Towards Parallel Large-scale Genomic Prediction by Coupling Sparse and Dense Matrix Algebra

Arne De Coninck*, Drosos Kourounis†, Fabio Verbosio†, Olaf Schenk†, Bernard De Baets*, Steven Maenhout‡ and Jan Fostier§
*KERMIT: Research Unit Knowledge-Based Systems, Faculty of Bioscience Engineering, Ghent University, Belgium
Email: arne.deconinck@ugent.be
†Advanced Computing Laboratory, Institute of Computational Science, USI, Lugano, Switzerland
‡Progeno, B-9052 Zwijnaarde, Belgium
§Department of Information Technology (INTEC), Faculty of Engineering and Architecture, Ghent University - iMinds, Belgium

***Abstract*—Genomic prediction for plant breeding requires taking into account environmental effects and variations of genetic effects across environments. The latter can be modelled by estimating the effect of each genetic marker in every possible environmental condition, which leads to a huge amount of effects to be estimated. Nonetheless, the information about these effects is only sparsely present, due to the fact that plants are only tested in a limited number of environmental conditions. In contrast, the genotypes of the plants are a dense source of information and thus the estimation of both types of effects in one single step would require as well dense as sparse matrix formalisms. This paper presents a way to efficiently apply a high performance computing infrastructure for dealing with large-scale genomic prediction settings, relying on the coupling of dense and sparse matrix algebra.**

Keywords*—genomic prediction; distributed computing; sparse matrix algebra; plant breeding;

I. Introduction

Genomic prediction methods most often rely on a linear mixed model framework that models at the same time fixed effects as well as random genetic effects [1]. These genetic effects are modelled by dedicating a small effect to every genetic marker, used to genotype the individuals. The most frequently used SNP arrays consist of 50,000 markers, but even genotypes with 700,000 markers are already available for dairy cattle [2]. DAIRRy-BLUP [3] was developed to use distributed systems for analyzing data sets with a large number of genotyped individuals, applying only dense linear algebra because genetic marker information is mainly dense.

In animal breeding such an analysis method can be very useful, because environmental conditions are more or less constant over the years, due to the fact that animals are mainly held in stables. However, when cultivating plants, the environment and some specific environmental conditions (e.g. soil moisture, solar radiation and air humidity) can have a much stronger impact on the phenotypic trait and effects of genetic markers may vary in different environments. It is thus recommended to also include so-called G × E effects in the analysis [4]. Since every marker is coupled to each environmental condition, modelling these effects increases significantly the problem size. Assuming that plants are tested in 100 different environmental conditions and genotyped for 50,000 markers, the number of G × E effects rises up to 5×10^6. Using only dense matrix algebra to provide estimates of all these effects would require huge distributed systems. Luckily, the information about the G × E effects is usually very sparse and thus a coupled method using sparse and dense linear algebra is presented here for analyzing such large-scale genomic prediction settings.

II. Materials and Methods

A. Statistical background

A linear mixed model for a genomic prediction setting with genetic marker effects and G × E effects can be written as:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Zu} + \mathbf{Td} + \mathbf{e}\,, \tag{1}$$

with $\mathbf{y}$ the vector of n observations, $\mathbf{b}$ the vector of m fixed effects, $\mathbf{u}$ and $\mathbf{d}$ vectors of respectively l and k random effects and $\mathbf{e}$ the residual error. The difference between both random effects is that $\mathbf{u}$, representing the G×E effects, is only sparsely coupled to the observations through the incidence matrix $\mathbf{Z}$ while $\mathbf{d}$, representing the marker effects, is densely coupled to the observations through incidence matrix $\mathbf{T}$. The fixed effects are coupled to the observations through incidence matrix $\mathbf{X}$, whose sparsity is irrelevant because the number of fixed effects is usually a lot smaller than the number of random effects.

Random effects are assumed to be drawn from a normal distribution, since it is expected that only a small part will have a significant effect, while the largest part will only affect the observations in a marginal way. Some other distributions for genetic marker effects have been proposed, but it has been shown that, when no major genes contribute to the trait, these Bayesian methods do not lead to improved prediction accuracies compared to linear predictions when assuming normally distributed marker effects ([5], [6], [7]). Although the exact distributions of the effects and the residual errors are not known, some assumptions are commonly made on the variance structure based on prior information:

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{d}\\ \mathbf{e}\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \begin{bmatrix}\phi\mathbf{E} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \gamma\mathbf{G} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}_n\end{bmatrix}\right). \tag{2}$$

$\mathbf{E}$ and $\mathbf{G}$ are constant square matrices of respective dimensions l and k. These are usually sparse since every random effect

is only correlated with a small number of other effects. It can thus be concluded that the observations are also normally distributed:

$$\begin{aligned} &\mathbf{y} \sim \mathcal{N}\left(\mathbf{Xb}, \mathbf{V}\right), \\ &\text{with } \mathbf{V} = \sigma^2 \left(\mathbf{I}_n + \phi \mathbf{ZEZ}^T + \gamma \mathbf{TGT}^T\right). \end{aligned} \tag{3}$$

When little information is available about the data, $\mathbf{E}$ and $\mathbf{G}$ may be simplified as being identity matrices, which is then also referred to as ridge regression BLUP (RR-BLUP) [8].

B. Estimating the effects

The Best Linear Unbiased Estimates and Predictions (BLUE and BLUP) of the fixed and random effects are linear estimates or predictions that minimize the mean squared error and exhibit no bias. These are also the solutions of the so-called Mixed Model Equations (MME) [9]:

$$\begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{Z} & \mathbf{X}^T\mathbf{T} \\ \mathbf{Z}^T\mathbf{X} & \mathbf{Z}^T\mathbf{Z} + \frac{1}{\phi}\mathbf{E}^{-1} & \mathbf{Z}^T\mathbf{T} \\ \mathbf{T}^T\mathbf{X} & \mathbf{T}^T\mathbf{Z} & \mathbf{T}^T\mathbf{T} + \frac{1}{\gamma}\mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \\ \hat{\mathbf{d}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{y} \\ \mathbf{Z}^T\mathbf{y} \\ \mathbf{T}^T\mathbf{y} \end{bmatrix}. \tag{4}$$

The coefficient matrix $\mathbf{C}$ of this equation is a square matrix of size $(m + l + k)$ and is made up of both sparse and dense blocks:

$$\mathbf{C} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{bmatrix}, \tag{5}$$

with

$$\mathbf{A} = \begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{Z} \\ \mathbf{X}^T\mathbf{Z} & \mathbf{Z}^T\mathbf{Z} + \frac{1}{\phi}\mathbf{E}^{-1} \end{bmatrix}, \tag{6}$$

$$\mathbf{D} = \mathbf{T}^T\mathbf{T} + \frac{1}{\gamma}\mathbf{G}^{-1}, \tag{7}$$

$$\mathbf{B}^T = \begin{bmatrix} \mathbf{T}^T\mathbf{X} & \mathbf{T}^T\mathbf{Z} \end{bmatrix}, \tag{8}$$

where $\mathbf{A}$ and $\mathbf{B}$ are sparse matrices and $\mathbf{D}$ is a dense matrix.

For solving matrix equation (4) it is thus appropriate to apply optimized routines for as well sparse as dense matrices. A step-by-step blockwise solution of this matrix equation boils down to:

1) Solve $\mathbf{A}\mathbf{w}_\mathbf{A} = \begin{bmatrix} \mathbf{X}^T\mathbf{y} \\ \mathbf{Z}^T\mathbf{y} \end{bmatrix}$ for $\mathbf{w}_\mathbf{A}$.
2) Calculate the Schur complement of $\mathbf{A}$: $\mathbf{S} = \mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$.
3) Solve $\mathbf{S}\hat{\mathbf{d}} = \mathbf{T}^T\mathbf{y} - \mathbf{B}^T\mathbf{w}_\mathbf{A}$ for $\hat{\mathbf{d}}$.
4) Solve $\mathbf{A}\begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{y} \\ \mathbf{Z}^T\mathbf{y} \end{bmatrix} - \mathbf{B}\hat{\mathbf{d}}$ for $\begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{u}} \end{bmatrix}$.

It can easily be seen that steps 1 and 4 only require methods optimized for sparse matrices, while steps 2 and 3 also involve dense matrices.

C. Estimating the variance components

In Eq. (2) the structure of the variance matrices was defined based on 3 variance components ϕ, γ and σ^2. These variance components need to be estimated based on the data, which is implemented using the Average Information Restricted Maximum Likelihood procedure (AI-REML) [10]. AI-REML is an iterative procedure where the variance components are updated every iteration as follows:

$$\boldsymbol{\kappa}_{n+1} = \boldsymbol{\kappa}_n - \mathbf{H}_{\mathbf{AI}_n}^{-1} \nabla l_{\text{REML}}(\boldsymbol{\kappa}_n), \tag{9}$$

where $\boldsymbol{\kappa}_n$ is the vector of variance components at iteration n (here $\boldsymbol{\kappa} = \left(\sigma^2, \phi, \gamma\right)$), $\mathbf{H}_{\mathbf{AI}_n}$ is the AI update matrix at iteration n and $\nabla l_{\text{REML}}(\boldsymbol{\kappa}_n)$ is the gradient of the REML log-likelihood with respect to $\boldsymbol{\kappa}$ evaluated for $\boldsymbol{\kappa}_n$.

The REML log-likelihood function can be written as:

$$\begin{aligned} l_{\text{REML}}(\sigma^2, \phi, \gamma) = -\frac{1}{2}\Big((n-m)\log\sigma^2 + l\log\phi \\ + k\log\gamma + \log|\mathbf{E}| + \log|\mathbf{G}| \\ + \log|\mathbf{C}| + \frac{1}{\sigma^2}\mathbf{y}^T\mathbf{P}\mathbf{y}\Big), \\ \text{with } \mathbf{P} = \mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}\left(\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{V}^{-1}. \end{aligned} \tag{10}$$

When ϕ and γ are known, an analytic solution for the maximization of this likelihood function with respect to σ^2 can be found, however, for maximization in function of ϕ and γ, we have to resort to the iterative AI-REML technique. For this iterative technique an evaluation of the first partial derivative of l_{REML} with respect to ϕ and γ is necessary:

$$\frac{\partial l_{REML}}{\partial \phi} = -\frac{1}{2}\left(\frac{l}{\phi} - \frac{\text{tr}(\mathbf{C}_{(2,2)}^{-1}\mathbf{E}^{-1})}{\phi^2} - \frac{\hat{\mathbf{u}}^T\mathbf{E}^{-1}\hat{\mathbf{u}}}{\sigma_e^2\phi^2}\right) \tag{11}$$

$$\frac{\partial l_{REML}}{\partial \gamma} = -\frac{1}{2}\left(\frac{k}{\gamma} - \frac{\text{tr}(\mathbf{C}_{(3,3)}^{-1}\mathbf{G}^{-1})}{\gamma^2} - \frac{\hat{\mathbf{d}}^T\mathbf{G}^{-1}\hat{\mathbf{d}}}{\sigma_e^2\gamma^2}\right), \tag{12}$$

with $\mathbf{C}_{(2,2)}^{-1}$ and $\mathbf{C}_{(3,3)}^{-1}$ the blocks of the inverse of $\mathbf{C}$ corresponding with the blocks in Eq. (4) containing respectively $\mathbf{Z}^T\mathbf{Z}$ and $\mathbf{T}^T\mathbf{T}$. For evaluating the traces in these equations, the only elements of the inverse of $\mathbf{C}$ to be calculated are those in the aforementioned blocks corresponding to non-zero elements of $\mathbf{E}^{-1}$ and $\mathbf{G}^{-1}$. However, because $\mathbf{T}^T\mathbf{T}$ is completely dense, calculating a sparse subset of $\mathbf{C}_{(3,3)}^{-1}$ is not efficient and thus the entire dense inverse will be calculated. It can also often be assumed that $\mathbf{E}$ is a diagonal matrix, which reduces the problem to calculating only the diagonal elements of $\mathbf{C}_{(2,2)}^{-1}$.

A stepwise approach for finding the required inverse elements of $\mathbf{C}_{(2,2)}^{-1}$ and $\mathbf{C}_{(3,3)}^{-1}$ is presented below:

1) Calculate the Schur complement of $\mathbf{A}$: $\mathbf{S} = \mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$.
2) Compute the inverse of $\mathbf{S}$ ($\mathbf{C}_{(3,3)}^{-1} = \mathbf{S}^{-1}$).
3) Compute a selected inverse of sparse matrix $\mathbf{A}$.
4) Solve $\mathbf{AY} = \mathbf{B}$ for $\mathbf{Y}$.
5) The required elements of $\mathbf{C}_{(2,2)}^{-1}$ can be calculated as: $\mathbf{C}_{(2,2)_{i,j}}^{-1} = \mathbf{A}_{i,j}^{-1} + \mathbf{Y}_i\mathbf{S}^{-1}\mathbf{Y}_j^T$, with $\mathbf{Y}_i$ the i-th row of $\mathbf{Y}$ and $\mathbf{Y}_j^T$ the j-th column of $\mathbf{Y}^T$.

III. Parallel implementation

The dimensions of the dense submatrix $\mathbf{D}$ depend on the number of genetic markers included in the analysis. Currently the number of markers for crop plants varies between 5,000 and 100,000 [11], which means that the dense submatrix can

consume up to 80 GB of memory when its elements are being stored in 64-bit floating-point format. Therefore, a critical step in efficient parallelization of this method is to apply distributed computing techniques such that matrix $\mathbf{D}$ is distributed across the private memories of all designated processes. To perform most common algebraic operations on such a distributed matrix, the standard libraries PBLAS [12] and ScaLAPACK [13] are employed, for which vendor-optimized versions are available.

Sparse matrices are stored in Compressed Sparse Row (CSR) format, which consists of an array with the values of the non-zero elements (floating-point), an array with the column indices of the non-zero elements (integer) and an array with indices indicating the start of a new row in the two other arrays (integer). The PARDISO library ([14], [15], [16]) was employed for solving sparse matrix equations and calculating a selected inverse of matrix $\mathbf{A}$, which only consists of the elements of the inverse corresponding to non-zero elements in the factorization of $\mathbf{A}$ [17]. This library can perform these operations multi-threaded on Shared Memory Processors (SMP), but not yet on a distributed system. To utilize this optimized library on a sparse matrix with a large dense submatrix, an algorithm was developed to couple PARDISO with distributed computing techniques, used for the algebraic operations on the dense submatrix.

A. Calculating the Schur complement of $\mathbf{A}$

The Schur complement $\mathbf{S}$ of $\mathbf{A}$ is needed for as well the solution of the MME (step 2 in Section II.B) as for calculating elements of the inverse of $\mathbf{C}$ (step 1 in Section II.C). Both sparse and dense matrices are involved in the computation of this Schur complement and moreover, the dense matrix is stored in a distributed fashion, while the sparse matrices are not.

The parallelization of this task is straightforward, because the calculation of $\mathbf{S}$ can be performed on each block $\mathbf{S}_{(i,j)}$ of $\mathbf{S}$ independently:

$$\mathbf{S}_{(i,j)} = \mathbf{D}_{(i,j)} - \mathbf{B}_i^T \mathbf{A}^{-1} \mathbf{B}_j \,, \tag{13}$$

where $\mathbf{S}_{(i,j)}$ and $\mathbf{D}_{(i,j)}$ are the (i,j)-th blocks of respectively $\mathbf{S}$ and $\mathbf{D}$, $\mathbf{B}_i^T$ are the rows of $\mathbf{B}^T$ corresponding to the rows of the (i,j)-th block of $\mathbf{D}$ and $\mathbf{B}_j$ are the columns of $\mathbf{B}$ corresponding to the columns of the (i,j)-th block of $\mathbf{D}$. The process possessing block (i,j) of $\mathbf{D}$ thus only needs some rows of sparse matrix $\mathbf{B}^T$ and some columns of $\mathbf{B}$. Only the root process assembles $\mathbf{B}^T$ entirely and this process thus has to send the desired rows of $\mathbf{B}^T$ and columns of $\mathbf{B}$ to the other processes using MPI [18]. Each process then still has to construct sparse matrix $\mathbf{A}$ from $\mathbf{X}$ and $\mathbf{Z}$, which are always stored in CSR format.

For each submatrix of $\mathbf{D}$ at the disposal of the process, the sparse matrix equation $\mathbf{A}\mathbf{Y}_j = \mathbf{B}_j$ is solved for $\mathbf{Y}_j$ with PARDISO and the product $\mathbf{B}_i^T \mathbf{Y}_j$, performed by PBLAS and thus stored as a dense matrix, is directly subtracted from $\mathbf{D}_{(i,j)}$. In this way, $\mathbf{D}$ is overwritten by $\mathbf{S}$, keeping it stored in a distributed way.

B. Calculating elements of the inverse of $\mathbf{C}$

The dense matrix equation in step 3 of Section II.B is directly solved by ScaLAPACK using the factorization of $\mathbf{S}$. ScaLAPACK can also compute the complete inverse of a distributed matrix using the factorization of this matrix, which is needed in step 2 of Section II.C.

For computing the selected inverse of a sparse matrix, it is not yet possible to utilize processes that do not share memory. Therefore, only the root process calculates the selected inverse of $\mathbf{A}$. In the previous section it was shown that each process computes the solution of $\mathbf{A}\mathbf{Y}_j = \mathbf{B}_j$ for $\mathbf{Y}_j$. This solution remains in memory and is identical for processes in the same column of the process grid. Additionally, it can be seen as if the entire solution $\mathbf{Y}$ of the matrix equation $\mathbf{A}\mathbf{Y} = \mathbf{B}$ is stored in a distributed way across the processes in the first row of the process grid. The matrix product $\mathbf{S}^{-1}\mathbf{Y}_j^T$ for step 5 in Section II.C can thus be performed by PBLAS and the result will be distributed across the processes of the first row of the process grid. Finally, a dot product between the distributed vectors $\mathbf{Y}_i$ and $\mathbf{S}^{-1}\mathbf{Y}_j^T$ can be calculated using PBLAS and the root process then adds this dot product to the corresponding element of $\mathbf{A}^{-1}$ to return the desired element of $\mathbf{C}^{-1}_{(2,2)}$.

IV. Results

A prototype of the parallel implementation was tested on a small data set from CIMMYT's[1] Global Wheat Program, which contains information on 599 wheat lines whose grain yield was evaluated in four environments. Genotypic information was available for 1279 markers using Diversity Array Technology. This data set is also publicly available with the BLR package of R [19] and was previously analyzed using single-environment [20] and multi-environment [21] models. Our model was fit to the data using four fixed environmental effects ($\mathbf{b}$), 1279 random marker effects ($\mathbf{d}$) and 5116 random marker-by-environment interaction effects ($\mathbf{u}$). Tenfold cross-validation was performed with the training sets missing all information of the lines in the test sets and the predictive ability of the model was evaluated using the correlation between predicted and observed values for the phenotypes in each environment. The tenfold cross-validation yielded a mean correlation of 0.439 over all environments, which is close to values obtained in [21].

Although this data set is too small to really observe the benefits of the parallelized approach, its scaling behavior was investigated for an increasing number of MPI processes. Therefore the entire data set was analyzed on a cluster consisting of nodes with 16 CPU cores (Dual Intel Xeon CPU E5-2670) and 64 GB RAM per node, which are linked through an FDR Inifiniband network. For these tests, one MPI process was mapped to each node, and on each node, all cores were employed using OpenMP. The total analysis of the data set required 442 seconds on 1 node, 108 seconds on 4 nodes and 64 seconds on 9 nodes. This already shows that some scaling was obtained with the prototype of the parallelized implementation, but further investigation is needed to check whether additional speedup can be gained.

[1] International Maize and Wheat Improvement Center, Mexico

V. Conclusion

Current genomic prediction settings can include information from ten thousands of animals or plants for whom genotypes are available with up to 100,000 markers. The analysis of such large-scale genomic datasets requires efficient usage of high performance computing infrastructures. In plant breeding the global environment and some specific environmental factors can play an important role. Phenotypes not only depend directly on these environmental effects, but marker effects may vary under the influence of certain environmental conditions. The so-called $\mathrm{G} \times \mathrm{E}$ effects are thus included in the analysis, which can lead to a dramatic increase in effects to be estimated Luckily, information on $\mathrm{G} \times \mathrm{E}$ effects is sparse, because each plant is only tested in a few environments and so sparse matrix formalisms can be employed to minimize memory requirements when processing large-scale data sets.

This paper presents a parallelized methodology for analyzing genomic prediction settings with $\mathrm{G} \times \mathrm{E}$ and genetic marker effects in a single step. The part of the MME arising from the genetic marker effects is stored in a distributed way and most algebraic operations on this dense matrix are performed by PBLAS and ScaLAPACK. To increase memory efficiency, the other parts of the MME are stored as sparse matrices, on which algebraic operations are performed by employing optimized routines from the PARDISO library. First tests with a prototype of this coupled parallel implementation have shown that prediction accuracy is comparable to other implementations and that it scales well when employing multiple compute nodes.

Acknowledgment

Some of the authors (ADC, BDB, JF) would like to thank the Ghent University Multidisciplinary Research Partnership Bioinformatics: From Nucleotides to Networks for funding. The computational resources (STEVIN Supercomputer Infrastructure) and services used in this work were kindly provided by Ghent University, the Flemish Supercomputer Center (VSC), the Hercules Foundation and the Flemish Government department EWI.

References

[1] T. H. E. Meuwissen, B. J. Hayes, and M. E. Goddard, "Prediction of total genetic value ysing genome-wide dense marker maps," *Genetics*, vol. 157, no. 4, pp. 1819–1829, 2001. [Online]. Available: http://www.genetics.org/content/157/4/1819.abstract

[2] J. B. Cole, S. Newman, F. Foertter, I. Aguilar, and M. Coffey, "Breeding and genetics symposium: Really big data: Processing and analysis of very large data sets," *Journal of Animal Science*, vol. 90, no. 3, pp. 723–733, 2012. [Online]. Available: http://www.journalofanimalscience.org/content/90/3/723.abstract

[3] A. De Coninck, J. Fostier, S. Maenhout, and B. De Baets, "DAIRRy-BLUP: A high-performance computing approach to genomic prediction," *Genetics*, vol. 197, no. 3, pp. 813–822, 2014. [Online]. Available: http://www.genetics.org/content/197/3/813.abstract

[4] T. Schulz-Streeck, J. O. Ogutu, A. Gordillo, Z. Karaman, C. Knaak, and H.-P. Piepho, "Genomic selection allowing for marker-by-environment interaction," *Plant Breeding*, vol. 132, no. 6, pp. 532–538, 2013. [Online]. Available: http://dx.doi.org/10.1111/pbr.12105

[5] B. Hayes, P. Bowman, A. Chamberlain, and M. Goddard, "Invited review: Genomic selection in dairy cattle: Progress and challenges," *Journal of Dairy Science*, vol. 92, no. 2, pp. 433–443, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022030209703479

[6] H. D. Daetwyler, M. P. L. Calus, R. Pong-Wong, G. de los Campos, and J. M. Hickey, "Genomic prediction in animals and plants: simulation of data, validation, reporting, and benchmarking," *Genetics*, vol. 193, no. 2, pp. 347–365, 2013. [Online]. Available: http://www.genetics.org/content/193/2/347.abstract

[7] A. Legarra, C. Robert-Grani, P. Croiseau, F. Guillaume, and S. Fritz, "Improved lasso for genomic selection," *Genetics Research*, vol. 93, pp. 77–87, 2 2011. [Online]. Available: http://journals.cambridge.org/article_S0016672310000534

[8] H.-P. Piepho, "Ridge regression and extensions for genomewide selection in maize," *Crop Science*, vol. 49, no. 4, pp. 1165–1176, 2009.

[9] C. R. Henderson, "Selection index and expected genetic advance," *Statistical genetics and plant breeding*, vol. 982, pp. 141–163, 1963.

[10] A. R. Gilmour, R. Thompson, and B. R. Cullis, "Average information REML: an efficient algorithm for variance parameter estimation in linear mixed models," *Biometrics*, pp. 1440–1450, 1995.

[11] M. Ganal, A. Polley, E.-M. Graner, J. Plieske, R. Wieseke, H. Luerssen, and G. Durstewitz, "Large SNP arrays for genotyping in crop plants," *Journal of Biosciences*, vol. 37, no. 5, pp. 821–828, 2012. [Online]. Available: http://dx.doi.org/10.1007/s12038-012-9225-3

[12] J. Choi, J. Dongarra, S. Ostrouchov, A. Petitet, D. W. Walker, and R. C. Whaley, "A proposal for a set of parallel basic linear algebra subprograms," in *Proceedings of the Second International Workshop on Applied Parallel Computing, Computations in Physics, Chemistry and Engineering Science*, ser. PARA '95. London, UK, UK: Springer-Verlag, 1996, pp. 107–114. [Online]. Available: http://dl.acm.org/citation.cfm?id=645779.666023

[13] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*. Society for Industrial and Applied Mathematics, 1997. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898719642

[14] A. Kuzmin, M. Luisier, and O. Schenk, "Fast methods for computing selected elements of the greens function in massively parallel nanoelectronic device simulations," in *Euro-Par 2013 Parallel Processing*, ser. Lecture Notes in Computer Science, F. Wolf, B. Mohr, and D. Mey, Eds. Springer Berlin Heidelberg, 2013, vol. 8097, pp. 533–544. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40047-6_54

[15] O. Schenk, M. Bollhöfer, and R. A. Römer, "On large-scale diagonalization techniques for the anderson model of localization," *SIAM Rev.*, vol. 50, no. 1, pp. 91–112, Feb. 2008. [Online]. Available: http://dx.doi.org/10.1137/070707002

[16] O. Schenk, A. Wchter, and M. Hagemann, "Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization," *Computational Optimization and Applications*, vol. 36, no. 2-3, pp. 321–341, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10589-006-9003-y

[17] K. Takahashi, J. Fagan, and M. Chin, "Formation of a sparse bus impedance matrix and its application to short circuit study," *8th Power Industry Computer Application Conference Proceedings*, p. 63, 1973.

[18] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI-The Complete Reference, Volume 1: The MPI Core*, 2nd ed. Cambridge, MA, USA: MIT Press, 1998.

[19] G. de los Campos and P. Prez, "BLR: Bayesian linear regression. version 1.3," 2012. [Online]. Available: http://cran.r-project.org/web/packages/BLR (verified 20 Nov. 2014)

[20] J. Crossa, G. d. l. Campos, P. Prez, D. Gianola, J. Burgueo, J. L. Araus, D. Makumbi, R. P. Singh, S. Dreisigacker, J. Yan, V. Arief, M. Banziger, and H.-J. Braun, "Prediction of genetic values of quantitative traits in plant breeding using pedigree and molecular markers," *Genetics*, vol. 186, no. 2, pp. 713–724, 2010. [Online]. Available: http://www.genetics.org/content/186/2/713.abstract

[21] J. Burgueño, G. de los Campos, K. Weigel, and J. Crossa, "Genomic prediction of breeding values when modeling genotype× environment interaction using pedigree and dense molecular markers," *Crop Science*, vol. 52, no. 2, pp. 707–719, 2012.