Efficient algorithms to identify duplicated genotypes in large datasets

M. H. Ferdosi[1]*, S. Masoodi[2] and M. Khansefid[3]

[1]AGBU, a joint venture of NSW Department of Primary Industries and University of New England, 2351, Armidale, Australia; [2]Computer Engineering Group, University of Payam Noor, Tehran, Iran; [3]Agriculture Victoria, AgriBio, Centre of AgriBioscience, 3083, Bundoora, Australia; *mferdos3@une.edu.au

Abstract

In this paper, we introduced two novel algorithms to identify duplicated genotypes. The runtime of these algorithms was compared with the widely adopted Exhaustive Search algorithm using simulated data. We found that both new algorithms could significantly reduce the execution time. Further, the optimised Matrix Algebra Approach algorithm was faster than the Dis-Similarity lookup table and could improve the performance nearly 34 times compared to Exhaustive Search.

Introduction

Thousands of farm animals are routinely genotyped with Single Nucleotide Polymorphism (SNP) panels every month in Australia. These new genotypes are continuously added to the genomic datasets for use in genomic evaluation or parent identification. To ensure data integrity, efficient algorithms and programs are needed to assess the quality of the data in large genomic datasets. Checking for mislabelled and duplicated genotypes is the first quality control step when new genotypes are added to a dataset or combined with genomic datasets. Identifying and removing repeated genotypes is also critical before imputation and haplotype phasing. Previously developed algorithms often use fingerprinting methods to identify duplicate genotypes with an approximate approach (Jin et al., 2017). In this paper, we proposed two efficient and deterministic algorithms to identify duplicated genotypes that could also handle missing and wrong genotypes.

Materials & Methods

Simulated Genotypes. Genotypes were simulated using QMSim V1.10 (Sargolzaei and Schenkel, 2009). Thirty chromosomes, each with 333 SNPs and a length of 100 centimorgans were simulated for 10 generations. Hence, in total 9,990 SNPs were simulated which was marginally more than Illumina BovineLD v2.0 BeadChip with 7,931 evenly spaced SNPs (recommended SNP-chip for cost-effective genotyping of cows in commercial herds). In each generation, 20 males were mated with 400 females. The genotypes across all generations (8,000 individuals) were used in this study. The genotypes of 16 individuals (every 500th animal) were duplicated. In the next step, two percent genotyping error and missing SNPs were added to the genotypes at random using R (R Core Team, 2021). The genotypes were stored in an integer matrix in which each element was 0, 1, 2 or 5, representing *aa*, *ab*, *bb* or a missing genotype, respectively.

We compared four methods for identification of duplicated genotypes.

Exhaustive Search (ES). In this method three nested loops were used to check every pair of individuals. The algorithm stopped the computation, and the two individuals were considered different when the error threshold (10 percent) between the genotypes was reached. We considered this algorithm as the base method to identify duplicated genotypes.

Matrix Algebra Approach (MAA). This method is an extension and modification of the previously described method to identify opposing homozygotes by Ferdosi and Boerner (2014). Suppose **M** is an integer matrix of 0, 1, 2 and 5, for *aa*, *ab*, *bb* and missing genotypes, respectively. The rows contain the genotypes for the individuals and the columns are associated with the markers. Therefore, $\mathbf{M}_{n\times m}$ is the matrix of genotypes, where *n* is the number of individuals and *m* is the number of markers. In the first step, a Boolean matrix for each genotype class (i.e., 0, 1, 2 and 5) should be constructed. For example, for *aa* (i.e., 0) genotypes, we need a matrix (**Z** matrix in Equation 1) with 1's and 0's; 1's for the presence of *aa* and 0's otherwise.

$$\begin{aligned}
&\mathbf{R} = 1 - (\mathbf{M}/5)\\
&\mathbf{Z} = -((\mathbf{M} - 2\mathbf{J})/2) \circ \mathbf{R}\\
&\mathbf{O} = (\mathbf{J} - (\mathbf{M} - \mathbf{J}) \circ (\mathbf{M} - \mathbf{J})) \circ \mathbf{R}\\
&\mathbf{T} = (\mathbf{M}/\mathbf{2}) \circ \mathbf{R}\\
&\mathbf{S} = \mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \mathbf{O}\mathbf{O}^{\mathrm{T}} + \mathbf{T}\mathbf{T}^{\mathrm{T}}
\end{aligned} \qquad (1)$$

where ∘ is the Hadamard product; **R** is a Boolean matrix in which each element is 0 or 1 corresponding to a missing or non-missing genotype, respectively. **Z**, **O** and **T** are incidence matrices containing 0's and 1's for presence of *aa* (i.e., 0), *ab* (i.e., 1) and *bb* (i.e., 2) genotypes in the **M** matrix, respectively. **J** is the matrix of ones and **S** is the similarity matrix that shows the number of common SNPs between two individuals. Individuals with high level of similarity (> 90% of genotypes) were considered as duplicated samples. The key point for optimisation of the MAA algorithm is that **Z**, **O** and **T** matrices are Boolean and their cross products is symmetric. Finally, the $1 - \left(\frac{\mathbf{S}}{\mathbf{R}\mathbf{R}^{\mathrm{T}}}\right)$ matrix shows the proportions of SNPs varying between individuals.

Dis-Similarity Lookup Table (DSLT). Firstly, a Dis-Similarity Lookup Table (**L**) was calculated. This table of dimensions 256 x 256 consists on each axis all the combinations of four SNPs (0000, 0001, 0002, 0005, 0010, …, 5555) and the values stored are the numbers of elements different between the four combinations of SNPs (0, 1, 2, 5). For example, the dissimilarity between 0000 and 0012 is 2. The error per SNP was calculated by dividing one by the total number of SNPs, such that when accumulated across all SNPs the maximum value is one. For example, the table value when the marker matrix contains 20 SNPs between 00 and 12 is 0.05. For computational simplicity, the marker matrix was packed, **P**, such that each four adjacent SNPs for an animal were combined into a single number between 1 and 256. The proportion of dissimilarity between two genotypes was then calculated as

$$Mismatch_{i,j} = \sum_{n=1}^{n=N \vee m>T} \mathbf{L}\left(\mathbf{P}_{i,n}, \mathbf{P}_{j,n}\right) \qquad (2)$$

where *T* is the threshold for the number of mismatching genotypes, *N* is total number of segments and *m* is the number of mismatches up to the n^{th} segment. Finally, it is possible to exit this summation early if mismatch is greater than some threshold *T* (10 percent of segments).

Matrix Algebra Approach Optimised- MAAO. This approach optimised the MAA algorithm further by firstly checking the 1/5 of the markers to identify the individuals which were different in at least 10% of total number markers (i.e., unduplicated genotypes). Subsequently, for the potentially duplicated individuals, all markers were checked to confirm the similarity between

genotypes was greater than 90%. Hence, the algorithm did not assess all markers if the number of dissimilar markers was above the threshold in the subset of markers.

All algorithms were implemented in C++ and parallelised with OpenMP Version 5 (OpenMP Architecture Review Board, 2018).

Comparison of algorithms. The speed of the algorithms was compared using the simulated genotypes. The runtime of the different methods was measured for 500, 1,000, 1,500, …, and 8,000 individuals. The codes were compiled with GNU Compiler Collection version 11 and tested on a Linux system with 64 GB of random-access memory (DDR4 2400) and Intel Central Processing Unit (CPU; Intel Core^{TM} i7-7700K Processor, 4 cores × 4.2 GHz).

Results and discussion

Figure 1 shows the logarithm of elapsed time to identify duplicated genotypes. Both MAA and DMS algorithms were faster than the ES method, especially when the genotypes of 8,000 individuals were checked for duplicates. DMS was around 14 times, MAA was around 10 times and OMAA was around 34 times faster than ES method (all single thread). The MAA and DMS could extract information from the raw genotypes, allowing less computation during pairwise comparison. This optimisation was more obvious in DMS algorithm. Multi-threading with quad-core CPU has similar effects on MAA and DMS algorithms and increased their performances nearly 3.8 times but increased the ES method's performance almost 4.8 times because the ES method did not require any initial optimisations and could benefit from vectorisation more effectively.

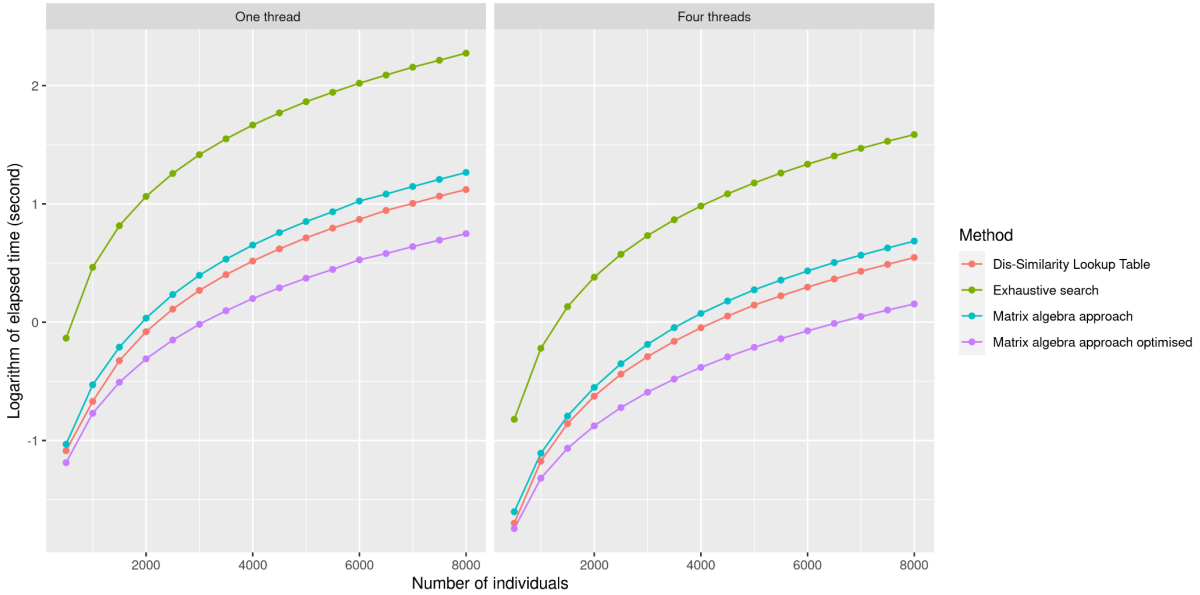

Figure 1. The figure compares the performance of four algorithms to identify duplicated genotypes.

Conclusion

In this article, we presented two novel deterministic algorithms for identification of duplicated genotypes. The performance of these methods can be improved further by identifying the duplicated genotypes with an approximate approach. We reported preliminary results in this paper, and further study is required to compare these algorithms to identify duplicated

genotypes with the previously developed algorithms such as *Genetic Relationship and Fingerprinting* as well (Jin et al., 2017).

Acknowledgements
This study was supported by Meat and Livestock Australia project L.GEN.1704.

References
Ferdosi, M. H. and V. Boerner (2014) Liv. Sci. 166:35–37. https://doi.org/10.1016/j.livsci.2014.05.026.
Jin, Y., A. A. Schäffer, S. T. Sherry and M. Feolo (2017) PloS one 12(6). https://doi.org/10.1371/journal.pone.0179106
OpenMP Architecture Review Board (2018) OpenMP application program interface version 5.0.
R Core Team (2021). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
Sargolzaei, M. and F. S. Schenkel (2009) Bioinformatics 25(5):680–681. https://doi.org/10.1093/bioinformatics/btp045