^{(b)} \end{bmatrix},$$

which is a $2C \times M_b$ matrix

$$\mathbf{R}_p^{-1} = \mathbf{S}^{-1}(([\mathbf{X}_p^{(b)})^T\mathbf{X}_p^{(b)} + \rho \mathbf{I}_{M_b}] + \mathbf{UEV})^{-1}\mathbf{S}^{-1}, \tag{6}$$

Now, using the Woodbury Matrix Identity, we can set $\mathbf{Q} = (\mathbf{X}_p^{(b)})^T\mathbf{X}_p^{(b)} + \rho \mathbf{I}_{M_b}$ and expand the inverse

$$\mathbf{R}_p^{-1} = \mathbf{S}^{-1}[\mathbf{Q}^{-1} - \mathbf{Q}^{-1}\mathbf{U}(\mathbf{E}^{-1} + \mathbf{VQ}^{-1}\mathbf{U})^{-1}\mathbf{VQ}^{-1}]\mathbf{S}^{-1}, \tag{7}$$

Note that $(\mathbf{E}^{-1} + \mathbf{VQ}^{-1}\mathbf{U})^{-1}$ is a very small inverse matrix which is $2C \times 2C$ in size. This allows us to compute this inverse explicitly without much issue. The above expression is mathematically equivalent to $\mathbf{R}_p^{-1}$ and thus can be substituted for any instance of $\mathbf{R}_p^{-1}$ in the original ADMM algorithm. This substitution is what becomes Algorithm 3—our improved ADMM algorithm for ridge regression with covariates.

Association testing

Here we outline how association testing is performed in SafeGENIE. For each variant g, we wish to calculate

$$s := \frac{\tilde{g}^T \hat{y}^*_{\text{resid,LOCO}}}{\hat{\sigma}_e \cdot \sqrt{\tilde{g}^T \tilde{g}}}.$$

Note that

$$\tilde{g} = \sigma_g^{-1} P g,$$

where $P = (I - Z(Z^T Z)^{-1} Z^T)$. Since we include a column of ones in the covariate matrix Z for mean shift correction, we do not need to apply mean correction in advance to g. Also, since the scaling factor σ_g^{-1} cancels between the denominator and the numerator, we can set it to one without affecting the result, i.e. let $\tilde{g} = Pg$. Thus, we can express the target quantity as

$$s = \frac{g^T P \hat{y}^*_{\text{resid,LOCO}}}{\hat{\sigma}_e \cdot \sqrt{g^T P g}}$$

We first directly calculate $\hat{y}^*_{\text{resid,LOCO}}$ and $\hat{\sigma}_e$. Then, we compute $g^T P \hat{y}^*_{\text{resid,LOCO}}$ by first computing

$$w := P \hat{y}^*_{\text{resid,LOCO}}$$

then computing

$$\tilde{g}^T \hat{y}^*_{\text{resid,LOCO}} = g^T w.$$

Note that w is computed once per chromosome and shared across all variants.

Next we consider $g^T P g$. We have access to a matrix R such that $RR^T = (Z^T Z)^{-1}$, using which we can express $P = I - ZRR^T Z^T$. Define

$$u := R^T Z^T g.$$

Then we can compute

$$g^T P g = g^T g - u^T u.$$

To further consolidate the computation above, we can take the following approach. We first compute

$$w := P \hat{y}^*_{\text{resid,LOCO}}.$$

Then, we take a single pass over the input genotype matrix to compute the following terms for each column g:

$$g^T g, \; Z^T g, \; w^T g.$$

Note that the first two are computed in plaintext and easily converted to secret shares, where each party sets their own share to the computed plaintext value such that the implicit sum of shares simply becomes the overall sum. The third term involves a ciphervector w and thus can be treated as ciphervector-plainmatrix multiplication.

Given these terms, we can compute u using the secret shares, then compute $g^T P g$ as well as its inverse square-root using the secret shares as well. Afterwards, we need only multiply the numerator with the inverse of the denominator (taking $\hat{\sigma}_e$ into account) and obtain the final statistics.