Efficient Computation of Base Generation Allele Frequencies

M.N. Aldridge[1], J. Vandenplas[1], and M.P.L. Calus[1]
[1] Wageningen University & Research, Animal Breeding and Genomics, 6700AH Wageningen, The Netherlands

Abstract

Several aspects of genomic prediction require use of allele frequencies that ideally reflect the base generation of the available pedigree. This includes computation of model-based reliabilities of direct genomic values (DGV) in the context of multi-step genomic evaluations, computation of genomic relationships to be used in single-step GBLUP, and computation of relationships among metafounders. In many cases, the allele frequencies computed from the currently genotyped population are used instead, motivated by the observation that computation of base generation allele frequencies is time consuming. Our aim was to compare the efficiency and accuracy of different methods to compute base generation allele frequencies. The first method employed the gene content method, by running a BLUP on the SNP genotypes and considering a heritability of 0.99. Either a univariate BLUP run for each SNP, or a multiple-trait BLUP run for several SNPs was performed, considering zero genetic correlations among the SNPs. The second method employed a general least squares estimator that is equivalent to the first method, albeit that it does not consider a residual variance. First analyses on simulated data without selection, missing genotypes or genotype errors in the data showed that the second method is superior in both accuracy and efficiency, but only if the inverse of the A matrix was computed using imputation on the fly. The implementation of the second method required less than two minutes to compute base generation allele frequencies for 1 670 SNPs based on 100 078 genotyped animals, and a total pedigree of 325 266 animals. Subsequent analyses with datasets simulating selection, missing genotypes and genotyping errors, that are closer to data used in practice, supported the results that the second method is more efficient and accurate.

Key words: base generation allele frequencies, genomic prediction

Introduction

The allele frequencies of the base generation in the pedigree are required for several aspects of genomic prediction. This includes computation of model-based reliabilities of direct genomic values (DGV) in the context of multi-step genomic evaluations, computation of genomic relationships to be used in single-step GBLUP, and computation of relationships among metafounders.

There is a need to find a computationally efficient and accurate method of estimating base generation allele frequencies rather than using the allele frequencies computed from the currently genotyped population. Two methods have been proposed to achieve this. The first method was a general least squares estimator (GLS) for each SNP (McPeek *et al.*, 2004; Garcia-Baccino *et al.,* 2017). The second method was best linear unbiased predictions (BLUP) for each SNP with a genetic correlation of zero and a heritability of 0.99 (Gengler *et al.*, 2007). The aim of this study was to test these methods and to determine which would be the most efficient and accurate.

Materials and Methods

Data Simulation

Datasets for a typical dairy population were simulated using the software QMSim (Sargolzaei and Schenkel, 2009). Each dataset had a historical population with 100 000 animals decreasing to 500 over 2 000 generations. Following the slow decline in population size, was a rapid increase to 25 000 animals over 10 generations to expand the population, while maintaining linkage disequilibrium.

A founder population was randomly selected from the last historical generation. The founder population was the base generation that the calculations for allele frequency were attempted. The founder population included 30 males and 24 970 females. The population structure of males to females was maintained across the following generations, which had a consistent effective population size of 120.

Four datasets were simulated using the above historic and founder population structures. All datasets had 12 generations proceeding the base generation. Generations 9 to 12 were fully genotyped. The pedigree included 325 266 animals with 100 078 animals genotyped. Genotyping included 1 670 SNPs, randomly positioned across the genome with a mutation rate of 2.5e^{-5}. QMSim provided frequencies of markers for each generation but only the founder generation was necessary, which were used to determine the accuracy of calculated allele frequencies.

The first dataset (Base Simulation) was used to determine computational efficiency. It had no selection, random mating and random culling. The genome consisted of 10 chromosomes, with 167 SNPs randomly positioned on each chromosome.

The second dataset (Simulation with selection) had selection and culling based on EBVs, with random matings and only a single chromosome, thereby decreasing variation.

The third dataset (Pedigree Errors) was a direct copy of Simulation with selection, but 25% of sires in generations 1 to 5 were replaced with an unknown sire. This was to simulate a scenario where animals erroneously may be considered as founder animals, only because they have unknown ancestry.

The fourth dataset (Genotyping Errors) was a replicate of Simulation with selection but with a random genotyping error rate of 0.02.

Calculating allele frequencies

To determine which method of calculating the base generation allele frequency was most efficient, the Base Simulation was used. The other three datasets were used to determine the effects that selection and variation, pedigree errors, and genotyping errors have on efficiency and accuracy of the tested methods.

Observed allele frequencies of genotyped animals

Two implementations of computing genotype frequencies were made. First, observed allele frequencies were calculated using all genotyped animals. Second, observed allele frequencies were calculated only using animals in the first genotyped generation (most closely related genotyped generation to the base generation). Both implementations were done using Python version 3.6.3.

Best Linear Unbiased Prediction

Using MiXBLUP (Ten Napel *et al.,* 2017), a series of BLUPs were run on the SNP genotypes (McPeek *et al.,* 2004; Gengler *et al.,* 2007). For each SNP the heritability was set at 0.99, with a genetic correlation between SNPs of zero. The series of SNPs included in BLUP runs ranged between a single SNP up to 60 SNPs. Each model had a convergence criteria of 1.0*10^{-12}. For 1 670 SNPs, 27 runs of 60 SNPs plus an additional run of 50 SNPs were run in parallel. The base generation allele frequency was estimated for each SNP as $\hat{\mu}/2$ where $\hat{\mu}$ was the estimate of the general mean of the model.

General least squares estimator

The equivalent GLS estimator to BLUP (McPeek *et al.,* 2004; Garcia-Baccino *et al.,* 2017) for the *i*-th SNP was:

$$\hat{\mu}_i = (\mathbf{1}'\mathbf{A}_{\mathbf{22}}^{-\mathbf{1}}\mathbf{1})^{-\mathbf{1}}\mathbf{1}'\mathbf{A}_{\mathbf{22}}^{-\mathbf{1}}\mathbf{Z}_i$$

where $\mathbf{1}$ is a vector of ones, $\mathbf{A}_{\mathbf{22}}^{-\mathbf{1}}$ is the inverse pedigree relationship matrix of genotyped animals and $\mathbf{Z}$ a matrix of genotypes coded as 0, 1, 2. Two approaches implemented this method differently.

The first implementation was similar to that of Strandén *et al.* (2017) which used the approach of McPeek *et al.* (2004), to estimate base generation allele frequencies. This was done by writing a Fortran program (allelefreq), whereby $\mathbf{A}_{\mathbf{22}}^{-\mathbf{1}}\mathbf{1}$ is computed as a multiplication of sparse matrices by a vector:

$$\mathbf{A}_{22}^{-1}\mathbf{1} = \left(\mathbf{A}^{22} - \mathbf{A}^{21}\left(\mathbf{A}^{11}\right)^{-1}\mathbf{A}^{12}\right)\mathbf{1}$$

where the product $\left(\mathbf{A}^{11}\right)^{-1}\mathbf{A}^{12}\mathbf{1} = \left(\mathbf{A}^{11}\right)^{-1}\mathbf{v}$ was solved as $\mathbf{A}^{11}\mathbf{x} = \mathbf{v}$ using Intel MKL-PARDISO (Schenk *et al.,* 2001).

Due to minor alleles close to being fixed with frequencies <0.001, there were instances where the estimated allele frequency of the base generation were outside of the parameter space. This was addressed by swapping the allele coding and using only estimates within the parameter space from the two runs of allelefreq.

The second implementation calculates the direct inverse of $\mathbf{A}_{22}$, which was computed and written using Calc_grm (Calus and Vandenplas, 2016). Using the same equivalent GLS estimator as allelefreq the base generation allele frequency was estimated. This was done to determine if it yielded the same results as the first method, and to evaluate the computational advantage of the first method.

Statistical Analysis

Computational efficiency

For each method the required processing time, observed wall clock time and memory was analysed. The reported processing time and Random Access Memory (RAM) was the maximum allocation required for the analysis as recorded by the computer. The wall clock time, was the observed time seen for the process to start and end, for multiple CPUs per task or running tasks in parallel. As such, the time required for frequency of all genotyped animals or the first genotyped generation included; reading the pedigree and genotype data for relevant generations, making the frequency calculation and writing the results.

The analysis of MiXBLUP has been reported for both the full processing time including the running of all 28 MiXBLUP runs, plus the time required to read the MiXBLUP results, calculate base generation allele frequency and write the final results. The MiXBLUP runs included 60 SNPs as the fewest number of required runs but single runs of 1 to 60 SNPs (5 SNP increments) were used to determine the computationally most efficient number of SNPs to be included. Increasing the number of correlated SNPs reduced reading the pedigree to once per iteration.

The time and memory reported for two threads of allelefreq, included both runs with and without swapping the allele coding. Which included; reading the pedigree, inbreeding coefficients and genotyped data, computing the base generation allele frequencies and writing the results.

Where Calc_grm was used time and memory reported includes the computation of the $\boldsymbol{A}_{22}^{-1}$ matrix, reading the corresponding results, reading the pedigree and genotype data, computation of the base generation allele frequency and writing the results.

The computational efficiency of each method was compared to a number of data structures and errors that occur in more realist scenarios.

All computations were run on a high performance cluster (HPC). The HPC was designed with 48 nodes: 16 cores, 64 GB memory, Intel Xeon, and 2.2 GHz. For the computation of $\mathbf{A}_{22}^{-1}$ with Calc_grm, one of two fat nodes with: 64 cores, 1 TB memory, AMD Opteron, and 2.3 GHz was used, with 16 threads. The use of the HPC cluster has been made possible by CAT-AgroFood (Shared Research Facilities Wageningen UR).

Accuracy of estimates

Known allele frequencies of the base generation provided by QMSim during data simulation were used to compare the accuracy of each method. To make the comparison between methods, the known frequency from QMSim and estimated allele frequency correlations were calculated using R version 3.4.0 (R Core Team. 2017). The accuracy of estimates for each method was compared to a number of more realistic data structures and errors.

Results and Discussion

Computational efficiency

It was shown that estimating base generation allele frequency can be estimated efficiently. The most computationally efficient method for calculating base generation allele frequency was with allelefreq, using a sparse computation of $\mathbf{A}_{22}^{-1}\mathbf{1}$. The full processing time required was just 1 minute and 28 seconds, and required 2.6 GB RAM, which was significantly lower compared to the other methods reported in Table 1. If only one of the allelefreq runs was used (without allele coding swapped) the time was reduced to 45 seconds.

Table 1. Computational requirements to complete the full process of each method for the Base Simulation.

Method	Process time	Wall clock time	Random Access Memory
All animals genotyped	0-00:03:44	0-00:03:44	7.8 GB
First animals genotyped	0-00:01:19	0-00:01:19	1.6 GB
29 MiXBLUPs	0-13:42:17	0-00:34:57	49.0 GB
1 MiXBLUP	0-00:29:50	0-00:29:50	1.8 GB
Allelefreq	0-00:01:28	0-00:00:52	2.6 GB
Calc_grm	50-20:12:16	1-05:42:00	165.9 GB

The program allelefreq was required to be run twice for each data simulation whereby the second run swaps the allele coding. The required time could have been greatly reduced if either the coding did not requiring swapping, or if the original and swapped run were run together so that the genotypes are only read once which is the most demanding process.

For the GLS method with full $\mathbf{A}_{22}^{-1}$ computed using Calc_grm, writing the full matrix with and then reading the matrix, was the most demanding process. The full process required over 50 days of processing time, and required over one day of actual observed wall clock time. This was the same reason for why Strandén *et al.* (2017) proposed the previous method with the imputed matrix. The efficiency could have been greatly improved if the GLS estimation was done within Calc_grm, to avoid writing the matrix to file. Due to the time and memory inefficiency this method was not recommended unless the $\mathbf{A}_{22}^{-1}$ was going to be calculated for other processes.

The results for computational efficiency for the methods based on GLS were presented for the Base Simulation. When the Simulations with selection, Pedigree errors, and Genotyping errors were used, the required processing time and memory was not significantly different. The reason there was no difference is because the majority of the computational requirements is reading the data not the calculation of allele frequency.

Running BLUPs for all 1 670 SNPs was inefficient in total processing time at over 13 hours, however the observed wall clock time when run in parallel was reduced to under 35 minutes. For the Simulation with selection, the required processing time increased to over 28 hours, with an observed wall clock time of over one hour. This was due to convergence issues with each MiXBLUP run requiring on average 330 iterations for the Base Simulations and 840 for the Simulation with selection. The results presented for the BLUP analysis used MiXBLUP, where 60 SNPs were run simultaneously, and genetic correlations set to zero. The 60 SNPs were used for convenience.

The most demanding process for the BLUP method was for solving the mixed model equations. Computationally the most time efficient number of simultaneously included SNPs was between 10 and 20 SNPs (Figure 1). The efficiency could have been improved by reducing the number of correlated SNPs.

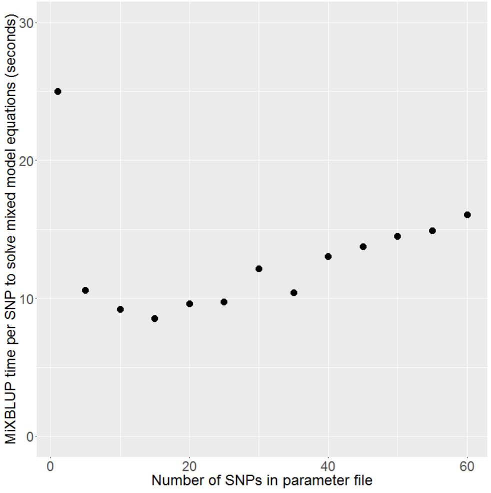

Figure 1. Average time per SNP for MiXBLUP to start and end, solving mixed model equations, with the Base Simulation dataset.

Using the allele frequency of the first fully genotyped generation to estimate the allele frequency, would have been the most computational efficient method. It required just 79 seconds and 1.6 GB of processing time and RAM to complete. It could have been further optimized by using a compiled language rather than Python. It however, was considered the least appropriate method as the accuracy of the estimates make it unsuitable as a method of estimating base generation allele frequency.

It should also be noted that estimating the base generation allele frequency does not need to be performed before each routine evaluation. Assuming the number of genotyped animals is large and representative enough. Similar to how variance components are not estimated for every evaluation, but are updated as necessary.

Accuracy of computation

The correlation between known allele frequency of the base generation and the estimated allele frequency for each method have been presented in Table 2. The correlations were used to gauge the accuracy of the estimated allele frequency.

Table 2. Correlation (± s.e.) between the known base generation allele frequency and estimated allele frequency.

Method	Base Simulation	Simulation with selection
All animals genotyped	0.99 ± 0.01	0.87 ± 0.01
First animals genotyped	0.99 ± 0.01	0.88 ± 0.01
MiXBLUP	0.99 ± 0.01	0.96 ± 0.01
Allelefreq	0.99 ± 0.01	0.97 ± 0.01
Calc_grm	0.99 ± 0.01	0.97 ± 0.01

Using the allele frequency of the most recent generation was a good estimate for the frequency of the base generation with a correlation of 0.99 ± 0.01 for the Base Simulation. For all the following methods the Base Simulation had a very high correlation of 0.99 ± 0.01, due to the fact that there was a limited difference between the base generation and later generations.

The issue was that once selection was introduced the correlation between the base generation allele frequency and the first genotyped generation decreased to 0.88 ± 0.01. The Simulation with selection and the dataset with Pedigree errors had identical genotypes and therefore the correlation between allele frequencies of the base and first genotyped generation were also 0.88 ± 0.01. When the Genotyping errors dataset was used the correlation was unaffected at 0.88 ± 0.01. Future analysis should test if larger proportions of genotyping errors reduces the accuracy of these estimates and should also be considered across the different methods. As other methods were more accurate and not much more time consuming, using the first genotyped generation to estimate the allele frequency of the base generation was not recommended. Further analysis of accuracy of estimates, of the base generation allele frequency focused on the Simulation with selection.

The correlation between the known base generation allele frequencies and estimates using MiXBLUP increased to 0.96 ± 0.01 for the Simulation with selection. This supported Gengler *et al.* (2008), that the accuracy of allele frequencies estimated by this method are reliable. There were however four outlier SNPs that had estimated allele frequencies outside of the parameter space by <0.01. These four SNPs had low minor allele frequencies (<0.001) in the base generation.

The most accurate estimates for base generation allele frequency was with the equivalent GLS estimator. The GLS method implemented in allelefreq gave a high correlation between the base generation allele frequency and the estimated frequency, of 0.97 ± 0.01 when ignoring estimates outside the parameter space.

There was the issue of 206 SNPs originally outside of the parameter space for the Base simulation, and when the allele coding was swapped 190 SNPs were outside the parameter space. If one of those allele coding estimates were within the parameter space and used, only three estimates remained outside the space (The same three SNPs observed outside parameter space with MiXBLUP). Only for those SNPs, the estimated allele frequencies deviated from the estimates obtained with the full A matrix from Calc_grm, which approach always returned estimates within the parameter space. It was assumed that these estimates would not be an issue as the known allele frequencies could be considered fixed at >0.99 in both the base generation and last generation. This has also been observed by Makgahlela *et al.* (2013), which suggested these estimates are due to the simplified model of Gengler *et al.* (2007), in the sense that a binomial model maybe more appropriate for SNPs with (very) low minor allele frequency, and does not impose restrictions on the parameter space.

Conclusions

Several processes for genomic prediction require base generation allele frequencies. A computationally efficient method was needed. This study compared time, memory use, and accuracy for a number of methods and different implementations. It was recommended that the generalized least squares method, with an pedigree relationship matrix computed using sparse matrices, be used to estimate base generation allele frequencies. If very high accuracies for base generation allele frequencies is needed a combination of the methods could be considered but this method should be suitably efficient and accurate for genomic prediction aspects.

Acknowledgements

This study was financially supported by the Dutch Ministry of Economic Affairs (TKI Agri & Food project 16022) and the Breed4Food partners Cobb Europe, CRV, Hendrix Genetics and Topigs Norsvin. The use of the HPC cluster has been made possible by CAT-AgroFood (Shared Research Facilities Wageningen UR).

References

Calus, M.P.L. & Vandenplas J. 2016. Calc_grm – a program to compute pedigree, genomic, and combined relationship matrices. *ABGC, Wageningen UR Livestock Research*.

Garcia-Baccino, C.A., Legarra, A., Christensen, O.F., Misztal, I., Pocrnic, I., Vitezica, Z.G. & Cantet, R.J. 2017. Metafounders are related to F_{st} fixation indices and reduce bias in single-step genomic evaluations. *Genetics Selection Evolution. 49*, 34.

Gengler, N., Mayeres. P. & Szydlowski, M. 2007. A simple method to approximate gene content in large pedigree populations: application to the myostatin gene in dual-purpose Belgian Blue cattle. *Animal 1*, 21-28.

Gengler, N., Abras, S., Verkenne, C., Vanderick, S., Szydlowski, M. & Renaville, R. 2008. Accuracy of prediction of gene content in large animal populations and its use for candidate gene detection and genetic evaluation. *Journal of Dairy Science 91*, 1652-1659.

Makgahlela, M., Strandén, I., Nielsen, U., Sillanpää, M. & Mäntysaari, E. 2013. The estimation of genomic relationships using breedwise allele frequencies among animals in multibreed populations. *Journal of Dairy Science 96*, 5364-75.

McPeek, M.S., Wu, X. & Ober, C. 2004. Best Linear Unbiased Allele-Frequency Estimation in Complex Pedigrees. *Biometrics 60,* 359-367.

R Core Team. 2017. R: A language and environment for statistical computing. *R Foundation for Statistical Computing, Vienna, Austria.*

Sargolzaei, M. & Schenkel, F.S. 2009. QMSim: a large-scale genome simulator for livestock. *Bioinformatics 25,* 680-681.

Schenk, O., Gärtner, K., Fichtner, W. & Stricker, A. 2001. PARDISO: a high-performance serial and parallel sparse linear solver in semiconductor device simulation. *Future Generation Computer Systems 18*, 69-78.

Ten Napel, J., Vandenplas, J., Lidauer, M., Stranden, I., Taskinen, M., Mäntysaari, V, Calus, M.P.L. & Veerkamp, R.F. 2017. MiXBLUP, user-friendly software for large genetic evaluation systems – Manual V2.1-2017-08, *Wageningen, the Netherlands.*