GENO-DIVER

Genetic Simulation Toolkit

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

[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.

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. 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 multiple 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 se-

lection 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; Prez-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 simulation program Geno-Diver and various parameters and their associated values that are utilized within the program. The software is free for academic and non-commercial 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 constantly. If you would like the program to do something that it currently does not do, don't hesitate to contact me at jthoward@ncsu.edu and I will make an attempt to include it in the simulation.

Please notify jthoward@ncsu.edu or cmaltec@ncsu.edu if you think the results are not correct or you have encountered a bug. We have wrote the program in a way for us to reconstruct the program you ran in order to more effectively solve the problem.

The bibliographical reference for this software is:

Download

Source code, executables and helper files are available at: GENO-DIVER

Linux Executable Files:

- macs
- msformatter
- GenoDiver

Source Code Files:

- Animal.h
- AnimalFun.cpp
- PopulationSimulator.cpp
- Simulation_Functions.cpp
- makefile

Helper Files:

- Geno-Diver Manual
- ADSA-ASAS National Meeting Presentation
- Replicate_Script.sh
- Batch_Script.sh
- complete_parameter_file.txt
- minimal_parameter_file.txt
- Example1.txt
- Example2.txt
- Example3.txt
- Example4.sh

Computing Environment

The code is written in C++11 language using object oriented techniques and the application runs on Linux 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". In order 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"). This will serve as your path in the make file 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 is downloaded or instruction can be obtained at Intel MKL Guide and depending on the computing system you are running, a guide to linking the Intel MKL libraries can be found at Intel MKL Linking Guide

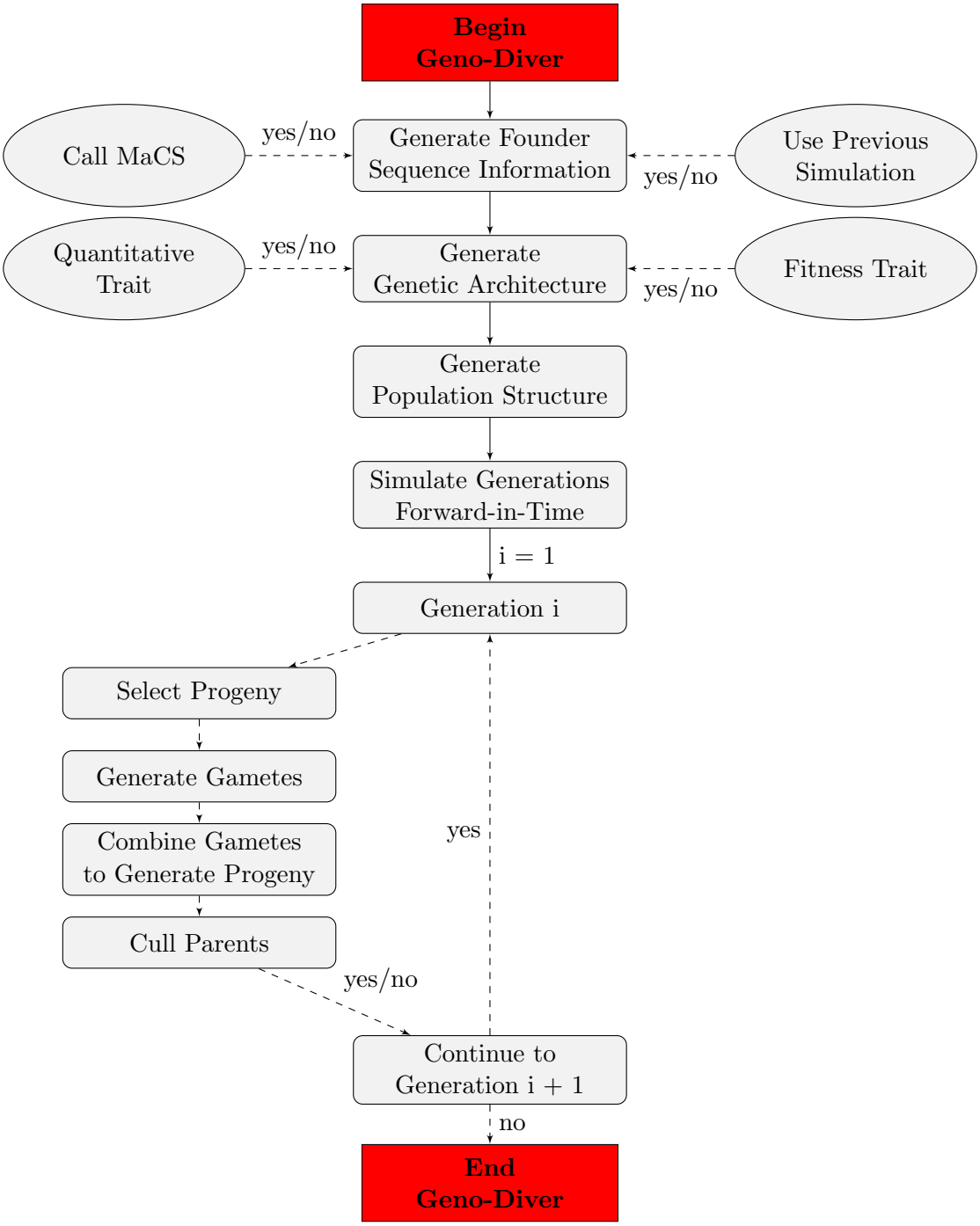
C++11 Version:
The C++11 standards start being supported in gcc 4.7 or newer. You would install or update to the correct version of gcc using the normal package manager or installer, depending on what type of OS you are using. Some website that can be used a reference include:
gcc helper 1 gcc helper 2

Compiling:
Once both EIGEN, Intel MKL libraries and gcc version 4.4 or newer have been 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.

In the makefile change lines 13 and 14 to the path which aligns to the directories where the EIGEN and MKL files are located.

After changing the path, type "make" command line and an executable file called "GenoDiver" will be created in your working directory.

Overview of Program



Running the Program

At the current time executable files are available only for linux operating environments and are located on the github page(GENO-DIVER) . To run the program place the following executable files in a folder:

- GenoDiver.
- macs.
- msformatter.

Prior to running the program you will need to check the permissions of the files and if needed make them executable (i.e. type "chmod 755 GenoDiver"). Once the permissions have been changed you need to generate a parameter file and place it in the same folder as the previous three files. If you are new to the program a minimal example is outlined in Example 1. The parameter file is read by searching for key words that are capitalized and then followed by a colon. Therefore any other phrase will be ignored when initializing parameters within the program.

To run the program type "./GenoDiver" and then the name of your parameter file. Minimal comments on the progress of the program will be printed to the screen. A more thorough description of the progress 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 top of the log file within the output folder to ensure the correct parameters were read in. The log file contains a great deal of information and is a great tool to ensure what you simulated is correct.

If the program is not running correctly the log file should provide knowledge on why and where the simulation crashed or exited.

If you are unsure of the problem an e-mail can be sent to jthoward@ncsu.edu with a copy of the log-file in order for me to replicate the results.

More advanced linux options in the form of bash scripts are described on the git-hub page such as running it in the background and running multiple replicates.

Program Parameters

A file that specifies all the parameters is outlined on the git-hub page and is called "complete_parameter_file.txt". Not all of them are required for the program to run. A file with the mandatory parameters for the simulation to run is outlined on the git-hub page and is called "minimal_parameter_file.txt". All key words should be in capital letters and the parameter(s) specified are separated by spaces. Only parameters after the key words impact the simulation.

A description of all the parameters that can be specified in the program is outlined below. The appendix also contains further hints and suggestions.

General and Starting Parameters

START

Description: - Determines where to start the simulation program.

Value:

- 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 will save time and space if done correctly. If you are using a different effective population size or MaCS diversity metric you always have to start at the sequence step, while any replicate that has the same MaCS parameters can start at the founder step and use the previously generated haplotypes saved within the output folder.

OUTPUTFOLDER

Description: - Directory name of files generated from simulation.
Value: - Any valid folder name.
Usage: - "OUTPUTFOLDER: GenoDiverFiles".
Type: - Optional. Default is "GenoDiverFiles".
Note:

- If there is already information in the file it will delete files depending on the value of the START parameter. If the START parameter is "sequence" everything is deleted. If the START parameter is "founder" everything but sequence information is deleted. The sequence files could potentially take up a large amount of memory. Only one sequence file can be generated at a time unless the mac, msformatter and GenoDriver are placed in separate folders.

SEED

Description: - Declares the seed number.
Value: - Can be any integer value.
Usage: - "SEED: 1501".
Type: - Optional. Default is the system time.
Note:

- If not declared by user it will appear in the beginning of the log file in order to generate the same data set again.

THREAD

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

Genome and Marker Information

CHR

Description: - Sets the number of chromosomes to generate.
Value: - Can be any integer value.
Usage: - "CHR: 3".
Type: - Mandatory.

CHR_LENGTH

Description: - Sets the length of each chromosome, in Megabases.
Value: - Integer Value.
Usage: - "CHR_LENGTH: 150, 150, 150"
Type: - Mandatory.
Note: - Can't be any spaces after final number.

NUM_MARK

Description:
- Sets the number of evenly spaced markers for each chromosome.

Value: - Integer Value.
Usage: - "NUM_MARK: 4000, 4000, 4000"
Type: - Mandatory.
Note:
- Quantitative and fitness QTL's cannot be markers at the current time. Can't be any spaces after the final number. The number of markers has to be less than the available number of mutations within a chromosome generated from MaCS.

MARKER_MAF

Description: - Minimum allele frequency allowed for markers.
Value: - Range from 0.0 to 0.50.
Usage: - "MARKER_MAF: 0.075"
Type: - Optional. Default is 0.05.

QTL

Description:
- The number of quantitative QTL for each chromosome. The quantitative QTL location is generated based on a uniform distribution from 0 to the length of the chromosome.

Value: - Integer. 5000 is max number of quantitative and fitness QTL.
Usage: - "QTL: 50"
Type: - Mandatory.

QUANTITATIVE_MAF

Description: - Minimum allele frequency allowed for quantitative QTL.
Value: - 0 to 0.5.
Usage: - "QUANTITATIVE_MAF: 0.05"
Type: - Optional. Default is 0.05..

FIT_LETHAL

Description:
- The number of lethal fitness QTL for each chromosome. The lethal fitness QTL location is generated based on a uniform distribution from 0 to the length of the chromosome.

Value: - Integer. 5000 is max number of quantitative and fitness QTL.
Usage: - "FIT_LETHAL: 25".
Type: - Optional. Default is 0.
Note: - These QTL can't have covariance with the quantitative trait.

FIT_SUBLETHAL

Description:
- The number of sub-lethal fitness QTL for each chromosome. The sub-lethal fitness QTL location is generated based on a uniform distribution from 0 to the length of the chromosome.

Value: - Integer. 5000 is max number of quantitative and fitness QTL.
Usage: - "FIT_SUBLETHAL: 25".
Type: - Optional. Default is 0.
Note: - These QTL can have covariance with the quantitative trait.

FITNESS_MAF

Description:
- Minimum allele frequency allowed for fitness QTL for lethal and sublethal. The first and second value pertains to lethals and sub-lethals, respectively.

Value: - Range from 0.0 to 0.5.
Usage: - "FITNESS_MAF: 0.02 0.08".
Type: - Optional. Default is 0.02 and 0.08.
Note:
- The minimum value is set to 0.01. Using the default values SNP will be found that range from 0.01 to 0.02 for lethals and 0.01 to 0.08 for sublethals. If the frequency is set to a high value a large number of founders and progeny will be killed and the simulation will exit due to a lack of individuals.

FOUNDER_HAPLOTYPES

Description:
- The number of haplotypes generated by MaCS. Need to ensure that it is greater than two times the total number of animals needed in the founder population.

Value: - 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. This is used to to generate haplotype based relationship matrices and summary statistics.

Value: - Integer Value.

Usage: - "HAPLOTYPE_SIZE: 50".

Type: - Optional. Default is 50.

RECOMBINATION

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

Value:
- Uniform: Recombination sampled from a Uniform distribution from 0 to 1.0.
- Beta: Recombination sampled from a Beta distribution (0.5, 0.5) from 0 to 1.0. Recombination's occurs more often at the end of the chromosome than towards the middle.

Usage: - "RECOMBINATION: Uniform".

Type: - Optional. Default is Uniform.

QTL Distributions

ADD_QUAN

Description:

- The parameters for the gamma distribution that generate the un-scaled additive effect for quantitative QTL. The effects are scaled such that the sum of the QTL variances in the founder population is equivalent to the proportion to the variance that is due to the additive gene action specified. A complete description is in Appendix II.

Value:

- 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 quantitative QTL. The effects are scaled such that the sum of the QTL variances in the founder population is equivalent to the proportion of the variance that is due to dominant gene action specified. A complete description is in Appendix II.

Value:

- 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 fitness QTL. No scaling is done. A complete description is in Appendix II.

Value:

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

Usage: - "LTHA: 1.6 0.1".
Type: - Optional. Default is 1.6 0.1.

LTHD

Description:

- The parameters for the normal distribution that generate the degree of dominance for the lethal fitness QTL. No scaling is done. A complete description is in Appendix II.

Value:

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

Usage: - "LTHD: 0.05 0.1".
Type: - Optional. Default is 0.05 0.1.

SUBA

Description:

- The parameters for the gamma distribution that generate the selection coefficients for sublethal fitness QTL. No scaling is done. A complete description is in Appendix II

Value:

- 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 fitness QTL. No scaling is done. A complete description is in Appendix II

Value:

- 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 QTL's associated with quantitative and sub-lethal traits. The relationship is a function of the number of QTL that are both quantitative and sub-lethal and the rank correlation between the effects. A complete description is in Appendix II

Value:

- Proportion of Pleiotropic QTL: Proportion of QTL that are both quantitative and sub-lethal
- Genetic correlation: The rank correlation between QTL effects.

Usage: - "COVAR: 0.5 0.2".

Type: - Optional. Default is 0.0 0.0..

Note:

- Care needs to be taken in order to generate the type of relationship that is desired between the quantitative and fitness trait. 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, a positive correlation should be given. 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 of the haplotypes generated from MaCS. There are multiple scenario's that can be called and represent a wide range of LD patterns or one can specify their own effective population size and population history. Each scenario has a slightly different population history parameter (i.e. "eN"). Lastly, a single value can be utilized and the historical population parameter will not be utilized. A description is outlined in Appendix III.

Value:

- 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 the minimal LD.
- Ne1000: A scenario that generates the minimal LD.
- CustomNe - Read in own population history parameters.
- Any value greater than 1: Utilizes the value as the effective population size and no population history assumed.

Usage: - "FOUNDER_Effective_Size: 50".

Type: - Mandatory.

MUTATION

Description:

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

Value:

- Mutation Rate: Probability of a new mutation occurring at a given base pair and follows the infinite alleles model. The total number of mutations occurring within a new gamete is generated from a Poisson distribution with a rate parameter equal to the mutation rate times the length of the chromosome in nucleotides.
- Proportion of Mutations that can be QTL: This parameter is utilized after the sequences are generated using MaCS and represents the number of non-neutral mutations that occurred out of the total number of new mutations. Each type of QTL (i.e. quantitative, lethal, sub-lethal) has an equal change of being chosen.

Usage: - "MUTATION: 2.5e-8 0.0".
Type: - Optional. Default is 2.5e-8 0.0.

VARIANCE_A
Description: - Proportion of variance due to additive gene action.
Value: - 0.0 to 1.0.
Usage: - "VARIANCE_A: 0.25".
Type: - Mandatory.

VARIANCE_D
Description: - Proportion of variance due to dominant gene action.
Value: - 0.0 to 1.0.
Usage: - "VARIANCE_D: 0.05".
Type: - Mandatory.
Note:

- Care needs to be taken in determining the additive and dominance variance and it implications on the number of quantitative QTL loci that display over-dominance or partial-dominance. Parameters to change that will impact the dominance variance include the QTL MAF frequency, mean and standard deviation of the normal distribution that generates the degree of dominance parameter and the ratio of additive to dominance variance.

Selection and Mating Parameters

GENERATIONS
Description: - Determines the number of generations you want to simulate.
Value: - Any integer value.
Usage: - "GENERATIONS: 10".
Type: - Mandatory.
Note:

- The maximum number generation will vary depending on the size of your computer memory.

INDIVIDUALS

Description:

- Determines the number of males and females in each generation and replacement rate for parents each generation. Care should be taken on picking the number of offspring in order to have enough for the next generation. If the number of animals falls below the input male or female value, the program will exit.

Value:

- Male Number: Number of males within each generation.
- Male Replacement: Proportion of males that are culled and replaced each generation.
- Female Number: Number of females within each generation.
- Female Replacement: Proportion of females that are culled and replaced 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. The distribution is generated from a Beta that is parameterized by two parameters. The number of mating pairs for a given age class is determined by splitting the cumulative distribution function (CDF) into quadrants based on the number of age classes that occur within a generation. The proportion of mating pairs out of the total that are appropriated to a given age class is then based on the proportion that fall within the CDF quadrant.

Value: - Both parameters have to be positive values.

Usage: - "PARITY_MATES_DIST: 1 1".

Type:

- Optional. 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. This may not be the actual number of progeny produced if lethal and/or sub-lethal QTL exist in the population. A progeny of a mating pair will be produced if the fitness value of the progeny is greater than a random value derived from a uniform distribution ranging from 0 to 1. The fitness value of the progeny is derived from the multiplicative effect of all lethal and sub-lethal QTL effects.

Value: - Ranges from 1 to 10.

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.

Value: - 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 individuals and favorable direction.

Value:

- Select: random, phenotype, true_bv or ebv.
- Direction: high or low.

Usage: - "SELECTION: ebv high".

Type: - Mandatory.

Note:

- For random selection the direction does not impact the results and therefore the direction value doesn't matter.

SOLVER_INVERSE

Description:

- Parameters specifying which relationship matrix is used to estimate breeding values, how they will be solved and how the inverse will be calculated.

Value:

- Relationship Matrix:
  - pedigree: constructed using genealogical information.
  - genomic: constructed from genomic information as outlined in Van Raden (2008) and computing strategies were constructed based on Aguilar et al. (2011).
  - ROH: constructed based on shared ROH haplotypes and is constructed similar to Howard et al. (Submitted).
- Solver:
  - direct: Uses Cholesky decomposition for Matrix Inversion. When only simulating a small number of generation this is advised.
  - pcg: Uses the iterative preconditioned conjugate gradient (PCG) method and is faster than direct when the number of animals is large.
- Inverse:
  - cholesky: update previous inverse based on the algorithm presented by Meyer et al. (2012).
  - recursion: utilizing the sequential update algorithm presented by Misztal et al. (2014).

Usage: - "SOLVER_INVERSE: pedigree pcg cholesky'.

Type: - Mandatory.

Note:

Inverse calculations are only called for genomic-based relationships. When pedigree based relationships are used only the inverse is calculated and the algorithm used is based on Meuwissen & Luo (1992).

MATING

Description: - Parameters that decide how animals are mated.

Value:

- random: males and females are randomly mated.
- random5: relationships ≥ 0.5 are not allowed to mate otherwise same as random.
- random25: relationships ≥ 0.25 are not allowed to mate otherwise same as random5.
- random125: relationships ≥ 0.125 are not allowed to mate otherwise same as random25.
- minPedigree: minimize parent co-ancestries based on pedigree.
- minGenomic: minimize parent co-ancestries based on genomic information (Van Raden 2008).
- minROH: minimize parent co-ancestries based on a ROH information (Howard et al. Submitted).

Usage: - "MATING: random5".

Type: - Mandatory.

Note:

At the current time relationships are minimized when applicable using the simulated annealing algorithm. For avoidance mating's (i.e. random5, random25, random 125) any coancestry below the threshold is zeroed out and a simulated annealing algorithm is used on the resulting matrix.

CULLING

Description:

- The metric used to cull individuals and at what age an animal is removed due to old age. The direction will be the same one that is used during the selection stage.

Value:

- cull: random, phenotype, true_bv or ebv.
- max age: any number greater than 1.

Usage: - "CULLING: ebv 5".

Type: - Mandatory.

Output Options

OUTPUT_LD

Description:
- Used to determine if you need to calculate the linkage disequilibrium decay based on the r2 metric for each generation.

Value: - yes or no.

Usage: -"OUTPUT_LD: no".

Type: - Optional. Default is no.

GENOTYPES

Description:
- Used to determine if genotypes should be exported for a given generation. If you don't need the all the genotypes this will save space and reduce running time.

Value:
- 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..

Incorporation of Future Modules

The overall goal of Geno-Diver is to be able to utilize either simulated or real data to see how current mating and selection strategies impact the performance, diversity and genetic load across the time horizon in the context of an individual herd or an entire population. In order to fulfill this goal multiple modules will be incorporated in the future including the use of external breeding value predictions, the use of advanced reproductive technologies, optimal contribution selection procedures and evolutionary algorithms. Furthermore, as more sophisticated methods get developed to identify and manage lethal recessives and regions that give rise to inbreeding depression, these routines will be introduced into the simulation. A pictorial description is outlined below.

Appendix I - Output Information

The folder that contains the information generated from the simulation program contains multiple files and each one is described below.

Data Summary Files

File: Summary_Statistics_QTL

Generation: Generation number.
Quant_Founder_Start: Number of quantitative QTL from founder generation segregating.
Quant_Founder_Lost: Number of quantitative QTL from founder generation fixed.
Mutation_Quan_Total: Number of quantitative QTL from new mutations segregating.
Mutation_Quan_Lost: Number of quantitative QTL from new mutations fixed.
Additive_Var: True additive genetic variance based on Σ 2pq[a+d(q-p)]^2.
Dominance_Var: True dominance genetic variance based on Σ (2pqd)^2.
Fit_Founder_Start: Number of fitness QTL from founder generation segregating.
Fit_Founder_Lost: Number of fitness QTLs derived from founder generation fixed.
Mutation_Fit_Total: Number of fitness QTLs derived from new mutations segregating.
Mutation_Fit_Lost: Number of fitness QTLs derived from new mutations fixed.
Avg_Haplotypes_Window: Mean haplotypes contained within a haplotype window
ProgenyDiedFitness: Number of progeny that died due to fitness.

File: 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. Submitted).
homozy: Mean proportion homozygous (i.e. 1 - homozy = Observed Heterozygosity).
ExpHet: Expected Heterozygosity (i.e. Σ (1 - p^2 - q^2))
fitness: Mean multiplicative fitness value of an individual.
homozlethal: Mean number of homozygous fitness QTL classified as lethal.
hetezlethal: Mean number of heterozygous fitness QTL classified as lethal.
homozysublethal: Mean number of homozygous fitness QTL classified as sub-lethal.
hetezsublethal: Mean number of heterozygous fitness QTL classified as sub-lethal.
lethalequiv: Mean lethal equivalents (Lethal equivalents = Σ s for an animal).

File: Summary_Statistics_DataFrame_Performance:

Generation: Generation number.
phen: Mean (variance) phenotypic value.
ebv: Mean (variance) estimated breeding value.
gv: Mean (variance) true genotypic breeding value (Σ (a + d)).
bv: Mean (variance) true genotypic breeding value (Σ a).
dd: Mean (variance) true dominance deviation (Σ d).
res: Mean (variance) residual value.

File: LD_Decay:

Is a file that has the average correlation (r^2) between two SNP across a range of distances. 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 D and (r^2) values are calculated as outlined below and the subscript refers to either SNP marker 1 or 2.

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

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

Data Files

File: 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.

File: Low_Fitness:

This file that describes the animals that died due to the fitness effect.

Sire: sire of dead progeny.

Dam: dam of dead progeny.

Fitness: Fitness value of individual.

QTL_Fitness: The genotypes for Fitness QTL for the individual.

File: Marker_Map:

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

File: Master_Genotypes:

A file that contains genotypic information for individuals that survived.
ID: Identification 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).

File: QTL_new_old _Class:

A file that contains information on QTL effects and frequency across generations.
Chr: Chromosome location.
Pos: Nucleotide position of QTL.
Additive_Selective: If it is a quantitative trait it refers to the additive effect and if it is a fitness QTL it refers to the selection coefficient.
Dominance: The dominance effect for the quantitative or fitness QTL.
Type: Refers to the type of QTL (2 = quantitative trait; 4 = fitness lethal; 5 = fitness sub-lethal).
Gen: Generation at which the mutation occured.
Freq: Gene frequency across generations with a "_" as the delimeter.

File: Master_DataFrame:

A file that contains multiple statistics for individuals that survived.
ID: Identification 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 either at the culling or selection stage.
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.
Fitness: Multiplicative Fitness value of the individual.
Phen: Phenotype.
EBV: Estimated Breeding Value.
Acc: Accuracy of EBV (not included yet).
GV: True genotypic value of individual (Σ (a + d)).
BV: True breeding value of individual (Σ a).
DD: True dominance deviation of individual (Σ d).
R: Residual value of individual.

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: G_Matrix:

- The Genomic relationship matrix in binary format.

File: Ginv_Matrix:

- The inverse of the Genomic relationship matrix in binary format.

File: Linv_Matrix:

- The inverse of the cholesky matrix from the previous generation that is utilized to construct inverse relationship matrix using the method outlined by Meyer et al. (2013) to

obtain the inverse. This file is in binary format.

File: Pheno_GMatrix:

- Dataframe utilized in constructed genomic relationship matrix.

File: Pheno_Pedigree:

- Used in constructing pedigree relationship matrix.

File: Previous_Beta_PCG:

- Estimates of solutions for previous generation.

File: SNPFreq:

- Frequency of SNP across all chromosomes derived from MaCS.

Appendix II - Generation of QTL Effects

The generation of QTL effects for the quantitative and fitness traits are important parameters that can have a large impact on the simulation results. The generation of QTL effects for both types of traits was undertaken using similar methods as previous articles and simulation programs have used. At the current time additive effects are simulated from a gamma distribution and dominances effects are simulated from a normal distribution in order to generate covariance more straightforward. The use of alternative distributions to generate effects and allow for covariance to occur between the two traits will occur in the future.

Quantitative Trait:

The additive effect (a), defined as half the difference in genotypic value between alternative homozygotes, was generated from a gamma distribution. The default parameters for the gamma distribution(0.4,1.66) result in a 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 an equal probability, one of the two alleles was 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 a priori dependencies between additive and dominance effects. Therefore, the degree of dominance (h) was initially generated from a normal distribution, which allows for you to vary the proportion that are positive or negative by altering the mean. Next, dominance effects (d) were generated by multiplying the degree of dominance by the absolute value of the additive effect ($d = h|a|$). It should be noted that this 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. This is outlined in the log file near the beginning after the QTLs have been placed along the genome.

Fitness Trait:

The generation of fitness effects was divided into lethal and sub-lethal genetic architectures in order 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 one 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 QTL 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 in terms of 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 has a fitness value of 1 - hs.

The selection coefficient was generated from a gamma distribution with different parameters for lethal and sublethal. The log-file outlines the mean selection coefficient for the lethals and sub-lethals. 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 was utilized as the dominance coefficient. The log-file outlines the mean dominance coefficient for the lethals and sub-lethals. 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 QTL 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. An animal survived if their individual fitness value was larger than a random number from a uniform distribution (0,1), if it was lower the animal died.

Covariance Between Traits:

The quantitative trait and the fitness trait can be correlated due to 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

the additive effects for the quantitative trait and the selection coefficient for the sub-lethal fitness effects to be correlated due to pleiotropy. The scaling of quantitative traits results in the additive effects for quantitative trait to change and therefore an alternative algorithm was utilized called Trivariate Reduction. The Trivariate Reduction algorithm only allows the correlation to be positive. For example high values with respect to the quantitive trait would result in the two traits being antagonistic if a positive correlation is given. 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 covariance between the two traits is generated by the Y_3 values. For the QTL that have a covariance with the quantitative trait the Y_2 value is sampled for each QTL 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 founder genomic information is generated from the MaCS program (Markovian Coalescence Simulator; Chen et al. 2009). Prior to using the program it is advisable to fully understand the coalescent process and a good review paper is Hudson (1991) and Chapter 5 of Charlesworth & Charleworth (2010). The use of MaCS allows for the ancestral population to vary in terms of its population history and size. This allows for different linkage disequilibrium (LD) patterns to be simulated and therefore a variety of species can be simulated. A large number of program utilize MaCS to generate their genomic information such as AlphaSim (Hickey & Gorjanc, 2012) and ms2gs (Pérez-Enciso & Legarra, 2016) and similar MaCS populations history parameters were also utilized in this simulation program.

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 your own 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" that is placed in the folder where the program is run. 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 only an integer is given than the value is the effective population size and no population history is simulated.

The generation of sequence information may take some time to compute and the files generated from the program may be large. Due to this, it is advisable to only 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 outlined below and is in a figure 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:

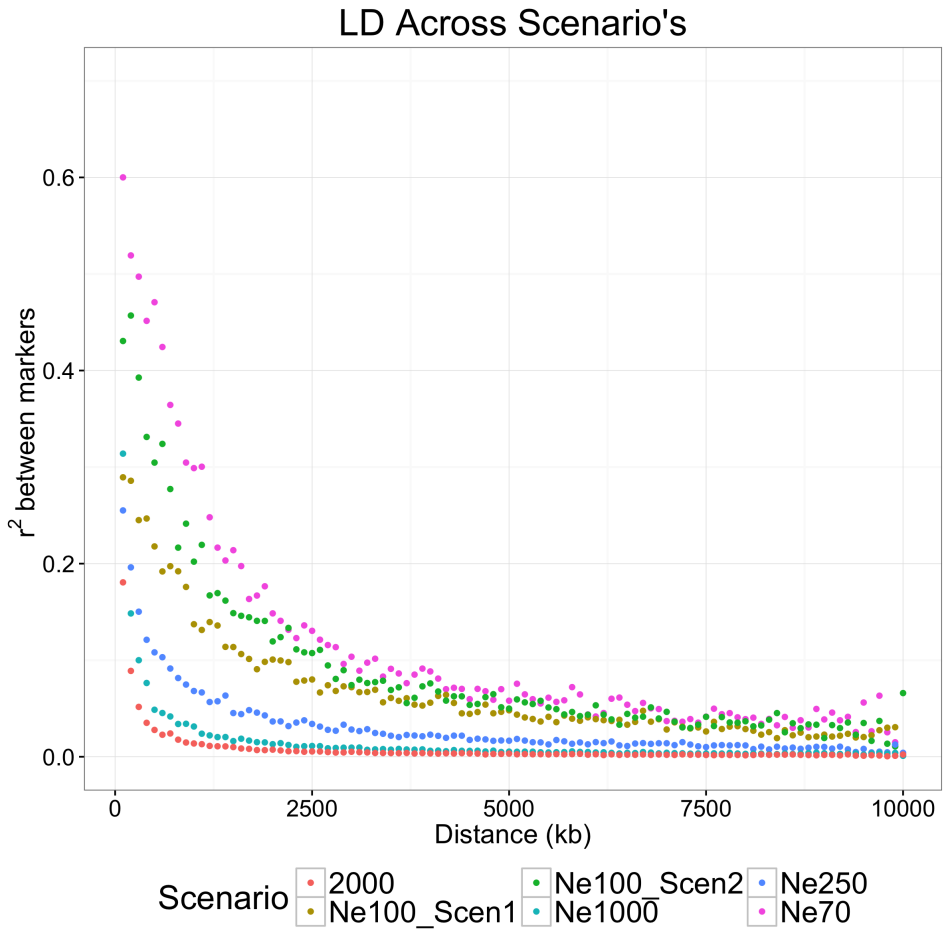
- Effective population size = "100".
- Historical population parameters: "-eN 50.00 200.0 -eN 75.00 300.0 -eN 100.00 400.0 -eN 125.00 500.0 -eN 150.00 600.0 -eN 175.00 700.0 -eN 200.00 800.0 -eN 225.00 900.0 -eN 250.00 1000.0 -eN 275.00 2000.0 -eN 300.00 3000.0 -eN 325.00 4000.0 -eN 350.00 5000.0 -eN 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., 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.
- 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.
- 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.
- 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
- 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.

Example 1 (Quantitative Trait - Single Progeny)

```
−−−−−−−|    Example 1 Parameter File     |−−−−−−−
−−−−−−−|       Starting Parameters       |−−−−−−−
START: sequence
−−−−−−−| Genome and Marker Information |−−−−−−−
CHR: 3
CHR_LENGTH: 150 150 150
NUM_MARK: 4000 4000 4000
QTL: 50
−−−−−−−|       Population Parameters     |−−−−−−−
FOUNDER_Effective_Size: Ne70
VARIANCE_A: 0.35
VARIANCE_D: 0.05
−−−−−−−| Selection and Mating Parameters |−−−−−−−
GENERATIONS: 10
INDIVIDUALS: 50 0.2 400 0.2
PROGENY: 1
SELECTION: ebv high
SOLVER_INVERSE: genomic pcg cholesky
MATING: random5
CULLING: ebv 5
```

The simulation starts by creating sequence data for 3 chromosome and the genome simulated has levels of short range LD (Ne70). This type of effective population size scenario will not create a large amount of available markers and due to this only a 12,000 marker panel is created (i.e. 4,000 markers per chromosome). For each chromosome 50 QTL are randomly placed along the genome and no fitness QTL are simulated in the current simulation. The narrow and broad sense heritability for the trait is 0.35 and 0.40, respectively. The phenotypic variance is by default set at 1.0 and therefore the residual variance is 0.6. 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, will be culled and replaced by new progeny. Selection and culling is based on keeping animals with a high estimated breeding values and a genomic relationship matrix is utilized. Lastly 1 progeny is produced for each mating pair and the number of generations simulated is 10.

Overview of Results:

- Prior to looking at any results or while the program is running go into the directory "GenoDiverFiles" and look over what parameters got called in the beginning of the log file.
- When simulating dominance be sure to check to see if the number that display over-dominance or partial dominance is what you were wanting. In this case with the given narrow and broad sense heritability, the proportion that are over-dominance is small (around 15 %).
- The table below depict the change in multiple parameters as the generations proceed and were all found in the following files:
  - "Summary_Statistics_DataFrame_Inbreeding".
  - "Summary_Statistics_DataFrame_Performance".

Generation	Phenotype	EBV	Pedigree Inbreeding	Genomic Inbreeding
0	-0.013	0.000	0	0.991
1	0.660	0.655	0	1.017
2	0.902	0.852	0.00042	1.027
3	1.137	1.045	0.00052	1.036
4	1.299	1.265	0.00349	1.046
5	1.496	1.484	0.00599	1.077
6	1.755	1.675	0.00918	1.098
7	1.845	1.85	0.01169	1.118
8	2.007	2.049	0.01587	1.149
9	2.265	2.239	0.02181	1.182
10	2.368	2.414	0.02519	1.220

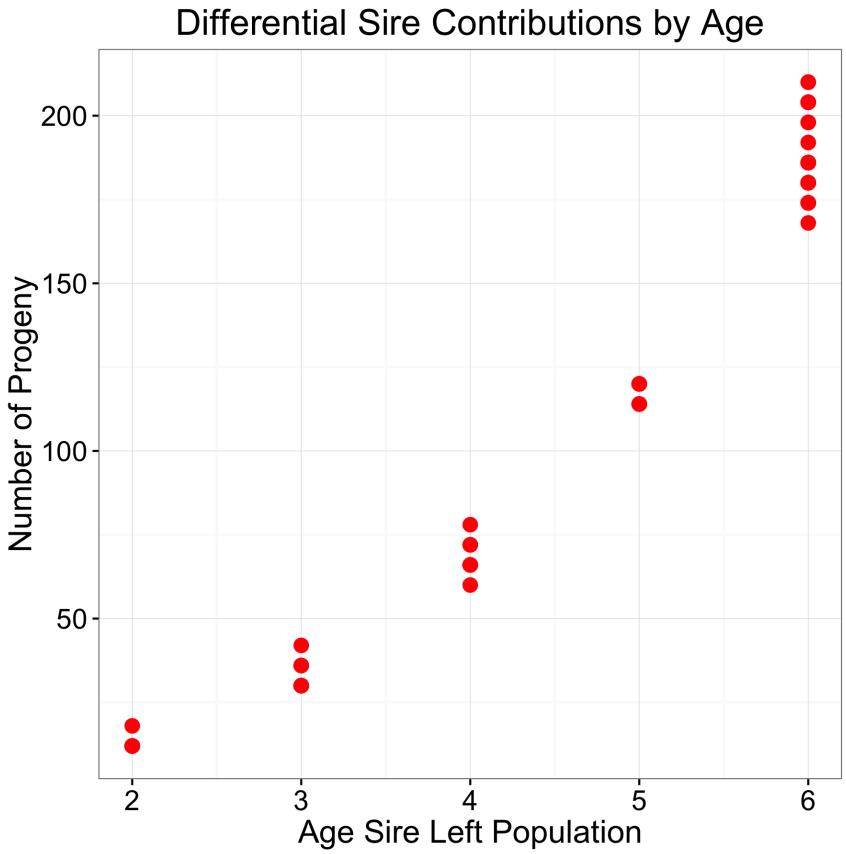
Example 2 (Quantitative Trait - Multiple Progeny)

```
--------|     Example 2 Parameter File      |--------
--------|        Starting Parameters         |--------
START: sequence
THREAD: 4
--------| Genome and Marker Information |--------
CHR: 3
CHR_LENGTH: 150 150 150
NUM_MARK: 4000 4000 4000
QTL: 50
--------|        Population Parameters       |--------
FOUNDER_Effective_Size: Ne70
VARIANCE_A: 0.35
VARIANCE_D: 0.05
--------| Selection and Mating Parameters |--------
GENERATIONS: 10
INDIVIDUALS: 50 0.2 300 0.2
PROGENY: 6
PARITY_MATES_DIST: 2.0 1.0
MAXFULLSIB: 2
SELECTION: ebv high
SOLVER_INVERSE: pedigree pcg cholesky
MATING: random125
CULLING: ebv 5
```

The simulation from Example 2 is similar in terms of its genome, markers and genetic architecture to Example 1. A few of the added complexities of the current simulation is that the number of progeny is set at 6 and therefore in order to limit selecting entire families the MAXFULLSIB parameter is set to 2. Furthermore, the number of matings allowed to a sire is dependent on the age of the animal and in the current setting older animals get a larger proportion of the total number of mating compared to younger animals. For each generation a total of 50 males and 300 females are in the population. A total of 10 and 60 (0.2 replacement rate) male and female parents, respectively, will be culled and replaced by new progeny. Selection and culling is based on keeping animals with a high estimated breeding values and a pedigree relationship matrix is utilized. The mating design is avoiding mating's with parental co-ancestries greater than 0.125 and mating below that is at random.

Overview of Results:

- The Beta distribution that is utilized in determining the number of mating that are given to a specific age group is shown graphically in the log file from lines 139 to 160. It can be seen visually that as an animals ages (approaches the max value 1.0) a greater proportion of the distribution is contained within that region.
- Furthermore, within each generation the log-file also outlines the number of times a given number of siblings were selected within a family for each generation. The use of a simple Linux "sed" statement can be utilized to grab section of the log file that are important to the user.
- The figure below depicts the number of offspring the sire produced as a function of when the sire left the population and was generated from data contained in the "Master_DataFrame" file. As shown below the number of offspring produced as a function of when the sire left the herd is not linear due to the particular Beta distribution utilized (2.0, 1.0).



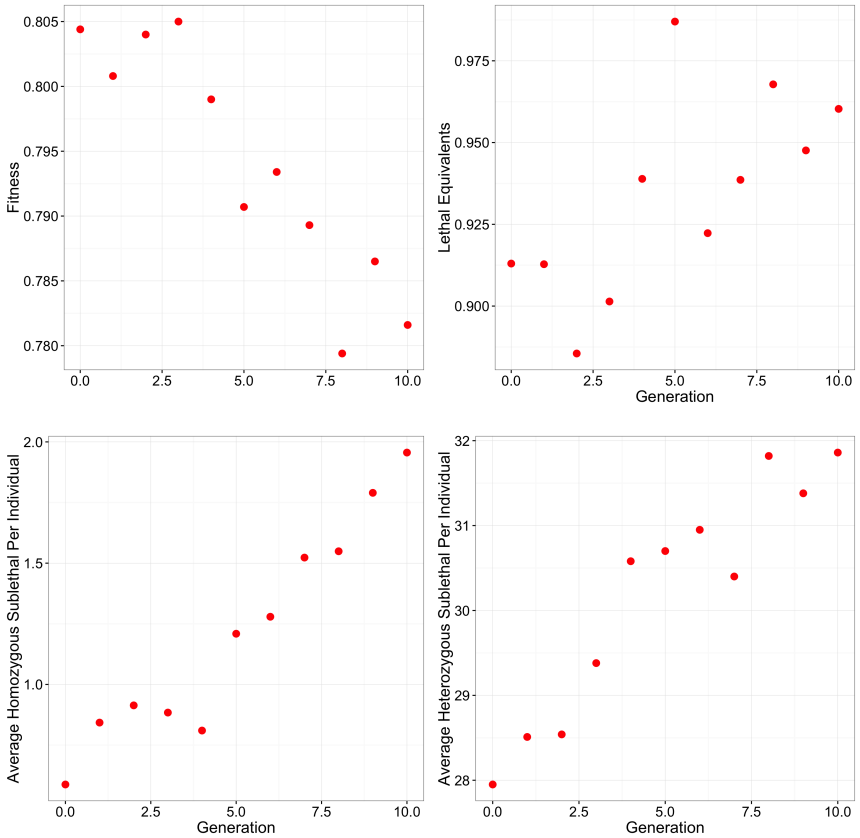
Example 3 (Quantitative Trait + Fitness Correlated)

```
−−−−−−−|     Example 3 Parameter File      |−−−−−−−
−−−−−−−|         Starting Parameters        |−−−−−−−
START: sequence
THREAD: 4
−−−−−−−| Genome and Marker Information |−−−−−−−
CHR: 3
CHR_LENGTH: 150 150 150
NUM_MARK: 4000 4000 4000
QTL: 50
FIT_LETHAL: 25
FIT_SUBLETHAL: 100
−−−−−−−|         Population Parameters       |−−−−−−−
FOUNDER_Effective_Size: Ne70
VARIANCE_A: 0.20
VARIANCE_D: 0.05
COVAR: 0.5 0.2
−−−−−−−| Selection and Mating Parameters |−−−−−−−
GENERATIONS: 10
INDIVIDUALS: 50 0.2 400 0.2
PROGENY: 1
SELECTION: ebv high
SOLVER_INVERSE: pedigree pcg cholesky
MATING: random
CULLING: ebv 5
```

This simulation highlights how a quantitative and a fitness trait can be simulated with a certain proportion (i.e. 50 %) of the QTL having both a quantitative and fitness effect. The simulation from Example 3 is similar in terms of its genome, markers and genetic architecture to Example 1. The only major difference is pedigree information is used to generate EBV instead of genomic information.

Overview of Results:

- A description of the number of progeny that died due to fitness and the number of the fitness QTL that were purged from the population is outlined in Summary_Statistics_QTL file. Furthermore, some of the results are displayed graphically below.
- In the log file it generate the mean selection coefficient and degree of dominance for the lethals and sublethals on line 132 to 137.
- When simulating fitness effects it should be noted that a certain proportion of the progeny will die due to fitness and if the population doesn't have enough progeny to stay at the value given it will exit the program. Therefore, careful consideration of the number and magnitude of fitness effects along with the number of progeny produced per mating pair needs to be addressed when constructing the parameter file.



Example 4 (Quantitative Trait Minimize Inbreeding)

```
# Delete old parameter file and director to place files if their
rm -rf ./Example4.txt | | true
rm -rf ./Example4_Output | | true
mkdir Example4_Output
# Create Parameter File
echo "--------|     Example 1 Parameter File      |--------" >>Example4.txt
echo "--------|         Starting Parameters        |--------" >>Example4.txt
echo "START: sequence" >>Example4.txt
echo "SEED: 1500" >>Example4.txt
echo "THREAD: 4" >>Example4.txt
echo "--------| Genome and Marker Information |--------" >>Example4.txt
echo "CHR: 3" >>Example4.txt
echo "CHR_LENGTH: 150 150 150" >>Example4.txt
echo "NUM_MARK: 4000 4000 4000" >>Example4.txt
echo "QTL: 50" >>Example4.txt
echo "--------|         Population Parameters      |--------" >>Example4.txt
echo "FOUNDER_Effective_Size: Ne70" >>Example4.txt
echo "VARIANCE_A: 0.35" >>Example4.txt
echo "VARIANCE_D: 0.05" >>Example4.txt
echo "--------| Selection and Mating Parameters |--------" >>Example4.txt
echo "GENERATIONS: 10" >>Example4.txt
echo "INDIVIDUALS: 50 0.2 600 0.2" >>Example4.txt
echo "PROGENY: 1" >>Example4.txt
echo "SELECTION: ebv high" >>Example4.txt
echo "SOLVER_INVERSE: pedigree pcg cholesky" >>Example4.txt
echo "MATING: random" >>Example4.txt
echo "CULLING: ebv 5" >>Example4.txt
# run Geno Driver
./GenoDiver <<BLK
Example4.txt
BLK
# Move inbreeding folder to permanent location
mv ./GenoDiverFiles/Summary_Statistics_DataFrame_Inbreeding ./Example4_Output/inbreeding_random

# Now do random 5 and start with founder
sed -i '/START: sequence/c\START: founder' Example4.txt
sed -i '/MATING: random/c\MATING: random5' Example4.txt
./GenoDiver <<BLK
Example4.txt
BLK
mv ./GenoDiverFiles/Summary_Statistics_DataFrame_Inbreeding ./Example4_Output/inbreeding_random5

# Now do random 25
sed -i '/MATING: random5/c\MATING: random25' Example4.txt
./GenoDiver <<BLK
Example4.txt
BLK
mv ./GenoDiverFiles/Summary_Statistics_DataFrame_Inbreeding ./Example4_Output/inbreeding_random25

# Now do random 125
sed -i '/MATING: random25/c\MATING: random125' Example4.txt
./GenoDiver <<BLK
Example4.txt
BLK
mv ./GenoDiverFiles/Summary_Statistics_DataFrame_Inbreeding ./Example4_Output/inbreeding_random125

# Now do minimize pedigree
sed -i '/MATING: random125/c\MATING: minPedigree' Example4.txt
./GenoDiver <<BLK
Example4.txt
BLK
mv ./GenoDiverFiles/Summary_Statistics_DataFrame_Inbreeding ./Example4_Output/inbreeding_pedigree

# Now do minimize genomic
sed -i '/MATING: minPedigree/c\:MATING minGenomic' Example4.txt
./GenoDiver <<BLK
Example4.txt
BLK
mv ./GenoDiverFiles/Summary_Statistics_DataFrame_Inbreeding ./Example4_Output/inbreeding_genomic

# Now do minimize genomic
sed -i '/MATING: minGenomic/c\MATING: minROH' Example4.txt
./GenoDiver <<BLK
Example4.txt
BLK
mv ./GenoDiverFiles/Summary_Statistics_DataFrame_Inbreeding ./Example4_Output/inbreeding_ROH
```

This example how you can use a simple bash script to change parameters in the parameter file across multiple scenarios and then output the results to a directory. Across all simulations they use the same founder sequence by first starting the random mating simulation from the sequence and then all remaining simulations were started from the founder generation. Furthermore, the same seed was utilized across all scenarios and therefore the only thing that changes across simulation replicates was how animals were mated. The graph below highlights the change across generations in pedigree across scenario's.