RESEARCH

Optimizing Sequencing Resources in Genotyped Livestock Populations Using Linear Programming

Hao Cheng[1*], Keyu Xu[1] and Kuruvilla Joseph Abraham[2]

*Correspondence:
qtlcheng@ucdavis.edu
[1]Department of Animal Science, University of California Davis, Davis, CA, 95616, US
Full list of author information is available at the end of the article

Abstract

Background: Low-cost genome-wide single-nucleotide polymorphisms (SNPs) are routinely used in animal breeding programs. Compared to SNP arrays, the use of whole-genome sequence data generated by the next-generation sequencing technologies (NGS) has great potential in livestock populations. However, a large number of animals are required to be sequenced to exploit the full potential of whole-genome sequence data. Thus, novel strategies are desired to allocate sequencing resources in genotyped livestock populations such that the entire population can be sequenced or imputed efficiently.

Methods: We present two applications of linear programming models called LPChoose for sequencing resources allocation. The first application is to identify the minimum number of animals for sequencing while meeting the criteria that each haplotype in the population is contained in at least one of the animals selected for sequencing. The second is to sequence a fixed number of animals whose haplotypes include as large a proportion as possible of the haplotypes present in the population given a limited sequencing budget. In both cases, we assume that all animals have been haplotyped. We present results from approximation algorithms, and motivate the use of approximations through the correspondence of the problems we address with problems in computer science for which there are no known efficient algorithms.

Results: In both applications LPChoose performed consistently better than some existing methods making similar assumptions.

Background

The discovery of genome-wide single-nucleotide polymorphisms (SNPs) and effective ways to assay them has revolutionized genetic analyses of quantitative traits in animal breeding [1, 2, 3, 4, 5]. In conventional breeding programs, low-cost SNP array data are routinely used in genomic selection to estimate breeding values. Compared to SNP arrays, the use of whole-genome sequence data generated by the next-generation sequencing technologies (NGS) has great potential in livestock populations for causal mutation detection [6] in genome-wide association studies and more stable or accurate prediction of breeding values [7] in genomic selection. However, a large number of animals are required to be sequenced to exploit the full potential of whole-genome sequence data. As sequencing is expensive, there is considerable interest in extracting as much information as possible by sequencing a limited number of animals. Several strategies have been proposed to allocate sequencing resources in genotyped livestock populations. One recent approach is the AlphaSeqOpt [8], where, as in this paper, the authors assume that all animals

are haplotyped. Furthermore, as in [8], we assume that if an animal is sequenced, the sequence information for all other animals in the population with same haplotypes can be imputed without errors.

The problem of selecting the best set of animals in a population capable of satisfying certain constraints is an optimization problem in a space whose dimension is equal to the total number of animals in the population. In practical applications and in the examples we will present, the optimization must be performed for tens of thousands of animals subject to tens of thousands of constraints. In general, solving an optimization problem in a space of such high dimension by either numerical or analytical approaches is intractable. However, the optimization problem we are trying to solve here can be addressed through the use of a well-established set of techniques known as linear programming (for an introduction to linear programming and references to original papers see [9]). In linear programming, exact or approximate optimum solutions through numerical approaches can be obtained by using two simplifying features: the objective function to be optimized is linear; the constraints to be satisfied are also linear. Linear programming has been previously applied in animal and plant breeding to optimize breeding decisions [10, 11]. However, the use of linear programming for the selection of the most suitable animals to allocate sequencing resources has not been studied. In this paper we will study the application of linear programming to the allocation of sequencing resources across animals in a population to address two different but related questions which may arise in breeding programs.

The first question is to determine the minimum number of animals needed to permit sequence imputation into all other members of the population, and then to identify these animals. Among all animals in the population, this subset of animals alone will be subsequently selected for sequencing. The rationale behind choosing only these animals for sequencing is as follows: upon sequencing any given animal in this subset, imputation in any other animal carrying the same haplotypes as the sequenced animal can be carried out; this is identical to the assumption made in [8]. Thus, once a subset of animals which contains all the haplotypes in the population has been identified, it suffices to sequence only the animals in that subset to permit imputation into the entire population. In order to minimize overall costs, we will attempt to find the smallest subset of animals with this property.

In practice, however, even after the smallest subset of such animals has been identified, it may not be possible to sequence all the animals identified in this manner due to budget constraints. If the budget for sequencing is limited, then it may become necessary to choose a limited subset of animals carrying haplotypes which may be considered to be representative of the population on the basis of being more frequent. Identifying this limited subset of animals is the second question which will be addressed in this paper. The objectives of this paper are to study the use of linear programming models to address both of these questions described above and compare their performance with existing methods. Both questions are related to the issue addressed in algorithm 1 in [8]. We do not discuss issues of haplotype phasing and resolution which are addressed in algorithm 2 in [8].

Methods

Before presenting details of our methods we establish some notations which will be used throughout this paper. For each animal we associate an indicator variable x_j such that $x_j \in (0,1)$ and $j \in (1,2,\ldots,n)$ where n is the total number of animals. $x_j = 1(0)$ means that the animal j is selected (not selected) for sequencing. We next introduce binary constants a_{ij} where $i \in (1,2,\ldots,p)$ and p is the total number of unique haplotypes. $a_{ij} = 1(0)$ if animal j carries haplotype i (or not). Since we are solving systems of inequalities in which the variables are binary valued, we actually work within the framework of integer linear programming, which is more restricted than linear programming.

The first application of integer linear programming is to identify the minimum number of animals for sequencing while meeting the criteria that each haplotype is contained in at least one of the animals selected for sequencing. If haplotype i is carried in at least one animal in the population, then we require that $\sum_{j=1}^{n} a_{ij}x_j \geq 1$. This condition can be satisfied only if $x_j = 1$ for at least one value of $1 \leq j \leq n$. A similar constraint must hold separately for all $i \in 1,2,\ldots,p$. Note that all these inequalities can be trivially satisfied by sequencing all animals, i.e., $x_j = 1$ for all $j \in 1,2,\ldots,n$. This solution almost never leads to the smallest number animals required for sequencing. In order to minimize the number of animals to be sequenced, we additionally require $z_1 = \sum_{j=1}^{n} x_j$, where z_1 is the number of selected animal, to be as small as possible. These inequalities can be collectively written as

$$\text{minimize} \quad z_1 = \sum_{j=1}^{n} x_j, \quad x_j \in 0,1 \tag{1}$$

$$\text{subject to the constraints} \quad \sum_{j=1}^{n} a_{ij}x_j \geq 1, \quad i = 1,2,3,\cdots,p \tag{2}$$

Equation 1 is the objective function to minimize the total number of selected animals. Equation 2 is the set of constraints which ensures that each haplotype is present in at least one of the animals selected for sequencing.

The objective function and the constraints can also be written in matrix notation as

$$\text{minimize} \quad z_1 = \mathbf{1}^T \boldsymbol{x}, \tag{3}$$

$$\text{subject to} \quad \mathbf{A}\boldsymbol{x} \geq \mathbf{1} \tag{4}$$

where $\mathbf{A}$ is the matrix of values a_{ij} and $\boldsymbol{x}$ denotes the vector of x_i values. This is a system of inequalities which can be solved by integer linear programming. The solution for x will contain some values equal to zero and others equal to one. The x_j which are equal to 1 correspond to the animals which should be sequenced. Once these animals are sequenced, by our earlier assumptions, imputation in the entire population becomes possible.

The second question, to select a fixed number of animals with most common haplotypes can also be addressed using integer linear programming. In order to prioritize animals with more frequent haplotypes we define a vector $\mathbf{h}$ with $1 \leq i \leq p$

whose ith element is the frequency of the ith haplotype. The vector $\mathbf{h}$ are then used to define another vector $\mathbf{c} = \mathbf{h}^t\mathbf{A}$. The number of elements of $\mathbf{c}$ is equal to the number of animals. Values of $\mathbf{c}$ are larger for animals which carry many more frequent haplotypes. Thus the values of $\mathbf{c}$ will be used as a guide to select animals when sequencing resources are limited. In addition, it is important to ensure that the same haplotype is not sequenced in a large number of different animals so as to maximize the haplotype diversity in the animals selected for sequencing. All these different requirements can be summarized in the following set of inequalities

$$\text{maximize} \quad z_2 = \sum_{j=1}^{n} c_j x_j, \quad x_j \in 0, 1 \tag{5}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_j \leq n_{max} \tag{6}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq r_{max} \quad i = 1, 2, 3, \cdots, p \tag{7}$$

The objective function and the constraints can also be written in matrix notation as

$$\text{maximize} \quad z_2 = \frac{1}{n}\mathbf{1}^T \boldsymbol{A}^T \boldsymbol{A}\boldsymbol{x} \tag{8}$$

$$\text{subject to} \quad \mathbf{1}^T \boldsymbol{x} \leq n_{max} \tag{9}$$

$$\mathbf{A}\boldsymbol{x} \leq r_{max} \tag{10}$$

Equation 5 is the objective function to maximize number of most frequent haplotypes represented by the selected animals. Equation 6 is the constraint to set the maximum number of animals to be sequenced n_{max}. Equation 7 is the constraint to ensure that each haplotype covered is at the most covered by r_{max} animals, where r_{max} should be a positive integer ≥ 1 and it is chosen to be considerably smaller than n_{max}. Ideally, the objective function is maximized with the value of r_{max} equal to 1 and the maximization of the objective function in equation 5 is performed in one step. In our data analysis, the value of r_{max} is set to be 2 to reduce the computational burden.

Linear programming problems described above cannot be solved analytically. However, they can be solved using branch and bound methods, which are guaranteed to converge to the global optimum. In practice, for reasons which will be discussed later, the convergence can be very slow. Thus we will use a fast approximation in the second application to select a fixed number of animals. Instead of selecting n_{max} animals at once, we will first select 2 animals (i.e., $n_{max} = 2$). Once the first two animals have been identified, these animals and all the haplotypes present in them are removed. Then the optimization in equations 5, 6, and 7 is repeated but with the objective function in equation 5 suitably modified to reflect the absence of the first two animals. This procedure can be carried out until the desired number of animals has been found. This approximation is very similar in spirit to that performed in [8], with one crucial difference. In [8], the animals are chosen one at a time. Thus, after the first animal is chosen and discarded along

with its haplotypes, there is no guarantee that the second animal will not share haplotypes with the first animal. Choosing animals two at a time this possibility can be rigorously excluded if r_{max} is chosen to be 1, and discouraged with r_{max} chosen to be equal to 2 or 3. Thus the scheme above can potentially lead to a greater coverage of haplotypes with the same number of animals.

Numerical example

A numerical example is shown below with haplotype information for 5 animals and 2 haplotypes blocks (table 1).

In table 1 numerical labels for haplotpyes should be interpreted as purely qualitative labels. As there are 5 animals and 13 distinct haplotypes, the corresponding **A** matrix has 5 columns and 13 haplotypes and is shown below.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

where the element from its ith row and jth column is a_{ij}, and a_{ij} is a binary variable such that $a_{ij} = 1(0)$ if animal j contains haplotype i (or not).

The first application of integer linear programming to select the minimum number of animals to identify all haplotypes is formulated as

$$\text{minimize} \quad z_1 = x_1 + x_2 + x_3 + x_4 + x_5, \quad x_j \in (0, 1) \tag{11}$$

$$\text{subject to} \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{12}$$

As the result, animal 1,2, 3, and 5 are selected to be sequenced.

The second application of integer linear programming to select a fixed number of animals, e.g., 2 animals, with most common haplotypes is formulated as

$$\text{maximize } z_2 = 0.286x_1 + 0.286x_2 + 0.524x_3 + 0.524x_4 + 0.238x_5, \tag{13}$$

$$\text{subject to} \quad x_1 + x_2 + x_3 + x_4 + x_5 \leq 2 \tag{14}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \tag{15}$$

At the end, animal 1 and animal 3 are selected.

Data analysis

Genotype data for five different scenarios were simulated using AlphaSimR [12]. The number of generations considered in these 5 scenarios are 5, 10, 15, 30, or 50. These scenarios resembled modern cattle populations. The genome of 10,000 segregating loci on 10 chromosomes was simulated using the "cattle genome" option in AlphaSimR. A quantitative trait controlled by 150 QTL of effects sampled from standard normal distributions distributed equally on 10 chromosomes was simulated. First, founders of 1,000 cattle of equal sex ratio were generated. At each generation, the best 25 males were selected as sires on the basis of their highest breeding value and mated to all 500 females as dams to produce next generations with 1000 cattle of equal sex ratio. In our analysis, 10 replicates were simulated for each of the five scenarios. From the above, all individuals had haplotypes for 10,000 SNPs distributed equally across the 10 chromosomes. In each population, haplotype blocks of length 100 SNPs were obtained across 10 chromosomes. A mismatch of up to 10% was used to ensure that haplotypes with small differences were considered as identical. In the results which will be presented, the selection of a fixed number animals will also be made after the exclusion of haplotypes with a frequency of less than 1%, similar to the strategy adopted in [13]. An open-source, publicly available Julia package called LPChoose (https://github.com/reworkhow/LPChoose.jl) has been developed. A publicly available implementation of linear programming, the GNU Linear Programming Kit (GLPK), is used in LPChoose. In particular, for the sake of the reproducibility of our results, we do not make use either directly or indirectly of any proprietary solvers for linear programming problems.

Results

The minimum numbers of animals identified in both LPChoose and AlphaSeqOpt to contain all unique haploptypes in five populations are shown in table 2. The minimum numbers of animals identified by LPChoose were consistently smaller than those from AlphaSeqOpt. As AlphaSeqOpt does not directly output the minimum number of animals needed to cover all haplotypes, the values in the last column of table 2 were obtained from repeated runs of AlphaSeqOpt with varying numbers of animals to be selected. The difference between LPChoose and AlphaSeqOpt in the minimum number of selected animals increased with the number of generations. However, the ratio of the difference and the total number of animals remained 0.1 - 0.2. The results of table 2 suggest that the difference between these methods in the smallest number of animals required grows as the size of the population increases. We emphasize that the results in column 3 of table 2 obtained from linear programming are obtained without any approximations, and are thus equal to the theoretically lowest values attainable for this kind of problem.

LPChoose and AlphaSeqOpt were also used to select a fixed number of animals (100 animals) whose haplotypes represent maximum proportions of the total haplotypes in the population. The results obtained are shown in tables 3 and 4. In table 3, the proportion of all haplotypes represented by 100 selected animals were compared. Animals identified in LPChoose consistently covered higher proportion of all haplotypes than those in AlphaSeqOpt, although only a small proportion (0.1 - 03) of all haplotypes were covered. In table 4, the proportion of common haplotypes represented by 100 selected animals were compared. LPChoose performed better consistently across different populations, and the proportions of common haplotypes identified in LPChoose were usually higher than 0.99, and those in AlphaSeqOpt are about 0.8 - 0.9.

Discussion

In this paper, linear programming models are used to answer two questions for the allocation of sequencing resources. The first question addressed is to identify the minimum number of animals whose haplotypes represent all the haplotypes in the population. The second question we address is to choose for sequencing a fixed number of animals whose haplotypes represent the maximum proportion of the total or common haplotypes in the population. In practice, the second question usually comes into play due to the limited sequencing budget. Because of restrictions in the sequencing budget, it may be possible to sequence only a fixed number of animals which may turn out to be less than the number of animals needed to cover all the haplotypes in the population. In this scenario, the animals selected should carry as large a proportion as possible of the haplotypes in the population, i.e., selecting animals carrying most frequent haplotypes in the population.

In the results presented, it has been implicitly assumed that once an animal has been sequenced then imputation on all other animals carrying the haplotypes of the sequenced animals can be carried out. Indeed, the structure of the linear programming equations and the resulting solutions rely on this assumption. The results we present suggest our methods offer an improvement over those previously published using these same criteria. Furthermore, LPChoose makes no explicit use of the pedigree once the haplotypes are known, in contrast to several other methods that rely

on detailed pedigree information [14]. Other criteria have been used to determine the best animals for sequencing, including the expected accuracy of genotype imputation [15, 16]. The methods in this paper could also be used as a guide to select animals which permit the best imputation accuracy. Thus this paper should be a valuable complement to existing published methods

There are some unavoidable limitations associated with the linear programming methods we have used. The presence of these limitations has already been suggested earlier, e.g., the results presented in table 4 are based on an approximate solution to the linear programming problem. To get additional insight into the technical difficulties which necessitate the use of approximations we will rephrase the problem of identifying the smallest set of animals which carry all the haplotypes in the population. We begin by defining a network consisting of the animals and haplotypes as follows. For each animal we define a vertex in an undirected graph and for each haplotype we also define a vertex in the same undirected graph. The edges in this undirected graph are defined as follows: there are no edges between vertices which define haplotypes; there are also no edges between vertices which represent animals. There is an edge between two vertices if one of the vertices represents an animal, the other represents a haplotype, and the animal contains the haplotype. This graph is thus a bipartite graph, with two subsets of vertices, representing animals and haplotypes, and the only edges present are those between the two subsets.

We now focus on just one subset of vertices, those representing the haplotypes. Let this subset be denoted by $\mathcal{H}$. $\mathcal{H}$ can be further partitioned into subsets in a very natural way. Let elements $h_{i_1}, h_{i_2}, \ldots, h_{i_o}$ be the haplotypes contained in animal i where $1 \leq i \leq n$. All these haplotypes can be grouped into one subset which we denote by $\tilde{h}_i$. If there are n animals, then there must be n such subsets and the union of all these n subsets is $\mathcal{H}$. Many of these subsets share elements as the same haplotype can appear in multiple animals. If there is a rare haplotype which appears in a single animal, then that haplotype will appear in a single subset. The problem of identifying the smallest subset of animals which contain all of the haplotypes amounts to identifying a minimal number of subsets, $\widetilde{h}_1, \widetilde{h}_2, \ldots, \widetilde{h}_m$ of $\mathcal{H}$ whose union is $\mathcal{H}$. The union of this minimal number of subsets contains all haplotypes and the number of subsets is m. m is the smallest number of animals needed to cover all haplotypes in the population.

The identification of $\widetilde{h}_1, \widetilde{h}_2, \ldots, \widetilde{h}_m$ is the well known set cover problem in graph theory [17]. The methods presented earlier in this paper for finding the minimal number of animals containing all the haplotypes in the population are essentially the solution to a set cover problem using integer linear programming, which is a very well established method for the solution of the set cover problem in a general context [17]. However, there is no known polynomial time algorithm for solving the set cover problem. Thus as the number of haplotypes grows, computational costs associated with an exact solution may become prohibitive.

The second computational issue we discuss, that of finding the best small subset of animals, can also be interpreted in terms of a set cover problem with additional weighting factors corresponding to haplotype frequencies. In light of the correspondence of the questions we address with a class of problems in combinatorial optimization for which there is no known efficient algorithm, it is not surprising

that approximations were required. Much of the difficulty arises due to the discrete nature of the x variables. In previous papers using continuously varying genomic fractions to optimize mating decisions to control for inbreeding [18, 19] this difficulty does not arise.

The approximations we have made use of are not the only approximations which can be employed in problems of this nature. Another class of approximation which can be easily incorporated into the framework of linear programming is to relax the requirement in equation 3 and equation 4 that the x variables be binary. Instead they can be set to be continuous in the range from 0 to 1. This relaxation transforms the original problem to a problem can be solved in polynomial time. However, after relaxation the solutions for the x variables may lie between 0 and 1. One possibility to decide which animals should be discarded and which should be sequenced is to choose a threshold (e.g., 0.8) such that all x values above the threshold are set equal to one and all values below the threshold are set equal to zero. In this way animals to be sequenced are identified. However, there is no guarantee that with this thresholding scheme the animals to be sequenced will contain all the haplotypes. If there are some haplotypes that have not been covered, then the additional animals to be sequenced can be identified by solving the relaxed versions of equation 3 and 4 subject to the additional requirements that the x variables for the animals found in the previous approximation be conditioned to 1. With this conditioning requirement, many of the constraints in equation 4 are now satisfied. With many constraints satisfied, Equation 3 and 4 can be solved again more easily. This cycle can be repeated until all constraints are satisfied. Once a sufficient number of constraints are satisfied, it may be that there is no longer any need to address the relaxed form of the equations and the remaining constraints be solved using binary x values. The scheme outlined offers a way to get approximate solutions for equation 3 and 4 when the size of the data set makes an exact solution computationally unfeasible.

Despite these fundamental limitations necessitating the use of approximations, linear programming methods offer many advantages. For example, with linear programming multiple solutions can be identified if needed. Another example of the flexibility of linear programming methods arises if one animal must be included in the set of animals to be sequenced. In this case that animal can be included by an additional constraint in equation 4, and the other animals selected in the most economical way after taking into account haplotypes present in the animal included by requirement. The requirement that multiple pre-selected animals have to be included can also be accommodated by the inclusion of multiple additional constraints. If animals with low frequency haplotypes should be preferentiallly included, as suggested in [20], then this preference can be accomodated in equation 3 by suitably changing the weights to favour animals carrying low frequency haplotypes.

Conclusions

We have illustrated the use of linear programming for optimizing the allocation of sequencing resources. The results presented show that linear programming offers certain clear advantages over some currently known approaches. The methodology

we have illustrated here could be used in conjunction with methods for phasing and haplotype resolution to maximally optimize the use of limited sequencing budgets.

Competing interests
The authors declare that they have no competing interests.

Author's contributions
KJA conceived the study. KJA, HC, KYU contributed to the development of the methodology. KJA, HC, KYU developed the LPChoose package. KYU preformed the simulation and analysis. HC, KJA wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgements
This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil(CAPES) - Finance Code 001.

Author details
[1]Department of Animal Science, University of California Davis, Davis, CA, 95616, US. [2]Department of Economics, University of São Paulo, Ribeirão Preto, 14040-905, Brazil.

References
1. VanRaden, P.M.: Efficient Methods to Compute Genomic Predictions. Journal of Dairy Science **91**(11), 4414–4423 (2008)
2. Hayes, B., Bowman, P., Chamberlain, A., Verbyla, K., Goddard, M.: Accuracy of genomic breeding values in multi-breed dairy cattle populations. Genetics Selection Evolution **41**(1), 51 (2009). doi:10.1186/1297-9686-41-51
3. VanRaden, P.M., Van Tassell, C.P., Wiggans, G.R., Sonstegard, T.S., Schnabel, R.D., Taylor, J.F., Schenkel, F.S.: Invited review: reliability of genomic predictions for north american holstein bulls. J Dairy Sci **92**(1), 16–24 (2009). doi:10.3168/jds.2008-1514
4. Habier, D., Tetens, J., Seefried, F.-R., Lichtner, P., Thaller, G.: The impact of genetic relationship information on genomic breeding values in german holstein cattle. Genetics Selection Evolution **42**(1), 5 (2010). doi:10.1186/1297-9686-42-5
5. Wolc, A., Arango, J., Settar, P., Fulton, J., Sullivan, N.P., Preisinger, R., Habier, D., Fernando, R., Garrick, D.J., Hill, W.G., Dekkers, J.C.M.: Genome-wide association analysis and genetic architcture of egg weight and egg uniformity in layer chickens. Animal Genetics **43 (Suppl 1)**, 87–96 (2012)
6. Daetwyler, H.D., Capitan, A., Pausch, H., Stothard, P., van Binsbergen, R., Brøndum, R.F., Liao, X., Djari, A., Rodriguez, S.C., Grohs, C., Esquerré, D., Bouchez, O., Rossignol, M.-N., Klopp, C., Rocha, D., Fritz, S., Eggen, A., Bowman, P.J., Coote, D., Chamberlain, A.J., Anderson, C., VanTassell, C.P., Hulsegge, I., Goddard, M.E., Guldbrandtsen, B., Lund, M.S., Veerkamp, R.F., Boichard, D.A., Fries, R., Hayes, B.J.: Whole-genome sequencing of 234 bulls facilitates mapping of monogenic and complex traits in cattle. Nature genetics **46**(8), 858–865 (2014)
7. Meuwissen, T., Goddard, M.: Accurate Prediction of Genetic Values for Complex Traits by Whole-Genome Resequencing. Genetics **185**(2), 623–631 (2010)
8. Gonen, A., Ros-Freixedes, R., Battagin, M., Gorjanc, G., Hickey, J.: A method for the allocation of sequencing resources in genotyped livestock populations. Genetics, Selection and Evolution **49**(47) (2017)
9. Luenberger, D., Ye, Y.: Linear and Nonlinear Programming. Springer, New York (2015)
10. Diaz, C., Toro, M., Rekaya, R.: Comparison of restricted selection strategies: an application to selection of cashmere goats. Livestock Production Science, 89–99 (1999)
11. Moeinizade, S., Hu, G., L, W., Schnable, P.: Optimizing selection and mating in genomic selection with a look-ahead approach: An operations research framework. G3Genes Genomes Genetics, 2123–2133 (2019)
12. Faux, A.M., Gorjanc, G., Gaynor, R.C., Battagin, M., Edwards, S.M., Wilson, D.L., Hearne, S.J., Gonen, S., Hickey, J.M.: AlphaSim: Software for Breeding Program Simulation. The Plant Genome **9**(3), 1–14 (2016)
13. Su, H., Koltes, J., Saatchi, M., Lee, J., Fernando, R., Garrick, D.: Characterizing haplotype diversity in ten us beef cattle breeds. Animal Industry Report: AS 660, ASL R2846 (2014)
14. Boichard, D.: Pedig: A fortran package for pedigree analysis suited for large populations. In: Presented at 7th World Congress on Genetics Applied to Livestock Production (2002)
15. Ros-Frexedes, R., Whalen, A., Gorjanc, G., Milaham, A., Hickey, J.: Evaluation of sequencing strategies for whole-genome imputation with hybrid peeling. Genetics Selection Evolution (2020)
16. Yu, X., Wooliams, J., Meuwisssen, T.: Prioritizing animals for dense genotyping in order to impute missing genotypes of sparsely genotyped animals. Genetics, Selection and Evolution (2014)
17. Vazirani, V.: Approximation Algorithms. Springer, ??? (2003)
18. Pong-Wong, R., Wolliams, J.: Optimisation of contribution of candidate parents to maximise genetic gain and restricting inbreeding using semidefinite programming. Genetics, Selection and Evolution, 3–25 (2007)
19. Meuwissen, T.: Maximizing the response of selection with a pre-defined rate of inbreeding. Journal of Animal Science, 934–990 (1997)
20. Bickhart, D., Hutchinson, J., Null, D., VanRaden, P., Cole, J.: Reducing animal sequencing redundancy by preferentially selecting animals with low-frequency haplotypes. Journal of Dairy Science, 5526–5534 (2016)

Tables

Table 1 haplotypes at 2 blocks for 5 individuals used in the numerical example

Animals	block 1	block 2
1	1 2	7 8
2	2 3	9 9
3	3 4	10 11
4	2 4	11 12
5	5 6	12 13

Table 2 Minimum number of animals identified representing all haplotypes in the population

scenario	total number of animals	minimum number of animals (mean± SE)	
		LPChoose	AlphaSeqOpt
1	6,000	4,086± 13	4,988 ± 30
2	11,000	6,663 ± 18	8,563 ± 51
3	16,000	8,648 ± 34	11,563 ± 132
4	31,000	12,229 ± 86	17,506 ± 275
5	51,000	14,241 ± 119	21,098 ± 494

Table 3 Proportion of all haplotypes represented by 100 selected animals in the population

scenario	total number of animals	proportion of all haplotypes (mean± SE)	
		LPChoose	AlphaSeqOpt
1	6000	0.2218 ± 0.0036	0.2052 ± 0.0048
2	11000	0.1700 ± 0.0020	0.1528 ± 0.0048
3	16000	0.1498 ± 0.0026	0.1319 ± 0.0047
4	31000	0.1249 ± 0.0032	0.1119 ± 0.0037
5	51000	0.1145 ± 0.0027	0.1084 ± 0.0036

Table 4 Proportion of common haplotypes represented by 100 selected animals in the population

scenario	total number of animals	proportion of common haplotypes (mean± SE)	
		LPChoose	AlphaSeqOpt
1	6000	0.9934 ± 0.0017	0.9239 ± 0.0032
2	11000	0.9911 ± 0.0028	0.9041 ± 0.0071
3	16000	0.9921 ± 0.0018	0.8766 ± 0.0171
4	31000	0.9899 ± 0.0021	0.8444 ± 0.0109
5	51000	0.9993 ± 0.0006	0.8335 ± 0.0088

Note. haplotypes frequency of common haplotypes is larger than 1.0%.