GENO-DIVER (V3)

Genetic Simulation Toolkit

Last Updated on July 2018

Jeremy T. Howard
Christian Maltecca[1]
Francesco Tiezzi[1]
Jennie E. Pryce[2,3]
Matthew L. Spangler[4]

[1]North Carolina State University, Raleigh, NC, USA
[2]Department of Economic Development, Jobs, Transport and Resources, Victoria, Australia
[3]La Trobe University, Victoria, Australia.
[4]University of Nebraska, Lincoln, NE, USA

Contents

Introduction

A variety of disciplines including conservation (McMahon et al. 2014), animal and plant breeding (De los Campos et al. 2013) and human genetics (Yang et al. 2010) are currently making use of large volumes of genetic marker information. In particular, within animal and plant breeding programs the use of genomic information to predict the genetic merit of individuals has become a routine practice (Jonas & Koning 2015; Morrell et al. 2012). This has resulted in a significant increase in the number of individuals within a herd/population having genomic information. Nonetheless, the ability to use this information to efficiently manage agricultural populations at the genomic level, both for preserving genetic diversity and lessening inbreeding depression, remains a challenge for the near future. Furthermore, the ability to utilize genotype information in an optimal manner such as deciding who and when to genotype an animal along with how genomic information impacts the amount of phenotype information that is required is an active area of research. Lastly, although the identification of lethal mutations of large effect segregating within livestock populations with genomic data is possible (VanRaden et al. 2011), optimal mating procedures that minimize the frequency of a large number of lethal and more importantly sublethal mutations across generations have not been fully implemented. As the popularity of genotyping breeding individuals increases across species, the possibility to utilize genomic information from multiple sources to manage genomic information will also increase. Methods that make effective use of information from multiple sources including performance, genome diversity and inbreeding load at the selection and/or mating step are in increasing need. The routine genotyping of individuals is a significant investment from several sectors of agriculture and the need to spread the costs across multi-

ple avenues are of considerable practical interest.

The use of simulation is a low cost alternative to assess and validate proposed methods to predict genetic values or to compare alternative selection, genotyping, phenotyping or mating strategies across time. A number of simulation programs have been developed that simulate breeding livestock/crops populations, but they primarily focus on testing strategies where the genetic architecture is based solely on a quantitative trait governed by additive effects (Sargolzaei & Schenkel 2009; Hickey et al. 2012; Cheng et al. 2015; Pérez-Enciso & Legarra 2016). Currently there is no single self contained software that can simulate complex traits involving both quantitative and fitness components along with the ability to generate complex pedigrees and genomic information ranging from sparse marker to sequence information. Consequently, determining how various selection and management practices impact the fitness and the overall genomic variability of a population undergoing selection for a quantitative trait remains challenging. Furthermore, the precise genetic architecture of complex traits is largely unknown although in general it is assumed that complex traits are affected by variation in a large number of genes, most of which have individually minor effects (Weiss 2008). Knowledge on the demographic history of a population and the extent of linkage disequilibrium is also being generated from dense marker panels or sequence information (McKay et al. 2007; Ai et al. 2013; Porto-Neto et al. 2014). As more information is generated on the genetic architecture of complex traits and variation across the genome, simulation can be employed to more effectively predict how current selection and management practices will impact future generations.

Disclaimer

This document outlines how to run the Geno-Diver simulation program and describes the parameters utilized within the program. The software is free for any person to use. The authors accept no responsibility for the accuracy of results obtained by using Geno-Diver software.

Geno-Diver is being updated with new tools and functions routinely. If you would like the program to do something that it currently does not do, don't hesitate to contact me at jeremy.howard06@gmail.com.

Please notify jeremy.howard06@gmail.com if you think the results are not correct or you have encountered a bug. We have written the program in a way for us to reconstruct the simulation scenario you ran to solve the problem. Lastly, the software was profiled using the Valgrind software (valgrind), and no memory leaks were identified.

The reference for the Geno-Diver software:

- Howard, J.T., F. Tiezzi, J.E. Pryce, C. Maltecca. A combined coalescence forward in time simulator software for pedigreed populations undergoing selection for complex traits. *Journal of Animal Breeding Genetics*, **134(6)**, 553-563.

The reference for the MaCS software:

- Chen, G. K., P. Marjoram & J. D. Wall. 2009. Fast and flexible simulation of DNA sequence data. *Genome Research*, **19**, 136-142.

Examples of how Geno-Diver can be utilized is outlined below in the following manuscript(s):

- Howard et al. 2017. A heuristic method to identify runs of homozygosity associated with reduced performance in livestock, *Journal of Animal Science*.
- Howard et al. 2018. The impact of truncating data on predictive ability for single-step genomic best linear unbiased prediction, *Journal of Animal Breeding Genetics*.
- Howard et al. 2018. The impact of selective genotyping on the response to selection using single-step genomic best linear unbiased prediction, *Journal of Animal Science*.

Download

Source code and executables (Mac and Linux) files are available at the GIT-HUB account.

Executable Files:

- GenoDiver
- macs
- msformatter

Source Code Files:

- ParameterClass.h
- ParameterClass.cpp
- OutputFiles.h
- OutputFiles.cpp
- Global_Population.h
- Global_Population.cpp
- Animal.h
- AnimalFun.cpp
- MatingDesignClasses.h
- MatingDesignClasses.cpp
- SetUpGenome.cpp
- EBV_Functions.cpp
- SelectionCullingFunctions.cpp
- Simulation_Functions.cpp
- PopulationSimulator.cpp
- Genome_ROH.h
- Genome_ROH.cpp
- HaplofinderClasses.h
- HaplofinderClasses.cpp
- zfstream.h
- zfstream.cpp
- makefile

Computing Environment

The code is written in the C++11 language using object-oriented techniques. The application has been tested to run on Linux and MAC platforms. The software makes use of two external libraries, Intel MKL and Eigen.

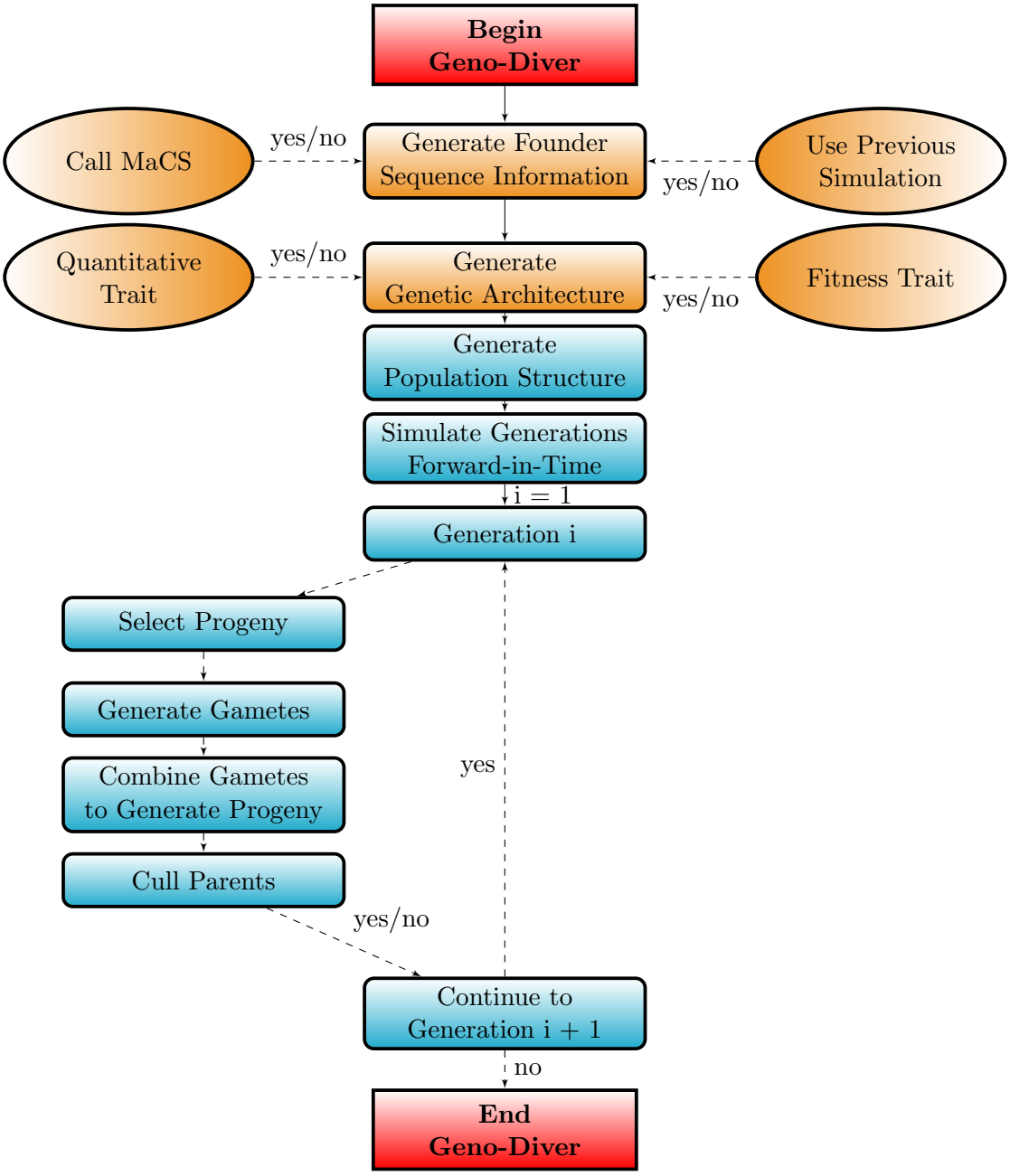
EIGEN Library:
EIGEN is freely available at: Eigen Site
Once at the site, download the latest stable release and uncompress it. For example, the current downloaded package is called "eigen-eigen-07105f7124f9.tar". To use it you just have to place it in the file where all of the other Geno-Diver files are located and uncompress the file. Once you uncompressed the file it will be a folder (e.g. "eigen-eigen-07105f7124f9"). The uncompressed file serves as your path in the makefile outlined below.

Intel MKL Library:
Intel MKL is a commercial library and is available for purchase. However, there is an opportunity to obtain the Intel MKL library (for Linux) free of charge for non-commercial use at the following website: Intel MKL Site
The Intel MKL can sometimes be tricky to download and link, but there is a step-by-step protocol within the folders that are downloaded or instructions are at Intel MKL Guide. Depending on the computing system you are running, a guide to linking the Intel MKL libraries are at Intel MKL Linking Guide.

C++11 Version:
Versions of gcc 4.7 or newer support the C++11 standards. You need to install or update to the correct version of gcc using the standard package manager or installer, depending on what type of OS you are using. Some helpful websites include:
gcc helper 1 gcc helper 2

Compiling:
Once both EIGEN, Intel MKL libraries and gcc version 4.7 or newer have been correctly installed and the folders placed in the directory where all of the Geno-Diver source code files are located the last thing you have to do is change the path for EIGEN and Intel MKL libraries and set which operating system you are using. A makefile has been provided for both a Linux and a MAC operating system. In the makefile change lines 13 and 14 to the path which aligns to the updated EIGEN and MKL path. After changing the path, rename the appropriate file to "makefile" and then type "make" on the command line. An executable file called "GenoDiver" is now in your working directory.

Overview of Program



Running the Program

At the current time executable files are available only for Linux and Mac operating environments and can be obtained by e-mailing 'jeremy.howard06@gmail.com'. The website provides multiple examples on how to simulate different scenario's along with snippets of R code to illustrate the results. To run the program place the following executable files in a folder:

- GenoDiver.
- macs.
- msformatter.

Before running the program, the file permissions need to be checked. After verifying the permissions, a parameter file needs to be generated and placed in the same folder as the previous three files. The parameter file has to be a linux file and not a windows file. If you are new to the simulation program, multiple examples are on the website and is a great place to fully understand the program and all of its available options. The simulation program reads the parameter file by searching for keywords that are capitalized and then followed by a colon. Therefore any phrase that does not meet the search criteria is ignored when initializing parameters within the program. Also, if you want to comment out a parameter just add "!!" within the key word and the program will skip over it. For example to skip over the "SEED" parameter just replace it with "SE!!ED" and it won't recognize the parameter any more.

To run the program type "./GenoDiver" and then the name of your parameter file. For example, if the parameter file is named "parameterfile", the program can be ran by typing "./GenoDiver parameterfile". During the simulation, the program outputs minimal comments on the status of the simulation. A more thorough description of the status of the program is printed to the log file (i.e. "log_file.txt").

After running the program it is a strongly recommended to check the parameters initialized at the top of the log file within the output folder. The log file contains a large amount of information and is a great tool to ensure that the parameters and outcome of the simulation is what is intended. If the program is not running correctly the log file should provide knowledge on why and where the simulation crashed or exited.

If you have problems send an e-mail to jeremy.howard06@gmail.com with a copy of the log file and parameter file.

Program Parameters

The complete list and description of the available options within the software are outlined below and the appendix also contains further discussion and suggestions. In order to make the software as user-frieindly as possible a small portion of the options are required for the program to run. All keywords should be in capital letters and the parameter(s) specified are separated by spaces. Only parameters after the keywords impact the simulation. Lastly, the order in which parameters appear does not matter.

General Parameters

START

Description: - Determines where to start the simulation.

Options:

- sequence: Starts at the sequence generation step.
- founder: Skips sequence generation and begins generating the founder population utilizing sequence information from a previous run.

Usage: - "START: sequence".

Type: - **Mandatory**.

Note:

The use of this option saves time and space due to the large size of sequence information generated from MaCS. If you are using a different effective population size or MaCS diversity metric you have to start at the sequence step, while any parameter file that has the same MaCS parameters can start at the founder step and use the previously generated haplotypes saved within the output folder. When calling multiple replicates, the simulation for any replicate after the first starts at the founder step by default.

SEED

Description: - Declares the seed number.

Option: - Any integer value.

Usage: - "SEED: 1501".

Type: - Optional. Default is the system time.

Note:

- The seed appears at the beginning of the log file if the user does not declare it.

OUTPUTFOLDER

Description: - Directory name of files generated from simulation.
Option: - Any valid folder name, but can't have spaces.
Usage: - "OUTPUTFOLDER: GenoDiverFiles".
Type: - Optional. Default is "GenoDiverFiles".
Note:

If the directory contains information from a previous simulation, it may be deleted depending on the value of the START parameter. If the START parameter is "sequence" everything is deleted. If the START parameter is "founder" everything except files related to generating sequence information are deleted.

THREAD

Description: - Declares the number of threads used for parallel processing.
Option: - Integer value based on number of cores available.
Usage: - "NTHREAD: 4".
Type: - Optional. Default is 1.

REPLICATES

Description:

- Declares the number of replicates to produce. The seed is increased by 1 after each replicate is finished.

Option: - Integer value.
Usage: - "NREP: 1".
Type: - Optional. Default is 1.
Note:

- All important replicate files are in a folder within the output folder path called "replicates". Each file has the seed number appended at the end of the file name to distinguish between replicates.

Genome and Marker Information

CHR

Description: - The number of chromosomes to generate.
Option: - Integer from 1 to 31.
Usage: - "CHR: 3".
Type: - **Mandatory**.

CHR_LENGTH

Description: - The length for each chromosome, in Megabases.
Option: - Integer Value.
Usage: - "CHR_LENGTH: 150 150 150"
Type: - **Mandatory**.
Note:

- The number of Chromosome length variables specified has to correspond to chromosome number.

NUM_MARK

Description:

- The number of markers within a chromosome. The marker location is generated based on a uniform distribution from 0 to the length of the chromosome.

Option: - Integer Value.
Usage: - "NUM_MARK: 4000 4000 4000"
Type: - **Mandatory**.
Note:

- Quantitative and fitness mutations cannot be markers. The program will exit if the number of markers is greater than the available number of mutations within a chromosome.

MARKER_MAF

Description: - Minimum allele frequency allowed for markers.
Option: - Range from 0.0 to 0.50.
Usage: - "MARKER_MAF: 0.05"
Type: - Optional. Default is 0.05.

QTL

Description:

- The number of quantitative trait loci (QTL) for each chromosome. The QTL location is generated based on a uniform distribution from 0 to the length of the chromosome.

Options:

- Integer. Number of QTL and fitness trait loci (FTL) can't exceed 5000.

Usage: - "QTL: 50 50 50"

Type: - **Mandatory**.

QUANTITATIVE_MAF

Description: - Minimum allele frequency allowed for QTL.

Value: - Range from 0.0 to 0.50.

Usage: - "QUANTITATIVE_MAF: 0.01"

Type: - Optional. Default is 0.01.

FIT_LETHAL

Description:

- The number of lethal fitness trait loci (FTL) for each chromosome. The lethal FTL location is generated based on a uniform distribution from 0 to the length of the chromosome.

Option: - Integer. Number of QTL & FTL can't exceed 5000.

Usage: - "FIT_LETHAL: 25 25 25".

Type: - Optional. Default is 0.

Note:

- Lethal FTL can not have a covariance with the quantitative trait. The number of lethal FTL variables specified has to correspond to chromosome number.

FIT_SUBLETHAL

Description:

- The number of sub-lethal fitness trait loci (FTL) for each chromosome. The sub-lethal FTL location is generated based on a uniform distribution from 0 to the length of the chromosome.

Option: - Integer. Number of QTL & FTL can't exceed 5000.

Usage: - "FIT_SUBLETHAL: 25 25 25".

Type: - Optional. Default is 0.

Note:

- Sub-lethal FTL can have a covariance with the quantitative trait. The number of sub-lethal FTL variables specified has to correspond to chromosome number.

FITNESS_MAF

Description:

- Parameters that describe the maximum FTL frequency allowed and the range in lethal FTL allele frequencies. If the parameter is given, the first two are required, while the last one is optional.

Options:

- double (0.0-0.50): Maximum allele frequency for the unfavorable allele allowed for lethal FTL.
- double (0.0-0.50): Maximum allele frequency for the unfavorable allele allowed for sub-lethal FTL.
- double (0.0-0.50): Maximum allele frequency range for the lethal FTL.

Usage: - "FITNESS_MAF: 0.02 0.08 0.01".

Type: - Optional. Default is 0.02, 0.08 and 0.01.

Note:

- Using the default values, the lethal FTL frequencies range from 0.01 to 0.02 and the sub-lethal frequencies range from 0.0 to 0.08. If the frequency of either lethal or sub-lethal FTL is too high, a large number of founders and progeny may not make it to breeding age. If the number of individuals is too small to generate the founder generation the simulation exits.

FOUNDER_HAPLOTYPES

Description:

- The number of haplotypes generated by MaCS.

Option: - Integer Value.

Usage: - "FOUNDER_HAPLOTYPES: 4000".

Type: - Optional. Default is based on male & female number.

HAPLOTYPE_SIZE

Description:
- The number of markers contained within a non-overlapping haplotype window.

Option: - Integer Value.

Usage: - "HAPLOTYPE_SIZE: 50".

Type: - Optional. Default is 50.

RECOMBINATION

Description:
- The type of distribution that generates the location of recombination events along the genome. The number of recombination events is generated from a Poisson distribution. The rate parameter fixed at 1.0 across all chromosomes.

Options:
- Uniform: Recombination sampled from a Uniform (0.0 to length of chromosome).
- Beta: Recombination sampled from a Beta (0.5, 0.5). Recombinations occurs more often at the end than in the middle of the chromosome.

Usage: - "RECOMBINATION: Uniform".

Type: - Optional. Default is Uniform.

MUTATION

Description:
Used in the MaCS software to generate scaled mutation parameter and in the simulation to generate new mutations as generations proceed.

Options:
- Mutation Rate (float): Probability of a new mutation occurring at a given base pair (i.e. infinite-site model). A Poisson distribution with a rate parameter equal to the mutation rate times the length of the chromosome (nucleotides) generates the total number of mutations occurring within a new gamete.
- Proportion of mutations that can be QTL (double): Within the forward-in-time part of the simulation program, represents the number of non-neutral mutations that occurred out of the total number of new mutations. Each trait (i.e. quantitative, lethal, sub-lethal) has an equal chance of being chosen.

Usage: - "MUTATION: 2.5e-8 0.0".

Type: - Optional. Default is 2.5e-8 0.0.

QTL Distributions

ADD_QUAN

Description:

- The parameters for the gamma distribution that generate the un-scaled additive effect for QTL. The effects are scaled to the user specified additive genetic variance. A complete description is in Appendix II.

Options:

- Shape: Shape of gamma distribution.
- Scale: Scale of gamma distribution.

Usage: - "ADD_QUAN: 0.4 1.66".

Type: - Optional. Default is 0.4 1.66.

DOM_QUAN

Description:

- The parameters for the normal distribution that generate the degree of dominance (h) for QTL. The effects are scaled to the user specified dominance genetic variance. A complete description is in Appendix II.

Options:

- Mean: Mean of a normal distribution.
- Standard Deviation (SD): SD of a normal distribution.

Usage: - "DOM_QUAN: 0.1 0.2".

Type: - Optional. Default is 0.1 0.2.

LTHA

Description:

- The parameters for the gamma distribution that generate the selection coefficients for lethal FTL. A complete description is in Appendix II.

Options:

- Shape: Shape of gamma distribution.
- Scale: Scale of gamma distribution.

Usage: - "LTHA: 900.0 0.001".

Type: - Optional. Default is 900.0 0.001.

LTHD

Description:

- The parameters for the normal distribution that generate the degree of dominance for the lethal FTL. A complete description is in Appendix II.

Options:

- Mean: Mean of a normal distribution.
- Standard Deviation (SD): SD of a normal distribution

Usage: - "LTHD: 0.001 0.0001".

Type: - Optional. Default is 0.001 0.0001.

SUBA

Description:

- The parameters for the gamma distribution that generate the selection coefficients for sublethal FTL. A complete description is in Appendix II.

Options:

- Shape: Shape of gamma distribution.
- Scale: Scale of gamma distribution.

Usage: - "SUBA: 0.1 0.2".

Type: - Optional. Default is 0.1 0.2.

SUBD

Description:

- The parameters for the normal distribution that generate the degree of dominance for sublethal FTL. A complete description is in Appendix II.

Options:

- Mean: Mean of a normal distribution.
- Standard Deviation (SD): SD of a normal distribution

Usage: - "SUBD: 0.3 0.1".

Type: - Optional. Default is 0.3 0.1.

COVAR

Description:

- Determines the relationship between the additive effect of the quantitative trait and selection coefficient for the sub-lethal trait. The relationship is a function of the number of QTL that impact the quantitative and sub-lethal trait and the rank correlation between the effects. A complete description is in Appendix II.

Options:

- Proportion of Pleiotropic QTL: Proportion of QTL that have both a quantitative and sub-lethal effect.
- Genetic correlation: Rank correlation between QTL and FTL effects.

Usage: - "COVAR: 0.5 0.2".

Type: - Optional. Default is 0.0 0.0.

Note:

- Fitness values range from 0 to 1 and higher values leading to a lower fitness value. If the two traits are antagonistic under the scenario of high values being favorable, the correlation should be positive. One just needs to change the favorable direction of the quantitative trait to alter the interpretation.

Population Parameters

FOUNDER_Effective_Size

Description:

- Used to generate the population history for the haplotypes generated from MaCS. There are multiple default scenarios and represent a wide range of LD patterns, or one can specify a custom effective population size and population history. Each scenario has a slightly different population history parameter (i.e. "eN"). Lastly, the population history can be a single value with no historical population. A description is in Appendix III.

Options:

- Ne70: A scenario that generates a large amount of short LD.
- Ne100_Scen1: A scenario that generates moderate short LD.
- Ne100_Scen2: A scenario that generates moderate short LD.
- Ne250: A scenario that generates minimal LD.
- Ne1000: A scenario that generates minimal LD.
- CustomNe: Read in own population history parameters.
- Integer: Utilizes the value as the effective population size and no population history assumed.

Usage: - "FOUNDER_Effective_Size: Ne70".
Type: - **Mandatory**.

MALE_FEMALE_FOUNDER
Description:
- Sets the number of founder male and female individuals along with the method founders were selected and how many generations it occurred.

Options:
- Integer value: Number of male founders.
- Integer value: Number of female founders.
- random: At the current time only random selection can be utilized.
- Integer value: Number of generations conducted.

Usage: - "MALE_FEMALE_FOUNDER: 50 600 random 5".
Type: - **Mandatory**.

VARIANCE_A
Description:
- Variance due to additive gene action and/or correlation between additive QTL effects if two traits are simulated.

Options:
- One Trait (double): additive genetic variation.
- Two Trait (double double double): additive genetic variance for trait 1, correlation between additive QTL effects for trait 1 and 2 and additive genetic variance for trait 2.

Usage: - "VARIANCE_A: 0.25".
Type: - **Mandatory**.

VARIANCE_D
Description:
- Variance due to dominant gene action.

Options:
- One Trait (double): dominance genetic variance.
- Two Trait (double double): dominance genetic variance for trait 1 and trait 2.

Usage: - "VARIANCE_D: 0.05".
Type: - Optional. Default is 0.0
Note:

- Care needs to be taken in choosing the additive and dominance variance and its implications on the number of QTL that display over-dominance or partial-dominance. Parameters to adjust include the QTL MAF frequency, mean and standard deviation of the normal distribution for dominance effects and the ratio of additive to dominance variance.

VARIANCE_R

Description:

- Variance due to environmental effects (i.e. residual).

Options:

- One Trait (double): environmental variance.
- Two Trait (double double double): environmental variance for trait 1, correlation between environmental effects for trait 1 and 2 and environmental variance for trait 2.

Usage: - "VARIANCE_R: 0.05".

Type:

- Optional. Default is that total variance is equal to 1.0 and a null correlation between traits if applicable.

Selection and Culling Parameters

GENERATIONS

Description:

- Determines the number of generations to simulate and includes the number of generations chosen with the MALE_FEMALE_FOUNDER option.

Option: - Any integer value.

Usage: - "GENERATIONS: 10".

Type: - **Mandatory**.

Note:

- The number of generations allowed vary depending on the size of your computer memory. It is advisable to start at a small number of generations and then build up to determine how much memory a scenario will use.

INDIVIDUALS

Description:

- Determines the number of males and females in each generation and replacement rate for parents each generation. If the number of parents falls below the user specified male or female value, the program exits.

Options:

- Integer value: Number of male parents within a generation.
- Double value (0.0 to 1.0): Proportion of male parents that are culled and replaced by progeny each generation.
- Integer value: Number of female parents within a generation.
- Double value (0.0 to 1.0): Proportion of female parents that are culled and replaced by progeny each generation.

Usage: - "INDIVIDUALS: 50 0.2 600 0.2".

Type: - **Mandatory**.

PARITY_MATES_DIST

Description:

- Determines the distribution of the number of mating pairs a sire has for each age group. A Beta (Alpha, Beta) is used to generate the distribution of mating pairs for a given age group. To generate the number of mating pairs by age class, the cumulative distribution function (CDF) is split into quadrants based on the number of age classes that occur within a generation. The total number of mating pairs within an age class is the proportion that falls within the CDF quadrant for a given age class.

Options:

- Double value (> 0.0): Alpha parameter in Beta distribution.
- Double value (> 0.0): Beta parameter in Beta distribution.

Usage: - "PARITY_MATES_DIST: 1 1".

Type:

- Optional. The default is both parameters being 1, which is very similar to a uniform distribution, such that all age classes have the same proportion of mating pairs.

PROGENY

Description:

- Determines the number of progeny produced by each mating pair. The number of progeny per mating pair may not be the actual number of progeny produced if FTL segregate in the population. The impact of a progeny fitness value on its ability to make it to breeding age is described in detal in Appendix II.

Option: - Integer value.

Usage: - "PROGENY: 4".

Type: - **Mandatory**.

MAXFULLSIB

Description:

- Determines the maximum number of full-sib progeny selected within a family. Once the maximum number is reached the full-sib with the lowest selection criteria is no longer selected and the next best animal is selected. This process is repeated across all families until all families are below the value.

Option: - Ranges from 1 to number of progeny.

Usage: - "MAXFULLSIB: 2".

Type: - Optional. Default set at number of progeny.

SELECTION

Description:

- The metric used to select offspring to serve as parents the next generation and favorable direction.

Options:

- Selection Criteria:
  - random: random selection.
  - phenotype: phenotypic selection.
  - tbv: selection based on true breeding value (TBV) .
  - ebv: selection based on estimated breeding value (EBV).
  - index_tbv: selection based on an index for the TBV across both traits.
  - index_ebv: selection based on an index for the EBV across both traits.
- Direction:
  - high: High values are favorable.
  - low: Low values are favorable.

Usage: - "SELECTION: ebv high".
Type: - **Mandatory**.
Note:

- For random selection, the direction does not impact the results and therefore the direction option does not matter. For index_tbv and index_ebv two traits have to be simulated. The index for animal 'i' based on TBV is constructed as follows:

$$\text{Index}_\text{i} = \frac{TBV_{Trait1}}{\sigma_{TBV_{Trait1}}} * weight1 + \frac{TBV_{Trait2}}{\sigma_{TBV_{Trait2}}} * weight2,$$

where weight1 and weight2 are the index weights for Trait1 and Trait2, respectively. The index based on EBV is the same except EBV are used instead of TBV. The standard deviation for Trait 1 and 2 is calculated from animals born a generation before truncation selection begins.

INDEX_PROPORTIONS
Description:

- Sets the weight that each trait is given when constructing the index true or estimated breeding value when selection is based on index_tbv or index_ebv.

Options:

- Double value : Weight given to Trait 1.
- Double value : Weight given to Trait 2.

Usage: - "INDEX_PROPORTIONS: 0.2 0.8".
Type: - Mandatory if selection based on 'index_tbv' or 'index_ebv'.
Note:

- The index weights must be such that their sum is equal to 1 and each weight is between 0.0 and 1.0.

EBV_METHOD:
Description:

- Parameter that specifies how estimated breeding values (EBV) are generated.

Options:

- Method:

  - pblup: The best linear unbiased prediction (BLUP) of EBV are obtained by Henderson's (Henderson 1975) mixed model equations. The mixed model equations are outlined below:

$$\begin{bmatrix} X'X & X'Z \\ Z'X & Z'Z + A^{-1}(\frac{\sigma_e^2}{\sigma_a^2}) \end{bmatrix} \begin{bmatrix} \hat{b} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} X'y \\ Z'y \end{bmatrix},$$

where X and Z refers to fixed and random design matrices, which relate records to fixed and random effects, $\mathbf{A^{-1}}$ refers to the inverse of the pedigree based relationship matrix. Fixed effects include the intercept and random effects include the effect of the animal. The mixed model equations are solved by the preconditioned conjugate gradient method or cholesky decomposition of the LHS of the matrix.
- gblup: The same as the pblup option, EBV are obtained by solving the mixed model equations, except the inverse of the pedigree based relationship matrix is replaced by the inverse of the genomic based relationship matrix.
- rohblup: The same as the pblup option, EBV are obtained by solving the mixed model equations, except the inverse of the pedigree based relationship matrix is replaced by the inverse of the run-of-homozygosity based relationship matrix.
- ssgblup: The same as the pblup option, EBV are obtained by solving the mixed model equations, except the inverse of the pedigree based relationship matrix is replaced by the inverse of a relationship matrix that augments pedigree relationships across all individuals based on contributions from animals with genomic information ($\mathbf{H^{-1}}$; Aguilar et al., 2010; Christensen & Lund, 2010).
- bayes: Marker effects and the associated EBV are estimated utilizing bayesian whole genome regression models based on the following model:

$$y_{\mathrm{i}} = \mu + \sum_{j=1}^{j=m} SNP_{\mathrm{j}} + \epsilon_{\mathrm{i}},$$

where y is the phenotype for individual$_{\mathrm{i}}$, μ is the intercept, SNP is the additive genetic effects that correspond to allele substitution effects for each marker and ϵ is the residual for individual $_{\mathrm{i}}$. SNP covariates had values of 0 for the homozygote, 1 for the heterozygote and 2 for the alternative homozygote. The intercept is assigned an uninformative prior and the SNP covariates can have 4 possible prior densities: 1.) Gaussian (BayesRidgeRegression); 2.) Scaled-t (BayesA); 3.) Two finite mixture priors: a mixture of a point of mass at zero and a Gaussian slab (BayesC); 4.) Two finite mixture priors: a mixture of a point of mass at zero and a Scaled-t slab (BayesB). Samples were drawn from the posterior density using a Gibbs

sampler with scalar updating.

Usage: - "EBV_METHOD: pblup'.

Type: - Only mandatory if selection is based on ebv.

Note:

- If left out and selection is not based on ebv the program will not calculate breeding values. If parameter included and selection is not based on ebv, the program will calculate breeding values.

BLUP_OPTIONS:

Description:

- When EBV are estimated based on 'pblup', gblup', 'rohblup' or 'ssgblup' this parameter can be included to remove older individuals from the analysis, specify solving method and how the inverse of the relationship matrix is calculated.

Options:

- Generations (integer): Number of generations to trace back from the current group of selection candidates. All animals 'n' generations back and all of the progeny of the associated parents will be included in the analysis. For example, if a value of 1 is specified the selection candidate's parents and all progeny that were parents of the selection candidates parents will only be included in the analysis. The default value is the number of generations simulated (i.e. all animals are used each generation).
- Solver:
    - direct: Uses Cholesky decomposition for matrix inversion. If EBV accuracy is needed or only simulating a small number of generations this is advised.
    - pcg: Uses the iterative preconditioned conjugate gradient (PCG) method and is faster than direct when the number of animals is large. The EBV accuracy is not calculated **(Default)**.
- Genomic Inverse (only applicable if 'gblup' is specified):
    - cholesky: Update previous Cholesky inverse (Meyer et al. 2012; **Default**).
    - recursion: Sequential update (Misztal et al. 2014).

Usage: - "BLUP_OPTIONS: 4 pcg cholesky".

Type: - Optional.

Note:

Parameters that specify inverse calculations are for genomic-based relationships. Calculation of pedigree-based relationships is generated based on Meuwissen & Luo (1992).

G_OPTIONS:

Description:

- When EBV are estimated based on 'gblup' this parameter can be included to determine how the genomic relationship matrix is calculated.

Options:

- How genomic relationship is constructed:
  - VanRaden: Constructed based on G=ZZ'/ 2Σ(pq).
- Allele frequencies calculated:
  - founder: Frequencies calculated based on founder genome.
  - current population: Frequencies calculated based on animals who are selection candidates..

Usage: - "G_OPTIONS: VanRaden observed".

Type: - Optional. Default is VanRaden and founder allele frequencies.

BLENDING_GA22

Description:

- When using the 'ssgblup' option, a weighted genomic relationship matrix (Gw), as proposed by VanRaden (2008), is utilized when constructing the H matrix. Gw is constructed as Gw = weight1 * G + weight * A22, where G is the raw genomic relationship matrix and A22 is the pedigree-based relationship for genotyped animals.

Options:

double value: Weight given to the G.
double value: Weight given to the A22.

Usage: - "BLENDING_GA22: 0.95 0.05".

Type: - Optional. Default values are 0.95 for G and 0.05 for A22

BAYESOPTIONS:

Description:

- When ebv are estimated based on the "bayes" option, this parameter has to be included to determine which method to utilize and MCMC options.

Options:

- Bayesian Model:

BayesRidgeRegression: Gaussian prior.
BayesA: Scaled-t prior.
BayesB: Spike with a point of mass at 0 and slab with the slab having a scaled-t prior.
BayesC: Spike with a point of mass at 0 and slab with the slab having a gaussian prior.

- Integer value: Number of MCMC iterations the first time a method is implemented (Required).
- Integer value: Number of MCMC iterations to burn-in the first time a method is implemented (Required).
- Integer value: Number of MCMC iterations for any generation after the first time a method is implemented (Required).
- Integer value: Number of MCMC iterations to burn-in for any generation after the first time a method is implemented (Required).
- Integer value: Thinning rate for MCMC iterations (Required).
- How pie is determined: (Only needed for BayesB & BayesC)
  - fix: Fix proportion of SNP with non-null effect.
  - estimate: Estimate proportion of SNP with non-null effect.
- Double value: Proportion of markers with non-null effect. (Only needed for BayesB & BayesC)
- Integer value: Number of generations to trace back from the current group of selection candidates. All animals 'n' generations back and all of the progeny of the associated parents will be included in the analysis. For example, if a value of 1 is specified the selection candidate's parents and all progeny that were parents of the selection candidates parents will only be included in the analysis. The default value is the number of generations simulated (i.e. all animals are used each generation) (Optional).

Usage: - "BAYESOPTIONS: BayesC 10000 2000 6000 1000 5 fix 0.10".

Type:

- If 'bayes' EBV option is utilized has to be included otherwise can't be in parameter file.

Note:

The number of iterations to run can be investigated by looking at the 'bayes_mcmc_samples' files within the output directory. For each generation the files gets replaced by the most recent results.

CULLING

Description:

- The metric used to cull parents and maximum age an animal can remain in the population before being removed due to old age. The direction is the same as the selection direction.

Options:

- Culling Criteria:
  - random: random culling.
  - phenotype: phenotypic culling.
  - tbv: culling based on true breeding value (TBV) .
  - ebv: culling based on estimated breeding value (EBV).
  - index_tbv: culling based on an index for the TBV across both traits.
  - index_ebv: culling based on an index for the EBV across both traits.
- Age Removed:
  - Integer: Age at which an animal has to be removed from the population.

Usage: - "CULLING: ebv 5".

Type: - **Mandatory**.

INTERIM_EBV

Description:

- Determines whether to calculate interim estimated breeding values (EBV) before or after culling of parents.

Options:

- no: Interim EBV are not calculated.
- before_culling: Interim EBV calculated prior to culling parents.
- after_culling: Interim EBV calculated after culling parents.

Usage: - "INTERIM_EBV: no".

Type: - Optional. Default is 'no'.

Genotyping and Phenotyping Option Parameters

PHENOTYPE_STRATEGY

Description:

- When simulating 1 quantitative trait, determines when and if an animal has a phenotype when estimated breeding value (ebv) are being predicted.

Options:

- Male Proportion Phenotyped:
  - Double value (0.0 to 1.0): Proportion of males that are phenotyped based on the male phenotyping strategy.
- Male Phenotyping Strategy:
  -pheno_atselection: Phenotype selection candidates prior to predicting EBV.
  -pheno_afterselection: Phenotype selection candidates after predicting EBV.
  -pheno_parents: Only phenotype selection candidates that were selected to serve as parents.
  -random_atselection: Randomly phenotype selection candidates prior to predicting EBV.
  -random_afterselection: Randomly phenotype selection candidates after predicting EBV.
  -random_parents: Randomly phenotype selection candidates that were selected to serve as parents.
  -ebv_atselection: Phenotype selection candidates based on EBV prior to predicting EBV.
  -ebv_afterselection: Phenotype selection candidates based on EBV after predicting EBV.
  -ebv_parents: Phenotype selection candidates that were selected based on EBV to serve as parents.
  -litterrandom_atselection: Randomly phenotype within each litter prior to predicting EBV.
  -litterrandom_atselection: Randomly phenotype within each litter after predicting EBV.
- Female Proportion Phenotyped:
  - Double value (0.0 to 1.0): Proportion of females that are phenotyped based on the female phenotyping strategy.
- Female Phenotyping Strategy:
  -pheno_atselection: Phenotype selection candidates prior to predicting EBV.

-pheno_afterselection: Phenotype selection candidates after predicting EBV.
-pheno_parents: Only phenotype selection candidates that were selected to serve as parents.
-random_atselection: Randomly phenotype selection candidates prior to predicting EBV.
-random_afterselection: Randomly phenotype selection candidates after predicting EBV.
-random_parents: Randomly phenotype selection candidates that were selected to serve as parents.
-ebv_atselection: Phenotype selection candidates based on EBV prior to predicting breeding values.
-ebv_afterselection: Phenotype selection candidates based on EBV after predicting breeding values.
-ebv_parents: Phenotype selection candidates that were selected based on EBV to serve as parents.
-litterrandom_atselection: Randomly phenotype within each litter prior to predicting EBV.
-litterrandom_atselection: Randomly phenotype within each litter after predicting EBV.

- Distribution sampling from when doing EBV based phenotype strategy:

-high: Phenotype animals with a high EBV.
-low: Phenotype animals with a low EBV.
-tails: Phenotype animals with a high and low EBV.

Usage: - "PHENOTYPE_STRATEGY: 1.0 atselection 1.0 atselection".

Type:

- Optional. The default setting is all male and female selection candidates are phenotyped prior to selection.

Note:

For the pheno_* the proportion has to be 0.0 or 1.0, while any other scenario has to be greater than 0.0 and less than 1.0. Both male and female proportions can't be 0.0.

PHENOTYPE_STRATEGY1

Description:

- When simulating 2 quantitative traits, determines when and if an animal has a phenotype when estimated breeding value (ebv) are being predicted for trait 1.

Options:

- Male Proportion Phenotyped:

- Double value (0.0 to 1.0): Proportion of males that are phenotyped based on the male phenotyping strategy.

- Male Phenotyping Strategy:

  -pheno_atselection: Phenotype selection candidates prior to predicting EBV.
  -pheno_afterselection: Phenotype selection candidates after predicting EBV.
  -pheno_parents: Only phenotype selection candidates that were selected to serve as parents.
  -random_atselection: Randomly phenotype selection candidates prior to predicting EBV.
  -random_afterselection: Randomly phenotype selection candidates after predicting EBV.
  -random_parents: Randomly phenotype selection candidates that were selected to serve as parents.
  -ebv_atselection: Phenotype selection candidates based on EBV prior to predicting EBV.
  -ebv_afterselection: Phenotype selection candidates based on EBV after predicting EBV.
  -ebv_parents: Phenotype selection candidates that were selected based on EBV to serve as parents.
  -litterrandom_atselection: Randomly phenotype within each litter prior to predicting EBV.
  -litterrandom_atselection: Randomly phenotype within each litter after predicting EBV.

- Female Proportion Phenotyped:

  - Double value (0.0 to 1.0): Proportion of females that are phenotyped based on the female phenotyping strategy.

- Female Phenotyping Strategy:

  -pheno_atselection: Phenotype selection candidates prior to predicting EBV.
  -pheno_afterselection: Phenotype selection candidates after predicting EBV.
  -pheno_parents: Only phenotype selection candidates that were selected to serve as parents.
  -random_atselection: Randomly phenotype selection candidates prior to predicting EBV.
  -random_afterselection: Randomly phenotype selection candidates after predicting EBV.
  -random_parents: Randomly phenotype selection candidates that were selected to serve as parents.
  -ebv_atselection: Phenotype selection candidates based on EBV prior to predicting EBV.

-ebv_afterselection: Phenotype selection candidates based on EBV after predicting EBV.
-ebv_parents: Phenotype selection candidates that were selected based on EBV to serve as parents.
-litterrandom_atselection: Randomly phenotype within each litter prior to predicting EBV.
-litterrandom_atselection: Randomly phenotype within each litter after predicting EBV.

- Distribution sampling from when doing EBV based phenotype strategy:

-high: Phenotype animals with a high EBV.
-low: Phenotype animals with a low EBV.
-tails: Phenotype animals with a high and low EBV.

Usage:

- "PHENOTYPE_STRATEGY1: 1.0 atselection 1.0 atselection".

Type:

- Optional. The default setting is all male and female selection candidates are phenotyped prior to predicting breeding values for the first trait.

Note:

For the pheno_* the proportion has to be 0.0 or 1.0, while any other scenario has to be greater than 0.0 and less than 1.0. Both male and female proportions can't be 0.0.

PHENOTYPE_STRATEGY2

Description:

- When simulating 2 quantitative traits, determines when and if an animal has a phenotype when estimated breeding value (ebv) are being predicted for trait 2.

Options:

- Male Proportion Phenotyped:

  - Double value (0.0 to 1.0): Proportion of males that are phenotyped based on the male phenotyping strategy.

- Male Phenotyping Strategy:

-pheno_atselection: Phenotype selection candidates prior to predicting EBV.
-pheno_afterselection: Phenotype selection candidates after predicting EBV.
-pheno_parents: Only phenotype selection candidates that were selected to serve as parents.

-random_atselection: Randomly phenotype selection candidates prior to predicting EBV.
-random_afterselection: Randomly phenotype selection candidates after predicting EBV.
-random_parents: Randomly phenotype selection candidates that were selected to serve as parents.
-ebv_atselection: Phenotype selection candidates based on EBV prior to predicting EBV.
-ebv_afterselection: Phenotype selection candidates based on EBV after predicting EBV.
-ebv_parents: Phenotype selection candidates that were selected based on EBV to serve as parents.
-litterrandom_atselection: Randomly phenotype within each litter prior to predicting EBV.
-litterrandom_atselection: Randomly phenotype within each litter after predicting EBV.

- Female Proportion Phenotyped:

  - Double value (0.0 to 1.0): Proportion of females that are phenotyped based on the female phenotyping strategy.

- Female Phenotyping Strategy:

  -pheno_atselection: Phenotype selection candidates prior to predicting EBV.
  -pheno_afterselection: Phenotype selection candidates after predicting EBV.
  -pheno_parents: Only phenotype selection candidates that were selected to serve as parents.
  -random_atselection: Randomly phenotype selection candidates prior to predicting EBV.
  -random_afterselection: Randomly phenotype selection candidates after predicting EBV.
  -random_parents: Randomly phenotype selection candidates that were selected to serve as parents.
  -ebv_atselection: Phenotype selection candidates based on EBV prior to predicting EBV.
  -ebv_afterselection: Phenotype selection candidates based on EBV after predicting EBV.
  -ebv_parents: Phenotype selection candidates that were selected based on EBV to serve as parents.
  -litterrandom_atselection: Randomly phenotype within each litter prior to predicting EBV.
  -litterrandom_atselection: Randomly phenotype within each litter after predicting EBV.

- Distribution sampling from when doing EBV based phenotype

strategy:

-high: Phenotype animals with a high EBV.
-low: Phenotype animals with a low EBV.
-tails: Phenotype animals with a high and low EBV.

Usage: - "PHENOTYPE_STRATEGY2: 1.0 atselection 1.0 atselection".

Type:

- Optional. The default setting is all male and female selection candidates are phenotyped prior to predicting breeding values for the second trait.

Note:

For the pheno_* the proportion has to be 0.0 or 1.0, while any other scenario has to be greater than 0.0 and less than 1.0. Both male and female proportions can't be 0.0.

GENOTYPE_STRATEGY

Description:

- If doing 'ssgblup' based BLUP determines who and at what time an animal is genotyped.

Options:

- Generation (integer value): Generation to begin genotyping.
- Male Proportion Genotyped:
  - Double value (0.0 to 1.0): Proportion of males that are genotyped based on the male genotyping strategy.
- Male Genotyping Strategy:

  -parents: Genotype all parents.
  -offspring: Genotype all selection candidates.
  -parents_offspring: Genotype all parents and selection candidates
  -random: Randomly genotype a proportion of the selection candidates.
  -parents_random: Randomly genotype a proportion of the selection candidates and all selection candidates that were selected to become parents.
  -ebv: Genotype a proportion of the selection candidates based on EBV.
  -parents_ebv: Genotype a proportion of the selection candidates based on EBV and all selection candidates that were selected to become parents.
  -litter_random: Genotype a random proportion of the animals within each litter.

-litter_parents_random: Genotype a random proportion of the animals within each litter and all selection candidates that were selected to become parents.

- Female Genotyping Strategy:

- Double value (0.0 to 1.0): Proportion of females that are genotyped based on the female genotyping strategy.

- Female Phenotyping Strategy:

-parents: Genotype all parents.
-offspring: Genotype all selection candidates.
-parents_offspring: Genotype all parents and selection candidates
-random: Randomly genotype a proportion of the selection candidates.
-parents_random: Randomly genotype a proportion of the selection candidates and all selection candidates that were selected to become parents.
-ebv: Genotype a proportion of the selection candidates based on EBV.
-parents_ebv: Genotype a proportion of the selection candidates based on EBV and all selection candidates that were selected to become parents.
-litter_random: Genotype a random proportion of the animals within each litter.
-litter_parents_random: Genotype a random proportion of the animals within each litter and all selection candidates that were selected to become parents.

- Distribution sampling from when doing EBV based genotyping strategy:

-high: Genotype animals with a high EBV.
-low: Genotype animals with a low EBV.
-tails: Genotype animals with a high and low EBV.

Usage: - "GENOTYPE_STRATEGY: 1.0 parents 1.0 parents".

Type:

- Mandatory if doing 'ssgblup' option.

Note:

For the parents, offspring and parents_offspring option the proportion has to be 0.0 or 1.0, while any other scenario has to be greater than 0.0 and less than 1.0. The distribution to sample is not required and if not provided it will be the direction animals are selected on.

Mating Parameters

MATING

Description: - Parameters that decide how animals are mated and the algorithm utilized to optimize matings.

Options:

- Mating Option:

  -random: Males and female parents are randomly mated.
  -random5: Relationships ≥ 0.5 are not allowed to mate otherwise same as random mating option.
  -random25: Relationships ≥ 0.25 are not allowed to mate otherwise same as random5 mating option.
  -random125: Relationships ≥ 0.125 are not allowed to mate otherwise same as random25 mating option.
  -minPedigree: Minimize parental co-ancestries based on a pedigree relationship matrix.
  -minGenomic: Minimize parental co-ancestries based on a genomic relationship matrix.
  -minROH: Minimize parental co-ancestries based on a ROH relationship matrix.
  -pos_assort: Positive assortative mating based on the value animals are being selected on.
  -neg_assort: Negative assortative mating based on the value animals are being selected on.
  -index: Mating is done based on an index of multiple values and is parameterized by 'MATING_INDEX' option.

- Optimization Method:

  -simu_anneal: Adaptive simulated annealing algorithm to assign mates (Ingber 1993).
  -linear_prog: Uses the Hungarian algorithm to find an exact solution to the best possible mating design.
  -sslr: Sequential selection of least related mates algorithm to assign mates (Pryce et al. 2011).
  -gp: Genetic programming to assign mating pairs (Chu & Beasley, 1997).

Usage: - "MATING: minPedigree sslr".

Type:

- **Mandatory**. Only the first option is required for random, pos_assort, and neg_assort.

Note:

For avoidance mating's (i.e. random5, random25, random 125) any coancestry below the threshold is zeroed out and mating are optimized. If utilizing the index mating parameter, the only options are can be utilized are 'gp' or 'sslr'. The 'gp' option can only be used for index mating.

Output Options

OUTPUT_LD

Description:

- Determines if the linkage dis-equilibrium decay based on the r2 metric for each generation across the genome and surrounding QTL is calculated each generation.

Option: - yes or no.

Usage: -"OUTPUT_LD: no".

Type: - Optional. Default is no.

GENOTYPES

Description:

- Determines whether genotypes are saved and at what generation to begin saving them.

Options:

- No or yes and if yes provide the generation at which to start exporting genotypes. Generation 0 is founder population.

Usage: - "GENOTYPES: yes 0".

Type: - Optional. Default is yes and starting at generation 0.

GENOME_ROH

Description:

- Calculates the proportion of the genome in a ROH for each individual. Also the frequency of a SNP being in a ROH and the length of ROH the SNP is contained in is calculated across the genome for a specified generation number.

Options:

- ROH cutoff
    - Mb (integer): ROH cutoff in Megabases.
- Generation number(s) ROH frequency and length are calculated:

- Generation (integer): The generations that the ROH statistics across the genome are calculated. Each generation is separated by a space.

Usage: - "GENOME_ROH: 5 5 10 15".
Type: - Optional.

WINDOWQTLVAR

Description:
- Calculates the additive ($\sum 2pq[a+d(q-p)]^2$) and dominance ($\sum(2pqd)^2$) variance within 1Mb non-overlapping windows for each generation.

Option: - yes or no.
Usage: -"WINDOWQTLVAR: no".
Type: - Optional. Default is no.

TRAINREFER_STATS

Description:
- For each generation selection has occurred, the estimated breeding value (EBV) along with their associated true breeding value (TBV) are outputted to a file for the parents and selection candidates. The output can be used to generate the accuracy and bias of EBV across generations.

Option: - yes or no.
Usage: -"TRAINREFER_STATS: no".
Type: - Optional. Default is no.

Appendix I - Output Information

The folder that contains the information generated by the simulation program contains multiple files, and a description of each one is below. Depending on the number of quantitative traits that are simulated the output files will have different columns. Outlined below is the output file when 2 quantitative traits are simulated, if only one quantitative trait is simulated the columns that refer to the second trait will not be included in the output files and the 1 will be removed from the column headings regarding the first trait.

Data Summary Files

Summary_Statistics_DataFrame_Performance:

Generation: Generation number.
phen1: Mean (variance) phenotypic value for trait 1.
ebv1: Mean (variance) estimated breeding value for trait 1.
tgv1: Mean (variance) true genotypic breeding value for trait 1.
tbv1: Mean (variance) true breeding breeding value for trait 1.
tdd1: Mean (variance) true dominance deviation for trait 1.
res1: Mean (variance) residual value for trait 1.
phen2: Mean (variance) phenotypic value for trait 2.
ebv2: Mean (variance) estimated breeding value for trait 2.
tgv2: Mean (variance) true genotypic breeding value for trait 2.
tbv2: Mean (variance) true breeding breeding value for trait 2.
tdd2: Mean (variance) true dominance deviation for trait 2.
res2: Mean (variance) residual value for trait 2.
index_tbv: Mean (variance) index true breeding value.

Summary_Statistics_DataFrame_Inbreeding:

Generation: Generation number.
ped_f: Mean pedigree based inbreeding parameter.
gen_f: Mean genomic relationship diagonal constructed based on Van Raden (2008).
h1_f: Mean diagonal of haplotype based relationship matrix (Hickey et al. 2012; H1).
h2_f: Mean diagonal of haplotype based relationship matrix (Hickey et al. 2012; H2).
h3_f: Mean diagonal of ROH based relationship matrix (Howard et al. 2017).
homozy: Mean proportion homozygous (i.e. 1 - homozy = Observed Heterozygosity).
PropROH: Mean proportion of the genome in ROH of a given length.
ExpHet: Expected Heterozygosity (i.e. Σ (1 - p^2 - q^2)).
fitness: Mean multiplicative fitness value of an individual.
homozlethal: Mean number of homozygous FTL classified as lethal.
hetezlethal: Mean number of heterozygous FTL classified as lethal.
homozysublethal: Mean number of homozygous FTL classified as sub-lethal.
hetezsublethal: Mean number of heterozygous FTL classified as sub-lethal.
lethalequiv: Mean lethal equivalents (Lethal equivalents = Σ s for an animal).

Summary_Statistics_QTL

Generation: Generation number.
Quant_Founder_Start: Number of QTL from founder generation segregating.
Quant_Founder_Lost: Number of QTL from founder generation fixed.
Mutation_Quan_Total: Number of QTL from new mutations segregating.
Mutation_Quan_Lost: Number of QTL from new mutations fixed.
Fit_Founder_Start: Number of FTL from founder generation segregating.
Fit_Founder_Lost: Number of FTL derived from founder generation fixed.
Mutation_Fit_Total: Number of FTL derived from new mutations segregating.
Mutation_Fit_Lost: Number of FTL derived from new mutations fixed.
Avg_Haplotypes_Window: Mean haplotypes contained within a haplotype window.
ProgenyDiedFitness: Number of progeny that died due to fitness.

LD_Decay:

A file that has the average correlation (r^2) between two SNP across a range of distances. The average was generated by moving across the genome in 10 Mb blocks and randomly grabbing two SNP and calculating their respective (r^2) and placing them in the correct bins based how far they were apart. Within a block 500 pairs of SNP are randomly sampled and once finished the window is shifted by 5 Mb and is repeated until the end of the chromosome. This is conducted within each chromosome. The distances are in the first row and are in Kilobases. Each row after the first row corresponds to the generation, such that line 2 is generation 0, line 3 is generation 1, etc. The formula to calculate the D and (r^2) values is below and the subscript refers to either SNP marker 1 or 2.

$$D = (A_1B_1 \times A_2B_2) - (A_1B_2 \times A_2B_1)$$

$$r^2 = \frac{D^2}{p_1(1-p_1)p_2(1-p_2)}$$

QTL_LD_Decay:

Similar to the "LD_Decay" file this file estimates the average correlation between a SNP and a QTL within window sizes of 0 - 0.5 Mb, 0.5 - 1.0 Mb, 1.0 - 1.5 Mb, 1.5 - 2.0 Mb and 2.0 - 2.5 Mb. For example if a SNP and QTL were 0.75 Mb apart it would get placed in the 0.5 - 1.0 Mb bin. Each line in the file represents a QTL. If the correlation could not be estimated due to lack of SNP in the window or if SNP are near fixation a '-5' will be produced. The first column represents the chromosome and the second column represents the position in Mb. The last column is the average correlation within a generation across the different window sizes. Correlations within a generation across windows are separated by a ":" and sets of correlations across generations are seperated by a "_".

Phase_Persistance:

The phase was calculated as the square root of (r^2) between a SNP and a QTL and was the same sign as D (de Roos et al. 2008). Each row represents a QTL with a particular SNP. The first and second column is the QTL and marker location. The third column determines whether the value is used in the correlation in phase across generation. Once a phase can't be estimated it can no longer be in the correlation calculation. The remaining columns is the phase estimate for each generation.

Phase_Persistance_Generation:

Using the output from the Phase_Persistance file, the correlation between phases across generations was estimated within window sizes of 0 - 0.5 Mb, 0.5 - 1.0 Mb, 1.0 - 1.5 Mb, 1.5 - 2.0 Mb and 2.0 - 2.5 Mb. The first column is the generation and the columns following would be the correlation between the current generation and preceding generations.

TrainReference:

ID: ID of individual.
T1_EBV: Estimated breeding value for trait 1.
T1_TBV: True breeding value for trait 1.
T2_EBV: Estimated breeding value for trait 2.
T2_TBV: True breeding value for trait 2.
Generation: Generation EBV was calculated.
Group: Whether animal was a parent or a selection candidate.

AmaxGeneration:

This file estimates the mean maximum relationship for individuals that were born in a previous generation with selection candidates.

Summary_Statistics_ROH_Freq:

The first two columns are the chromosomal and nucleotide position of the SNP and the remaining columns are the frequency of that SNP being in an ROH of the length that was specified for a given generation. A SNP may not be in a window of a given length and therefore is set to -5.

Summary_Statistics_ROH_Length:

The first two columns are the chromosomal and nucleotide position of the SNP and the remaining columns are the mean and median length (i.e. "Mean_Median") of ROH for that given SNP when that SNP is in a ROH for a given generation. A SNP may not be in a window of a given length and therefore is set to -5.

WindowAdditiveVariance:

Provides the true additive genetic variance for a given 1-Mb window across generations. The first row is the chromosome and mid-point in the window and the remaining lines are the associated additive genetic variance estimates for a given window.

WindowDominanceVariance:

Provides the true dominance genetic variance for a given 1-Mb window across generations. The first row is the chromosome and mid-point in the window and the remaining lines are the associated dominance genetic variance estimates for a given window.

Data Files

log_file.txt:

This file displays a great deal of information on specifics within each generation and it is advisable that one should look over it after you try a new simulation protocol.

Master_DataFrame:

A file that contains multiple statistics for individuals that survived. If only one quantitative trait is simulated, columns for trait 2 are not displayed.
ID: ID of individual.
Sire: Sire Identification of individual.
Dam: Dam Identification of individual.
Sex: Sex of individual (0 = male and 1 = female).
Gen: Generation the animal was born.
Age: Age the animal was removed from the population.
Progeny: Number of progeny.
Dead: Number of dead progeny.
Ped_F: Pedigree based inbreeding metric.
Gen_F: Diagonal of genomic based relationship constructed based on Van Raden (2008).
Hap1_F: Diagonal of haplotype 1 based relationship matrix.
Hap2_F: Diagonal of haplotype 2 based relationship matrix.
Hap3_F: Diagonal of ROH based relationship matrix.
Homolethal: Number of homozygous lethal genotypes.
Heterlethal: Number of heterozygous lethal genotypes.
Homosublethal: Number of homozygous sub-lethal genotypes.
Hetersublethal: Number of heterozygous sub-lethal genotypes.
Letequiv: Lethal equivalent value.
Homozy: Proportion of the genome homozygous.
PropROH: Proportion of the genome in an ROH of a given length.
Fitness: Multiplicative Fitness value of the individual.
Phen1: Phenotype for trait 1.
EBV1: Estimated breeding value for trait 1.
Acc1: Accuracy of estimated breeding value for trait 1.
TGV1: True genotypic value for trait 1 (Σ (a + d)).
TBV1: True breeding value for trait 1 (Σ a).
TDD1: True dominance deviation for trait 1 (Σ d).
R1: Residual value for trait 1.
Phen2: Phenotype for trait 2.
EBV2: Estimated breeding value for trait 2.
Acc2: Accuracy of estimated breeding value for trait 2.
TGV2: True genotypic value for trait 2 (Σ (a + d)).
TBV2: True breeding value for trait 2 (Σ a).
TDD2: True dominance deviation for trait 2 (Σ d).
R2: Residual value for trait 2.

Master_Genotypes.gz:
A zipped file that contains genotypic information for individuals that survived.
ID: ID of individual.
Marker: Marker genotypes of individual (0-11; 2-22; 3-12; 4-21).
QTL: QTL genotypes of individual (0-11; 2-22; 3-12; 4-21).

Marker_Map:
chr: Chromosome location.
pos: Nucleotide position of marker.

QTL_new_old_Class:
A file that contains information on QTL effects and frequency across generations. If only one quantitative trait is simulated, columns for trait 2 are not displayed.
Chr: Chromosome location.
Pos: Nucleotide position of QTL.
Additive_Selective1: If it is a QTL it refers to the additive effect and if it is a FTL it refers to the selection coefficient for trait 1.
Dominance2: The dominance effect for the QTL or degree of dominance for FTL for trait 1.
Additive_Selective2: If it is a QTL it refers to the additive effect and if it is a FTL it refers to the selection coefficient for trait 2.
Dominance2: The dominance effect for the QTL or degree of dominance for FTL for trait 2.
Type: Refers to the type of loci (2 = quantitative trait; 4 = fitness lethal; 5 = fitness sub-lethal).
Gen: Generation at which the mutation occured.
Freq: Allele frequency in the progeny. Each generation is separated by a underscore.

Low_Fitness:
This file has a number of metrics for each individual that did not survive to breeding age.
Sire: ID of individual.
Dam: Dam ID of individual.
Gen: Generation the animal was born.
Ped_F: Pedigree based inbreeding metric.
Gen_F: Diagonal of genomic based relationship constructed based on Van Raden (2008).
Hap3_F: Diagonal of ROH based relationship matrix.
Homozy: Proportion of the genome homozygous.
Homolethal: Number of homozygous lethal genotypes.
Heterlethal: Number of heterozygous lethal genotypes.
Homosublethal: Number of homozygous sub-lethal genotypes.
Hetersublethal: Number of heterozygous sub-lethal genotypes.
Letequiv: Lethal equivalent value.
Fitness: Multiplicative Fitness value of the individual.
TGV1: True genotypic value of individual for trait 1 (Σ (a + d)).
TBV1: True breeding value of individual for trait 1 (Σ a).
TDD1: True dominance deviation of individual for trait 1 (Σ d).
TGV2: True genotypic value of individual for trait 2 (Σ (a + d)).
TBV2: True breeding value of individual for trait 2 (Σ a).
TDD2: True dominance deviation of individual for trait 2 (Σ d).

QTL_Fitness: QTL genotypes of individual (0-11; 2-22; 3-12; 4-21).

Animal_GenoPheno_Status:

This file outlines when and if an animal was genotyped along with whether they have a phenotype.
ID: ID of individual.
GenoStatus: Genotype Status.
GenoStage: At what stage an animal was genotyped (i.e. selection candidate or parent).
Pheno1Status: Phenotype status for trait 1.
Pheno2Status: Phenotype status for trait 2.
When Culled: The stage of animal which includes selection candidate (popselcandidate), parent (popparent), culled as a selection candidated (popculled_selcandidate) and culled as a parent (popculled_parent).
Generation Born: Generation when animal was born.
Age Culled: Age of an animal when it was culled.
Progeny: Number of progeny.
Sex: Sex of individual (0 = male and 1 = female).

Supplementary Files

File: CH*SNP.txt:

- Haplotype sequence for each chromosome simulated from MaCS.

File: MAP*.txt:

- map file corresponding to haplotypes sequence in CH*SNP.txt.

File: FounderGenotypes:

- Genotypes across chromosomes for each founder. The line number corresponds to the founder ID and the first column represents the row number of the two haplotypes that created the genotype, followed by the genotype string.

File: Pheno_Pedigree:

- Used in constructing pedigree relationship matrix.

File: SNPFreq:

- Frequency of SNP across all chromosomes derived from MaCS.

Appendix II - Generation of Effects

The generation of effects for the quantitative, fitness, and the covariance between additive effects when simulation multiple traits are important parameters that can have a large impact on the simulation results. The methods to generate effects for both types of traits is similar to previous articles and methods other simulation programs have used. At the current time, the sampling of additive effects is from a gamma distribution and dominances effects are simulated from a normal distribution to generate the covariance between quantitative and fitness traits.

Quantitative Trait:

The additive effect (a), defined as half the difference in genotypic value between alternative homozygotes, is generated from a gamma distribution. The default parameters for the gamma distribution(0.4,1.66) result in an L-shaped distribution of QTL effects and implies that the majority of effects are small and a few have large effects. The gamma distribution only generates positive values, therefore, with equal probability, one of the two alleles is chosen to be positive or negative based on a binomial distribution ($p = 0.5$).

The dominance effect, defined as the deviation of the value of the heterozygote from the mean of the two homozygotes, was generated using a multistep procedure. Independence between additive and dominance effects is the classical treatment (Falconer & Mackay, 1996) and it is convenient because it allows orthogonally of the additive and dominance estimates. However, this independence is contradictory with the phenomena of inbreeding depression and hybrid vigor that indicates dominance is directional (Lynch & Walsh, 1998) and results from real data (Wellmann & Bennewitz 2011; Wellmann & Bennewitz 2012), which suggest an a priori dependency between additive and dominance effects. Therefore, the degree of dominance (h) is sampled from a normal distribution, which allows for the user to vary the proportion of positive or negative dominance effects by altering the mean. Next, dominance effects (d) were generated by multiplying the degree of dominance by the absolute value of the additive effect ($d = h|a|$). The use of this simulation method results in the additive and dominance effects to now be dependent on each other. Lastly, the choice of parameters specifying the normal distribution and the minor allele frequency for the quantitative QTL has an impact on the proportion of dominance effects that display partial or over-dominance. The proportion that display partial or over-dominance is outlined near the beginning of the log file.

Two quantitative traits with a given covariance structure between the additive effects are simulated based on methods similar to Zhang et al. (2015) and Hayashi & Iwata (2013). Within each trait, similar to how additive effects are generated for one trait, additive effects are sampled from gamma

distributions and the marginal distribution across both traits are assigned the same shape and scale parameter. Due to the marginal distributions being the same across the two traits, a correlation between the additive effects for the two traits can be generated by sampling from three independent gamma distributions and the associated samples combined to generate additive effects for trait 1 and trait 2. Assuming the marginal distribution across both traits are 0.4 and 1.66 for the scale and shape parameter, respectively, the following gamma distributions were generated:

x1 $\sim$ gamma(0.4*r_g,1.66)
x2 $\sim$ gamma(0.4*(1-r_g),1.66)
x3 $\sim$ gamma(0.4*(1-r_g),1.66)

Samples from the associated gamma variables were then combined to generate Trait1 as x1 + x2 and Trait2 as x1+x3.

Fitness Trait:

The generation of fitness effects was divided into lethal and sub-lethal genetic architectures to allow for full flexibility. The distribution of fitness effects and their associated frequency in the genome have been hypothesized to come from two competing results from the literature. The first one is based on the results obtained by (Mukai et al., 1972) and is what we called the "Mukai scenario", where mutations are assumed to be numerous and of small effect. The second hypothesis is based on more recent results from mutation-accumulation studies and assume that mutations are considerable less frequent but of larger effect (Caballero & Keightley, 1994; Garcia-Dorado & Caballero, 2000). For both lethal and sub-lethal FTL the fitness was defined as relative fitness and is parameterized by two coefficients and they include the selection coefficient (s) and the dominance coefficient (h). The s value measures how much worse the unfit allele is, compared to the fittest allele. The h value measures the degree of dominance that the heterozygote shows regarding the reduced fitness compared to the unfit homozygote (Wright 1931). The normalization procedure forces the fittest homozygote genotype to have a value of 1, and the other homozygote genotype has a value of 1 - s. Lastly, heterozygote genotypes have a fitness value of 1 - hs.

The selection coefficient was generated from a gamma distribution with different parameters for lethal and sublethal. The logfile outlines the mean selection coefficient for the lethal and sub-lethal FTL. As a reference when altering the shape and scale parameter, the mean of a gamma distribution is the shape X scale.

The dominance coefficient was generated from a normal distribution with different parameters for the lethal and sublethal. The absolute value of the sample is taken as the dominance coefficient. The logfile outlines the mean

dominance coefficient for the lethal and sub-lethal FTL. As a reference when altering the shape and scale parameter, the mean of a gamma distribution is the shape X scale.

The fitness of an individual was then calculated as the multiplicative effect of each fitness genotype across both lethal and sub-lethal FTL with a maximum value of 1 and minimum of 0. A value closer to 1 has a higher fitness and is more likely to survive. In order to simulate environmental stochasticity, a random number was generated from a uniform distribution between 0 and 1 and compared with the fitness value for an individual. If the fitness value was less than the random value from the uniform distribution, the individual did not survive to breeding age and if it was greater than or equal to the animal survived to breeding age.

Covariance Between Quantitative and Fitness Traits :
The correlation between the quantitative trait and the fitness trait can be due linkage or pleiotropy. Setting the COVAR parameters both to 0 results in linkage to be the only possible source of correlation between fitness and quantitive traits. Setting the COVAR parameters to a value greater than 0 results in a pleiotropic correlation between the additive effects for the quantitative trait and the selection coefficient for the sub-lethal fitness traits. The scaling of quantitative traits results in the additive effects for the quantitative trait to change and therefore covariance was generated based on Trivariate Reduction algorithm. The Trivariate Reduction algorithm only allows the correlation to be positive. For example, high values for the quantitive trait would result in the two traits being antagonistic based on a positive correlation. One just needs to change the favorable direction of the quantitative trait to alter the interpretation.

Trivariate Reduction for Gamma1 (a_1,b_1) and Gamma2 (a_2,b_2)
- Correlation (ρ) Bounded between: $0 \leq \rho \leq \min(a_1,a_2) \; / \; \sqrt{a_1 a_2}$.
- Steps:
1.) Generate $Y_1 \sim$ gamma(a_1 - $\rho\sqrt{a_1 a_2}$,1)
2.) Generate $Y_2 \sim$ gamma(a_2 - $\rho\sqrt{a_1 a_2}$,1)
3.) Generate $Y_3 \sim$ gamma($\rho\sqrt{a_1 a_2}$,1)
4a.) Generate Value for Gamma1: $b_1(Y_1 + Y_3)$
4b.) Generate Value for Gamma2: $b_2(Y_2 + Y_3)$

The Y_3 value generate the covariance between the two traits. For FTL that have a covariance with the quantitative trait the Y_2 value is sampled for each FTL within an iteration and the rank correlation is calculated. Once the rank correlation gets within a 1.5 percent of the value specified it then generates the selection coefficient and dominance values using the current iterations Y_2 values.

Appendix III - MaCS Sequence Simulation

The MaCS program (Markovian Coalescence Simulator; Chen et al. 2009) generates the founder genome. Before using the program, it is advisable to understand the coalescent process and a good review paper is Hudson (1991) and Chapter 5 of Charlesworth & Charleworth (2010). We have chosen to employ a coalescent simulator (specifically MaCS) in this step due to the flexibility of the approach in generating haplotype sequences for a wide range of population scenarios in terms of the size and structure of the ancestral population across time and genome scenarios with varying mutation and recombination rates. We have chosen the default scenarios to resemble options specified by AlphaSim (Hickey & Gorjanc, 2012) as they represent typical agricultural species LD patterns.

There are 5 default scenarios that represent a range of linkage disequilibrium (LD) patterns that can be called within the simulation, as outlined in the figure below. The five scenarios are called by specifying either "Ne70", "Ne100_Scen1", "Ne100_Scen2", "Ne250" or "Ne1000" after the FOUNDER_Effective_Size parameter in the parameter file. The user can input a custom effective population size and historical population parameters by using "CustomNe" as the parameter. An easy way to generate custom parameters for MaCS is to utilize a default scenario that resembles the pattern you are wanting and to change the effect population size of the population and determine how the LD pattern changes. If this is specified the program looks for a file called "CustomNe" within the folder where the execution of the program occurred. The file should contain two rows, with the first one being the effective population size parameter and the last one being the historical population size parameters. Lastly, if the program only reads an integer value, then the value is the effective population size with no population history.

The generation of sequence information may take some time to compute. The files generated from the program may be large. Due to this, it is advisable only to generate sequence data once for a given scenario and then adjust narrow-sense heritability, broad sense heritability, selection or mating parameters and start with generating the founder generation.

The default scenarios are below. An illustration of the LD decay associated with each scenario is on the following page:

Ne70:

- Effective population size = "70".
- Historical population parameters: "-eN 0.18 0.71 -eN 0.36 1.43 -eN 0.54 2.14 -eN 0.71 2.86 -eN 0.89 3.57 -eN 1.07 4.29 -eN 1.25 5.00 -eN 1.43 5.71".

Ne100_Scen1:

- Effective population size = "100".
- Historical population parameters: "-eN 0.06 2.0 -eN 0.13 3.0 -eN 0.25 5.0 -eN 0.50 7.0 -eN 0.75 9.0 -eN 1.00 11.0 -eN 1.25 12.5 -eN 1.50 13.0 -eN 1.75 13.5 -eN 2.00 14.0 -eN 2.25 14.5 -eN 2.50 15.0 -eN 5.00 20.0 -eN 7.50 25.0 -eN 10.00 30.0 -eN 12.50 35.0 -eN 15.00 40.0 -eN 17.50 45.0 -eN 20.00 50.0 -eN 22.50 55.0 -eN 25.00 60.0 -eN 50.00 70.0 -eN 100.00 80.0 -eN 150.00 90.0 -eN 200.00 100.0 -eN 250.00 120.0 -eN 500.00 200.0 -eN 1000.00 400.0 -eN 1500.00 600.0 -eN 2000.00 800.0 -eN 2500.00 1000.0".

Ne100_Scen2:

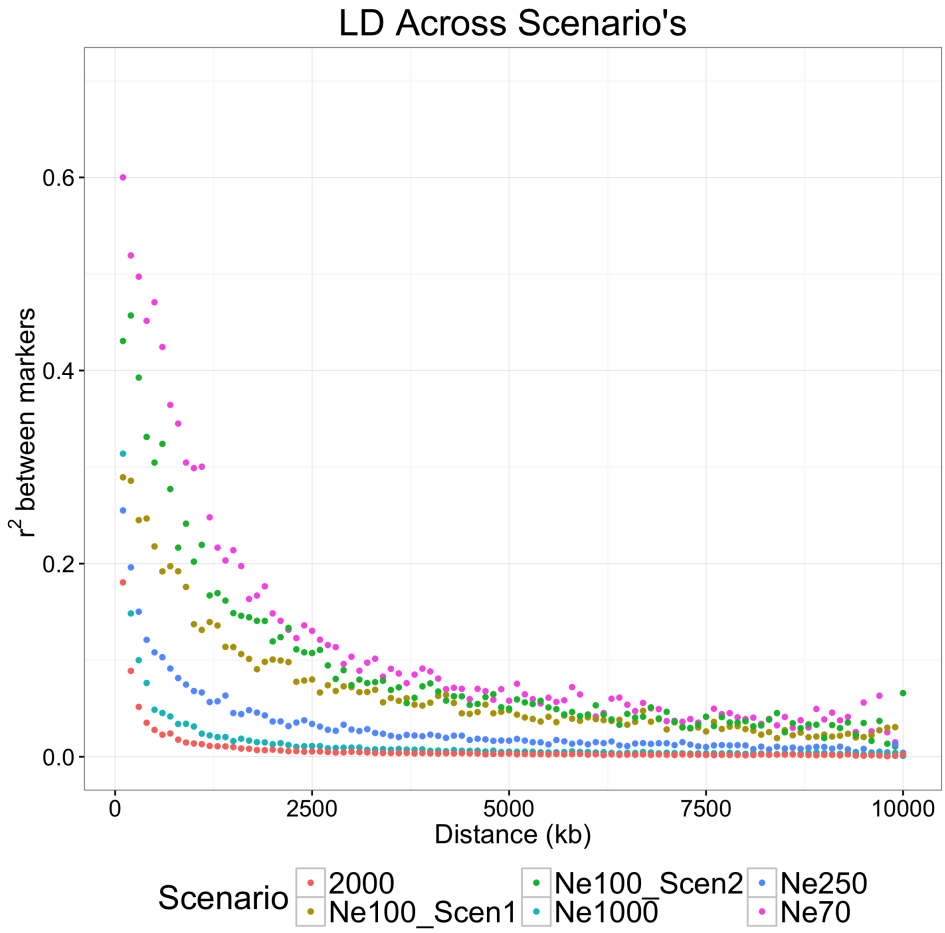
- Effective population size = "100".
- Historical population parameters: "-eN 50.00 200.0 -eN 75.00 300.0 -eN 100.00 400.0 -eN 125.00 500.0 -eN 150.00 600.0 -eN 175.00 700.0 -eN 200.00 800.0 -eN 225.00 900.0 -eN 250.00 1000.0 -eN 275.00 2000.0 -eN 300.00 3000.0 -eN 325.00 4000.0 -eN 350.00 5000.0 -eN 375.00 6000.0 -eN 400.00 7000.0 -eN 425.00 8000.0 -eN 450.00 9000.0 -eN 475.00 10000.0".

Ne250:

- Effective population size = "250".
- Historical population parameters: "-eN 0 1.04 -eN 0 1.08 -eN 0 1.12 -eN 0 1.16 -eN 0.01 1.2 -eN 0.03 1.6 -eN 0.05 2.0 -eN 0.1 2.8 -eN 0.2 4.8 -eN 0.3 5 -eN 0.4 5.2 -eN 0.5 5.4 -eN 0.6 5.6 -eN 0.7 5.7 -eN 0.8 5.8 -eN 0.9 5.9 -eN 1 6 -eN 1 4 -eN 2 8 -eN 3 10 -eN 4 12 -eN 5 14 -eN 6 16 -eN 7 18 -eN 8 20 -eN 9 22 -eN 10 24 -eN 20 28 -eN 40 32 -eN 60 36 -eN 80 40 -eN 100 48 -eN 200 80 -eN 400 160 -eN 600 240 -eN 800 320 -eN 1000 400".

Ne1000:

- Effective population size = "1000".
- Historical population parameters: "-eN 0.50 2.00 -eN 0.75 2.50 -eN 1.00 3.00 -eN 1.25 3.20 -eN 1.50 3.50 -eN 1.75 3.80 -eN 2.00 4.00 -eN 2.25 4.20 -eN 2.50 4.50 -eN 5.00 5.46 -eN 10.00 7.37 -eN 15.00 9.28 -eN 20.00 11.19 -eN 25.00 13.10 -eN 50.00 22.66 -eN 100.00 41.77 -eN 150.00 60.89 -eN 200.00 80.00".


LD Across Scenario's
r2 between markers
0.0
0.2
0.4
0.6
0
2500
5000
7500
10000
Distance (kb)
Scenario
2000
Ne100_Scen1
Ne100_Scen2
Ne1000
Ne250
Ne70

Literature Cited

- Aguilar, I., Misztal, I., Johnson, D. L., Legarra, A., Tsuruta, S. & Lawlor, T. J. 2010. Hot topic: A unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of Holstein final score. J. Dairy Sci., 93, 743-752.

- Aguilar, I., I. Misztal , A. Legarra & S. Tsuruta. 2011. Efficient computation of the genomic relationship matrix and other matrices used in single-step evaluation. J. Anim. Breed. Genet. 128:422-428.

- Ai H., L. Huang, J. Ren. 2013. Genetic diversity, linkage disequilibrium and selection signatures in chinese and Western pigs revealed by genome-wide SNP markers. PLoS One 8(2):e56001.

- Caballero A. & P. Keightley. 1994. A pleiotropic nonadditive model of variation in quantitative traits. Genetics 138:883-900.

- de los Campos, G., J. M. Hickey, R. Pong-Wong, H. D. Daetwyler, & M. P. L. Calus. 2013. Whole-Genome Regression and Prediction Methods Applied to Plant and Animal Breeding. Genetics, 193, 327-345.

- de Roos, A. P. W., B. J. Hayes, R. J. Spelman, & M. E. Goddard. 2008. Linkage Disequilibrium and Persistence of Phase in Holstein-Friesian, Jersey and Angus Cattle. Genetics, 2008, 1503-1512.

- Charlesworth, B., & D. Charlesworth. 2010. Elements of Evolutionary Genetics. Greenwoord Village, Colorado, USA: Roberts and Company.

- Chen, G. K., P. Marjoram & J. D. Wall. 2009. Fast and flexible simulation of DNA sequence data. Genome Res. 19(1):136-42.

- Cheng H., D. J. Garrick, R. L. Fernando. 2015. XSim: simulation of descendants from ancestors with sequence data. G3 5:1415-1417.

- Christensen, O. F. & Lund, M. S. 2010. Genomic predictions when some animals are not genotyped. Genet. Sel. Evol., 42, 2.

- Falconer D. S. & T. F. S. Mackay. Introduction to quantitative genetics. 4th ed. New York, NY: Longman Scientific and Technical; 1996.

- Garcia-Dorado, A. & A. Caballero. 2000. On the average degree of dominance of deleterious spontaneous mutations. Genetics, 155, 1991-2001.

- Hayashi, T. & H. Iwata. 2013. A Bayesian method and its variational approximation for prediction of genomic breeding values in multiple traits. BMC Bioinformatics 14:34.

- Hickey, J. M., & G. Gorjanc. 2012. Simulated Data for Genomic Selection and Genome-Wide Association Studies Using a Combination of Coalescent and Gene Drop Methods. G3 2:425-427.

- Howard J. T., F. Tiezzi1, Y. Huang, K.A. Gray & C. Maltecca. The use of alternative genomic metrics in nucleus herds to manage the diversity of purebred and crossbred animals. Genet. Sel. Evol. 48:91.

- Howard, J. T., T. A. Rathje, C. E. Bruns, D. F. Wilson-Wells, S. D. Kachman & M. L. Spangler. 2018. The impact of truncating data on predictive ability for single-step genomic best linear unbiased prediction. J. Anim. Breed. Genet.

- Howard, J. T., F. Tiezzi, Y. Huang, K.A. Gray, C. Maltecca. 2017. A heuristic method to identify runs of homozygosity associated with reduced performance in livestock. J. Anim. Sci. 95(10): 4318-4332.

- Hudson R. R. 1990. Gene genealogies and the coalescent process. Oxford Surveys in Evolutionary Biology. 7:1-45.

- Jonas, E. & D. -J. de Koning. 2015. Genomic Selection Needs to Be Carefully Assessed to Meet Specific Requirements in Livestock Breeding Programs. Front. Genet., 6, 49.

- Lynch, M. & B. Walsh. 1998. Genetics and analysis of quantitative traits. Vol. 1. Sunderland, MA: Sinauer.

- McKay, S. D., R. D. Schnabel, B. M. Murdoch, L. K. Matukumalli, J. Aerts, W. Coppieters, D. Crews, E. Dias Neto, C. A. Gill, C. Gao, H. Mannen, P. Stothard, Z. Wang, C. P. Van Tassell, J. L. Williams, J. F. Taylor, S. S. Moore. 2007. Whole genome linkage disequilibrium maps in cattle. BMC Genet. 8:74.

- McMahon, B. J., E. C. Teeling, J. Hoglund. 2014. How and Why Should We Implement Genomics into Conservation? Evol. Appl., 7, 999-1007.

- Meuwissen, T. H. E., & Z. Luo. 1992. Computing inbreeding coefficients in large populations. Genet Sel Evol. 24(4): 305-313.

- Meyer, K., B. Tier & H. U. Graser. 2013. Technical note: Updating the inverse of the genomic relationship matrix. J. Anim. Sci. 91:2583-2586.

- Misztal, I., A. Legarra & I. Aguilar. 2014. Using recursion to compute the inverse of the genomic relationship matrix.
- Morrell, P. L., E. S. Buckler, J. Ross-Ibarra. 2012. Crop genomics: advances and applications. Nat. Rev. Genet. 13, 85-96.
- Mukai, T., S. I. Chigusa, L. E. Mettler, J. F. Crow. 1972. Mutation rate and dominance of genes affecting viability in drosophila melanogaster. Genetics 72:333-355.
- Pérez-Enciso M., & A. Legarra. 2016. A combined coalescence gene-dropping tool for evaluating genomic selection in complex scenarios (ms2gs). J. Anim. Breed. Genet. 133(2):85-91.
- Porto-Neto L. R., T. S. Sonstegard, G. E. Liu, D. M. Bickhart, M. V. Da Silva, M. A. Machado, Y. T. Utsunomiya, J. F. Garcia, C. Gondro, C. P. Van Tassell. 2013. Genomic divergence of zebu and taurine cattle identified through high-density SNP genotyping. BMC Genomics. 14:876.
- Sargolzaei, M. & F. S. Schenkel. 2009. QMSim: a large-scale genome simulator for livestock. Bioinformatics, 25, 680-1.
- VanRaden P. M. 2008. Efficient Methods to Compute Genomic Predictions. J. Dairy Sci., 91, 4414-4423.
- VanRaden P. M., K. M. Olson, D. J. Null, J. L. Hutchison. 2011. Harmful recessive effects on fertility detected by absence of homozygous haplotypes. J. Dairy Sci., 94, 6153-6161.
- Wellmann, R. & J. Bennewitz. 2011. The contribution of dominance to the understanding of quantitative genetic variation. Genet Res (Camb) 93:139-154.
- Wellmann, R. & J. Bennewitz. 2012. Bayesian models with dominance effects for genomic evaluation of quantitative traits. Genet Res (Camb) 94:21?37.
- Wright, S. 1931. Evolution in Mendelian populations. Genetics 16:97-159.
- Yang, J., B. Benyamin, B. P. McEvoy, S. Gordon, A. K. Henders, D. R. Nyholt, P. A. Madden, A. C. Heath, N. G. Martin, G. W. Montgomery, M. E. Goddard, P. M. Visscher. 2010. Common SNPs explain a large proportion of the heritability for human height. Nat Genet., 42, 565-569.
- Zhang, Z., X. Li, X. Ding, J. Li, Q. Zhang. 2015. GPOPSIM: a simulation tool for whole-genome genetic data. BMC Genetics 16:10

Example 1: Running the Simulation Software

Quantitative Trait - Single Progeny

```
−−−−−−−|     Running the Program Example      |−−−−−−−
−| General |−
START: sequence
SEED: 1500
−| Genome & Marker |−
CHR: 3
CHR_LENGTH: 150 150 150
NUM_MARK: 4000 4000 4000
QTL: 150 150 150
−| Population |−
FOUNDER_Effective_Size: Ne70
MALE_FEMALE_FOUNDER: 50 400 random 3
VARIANCE_A: 0.10
−| Selection |−
GENERATIONS: 15
INDIVIDUALS: 50 0.2 400 0.2
PROGENY: 1
SELECTION: ebv high
EBV_METHOD: pblup
CULLING: ebv 5
−| Mating |−
MATING: random125 simu_anneal
```

To run the program place, "GenoDiver", "macs" and "msformatter" executable files in the folder where the program will run. Before running the program, the file permissions need to be checked. After verifying the permissions, a parameter file, outlined above, will need to be generated and placed in the same folder as the previous three executable files. A parameter file can be generated using any text editor. The simulation program reads the parameter file by searching for keywords that are capitalized and followed by a colon. Therefore any phrase that does not meet the search criteria is ignored when initializing parameters within the program.

To run the program type "./GenoDiver" and the name of your parameter file. For example, if the parameter file is named "parameterfile", the program is run by typing in "./GenoDiver parameterfile". During the simulation, the program outputs minimal comments on the progress. A more thorough description of the program's status is printed to the log file (i.e. "log_file.txt").

Parameter File Summary

Sequence information is generated for three chromosomes with a length of 150 Megabases (Mb). The simulated genome has a high degree of short-range LD (Ne70). The SNP panel contains 12,000 markers (i.e. 4,000 markers per chromosome). For each chromosome, 150 randomly placed QTL and zero FTL mutations were generated. The quantitative trait simulated has a narrow sense heritability of 0.10 and only additive effects are generated (i.e. no dominance). The phenotypic variance is by default set at 1.0, and therefore the residual variance is 0.90. The founder population consisted of 50 males and 400 females. For each generation, a total of 50 males and 400 females are in the population. Random selecton of progeny and culling of parents was conducted for 3 generations in order to build up the pedigree. A total of 10 and 80 (0.2 replacement rate) male and female parents, respectively, are culled and replaced by new progeny each generation. After 3 generations, animals with a high EBV are selected or culled each generation. Fifteen generations are simulated. The EBV are estimated using an animal model with a pedigree-based relationship matrix. Each mating pair produced one progeny. Parents with a pedigree-based relationships greater than 0.125 were avoided, and this was optimized based on the simulated annealing method.

Overview of Results

After the program has finished it is a strongly recommended to check the parameters initialized at the top of the log file (i.e. "log_file.txt") within the output folder. The log file contains a large amount of information and is a great tool to ensure that the parameters and outcome of the simulation is what is intended. If the program is not running correctly the log file should provide knowledge on why and where the simulation crashed or exited. The files are by default placed in the "GenoDiverFiles" directory. If the "OUTPUTFOLDER" option is utilized, the files will be placed in the user-specified directory. Below is a screenshot of the files that are generated from the simulation software based on the parameter file outlined above.

A number of files are generated, but only a few are needed to generate summary statistics on the simulation program and include:

- **Master_DataFrame**: File with phenotype, inbreeding and pedigree information across all animals.
- **Master_Genotypes**: File with genotype information for each animal.
- **QTL_new_old_Class**: File with information for each QTL/FTL mutation.

- **Marker_Map**: Location of markers.
- **Summary_Statistics_DataFrame_Performance**: Summary statistics by generation on performance metrics.
- **Summary_Statistics_DataFrame_Inbreeding**: Summary statistics by generation on inbreeding metrics.
- **Summary_Statistics_QTL**: Summary statistics by generation QTL/FTL metrics.

Overview of Results:

- The table below outlines the change in multiple parameters as the generations proceed

Generation	Phenotype	EBV	Pedigree Inbreeding	Genomic Inbreeding
0	0.000	0.000	0	0.978
1	0.004	0.004	0	0.997
2	0.308	0.254	0	0.999
3	0.459	0.476	0	0.999
4	0.765	0.752	0.0008	1.007
5	1.030	1.014	0.0022	1.018
6	1.207	1.214	0.0078	1.021
7	1.507	1.463	0.0143	1.047
8	1.677	1.670	0.0200	1.058
9	1.887	1.871	0.0251	1.066
10	2.099	2.063	0.0296	1.080
11	2.246	2.256	0.0350	1.099
12	2.519	2.465	0.0389	1.111
13	2.617	2.619	0.0435	1.123
14	2.765	2.791	0.0485	1.148
15	2.965	2.971	0.0510	1.153

Example 2: Quantitative Trait - Multiple Progeny

```
−−−−−−−|     Multiple Progeny Quantitative Trait      |−−−−−−−
−| General |−
START: sequence
SEED: 1501
−| Genome & Marker |−
CHR: 3
CHR_LENGTH: 150 150 150
NUM_MARK: 4000 4000 4000
QTL: 150 150 150
−| Population |−
FOUNDER_Effective_Size: Ne70
MALE_FEMALE_FOUNDER: 25 200 random 3
VARIANCE_A: 0.35
−| Selection |−
GENERATIONS: 10
INDIVIDUALS: 25 0.2 200 0.2
PROGENY: 8
MAXFULLSIB: 2
SELECTION: ebv high
EBV_METHOD: pblup
CULLING: ebv 5
−| Mating |−
MATING: random125 simu_anneal
```

The parameter file outlined below illustrates how to simulate a prolific species (i.e. Swine) with a threshold on the minimum number of progeny to select within each family. Once the program has finished, inspection of the log file will provide details on the impact of the "MAXFULLSIB" option. Within the log file, after the "Begin ebv Selection of offspring" section, the number of times "n" number of siblings were selected within a family is outlined for each generation. This provides an overview of the degree of co-selection of siblings that occur based on the simulation design outlined above. If this option would not have been included, the maximum number of siblings that can be selected within a family would be 8. Given the low narrow-sense heritability of the trait, the number of siblings selected within a family would be higher if this parameter wasn't included.

Parameter File Summary

Sequence information is generated for three chromosomes with a length of 150 Megabases (Mb). The genome simulated has a high degree of short-

range LD (Ne70). The SNP panel contains 12,000 markers (i.e. 4,000 markers per chromosome). For each chromosome, 150 randomly placed QTL and zero FTL mutations were generated. The quantitative trait simulated has a narrow sense heritability of 0.35 and only additive effects are generated (i.e. no dominance). The phenotypic variance is by default set at 1.0, and therefore the residual variance is 0.65. The founder population consisted of 25 males and 200 females. Random selecton of progeny and culling of parents was conducted for 3 generations. For each generation, a total of 25 males and 200 females are in the population. A total of 5 and 40 (0.2 replacement rate) male and female parents, respectively, are culled and replaced by new progeny each generation. After 3 generations, animals with a high EBV were selected or culled each generation. A total of 10 generations were simulated. The EBV are estimated using an animal model with a pedigree-based relationship matrix. Each mating pair produced eight progeny. Within each full-sib family a maximum of 2 progeny can be selected. Parents that had pedigree-based relationships greater than 0.125 were avoided, and this was optimized based on the simulated annealing method.

Overview of Results

Once the program has finished, inspection of the log file will provide details on the impact of the "MAXFULLSIB" option. Within the log file, after the "Begin ebv Selection of offspring" section, the number of times "n" number of siblings were selected within a family is outlined for each generation. This provides an overview of the degree of co-selection of siblings that occur based on the simulation design outlined above. If this option would not have been included, the maximum number of siblings that can be selected within a family would be 8. Given the low narrow-sense heritability of the trait, the number of siblings selected within a family would be higher if this parameter wasn't included.

Example 3: Differential Sire Contribution by Age

```
−−−−−−−−|     Differential Sire Contribution by Age      |−−−−−−−−
−| General |−
START: sequence
SEED: 1501
−| Genome & Marker |−
CHR: 3
CHR_LENGTH: 150 150 150
NUM_MARK: 4000 4000 4000
QTL: 150 150 150
−| Population |−
FOUNDER_Effective_Size: Ne70
MALE_FEMALE_FOUNDER: 50 400 random 3
VARIANCE_A: 0.35
−| Selection |−
GENERATIONS: 15
INDIVIDUALS: 50 0.2 400 0.2
PROGENY: 1
PARITY_MATES_DIST: 2.0 1.0
SELECTION: ebv high
EBV_METHOD: pblup
CULLING: ebv 5
−| Mating |−
MATING: random125 simu_anneal
```

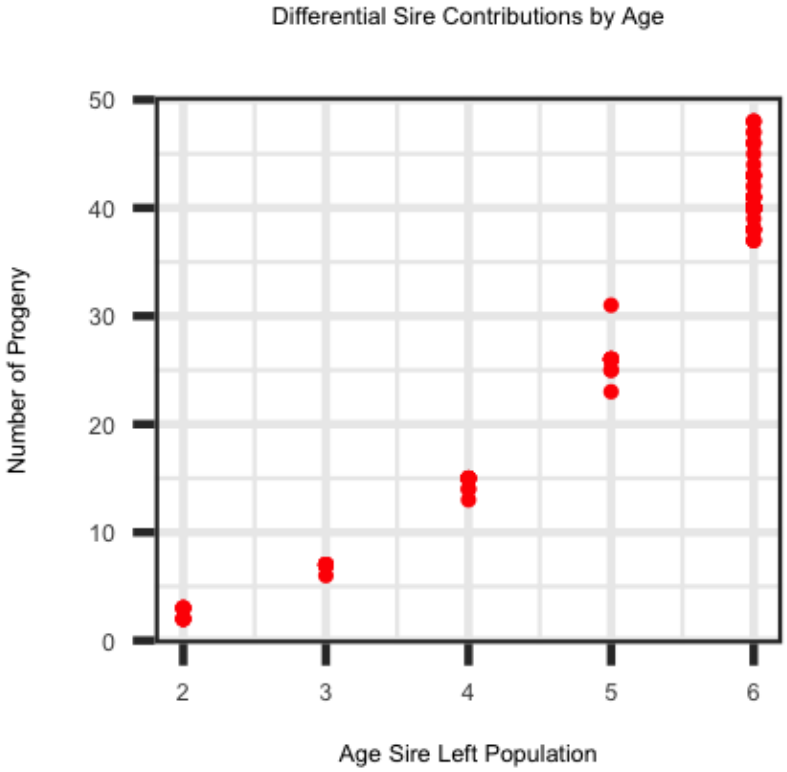
The parameter file outlined above illustrates how to simulate a population where older animals are assigned a larger number of mating pairs compared to younger animals. This type of scenario is generated by utilizing the "PARITY_MATES_DIST" parameter with the following values "2.0 1.0". These two values are utilized to generate the distribution of mating pairs. A Beta distribution, which is parameterized by two parameters, is used to generate the distribution of mating pairs. A beta distribution was utilized in order to allow for a wide range of mating scenarios. For example, to generate a scenario where younger animals are assigned a larger number of mating pairs compared to older animals, the parameters need to be changed to "1.0 2.0". The number of mating pairs by age class are generated by splitting the cumulative distribution function (CDF) into quadrants based on the number of age classes that occur within a generation. The total number of mating pairs within an age class is the proportion that falls within the CDF quadrant for a given age class.

Parameter File Summary

Sequence information is generated for three chromosomes with a length of 150 Megabases. The genome simulated has a high degree of short-range LD (Ne70). The SNP panel contains 12,000 markers (i.e. 4,000 markers per chromosome). For each chromosome, 150 randomly placed QTL and zero FTL mutations were generated. The quantitative trait simulated has a narrow sense heritability of 0.35 and only additive effects are generated (i.e. no dominance). The phenotypic variance is by default set at 1.0, and therefore the residual variance is 0.65. The founder population consisted of 50 males and 400 females. Random selecton of progeny and culling of parents was conducted for 3 generations. For each generation, a total of 50 males and 400 females are in the population. A total of 10 and 80 (0.2 replacement rate) male and female parents, respectively, are culled and replaced by new progeny each generation. After 3 generations, animals with a high EBV were selected or culled each generation. A total of 15 generations were simulated. The EBV are estimated using an animal model with a pedigree-based relationship matrix. Each mating pair produced one progeny. The mating distribution was skewed so that older animals had more mating pairs than younger animals. Parents that had pedigree-based relationships greater than 0.125 were avoided, and this was optimized based on the simulated annealing method.

Overview of Results

Once the program has finished, inspection of the log file will provide details on the impact of the "PARITY_MATES_DIST" option. Within the log file (lines 148-169), the mating distribution CDF is illustrated and is outlined below. When generating the number of matings for a given age class for each generation the CDF outlined below is split into quadrants based on the number of age classes that occur within a generation. A potential reasons for putting this into a simulation is to generate some sires with a large number of progeny while other sires have very few progeny. The sires that generate a large number of progeny would then have a large impact on the genome of future generations which may include the spread of a lethal/sublethal mutation that the sire(s) carry. The plot below depicts the non-linear relationship of number of progeny left by a sire across different ages at which a sire left the herd.


Differential Sire Contributions by Age
Number of Progeny
50
40
30
20
10
0
2
3
4
5
6
Age Sire Left Population

Example 4 Fitness Trait

```
−−−−−−−−|     Simulating a Fitness Trait      |−−−−−−−−
−| General |−
START: sequence
SEED: 1500
−| Genome & Marker |−
CHR: 3
CHR_LENGTH: 150 150 150
NUM_MARK: 4000 4000 4000
QTL: 0 0 0
FIT_LETHAL: 50 50 50
FIT_SUBLETHAL: 50 50 50
−| Population |−
FOUNDER_Effective_Size: 500
MALE_FEMALE_FOUNDER: 150 600 random 0
VARIANCE_A: 0.0
−| Selection |−
GENERATIONS: 25
INDIVIDUALS: 10 0.2 400 0.2
PROGENY: 1
SELECTION: random high
CULLING: random 10
−| Mating |−
MATING: random
−| Output Options |−
GENOTYPES: no
```

The parameter file below illustrates how to simulate a fitness trait only. When simulating a fitness trait it is important to ensure that you have enough founder individuals because a portion will not make it to breeding age. Therefore, an extra 140 males and 200 females were added to the founder population to ensure enough individuals are available to generate the breeding population. This will also impact the replacement rate because if enough progeny aren't available to remain at the chosen male and female population size the simulation will exit. A full description of how the fitness value of an individual impacts its ability to make it to breeding age is described in Appendix 2.

Parameter File Summary

Sequence information is generated for three chromosomes with a length of 150 Megabases. The genome simulated has a low degree of short-range LD (250). The SNP panel contains 12,000 marker (i.e. 4,000 markers per chromosome). For each chromosome, 50 lethal and 50 sub-lethal mutations were generated. The quantitative trait has a broad sense heritability of 0.0 and therefore an animals phenotype is only a function of random environmental deviations with a variance of 1.0. The founder population consisted of 150 males and 600 females. For each generation, a total of 10 males and 400 females are in the population. A total of 10 and 80 (0.2 replacement rate) male and female parents, respectively, are culled and replaced by new progeny each generation. Across all generations animals are randomly selected or culled each generation. The maximum number of generations an animal can remain in the breeding population is 10. Each mating pair produced one progeny and parents were mated at random. The genotypes are not saved to a file.

Overview of Results:

- The number of progeny that died due to fitness and the number of the FTL purged from the population is in Summary_Statistics_QTL file.
- The log file contains the mean selection coefficient and degree of dominance for the lethal and sublethal FTL.
- When simulating fitness effects, a certain proportion of the progeny will die due to fitness and if the population does not have enough progeny to stay at the value given the program exits. Therefore, careful consideration of the number and magnitude of fitness effects along with the number of progeny produced per mating pair needs to be carefully considered when constructing the parameter file.

Example 5 (Quantitative Trait + Fitness Correlated)

```
−−−−−−−−|     Quantitative and Fitness Trait      |−−−−−−−−
−| General |−
START: sequence
SEED: 1501
−| Genome & Marker |−
CHR: 3
CHR_LENGTH: 150 150 150
NUM_MARK: 4000 4000 4000
QTL: 150 150 150
FIT_LETHAL: 15 15 15
FIT_SUBLETHAL: 100 100 100
−| Population |−
FOUNDER_Effective_Size: Ne70
MALE_FEMALE_FOUNDER: 100 400 random 3
VARIANCE_A: 0.20
VARIANCE_D: 0.05
COVAR: 0.5 0.2
−| Selection |−
GENERATIONS: 20
INDIVIDUALS: 50 0.2 250 0.2
PROGENY: 1
SELECTION: ebv high
EBV_METHOD: pblup
CULLING: ebv 10
−| Mating |−
MATING: random
```

The parameter file outlined below illustrates how to simulate a quantitative trait that has a proportion (i.e. 50%) of the quantitative trait loci (QTL) also having a fitness effect. The relationship between the QTL with a fitness effect has a positive correlation of 0.20, but the two traits are antagonistic based on the way the fitness value of a loci is parameterized. High selection coefficients (s) result in the unfavorable homozygote genotype to be less fit and is described in detail in the 'QTL/FTL Distributions' link. As outlined in the previous example, it is important to add a few extra animals in the founder population since a portion will die. If the number of male or female founder animals is smaller than the population size it will exit out of the program.

Parameter File Summary

Sequence information is generated for three chromosomes with a length of 150 Megabases (Mb). The simulated genome has a high degree of short-range LD (Ne70). The SNP panel contains 12,000 markers (i.e. 4,000 markers per chromosome). For each chromosome, 150 randomly placed QTL were generated. Also, 15 lethal and 100 sub-lethal mutations were generated for each chromosome. The narrow and broad sense heritability for the quantitative trait is 0.20 and 0.25, respectively. The phenotypic variance is by default set at 1.0, and therefore the residual variance is 0.75. Half of the QTL also have a fitness effect and the correlation between the additive QTL effects and sub-lethal selection coefficients is 0.20. The founder population consisted of 100 males and 400 females prior to determining whether an animals survived to breeding age. For each generation, a total of 50 males and 250 females are in the population. Random selecton of progeny and culling of parents was conducted for 3 generations in order to build up the pedigree. A total of 10 and 50 (0.2 replacement rate) male and female parents, respectively, are culled and replaced by new progeny each generation. After 3 generations, animals with a high EBV were selected or culled each generation. Twenty generations are simulated.The EBV are estimated using an animal model with a pedigree-based relationship matrix. Each mating pair produced one progeny. Parents were mated at random.