Border and Becker *BMC Bioinformatics* (2019) 20:411
https://doi.org/10.1186/s12859-019-2978-z

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

Stochastic Lanczos estimation of genomic variance components for linear mixed-effects models

Richard Border[1,2*] and Stephen Becker[3]

Abstract

Background: Linear mixed-effects models (LMM) are a leading method in conducting genome-wide association studies (GWAS) but require residual maximum likelihood (REML) estimation of variance components, which is computationally demanding. Previous work has reduced the computational burden of variance component estimation by replacing direct matrix operations with iterative and stochastic methods and by employing loose tolerances to limit the number of iterations in the REML optimization procedure. Here, we introduce two novel algorithms, *stochastic Lanczos derivative-free REML* (`SLDF_REML`) and *Lanczos first-order Monte Carlo REML* (`L_FOMC_REML`), that exploit problem structure via the principle of Krylov subspace shift-invariance to speed computation beyond existing methods. Both novel algorithms only require a single round of computation involving iterative matrix operations, after which their respective objectives can be repeatedly evaluated using vector operations. Further, in contrast to existing stochastic methods, `SLDF_REML` can exploit precomputed genomic relatedness matrices (GRMs), when available, to further speed computation.

Results: Results of numerical experiments are congruent with theory and demonstrate that interpreted-language implementations of both algorithms match or exceed existing compiled-language software packages in speed, accuracy, and flexibility.

Conclusions: Both the `SLDF_REML` and `L_FOMC_REML` algorithms outperform existing methods for REML estimation of variance components for LMM and are suitable for incorporation into existing GWAS LMM software implementations.

Keywords: GWAS, Linear mixed-effects models, Variance components, REML, Conjugate gradients, Stochastic trace estimation, Stochastic Lanczos quadrature

Background

Linear mixed-effects modeling (LMM) is a leading methodology employed in genome-wide association studies (GWAS) of complex traits in humans, offering the dual benefits of controlling for population stratification while permitting the inclusion of data from related individuals [1]. However, the implementation of LMM comes at the cost of increased computational burden relative to ordinary least-squares regression, particularly in performing residual maximum likelihood (REML) estimation of genomic variance components. Conventional REML algorithms require multiple $\mathcal{O}\left(n^3\right)$ or $\mathcal{O}\left(mn^2\right)$ matrix operations, where m and n are the numbers of markers and individuals, respectively, rendering them infeasible for large biobank scale data sets. Further, common numerical methods for REML estimation rely on sparse matrix methods suitable for traditional LMM applications (e.g., pedigree data or experiments with repeated measures [2]) that are inapplicable to genomics variance components models since these models involve dense relatedness matrices. As a result, the problem of increasing the computational efficiency of REML estimation of genomic

*Correspondence: richard.border@colorado.edu
[1]Institute for Behavioral Genetics, University of Colorado Boulder, 80309, Boulder, CO, USA
[2]Department of Psychology and Neuroscience, University of Colorado Boulder, 80309, Boulder, CO, USA
Full list of author information is available at the end of the article

variance components has generated considerable research activity [3–8].

In the case of the standard two variance component model (1), the estimation of which is the focus of the current research, previous efforts toward increasing computational efficiency fit into two primary categories: 1., reducing the number of cubic time complexity matrix operations needed to achieve convergence; and 2., substituting stochastic and iterative matrix operations for deterministic, direct methods to obtain procedures with quadratic time complexity. The first approach is embodied by the methods implemented in the FaST-LMM and GEMMA packages [3, 5, 6], which take advantage of the fact that the genetic relatedness matrix (GRM) and identity matrix comprising the covariance structure are simultaneously diagonalizable. As a result, after performing a single spectral decomposition of the GRM and a small number of matrix-vector multiplications, the REML criterion (3) and its gradient and Hessian can be repeatedly evaluated using only vector operations. The second approach is exemplified by the popular BOLT-LMM software [7, 8], which avoids all cubic operations by solving linear systems via the method of conjugate gradients (CG) and employing stochastic trace estimators in place of deterministic computations.

In the current research, we propose two algorithms, stochastic Lanczos derivative-free residual maximum likelihood (`SLDF_REML`; Algorithm 3) and Lanczos first-order Monte Carlo residual maximum likelihood (`L_FOMC_REML`; Algorithm 4), that combine features of both approaches (Fig. 1). Here, we translate the simultaneous diagonalizability of the heritable and non-heritable components of the covariance structure to stochastic and iterative methods via the principle of Krylov subspace shift-invariance. As a result, we only need to compute the costliest portions of the objective function once (via stochastic/iterative methods), computing all subsequent iterations of the REML optimization problem only using vector operations. We develop the theory underlying these methods and demonstrate their performance relative to previous methods via numerical experiment.

Results

Across 20 replications per condition for random subsamples of n=16,000 to 256,000 unrelated European-ancestry individuals, both `SLDF_REML` and `L_FOMC_REML` produced heritability estimates for height consistent with those generated by the GCTA software package (Figs. 4 and 5). For large samples, the novel algorithms achieved greater accuracy than either version of BOLT-LMM (e.g., for n=250,000, mean-squared error was 1.74×10^{-6} for BOLT-LMM v2.3.2 versus 1.24×10^{-7} for `L_FOMC_REML`). Particularly, the time required per additional iteration after initial overhead computations was low for the novel algorithms (e.g., $\bar{t}$=20.07 min for BOLT-LMM v2.3.2 versus 2.06 min for `L_FOMC_REML`; Table 2), enabling increased precision at minor cost. With respect to total timings, `SLDF_REML` dramatically outperformed all other methods when the precomputed GRM was available (Table 2 and Fig. 3), which we expect whenever the number of markers exceeds the sample size. Examining methods taking genotype matrices as inputs, `SLDF_REML` and `L_FOMC_REML` performed similarly, whereas BOLT-LMM v2.3.2 converged more quickly than either in smaller samples (Fig. 3), though the differences for n=256,000 were relatively minor (e.g.,

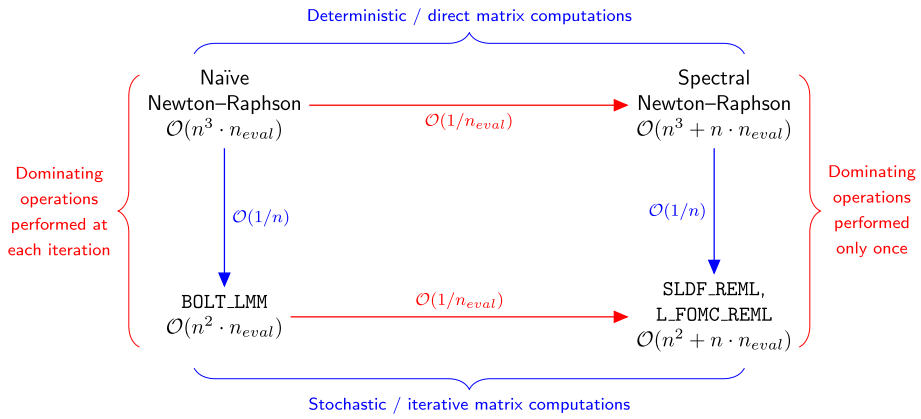

Fig. 1 Time complexity analogies with respect to existing and proposed methods. Heuristically, the novel algorithms (bottom right) are to the stochastic, iterative algorithm implemented in the BOLT-LMM software [7, 8] (bottom left) as the direct methods exploiting the shifted structure of the two component genomic variance component model (1) (e.g., FaST-LMM and GEMMA [3, 5]; top right) are to standard direct methods (top left). For simplicity, we assume here that the number of markers is equal to the number of observations and omit low-order terms related to the spectral conditioning of the covariance structure and the number of random vectors generated by the stochastic methods; further details are provided in Table 1. n_{eval} denotes the number of objective function evaluations needed to achieve convergence

$\bar{t}$ =91.09 min for BOLT-LMM v2.3.2 versus 102.21 min for `L_FOMC_REML`; Table 2). The older version of BOLT-LMM, v2.1, performed significantly more slowly than any of the other implementations examined (e.g., average wall clock time was 177.95 min at n=256,000), demonstrating the importance of implementation optimization.

As the computations needed to compute the Lanczos decompositions in `L_FOMC_REML` and BOLT-LMM v2.3.2 are equivalent in time and memory complexity, we expect that an optimized compiled-language implementation of `L_FOMC_REML` would reduce the overhead computation time by a significant linear factor (≈3 for n=256,000, comparing the sum of the overhead time and single objective function evaluation time for BOLT-LMM v2.3.2 to its total running time; Table 2). Consistent with theory, the wall clock times per objective function evaluation for the novel algorithms were trivial given the Lanczos decompositions (e.g., for n=256,000, $\bar{t}$ = 2.06 versus 20.07 min for `L_FOMC_REML` and BOLT-LMM v2.3.2, respectively; Table 2 and Fig. 2).

Discussion

We have proposed stochastic algorithms for estimating the two variance component model (1), both of which theoretically offer substantial time savings relative to existing methods. Our methods capitalize on the principle of Krylov subspace shift invariance to reduce the number of steps involving $\mathcal{O}\left(n^2\right)$ or $\mathcal{O}(mn)$ computations to one, whereas existing methods perform equivalent computations at each iteration of the REML optimization procedure. For large samples, when taking genotype matrices as inputs, our interpreted-language implementations of `L_FOMC_REML` and `SLDF_REML` [9] produced more accurate variance component estimates than the highly-optimized, compiled BOLT-LMM implementations, while taking similar amounts of time. Thus, we expect comparably-optimized implementations of the novel algorithms to compute high accuracy REML estimates in close to the time required by BOLT-LMM v2.3.2 for a *single* objective function evaluation. Further, in contrast to the `BOLT_LMM` algorithm, which requires the genotype matrix, `SLDF_REML` can exploit precomputed GRMs to reduce operation count by an $\mathcal{O}(2m/n)$ factor (Table 1), which yields dramatic time savings when the number of markers greatly exceeds the number of individuals (Fig. 3). While GRM precomputation is itself $\mathcal{O}\left(mn^2\right)$, it can be effectively and asynchronously parallelized across multiple compute nodes, substantially mitigating computational burden (though we note that serial input/output constraints can interfere with efficient parallelization). However, as the `L_FOMC_REML` algorithm involves the computation of BLUPs of SNP effects, `L_FOMC_REML` is preferable to `SLDF_REML` when BLUP estimates are desired for prediction (as in [10]).

There are several limitations to the proposed approaches. First, `SLDF_REML`, which benefits from the ability to take GRMs as input, depends linearly on the number of included covariates, which might grow prohibitive in samples spanning numerous genotyping batches and ascertainment locations. However, as in `BOLT_LMM`, `L_FOMC_REML` requires $\mathcal{O}(mn)$ matrix multiplications for BLUP computation at each step of the REML optimization procedure, whereas `SLDF_REML` requires only vector operations. Thus, though the options provided by the two novel algorithms increase researchers' flexibility overall, the choice of whether to employ `SLDF_REML` versus `L_FOMC_REML` is problem-specific and necessitates greater researcher attention to resource allocation. For example, even when a precomputed GRM is available, it might be preferable to use `L_FOMC_REML` if BLUPs of latent SNP effects are desired. On the other hand, if a researcher intends to sequentially analyze a large number of phenotypes in a relatively small sample of individuals, it might prove most efficient to compute a GRM, despite the involved computational burden, in order to speed subsequent computations by supplying the GRM to the `SLDF_REML` algorithm. Second, neither algorithm mitigates the substantial $\mathcal{O}(mn)$ or $\mathcal{O}\left(n^2\right)$ memory complexity common to all algorithms for REML estimation of genomic variance components, requiring that researchers have access to high-memory compute nodes to work with large samples (though we note that neither of the novel algorithms substantial increases this burden either). Finally, for the same reasons that the spectral decomposition-based direct methods implemented in the FaST-LMM and GEMMA packages [3, 5, 6] are restricted to the simple two component model (1) (i.e., whereas the GRM and identity matrix are simultaneously diagonalizable, the same doesn't hold for arbitrary collections of three or more symmetric positive semidefinite matrices), the shift-invariance property exploited by the proposed methods does not extend to multiple genomic variance components. Given that the two component model is insufficient for precise heritability estimation for many complex traits [11], our novel algorithms apply to the particular, though common, tasks of variance component and BLUP estimation for LMM in association studies.

Despite these limitations, the proposed algorithms have clear advantages over existing methods in terms of flexibility, accuracy, and speed of computation. We provide both pseudocode and heavily annotated Python 3 implementations [9] to facilitate their incorporation into existing software packages. Further, though our algorithms are restricted to the two variance component model, they can be used to generate the inputs necessary for estimation of more complex models, such as the mixture model estimated via variational approximation implemented in [7], and thus have applications to non-infinitesimal models.

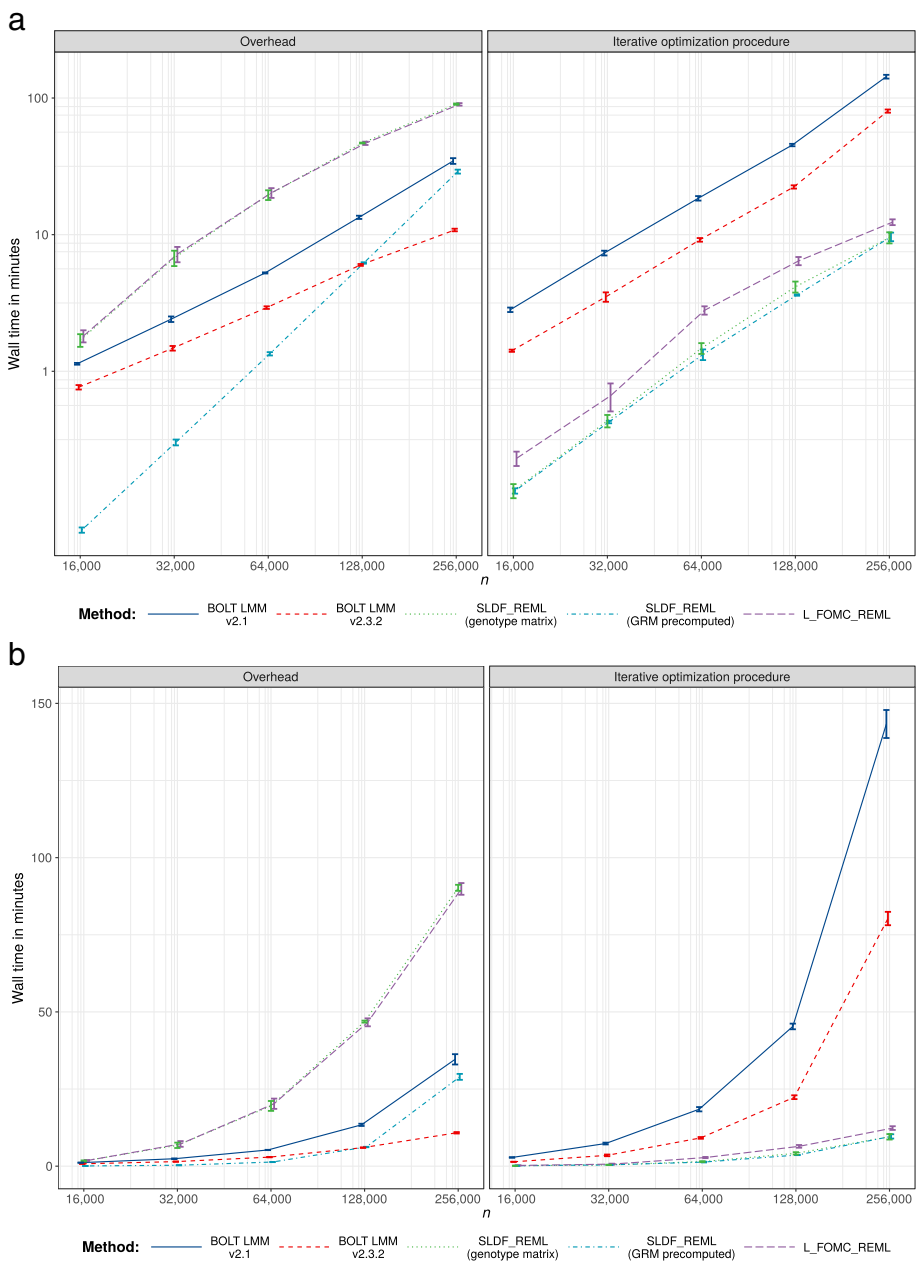

Fig. 2 Overhead versus iterative optimization procedure timing results. Trimmed mean wall clock time for overhead computations and iterative REML optimization procedures across twenty replications per condition on the $\log_{10}$ scale (**a**) and natural scale (**b**). Error bars reflect per condition standard errors and lines connect per condition means

Finally, we suggest that the methods presented in our theoretical development, in particular stochastic trace estimation and stochastic Lanczos quadrature, are likely to find uses in REML estimation of other models of interest to researchers in genomics. In particular, we suggest the development of models that exploit Krylov subspace shift-invariance to speed up variance/covariance component estimation for the case of multivariate phenotypes as a target for future research. Such models necessarily involve the computation or approximation of Hessian matrices, thereby introducing additional complexity in comparison to the univariate case considered above. However, the extension of fast cubic complexity methods based on the spectral decomposition of the covariance matrix [3, 5] to the multivariate case [6] suggests the potential for multivariate analogues of the algorithms presented here.

Table 1 Time complexity of stochastic algorithms

Method		Overhead	Objective function evaluation
`SLDF_REML`	with precomputed GRM	$\mathcal{O}\left(n^2 \cdot (n_{\text{rand}} + c) \cdot n_\kappa\right)$	$\mathcal{O}(n \cdot c \cdot n_\kappa)$
	with genotype matrix	$\mathcal{O}(2m \cdot n \cdot (n_{\text{rand}} + c) \cdot n_\kappa)$	$\mathcal{O}(n \cdot c \cdot n_\kappa)$
`L_FOMC_REML`		$\mathcal{O}(4m \cdot n \cdot n_{\text{rand}} \cdot n_\kappa)$	$\mathcal{O}(m \cdot n \cdot n_{\text{rand}})$
`BOLT_LMM`		$\mathcal{O}\left(n \cdot c^2 + m \cdot c\right)$	$\mathcal{O}(4m \cdot n \cdot n_{\text{rand}} \cdot n_\kappa)$

n denotes the number of individuals, m the number of markers, and c the number of covariates. n_{rand} indicates the number of random probing vectors and is fixed at 15 in all numerical experiments. n_κ reflects the number of conjugate gradient iterations required to achieve convergence at a specified tolerance and can be bounded in terms of the spectral condition number of H_0. As noted in [8], implicit preconditioning of H_0 can be achieved by including the first few right singular vectors of the genotype matrix (or eigenvectors of the GRM) as covariates

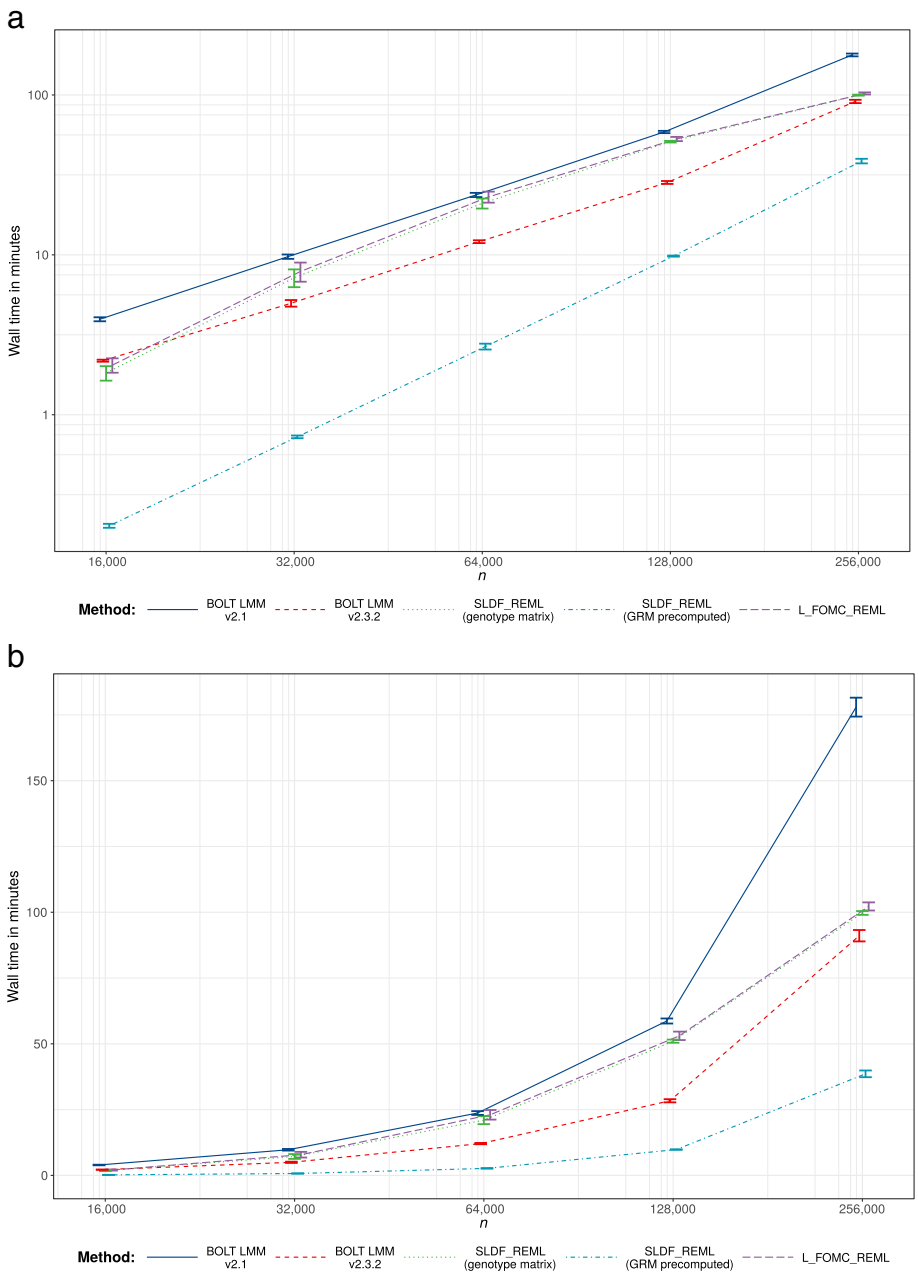

Fig. 3 Timing results. Trimmed mean wall clock time across twenty replications for per condition on the $\log_{10}$ scale (**a**) and natural scale (**b**). Error bars reflect per condition standard errors and lines connect per condition trimmed means

Conclusions

The proposed algorithms, `SLDF_REML` and `L_FOMC_REML`, unify previous approaches to estimating the two variance component model (1) by exploiting the simultaneous diagonalizability of the covariance structure components while avoiding matrix operations with cubic time complexity. As a result, the most expensive operations only need to be performed once, as with the spectral decomposition performed in the FaST-LMM and GEMMA software packages [3, 5, 6], but these operations consist only of matrix-vector products, as in the BOLT-LMM software package [7, 8]. All but one iteration of the REML optimization procedure requires only vector operations, yielding increased speed and numerical precision relative to existing methods. Furthermore, the unique strengths of the two methods lead to a flexible approach depending on researcher goals: `SLDF_REML` is capable of operating on precomputed GRMs when available, whereas `L_FOMC_REML` can generate BLUPs of latent SNP effects without added computational burden. We recommend these algorithms for incorporation into GWAS LMM implementations.

Method

We consider the two component genomic variance components model commonly employed in LMM association studies [1], which is of the form

$$\begin{aligned} y &= X\beta + \frac{1}{\sqrt{m}}Zu + e, \\ u &\overset{i.i.d.}{\sim} \mathcal{N}\left(0, \sigma_g^2\right), \quad e \overset{i.i.d.}{\sim} \mathcal{N}\left(0, \sigma_e^2\right), \end{aligned} \tag{1}$$

where y is a measured phenotype, the $c \ll n$ columns of $X \in \mathbb{R}^{n\times c}$ are covariates (including an intercept term) with corresponding fixed effects β, and $Z \in \mathbb{R}^{n\times m}$ is a matrix of n individuals' standardized genotypes at m loci. Without loss of generality, we assume that X has full column rank; in the case of numerical rank deficiency we can simply replace X by the optimal full rank approximation generated by its economy singular value decomposition or rank revealing QR decomposition. The latent genetic effects $u \in \mathbb{R}^m$ and residuals $e \in \mathbb{R}^n$ are random variables with distributions parametrized by the heritable and non-heritable variance components, σ_g^2 and σ_e^2, respectively. The REML criterion corresponds to the marginal likelihood of $\sigma_g^2, \sigma_e^2 | K^T y$, where K^T projects to an $(n-c)$-dimensional subspace orthogonal to the covariate vectors such that the null space of K^T is exactly the column space of X [12]. In other words $K^T : \mathbb{R}^n \to \mathcal{S} \subset \mathbb{R}^{n-c}$ such that $\mathbb{R}^n = \mathcal{S} \oplus \text{col } X$. The transformed random variable $K^T y$ has the marginal distribution $K^T y \sim \mathcal{MVN}\left(0, \sigma_g^2 \frac{1}{m} K^T Z Z^T K + \sigma_e^2 K K^T\right)$, which we reparametrize as $K^T y \sim \mathcal{MVN}\left(0, \sigma_g^2 K^T H_\tau K\right)$, where

$$H_\tau = \frac{1}{m} ZZ^T + \tau I_n, \quad \tau = \sigma_e^2 / \sigma_g^2. \tag{2}$$

Here, $\frac{1}{m} ZZ^T$, which indicates the average covariance between individuals' standardized genotypes, is often referred to as the *genomic relatedness matrix* (GRM). The *REML criterion*, or marginal log likelihood, can be expressed as a function of τ:

$$\begin{aligned} \ell\left(\tau | K^T y\right) \propto &- (n-c) \ln\left(\hat{\sigma}_g^2(\tau)\right) - \hat{\sigma}_e^2(\tau)^{-1} y^T P_\tau y \\ &- \ln\left(\det\left(K^T H_\tau K\right)\right), \end{aligned} \tag{3}$$

where $P_\tau = K\left(K^T H_\tau K\right)^{-1} K^T$, and, as implied by the REML first-order (stationarity) conditions, $\hat{\sigma}_e^2(\tau)$ is the expected residual variance component given τ and $\hat{\sigma}_g^2(\tau) = \hat{\sigma}_e^2(\tau)/\tau$ [12, 13]. In practice, K is never explicitly formed.

Naïve procedures for maximizing the REML criterion require evaluating (3) or its derivatives at each iteration of the optimization procedure. Previous methods either reduce the number of necessary cubic time complexity operations to one by exploiting problem structure, or substitute quadratic time complexity iterative and stochastic matrix operations for direct computations (Fig. 1). Here, we unify these approaches via the principle of Krylov subspace shift invariance to achieve methods that only require a single iteration of quadratic time complexity operations.

In what follows, we first present a brief survey of the Lanczos process, its applications to families of shifted linear systems, and its use in constructing Gaussian quadratures for spectral matrix functions. We assume familiarity with the method of conjugate gradients, an iterative procedure for approximating solutions to symmetric positive definite linear systems, and Gaussian quadrature, a method for approximating the integral of a given function by a well chosen weighted sum of its values; if not, see [14] and [15], respectively. We present these methods toward the goal of efficiently evaluating the quadratic form and log-determinant terms appearing in the REML criterion (3). We then present the details of the `SLDF_REML` and `L_FOMC_REML` algorithms, both of which exploit problem structure via Lanczos process-based methods in order to speed computation. Finally, we derive expressions for the computational complexity of the present algorithms, which we confirm via numerical experiment.

Preliminaries

The notation in this section is self-contained. Our presentation borrows from the literature extensively; further details on the (block) Lanczos procedure [14, 16], conjugate gradients for shifted linear systems [17, 18],

stochastic trace estimation [19, 20], and stochastic Lanczos quadrature [21–23] are suggested in the bibliography.

Krylov subspaces

Consider a symmetric positive-definite matrix A and nonzero vector b. Define the m^{th} Krylov subspace by the span of the first $m-1$ monomials in A applied to b; that is, $\mathcal{K}_m(A,b) = \text{span}\left\{A^k b : k = 0,\ldots,m-1\right\}$. Krylov subspaces are *shift invariant*—i.e., for real numbers σ, we have $\mathcal{K}_m(A,b) = \mathcal{K}_m(A+\sigma I, b)$.

The Lanczos procedure

The Lanczos procedure generates the decomposition $AU_m = U_m T_m$, where the columns $u_1,\ldots,u_m$ of U_m form an orthonormal basis for $\mathcal{K}_m(A,b)$ and the *Jacobi matrices* $T_m \in \mathbb{R}^{m\times m}$ are symmetric tridiagonal. Choosing $u_1 = b/\|b\|$, successive columns are uniquely determined by the sequence of Lanczos polynomials $\{p_k\}_{k=1}^{m-1}$ such that each $u_k = p_{k-1}(A)u_1$ and each p_k is the characteristic polynomial of Jacobi matrix T_k consisting of the first k rows and columns of T_m. The Lanczos procedure is equivalent to the well-known method of conjugate gradients (CG) for solving the linear system $Ax = b$ in that the m^{th} step CG approximate solution $x^{(m)}$ is obtained from the above decomposition using only vector operations (see Algorithm 1). The number of steps m prior to termination corresponds to the number of CG iterations need to bound the norm of the residual below a specified tolerance: $\left\|Ax^{(m)} - b\right\| < \epsilon$. The rate of convergence depends on the spectral properties of A and can be controlled in terms of the spectral condition number $\kappa(A)$. In the present application, the fact that all complex traits of interest generally have a non-trivial non-heritable component results in well-conditioned systems [7, 9].

Solving families of shifted linear systems

Having applied the Lanczos process to the *seed system* $Ax = b$, shift-invariance can be exploited to obtain the m^{th} step CG approximate solution $x_\sigma^{(m)}$ to the *shifted*

Algorithm 1: Lanczos conjugate gradients solver for shifted systems (`L_Solve`)

input : shift $\sigma \geq 0$, right hand sides $B = \{b_j \in \mathbb{R}^n\}_{j=1}^c$ and their Lanczos decompositions $U_j \in \mathbb{R}^{n\times m}$, $T_j \in \mathbb{R}^{m\times m}$ where col $U_j = \mathcal{K}_m(A, b_j)$.

output: Approximate solution $X_\sigma^{(m)} \approx (A+\sigma I)^{-1}B$.

```
begin
    for j = 1,...,c do                                          // iterate over RHSs
        // initialize coefficients:
        δ_{1:m} ← {(T_j)_{i,i}}_{i=1}^m + σ1⃗_m;                  // recycle Jacobi
        β_{2:m} ← {(T_j)_{i,i-1}}_{i=2}^m;                       //   coefficients
        ω_0 ← 0;
        γ_0 ← 1;
        ρ_1 ← ||b_j||;
        // initialize vectors:
        x_j ← 0⃗_n;                                              // CG approx solutions
        r_j ← b_j;                                              // CG residuals
        p_j ← b_j;                                              // search directions
        // main loop:
        for k = 1,...,m do
            // update coefficients
            γ_k ← (δ_k − ω_{k−1}/γ_{k−1})^{−1};
            ω_k ← (β_{k+1}γ_k)^2;
            ρ_{k+1} ← −β_{k+1}γ_k ρ_k;
            // update CG soln, residual, search dir
            x_j ← x_j + γ_k p_j;
            r_j ← ρ_{k+1}({U_j}_{k+1});                         // recycle basis
            p_j ← r_j + ω_k p_j;
        end
    end
    return X_σ^{(m)} = [x_1|···|x_c]
end
```

linear system $A_\sigma x_\sigma = (A + \sigma I)x_\sigma = b$, only using vector operations [17]. It can be shown that any positive shift by $\sigma \geq 0$ improves the rate of convergence such that $\left\|A_\sigma x_\sigma^{(m)} - b\right\| = \frac{\delta_m}{\delta_m+\sigma}\left\|Ax^{(m)} - b\right\|$, where $\delta_m > 0$ is the m^{th} diagonal element of the Lanczos Jacobi matrix corresponding to $\mathcal{K}_m(A, b)$.

Lanczos polynomials and Gaussian quadrature

Additionally, the Lanczos polynomials comprise a sequence of orthogonal polynomials with respect to the *spectral measure*

$$\mu_{A,v}(t) = \sum_{j=1}^{\ell:\lambda_\ell \leq t} \left(Q^T v\right)_j^2,$$

where $A = Q\Lambda Q^T$ is the spectral decomposition [21, 22]. Quadratic forms $v^T f(A)v$ involving *spectral functions* $f(A) = Qf(\Lambda)Q^T$, e.g., for the matrix logarithm, $v^T(\log A)v = \sum_{i=1}^n \left[\ln(\lambda_i)\left(Q^T v\right)_i^2\right]$, can be written as Riemann–Stieltjes integrals of the form

$$v^T Qf(\Lambda)Q^T v = \int_{\lambda_1}^{\lambda_n} f(t)d\mu_{A,v}(t). \tag{4}$$

The Lanczos decomposition $AU_m = U_m T_m$ generates the weights and nodes for an m-point Gaussian quadrature approximating the above integral. Denoting the spectral decomposition of the j^{th} Jacobi matrix $T_j = W_j D_j W_j^T$ for $j = 1, \ldots, m$, we approximate (4) as

$$\int_{\lambda_1}^{\lambda_n} f(t)d\mu_{A,v}(t) \approx \sum_{\ell=1}^{m} \omega_{j,\ell} f(\theta_{j,\ell}),$$

where $\theta_{j,\ell} = \{D_j\}_{\ell,\ell}$ and $\omega_{j,\ell} = \left\{e_1^T W_j\right\}_\ell$. As m here corresponds to the number of CG iterations needed to ensure that $\|Ax^{(m)} - v\|$ is smaller than a specified tolerance, the tridiagonal Jacobi matrices are small and calculating their spectral decompositions is computationally trivial.

Stochastic Lanczos quadrature

Stochastic Lanczos quadrature (SLQ) combines the above quadrature formulation with Hutchinson-type stochastic trace estimators [21]. Such estimators approximate the trace of a matrix $H \in \mathbb{R}^{n\times n}$ by a weighted sum of quadratic forms $\mathrm{tr}(H) \approx \frac{n}{n_{\text{rand}}} \sum_{k=1}^{n_{\text{rand}}} v_k^T H v_k$ for normalized, suitably distributed i.i.d. random *probing vectors* $\{v_j\}_{j=1}^{n_{\text{rand}}}$ [19]. The SLQ approximate trace of a spectral function of a matrix, $\mathrm{tr}(f(A))$, is then

$$\begin{aligned}
\mathrm{tr}(f(A)) &\approx \frac{n}{n_{\text{rand}}} \sum_{k=1}^{n_{\text{rand}}} v_k^T Qf(A)Q^T v_k \\
&= \frac{n}{n_{\text{rand}}} \sum_{k=1}^{n_{\text{rand}}} \int_{\lambda_1}^{\lambda_n} f(t)d\mu_{A,v_k}(t) \\
&\approx \frac{n}{n_{\text{rand}}} \sum_{k=1}^{n_{\text{rand}}} \sum_{\ell=1}^{m_\kappa} \omega_{k,\ell} f(\theta_{k,\ell}).
\end{aligned} \tag{5}$$

Whereas the number of probing vectors n_{rand} is chosen a priori, the number quadrature nodes m_κ corresponds to the number of conjugate gradient iterations needed to ensure $\left\|A_\sigma x_{j\sigma}^{(m_\kappa)} - v_j\right\|$ is less than a specified tolerance for each $j = 1, \ldots, n_{\text{rand}}$.

SLQ and shift invariance

For a fixed probing vector v_i, we can exploit the shift invariance of $\mathcal{K}_m(A, v_i)$ to efficiently update Gaussian quadrature generated by the corresponding Lanczos decomposition $AU_m = U_m T_m$. Again denoting the spectral decomposition of the Jacobi matrix $T_i = W_i D_i W_i^T$, the Lanczos decomposition of the shifted system is simply $A_\sigma U_m = U_m W_m(D_m + \sigma I_m)W_m^T$. Thus, given the approximation (5) for $\mathrm{tr}(f(A))$, we can efficiently compute an approximation of $\mathrm{tr}(f(A_\sigma))$ for any $\sigma > 0$. In Algorithm 2 we implement a method for estimating $\mathrm{tr}(\log(A_\sigma))$ in $\mathcal{O}(n_{\text{rand}})$ operations given the spectral decompositions of the Jacobi matrices corresponding to $\mathcal{K}_m(A, v_j)$ for probing vectors $\{v_j\}_{j=1}^{n_{\text{rand}}}$.

Block methods

For multiple right hand sides $B = [b_1|\cdots|b_c]$, the Lanczos procedure can be generalized to the *block Krylov subspace* $\mathcal{K}_m(A, B) = \bigotimes_{j=1}^c \mathcal{K}_m(A, b_j)$, resulting in a collection of Lanczos decompositions $AU_j = U_j T_j$ such that $\{U_j\}_1 = b_j/\|b_j\|$ for $j = 1, \ldots, c$. This process is equivalent to block CG methods in that the Jacobi matrices can again be used to generate an approximate solution $X^{(m)}$ to the matrix equation $AX^{(m)} = B$. We provide an implementation of the block Lanczos procedure in `L_Seed` [9], employing a conservative convergence criterion defined in terms of the magnitude of the $(1, 2)$ operator norm of the residual $\left\|AB - X^{(m)}\right\|_{1\to 2} = \max_j \left\|Ab_j - x_j^{(m)}\right\|_2$. Compared to performing c separate Lanczos procedures with respect to $\{\mathcal{K}_m(A, b_j)\}_{j=1}^c$, block Lanczos with respect to $\mathcal{K}_m(A, B)$, with $B = [b_1|\cdots|b_c]$, produces the same result (for a fixed number of steps). However, block Lanczos employs BLAS-3 operations and is thus more performant, especially when implemented on top of parallelized linear algebra subroutines.

A derivative-free REML algorithm

We propose the stochastic Lanczos derivative-free residual maximum likelihood algorithm (`SLDF_REML`;

Algorithm 3), a method for efficiently and repeatedly evaluating the REML criterion, which is then subject to a zeroth-order optimization scheme. To achieve this goal, we first identify the parameter space of interest with a family of shifted linear systems. We then develop a scheme for evaluating the quadratic form $y^T P_\tau y$ and log determinant $\ln\left(\det\left(K^T H_\tau K\right)\right)$ terms in the REML criterion (3) that use the previously discussed Lanczos methods to exploit this shifted structure. Specifically, after obtaining a collection of Lanczos decompositions, we can repeatedly solve the linear systems involved in the quadratic form term via Lanczos conjugate gradients and approximate the log determinant term via stochastic Lanczos quadrature.

The parameter space as shifted linear systems

Given a range of possible values of the *standardized genetic variance component*, or *heritability*,

$$h^2 = \sigma_g^2 / \left(\sigma_g^2 + \sigma_e^2\right), \quad h^2 \in \left[h^2_{\min}, h^2_{\max}\right], \tag{6}$$

we set $\tau_0 = (1 - h^2_{\max})/h^2_{\max}$ and define $H_0 = H_{\tau_0}$, noting that for all $\tau \in \Theta = \left\{\left(1 - h^2\right)/h^2 : h^2 \in \left[h^2_{\min}, h^2_{\max}\right]\right\}$, the spectral condition number of H_τ will be less than that of H_0 as the identity component of H_τ will only increase. Further, we have now identified elements of our parameter space $\tau \in \Theta$ with the family of shifted linear systems

$$\mathcal{H}_{\tau_0} = \{H_\sigma = H_\tau = H_0 + \sigma I_n : \sigma = \tau - \tau_0\}.$$

For any vector v for which we have computed the Lanczos decomposition $H_0 U = UT$ with the first column of U equal to $v/\|v\|$, we can use Algorithm 1 to obtain the CG approximate solution $x_\sigma \approx H_\sigma^{-1} v$ for all $\sigma \geq 0$ in $O(n)$ operations.

The quadratic form

Directly evaluating the quadratic form

$$y^T P_\tau y = y^T K \left(K^T H_\tau K\right)^{-1} K^T y \tag{7}$$

is computationally demanding and is typically avoided in direct estimation methods [12, 13]. Writing the complete QR decomposition of the covariate matrix $X = [\,Q_X | Q_{X^\perp}\,]\,R$ allows us to define $K^T = Q_{X^\perp}^T$, noting that substituting $Q_{X^\perp} Q_{X^\perp}^T$ for K^T preserves the value of (7). $Q_{X^\perp} Q_{X^\perp}^T$ is equivalent to the orthogonal projection operator $S : v \mapsto v - Q_X Q_X^T v$, which admits an efficient implicit construction and is computed in $O\left(nc^2\right)$ operations via the *economy* QR decomposition $X = Q_X R_X$. Then, reexpressing (7) as $y^T S (S H_\tau S)^\dagger S y$, we can use the Lanczos process to construct an orthonormal basis and corresponding Jacobi matrix for the Krylov subspace $\mathcal{K}(SH_0S, Sy)$. We can then obtain the CG approximation of $y^T S (S H_\sigma S)^{-1} S y$ using vector operations as, for any shift σ, we have $y^T S (S H_\sigma S)^\dagger S y = y^T S (S H_0 S + \sigma I_n)^{-1} S y$ (see Lemma 1 in Additional file 1 for proof).

The log determinant

We use an equivalent formulation [12, 24] of the term $\ln(\det(K^T H_\tau K))$, rewriting it as

$$\ln(\det(H_\tau)) + \ln\left(\det\left(X^T H_\tau^{-1} X\right)\right) - \ln\left(\det\left(X^T X\right)\right).$$

The $\det\left(X^T X\right)$ term is constant with respect to τ and can be disregarded. For $c \ll n$, $\det\left(X^T H_\tau^{-1} X\right)$ is computationally trivial via direct methods given $H_\tau^{-1} X$, which we can compute for all parameter values of interest in $O(n)$ operations having first applied the block Lanczos process with respect to $\mathcal{K}(H_0, X)$. Computing the block Lanczos decomposition corresponding to $\mathcal{K}(H_0, X)$, which is only

Algorithm 2: Stochastic Lanczos quadrature approximate log determinant of shifted systems (`SLQ_LDet`)

input : shift $\sigma \geq 0$, eigenvectors and eigenvalues $W_{V_j} \in \mathbb{R}^{m \times m}$, $D_{V_j} \in \mathbb{R}^m$ of Jacobi matrices corresponding to $\mathcal{K}(A, v_j)$ for each probing vector, $j = 1, \ldots, n_{\text{rand}}$

output: approximate log determinant soln $\approx \log(\det(A + \sigma I))$

```
begin
    soln = 0;
    for j = 1, ..., n_rand do
        for i = 1, ..., m do
            soln ← soln + (W_{V_j})^2_{i,1} ln((D_{V_j})_i + σ);
        end
    end
    return (n/n_rand)soln
end
```

Algorithm 3: Stochastic Lanczos derivative-free residual maximum likelihood (`SLDF-REML`)

input : standardized genotype matrix $Z \in \mathbb{R}^{n\times m}$ or genomic relatedness matrix $ZZ^T \in \mathbb{R}^{n\times n}$, phenotype vector $y \in \mathbb{R}^n$ covariate matrix $X \in \mathbb{R}^{n\times c}$ with $c \ll n$, range of values to consider for standardized genomic variance component $\Theta = [h^2_{\text{min}}, h^2_{\text{max}}]$, number of probing vectors for trace estimator n_{rand}, scalar optimization routine over search interval `optimize`$(f : \Theta \to \mathbb{R}, a, b)$

output: estimated variance components $\hat{\sigma}^2_g, \hat{\sigma}^2_e$

define : `qr`: economy QR decomposition, `Rademacher`: generates Rademacher random samples, `L_Seed`: block Lanczos procedure as implemented in [21], `eigh_tridiagonal`: spectral decomposition of Hermitian tridiagonal matrix

begin
- $\tau_0 \leftarrow (1 - h^2_{\text{max}})/h^2_{\text{max}}$; // minimum value of τ
- $\tau_{\text{max}} \leftarrow (1 - h^2_{\text{min}})/h^2_{\text{min}}$; // maximal value of τ
- $Q, R \leftarrow$ `qr`(X); // economy QR decomp. of X
- $H_0 : u \mapsto \frac{1}{m} ZZ^T u + \tau_0 u$; // LHS of seed system
- $S : u \mapsto u - QQ^T u$; // projection to (col $X)^\perp$
- **for** $j = 1, \ldots, n_{rand}$ **do** // draw random probes
  - $V_j \leftarrow$ `Rademacher`(n);
  - $V_j \leftarrow V_j / \|V_j\|$
- **end**
- `// Lanczosdecompositions of seed systems:`
- $U_y, T_y \leftarrow$ `L_Seed`(SH_0S, Sy); // proj. pheno.
- **for** $j = 1, \ldots, c$ **do**
  - $U_{X_j}, T_{X_j} \leftarrow$ `L_Seed`(H_0, X_j); // covariates
- **end**
- **for** $j = 1, \ldots, n_{rand}$ **do**
  - $U_{V_j}, T_{V_j} \leftarrow$ `L_Seed`(H_0, V_j); // probes
  - `// decompose Jacobi matrices for SLQ:`
  - $W_{V_j}, D_{V_j} =$ `eigh_tridiagonal` (T_{V_j})
- **end**
- `// construct REML criterion function:`
- **def** `REML_criterion` $(h^2 \leq h^2_{\text{max}})$**:**
  - **global** $\hat{\sigma}^2_g, \hat{\sigma}^2_e$;
  - $\tau = (1 - h^2)/h^2$;
  - $\sigma \leftarrow \tau - \tau_0$;
  - $\gamma \leftarrow (1 + \tau)^{-1}$;
  - ldet $\leftarrow X^T$(`L_Solve`$(\sigma, \{U_{X_j}, T_{X_j}\}_{j=1}^c)$);
  - ldet $\leftarrow$ldet $+$`SLQ_LDet`$(\sigma, \{W_{V_j}, D_{V_j}\}_{j=1}^{n_{rand}})$;
  - qform $\leftarrow y^T S($`L_Solve`$(\sigma, U_y, T_y))$;
  - $\hat{\sigma}^2_e \leftarrow$ qform$/(n - c)$;
  - $\hat{\sigma}^2_g \leftarrow \hat{\sigma}^2_e/\tau$;
  - **return** $(n - c)\ln(\hat{\sigma}^2_g) -$ ldet $-$ qform$/\hat{\sigma}^2_e$
- `// apply zeroth-order optimization routine:`
- `optimize`(`REML_criterion`,$h^2_{\text{min}}, h^2_{\text{max}}$);
- **return** $\hat{\sigma}^2_g, \hat{\sigma}^2_e$

end

performed once, unfortunately scales with the number of covariates c, a disadvantage not shared by our second algorithm (Algorithm 4). The remaining term, $\ln(\det(H_\tau))$, is approximated by applying SLQ (Algorithm 2) to a special case of (5): We rewrite the log determinant as the trace of the matrix logarithm

$$\begin{aligned}\ln(\det(H_\tau)) &= \text{tr}\,(\log(H_\tau)) \\ &= \text{tr}\, Q[\ln(\lambda_1 + \sigma)| \cdots |\ln(\lambda_n + \sigma)]\, Q^T,\end{aligned}$$

where we have spectrally decomposed $H_0 = Q\Lambda Q^T$ for some $\tau_0 \leq \tau$ with $\sigma = \tau - \tau_0$. We draw n_{rand} *i.i.d.* normalized Rademacher random vectors $v_1, \ldots, v_{n_{\text{rand}}}$, where

each element of each vector v_i takes values of either $1/\|v_i\|$ or $-1/\|v_i\|$ with equal probability. The SLQ approximate of the log determinant for the seed system is

$$\ln(\det(H_\sigma)) \approx \frac{n}{n_{\text{rand}}} \sum_{i=1}^{n_{\text{rand}}} \sum_{\ell=1}^{m_i} \omega_{i,\ell} \ln(\theta_{i,\ell} + \sigma),$$

where the weights $w_{i,\ell}$ and nodes $\theta_{i,\ell}$ are respectively derived by using the Lanczos process to construct orthonormal bases for $\mathcal{K}(H_0, v_i)$ (in practice, we apply block Lanczos to $\mathcal{K}(H_0, (v_1, \ldots, v_{n_{\text{rand}}}))$) [21, 22].

The SLDF_REML algorithm

Stochastic Lanczos derivative-free residual maximum likelihood (`SLDF_REML`; Algorithm 3), conceptually similar to the derivative-free algorithm of Graser and colleagues [13], applies the previously introduced Lanczos methods to approximate the above reparametrization of the REML criterion. Shift-invariance is then exploited such that, with the exception of the initial Lanczos decompositions, the REML log likelihood can be repeatedly evaluated using only vector operations. `SLDF_REML` takes a phenotype vector $y \in \mathbb{R}^n$, a covariate matrix $X \in \mathbb{R}^{n \times c}$, either the genetic relatedness matrix $ZZ^T \in \mathbb{R}^{n \times n}$ or the standardized genotype matrix $Z \in \mathbb{R}^{n \times m}$ (in which case the action of the GRM as a linear operator is coded implicitly as $v \mapsto Z\left(Z^T v\right)$), and a range of possible standardized genomic variance component values $\Theta = \left[h^2_{\min}, h^2_{\max}\right]$ as arguments and generates a function `REML_criterion`: $\Theta \to \mathbb{R}$ that efficiently computes the log-likelihood of $\tau | K^T y$. This function is then subject to scalar optimization via Brent's method, which is feasible given the low cost of evaluation and low dimension of Θ. Hyperparameters include the number of probing vectors to be used for the SLQ approximation of the log determinant n_{rand}, as well as tolerances corresponding to the REML criterion, parameter estimates, and the Lanczos residual norms. Convergence to a given tolerance on a sensible scale is ensured by optimizing with respect to the heritability $h^2 \in \Theta \subseteq [0, 1]$ and evaluating the REML criterion at $\tau = \left(1 - h^2\right)/h^2$. The REML criterion can be repeatedly evaluated in $\mathcal{O}(n)$ operations, making high accuracy computationally feasible.

A first-order Monte Carlo REML algorithm

We additionally propose the Lanczos first-order Monte Carlo residual maximum likelihood algorithm (`L_FOMC_REML`; Algorithm 4), which also takes advantage of the shifted structure of the standard genomic variance components model to speed computation. We first present the related first-order algorithm implemented in the efficient and widely-used BOLT-LMM software [7, 8], which we refer to as `BOLT_LMM` and of which the proposed `L_FOMC_REML` algorithm is a straightforward extension.

`BOLT_LMM` ***(First-order Monte Carlo REML)***

The `BOLT_LMM` algorithm is based on the observation that at stationary points of the REML criterion (3), the first-order REML conditions (i.e., $\nabla \ell = 0$) imply that

$$\mathbb{E}\left[\tilde{u}^T \tilde{u} | y\right] = \tilde{u}^T \tilde{u}, \quad \mathbb{E}\left[\tilde{e}^T \tilde{e} | y\right] = \tilde{e}^T \tilde{e}, \tag{8}$$

where $\tilde{u}$ and $\tilde{e}$ are the best linear unbiased predictions (BLUPs) of the latent genetic effects and residuals, respectively [25]. The BLUPs are functions of τ given by

$$\begin{aligned} \tilde{u}(\tau) &= m^{-1/2} Z^T S \acute{H}_\tau^{-1} S y, \\ \tilde{e}(\tau) &= \tau \acute{H}_\tau^{-1} S y, \end{aligned} \tag{9}$$

where we have defined $\acute{H}_\tau = \frac{1}{m} SZZ^T S + \tau I_n$. The expectations (8) are approximated via the following stochastic procedure: Monte Carlo samples of the latent variables, $\check{u}_k \overset{i.i.d}{\sim} \mathcal{MVN}(0, I_m)$, $\check{e}_k \overset{i.i.d}{\sim} \mathcal{MVN}(0, S)$ are used to generate samples of the projected phenotype vector

$$\check{y}_k = m^{-1/2} S Z \check{u}_k + \check{e}_k, \quad k = 1, \ldots n_{\text{rand}}.$$

BLUPs are then computed as in (9), yielding the approximations

$$\begin{aligned} \underset{\text{MC}}{\mathbb{E}}\left[\tilde{u}^T \tilde{u} | y\right] &= \frac{n_{\text{rand}}^{-1}}{\sqrt{m}} \sum_{k=1}^{n_{\text{rand}}} \left\| Z^T S \acute{H}_\tau^{-1} S \check{y}_k \right\|^2, \\ \underset{\text{MC}}{\mathbb{E}}\left[\tilde{e}^T \tilde{e} | y\right] &= n_{\text{rand}}^{-1} \sum_{k=1}^{n_{\text{rand}}} \left\| \tau \acute{H}_\tau^{-1} S \check{y}_k \right\|^2. \end{aligned}$$

Using the above expressions, Loh et al. [7, 8] apply a zeroth-order root-finding algorithm to the quantity

$$f_r(\tau) = \ln\left[\frac{\tilde{u}^T \tilde{u}}{\tilde{e}^T \tilde{e}}\right] - \ln\left[\frac{\mathbb{E}_{\text{MC}}\left[\tilde{u}^T \tilde{u} | y\right]}{\mathbb{E}_{\text{MC}}\left[\tilde{e}^T \tilde{e} | y\right]}\right],$$

noting that $f_r = 0$ is a necessary condition (and, in practice, a sufficient condition) for (8). Using CG to approximate solutions to the linear systems involved in BLUP computations results in an efficient REML estimation procedure involving $O(n \cdot m \cdot n_{\text{rand}})$ operations for well-conditioned covariance structures (i.e., for nontrivial non-heritable variance component values). As noted in [8], implicit preconditioning of H_0 can be achieved by including the first few right singular vectors of the genotype matrix (or eigenvectors of the GRM) as columns of the covariate matrix X.

The L_FOMC_REML algorithm

The `BOLT_LMM` algorithm described above involves solving $n_{\text{rand}} + 1$ linear systems

$$\acute{H}_{\tau_\ell}^{-1} S \check{y}, \; \acute{H}_{\tau_\ell}^{-1} S \check{y}_1, \; \ldots, \; \acute{H}_{\tau_\ell}^{-1} S \check{y}_{n_{\text{rand}}},$$

Algorithm 4: Lanczos first-order Monte Carlo residual maximum likelihood (`L_FOMC_REML`)

input : standardized genotype matrix $Z \in \mathbb{R}^{n \times m}$, phenotype vector $y \in \mathbb{R}^n$, covariate matrix $X \in \mathbb{R}^{n \times c}$ with $c \ll n$, range of values to consider for standardized genomic variance component $\Theta = [h^2_{\mathsf{min}}, h^2_{\mathsf{max}}]$, number of Monte Carlo samples n_{rand}, zeroth order scalar root finding routine $\texttt{root}(f : \Theta \to \mathbb{R}, h^2_{\mathrm{min}}, h^2_{\mathrm{max}})$

output: estimated value of heritable variance component $\hat{\sigma}^2_g$

define : `qr`: economy QR decomposition, `Gaussian`: generates standard normal random samples, `L_Seed`: block Lanczos procedure as implemented in [21]

begin

- $\tau_0 \leftarrow (1 - \sigma^2_{g\mathsf{max}})/\sigma^2_{g\ \mathsf{max}}$; `// minimum value of` τ
- $\tau_{\mathsf{max}} \leftarrow (1 - \sigma^2_{g\ \mathsf{min}})/\sigma^2_{g\ \mathsf{min}}$; `// maximal value of` τ
- $Q, R \leftarrow \texttt{qr}(X)$; `// economy QR decomp. of` X
- $S : u \mapsto u - QQ^T u$; `// projection to` $(\mathrm{col}\ X)^\perp$
- $\acute{H}_0 : u \mapsto \frac{1}{m} SZZ^T Su + \tau_0 u$; `// LHS of seed system`
- **for** $k = 1, \ldots, n_{\mathrm{rand}}$ **do** `// sample latent variables`
  - $\breve{u}_k \leftarrow \texttt{Gaussian}(m)$;
  - $\breve{e}_k \leftarrow \texttt{Gaussian}(n)$;
  - $\breve{\breve{e}}_k \leftarrow S\breve{e}_k$; `// latent residual`
  - $\breve{y}_k \leftarrow m^{-1/2} SZ\breve{u}_k$; `// latent genetic value`
- **end**
- `// Lanczos decompositions of seed systems:`
- $U_y, T_y \leftarrow \texttt{L\_Seed}(\acute{H}_0, Sy)$;
- **for** $k = 1, \ldots, n_{\mathrm{rand}}$ **do** `// can use block Lanczos`
  - $U_{\breve{e}_k}, T_{\breve{e}_k} \leftarrow \texttt{L\_Seed}(H_0, \breve{e}_k)$;
  - $U_{\breve{y}_k}, T_{\breve{y}_k} \leftarrow \texttt{L\_Seed}(H_0, \breve{y}_k)$;
- **end**
- `// construct objective for root finding:`
- **def** `f_reml` $(h^2 \leq h^2_{\mathrm{max}})$**:**
  - **global** $\hat{\sigma}^2_e$;
  - $\tau = (1 - h^2)/h^2$;
  - $\sigma \leftarrow \tau - \tau_0$;
  - `// compute BLUPs:`
  - $\mathsf{soln} \leftarrow \texttt{L\_Solve}(\sigma, U_y, T_y)$;
  - $\tilde{u} \leftarrow m^{-1/2} Z^T S(\mathsf{soln})$;
  - $\tilde{e} \leftarrow \sqrt{\tau}(\mathsf{soln})$;
  - **for** $k = 1, \ldots, n_{\mathrm{rand}}$ **do** `// MC samples`
    - $\mathsf{soln\_u}[k] \leftarrow \texttt{L\_Solve}(\sigma, U_{\breve{u}_k}, T_{\breve{u}_k})$;
    - $\mathsf{soln\_e}[k] \leftarrow \texttt{L\_Solve}(\sigma, U_{\breve{e}_k}, T_{\breve{e}_k})$;
    - $\breve{\tilde{u}}_k \leftarrow m^{-1/2} Z^T S(\mathsf{soln\_u}[k] + \sqrt{\tau}\mathsf{soln\_e}[k])$;
    - $\breve{\tilde{e}}_k \leftarrow \sqrt{\tau}(\mathsf{soln\_u}[k] + \sqrt{\tau}\mathsf{soln\_e}[k])$;
  - **end**
  - $\mathsf{E}[\tilde{u}^T\tilde{u}] \leftarrow n^{-1}_{\mathrm{rand}} \sum_{k=1}^{n_{\mathrm{rand}}} \|\breve{\tilde{u}}_k\|^2$;
  - $\mathsf{E}[\tilde{e}^T\tilde{e}] \leftarrow n^{-1}_{\mathrm{rand}} \sum_{k=1}^{n_{\mathrm{rand}}} \|\breve{\tilde{e}}_k\|^2$;
  - $\hat{\sigma}^2_e \leftarrow \tilde{e}^T\tilde{e}/(n - c)$;
  - **return** $\ln(\tilde{u}^T\tilde{u}/\tilde{e}^T\tilde{e}) - \ln(\tilde{u}^T\tilde{u}/\mathsf{E}[\tilde{e}^T\tilde{e}])$
- `// apply zeroth-order root finding routine:`
- $\hat{h}^2 \leftarrow \texttt{root}(\texttt{f\_reml}, h^2_{\mathrm{min}}, h^2_{\mathrm{max}})$;
- $\hat{\sigma}^2_g \leftarrow \hat{\sigma}^2_e(\hat{h}^2/(1 - \hat{h}^2))$;
- **return** $\hat{\sigma}^2_g, \hat{\sigma}^2_e$

end

at each iteration of the optimization scheme in order to compute BLUPs of the latent variables for the observed phenotype vector and each of the Monte Carlo samples. However, each iteration involves spectral shifts of the left hand side of the form

$$\acute{H}^{-1}_{\tau_{\ell+1}} = \left(\acute{H}_{\tau_\ell} + \sigma I_n\right)^{-1}, \quad \sigma = (\tau_{\ell+1} - \tau_\ell).$$

As in the `SLDF_REML` algorithm, the underlying block Krylov subspace is invariant to these shifts (i.e., $\mathcal{K}_m\left(\acute{H}_\tau, Y\right) = \mathcal{K}_m\left(\acute{H}_\tau + \sigma I, Y\right)$, where $Y = \left[y|\check{y}_1|\cdots|\check{y}_{n_{\text{rand}}}\right]$). Thus, having performed the Lanczos process for an initial parameter value τ_0, we can use `L_Solve` (Algorithm 1) to obtain the block CG approximate solution $X^{(m)}_\sigma \approx \acute{H}^{-1}_{\tau+\sigma} Y$ in $O(n \cdot n_{\text{rand}})$ operations. We are thus able to avoid solving linear systems in all subsequent iterations, though the relatively small number of matrix-vector products involved in computing BLUPs for the latent genetic effects at each step are unavoidable. The requirement of the genotype matrix for computing (9) prevents both `L_FOMC_REML` and `BOLT_LMM` from efficiently exploiting precomputed GRMs.

Comparison of methods

We compare theoretical and empirical properties of our proposed algorithms, `SLDF_REML` and `L_FOMC_REML`, to those of `BOLT_LMM`.

Computational complexity

In contrast to `BOLT_LMM`, the Lanczos-decomposition based algorithms we have proposed only need to perform the computationally demanding operations necessary to evaluate the REML criterion once. As such, we differentiate between *overhead* computations, which occur once and do not depend on the number of iterations needed to achieve convergence, and *per-iteration* computations, which are repeated until convergence of the optimization process (Table 1 and Fig. 2).

The overhead computations of `SLDF_REML` are dominated by the need to construct bases for the $n_{\text{rand}} + c + 1$ subspaces $\mathcal{K}(H_0, [\check{\nu}_1, \ldots, \check{\nu}_{n_{\text{rand}}}, x_1, \ldots, x_c, y])$, and are thus $\mathcal{O}\left(n^2(n_{\text{rand}} + c)n_\kappa\right)$ when a precomputed GRM is available and $\mathcal{O}(2m \cdot n(n_{\text{rand}} + c)n_\kappa)$ otherwise. Here, n_κ denotes the number of Lanczos iterations needed to achieve convergence at a pre-specified tolerance and increases with $h^2_{\max}$. Subsequent iterations are dominated by the cost of solving $c + 1$ shifted linear systems via `L_Solve` and are thus $\mathcal{O}(n \cdot c \cdot n_\kappa)$. The overhead computations in `L_FOMC_REML` are dominated by the Lanczos decompositions corresponding to the $2n_{\text{rand}} + 1$ seed systems, where the GRM is implicitly represented in terms of the standardized genotype matrix, and is thus $\mathcal{O}(4m \cdot n \cdot n_{\text{rand}} \cdot n_\kappa)$. Operations of equivalent complexity are needed at *every* iteration of `BOLT_LMM`.

Numerical experiments

We compared wall clock times for genomic variance component estimation for height in nested random subsets of 16,000, 32,000, 64,000, 128,000, and 256,000 unrelated ($\hat{\pi} < .05$) European ancestry individuals from the widely used UK Biobank data set [26]. All analyses included 24 covariates consisting of age, sex, and testing center and used hard-called genotypes from 330,723 array SNPs remaining after enforcing a 1% minor allele frequency cutoff. We compared `SLDF_REML`, with and without a precomputed GRM, to `L_FOMC_REML` which requires the genotype matrix. For the novel algorithms, absolute tolerances for the Lanczos iterations and the REML optimization procedure were set to 5e-5 and 1e-5, respectively. Additionally, we compared our interpreted Python 3.6 code to BOLT-LMM versions 2.1 and 2.3.3 (C++ code compiled against the Intel MKL and Boost libraries) [7, 8, 27, 28]. We ran each algorithm twenty times per condition, trimming away the two most extreme timings in each condition. Mirroring the default settings of the BOLT-LMM software packages, we set $n_{\text{rand}} = 15$ across both of our proposed methods.

Novel algorithms were implemented in the Python v3.6.5 computing environment [9], using NumPy v1.14.3 and SciPy v1.1.0 compiled against the Intel Math Kernel Library v2018.0.2 [28–30]. Optimization was performed using SciPy's implementation of Brent's method, with convergence determined via absolute tolerance of the standardized genomic variance component $\hat{h}^2$. Timing

Table 2 Overhead and per objective function evaluation timings of stochastic algorithms for n=256,000

Method		Overhead	Per evaluation	Evaluation count	Total
BOLT-LMM v2.1		34.63	35.83	4	177.95
BOLT-LMM v2.3.2		10.82	20.07	4	91.09
`L_FOMC_REML`		89.87	2.06	6	102.21
`SLDF_REML`	with genotype matrix	90.22	1.06	9	99.73
	with precomputed GRM	28.95	1.07	9	38.60

Data reflect trimmed mean wall clock time in minutes over 20 iterations per condition

results (Table 2 and Figs. 3 and 5) do not include time required to read genotypes into memory, or, when applicable, to compute GRMs, and reflect total running time on an Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz with 32 physical cores and 1 terabyte of RAM. Timing experiments excluded methods with cubic time complexity, including GCTA, FaST-LMM, and GEMMA. Accuracy was assessed by comparing heritability estimates generated by the stochastic algorithms to those estimated via the direct, deterministic average-information Newton–Raphson algorithm as implemented in the GCTA software package v1.92.0b2 [4] (Figs. 4 and 5).

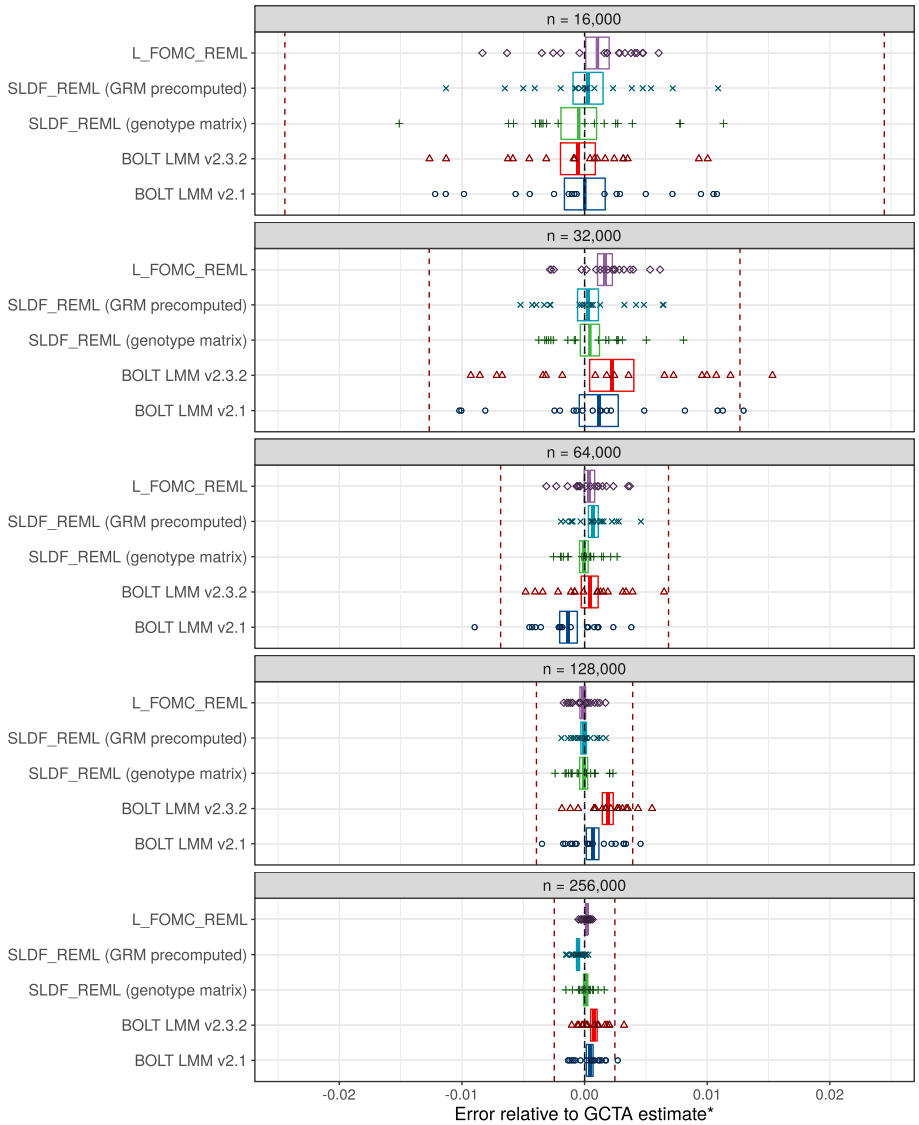

Fig. 4 Accuracy results. Comparison of heritability estimates for height generated by BOLT-LMM versions 2.1 and 2.3.2, `SLDF_REML`, and `L_FOMC_REML` versus those generated by the deterministic algorithm implemented in the GCTA software package* [4], for varying sub-samples of 16,000 to 256,000 unrelated European-ancestry UK Biobank participants. Data are comprised of twenty independent replications per condition. Red dashed lines indicate standard errors of GCTA estimate. Points represent individual observations whereas boxes indicate the 95% confidence intervals for the trimmed mean estimate after a Bonferroni correction for 25 comparisons. The bias evidenced by the BOLT-LMM estimators is likely due to the combination of performing a small number of secant iterations with fixed start values and loose tolerances for determining convergence. *For n=256,000, memory requirements prohibited the use of GCTA, so we instead averaged ten estimates generated by the high-accuracy stochastic estimator implemented in BOLT-REML [31] (standard errors were 6.32e-5 and 2.45e-7 for the mean REML heritability estimate and its standard error, respectively)

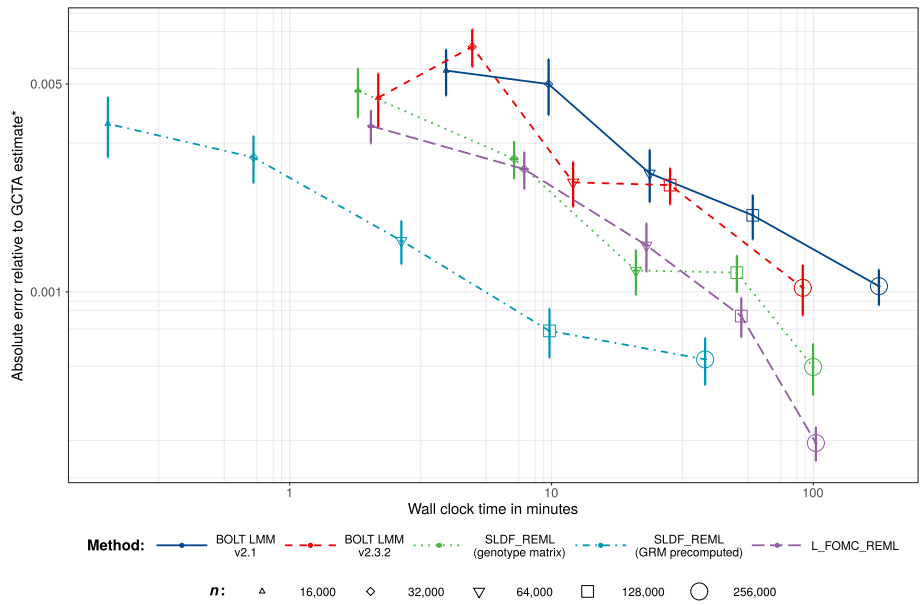

Fig. 5 Numerical experiments: accuracy versus time. Average absolute error on the $\log_{10}$ scale with respect to the GCTA estimate* versus trimmed mean wall clock time across twenty replications per condition. Error bars reflect per condition standard errors and lines connect per condition trimmed means. *For n=256,000, memory requirements prohibited the use of GCTA, so we instead averaged ten estimates generated by the high-accuracy stochastic estimator implemented in BOLT-REML v2.3.2 [31] (standard errors were 6.32e-5 and 2.45e-7 for the mean heritability and its standard error, respectively)

Additional file

Additional file 1: Proof of result used to efficiently compute the quadratic form (7). (PDF 125 kb)

Abbreviations

BLAS: Basic linear algebra subprogram; BLUP: Basic linear unbiased prediction; CG: Conjugate gradients method; GCTA: Genome-wide complex trait analysis [4]; GRM: Genomic relatedness matrix; GWAS: Genome-wide association study; LMM: Linear mixed-effects model; REML: Residual maximum likelihood; SLQ: Stochastic Lanczos quadrature

Acknowledgements

The authors wish to thank UK Biobank participants. Additionally, the authors thank Matthew C. Keller and Luke M. Evans for their thoughtful comments and provision of computational resources. Publication of this article was funded by the University of Colorado Boulder Libraries Open Access Fund.

Authors' contributions

RB wrote the manuscript, developed the algorithms, wrote the code used in numerical experiments, and analyzed the data. SB supervised the project and contributed to the development of the algorithms and the writing of the manuscript. Both authors read and approved the final manuscript.

Funding

Richard Border was supported by a training grant from the National Institute of Mental Health (T32 MH016880) and by the Institute for Behavioral Genetics. Stephen Becker acknowledges funding by NSF grant DMS-1819251.

Availability of data and materials

The UK Biobank data are available to qualified researchers via the UK Biobank Access Management System (https://bbams.ndph.ox.ac.uk/ams). The code used in the numerical experiments is available on Github (https://github.com/rborder/SL_REML).

Ethics approval and consent to participate

UK Biobank data collection procedures were approved by the UK Biobank Research Ethics Committee (reference 11/NW/0382).

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Institute for Behavioral Genetics, University of Colorado Boulder, 80309, Boulder, CO, USA. [2]Department of Psychology and Neuroscience, University of Colorado Boulder, 80309, Boulder, CO, USA. [3]Department of Applied Mathematics, University of Colorado Boulder, 80309, Boulder, CO, USA.

Received: 15 April 2019 Accepted: 30 June 2019
Published online: 30 July 2019

References

1. Yang J, Zaitlen NA, Goddard ME, Visscher PM, Price AL. Advantages and Pitfalls in the Application of Mixed Model Association Methods. Nat Genet. 2014;46(2):100–6.
2. Bates D, Mächler M, Bolker B, Walker S. Fitting Linear Mixed-Effects Models Using Lme4. 2014. arXiv preprint arXiv:14065823.
3. Zhou X, Stephens M. Genome-Wide Efficient Mixed Model Analysis for Association Studies. Nat Genet. 2012;44(7):821–4.
4. Yang J, Lee SH, Goddard ME, Visscher PM. GCTA: A Tool for Genome-Wide Complex Trait Analysis. Am J Hum Genet. 2011;88(1):76–82.
5. Lippert C, Listgarten J, Liu Y, Kadie CM, Davidson RI, Heckerman D. FaST Linear Mixed Models for Genome-Wide Association Studies. Nat Methods. 2011;8(10):833–5.
6. Zhou X, Stephens M. Efficient Multivariate Linear Mixed Model Algorithms for Genome-Wide Association Studies. Nat Methods. 2014;11(4):407.

7. Loh PR, Tucker G, Bulik-Sullivan BK, Vilhjálmsson BJ, Finucane HK, Salem RM, et al. Efficient Bayesian Mixed-Model Analysis Increases Association Power in Large Cohorts. Nat Genet. 2015;47(3):284–90.
8. Loh PR, Kichaev G, Gazal S, Schoech AP, Price AL. Mixed-Model Association for Biobank-Scale Datasets. Nat Genet. 2018;50:906–8.
9. Border R. Stochastic Lanczos Likelihood Estimation of Genomic Variance Components. Appl Math Grad Theses Dissertations. 2018;120.
10. de los Campos G, Vazquez AI, Fernando R, Klimentidis YC, Sorensen D. Prediction of Complex Human Traits Using the Genomic Best Linear Unbiased Predictor. PLoS Genet. 2013;9(7):e1003608.
11. Evans LM, Tahmasbi R, Vrieze SI, Abecasis GR, Das S, Gazal S, et al. Comparison of Methods That Use Whole Genome Data to Estimate the Heritability and Genetic Architecture of Complex Traits. Nat Genet. 2018;50(5):737–45.
12. Searle SR, Casella G, McCulloch CE. Variance Components, vol. 391. United States: Wiley; 2009.
13. Graser HU, Smith SP, Tier B. A Derivative-Free Approach for Estimating Variance Components in Animal Models by Restricted Maximum Likelihood. J Anim Sci. 1987;64(5):1362–70.
14. Björck A. Numerical Methods in Matrix Computations, vol. 59. Switzerland: Springer; 2015.
15. Atkinson KE. An Introduction to Numerical Analysis. United Kingdom: Wiley; 2008.
16. O'Leary DP. The Block Conjugate Gradient Algorithm and Related Methods. Linear Algebra Appl. 1980;29:293–322.
17. Frommer A, Maass P. Fast CG-Based Methods for Tikhonov-Phillips Regularization. SIAM J Sci Comput. 1999;20(5):1831–50.
18. Sogabe T. A Fast Numerical Method for Generalized Shifted Linear Systems with Complex Symmetric Matrices. Recent Dev Num Anal Num Comput Algoritm. 2010;13.
19. Hutchinson MF. A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines. Commun Stat Simul Comput. 1990;19(2):433–50.
20. Avron H, Toledo S. Randomized Algorithms for Estimating the Trace of an Implicit Symmetric Positive Semi-Definite Matrix. J ACM. 2011;58(2): 8:1–8:34.
21. Golub GH, Matrices MG. Moments and Quadrature with Applications. Princeton: Princeton University Press; 2009.
22. Ubaru S, Chen J, Saad Y. Fast Estimation of Tr(f(A)) via Stochastic Lanczos Quadrature. SIAM J Matrix Anal Appl. 2017;38(4):1075–99.
23. Chen J, Saad Y. A Posteriori Error Estimate for Computing Tr(f(A)) by Using the Lanczos Method. 2018. arXiv:180204928 [math].
24. Zhu S, Wathen AJ. Essential Formulae for Restricted Maximum Likelihood and Its Derivatives Associated with the Linear Mixed Models. 2018. arXiv:180505188 [stat].
25. McCulloch C, Searle SR, Neuhaus JM. Generalized, Linear, and Mixed Models. Hoboken: Wiley; 2008.
26. Sudlow C, Gallacher J, Allen N, Beral V, Burton P, Danesh J, et al. UK Biobank: An Open Access Resource for Identifying the Causes of a Wide Range of Complex Diseases of Middle and Old Age. PLoS Med. 2015;12(3): e1001779.
27. Schling B. The Boost C++ Libraries. USA: XML Press; 2011.
28. Wang E, Zhang Q, Shen B, Zhang G, Lu X, Wu Q, et al. Intel Math Kernel Library. In: High-Performance Computing on the Intel®Xeon Phi™. New York; 2014. p. 167–88.
29. Oliphant T. NumPy: A Guide to NumPy. 2006.
30. Jones E, Oliphant T, Peterson P, et al. SciPy: Open Source Scientific Tools for Python. 2001.
31. Loh PR, Bhatia G, Gusev A, Finucane HK, Bulik-Sullivan BK, Pollack SJ, et al. Contrasting Genetic Architectures of Schizophrenia and Other Complex Diseases Using Fast Variance Components Analysis. Nat Genet. 2015;47(12):1385–92.

Publisher's Note

Let A^+ denote any generalized inverse and let $A^\dagger$ denote the Moore-Penrose pseudoinverse.

Fact 1 (Seber [2008], page 457)**.** *Let $S \in \mathbb{R}^{n\times n}$ be an orthogonal projection operator. Then for any $B \in \mathbb{R}^{m\times n}$,*

$$S(BS)^\dagger = (BS)^\dagger$$

and

$$(SB)^\dagger S = (SB)^\dagger$$

Fact 2 (Lewis and Newman [1968], Theorem 6)**.** *Let $A \in \mathbb{R}^{n\times n}$ be such that $A = A^T \succeq 0$, $C \in \mathbb{R}^{n\times c}$, and define $\tilde{A} = A + C^T C$. Let $A^+ \succeq 0$ be a generalized inverse of A. Then*

$$\tilde{A}^+ = A^+ - A^+C^T(I + CA^+C^T)^{-1}CA^+$$

is a generalized inverse of $\tilde{A}$ if and only if $\ker A \subseteq \ker C$. Further, if this is the case, we have

$$\begin{gathered}\tilde{A}^+\tilde{A} = A^+A,\\ \tilde{A}\tilde{A}^+ = AA^+,\\ \tilde{A}^+ \succeq 0,\\ \text{and } \tilde{A}^+ = \tilde{A}^\dagger \iff A^+ = A^\dagger.\end{gathered}$$

Lemma 1. *Write the full QR decomposition of $X \in \mathbb{R}^{n\times c}$ as $X = QR = [Q_X|Q_{X^\perp}]R$ such that $Q_X \in \mathbb{R}^{n\times(n-c)}$, $Q_{X^\perp} \in \mathbb{R}^{n\times c}$ and define $S = I_n - Q_XQ_X^T$. Then,*

$$y^TS(S(H+\sigma I)S)^\dagger Sy = y^TS(SHS+\sigma I)^{-1}Sy$$

Proof. Applying the first fact,

$$y^TS(S(H+\sigma I)S)^\dagger Sy = y^T(S(H+\sigma I)S)^\dagger y.$$

Rewrite the left hand side as

$$\begin{aligned}y^T(S(H+\sigma I)S)^\dagger y &= \sigma^{-1}y^T(\underbrace{\sigma^{-1}SHS}_{\equiv A} + SS)^\dagger y\\ &= \sigma^{-1}y^T\tilde{A}^\dagger y,\end{aligned}$$

where we define $\tilde{A} = A + S^TS$. Note that the first fact also yields

$$SA^\dagger S = SA^\dagger = A^\dagger S = A^\dagger$$

Using the second fact, we then have

$$\begin{aligned}\sigma^{-1}y^T\tilde{A}^\dagger y &= \sigma^{-1}y^T(A^\dagger - A^\dagger S(I + SA^\dagger S)^{-1}SA^\dagger)y\\ &= \sigma^{-1}y^T(A^\dagger - A^\dagger(I + A^\dagger)^{-1}A^\dagger)y\\ &= \sigma^{-1}y^T(A + I)^{-1}y\\ &= y^TS(SHS+\sigma I)^{-1}Sy\end{aligned}$$

□

References

T. O. Lewis and T. G. Newman. Pseudoinverses of Positive Semidefinite Matrices. *SIAM Journal on Applied Mathematics*, 16(4):701–703, 1968. ISSN 0036-1399. URL `https://www.jstor.org/stable/2099121`.

George AF Seber. *A Matrix Handbook for Statisticians*, volume 15. John Wiley & Sons, 2008.