ONLINE METHODS

Linear mixed model and target optimization functions

We consider the following standard linear mixed model:

$$\mathbf{y} = \mathbf{W}\boldsymbol{\alpha} + \mathbf{x}\beta + \mathbf{Z}\mathbf{u} + \boldsymbol{\varepsilon} \quad \mathbf{u} \sim \mathrm{MVN}_m\left(0, \lambda\tau^{-1}\mathbf{K}\right) \quad \boldsymbol{\varepsilon} \sim \mathrm{MVN}_n\left(0, \tau^{-1}\mathbf{I}_n\right)$$

where n is the number of individuals and m is the number of groups/strains/clusters; $\mathbf{y}$ is a n by 1 vector of quantitative traits; $\mathbf{W}=(\mathbf{w}_1, \ldots, \mathbf{w}_c)$ is a n by c matrix of covariates (fixed effects) including a column vector of 1s; $\boldsymbol{\alpha}$ is a c by 1 vector of corresponding coefficients including the intercept; $\mathbf{x}$ is a n by 1 vector of marker genotypes; β is the effect size of the marker; $\mathbf{Z}$ is a n by m loading matrix and $\mathbf{u}$ is a m by 1 vector of random effects; $\boldsymbol{\varepsilon}$ is a n by 1 vector of errors; τ^{-1} is the variance of the residual errors; λ is the ratio between the two variance components; $\mathbf{K}$ is a known m by m relatedness matrix; $\mathbf{I}_n$ is a n by n identity matrix; and MVN denotes multivariate normal distribution.

In the case of the HMDP data set, m is the number of strains, n is the number of animals, and matrix $\mathbf{Z}$ indicates which strain each animal arises from ($z_{ij}=1$ if individual i comes from strain j, and 0 otherwise). In the case of the WTCCC data set, m=n and $\mathbf{Z}$ is an identity matrix. Multiple covariates such as cluster memberships or eigenvectors [5-7] can be incorporated into $\mathbf{W}$.

We are interested in obtaining both the maximum likelihood estimates (MLE) and the restricted/residual maximum likelihood (REML) estimates, and further exact test statistics. We used the term "exact" in the main text and here for brevity, although a more precise term would be "effectively exact". This is because computing the statistics involves an optimization problem that is not guaranteed to be convex, and so in general one cannot guarantee to found the global optimum. However, existing optimization methods appear to be highly effective in practice. The following description and derivation of GEMMA algorithm uses a few properties that appeared before [18].

The log-likelihood and log-restricted likelihood functions for the standard linear mixed model are

$$l(\lambda,\tau,\boldsymbol{\alpha},\beta)=\frac{n}{2}\log(\tau)-\frac{n}{2}\log(2\pi)-\frac{1}{2}\log|\mathbf{H}|-\frac{1}{2}\tau(\mathbf{y}-\mathbf{W}\boldsymbol{\alpha}-\mathbf{x}\beta)^{T}\mathbf{H}^{-1}(\mathbf{y}-\mathbf{W}\boldsymbol{\alpha}-\mathbf{x}\beta),\ (1)$$

and

$$\begin{aligned}l_r(\lambda,\tau)&=\frac{n-c-1}{2}\log(\tau)-\frac{n-c-1}{2}\log(2\pi)+\frac{1}{2}\log\left|(\mathbf{W},\mathbf{x})^{T}(\mathbf{W},\mathbf{x})\right|\\&-\frac{1}{2}\log|\mathbf{H}|-\frac{1}{2}\log\left|(\mathbf{W},\mathbf{x})^{T}\mathbf{H}^{-1}(\mathbf{W},\mathbf{x})\right|-\frac{1}{2}\tau\mathbf{y}^{T}\mathbf{P}_x\mathbf{y}\end{aligned},\qquad(2)$$

where

$$\mathbf{G}=\mathbf{ZKZ}^{T},\ \mathbf{H}=\lambda\mathbf{G}+\mathbf{I}_n,\ \mathbf{P}_x=\mathbf{H}^{-1}-\mathbf{H}^{-1}(\mathbf{W},\mathbf{x})\left((\mathbf{W},\mathbf{x})^{T}\mathbf{H}^{-1}(\mathbf{W},\mathbf{x})\right)^{-1}(\mathbf{W},\mathbf{x})^{T}\mathbf{H}^{-1}.$$

If λ is known, we can easily obtain $\hat{\boldsymbol{\alpha}}$, $\hat{\beta}$ and $\hat{\tau}$ for both log-likelihood and log-restricted likelihood functions (see Supplementary Note for details). Therefore, finding MLE and REML estimates is equivalent to optimizing the following target functions with respect to λ:

$$l(\lambda)=\frac{n}{2}\log\left(\frac{n}{2\pi}\right)-\frac{n}{2}-\frac{1}{2}\log|\mathbf{H}|-\frac{n}{2}\log\left(\mathbf{y}^{T}\mathbf{P}_x\mathbf{y}\right),\qquad(3)$$

$$\begin{aligned}l_r(\lambda)&=\frac{n-c-1}{2}\log\left(\frac{n-c-1}{2\pi}\right)-\frac{n-c-1}{2}+\frac{1}{2}\log\left|(\mathbf{W},\mathbf{x})^{T}(\mathbf{W},\mathbf{x})\right|\\&-\frac{1}{2}\log|\mathbf{H}|-\frac{1}{2}\log\left|(\mathbf{W},\mathbf{x})^{T}\mathbf{H}^{-1}(\mathbf{W},\mathbf{x})\right|-\frac{n-c-1}{2}\log\left(\mathbf{y}^{T}\mathbf{P}_x\mathbf{y}\right)\end{aligned}.\qquad(4)$$

Optimization method overview

A direct naive evaluation of the likelihood function or the restricted-likelihood function has a computational time that increases with the cube of the number of individuals, because it involves calculating a matrix determinant and a matrix inversion. A similarly expensive computation, involving a matrix inversion and a few matrix-vector multiplications, is used for each update step in the standard Henderson's iterative optimization procedure [19]. Therefore, Henderson's optimization algorithm is relatively slow. The algorithm EMMA [3] solves this problem by eigen-decompositions of matrix **G** and matrix $\mathbf{P}_x$ before optimization. After that, each target function involves only a summation of n scalar functions, thus making the derivation of the derivatives straightforward and their evaluations efficient. As a result, EMMA performs a single expensive calculation for each marker (decomposition of $\mathbf{P}_x$) followed by an iterative maximization scheme that involves only cheap operations (linear complexity in the number of individuals each iteration).

We take a different approach and obtain the first and second derivatives in vector/matrix forms, before eigen-decomposition of the relatedness matrix $\mathbf{G}$. Using three key recursions, we further show that both target functions and derivatives in vector/matrix forms, for each marker, despite their complicated appearance, are easy and efficient to evaluate during each optimization step. Therefore, we effectively replace the expensive eigen-decompostion of matrix $\mathbf{P}_x$ for each SNP with a cheap matrix-vector multiplication followed by a few recursions involving only scalar multiplications. Like EMMA, each iteration of our iterative maximization involves only cheap operations (linear complexity in number of individuals, quadratic complexity in the number of covariates c).

For numeric optimization, we start with Brent's method on the first derivative for stability, and follow with Newton-Raphson's method using the second derivative for efficiency. Details are given in Supplementary Note.

Note that the eigen-decomposition can be made faster when $m<n$ with a modification of the Gram-Schmidt process[3]. However, this trick is not expected to make much improvement for a genome-wide analysis and has not been implemented in the current version of the software.

Derivatives of target functions

We obtain the first and second derivatives for the log-likelihood function:

$$\frac{\partial l(\lambda)}{\partial \lambda} = -\frac{1}{2}\mathrm{trace}\left(\mathbf{H}^{-1}\mathbf{G}\right) + \frac{n}{2}\frac{\mathbf{y}^T\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{y}}{\mathbf{y}^T\mathbf{P}_x\mathbf{y}}, \tag{5}$$

$$\frac{\partial^2 l(\lambda)}{\partial \lambda^2} = \frac{1}{2}\text{trace}\left(\mathbf{H}^{-1}\mathbf{G}\mathbf{H}^{-1}\mathbf{G}\right) - \frac{n}{2}\frac{2\left(\mathbf{y}^T\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{y}\right)\left(\mathbf{y}^T\mathbf{P}_x\mathbf{y}\right) - \left(\mathbf{y}^T\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{y}\right)^2}{\left(\mathbf{y}^T\mathbf{P}_x\mathbf{y}\right)^2}, \tag{6}$$

and the first and second derivatives for the log-restricted likelihood function:

$$\frac{\partial l_r(\lambda)}{\partial \lambda} = -\frac{1}{2}\text{trace}\left(\mathbf{P}_x\mathbf{G}\right) + \frac{n-c-1}{2}\frac{\mathbf{y}^T\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{y}}{\mathbf{y}^T\mathbf{P}_x\mathbf{y}}, \tag{7}$$

$$\frac{\partial^2 l_r(\lambda)}{\partial \lambda^2} = \frac{1}{2}\text{trace}\left(\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{G}\right) - \frac{n-c-1}{2}\frac{2\left(\mathbf{y}^T\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{y}\right)\left(\mathbf{y}^T\mathbf{P}_x\mathbf{y}\right) - \left(\mathbf{y}^T\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{y}\right)^2}{\left(\mathbf{y}^T\mathbf{P}_x\mathbf{y}\right)^2}. \tag{8}$$

The above equations are obtained using a few matrix calculus properties listed in detail in Supplementary Note. Here, ⊗ denotes Kronecker product and vec denotes matrix vectorization (by stacking columns).

Several quantities require efficient evaluation

There are a few quantities we need to efficiently evaluate for each genetic marker in each optimization step. For the log-likelihood and log-restricted likelihood functions (3)-(4), we need to evaluate three quantities: $|\mathbf{H}|$, $\left|(\mathbf{W},\mathbf{x})^T\mathbf{H}^{-1}(\mathbf{W},\mathbf{x})\right|$ and $\mathbf{y}^T\mathbf{P}_x\mathbf{y}$. For the derivatives of the log-likelihood and log-restricted likelihood functions (5)-(8), we need to evaluate two types of quantities: trace terms (trace($\mathbf{H}^{-1}\mathbf{G}$), trace($\mathbf{H}^{-1}\mathbf{G}\mathbf{H}^{-1}\mathbf{G}$), trace($\mathbf{P}_x\mathbf{G}$), trace($\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{G}$)) and vector-matrix-vector product terms ($\mathbf{y}^T\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{y}$, $\mathbf{y}^T\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{G}\mathbf{P}_x\mathbf{y}$). We notice that the trace terms can be derived from trace($\mathbf{H}^{-1}$), trace($\mathbf{H}^{-1}\mathbf{H}^{-1}$), trace($\mathbf{P}_x$), trace($\mathbf{P}_x\mathbf{P}_x$), and the vector-matrix-vector product terms can be derived from $\mathbf{y}^T\mathbf{P}_x\mathbf{y}$, $\mathbf{y}^T\mathbf{P}_x\mathbf{P}_x\mathbf{y}$, $\mathbf{y}^T\mathbf{P}_x\mathbf{P}_x\mathbf{P}_x\mathbf{y}$ (details in Supplementary Note). Therefore, we need to efficiently evaluate three types of quantities for each SNP for any given λ:

1. Determinant terms $|\mathbf{H}|$ and $\left|(\mathbf{W},\mathbf{x})^T\mathbf{H}^{-1}(\mathbf{W},\mathbf{x})\right|$.

2. Trace terms trace($\mathbf{H}^{-1}$), trace($\mathbf{H}^{-1}\mathbf{H}^{-1}$), trace($\mathbf{P}_x$), trace($\mathbf{P}_x\mathbf{P}_x$).

3. Vector-matrix-vector product terms $\mathbf{y}^T\mathbf{P}_x\mathbf{y}$, $\mathbf{y}^T\mathbf{P}_x\mathbf{P}_x\mathbf{y}$ and $\mathbf{y}^T\mathbf{P}_x\mathbf{P}_x\mathbf{P}_x\mathbf{y}$**.**

We separate the above terms into basic quantities (which involve matrix **H**) and induced quantities (which involve matrix $\mathbf{P}_x$), and describe their evaluations in the next two sections, respectively.

Calculation of the basic quantities

Here, we describe the efficient calculations of three basic quantities: the determinant term $|\mathbf{H}|$, the trace terms trace($\mathbf{H}^{-1}$) and trace($\mathbf{H}^{-1}\mathbf{H}^{-1}$), and the vector-matrix-vector product terms in the forms of $\mathbf{a}^T\mathbf{H}^{-1}\mathbf{b}$, $\mathbf{a}^T\mathbf{H}^{-1}\mathbf{H}^{-1}\mathbf{b}$ and $\mathbf{a}^T\mathbf{H}^{-1}\mathbf{H}^{-1}\mathbf{H}^{-1}\mathbf{b}$, for **a** and **b** being equal to one of $\mathbf{w}_i$, **x** and **y**. We show in the next subsection that all the other terms can be derived from these basic quantities using recursions.

Before the genome-wide analysis, we first obtain an eigen-decomposition $\mathbf{G}=\mathbf{UDU}^T$ with time complexity $O(mn^2)$, where $\mathbf{D}=\mathrm{diag}(\delta_{1,} {}_{\dots,}\ \delta_n)$ and δ_is are the eigen values. Since $\mathbf{I}_n=\mathbf{UU}^T$, we have $\mathbf{H}=\mathbf{U}\mathrm{diag}(\lambda\delta_1+1_{,\dots,}\lambda\delta_n+1)\mathbf{U}^T$. Therefore, during each optimization step, the determinant term can be calculated with time complexity $O(n)$: $|\mathbf{H}|=\prod_{i=1}^{n}(\lambda\delta_i+1)$.

Similarly the trace terms can be evaluated with time $O(n)$:

$$\mathrm{trace}(\mathbf{H}^{-1})=\sum_{i=1}^{n}(\lambda\delta_i+1)^{-1}\ ,\ \mathrm{trace}(\mathbf{H}^{-1}\mathbf{H}^{-1})=\sum_{i=1}^{n}(\lambda\delta_i+1)^{-2}\ .$$

Next, we define and calculate $(\boldsymbol{v}_{w1},\ \dots,\ \boldsymbol{v}_{wc})=\mathbf{U}^T\mathbf{W}$, $\mathbf{v}_y=\mathbf{U}^T\mathbf{y}$ and $\mathbf{v}_x=\mathbf{U}^T\mathbf{x}$, each with time complexity $O(n^2)$ and only $\mathbf{v}_{\mathrm{x}}$ needs to be calculated for each SNP. Now, for any **a** and **b** being equal to one of $\mathbf{w}_i$, **x** and **y**, during each optimization step with time complexity $O(n)$, we obtain:

$$\mathbf{a}^T\mathbf{H}^{-1}\mathbf{b}=\sum_{i=1}^{n}v_{ai}v_{bi}(\lambda\delta_i+1)^{-1}\ ,$$

$$\mathbf{a}^T\mathbf{H}^{-1}\mathbf{H}^{-1}\mathbf{b}=\sum_{i=1}^{n}v_{ai}v_{bi}(\lambda\delta_i+1)^{-2}\ ,$$

$$\mathbf{a}^T\mathbf{H}^{-1}\mathbf{H}^{-1}\mathbf{H}^{-1}\mathbf{b}=\sum_{i=1}^{n}v_{ai}v_{bi}(\lambda\delta_i+1)^{-3}\ ,$$

where v_{ai} and v_{bi} are the corresponding ith elements in the vectors $\mathbf{U}^T\mathbf{a}$ and $\mathbf{U}^T\mathbf{b}$, respectively.

Recursions for the induced quantities

Here, we describe three recursions to efficiently evaluate the induced quantities from the basic quantities. The induced quantities are: the determinant term $\left|(\mathbf{W},\mathbf{x})^T\mathbf{H}^{-1}(\mathbf{W},\mathbf{x})\right|$, the trace terms trace($\mathbf{P}_x$) and trace($\mathbf{P}_x\mathbf{P}_x$), and the vector-matrix-vector product terms $\mathbf{y}^T\mathbf{P}_x\mathbf{y}$, $\mathbf{y}^T\mathbf{P}_x\mathbf{P}_x\mathbf{y}$ and $\mathbf{y}^T\mathbf{P}_x\mathbf{P}_x\mathbf{P}_x\mathbf{y}$.

First, we define $\mathbf{P}_0=\mathbf{H}^{-1}$, $\mathbf{P}_{c+1}=\mathbf{P}_x$, $\mathbf{w}_{c+1}=\mathbf{x}$ and $\mathbf{W}_i=(\mathbf{w}_1,\ldots,\mathbf{w}_i)$, $\mathbf{P}_i=\mathbf{H}^{-1}-\mathbf{H}^{-1}\mathbf{W}_i\left(\mathbf{W}_i^T\mathbf{H}^{-1}\mathbf{W}_i\right)^{-1}\mathbf{W}_i^T\mathbf{H}^{-1}$ for $i\in\{1,\ldots,c+1\}$. With Leibniz formula, we obtain a recursion for the determinant term: $\left|\mathbf{W}_i^T\mathbf{H}^{-1}\mathbf{W}_i\right|=\left|\mathbf{W}_{i-1}^T\mathbf{H}^{-1}\mathbf{W}_{i-1}\right|\left(\mathbf{w}_i^T\mathbf{P}_{i-1}\mathbf{w}_i\right)$.

Next, with blockwise matrix inversion, we obtain $\mathbf{P}_i=\mathbf{P}_{i-1}-\mathbf{P}_{i-1}\mathbf{w}_i\left(\mathbf{w}_i^T\mathbf{P}_{i-1}\mathbf{w}_i\right)^{-1}\mathbf{w}_i^T\mathbf{P}_{i-1}$. This leads to a recursion for the trace terms trace($\mathbf{P}_i$) and trace($\mathbf{P}_i\mathbf{P}_i$), and another recursion for the vector-matrix-vector product terms $\mathbf{a}^T\mathbf{P}_i\mathbf{b}$, $\mathbf{a}^T\mathbf{P}_i\mathbf{P}_i\mathbf{b}$ and $\mathbf{a}^T\mathbf{P}_i\mathbf{P}_i\mathbf{P}_i\mathbf{b}$, for any vectors **a**, **b** of the right size (details in Supplementary Note).

All the above recursions only involve scalar multiplications and calculations do not depend on the number of individuals. Therefore, the overall time complexity for GEMMA is $O(mn^2)$ (eigen-decompostion of **G**) + $O(cn^2)$ (evaluations of $\mathbf{v}_{wi}$ and $\mathbf{v}_y$)+ $O(pn^2)$ (evaluation of $\mathbf{v}_x$ for each SNP)+$O(ptc^2n)$ (evaluations of the basic quantities for each SNP during each optimization iteration)=$O(mn^2+cn^2+pn^2+ptc^2n)$.