



# Efficient ReML inference in variance component mixed models using a Min-Max algorithm

Fabien Laporte, Alain Charcosset, Tristan Mary-Huard

## ► To cite this version:

Fabien Laporte, Alain Charcosset, Tristan Mary-Huard. Efficient ReML inference in variance component mixed models using a Min-Max algorithm. PLoS Computational Biology, 2022, 18 (1), pp.e1009659. 10.1371/journal.pcbi.1009659 . hal-04068810

**HAL Id: hal-04068810**

**<https://hal.inrae.fr/hal-04068810>**

Submitted on 14 Apr 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

## RESEARCH ARTICLE

# Efficient ReML inference in variance component mixed models using a Min-Max algorithm

Fabien Laporte<sup>1</sup>, Alain Charcosset<sup>1</sup>, Tristan Mary-Huard<sup>1,2\*</sup>

**1** Université Paris-Saclay, INRAE, CNRS, AgroParisTech, Génétique Quantitative et Evolution - Le Moulon, Gif-sur-Yvette, France, **2** INRAE, AgroParisTech, Université Paris-Saclay, MIA-Paris, Paris, France

\* [tristan.mary-huard@agroparistech.fr](mailto:tristan.mary-huard@agroparistech.fr)



## OPEN ACCESS

**Citation:** Laporte F, Charcosset A, Mary-Huard T (2022) Efficient ReML inference in variance component mixed models using a Min-Max algorithm. PLoS Comput Biol 18(1): e1009659. <https://doi.org/10.1371/journal.pcbi.1009659>

**Editor:** Mona Singh, Princeton University, UNITED STATES

**Received:** June 4, 2020

**Accepted:** November 18, 2021

**Published:** January 24, 2022

**Copyright:** © 2022 Laporte et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** Datasets are available as downloadable figures, tables, and supplementary files in the following articles: (NAM: <https://doi.org/10.1534/genetics.117.300305>) (Factorial: <https://doi.org/10.1534/genetics.114.165860>) (Flint: <https://doi.org/10.1007/s00122-014-2379-7>). Data and Code are available as zip file in [S1](#), [S2](#) and [S3](#) Codes.

**Funding:** This work was funded by the French National Research Agency (ANR) as part of the "Investissements d'Avenir" program, ANR-10-BTBR-0001, Amaizing (<https://anr.fr>) to FL, AC, and

## Abstract

Since their introduction in the 50's, variance component mixed models have been widely used in many application fields. In this context, ReML estimation is by far the most popular procedure to infer the variance components of the model. Although many implementations of the ReML procedure are readily available, there is still need for computational improvements due to the ever-increasing size of the datasets to be handled, and to the complexity of the models to be adjusted. In this paper, we present a Min-Max (MM) algorithm for ReML inference and combine it with several speed-up procedures. The ReML MM algorithm we present is compared to 5 state-of-the-art publicly available algorithms used in statistical genetics. The computational performance of the different algorithms are evaluated on several datasets representing different plant breeding experimental designs. The MM algorithm ranks among the top 2 methods in almost all settings and is more versatile than many of its competitors. The MM algorithm is a promising alternative to the classical AI-ReML algorithm in the context of variance component mixed models. It is available in the `MM4LMM R`-package.

## Author summary

Mixed models have been a cornerstone of the quantitative genetics methodology for decades. Due to the growing size of datasets, their associated computational cost is a major burden, particularly in genome-wide association studies that routinely address Millions of markers, requiring as many mixed models to be fitted. In the particular case of a 2-variance component mixed models efficient procedures such as `FaST-LMM` or `GEMMA` have been developed to analyze panels with tens of thousands of individuals. However, there is room for improvement in cases where the computational burden is due to the number of variance components in the model rather than the panel size, a classical situation in plant genetics where several variance components are required to handle various polygenic effects. We consider a "MM" (Min-Max) algorithm as an alternative to the by-default AI (Average Information) algorithm used to perform inference in mixed models. The MM algorithm can be combined to the classical tricks used to accelerate the inference

TMH. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

process (e.g. simultaneous orthogonalization or squared iterative acceleration methods). The MM procedure is implemented in an `MM4LMM` package, and is competitive compared to classical algorithms used in plant genetics. This package should help geneticists handling more complex models to analyze their data than today.

This is a *PLOS Computational Biology* Methods paper.

## Introduction

Since their formal introduction in the early 50's [1, 2], mixed models have become an indispensable tool of modern statistics. They have been successfully used in many application fields [3] to model data with multiple sources of biological or technical variations. Starting with the work of [4] and [5], mixed models have been a favored methodology in quantitative genetics, and are still widely used in the context of Genome-Wide Association Studies (GWAS) and Genomic Selection (GS).

With the development of high throughput technologies, a special care has been dedicated to the development of efficient algorithmic procedures for the inference of mixed models [6–9]. This is illustrated by the availability of many tools/software that either perform inference in a mixed model including many (fixed and random) effects on large datasets, or alternatively that efficiently fit hundreds of thousands of mixed models with a limited number of variance components. Many of these tools were developed in the context of animal or human quantitative genetics, or single environment plant studies, where a popular strategy for GWAS analysis is the one presented in [10] that requires the fitting of a simple 2-component mixed model per marker, a favorable case for which efficient algorithms exist that allow the analysis of very large panels with  $10^4 - 10^5$  individuals genotyped at  $10^6$  markers [7, 8].

In the context of plant breeding, panels are often of moderate size, including a few hundreds/thousands of individuals only due to experimental/cost constraints. At the same time, the mixed models used for the statistical analysis may be more complex than the ones used in human genetics, since the modeling should account for the specificities of the experimental design (e.g. when a multi-site experiment involving the same varieties in different environments has been carried out) or of the crossing design (e.g. when considering hybrids obtained from parental lines belonging to different populations). One then aims at choosing among the many algorithmic procedures the one that is best suited to cope with these specific features.

From a technical point of view, the Average Information (AI) algorithm [11, 12] has become the reference algorithm for Restricted Maximum Likelihood (ReML) estimation. It has been implemented in many packages [9, 13, 14] and in different contexts, the practical implementation being sometimes slightly different from a package to another (see the [Methods](#) section). Although in some very specific cases alternative algorithms may be more computationally efficient, it is considered as a reference due to its versatility and its competitive computational performance.

In this article we present a Min-Max (MM) algorithm for the ReML inference in Gaussian Variance Component (VC) mixed model. MM algorithms have been previously described [15], and applied to ML inference in mixed models [16]. We first present here the full derivation of the MM procedure applied to ReML estimation, along with the way it can be combined

with some classical computational tricks to significantly speed-up the initial procedure. We then provide a thorough benchmark comparison to illustrate that the proposed MM algorithm can compete with several state-of-the-art algorithms that are widely used in quantitative genetics to perform ReML estimation, including BOLT-LMM [17], FaST-LMM [7], *gaston* [9], GEMMA [18] and GridLMM [19]. In our study algorithms are compared on different plant breeding scenarios, with a focus for those that may require numerous runs of the model (e.g. GWAS). The study reveals that differences in terms of precision may be marginal between exact ReML procedures, but differences in computational performance may be important. In many scenarios the MM algorithm proves to be competitive with respect to its competitors.

## Results

### Setting

We considered several procedures corresponding to different implementations of the optimization algorithms described in the Methods section. Our candidate procedure, called hereafter MM4LMM, corresponds to the implementation of the Min-Max algorithm. MM4LMM is compared to the following 3 state-of-the-art procedures:

- ★ *gaston* [9] (version 1.5.7) is an R package that implements the AI algorithm. When  $K = 2$  *gaston* performs ReML estimation using Newton method.
- ★ FaST-LMM [7] (C version 2014) fits mixed models with two variance components only, using the joint orthogonalization and profiling tricks.
- ★ GEMMA [18] (version 0.98.1) fits mixed models using the AI algorithm for variance component estimation, combined with the joint orthogonalization trick when applied to a two variance component model.

All aforementioned algorithms aim at optimizing the restricted maximum likelihood (6). Additionally, we considered Grid-LMM [19] that performs an approximate ReML optimization over a grid of candidate variance values, and BOLT-LMM that performs GWAS using a Gaussian mixture modeling of SNP effects (see [17] for details).

In order to compare the different algorithms in terms of precision of variance parameter estimation, one needs a reference procedure that provides the “true” values of the ReML estimates. In the present context we considered the `optimizeLmer` function of the *lme4* [20] R package which performs inference using the BOBYQA algorithm [21]. This algorithm was used as the reference procedure, *lme4* being one of the most popular (and consequently one of the most debugged) R package to fit mixed models.

We focus here on the application of mixed models to statistical plant genetics, with two specific application cases in mind: genome-wide association study (GWAS) and variance component analysis (VCA). In GWAS, the goal is to identify markers that are significantly associated with phenotypic variation. The relationship between a given marker and the phenotype needs to be tested in a model that accounts for both the marker effect and background genetic effects specific to each individual.

In classical human and animal studies and also simple plant experiments (one trait in one environment, single population), it is classical to adjust LMM with two variance components (genetic background and residual). Therefore, many softwares exist to fit LMM with only two variance components. However, a number of plant and animal genetic studies address phenotypic data of hybrids (crosses in animals) between different populations, referred to in plants as heterotic groups (e.g. flint and dent for maize genetics). Background genetic effects involve

in this case more than one source of variation, at least four when considering the additive variation contributed by each group, the interaction between the two groups and the error term.

A further characteristic of plant experiments is that phenotyping of varieties is generally conducted in multiple environments. These different environments generally reveal different magnitudes of genetic and error variations and complex genotype  $\times$  environment patterns. In this case, algorithms have to be flexible regarding the high number of possible random effects considered within the LMM.

Most GWAS algorithms fit mixed model including two or more variance components for each tested marker, meaning that computational efficiency is of major importance. Alternatively, some algorithms build on an approximate procedure where only the parameters associated to the fixed part of the model are updated for each marker (e.g. EMMAX or the approximate version of FaST-LMM, see [7, 22]), resulting in a significant computational speed-up. However, one should be sure that such speed-up approximations in the optimization algorithm yield accurate  $p$ -values.

### Algorithm comparison

The three datasets presented in the next sections were selected as representative of classical GWAS/GS/VCA panels in plant quantitative genetics, both in terms of scale (numbers of trials, individuals per trials and replicates per individuals) and complexity (lines or hybrids, need for genotype  $\times$  environment interaction in the statistical modeling. . .). All datasets are publicly available, see the Data Availability section. Also we stress out that the models considered for each dataset were selected in order to illustrate the many different aspects that may impact the computational performance of the algorithms to be compared rather than providing general guidelines about statistical modeling in quantitative genetics.

All algorithms were run and compared on a server using a Intel(R) Xeon(R) CPU E5-2420 0 @ 1.90GHz processor, applying their by-default configuration settings. Each dataset was analyzed using the same number of cores and memory setting for all algorithms.

**Two variance components. Dataset.** We consider the CornFed Flint Association panel—named Flint hereafter [23]. It consists in 259 maize lines of the Flint heterotic group, genotyped at 39,076 biallelic markers (after quality control). These hybrids were evaluated in an 11 location network. In the following GWAS we considered least square means of the individual phenotypes as the studied trait as in the original publication. We present here the results for two phenotypes, DM\_Y\_Flo and Tass, the results being similar for the remaining ones.

**Statistical analysis.** For each marker  $\ell$  and for a given trait, the analysis was performed using the following model:

$$Y = \mu + X_{\ell}\beta_{\ell} + U + E, \\ U \sim \mathcal{N}(0, \sigma_G^2 K), \quad E \sim \mathcal{N}(0, \sigma_E^2 I) \quad \text{and} \quad U \perp E,$$

with  $Y$  the vector of phenotypes,  $X_{\ell}$  the vector corresponding to the number of copies of allele 1 present at marker  $\ell$ ,  $\beta_{\ell}$  the effect associated with allele 1,  $U$  the random effect accounting for the genetic background, and  $E$  the error vector.  $K$  is the matrix of kinship between lines. To identify markers associated to phenotypic variation the null hypothesis  $H_0: \{\beta_{\ell} = 0\}$  was tested using a Wald test procedure (see S1 Appendix for details).

**Computational time.** The computational times corresponding to the whole genome analysis are displayed in Table 1. Recall that 1me4 represents the reference in terms of inference performance (i.e. sharp estimation of parameters), but is not expected to perform well in terms of computational time, being not optimized for GWAS analysis. Although all methods achieve the GWAS analysis in less than 30s, a factor of at least 10 may be observed between the most

**Table 1. Computational time (in sec.) associated to the different algorithms for the complete GWAS analysis of the Flint dataset, trait by trait.**

	<b>gaston</b>	<b>MM4LMM</b>	<b>FaST-LMM</b>	<b>GEMMA</b>	<b>BOLT-LMM</b>	<b>GridLMM</b>	<b>lme4</b>
DM_Y	3	6	28	15	12	9	12886
Tass	5	17	28	15	325	5	34852

<https://doi.org/10.1371/journal.pcbi.1009659.t001>

efficient (*gaston*) and the less efficient (*FaST-LMM* or *BOLT-LMM*) methods (excluding *lme4*). There is no gain in using *GridLMM* in this context since all exact methods are using the simultaneous orthogonalization trick, making the inference very efficient. One also observes that the computational time of *BOLT-LMM* is quite unstable from one trait to another.

**QTL detection.** In order to check the precision of the methods, we applied a full marker identification procedure accounting for multiple testing with each algorithm. A Bonferroni correction using  $M_{eff}$  as the number of tests was performed, with  $M_{eff}$  being the effective number of tests as estimated using the Gao procedure [24]. For the present application one obtains  $M_{eff} = 3, 527$ .

We found the GWAS results to be very similar across all algorithms except *BOLT-LMM*. For the other algorithms all GWAS analyzes led to the same list of markers (provided in [Table 2](#)), and the correlation between the different lists of (log-transformed) p-values were found to be higher than 0.99. The correlation observed between *BOLT-LMM* and the other methods is 0.97, leading to a different list of significant markers. This could be expected since *BOLT-LMM* does not rely on the same modeling than the other algorithms.

Several approximate approaches have been proposed in order to further reduce the computational burden. In *gaston* a score test procedure is provided. *FaST-LMM* also comes with an approximate version of the test procedure, where the variance components are estimated only once without further refitting of the restricted maximum likelihood for each marker. Although such strategies speed up the computational performance, they result in p-values that are poorly estimated, and possibly to different lists of significant markers (see [S1 Code: TwoVarianceComponents/Rmd/SupplementaryResults\\_TVC.html](#)). Alternatively, it has been advocated to perform GWAS by considering each chromosome separately for the variance component estimation, using a kinship matrix estimated on markers that do not belong to the candidate chromosome under study [25]. Applying this strategy led to an increase of power (i.e. smaller p-values for the best candidates) for the approximate approaches. However in the present example this gain in power did not increase the number of markers detected by the approximate methods up to that detected by the exact ones. Additionally, note that exact procedures can also be combined to this “leave-one-chromosome-out” approach, (see also [S1 Code: TwoVarianceComponents/Rmd/SupplementaryResults\\_TVC.html](#)).

**Table 2.  $-\log_{10}(\text{p-value})$  of markers detected by all exact algorithms for DM\_Y\_Flo.**

<b>Marker</b>	<b><math>-\log_{10}(pval)</math></b>
SYN10537	5.61
SYN10528	5.61
PZE-101030022	5.07
PZE-101123079	4.87
SYN13856	5.19

List of significant markers at a nominal level of 5% (Gao correction for multiple testing).

<https://doi.org/10.1371/journal.pcbi.1009659.t002>



In the next two sections BOLT-LMM and FaST-LMM are not considered since they do not handle models with more than 2 variance components.

**Four variance components. Dataset.** The maize dataset (called Factorial in the following) consists in hybrids derived from an incomplete factorial crossing design between flint and dent lines [26]. A total of  $n_D = 123$  parental dent lines and  $n_F = 86$  parental flint lines were crossed to obtain  $n_H = 1,254$  hybrids. Parental lines were genotyped at 32,486 markers. Two phenotypic traits were quantified: grain yield (GY) and grain moisture (GM). We present the results obtained on GM. The association study was performed using the following model: at a marker  $\ell$  and for a given trait, one has:

$$\begin{aligned} Y &= 1\mu + X_\ell\beta_\ell + Z_F G_F + Z_D G_D + Z_H G_H + E \\ G_F &\sim \mathcal{N}(0, \sigma_F^2 K_F), \quad G_D \sim \mathcal{N}(0, \sigma_D^2 K_D) \\ G_H &\sim \mathcal{N}(0, \sigma_H^2 \Phi), \quad E \sim \mathcal{N}(0, \sigma_E^2 I) \\ G_F &\perp G_D \perp G_H \perp E \end{aligned} \quad (1)$$

with  $Y$  is the vector of phenotype,  $X_\ell$  is the vector corresponding to the number of copies of allele 1 present at marker  $\ell$ ,  $\beta_\ell$  the effect associated with the allele 1,  $G_F$ ,  $G_D$  and  $G_H$  are the random polygenic effects corresponding to the flint parent, the dent parent and their specific interaction, respectively, and  $Z_F$ ,  $Z_D$  and  $Z_H$  are the associated incidence matrices. Correlation matrices  $K_F$  and  $K_D$  correspond to the kinship matrices between the dent (resp. flint) parental lines. Matrix  $\Phi$  corresponds to the double relatedness matrix between hybrids, of general term

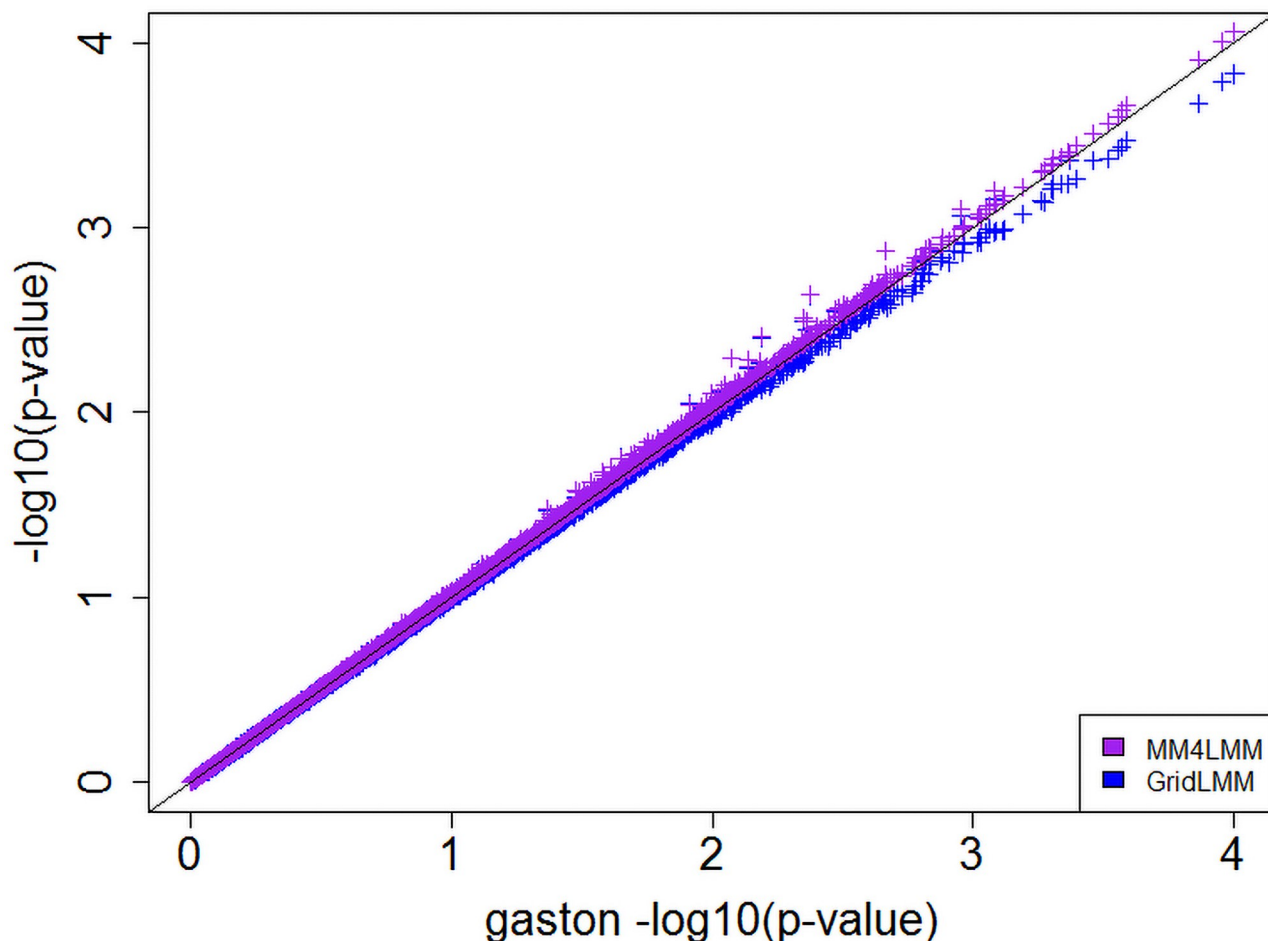
$$\hat{\Phi}_{h,h'} = (\hat{K}_F)_{f,f'} \times (\hat{K}_D)_{d,d'}$$

where  $h$  (resp.  $h'$ ) is the hybrid resulting from the crossing between the flint and dent lines  $f$  and  $d$  (resp.  $f'$  and  $d'$ ). Lastly,  $E$  is the error vector.

**Algorithm performances.** The analysis of the Factorial dataset required 11.25h with `gaston`, 2.5h with `MM4LMM` and 107 seconds with `GridLMM`. The performance of `GEMMA` is not reported since it exceeded 20h. In terms of p-values the two exact methods led to very similar results (correlations between  $-\log_{10}(\text{p-values})$  series  $>0.999$ ). The correlation between any exact method and `GridLMM` was found to be of the same order but with p-values being bigger for `GridLMM` compared with exact methods, see Fig 1. Although in the present example no marker was declared significant whatever the method, the p-value inflation could result in a slight loss power in some applications.

Importantly, the model considered for the analysis only assumed an additive effect for the marker. However in most cases the analysis of a hybrid panel requires one to account for both an additive and a dominance fixed effects. Including a dominance effect is possible when using `MM4LMM`, `GEMMA` and `gaston`, but not when using `GridLMM` that requires the marker effect to be included in the model through a single numeric incidence vector. This constitutes a significant limitation for the use of `GridLMM` applied to GWAS in the context of plant or animal genetics.

**More than 4 variance components. Dataset.** The dataset (called the NAM dataset hereafter) is constituted of  $n_H = 951$  maize hybrids derived from an incomplete factorial crossing design between  $n_D = 875$  dent lines and  $n_F = 883$  flint lines [27]. All hybrids were evaluated for 4 phenotypes (i.e. response variable). Here we focus on the Dry Matter Yield (hereafter DMY), the results obtained with the other traits being very similar. Hybrids were evaluated in 8 different trials performed in two countries, with a number of measurements per hybrid in a trial going from 0 to 2, most hybrids being measured once. The number of measurements per trial



**Fig 1.** Log-transformed p-values concordance between *gaston* and GridLMM, and *gaston* and MM4LMM.

<https://doi.org/10.1371/journal.pcbi.1009659.g001>

goes from 896 to 1001, the total number of measurements being 7,725. The goal of the study was to evaluate the contribution of the 2 parental populations (dent and flint) to the phenotypic variability.

**Statistical analysis.** Two different strategies were considered for the statistical analysis. The first strategy consisted in a 2-step analysis. In step 1 a first model was fitted to correct the phenotypic data for field effects—see [S2 Appendix](#) for details. In a second step a Variance Component Analysis was performed using the following model:

$$\begin{aligned}
 Y &= 1\mu + X_T\beta_T + Z_F G_F + Z_D G_D + Z_H G_H + E \\
 G_F &\sim \mathcal{N}(0, \sigma_F^2 K_F), \quad G_D \sim \mathcal{N}(0, \sigma_D^2 K_D) \\
 G_H &\sim \mathcal{N}(0, \sigma_H^2 \Phi), \quad E \sim \mathcal{N}(0, \sigma_E^2 I) \\
 G_F &\perp G_D \perp G_H \perp E
 \end{aligned} \tag{2}$$

where  $Y$  here stands for the corrected phenotypes obtained in step 1. Here  $\beta_T$  is the the vector of trial fixed effects and  $X_T$  is the associated incidence matrix, the other terms being defined as in [Eq \(1\)](#).



The second strategy consisted in performing a one-step analysis on the whole dataset, including all trials in a single analysis. The model has to account for both genetics and trial/field effects, but also for gene×environment interactions, a feature that is relevant whenever trials are expected to be diverse (in terms of environmental conditions) and genetic effects sensitive to the environment. The random interaction terms are assumed to have specific variances in each trial, leading to

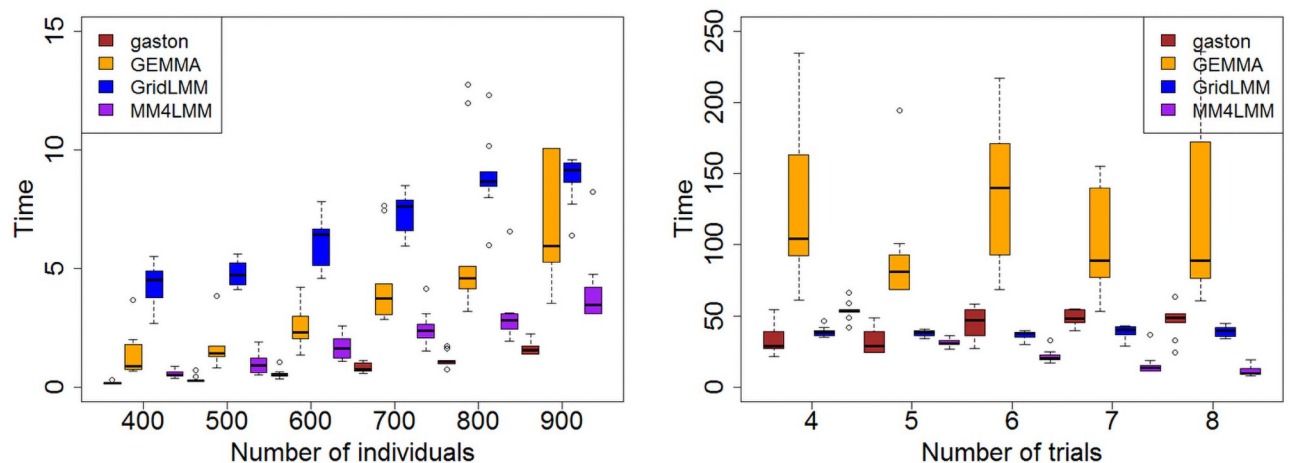
$$\begin{aligned}
 Y = & 1\mu + X_T\beta_T + Z_F G_F + Z_D G_D + Z_H G_H \\
 & + \sum_{k=1}^{N_T} [Z_{FT_k} G_{FT_k} + Z_{DT_k} G_{DT_k} + Z_{HT_k} G_{HT_k} \\
 & + Z_{rowT_k} G_{rowT_k} + Z_{colT_k} G_{colT_k} + Z_{T_k} E_k] \\
 G_{FT_k} \sim & \mathcal{N}(0, \sigma_{FT_k}^2 K_F), \quad G_{DT_k} \sim \mathcal{N}(0, \sigma_{DT_k}^2 K_D) \\
 G_{HT_k} \sim & \mathcal{N}(0, \sigma_{HT_k}^2 \Phi), \quad G_{rowT_k} \sim \mathcal{N}(0, I_{n_k}) \\
 G_{colT_k} \sim & \mathcal{N}(0, I_{n_k}), \quad E_k \sim \mathcal{N}(0, \sigma_{ET_k}^2 I_{n_k}) \\
 G_F \perp & G_D \perp G_H \perp G_{FT_k} \perp G_{DT_k} \perp G_{HT_k} \perp E_k,
 \end{aligned} \tag{3}$$

with  $N_T$  the number of trials,  $G_{FT_k}$  (resp.  $G_{DT_k}$  and  $G_{HT_k}$ ) the polygenic effect associated to the flint lines (resp. dent lines and flint-dent line interaction) within trial  $k$ ,  $Z_{FT_k}$ ,  $Z_{DT_k}$  and  $Z_{HT_k}$  the associated incidence matrices,  $G_{rowT_k}$  and  $G_{colT_k}$  are row and column effects within trial  $k$ ,  $Z_{rowT_k}$  and  $Z_{colT_k}$  the associated incidence matrices,  $n_{r_k}$  and  $n_{c_k}$  are the numbers of row and column within trial  $k$  and  $E_k$  the residual effect within the trial  $k$ ,  $n_k$  the number of hybrids within trial  $k$  and  $Z_{T_k}$  the incidence matrix associated to  $E_k$ . Note that residual variances are also assumed to be specific to each trial.

In multi-trial analyzes different factors may impact the computational efficiency of the inference algorithm, including the number of observations, the number of trials, the computational tricks that may be implemented and the number of variance components. In order to quantify the effect of these factors on the different algorithms considered here, we first considered simulated data mimicking a subsample of the full dataset. Synthetic phenotypes were simulated based on the observed genotypic data and the experimental design of the NAM dataset. More precisely, we subsampled sets of hybrids from the NAM dataset and then simulated the phenotype based on Model (2), using the observed kinship matrices and considering no fixed effects (i.e. all observations have a null mean). The computational performance of the 4 algorithms are presented on both the simulated and the NAM data, analyzed using either Model (2) or (3). Note that Model (3) has 44 components when assuming a common error variance, and 51 otherwise, making the model fitting significantly more involving than the previous analyses. Consequently we considered variance component analysis rather than association analysis.

**Estimation.** In terms of variance estimation, all algorithms yielded the same results when applied to the complete NAM dataset, using Model (2). The table of the variance component estimates is given in S3 Appendix. Similar conclusions were obtained when considering other phenotypic traits (results not shown).

**Computational time.** We investigated how the number of observations impacts the computational performance of the different algorithms. We first considered a “one-trial” simulation setting where  $n = 400, 500, \dots, 900$  hybrids were randomly selected from the 951 available ones and phenotypes were generated as described in the previous paragraph. This process was repeated 10 times. The data were then analyzed using Model (2). The results are displayed



**Fig 2. Computational time for variance component analysis with simulated data.** Computational time of the algorithms with respect to the number of observations (left) and the number of trials (right).

<https://doi.org/10.1371/journal.pcbi.1009659.g002>

in Fig 2 (left). As expected the computational time of all procedures increases with  $n$ , with *gaston* and *MM4LMM* being the top two algorithms. Compared to the Hybrid data application the lower performance of *GridLMM* comes from the fact that here only a single model is fitted.

In a second setting we investigated the impact of the number of trials on the computational performance of the different algorithms. To this end, we fixed the total number of observations at  $n = 2400$ , and randomly sampled hybrids in  $n_T = 4, \dots, 8$  trials (balancing the contribution of the different trials). For each number of trials, the simulation process was repeated 10 times and Model (2) was used for the analysis. Results are presented in Fig 2 (right). Apart from *MM4LMM*, all algorithms are insensitive to  $n_T$ : as soon as the number of variance components in the model does not depend on  $n_T$  the algorithms scale with  $n$  (which is fixed here) only. The behavior of *MM4LMM* differs from the other algorithms because *MM4LMM* automatically selects whether the MME trick (described in the Methods section) should be used or not. In the present setting one can show that the algorithmic complexity of the MM algorithm is  $O(n^3 + K \times n^2 + p^3)$  whereas the one of MM combined with the MME trick is  $O((n_F + n_D + n_H + p)^3 + (n_F^2 + n_D^2 + n_H^2 + \sum_{k=1}^K n_k) + p^3)$ , where  $p = \text{rank}(X)$ . Here quantities  $n_F$ ,  $n_D$  and  $n_H$  decrease with increasing values of  $n_T$ , and the computational time of *MM4LMM* decreases accordingly. Consequently, depending on the balance between the number of random effects and of measurements, the MME trick may be beneficial (e.g. when all trials are considered) or detrimental (eg when only 2 or 4 trials are included in the analysis).

Although the previous simulated settings allow one to disentangle the effect of the number of observations and the number of trials, we considered a more realistic setting where both numbers increase together, i.e. a setting where the number of slots in a trial does not depend on  $n_T$ . Here we used the real NAM data, built intermediate versions of the complete dataset by selecting subsets of 2, 4 or 6 trials among the 8 available ones and analyzed these subsets with Model (2). Table 3 displays the computational time associated with each algorithm. One observes that differences in terms of performance may be important, as quantified using the ratio between the worst and the best computational time obtained (last column), with no single algorithm being uniformly the most efficient. Note that the Bayesian estimation procedure implemented in the *MCMCglmm* package [28] was also considered for this analysis. Results

**Table 3. Computational time (in sec.) associated to the analysis of different subsamples of trials of the NAM dataset, using Model (2).** Bold numbers correspond to the best performance.

Nb Trials	Avg Nb Obs	gaston		MM4LMM		GEMMA		GridLMM		Ratio
		mean	sd	mean	sd	mean	sd	mean	sd	
2	1,931.25	<b>16.28</b>	4.43	42.15	7.05	58.35	16.48	23.02	4.45	3.6
4	3,862.50	170.44	24.15	288.08	44.62	386.03	30.02	<b>106.74</b>	14.28	3.6
6	5,793.75	659.35	72.58	<b>308.11</b>	25.87	1244.19	152.15	326.58	37.15	4.0
8	7,725	1786.87		<b>333.15</b>		3100.94		792.59		9.3

<https://doi.org/10.1371/journal.pcbi.1009659.t003>

**Table 4. Computational time (in sec.) associated to the analysis of the NAM dataset using Model (3).**

gaston	MM4LMM	GEMMA
5207	15739	>30000

<https://doi.org/10.1371/journal.pcbi.1009659.t004>

were similar in terms of parameter estimation but are not reported here due to prohibitive computational costs (>24h for a single 2-trial analysis).

We then considered the analysis of all trials using Model (3). In this context the use of the MME trick would be detrimental since the cumulative size of the correlation matrices may become significantly high. The computational times obtained are summarized in Table 4. Note that no performance is reported for GridLMM since it cannot handle such a large number of variance components (the memory size required for matrix storage—>80Go for the first iteration—becomes highly prohibitive). The table of the variance component estimates is given in S4 Appendix.

Lastly, note that Model (3) assumes a homogeneous error variance across trials, a strong hypothesis that is highly unlikely in practice. The third strategy is then to analyze the full dataset using Model (3), except that one now assumes that  $E_k \sim \mathcal{N}(0, \sigma_{ET_k}^2 I_{n_k})$  for each trial  $k$ . A comparison between the homogeneous and trial specific error variance models based on the BIC criterion confirms that the heterogeneous error variance model is to be preferred for the NAM dataset (BIC(homogeneous)=15,279, BIC(specific)=14,850, smaller is better). Although highly desirable for the statistical analysis, this last model cannot be fitted by the algorithms presented here except for MM4LMM, that run the analysis in 4410 seconds.

## Discussion

As illustrated in the Results section, the MM algorithm is a competitive algorithm in the context of ReML inference for VC mixed models. When  $K = 2$ , it can be combined with the simultaneous orthogonalization trick to compete with algorithms dedicated to the 2-component case such as FaST-LMM or BOLT-LMM when applied to datasets of moderate size. For large values of  $K$  the number of available methods reduces to gaston, GEMMA and MM4LMM, the last one being the more versatile to e.g. handle models including heterogeneous error variances. This versatility is important in the context of GS, GWAS or VCA since it gives access to models more complex than the “reference” model that only includes a polygenic and an error components (as proposed in [10]). Note that the MM4LMM R package also provides estimated standard error values for the variance parameters that help to better interpret the results when the number of variance components is high. These features make MM4LMM a method of choice for i) the analysis of multi-environment trials and ii) the analysis of crossing designs where the modeling of non-additive marker effects is at stake, as these two settings are usually characterized by a large number of variance components.

The analysis of the NAM dataset showed that the MME trick is only beneficial if the number of observations is much higher than the total number of latent effects, but may be detrimental otherwise. Also note that the MME trick requires each matrix  $V_k$  in the model to be invertible, a stringent condition that is not required by the ReML procedure itself, and is not satisfied in many applications. For these reasons the use of the MME trick should be restricted to some applications only, an optional strategy that is available in the `MM4LMM` package but not in most packages/softwares.

## Conclusion

The Min-Max algorithm is a simple alternative to the by-default AI-ReML algorithm that is commonly implemented in most packages. We demonstrated that most classical speed-up tricks used in the context of mixed model inference can be easily combined with the MM algorithm, yielding an efficient estimation procedure that can compete with state of the art competitors in most contexts that are commonly encountered in animal or plant genetics—even those for which efficient alternatives exist (such as the  $K = 2$  case). This first study opens a way for new algorithmic developments in the field of VC mixed models and more generally in ReML inference for other classes of mixed models. A limitation for such further developments is the fact that MM methods require the derivation of a specific surrogate function for each class of mixed model to be considered, making the extension of the inference procedure to e.g. auto-regressive or factor analytic models [29] not straightforward.

## Methods

### Variance component mixed model

In this article we focus on variance component models of the form:

$$Y \sim \mathcal{N}(X\beta, \sum_{k=1}^K \sigma_k^2 V_k) \quad (4)$$

where  $Y$  is a vector of  $n$  observations,  $X$  is an incidence matrix,  $\beta$  is the vector of fixed effects,  $V_k$  is the (known) correlation matrix associated to the  $k^{\text{th}}$  random effect and  $\gamma = (\sigma_1^2, \dots, \sigma_K^2)$  is the vector of variances associated to the  $K$  random effects. In what follows  $X$  is assumed to be a full rank matrix. A special case of Model (4) is the following mixed model:

$$Y = X\beta + \sum_{k=1}^{K-1} Z_k U_k + E$$

$$\text{with } \begin{cases} U_k \sim \mathcal{N}(0, \sigma_k^2 R_k), & k = 1, \dots, K-1, \\ E \sim \mathcal{N}(0, \sigma_K^2 V_K) \\ U_1 \perp \dots \perp U_{K-1} \perp E \end{cases}, \quad (5)$$

where  $U_k$  is the  $k^{\text{th}}$  random effect vector of size  $n_k$ ,  $Z_k$  (resp.  $R_k$ ) is the incidence matrix (resp. the correlation matrix) associated with random effect  $U_k$ ,  $E$  is an error vector, and notation  $A \perp B$  stands for “ $A$  and  $B$  are independent”. Model (5) boils down to Model (4) where  $V_k = Z_k R_k Z_k^T$ . Lastly, we introduce  $\Sigma_\gamma = \sum_{k=1}^K \sigma_k^2 V_k$ , the covariance matrix of vector  $Y$ .

The goal is to infer the unknown fixed effects and variance parameters  $\beta$  and  $\gamma$ . Here we consider the Restricted Maximum Likelihood (ReML) estimation procedure [30, 31].

Let  $\Pi_{X^\perp} = I - X(X^T X)^{-1} X^T$  be the projection matrix on  $\text{span}(X)^\perp$ , and  $M$  be any matrix built from the columns of  $\Pi_{X^\perp}$  such that  $M$  is of full rank and  $\text{rank}(M) = \text{rank}(\Pi_{X^\perp}) = m$ .

Applying  $M$  to the initial data vector  $y$  allows one to get rid of the fixed effects. The restricted (log-) likelihood corresponds to the (log-) likelihood of the transformed data  $My$ , and has the following expression (up to a constant):

$$\mathcal{L}_R(\gamma) = -\frac{1}{2} \left[ \log(|X^T \Sigma_\gamma^{-1} X|) + \log(|\Sigma_\gamma|) + y^T P_\gamma y \right] \quad (6)$$

where  $|H|$  stands for the determinant of matrix  $H$  and

$$\begin{aligned} P_\gamma &= M^T (M \Sigma_\gamma M^T)^{-1} M \\ &= \Sigma_\gamma^{-1} - \Sigma_\gamma^{-1} X (X^T \Sigma_\gamma^{-1} X)^{-1} X^T \Sigma_\gamma^{-1}. \end{aligned}$$

Note that  $\mathcal{L}_R(\gamma)$  does not depend on  $\beta$  (since  $MX = 0$  by construction), nor on the specific choice of  $M$  thanks to the second expression of  $P_\gamma$  above [32]. Variance parameters  $\gamma$  can be estimated by applying the classical Maximum Likelihood procedure to  $\mathcal{L}_R$ , then fixed effects can be obtained using the following formula:

$$\hat{\beta} = (X^T \Sigma_\gamma^{-1} X)^{-1} X^T \Sigma_\gamma^{-1} y.$$

Although quite popular, the ReML procedure may be quite challenging from a computational point of view, the bottleneck being the maximization of the log-likelihood (6) w.r.t.  $\gamma$ . Although the first derivative of  $\mathcal{L}_R$  with respect to  $\sigma_k^2$  has a simple expression:

$$\frac{\partial \mathcal{L}_R}{\partial \sigma_k^2} = -\frac{1}{2} [tr(P_\gamma V_k) - y^T P_\gamma V_k P_\gamma y],$$

solving the  $K$  equations  $\frac{\partial \mathcal{L}_R}{\partial \sigma_k^2} = 0, k = 1, \dots, K$  does not lead to a closed form expression for  $\hat{\gamma}$ . Consequently likelihood maximization has to be performed numerically. The next section presents the Newton optimization algorithm and its derivatives to obtain the ReML variance estimates.

## Newton based algorithms

**Newton algorithm.** Let first rewrite Model (4) as

$$Y = X\beta + ZU + E$$

where  $Z = (Z_1 | \dots | Z_{K-1})$ ,  $U = (U_1^T | \dots | U_{K-1}^T)^T$ . The joint distribution of  $(U, E)$  is

$$\begin{aligned} \begin{bmatrix} U \\ E \end{bmatrix} &\sim \mathcal{N} \left( 0, \sigma_K^2 \begin{bmatrix} G_\delta & 0 \\ 0 & V_K \end{bmatrix} \right) \\ \text{where } \delta &= \left( \frac{\sigma_1^2}{\sigma_K^2}, \dots, \frac{\sigma_{K-1}^2}{\sigma_K^2} \right) \\ \text{and } G_\delta &= \begin{pmatrix} \delta_1 R_1 & 0 & \dots & 0 \\ 0 & \delta_2 R_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \delta_{K-1} R_{K-1} \end{pmatrix}. \end{aligned}$$

The restricted likelihood  $\mathcal{L}_R$  can be reformulated as:

$$\begin{aligned} \mathcal{L}_R(\delta, \sigma_K^2) &= -\frac{1}{2} [m \log(\sigma_K^2) + \log(|X^T \Sigma_\delta^{-1} X|) \\ &\quad + \log(|\Sigma_\delta|) + \frac{y^T P_\delta y}{\sigma_K^2}] \end{aligned}$$

where  $\Sigma_\delta = ZG_\delta Z^T + V_K$ ,  $P_\delta = \sigma_K^2 P_\gamma$  and  $m = \text{rank}(M)$ . Starting from this last expression, one can perform optimization of  $\mathcal{L}_R$  using an iterative scheme like the Newton algorithm that requires the first and second derivatives of  $\mathcal{L}_R$  w.r.t. both  $\delta$  and  $\sigma_K^2$ . The first derivatives are

$$\begin{aligned} [\nabla \mathcal{L}_R(\delta, \sigma_K^2)]_k &= -\frac{1}{2} \left( \text{tr}(P_\delta V_k) - \frac{y^T P_\delta V_k P_\delta y}{\sigma_K^2} \right) \quad 1 \leq k < K, \\ [\nabla \mathcal{L}_R(\delta, \sigma_K^2)]_K &= -\frac{1}{2} \left( \frac{m}{\sigma_K^2} - \frac{y^T P_\delta y}{\sigma_K^4} \right). \end{aligned}$$

Similarly, the second derivatives are

$$\begin{aligned} [\mathcal{H} \mathcal{L}_R(\delta, \sigma_K^2)]_{kk'} &= \frac{1}{2} \text{tr}(P_\delta V_k P_\delta V_{k'}) - \frac{y^T P_\delta V_k P_\delta V_{k'} P_\delta y}{\sigma_K^2}, \\ [\mathcal{H} \mathcal{L}_R(\delta, \sigma_K^2)]_{kK} &= -\frac{1}{2} \frac{y^T P_\delta V_k P_\delta y}{\sigma_K^4}, \\ [\mathcal{H} \mathcal{L}_R(\delta, \sigma_K^2)]_{KK} &= \frac{m}{2\sigma_K^4} - \frac{y^T P_\delta y}{\sigma_K^6}. \end{aligned}$$

Denoting  $\nabla_{\mathcal{L}_R}^{(t)}$  and  $\mathcal{H}_{\mathcal{L}_R}^{(t)}$  the gradient and the Hessian matrix of  $\mathcal{L}_R$  evaluated at point  $(\delta^{(t)}, \sigma_K^{2(t)})$  respectively, the Newton method then iterates the following recursion:

$$\begin{pmatrix} \delta^{(t+1)} \\ \sigma_K^{2(t+1)} \end{pmatrix} = \begin{pmatrix} \delta^{(t)} \\ \sigma_K^{2(t)} \end{pmatrix} - [\mathcal{H}_{\mathcal{L}_R}^{(t)}]^{-1} \nabla_{\mathcal{L}_R}^{(t)}.$$

A classical shortcut consists in making use of the fact that the last gradient component leads to an explicit expression of  $\sigma_K^2$  when the ratio  $\delta$  is known:

$$\hat{\sigma}_K^2(\delta) = \frac{y^T P_\delta y}{m}. \quad (7)$$

One can then apply the Newton algorithm to update  $\delta$  only, which reduces the number of unknown parameter by one in the Newton update procedure. This trick is classically known as the “profiling” trick. Additional computational shortcuts are presented in the Computational shortcuts section.

**Fisher scoring and average information.** It has been suggested [11, 30] that the use of alternative matrices in place of the Hessian matrix in the Newton procedure could be beneficial in terms of convergence rate and/or computational burden. The first alternative, known as the Fisher algorithm, consists in replacing  $\mathcal{H}_{\mathcal{L}_R}^{(t)}$  by its expected value. The expectations of the Hessian matrix terms are

$$\begin{aligned} \mathbb{E}[\mathcal{H} \mathcal{L}_R(\delta, \sigma_K^2)]_{kk'} &= -0.5 \times \text{tr}(P_\delta V_k P_\delta V_{k'}), \\ \mathbb{E}[\mathcal{H} \mathcal{L}_R(\delta, \sigma_K^2)]_{kK} &= -0.5 \times \text{tr}(P_\delta V_k) / \sigma_K^4, \\ \mathbb{E}[\mathcal{H} \mathcal{L}_R(\delta, \sigma_K^2)]_{KK} &= -0.5 \times m / \sigma_K^4. \end{aligned}$$

A second alternative is the use of the Average Information (AI) matrix [12]. The AI matrix is defined as the average of the Hessian and its expectation. The efficiency of this strategy leads

in the general expression of the resulting matrix. One has

$$AI_{kk'} = \frac{y^T P_\delta V_k P_\delta V_{k'} P_\delta y}{2\sigma_k^2},$$

$$AI_{kk} \approx \frac{y^T P_\delta V_k P_\delta y}{2\sigma_k^4}, \quad AI_{KK} = \frac{y^T P_\delta y}{2\sigma_K^6},$$
(8)

where for the second term the approximation  $\text{tr}(V_k P_\delta) \approx y^T P_\delta V_k P_\delta y$  is used. Compared with the previous expressions obtained for the Newton and FS algorithms, computing the AI matrix does not involve any trace computation anymore. Note that  $P_\delta$  is computed at each step using formula

$$P_\delta = \Sigma_\delta^{-1} - \Sigma_\delta^{-1} X (X^T \Sigma_\delta^{-1} X)^{-1} X^T \Sigma_\delta^{-1}$$

where  $\delta$  and  $\sigma_k^2$  are fixed at their current value.

**Computational shortcuts. Simultaneous orthogonalization.** As mentioned in the Newton based algorithms section, the profiling trick reduces the computational complexity by discarding one of the variance component in the update procedure: the numerical optimization only applies to  $\delta$ ,  $\sigma_k^2$  being estimated afterwards using its explicit expression (7). When applied to the case where  $K = 2$ , profiling may be combined to the simultaneous orthogonalization of the two covariance matrices to obtain an even simpler expression of the restricted likelihood. Assuming one of the two matrices (say  $V_2$ ) is positive definite, then there exist a matrix  $\Lambda$  and a diagonal matrix  $D$  such that

$$\Lambda V_1 \Lambda^T = D \quad \text{and} \quad \Lambda V_2 \Lambda^T = I_n.$$

One can reexpress the restricted log-likelihood associated with Model (4) as a function of  $\sigma_2^2$ ,  $\delta$  and  $D$  as follows:

$$\mathcal{L}_R(\delta, \sigma_2^2) = -\frac{1}{2} [m \log(\sigma_2^2) + \log(|D\delta + I_n|)$$

$$+ \log(|\tilde{X}^T (D\delta + I_n)^{-1} \tilde{X}|) + \frac{1}{\sigma_2^2} \tilde{y}^T P_\delta \tilde{y}],$$
(9)

where  $\tilde{y} = \Lambda y$  and  $\tilde{X} = \Lambda X$ . The expression of  $\hat{\sigma}_2^2(\delta)$  can then be plugged back into Eq (9) to obtain a function that depends on  $\delta$  only:

$$\mathcal{L}_R(\delta) = -\frac{1}{2} [m \log(\hat{\sigma}_2^2(\delta)) + \log(|D\delta + I_n|)$$

$$+ \log(|\tilde{X}^T (D\delta + I_n)^{-1} \tilde{X}|) + m].$$
(10)

This last expression can then be optimized w.r.t.  $\delta$ . This strategy is implemented in the R package `gaston` where a Newton Raphson algorithm, followed by a Brent algorithm (if the procedure has not already converged) are used for the optimization of (10), and also in FaST-LMM where the optimization is first performed on a grid then refined using the Brent algorithm [7, 9]. One can observe that the simultaneous orthogonalization trick drastically reduces the computational burden whenever many models with identical random effects but different fixed effects have to be adjusted, the orthogonalization being performed only once (i.e.  $\Lambda$  is common to all models).

**Henderson equation shortcut.** As mentioned earlier, one needs to invert matrix  $\Sigma_\delta$  at each step to update matrix  $P_\delta$ . This operation is the computational bottleneck of the optimization



procedure and may become cumbersome when the number of measurements  $n$  is large. In some configurations this step may be relaxed by deriving the quantities required for the update of  $\delta$  from the Henderson Mixed Model Equations (MME)

$$\begin{bmatrix} X^T V_K^{-1} X & X^T V_K^{-1} Z \\ Z^T V_K^{-1} X & Z^T V_K^{-1} Z + G_\delta^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X^T V_K^{-1} y \\ Z^T V_K^{-1} y \end{bmatrix} \quad (11)$$

Noting  $C$  the coefficient matrix appearing in the left hand side of the equation, one first notices that solving the system requires the inversion of  $C$ , of size  $\sum_k n_k + p$ , where  $n_k$  is the length of vector  $U_k$  and  $p$  is the rank of matrix  $X$ . Second, it has been shown that the quantities appearing in AI matrix (8) can be reexpressed using  $C^{-1}$ , details are given in S5 Appendix [12]. One has:

$$AI_{kk'} = \frac{[y^T P_\delta Z_k] R_k [Z_k^T P_\delta Z_{k'}] R_{k'} [Z_{k'}^T P_\delta y]}{2\sigma_K^2},$$

$$AI_{kk} = \frac{[y^T P_\delta Z_k] R_k [Z_k^T P_\delta y]}{2\sigma_K^4}, \quad AI_{KK} = \frac{y^T P_\delta y}{2\sigma_K^6}$$

and

$$\begin{aligned} Z^T P_\delta y &= G_\delta^{-1} \hat{u} \\ Z^T P_\delta Z &= G_\delta^{-1} - G_\delta^{-1} [C^{-1}]_{uu} G_\delta^{-1} \\ P_\delta y &= V_K^{-1} \hat{e}. \end{aligned} \quad (12)$$

where  $\hat{e} = y - X\hat{\beta} - Z\hat{u}$  and  $[C^{-1}]_{uu}$  corresponds to the submatrix of  $C^{-1}$  associated with the random component  $u$ . Assuming all matrices  $V_k$  are invertible (i.e. definite positive), all these expressions are easily obtained from  $\hat{\beta}$ ,  $\hat{u}$  and  $C^{-1}$ . As soon as  $\sum_k n_k + p \ll n$  it becomes computationally efficient to compute the AI matrix through the MME rather than through direct inversion of  $P_\delta$ . In the following, this shortcut will be referred to as the “MME trick”.

## Min-Max algorithm for ReML

**MM algorithm for ReML inference.** MM algorithms represent another class of iterative schemes [15]. We provide a brief overview of the MM principle based on the previous reference. Consider an optimization problem where one aims at finding the minimizer  $\theta^*$  of a function  $f(\theta)$  (in our setting  $f = -\mathcal{L}_R$  and  $\theta = \gamma$ ), one builds at each step  $t$  a surrogate function  $g^{(t)}$  satisfying

$$g^{(t)}(\theta) \geq f(\theta) \quad \text{and} \quad g^{(t)}(\theta^{(t-1)}) = f(\theta^{(t-1)}),$$

where  $\theta^{(t-1)}$  is the current evaluation of  $\theta^*$ . Assuming the surrogate function can be minimized easily, one defines

$$\theta^{(t)} = \arg \min_{\theta} g^{(t)}(\theta).$$

One can show that the sequence  $(\theta^{(t)})_{t \geq 1}$  satisfies the descent property  $f(\theta^{(t+1)}) \leq f(\theta^{(t)})$ . In practice, the convergence is assessed using a convergence criterion such as

$$\|\theta_t - \theta_{t-1}\|_\infty = \max_i (|\theta_{t,i} - \theta_{t-1,i}|) < \epsilon$$

or

$$f(\theta_{t-1}) - f(\theta_t) < \epsilon.$$

In the present article both criteria were used with  $\epsilon = 10^{-5}$ .

In the context of variance component mixed models, a MM method has been proposed to maximize the likelihood [16]. Following the same line of proof, we present an MM algorithm for ReML inference. The main difficulty to apply MM optimization is to derive the sequence of surrogate functions. Proposition 1 provides the surrogate function at step  $t + 1$  for the ReML optimization problem:

**Proposition 1** Define function  $g^{(t+1)}(\gamma)$  as

$$g^{(t+1)}(\gamma) = \frac{1}{2} \sum_{k=1}^K \left[ \sigma_k^2 \text{tr}(P_\gamma^{(t)} V_k) + \frac{\sigma_k^{4(t)}}{\sigma_k^2} y^T P_\gamma^{(t)} V_k P_\gamma^{(t)} y \right] + \log(|M \Sigma_\gamma^{(t)} M^T|) - m, \quad (13)$$

where  $m = \text{rank}(M)$ . Then

$$g^{(t+1)}(\gamma) \geq -\mathcal{L}_R(\gamma)$$

where equality holds at point  $\gamma^{(t)}$ .

The proof of Proposition 1 is adapted from the one given for ML inference [16] and is given in [S6 Appendix](#). Because at each step  $t$  the surrogate function  $g^{(t)}$  is linear with respect to  $\sigma_1^2, \dots, \sigma_K^2$ , one easily obtains its optimizer by setting its gradient at 0. This provides the following update for the variance parameters:

$$\sigma_k^{2(t+1)} = \sigma_k^{2(t)} \sqrt{\frac{y^T P_\gamma^{(t)} V_k P_\gamma^{(t)} y}{\text{tr}(P_\gamma^{(t)} V_k)}}. \quad (14)$$

The next section presents the adaptation of the computational tricks presented in the Newton based algorithms section to the ReML MM procedure.

**Computational shortcuts for the ReML MM procedure.** **Two matrix shortcut.** In the particular case when  $K = 2$  the correlation matrices can be jointly orthogonalized. Similar to the profiling trick, we introduce

$$g^{(t+1)}(\delta) = \frac{1}{2} \left[ m \log \left( \frac{1}{m} \tilde{y}^T P_\delta^{(t)} \left( \frac{\delta^{2(t)}}{\delta} D + I_n \right) P_\delta^{(t)} \tilde{y} \right) + \text{tr}(P_\delta^{(t)} (D\delta + I)) + c^{(t)} \right],$$

where  $c^{(t)}$  is an irrelevant constant. As detailed in [S7 Appendix](#), one can show that optimizing function  $g^{(t+1)}$  boils down to solving a quadratic function that admits a unique positive solution corresponding to  $\delta^{(t+1)}$ .

**K matrix shortcut.** When relevant, the MME trick can be applied to speed up the computation of the quantities appearing in the surrogate function (13). Since  $P_\delta = \sigma_K^2 P_\gamma$  and  $V_k = Z_k R_k Z_k^T$ , the update formulas (14) can be rewritten as follows:

$$\sigma_k^{2(t+1)} = \sigma_k^{2(t)} \sqrt{\frac{y^T P_\delta^{(t)} Z_k R_k Z_k^T P_\delta^{(t)} y}{\sigma_K^{2(t)} \text{tr}(Z_k^T P_\delta^{(t)} Z_k R_k)}}$$

and can be computed using expressions (12) for  $k = 1, \dots, K - 1$  (recall that  $Z$  does not include  $Z_K$ ). For the case  $k = K$ , the numerator can be easily obtained from the expression of  $P_\delta^{(t)} y$  in (12), and  $\text{tr}(P V_K)$  can be calculated using the following proposition:

**Proposition 2** Define  $S = V_K^{-1} - V_K^{-1}X(X^T V_K^{-1}X)^{-1}X^T V_K^{-1}$ .

$$\text{Then } \text{tr}(PV_K) = m - \text{tr}(Z^T SZ[C^{-1}]_{uu}).$$

Note that  $S$  does not need any update and can be computed at once. The demonstration is given in [S8 Appendix](#).

**MM Acceleration.** Similar to EM algorithms, MM algorithms can benefit from accelerating strategies to achieve better rates of convergence (by reducing the number of iterations required to achieve a given precision). Here we combined the MM algorithm with a squared iterative method [33]. Assuming one aims at minimizing a function  $f$  using a MM algorithm, note  $\theta^{(t-2)}$ ,  $\theta^{(t-1)}$  and  $\theta^{(t)}$  the MM estimates obtained at steps  $t-2$ ,  $t-1$  and  $t$ , respectively. At step  $t$  one also computes

$$\begin{aligned} r &= \theta^{(t-1)} - \theta^{(t-2)}, \quad v = \theta^{(t)} - \theta^{(t-2)}, \quad \alpha = -\|r\|_2 / \|v\|_2 \\ \theta^{(t')} &= \theta^{(t-2)} - 2\alpha r + \alpha^2 v \end{aligned}$$

If  $f(\theta^{(t')}) < f(\theta^{(t)})$  then  $\theta^{(t)} \leftarrow \theta^{(t')}$ . The acceleration process is then iterated with  $\theta^{(t+2)}$ .

## Supporting information

**S1 Code. Data and Code for Two variance component analyses available as zip file.**  
(ZIP)

**S2 Code. Data and Code for Four variance component GWAS available as zip file.**  
(ZIP)

**S3 Code. Data and Code for Four variance component VCA available as zip file.**  
(ZIP)

**S1 Appendix. Wald test procedure.**  
(PDF)

**S2 Appendix. Data correction step for the NAM dataset.**  
(PDF)

**S3 Appendix. Table of variance component estimates for model (2).**  
(PDF)

**S4 Appendix. Table of variance component estimates for model (3) with common error variance.**  
(PDF)

**S5 Appendix. Proof of equalities in Eq (12).**  
(PDF)

**S6 Appendix. MM surrogate function for the ReML procedure.**  
(PDF)

**S7 Appendix. Two variance component shortcut for the MM algorithm.**  
(PDF)

**S8 Appendix. Proof of Proposition 2.**  
(PDF)

## Acknowledgments

The authors would like to thank Simon Rio, Aurélien Beugnot and Dimitri Sanchez for their valuable feedback on the R package.

## Author Contributions

**Conceptualization:** Alain Charcosset, Tristan Mary-Huard.

**Formal analysis:** Fabien Laporte.

**Funding acquisition:** Alain Charcosset.

**Methodology:** Fabien Laporte, Tristan Mary-Huard.

**Software:** Fabien Laporte, Tristan Mary-Huard.

**Supervision:** Alain Charcosset, Tristan Mary-Huard.

**Writing – original draft:** Fabien Laporte, Alain Charcosset, Tristan Mary-Huard.

**Writing – review & editing:** Fabien Laporte, Alain Charcosset, Tristan Mary-Huard.

## References

1. Henderson CR. Estimation of variance and covariance components. *Biometrics*. 1953; 9(2):226–252. <https://doi.org/10.2307/3001853>
2. Scheffe H. A “mixed model” for the analysis of variance. *The Annals of Mathematical Statistics*. 1956; p. 23–36. <https://doi.org/10.1214/aoms/1177728348>
3. Gibbons RD, Hedeker D. Applications of mixed-effects models in biostatistics. *Sankhyā: Indian J Stat, Series B*. 2000; p. 70–103.
4. Griffing B. Concept of general and specific combining ability in relation to diallel crossing systems. *Australian journal of biological sciences*. 1956; 9(4):463–493. <https://doi.org/10.1071/BI9560463>
5. Henderson CR. Sire evaluation and genetic trends. *Journal of Animal Science*. 1973; 1973(Symposium):10–41. <https://doi.org/10.1093/ansci/1973.Symposium.10>
6. Kang HM, Zaitlen NA, Wade CM, Kirby A, Heckerman D, Daly MJ, et al. Efficient control of population structure in model organism association mapping. *Genetics*. 2008; 178(3):1709–1723. <https://doi.org/10.1534/genetics.107.080101> PMID: 18385116
7. Lippert C, Listgarten J, Liu Y, Kadie CM, Davidson RI, Heckerman D. FaST linear mixed models for genome-wide association studies. *Nature methods*. 2011; 8(10):833–835. <https://doi.org/10.1038/nmeth.1681> PMID: 21892150
8. Zhou X, Stephens M. Genome-wide efficient mixed-model analysis for association studies. *Nat genet*. 2012; 44(7):821–824. <https://doi.org/10.1038/ng.2310> PMID: 22706312
9. Dandine-Roulland C, Perdy H. Genome-wide data manipulation, association analysis and heritability estimates in R with Gaston 1.5. *Human heredity*. 2017; 83(1):6–6.
10. Yu J, Pressoir G, Briggs WH, Bi IV, Yamasaki M, Doebley JF, et al. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nat Genet*. 2006; 38(2):203–208. <https://doi.org/10.1038/ng1702> PMID: 16380716
11. Gilmour AR, Thompson R, Cullis BR. Average information REML: efficient algorithm for variance parameter estimation in linear mixed models. *Biometrics*. 1995; p. 1440–1450. <https://doi.org/10.2307/2533274>
12. Johnson DL, Thompson R. Restricted maximum likelihood estimation of variance components for univariate animal models using sparse matrix techniques and average information. *Journal of dairy science*. 1995; 78(2):449–456. [https://doi.org/10.3168/jds.S0022-0302\(95\)76654-1](https://doi.org/10.3168/jds.S0022-0302(95)76654-1)
13. Gilmour AR, Gogel BJ, Cullis BR, Welham S, Thompson R. ASReml user guide release 4.1. VSN International Ltd, Hemel Hempstead, UK. 2015;.
14. Speed D, Cai N, Johnson MR, Nejentsev S, Balding DJ. Reevaluation of SNP heritability in complex human traits. *Nat Genet*. 2017; 49(7):986. <https://doi.org/10.1038/ng.3865> PMID: 28530675
15. Hunter DR, Lange K. A tutorial on MM algorithms. *The American Statistician*. 2004; 58(1):30–37. <https://doi.org/10.1198/0003130042836>

16. Zhou H, Hu L, Zhou J, Lange K. MM algorithms for variance components models. *Journal of Computational and Graphical Statistics*. 2019; 28(2):350–361. <https://doi.org/10.1080/10618600.2018.1529601> PMID: 31592195
17. Loh PR, Tucker G, Bulik-Sullivan BK, Vilhjalmsón BJ, Finucane HK, Salem RM, et al. Efficient Bayesian mixed-model analysis increases association power in large cohorts. *Nature genetics*. 2015; 47.3: 284. <https://doi.org/10.1038/ng.3190> PMID: 25642633
18. Zhou X. A unified framework for variance component estimation with summary statistics in genome-wide association studies. *The annals of applied statistics*. 2017; 11(4):2027. <https://doi.org/10.1214/17-AOAS1052> PMID: 29515717
19. Runcie DE, Crawford L. Fast and general-purpose linear mixed models for genome-wide genetics. *PLoS genetics*. 2019; 15.2:e1007978. <https://doi.org/10.1371/journal.pgen.1007978> PMID: 30735486
20. Bates D, Mächler M, Bolker B, Walker S. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*. 2015; 67(1):1–48. <https://doi.org/10.18637/jss.v067.i01>
21. Powell MJ. The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06*, University of Cambridge, Cambridge. 2009;26–46.
22. Kang HM, Sul JH, Service SK, Zaitlen NA, Kong SY, Freimer NB, et al. Variance component model to account for sample structure in genome-wide association studies. *Nature genetics*. 2010; 42(4): 348–354. <https://doi.org/10.1038/ng.548> PMID: 20208533
23. Rincent R, Nicolas S, Bouchet S, Altmann T, Brunel D, Revilla P, et al. Dent and Flint maize diversity panels reveal important genetic potential for increasing biomass production. *Theo Appl Genet*. 2014; 127(11):2313–2331. <https://doi.org/10.1007/s00122-014-2379-7> PMID: 25301321
24. Li J, Ji L. Adjusting multiple testing in multilocus analyses using the eigenvalues of a correlation matrix. *Heredity*. 2005; 95(3):221–227. <https://doi.org/10.1038/sj.hdy.6800717> PMID: 16077740
25. Rincent R, Moreau L, Monod H, Kuhn E, Melchinger AE, Malvar RA, et al. Recovering power in association mapping panels with variable levels of linkage disequilibrium. *Genetics*. 2014; 197(1):375–387. <https://doi.org/10.1534/genetics.113.159731> PMID: 24532779
26. Technow F, Schrag TA, Schipprack W, Bauer E, Simianer H, Melchinger AE. Genome properties and prospects of genomic prediction of hybrid performance in a breeding program of maize. *Genetics*. 2014; 197(4):1343–1355. <https://doi.org/10.1534/genetics.114.165860> PMID: 24850820
27. Giraud H, Bauland C, Falque M, Madur D, Combes V, Jamin P, et al. Reciprocal Genetics: Identifying QTL for General and Specific Combining Abilities in Hybrids Between Multiparental Populations from Two Maize (*Zea mays* L.) Heterotic Groups. *Genetics*. 2017; 207(3):1167–1180. <https://doi.org/10.1534/genetics.117.300305> PMID: 28971957
28. Hadfield JD. MCMC Methods for Multi-Response Generalized Linear Mixed Models: The MCMCglmm R Package. *Journal of Statistical Software*. 2010; 33(2):1–22. <https://doi.org/10.18637/jss.v033.i02>
29. Piepho HP. Analyzing genotype-environment data by mixed models with multiplicative terms. *Biometrics*. 1997; p. 761–766. <https://doi.org/10.2307/2533976>
30. Patterson HD, Thompson R. Recovery of inter-block information when block sizes are unequal. *Biometrika*. 1971; 58(3):545–554. <https://doi.org/10.1093/biomet/58.3.545>
31. Harville DA. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*. 1977; 72(358):320–338. <https://doi.org/10.2307/2286798>
32. Searle SR, Casella G, McCulloch CE. Variance components. vol. 391. John Wiley & Sons; 1992.
33. Varadhan R, Roland C. Simple and globally convergent methods for accelerating the convergence of any EM algorithm. *Scand J Stats*. 2008; 35(2):335–353. <https://doi.org/10.1111/j.1467-9469.2007.00585.x>