QMSim

User's Guide

Version 1.10

(Last update: July 12, 2013)

Centre for Genetic Improvement of Livestock
Department of Animal and Poultry Science
University of Guelph
Guelph, Canada

Mehdi Sargolzaei and Flavio S. Schenkel
July 2013

Disclaimer

The bibliographical reference for this software is:

Sargolzaei, M. and F. S. Schenkel. 2009. QMSim: a large-scale genome simulator for livestock. Bioinformatics, 25: 680-681. First published January 28, 2009, doi:10.1093 /bioinformatics/btp045.

Introduction

Linkage disequilibrium (LD) and linkage analyses have been extensively used to identify quantitative trait loci (QTL) in human and livestock species. Recently, interest in whole genome fine mapping and especially genome-wide selection has grown as a result of the dramatic increase in the number of known single nucleotide polymorphisms (SNP) and the decrease in genotyping costs. The access to dense marker maps has opened up the possibility for new approaches for fine mapping and genome wide selection. However, even though genotyping costs have substantially decreased, large scale genome-wide association studies are still costly. For this reason most of studies suffer from small sample sizes or from low marker density. Simulation is a highly valuable tool for assessing and validating proposed methods for QTL mapping and genome wide selection at very low cost, allowing also for the prediction of future changes in genetic parameters. During the last few decades, simulation has played a major role in population genetics and genomics.

Simulating and analyzing livestock genomic data differ in several aspects from analyses carried out in humans. For instance, in livestock a common strategy for detecting QTL is either to use multigenerational pedigrees, large family sizes, in which artificial insemination is practiced, or to design a crossbreeding program, such as F2 and back crosses (Andersson, 2001). More importantly, human populations have been experiencing an expansion in effective population size (Ne), while Ne in livestock populations has decreased. Consequently LD in livestock extends over longer distances than in humans. Moreover, combined LD and linkage QTL mapping has attracted more attention in livestock due to strong family structure (Meuwissen et al., 2002). Therefore, the population structure is crucial to identify and correctly interpret the associations between molecular and phenotypic diversity (Pritchard and Rosenberg 1999; Buckler and Thornsberry 2002).

QMSim was designed to simulate a wide range of genetic architectures and population structures in livestock. Large scale genotyping data and complex pedigrees can be efficiently simulated. Simulation is basically carried out in two steps. In the first step, historical generations are simulated to create desirable level of LD and, in the second step, recent population structures are generated, which can be very complex. QMSim allows for a wide range of parameters to be incorporated in the simulation models in order to produce appropriate simulation data.

In order to create initial LD and to establish mutation-drift equilibrium, a historical population can be simulated by considering only two evolutionary forces: mutation and drift. Mutation constantly introduces new variation and genetic drift shifts the variation to fixation. Here, the mating system is based on the union of gametes randomly sampled from the male and female gametic pools. After generating historical generations, the recent populations are simulated. Expansion and contraction of both historical and recent populations are allowed. In the recent generations selection and culling can be implemented based on different criteria such as phenotypes, true genetic values and estimated breeding values for single trait with predefined heritability and phenotypic variance. Estimated breeding values may be generated using four different approaches: 1) Best linear unbiased prediction (BLUP) via an animal model, 2) based on predefined accuracy, 3) approximated based on the number of offspring with record and 4) breeding values for each generation can be externally estimated by any method and then be inputted to QMSim. Mating design can be random, assortative (positive or negative) or optimized to minimize or maximize inbreeding. The mating design that maximizes inbreeding allows one to quickly create an inbred line. Optimization of inbreeding is carried out using simulated Annealing method (Sonesson and Meuwissen, 2000). The program can simulate sex limited traits, such as milk production. Owing to the object-oriented programming, it is easy to simulate multiple populations with different structures and selection criteria. QMSim has flexibility in simulating wide range of population structures. For example, in livestock, some of QTL mapping designs involve line crosses produced from inbred lines with divergent phenotypes. In this case, the associated mutations are expected to have high frequencies in opposite directions. Another example is simulation of two lines coming from the same base population to assess accuracy of genomic breeding values for a particular genomic evaluation method. Here, one line can be treated as training set and the other one as validation set. These scenarios can readily be simulated by QMSim.

A wide range of parameters can be specified for simulating the genome, such as: mutation rate, crossover interference, number of chromosomes, markers and QTL, location of markers and QTL, number of alleles, allelic frequencies, allelic effects, missing marker rate, and genotyping error rate. This flexibility permits for a wide variety of genetic architectures to be considered. No allelic effects are simulated for markers, so they are treated as neutral. For QTL, additive allelic effects can be sampled from gamma, normal or uniform distributions. Alternatively, predefined relative additive variance for each QTL can be supplied by the user.

One important aspect of genome simulation is to model the recombination appropriately to produce realistic level of LD, given the recent and past population structures. QMSim models crossover process, using a Poisson model. This is done by sampling the number of crossovers from a Poisson distribution and then the crossovers are located randomly across the chromosome. Because the input map is in centiMorgan it is straightforward to take into account the pattern of recombination hotspots and coldspots along the genome by adjusting the distances between markers. Moreover a simple algorithm is applied to account for crossover interference. To establish mutation-drift equilibrium in the historical generations either infinite-allele mutation model or recurrent mutation model is used. The infinite-allele mutation model assumes that a mutation creates a new allele, while the recurrent mutation model assumes that a mutation alters an allelic state to another and does not create a new allele. In the recurrent model, transition probabilities from one allelic state to another are assumed equal. Different mutation rates for markers and QTL can be specified. Effect of mutant QTL alleles can be drawn from gamma, normal or uniform distribution. The number of mutations is sampled from a Poisson distribution and it is assumed that mutation rates are equal for all loci within markers and within QTL.

无限等位基因突变模型假设突变产生新的等位基因，而复发突变模型假设突变将等位基因状态改变为另一个并且不产生新的等位基因。

The most important parameters affecting CPU time and memory are the number of loci (markers and QTL) and population size. However, depending on the simulation scenario, other parameters such as high mutation rate, calculation of LD for dense marker panels, calculation of inbreeding, saving huge outputs, etc might become a computational bottleneck. QMSim provides the user with enough options to manage the outputs or turn off unwanted computations. For example, one may optionally save genotypes for last few generations avoiding large outputs.

There are two versions of QMSim, a 8-bit version and a 16-bit version. The 8-bit version allows for maximum 255 alleles per locus while 16-bit version allows for maximum 65,535 alleles per locus. The 16-bit version might be useful for specific scenarios such as one where unique alleles for each founder are to be simulated.

QMSim main features:

- ✓ Simulates historical generations to establish mutation-drift equilibrium and create linkage disequilibrium.
- ✓ Recombination is appropriately modeled. Interference is allowed.
- ✓ Multiple chromosomes, each with different or similar density of markers and QTL maps, can be generated.
- ✓ Very dense marker map and also sequence data can be simulated.
- ✓ Missing genotypes and genotyping errors can be simulated.
- ✓ Markers can be either SNP or microsatellites.
- ✓ Males and females can have different genome length in cM.
- ✓ Unbalanced sex ratio is also allowed in historical populations.
- ✓ Additive QTL effects can be simulated with different distributions, such as gamma, normal or uniform.
- ✓ In addition to QTL effects, polygenic effect can be included.
- ✓ After mutation-drift equilibrium, polymorphic marker panel with pre-defined MAF threshold and QTL can be selected.
- ✓ 16-bit version allows for assigning unique alleles to each founder.
- ✓ Complete linkage disequilibrium in the first historical generation can be generated.
- ✓ Calculates LD decay in specified generations.
- ✓ Population expansion or bottleneck is allowed for both historical and recent populations.
- ✓ Selection and culling of breeding population can be carried out based on different criteria, such as phenotypes, estimated breeding values (can be calculated and inputted by the user) and true breeding values.

- ✓ More than one litter size with predefined probabilities can be considered.
- ✓ Multiple recent populations or lines can be simulated. Crossing between populations or lines is allowed.
- ✓ Multiple populations can be analyzed jointly for estimating breeding values and computing inbreeding.
- ✓ Creates detailed output files. Outputs can be customized to avoid saving unnecessary data.
- ✓ Equipped with fast and high-quality pseudo-random number generators.
- ✓ Computationally efficient.

Computing environment:

The code is written in C++ language using object oriented techniques and the application runs on Windows and Linux platforms.

Download:

The executable files are available for Windows and Linux at
http://www.aps.uoguelph.ca/~msargol/qmsim/

Input parameter file

The program requires a parameter file, in which various parameters for the simulation should be specified. The input parameter file must be in ASCII format. The C++ like comments can be used to add descriptive anywhere in the parameter file. The parameter file consists of five main sections. The first part describes global parameters, the second part describes the historical population, the third part describes parameters for subpopulations and generations, the fourth part contains genome parameters and the fifth part is related to the output options. The order of commands within each section is not normally important. All commands end with a semicolon. Failure to include the semicolon will cause an error message and program exits.

1- GLOBAL PARAMETERS SECTION:

title

Description:	Set a title.	
Usage:	title = "string";	
	string	indicates an arbitrary title.
Type:	Optional	
Default:	None	

seed

Description:	Set a user-specified seed.	
Usage:	seed = "filename";	
	filename	indicates a seed file name.
Type:	Optional	
Default:	"seed"	
Note:		

Initialization of the random number generator requires a seed. If seed file is not specified, the random number generator will be seeded from the system clock. For each run the initial seed numbers will be backed up in output folder. This allows one to generate the same simulated data. Therefore, one can backup only the seed file instead of backing up whole data and, if needed, generate the same data set in the future or on a remote system with the same version of the software as the one the seed number was generated.
At beginning of each run initial seed numbers are written in “seed.prv”. For next run, the seed derived from the system clock will be compared with “seed.prv” to ensure that the same seed is not used. “seed.prv” file can be accessed by only one process at a time, therefore if two jobs are started at the same time and in the same folder they will be seeded with different seed.
The Mersenne Twister algorithm, which is a high-quality fast random number generator, is used to generate random numbers (Matsumoto and

Nishimura, 1998).

---

nthread

Description:	Number of threads for parallel processing.	
Usage:	nthread = value;	
	value	is the number of threads.
		Range: 1 – 200
Type:	Optional	
Default:	Auto	
Note:	Number of CPUs is automatically detected. If you are running on a busy server then you may want to restrict the number of threads to available CPUs. Please note that you may not observe a linear relation between speed and the number of threads since the main bottleneck is memory access. Some functions run only on single processor like GBLUP function or Inbreeding Optimization function. It is important to note that when QMSim runs with the same seed number but on two machines with different number of CPU cores, different data sets may be generated. To be able to re-generate the same data set using a stored seed on two machines, the two machines should have same number of CPU cores or "nthread" should be set to 1 for both runs.	

---

nrep

Description:	Number of replicates.	
Usage:	nrep = value;	
	value	is the number of replicates.
		Range: 1 – 20,000
Type:	Mandatory	
Note:	While the simulation is in process, user can adjust (decrease or increase) the number of replicates by creating file "nrep.txt" in output folder. The content of the file should be one integer number only, which indicates the adjusted number of replicates.	

---

h2

Description:	Overall heritability (Polygenic plus QTL)	
Usage:	h2 = value;	
	value	is heritability.
		Range: 0 – 1
Type:	Mandatory	

---

qtlh2

Description:	QTL heritability

Usage:	qtlh2 = value;	
	value	is heritability attributable to QTL. Range: 0 – 1
Type:	Mandatory	
Note:	In the last historical generation, the QTL allelic effects are scaled to ensure the desired QTL variance. If qtlh2 is set to a value smaller than h2 then a polygenic (infinitesimal) effect is also simulated. When qtlh2 is zero, no QTL effect is simulated and therefore pure polygenic effect is simulated.	

phvar

Description:	Phenotypic variance	
Usage:	phvar = value;	
	value	is phenotypic variance. Range: 0 – 10,000,000
Type:	Mandatory	

no_male_rec

Description:	No record for males will be simulated.
Usage:	no_male_rec;
Type:	Optional
Note:	This option is used to simulate a sex limited trait like milk yield. When males do not have records, but the selection or culling design is based on phenotypes, males will be randomly selected or culled.

skip_inbreeding

Description:	Inbreeding will not be calculated.
Usage:	skip_inbreeding;
Type:	Optional
Note:	Inbreeding is calculated in recent populations only. Calculation of inbreeding is a time consuming task in large populations. To gain speed in simulating large populations, one can safely turn off calculation of inbreeding when: 1) no polygenic effect is simulated (i.e., all the genetic variance is explained by QTL), 2) BLUP-breeding values are not estimated and 3) matings are not optimized to minimize or maximize inbreeding. Inbreeding can be skipped for the above mentioned situations but results should be interpreted with caution.

joint_pop

Description:	When more than one population is defined, populations will be analyzed jointly.
Usage:	joint_pop;

Type:	Optional
Default:	If not specified populations will be analyzed separately.

---

system

Description:	Runs a system executable file at the end of each replicate.		
Usage:	system = "command" /option;		
	command	is a system command.	
	option	/nrep	passes the current replicate number as the last argument to the external program.
Type:	Optional		
Note:	System command is useful when simulating larger number of replicates each with large genotype data. After each replicate the genotype data can be analyzed with an external program and then be removed.		

2- HISTORICAL POPULATION SECTION:

In order to create initial LD and to establish mutation-drift equilibrium, a historical population can be simulated by considering equal number of individuals from both sexes, discrete generations, random mating, no selection and no migration. Offspring are produced by random union of gametes, each from the male and female gametic pools. Expansion and contraction of the historical population size are allowed. Historical population is simulated based on forward-time approach. The current version of QMSim can only simulate a single historical population.

---

begin_hp & end_hp

Description:	Beginning and end of the historical population parameters
Usage:	begin_hp;
	end_hp;
Type:	Mandatory

---

hg_size

Description:	Historical generation sizes	
Usage:	hg_size = v1 [v2] …;	
	v1	indicates the historical generation size
		Range: 2 – 100,000
	v2	indicates the historical generation number
		Range: 0 – 150,000
Type:	Mandatory	
Example1:	hg_size = 500 [0] 500 [1000];	
	One thousand historical generations with constant size of 500	

Example2:	hg_size = 2000 [0] 400 [1000]; One thousand historical generations with gradual decrease in size from 2000 to 400. The decrease is linear.
Note:	By using appropriate historical generation sizes different form of historical bottleneck or expansion can be simulated. Generation number starts always from zero therefore the first historical generation number must be always zero.

---

nmfhg

Description: The number of males in the first historical generation

Usage: nmfhg = value;

value	indicates the number of males in the first historical generation. Range: 1 – 100,000

Type: Optional

Default: equal number of individuals from each sex

Note: The sex ratio will be constant across historical generations. However, if one wishes the sex ratio can be changed in the last historical generation only.

---

nmlhg

Description: The number of males in the last historical generation

Usage: nmlhg = value;

value	indicates the number of males in the last historical generation. Range: 1 – 100,000

Type: Optional

Default: same sex ratio as the first historical generation

3- POPULATION SECTION:

This section can contain more than one recent population and parameters for each population should surrounded by begin_pop and end_pop. Founders for each population can be chosen from the historical population or from the other recent populations. Multiple recent populations can be analyzed separately (by creating one distinct pedigree for each population) or jointly (by creating one pedigree for all populations) for inbreeding and estimated breeding values. To perform joint analysis use joint_pop command. Up to 1,000 recent populations can be simulated.

---

begin_pop & end_pop

Description: Beginning and end of the population parameters

Usage: begin_pop = "string";
end_pop;

	string	indicates an arbitrary name of population (maximum 20 characters).
Type:	Mandatory	

Choosing founders for a population:

begin_founder; and end_founder; should be used at the beginning and end of this subsection.
For the first defined recent population founders must come from the last historical generation. However, for subsequent populations (if defined) founders can be also chosen from one or more (up to 10) previously defined populations. Founders can be chosen from specified generations of previous populations based on different criteria. One can simulate migration by choosing founder groups from different populations (see example 12).
Note that for separate analysis of multiple populations (when EBV is to be estimated) EBVs for founders of each population are estimated separately and based on no pedigree information, disregarding which population they are selected from.
If founders from two or more populations are to be selected (not randomly) from a particular generation of the population, then the order in which populations are defined becomes important. Note that populations are processed in the order they are defined.

begin_founder & **end_founder**

Description:	Beginning and end of parameters for choosing founders

Usage:	begin_founder;
	end_founder;

Type:	Mandatory

male

Description:	Selecting base males from particular population and generation		
Usage1:	male [n = v1, pop = "s1", gen = v2, select = v3 option1] option2;		
Usage2:	male [n = v1, pop = "hp", select = v3 option1] option2;		
	v1	is the number of male individuals to be selected.	
		Range: 1 –50,000	
	s1	is the name of population.	
	hp	historical population (last historical generation).	
	v2	is the generation number.	
		Range: 0 – 2,000	
	v3	indicates the type of selection (optional).	
		Range: rnd, phen, tbv and ebv	
		Default: rnd	
	option1	/l	to select low values
		/h	to select high values
		Default	/h

	option2	/not_founder_yet	to select individuals that have not appeared as founders of any recent populations
Type:	Mandatory		
Note:	options /l and /h cannot be used when selection method is random. In Usage2, founders are always chosen from the last historical generation. When choosing founders from the last historical generation, selection method cannot be based on 'ebv'. This command can be used more than once in order to choose male founders from different generations or populations.		

female

Description:	Selecting base females from particular population and generation
Usage:	the same as for males

ng

Description:	the number of generations for current population	
Usage:	ng = value;	
	value	is the number of generations for current population. Range: 0 – 2,000
Type:	Mandatory	

ls

Description:	Litter size	
Usage1:	ls = value;	
Usage2:	ls = v1 v2 [p2] v3 [p3] ...;	
	value v1 v2 v3	are the litter sizes or the numbers of progeny per dam. Range: 0 – 1,000
	p2 p3	are the probabilities of the litter sizes. Range: 0 – 1
Type:	Mandatory	
Note:	The litter size is considered to not be controlled genetically. The litter sizes are uniformly sampled based on the provided litter size probabilities. In usage2, the probability of the first litter size is not required because the sum of probabilities is 1.	

pmp

Description:	The proportion of male progeny	
Usage:	pmp = value option;	
	value	is the proportion of male progeny. Range: 0 – 1

	option	/fix	sex is assigned at random, but it is ensured that observed proportion of males will be equal to the specified value.
		/fix_litter	this causes sex ratio be fixed within litters (progeny of a dam). For example, the command “pmp = 0.5 /fix_litter;”, when litter size is 2 will always generate one male and one female progeny per dam.
Type:	Optional		
Default:	0.5		

---

md

Description:	Mating design		
Usage:	md = value option;		
	value	is the type of mating design Range: rnd, rnd_ug, minf, maxf, p_assort, n_assort	
	option	/phen	assortative mating base on phenotype
		/ebv	assortative mating base on ebv
		/tbv	assortative mating base on tbv
Type:	Optional		
Default:	rnd		

Note: Option will be used only for assortative mating design (i.e., p_assort and n_assort).

rnd_ug stands for random union of gametes. With this design offspring are produced by union of two gametes, one randomly sampled from the male gametic pool and one from female gametic pool. Therefore a dam can mate with more than one sire in each generation. **Note that with rnd_ug mating design, ‘ls’ (litter size) is treated as average number of progeny per dam.**

p_assort and n_assort stand for positive assortative and negative assortative, respectively. In the assortative mating design, sires mate to dams based on similarity (positive) or dissimilarity (negative). Similarity and dissimilarity can be based on true breeding values (tbv), estimated breeding values (ebv) or phenotypes (phen).

Designs minf and maxf minimizes and maximizes inbreeding, repectively. In the optimized mating design, pairs of mates are chosen so that inbreeding is minimized or maximized in the next generation. To minimize or maximize inbreeding, simulated annealing method is used (Sonesson and Meuwissen, 2000). The simulated annealing method is an adaptation of the Metropolis-Hastings algorithm for the global optimization problem. The initial temperature was set to 0.5 and then decreased by a factor of 0.9.

In culling of parents and selection of progeny, minimization or maximization of inbreeding is not considered.

---

sr

Description:	Sire replacement		
Usage:	sr = v1 v2 [v3] ... option;		
	v1	is the proportion of sire population to be culled	
		Range: 0 – 1	
	v2	is the sire population growth rate	
		Range: -1 – 5	
	v3	is starting generation number	
	option	/e	exponential growth rate
Type:	Optional		
Default:	1 (i.e., discrete generations)		
Example1:	sr = 0.5; 50% of sires would be replaced in all generations (sire population growth rate is zero)		
Example2:	sr = 0.5 0.2; 50% of current sire population would be culled in each generation and sire population size grows constantly across generations at rate 0.2. The number of male progeny to be selected would be larger than the number of culled sires because of increase in the sire population size		
Example3:	sr = 0.4 [1] 0.5 [5]; 40% of sire population would be culled from generation 1 to generation 5 and 50% from generation 5 afterward (sire population growth rate is zero)		
Example4:	sr = 0.5 -0.1 [1] 0.5 0.0 [5]; 50% of current sire population would be culled in each generation but sire population growth rates are -0.1 for the first 5 generation and zero afterward		
Note:	If there are not enough male progeny to replace the culled sires, the culling rate will be reduced. If sire population growth rate is negative and if the decline rate is larger than the culling rate, the culling rate will increase. In both situations, a warning message will be displayed. If option /e is specified, size of the current generation increases/decreases by v2 * size of previous generation. If this option is not specified size of the current generation increases/decreases by v2 * size of starting generation (i.e., generation v3).		

---

dr

Description:	Dam replacement	
Usage:	dr = v1 v2 [v3] ... option;	
	v1	is the proportion of dam population to be culled
		Range: 0 – 1
	v2	is the dam population growth rate
		Range: -1 – 5

	v3	is starting generation number	
	option	/e	exponential growth rate
Type:	Optional		
Default:	1 (i.e., discrete generations)		
Example:	See sr for examples		

---

sd

Description:	Selection design		
Usage:	sd = value option;		
	value	is the type of selection design	
		Range: rnd, phen, tbv and ebv	
	option	/l	to select low values
		/h	to select high values
		Default	/h
Type:	Optional		
Default:	rnd		
Note:	options /l and /h cannot be used when selection design is random. option /l forces the program to select individuals with low phenotype, tbv or ebv values and option /h does the opposite.		

---

cd

Description:	Culling design		
Usage:	cd = value option;		
	value	is the type of culling design	
		Range: rnd, phen, tbv, ebv and age	
	option	/l	to select low values
		/h	to select high values
		Default	/h
Type:	Optional		
Default:	age		
Note:	options /l and /h cannot be used when culling design is random or age. option /l forces the program to select individuals with low phenotype, tbv or ebv values and option /h does the opposite. If culling design is age and selection design is not random then culling is first based on age and second based on the selection design with opposite direction within age group. For instance, if culling is set to age and selection is set to tbv /h, the culling within each age group is based on tbv /l.		

---

ebv_est

Description:	Breeding value estimation method	
Usage1:	ebv_est = blup option;	
Usage2:	ebv_est = approx option1;	
	option1	/mnp_male v1

/mnp_female v2

v1, v2 are minimum number of progeny with record for male and female individuals (default is zero)
Range: 0 – 1,000

Usage3: ebv_est = accur v option;

Usage4: ebv_est = accur_male v1 accur_female v2 option;

v, v1, v2 are predefined accuracies
Range: 0 – 1

option /true_av true additive genetic variance for each generation will be considered

Usage5: ebv_est = external_bv "external solver filename";

Type: Optional

Default: None

Note: If ebv_est statement is specified, QMSim will estimate breeding value for each individual.
Breeding value can be defined as a measure of the additive genetic value of an individual as a parent.
Three methods are implemented to estimate breeding values:

1) blup
The best linear unbiased prediction (BLUP) of breeding values are obtained by Henderson’s (Henderson, 1975) mixed linear model. The BLUP predictor has the smallest prediction error variance among all possible linear unbiased predictors. Two pieces of information are used: phenotypic records and pedigree data. The numerator relationship matrix (**A**) is used in the following mixed model equations to derive BLUP of random additive effects (including polygenes and QTL):

$$\left[\mathbf{Z}'\mathbf{Z} + \mathbf{A}^{-1}\frac{\sigma_e^2}{\sigma_a^e}\right]\left[\hat{\mathbf{a}}\right] = \left[\mathbf{Z}'\mathbf{y}\right],$$

where **y** is the vector of phenotypic records, **Z** is the incidence matrix relating the records to the random additive effects (**a**), σ_e^2 is the residual variance and σ_a^2 is the additive genetic variance.
The mixed model equations are solved by the conjugate gradient method.
For multiple population case, all individuals are included in the mixed model equations.

2) approx
When there is enough number of half-sib offspring per individual, EBV can be approximated with a good accuracy. More importantly, when dealing with a sex limited trait like milk yield, records on relatives such as half-sib

progeny can be used to generate estimated breeding values for the sires, using the following simple formula:

$$EBV = \frac{N \times (TBV + \frac{2 \times NRND \times \sigma_e^*}{\sqrt{N}})}{N + \frac{4 - h^2}{h^2}}$$

where N is the progeny size, TBV is true breeding value, σ_e^* is residual standard deviation for a sire model ($\sigma_e^* = \sqrt{\sigma_e^2 + \frac{3}{4}\sigma_a^2}$) , h^2 is heritability and *NRND* is normal random deviate.

It is assume that progeny are paternal half-sibs and additive genetic variance is constant across generations. For individuals with no progeny, parental average is calculated (for founders own record is used to estimate EBV at the start). If minimum number of progeny with record is specified and an individual has fewer progeny with record in the pedigree then some dummy progeny will be considered to reach the minimum number.

3) accur

In this case, breeding values are approximated based on a predefined accuracy. Some methods for estimating breeding values might be very time consuming such as Bayesian methods to estimate breeding values using dense genetic markers (Meuwissen et al., 2001). But, knowing the accuracy for a method, computing breeding values with predefined accuracy allows one to roughly evaluate the future changes in the genetic parameters based on the assumed method.

Note that accuracy would be constant across generations if true additive genetic variance for each generation is considered, otherwise the initial additive genetic variance will be used then the loss of genetic variation will affect the accuracy.

4) external_bv

With this option, user should estimate breeding values from temporary data outputted by QMSim in each generation. The estimated breeding values should then be provided to QMSim for selection and culling.

In each generation, QMSim writes the temporary data in "data.tmp". Then it executes the "external solver". The external solver should read "data.tmp" and genotypes information (if required) and calculates breeding values then it should write EBVs in "my_bv.txt". After external solver finished, QMSim reads "my_bv.txt" and use the EBVs to select or cull parents for simulation of next generation.

File "data.tmp" contains a header line, animals ID, sire ID, dam ID, FLAG,

sex, generation number, number of male and female progeny, inbreeding, homozygosity, phenotype, residual, Polygene and QTL. Flag indicates whether the animal is sire (S), dam (D) or progeny (P).
File "my_bv.txt" should contain a header line, animals ID and EBV.

Population specific parameters for saving outputs:

begin_popoutput & end_ popoutput

Description: Beginning and end of output options for the population
Usage: begin_popoutput;
end_ popoutput;
Type: Mandatory

crossover

Description: Save all crossovers that occurred in the recent population
Usage: crossover option;

option	/gen v1 v2 v3 ...	save crossovers occurred in specified generations. v1 v2 v3 ... are generation numbers for which crossovers should be saved.

Type: Optional
Note: File name = “population name”_crosso_”replicate number”.txt

data

Description: Save individual’s data except their genotypes
Usage: data option;

option	/gen v1 v2 v3 ...	save data for specified generations. v1 v2 v3 ... are generation numbers for which data should be saved.

Type: Optional
Note: File name = “population name”_data_”replicate number”.txt
The data file contains pedigree, sex, generation number, number of male and female progeny, inbreeding, homozygosity, and genetic and residual values.

stat

Description: Save brief statistics on simulated data
Usage: stat;
Type: Optional
Note: File name = “population name”_stat_”replicate number”.txt
The stat file contains mean and standard deviation for inbreeding, homozygosity, phenotypes and its components, and information on structure

of simulated population.

allele_freq

Description:	Save allele frequencies		
Usage:	allele_freq option;		
	option	/mafbin v	v is the number of bin for minor allele frequency distribution
		/gen v1 v2 v3 ...	save allele frequencies for specified generations. v1 v2 v3 ... are generation numbers for which allele frequencies should be saved.
Type:	Optional		
Note:	Marker file name = “population name”_freq_mrk_”replicate number”.txt QTL file name = “population name”_ freq_qtl_”replicate number”.txt To save memory only non-zero allele frequencies are stored. Minor allele frequency distribution is printed at the end of the file.		

genotype

Description:	Save genotype data		
Usage:	genotype option;		
	option	/binary	save genotypes in binary format (to save memory)
		/snp_code	save SNP as genotype code (i.e. 0 2 3 4 5; 0=homozygote for allele 1, 2=homozygote for allele 2, 3=heterozygote the first allele is from sire and the second allele is from dam, 4=heterozygote the first allele is from dam and the second allele is from sire, 5=missing)
		/seq	save SNP and bi-allelic QTL in the same file with genotype code: 0 2 3 4 5 (see /snp_code for genotype code description)
		/gen v1 v2 v3 ...	save genotype for specified generations. v1 v2 v3 ... are generation numbers for which genotype should be saved.
Type:	Optional		
Note:	Marker file name = “population name”_mrk_”replicate number”.txt QTL file name = “population name”_qtl_”replicate number”.txt The file contains animal ID and genotypes (two alleles for each locus). The order of genotypes is as same as the order in linkage map. The format of binary file is: 4 bytes for animal ID and 4 bytes for each		

marker (2 bytes for each allele). This format is repeated for every individual. Options binary and snp_code cannot be used together.

---

ld

Description:	Save linkage disequilibrium stat		
Usage:	ld option;		
	option	/maft v	v is minor allele frequency threshold for ld calculation. Markers with minor allele frequency less than v will be excluded. Default is 0.05.
		/max_distance v	v is the maximum distance between two markers. Marker pairs with greater distance are ignored. Default is 5 cM.
		/nmrk v	v is the number of markers to be considered for LD calculation. Marker will be selected randomly.
		/chr v1 v2 v3 …	Calculate LD statistics for specified chromosome. v1 v2 v3 ... are chromosome numbers. Default is the longest chromosome.
		/gen v1 v2 v3 ...	save ld data for specified generations. v1 v2 v3 ... are generation numbers for which ld data should be saved
Type:	Optional		
Note:	File name = “population name”_ld_”replicate number”.txt		

The extent of LD is an important factor in association studies. Different LD measures can give substantially different estimates of LD. Each measure might be preferable in certain situations. Therefore extend of LD is calculated based on three different measures of LD and the user should decide which one to use.

1) Pooled square of the correlation between loci *A* and *B*:

$$r^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} p(A_i)p(B_j)r_{ij}^2$$

where

$$r_{ij}^2 = \frac{D_{ij}^2}{p(A_i)(1-p(A_i))p(B_j)(1-p(B_j))}$$

and

$$D_{ij} = p(A_iB_j) - p(A_i)p(B_j)$$

2) Lewontin's LD measure:

$$D' = \sum_{i=1}^{m}\sum_{j=1}^{n} p(A_i)p(B_j)\left|\frac{D_{ij}}{D_{ij}^{\max}}\right|$$

If $D_{ij} < 0$

$$D_{ij}^{\max} = \min(p(A_i)p(B_j),(1-p(A_i))(1-p(B_j)))$$

If $D_{ij} \geq 0$

$$D_{ij}^{\max} = \min(p(A_i)(1-p(B_j)),(1-p(A_i))p(B_j))$$

3) Standardized chi-square measure:

$$\chi^2 = 2N\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{D_{ij}^2}{p(A_i)p(B_j)} \Big/ 2N\min(m,n)$$

where 2N is the number of haplotypes.

4- GENOME SECTION:

begin_genome & **end_genome**

Description:	Beginning and end of the genome section
Usage:	begin_genome;
	end_genome;
Type:	Mandatory

Defining chromosomes:

begin_chr & **end_chr**

Description:	Beginning and end of chromosome parameters	
Usage:	begin_chr = value;	
	value	is the number of chromosomes with the same parameters
		Range: 1 – 500
	end_ chr;	
Type:	Mandatory	
Note:	Chromosomes with different parameters should be defined separately. In total 500 chromosomes can be simulated.	

chrlen

Description:	Chromosome length	
Usage:	chrlen = value;	
	value	is the chromosome length

		Range: 1 – 10,000 cM	
Type:	Mandatory		

nmloci

Description:	Number of marker loci on the chromosome	
Usage:	nmloci = value;	
	value	is the number of marker loci
		Range: 0 –400,000
Type:	Mandatory	

mpos

Description:	Marker positions		
Usage1:	mpos = even option;		
	even	evenly spaced	
Usage2:	mpos = rnd option;		
	rnd	random; samples from uniform distribution in each replicate	
Usage3:	mpos = rnd1 option;		
	rnd1	random; samples from uniform distribution in the first replicate only (fixed across replicates)	
	option	/start v	v is start position
		/end v	v is end position
Usage4:	mpos = pd v1 v2 v3 …;		
	pd	predefined	
	v1 v2 v3 …	are marker positions	
Type:	Mandatory		

nma

Description:	Number of marker alleles in the first historical generation	
Usage1:	nma = all v;	
	all	all loci would have same number of alleles initially
	v	number of alleles
Usage2:	nma = rnd v1 v2 v3 …;	
	rnd	random; samples from uniform distribution in each replicate
	v1 v2 v3 …	number of marker alleles
Usage3:	nma = rnd1 v1 v2 v3 …;	
	rnd1	random; samples from uniform distribution in the first replicate only (fixed across replicates)
	v1 v2 v3 …	number of marker alleles
Usage4:	nma = pd v1 v2 v3 …;	
	pd	predefined
	v1 v2 v3 …	number of marker alleles

		Range: 1 – 255 for 8-bit version and 1 – 65535 for 16-bit version
Usage5:	nma = unique;	number of alleles is 2 times of the number of individuals in the first historical generation
Type:	Mandatory	
Note:	In the subsequent historical generations, the number of alleles per locus might increase by mutation.	

---

maf

Description:	Marker allele frequencies in the first historical generation	
Usage1:	maf = eql;	
	eql	equal
Usage2:	maf = rnd;	
	rnd	random; samples from uniform distribution in each replicate
Usage3:	maf = rnd1;	
	rnd1	random; samples from uniform distribution in the first replicate only (fixed across replicates)
Usage4:	maf = all v1 v2 v3 …;	
	all	all loci would have the same allele frequencies (The number of alleles for all loci should be the same)
	v1 v2 v3 …	are marker allele frequencies (The allele frequencies must sum to one). The number of input values is the number of alleles. Range: 0 – 1
Usage5:	maf = pd v1 v2 v3 …;	
	pd	predefined
	v1 v2 v3 …	are marker allele frequencies (The allele frequencies must sum to one within each marker). The total number of input values should be sum of the number of alleles for all loci. Range: 0 – 1
Type:	Mandatory	
Note:	Allele frequencies will be simulated in the first historical generation. In the subsequent generations, allele frequencies might be changed by drift and mutation.	

---

nqloci

Description:	Number of QTL loci on the chromosome	
Usage:	nqloci = value;	
	value	is the number of QTL loci Range: 0 – 50,000
Type:	Mandatory	

qpos

Description:	QTL positions		
Usage1:	qpos = even option;		
	even	evenly	
Usage2:	qpos = rnd option;		
	rnd	random; samples from uniform distribution in each replicate	
Usage3:	qpos = rnd1 option;		
	rnd1	random; samples from uniform distribution in the first replicate only (fixed across replicates)	
	option	/start v	v is start position
		/end v	v is end position
Usage4:	qpos = pd v1 v2 v3 …;		
	pd	predefined	
	v1 v2 v3 …	are QTL positions	
Type:	Mandatory		

nqa

Description:	Number of QTL alleles in the first historical generation	
Usage1:	nqa = all v;	
	all	all loci would have same number of alleles initially
	v	number of alleles
Usage2:	nqa = rnd v1 v2 v3 …;	
	rnd	random; samples from uniform distribution in each replicate
	v1 v2 v3 …	number of QTL alleles
Usage3:	nqa = rnd1 v1 v2 v3 …;	
	rnd1	random; samples from uniform distribution in the first replicate only (fixed across replicates)
	v1 v2 v3 …	number of QTL alleles
Usage4:	nqa = pd v1 v2 v3 …;	
	pd	predefined
	v1 v2 v3 …	number of QTL alleles Range: 1 – 255 for 8-bit version and 1 – 65535 for 16-bit version
Type:	Mandatory	
Note:	In the subsequent generations, the number of alleles per locus might increase by mutation.	

qaf

Description:	QTL allele frequency in the first historical generation	
Usage1:	qaf = eql;	
	eql	equal

Usage2:	qaf = rnd;	
	rnd	random; samples from uniform distribution in each replicate
Usage3:	qaf = rnd1;	
	rnd1	random; samples from uniform distribution in the first replicate only (fixed across replicates)
Usage4:	qaf = all v1 v2 v3 …;	
	all	all loci would have the same allele frequencies (The number of alleles for all loci should be the same)
	v1 v2 v3 …	are QTL allele frequencies (The allele frequencies must sum to one). The number of input values is the number of alleles. Range: 0 – 1
Usage4:	qaf = pd v1 v2 v3 …;	
	pd	predefined
	v1 v2 v3 …	are QTL allele frequencies (The allele frequencies must sum to one). The total number of input values should be sum of the number of alleles for all loci. Range: 0 – 1
Type:	Mandatory	
Note:	Allele frequencies will be simulated in the first historical generation. In the subsequent generations, allele frequencies might be changed by drift and mutation.	

qae

Description:	QTL allele effect	
Usage1:	qae = pd v1 v2 v3 …;	
	pd	predefined
	v1 v2 v3 …	are percentage of QTL variances (should sum up to one)
	Note:	QTL allelic effects will be sampled from gamma distribution with shape 0.4
Usage2:	qae = rndg v;	
	rndg	Effects are sampled from gamma distribution
	v	gamma shape
Usage3:	qae = rndn;	
	rndn	Effects are sampled from normal distribution
Usage4:	qae = rnd;	
	rnd	Effects are sampled from uniform distribution
Type:	Mandatory	
Note:	QTL allelic effects are simulated in the last historical generation. QTL allelic effects are first sample based on the specified distribution (i.e., gamma, normal or uniform distribution) and then are scaled such that the sum of QTL variances in the last historical generation equals the input QTL variance.	

Scaling is done as follows:

1) QTL value at each locus is calculated as sum of alleles 1 and 2
2) For each QTL, mean in the last HP is computed
3) For each animal in the last HP, sum of QTL values is calculated. This is true genetic value due to QTLs. Then observed variance is obtained for the last HP.
4) Scale = sqrt(Input QTL variance) / sqrt(Observed QTL variance)
5) Then QTL allelic effects at each locus are scaled as (allelic effect – QTL mean) × Scale

When relative QTL variances are specified as input values (predefined), allelic effects of each QTL are scaled separately to ensure the right variance for each QTL.

---

male_map_scale

Description:	Scale factor to shrink or expand linkage map for males	
Usage:	male_map_scale = value;	
	value	is the scale factor
		Range: 0 – 5
Type:	Optional	
Default:	1	
Note:	Crossover will be modeled based on the scaled map. If interference has been specified, interference will also be scaled and therefore interference might become different between sexes. If male_map_scale is specified, new map for males will be printed in linkage map output file.	

---

female_map_scale

Description:	Scale factor to shrink or expand linkage map for females	
Usage:	female_map_scale = value;	
	value	is the scale factor
		Range: 0 – 5
Type:	Optional	
Default:	1	
Note:	Crossover will be modeled based on the scaled map. If interference has been specified, interference will also be scaled and therefore interference might become different between sexes. If female_map_scale is specified, new map for females will be printed in linkage map output file.	

---

cld

Description:	Generate complete linkage disequilibrium in the first historical generation	
Usage1:	cld = m;	generate complete LD between markers
Usage2:	cld = q;	generate complete LD between QTL

Usage3:	cld = m q;	generate complete LD between markers and between QTL, separately (i.e., no LD between markers and QTL)
Usage4:	cld = mq;	generate complete LD between markers, between QTL and between markers and QTL. In this case markers and QTL should have the same number of alleles.

Type: Optional

Note: To generate complete LD for markers or QTL the following requirements should be met: 1) number of alleles should be the same for all loci (i.e., 'all n' for markers or/and QTL) 2) allele frequencies should be the same for all loci (i.e., 'eql' or 'all …').
When complete LD between QTL is considered, there should be few historical generations to create variation between individuals before starting recent populations.

---

select_seg_loci

Description: Select segregating loci in the last historical generation. Only selected loci are used to simulate recent population(s)

Usage: select_seg_loci option;

option	/maft v	v is minor allele frequency threshold. Loci with minor allele frequency larger than or equal to v will be selected
	/nmrk v	v is the number of markers to be selected randomly
	/nqtl v	v is the number of qtl to be selected randomly

Type: Optional

Default: None

Note: When one simulate a large number of historical generations to approach mutation-drift equilibrium, a large proportion of loci are fixed in the last historical generation. Because QMSim does not simulate mutation in recent populations, non-segregating loci are not useful. Therefore, getting rid of the non-segregating loci in the last historical generation can save substantial amount of time and memory.
If no option is specified with this command all segregating loci in the last historical generation will be selected. If 'maft' is specified selection will be first on the minor allele frequency and then based on the number of markers or QTL.
If the number of segregating markers or QTL is smaller than that specified, a warning message will be displayed.

---

r_mpos_g

Description:	Randomize marker positions across genome in each replicate
Usage:	r_mpos_g;
Type:	Optional
Default:	None

---

r1_mpos_g

Description:	Randomize marker positions across genome in the first replicate only (fixed across replicates)
Usage:	r1_mpos_g;
Type:	Optional
Default:	None

---

r_qpos_g

Description:	Randomize QTL positions across genome in each replicate
Usage:	r_qpos_g;
Type:	Optional
Default:	None

---

r1_qpos_g

Description:	Randomize QTL positions across genome in the first replicate only (fixed across replicates)
Usage:	r1_qpos_g;
Type:	Optional
Default:	None

---

rmmg

Description:	Rate of missing marker genotypes		
Usage:	rmmg = value option;		
	value	is the rate of missing marker genotypes Range: 0 – 0.5	
	option	/rndg v	Missing rates are sampled from gamma distribution with shape v
Type:	Optional		
Default:	0		
Note:	If one is interested in having the true genotypes as well, the scenario should be re-run with the same seed with rmmg, rmqg, rmge and rqge commented out. Missing genotypes are simulated in the last step when writing the genotypes in output files. Therefore, inheritance of alleles from one generation to another is not interrupted and all reported statistics are based on full genotypes.		

rmqg

Description:	Rate of missing QTL genotypes
Usage:	The same as for rmmg

rmge

Description: Rate of marker genotyping error

Usage: rmge = value option;

value	is the rate of marker genotyping error Range: 0 – 0.2	
option	/rndg v	Error rates are sampled from gamma distribution with shape v

Type: Optional

Default: 0

Note: Genotyping errors are generated by randomly sampling two alleles from a distribution of equally frequent alleles. It is ensured, however, that the sampled genotypes are different from the correct ones. Sampled alleles come from the existing alleles in the population. If an allele is fixed it is treated as bi-allelic.

If one is interested in having the true genotypes as well, the scenario should be re-run with the same seed and without rmmg, rmqg, rmge and rqge.

Genotyping errors are simulated in the last step when writing the genotypes into the output files. Therefore, all reported statistics are based on true genotypes.

rqge

Description:	Rate of QTL genotyping error
Usage:	The same as for rmge

mmutr

Description: Marker mutation rate in historical population

Usage: mmutr = value option;

value	is mutation arte Range: 0 – 0.01	
option	/recurrent	recurrent mutation

Type: Optional

Default: 0

Note: By default, the mutation model is infinite-allele model.

In recurrent mutation process, no new allele is generated instead the allele state is altered from one to another. **However, with recurrent mutation model, if the number of alleles for a locus is one the locus will be treated as bi-allelic locus. This allows one to simulate SNP when the first historical generation is fixed for all loci (i.e., non-segregating).**

For SNP recurrent mutations are generally very rare and there is no evidence to indicate that mutation contributes significantly to the erosion of LD between SNP (Ardlie et al., 2002).
The number of mutations per individual is sampled from a Poisson distribution with mean u (u = 2 * number of loci * mutation rate) and then each mutation is assigned to a random locus in the genome.
It is important to note that in recent populations no mutation is generated. This is because usually recent populations involve small number of generations for which the effect of mutation might be neglected.

---

qmutr

Description:	QTL mutation rate in historical population		
Usage:	qmutr = value option;		
	value	is mutation arte	
		Range: 0 – 0.01	
	option	/recurrent	recurrent mutation
Type:	Optional		
Default:	0		
Note:	See mmutr		

---

interference

Description:	Interference		
Usage:	interference = value option;		
	value	is interference based on distance in cM	
		Range: 0.01 – 500.0	
	option	/c	complete interference
Type:	Optional		
Default:	no interference		

Note: Crossover is a key factor that gradually breakdowns the allelic associations or LD. Crossover interference can be defined as the nonrandom placement of crossovers along chromosomes in meiosis. The Poisson distribution can describe the distribution of crossovers along a chromosome in meiosis. However, in the present of crossover interference the number of crossover events does not follow Poisson distribution and is low compared with the absent of crossover interference. In order to account for interference we used the following simple algorithm:

- The number of crossover events is sampled from a Poisson distribution with mean equal to the length of chromosomes in Morgan (or with user-specified mean; see mean_crossover)
- Locations of crossovers along chromosomes are assigned at random
- If distance between pair of crossovers (*dis*) is smaller than the specified value

If interference is complete or if URND < $(1-\frac{dis}{value})^2$: delete one of the two crossovers at random

URND is a uniform random deviate.
No chromatid interference is assumed.

---

mean_crossover

Description:	Mean crossover per 1 Morgan	
Usage:	mean_crossover = value;	
	value	is the mean crossover per 1 Morgan Range: 0 – 5
Type:	Optional	
Default:	1	

5- OUTPUT SECTION:

---

begin_output & end_output

Description:	Beginning and end of the output section

Usage:	begin_output;
	end_output;

Type:	Mandatory

---

output_folder

Description:	Output folder	
Usage:	output_folder = string;	
	string	path to the folder where output files will be written
Type:	Optional	
Default:	Default output folder name is r_"parameter file name"	

---

attach_rep

Description:	Append output files over replicates
Usage:	attach_rep;
Type:	Optional

---

linkage_map

Description:	Save linkage map
Usage:	linkage_map;
Type:	Optional

Note:	Marker linkage map file name = lm_mrk_"replicate number".txt QTL linkage map file name = lm_qtl_"replicate number".txt

allele_effect

Description:	Save allele effects
Usage:	allele_effect;
Type:	Optional
Note:	File name = effect_qtl_"replicate number".txt

allele_label

Description:	Save starting allele labels
Usage:	allele_label;
Type:	Optional
Note:	If this command is not specified, allele labels are converted to consecutive numbers starting from 1 at last historical generation. This recoding reduces the genotype file size. If one needs to trace alleles from founders (when starting alleles are unique) then allele_label should be specified.

hp_stat

Description:	Save brief statistics on historical population
Usage:	hp_stat;
Type:	Optional
Note:	File name = hp_stat_"replicate number".txt The output mainly contains statistics on the genome in the last historical generations

monitor_hp_homo

Description:	Save the mean marker and QTL homozygosity of the historical population		
Usage:	monitor_hp_homo option;		
	option	/freq value	
		value	is the output frequency. For example, if value is 100, the mean homozygosity will be reported every 100 generations. Range: 1 – 10,000 Default: 50
Type:	Optional		
Note:	Marker file name = hp_homo_mrk_"replicate number".txt QTL file name = hp_homo_qtl_"replicate number".txt Monitoring homozygosity in HP generations helps to find optimum number of generations required to reach mutation-drift equilibrium. Once the optimum number is found this command can be disabled to speed up the		

simulation of HP.

Running the application

QMSim [parameter filename] -o

If *parameter file name* is not specified, program will prompt the user to enter it. Option *-o* forces the program to overwrite output folder if already exists.

Examples

Example 1: Simulating 10k SNP panel in population of 10 discrete generations and with no historical generations. All loci are in linkage equilibrium in the founders.

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Example 1 - 10k SNP panel";
nrep  = 1;                  //Number of replicates
h2    = 0.2;                //Heritability
qtlh2 = 0.2;                //QTL heritability
phvar = 1.0;                //Phenotypic variance

/*******************************
 **   Historical population   **
 *******************************/
begin_hp;
   hg_size = 420 [0];       //Size of the historical generations
   nmlhg   = 20;            //Number of males in the last generation
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =  20, pop = "hp"];
      female [n = 400, pop = "hp"];
   end_founder;
   ls  = 2;                 //Litter size
   pmp = 0.5 /fix;          //Proportion of male progeny
   ng  = 10;                //Number of generations
   begin_popoutput;
        data;
        stat;
        genotype /gen 8 9 10;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;         //Chromosome length
      nmloci = 333;         //Number of markers
      mpos   = rnd;         //Marker positions
      nma    = all 2;       //Number of marker alleles
      maf    = eql;         //Marker allele frequencies
      nqloci = 25;          //Number of QTL
      qpos   = rnd;         //QTL positions
      nqa    = rnd 2 3 4;      //Number of QTL alleles
      qaf    = eql;            //QTL allele frequencies
      qae    = rndg 0.4;       //QTL allele effects
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   linkage_map;
end_output;
```

Example 2: Simulating 10k SNP panel in population of 10 overlapping generations with 200 historical generations. LD decay in the founders (generation 0) and the last generation will be reported.

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Example 2 - 10k SNP panel";
nrep  = 1;                    //Number of replicates
h2    = 0.2;                  //Heritability
qtlh2 = 0.2;                  //QTL heritability
phvar = 1.0;                  //Phenotypic variance

/*******************************
 **   Historical population   **
 *******************************/
begin_hp;
   hg_size = 420 [0]          //Size of the historical generations
             420 [200];
   nmlhg   = 20;              //Number of males in the last generation
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =  20, pop = "hp"];
      female [n = 400, pop = "hp"];
   end_founder;
   ls  = 1 2 [0.05];          //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 10;                  //Number of generations
   md  = rnd;                 //Mating design
   sr  = 0.4;                 //Replacement ratio for sires
   dr  = 0.2;                 //Replacement ratio for dams
   sd  = rnd;                 //Selection design
   cd  = age;                 //Culling design
   begin_popoutput;
        ld /maft 0.1 /gen 0 10;
        data;
        genotype /snp_code /gen 10;
        allele_freq /gen 10;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;           //Chromosome length
      nmloci = 333;           //Number of markers
      mpos   = rnd;           //Marker positions
      nma    = all 2;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 25;            //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = rnd 2 3 4;     //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndg 0.4;      //QTL allele effects
   end_chr;
   mmutr     = 2.5e-5 /recurrent; //Marker mutation rate
   qmutr     = 2.5e-5;            //QTL mutation rate
   interference = 25;
   r_mpos_g;                  //Randomize marker positions across genome
   r_qpos_g;                  //Randomize QTL positions across genome
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   linkage_map;
   allele_effect;
end_output;
```

Example 3: Simulating two lines selected by divergent phenotypic selection for 20 overlapping generations.

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Example 3 - Creating two divergent lines - 5k SNP panel";
nrep  = 1;                    //Number of replicates
h2    = 0.2;                  //Heritability
qtlh2 = 0.2;                  //QTL heritability
phvar = 1.0;                  //Phenotypic variance

/*******************************
 **   Historical population   **
 *******************************/
begin_hp;
   hg_size = 840 [0]          //Size of the historical generations
             840 [200];
   nmlhg   = 40;              //Number of males in the last generation
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "Line 1";
   begin_founder;
      male   [n =  20, pop = "hp"];
      female [n = 400, pop = "hp"];
   end_founder;
   ls  = 1 2 [0.05];          //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 20;                  //Number of generations
   md  = maxf;                //Mating design
   sr  = 0.4;                 //Replacement ratio for sires
   dr  = 0.2;                 //Replacement ratio for dams
   sd  = phen /h;             //Selection design
   cd  = age;                 //Culling design
   begin_popoutput;
        ld /maft 0.1 /gen 0;
        data;
        genotype /snp_code /gen 10;
        allele_freq /gen 10;
   end_popoutput;
end_pop;

begin_pop = "Line 2";
   begin_founder;
      male   [n =  20, pop = "hp"] /not_founder_yet;
      female [n = 400, pop = "hp"] /not_founder_yet;
   end_founder;
   ls  = 1 2 [0.05];          //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 20;                  //Number of generations
   md  = maxf;                //Mating design
   sr  = 0.4;                  //Replacement ratio for sires
   dr  = 0.2;                  //Replacement ratio for dams
   sd  = phen /l;              //Selection design
   cd  = age;                  //Culling design
   begin_popoutput;
        data;
        genotype /snp_code /gen 10;
        allele_freq /gen 10;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;            //Chromosome length
      nmloci = 166;            //Number of markers
      mpos   = rnd;            //Marker positions
      nma    = all 2;          //Number of marker alleles
      maf    = eql;            //Marker allele frequencies
      nqloci = 25;             //Number of QTL
      qpos   = rnd;            //QTL positions
      nqa    = rnd 2 3 4;      //Number of QTL alleles
      qaf    = eql;            //QTL allele frequencies
      qae    = rndg 0.4;       //QTL allele effects
   end_chr;
   mmutr     = 2.5e-5 /recurrent; //Marker mutation rate
   qmutr     = 2.5e-5;            //QTL mutation rate
   interference = 25;
   r_mpos_g;                   //Randomize marker positions across genome
   r_qpos_g;                   //Randomize QTL positions across genome
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   linkage_map;
   allele_effect;
end_output;
```

Example 4: Considering different genome lengths for males and females. In this example female genome is 25% longer than male genome.

```
/******************************
 **     Global parameters     **
 ******************************/
title = "Example 4 - Different genome lengths for males and females - 10k
SNP panel";
nrep  = 1;                  //Number of replicates
h2    = 0.2;                //Heritability
qtlh2 = 0.2;                //QTL heritability
phvar = 1.0;                //Phenotypic variance

/******************************
 **   Historical population   **
 ******************************/
begin_hp;
   hg_size = 420 [0]        //Size of the historical generations
             420 [200];
   nmlhg   = 20;            //Number of males in the last generation
end_hp;

/******************************
 **        Populations        **
 ******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =  20, pop = "hp"];
      female [n = 400, pop = "hp"];
   end_founder;
   ls  = 2;                 //Litter size
   pmp = 0.5 /fix;          //Proportion of male progeny
   ng  = 10;                //Number of generations
   begin_popoutput;
        data;
        genotype /snp_code /gen 10;
   end_popoutput;
end_pop;

/******************************
 **          Genome           **
 ******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;            //Chromosome length
      female_map_scale=1.25;  //Scale factor for female map
      nmloci = 333;           //Number of markers
      mpos   = rnd;           //Marker positions
      nma    = all 2;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 25;            //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = rnd 2 3 4;     //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndg 0.4;      //QTL allele effects
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   linkage_map;
   hp_stat;
end_output;
```

Example 5: Simulating hotspots and coldspots.

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Example 5 - Hotspots and coldspots";
nrep  = 1;                    //Number of replicates
h2    = 0.2;                  //Heritability
qtlh2 = 0.2;                  //QTL heritability
phvar = 1.0;                  //Phenotypic variance

/*******************************
 **   Historical population   **
 *******************************/
begin_hp;
   hg_size = 420 [0]          //Size of the historical generations
             420 [200];
   nmlhg   = 20;              //Number of males in the last generation
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =  20, pop = "hp"];
      female [n = 400, pop = "hp"];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 10;                  //Number of generations
   begin_popoutput;
        data;
        genotype;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 1;                         //Chromosome 1
      chrlen = 150;                       //Chromosome length
      nmloci = 12;                        //Number of markers
      mpos   = pd 0.0001 .1 .2 .3         //Marker positions
                  100 101 102 103
                  135 140 145 150;
      nma    = all 2;                     //Number of marker alleles
      maf    = rnd;                       //Marker allele frequencies
      nqloci = 12;                        //Number of QTL
      qpos   = pd .05 .15 .25 .35         //QTL positions
                  100.5 101.5 102.5 103.5
                  132   137   142   147;
      nqa    = all 4;                     //Number of QTL alleles
      qaf    = rnd;                           //QTL allele frequencies
      qae    = rndn;                          //QTL allele effects
   end_chr;

   begin_chr = 1;                             //Chromosome 2
      chrlen = 100;                           //Chromosome length
      nmloci = 12;                            //Number of markers
      mpos   = pd 10.00 10.01 10.02 10.03 10.04 10.05  //Marker positions
                  60.05 60.06 60.07 60.08 60.09 60.10;
      nma    = all 2;                         //Number of marker alleles
      maf    = rnd;                           //Marker allele frequencies
      nqloci = 2;                             //Number of QTL
      qpos   = pd 10.025 60.075;              //QTL positions
      nqa    = all 4;                         //Number of QTL alleles
      qaf    = rnd;                           //QTL allele frequencies
      qae    = rndn;                          //QTL allele effects
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   linkage_map;
end_output;
```

Example 6: Simulating a historical bottleneck.

```
/******************************
 **     Global parameters     **
 ******************************/
title = "Example 6 - Historical bottleneck";
nrep  = 1;                  //Number of replicates
h2    = 0.2;                //Heritability
qtlh2 = 0.2;                //QTL heritability
phvar = 1.0;                //Phenotypic variance

/******************************
 **   Historical population   **
 ******************************/
begin_hp;
   hg_size = 1000 [0]          //Size of the historical generations
              200 [70]
              200 [80]
              420 [100];
   nmlhg   = 20;               //Number of males in the last generation
end_hp;

/******************************
 **        Populations        **
 ******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =  20, pop = "hp"];
      female [n = 400, pop = "hp"];
   end_founder;
   ls  = 2;                 //Litter size
   pmp = 0.5 /fix;          //Proportion of male progeny
   ng  = 0;                 //Number of generations
   begin_popoutput;
        data;
        genotype /gen 0;
        allele_freq /gen 0;
   end_popoutput;
end_pop;

/******************************
 **          Genome           **
 ******************************/
begin_genome;
   begin_chr = 5;
      chrlen = 150;         //Chromosome length
      nmloci = 100;         //Number of markers
      mpos   = rnd;         //Marker positions
      nma    = all 2;       //Number of marker alleles
      maf    = rnd;         //Marker allele frequencies
      nqloci = 25;          //Number of QTL
      qpos   = rnd;         //QTL positions
      nqa    = rnd 2 3 4;   //Number of QTL alleles
      qaf    = rnd;         //QTL allele frequencies
      qae    = rndg 0.4;      //QTL allele effects
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   linkage_map;
   hp_stat;
end_output;
```

Example 7: Simulating population expansion in a recent population.

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Example 7 - population expansion in a recent population";
nrep  = 1;                     //Number of replicates
h2    = 0.2;                   //Heritability
qtlh2 = 0.2;                   //QTL heritability
phvar = 1.0;                   //Phenotypic variance

/*******************************
 **  Historical population    **
 *******************************/
begin_hp;
   hg_size = 420 [0]           //Size of the historical generations
             420 [200];
   nmlhg   = 20;               //Number of males in the last generation
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =  20, pop = "hp"];
      female [n = 400, pop = "hp"];
   end_founder;
   ls  = 2;                    //Litter size
   pmp = 0.5 /fix;             //Proportion of male progeny
   ng  = 15;                   //Number of generations
   md  = rnd;                  //Mating design
   sr  = 0.4 0.0 [1]           //Replacement ratio for sires
         0.4 0.25 [5]
         0.4 0.0 [10];
   dr  = 0.2 0.0 [1]           //Replacement ratio for dams
         0.2 0.25 [5]
         0.2 0.0 [10];
   sd  = rnd;                  //Selection design
   cd  = age;                  //Culling design
   begin_popoutput;
        data;
        stat;
        genotype /snp_code /gen 15;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;            //Chromosome length
      nmloci = 167;            //Number of markers
      mpos   = rnd;           //Marker positions
      nma    = all 2;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 25;            //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = rnd 2 3 4;     //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndg 0.4;      //QTL allele effects
   end_chr;
   mmutr     = 2.5e-5 /recurrent; //Marker mutation rate
   qmutr     = 2.5e-5;            //QTL mutation rate
end_genome;

/******************************
 **       Output options      **
 ******************************/
begin_output;
   linkage_map;
end_output;
```

Example 8: Simulating a pure polygenic trait (no markers and no QTLs are simulated).
As QMSim simulates a genome, the genome section in the parameter file is mandatory.
However, for simulating a pure polygenic trait, user should do the following:

1- Set qtlh2 to zero
2- No historical generations are needed, so set nhg to zero
3- Define one chromosome
4- Set the number of markers and QTL to zero
5- Set other genome parameters to valid arbitrary values

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Example 8 - Simulating a pure polygenic trait";
nrep  = 1;                    //Number of replicates
h2    = 0.2;                  //Heritability
qtlh2 = 0.0;                  //QTL heritability
phvar = 1.0;                  //Phenotypic variance

/*******************************
 **   Historical population   **
 *******************************/
begin_hp;
   hg_size = 420 [0];         //Size of the historical generations
   nmlhg   = 20;              //Number of males in the last generation
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =  20, pop = "hp"];
      female [n = 400, pop = "hp"];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 10;                  //Number of generations
   begin_popoutput;
        data;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 1;
      chrlen = 10;            //Chromosome length
      nmloci = 0;             //Number of markers
      mpos   = rnd;           //Marker positions
      nma    = all 2;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 0;             //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = all 2;         //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndg 0.4;      //QTL allele effects
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
end_output;
```

Example 9: Simulating selection and culling designs. In the following example breeding individuals with inferior EBVs are replaced with progeny with superior phenotypes.

```
/******************************
 **     Global parameters     **
 ******************************/
title = "Example 9 - Replacing breeding individuals with inferior EBVs
with progeny with superior phenotypes - 5k SNP panel";
nrep  = 1;                  //Number of replicates
h2    = 0.3;                //Heritability
qtlh2 = 0.1;                //QTL heritability
phvar = 1.0;                //Phenotypic variance

/******************************
 **   Historical population   **
 ******************************/
begin_hp;
   hg_size = 2550 [0]          //Size of the historical generations
             2550 [200];
   nmlhg   = 50;               //Number of males in the last generation
end_hp;

/******************************
 **        Populations        **
 ******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =   50, pop = "hp"];
      female [n = 2500, pop = "hp"];
   end_founder;
   ls  = 1 2 [0.2];            //Litter size
   pmp = 0.5;                  //Proportion of male progeny
   ng  = 10;                   //Number of generations
   md  = rnd;                  //Mating design
   sr  = 0.4;                  //Replacement ratio for sires
   dr  = 0.2;                  //Replacement ratio for dams
   sd  = phen /h;              //Selection design
   cd  = ebv /l;               //Culling design
   ebv_est = blup /true_av;
   begin_popoutput;
        ld /maft 0.1 /gen 0;
        data;
        stat;
        genotype /snp_code /gen 10;
   end_popoutput;
end_pop;

/******************************
 **          Genome           **
 ******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;            //Chromosome length
      nmloci = 167;            //Number of markers
      mpos   = rnd;           //Marker positions
      nma    = all 2;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 25;            //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = rnd 2 3 4;     //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndg 0.4;      //QTL allele effects
   end_chr;
   mmutr     = 2.5e-5 /recurrent; //Marker mutation rate
   qmutr     = 2.5e-5;            //QTL mutation rate
   r_mpos_g;                  // Randomize marker positions across genome
   r_qpos_g;                  // Randomize marker positions across genome
end_genome;

/********************************
 **       Output options      **
 ********************************/
begin_output;
   linkage_map;
end_output;
```

Example 10: Simulating a sex limited trait.

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Example 10 - Simulating a sex limited trait - 5k SNP panel";
nrep  = 1;                   //Number of replicates
h2    = 0.3;                 //Heritability
qtlh2 = 0.3;                 //QTL heritability
phvar = 1.0;                 //Phenotypic variance
no_male_rec;                 //Males have no record

/*******************************
 **   Historical population   **
 *******************************/
begin_hp;
   hg_size = 2550 [0]          //Size of the historical generations
             2550 [200];
   nmlhg   = 50;               //Number of males in the last generation
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "p1";
   begin_founder;
      male   [n =   50, pop = "hp"];
      female [n = 2500, pop = "hp"];
   end_founder;
   ls  = 1 2 [0.2];           //Litter size
   pmp = 0.5;                 //Proportion of male progeny
   ng  = 10;                  //Number of generations
   md  = rnd;                 //Mating design
   sr  = 0.4;                 //Replacement ratio for sires
   dr  = 0.2;                 //Replacement ratio for dams
   sd  = phen /h;             //Selection design
   cd  = ebv /l;              //Culling design
   ebv_est = blup /true_av;
   begin_popoutput;
        ld /maft 0.1 /gen 0;
        data;
        stat;
        genotype /snp_code /gen 10;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;           //Chromosome length
      nmloci = 167;           //Number of markers
      mpos   = rnd;           //Marker positions
      nma    = all 2;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 25;            //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = rnd 2 3 4;     //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndg 0.4;      //QTL allele effects
   end_chr;
   mmutr     = 2.5e-5 /recurrent; //Marker mutation rate
   qmutr     = 2.5e-5;            //QTL mutation rate
   r_mpos_g;                  // Randomize marker positions across genome
   r_qpos_g;                  // Randomize marker positions across genome
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   linkage_map;
end_output;
```

Example 11: F2 and backcross designs.

```
/******************************
 **     Global parameters     **
 ******************************/
title = "Example 11 - F2 and backcross designs";
nrep  = 1;                    //Number of replicates
h2    = 0.2;                  //Heritability
qtlh2 = 0.05;                 //QTL heritability
phvar = 1.0;                  //Phenotypic variance

/******************************
 **   Historical population   **
 ******************************/
begin_hp;
   hg_size = 10000 [0]        //Size of the historical generations
             10000 [100];
end_hp;

/******************************
 **        Populations        **
 ******************************/
begin_pop = "line1";
   begin_founder;
      male   [n = 20,  pop = "hp", select = tbv /h];
      female [n = 400, pop = "hp", select = tbv /h];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 20;                  //Number of generations
   md  = p_assort /tbv;       //Mating design
   sd  = tbv /h;              //Selection design
end_pop;

begin_pop = "line2";
   begin_founder;
      male   [n = 20,  pop = "hp", select = tbv /l];
      female [n = 400, pop = "hp", select = tbv /l];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 20;                  //Number of generations
   md  = p_assort /tbv;       //Mating design
   sd  = tbv /l;              //Selection design
end_pop;

//Cross between line1 and line 2 to generate F2
begin_pop = "cross";
   begin_founder;
      male   [n = 20, pop = "line1", gen = 20];
      female [n = 400, pop = "line2", gen = 20];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 2;                   //Number of generations
   md  = rnd;                 //Mating design
   sr  = 1;                   //Replacement ratio for sires
   dr  = 1;                   //Replacement ratio for dams
   sd  = rnd;                 //Selection design
   cd  = rnd;                 //Culling design
   begin_popoutput;
        data;
        genotype /gen 1 2;
        stat;
   end_popoutput;
end_pop;

//Backcrossing F1 to line1
begin_pop = "bckcross";
   begin_founder;
      male   [n = 20, pop = "line1", gen = 20];
      female [n = 400, pop = "cross", gen = 1];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 1;                   //Number of generations
   md  = rnd;                 //Mating design
   begin_popoutput;
        data;
        genotype /gen 1;
        stat;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 1;
      chrlen = 100;           //Chromosome length
      nmloci = 8;             //Number of markers
      mpos   = pd 30 35 39 40.001 60.001 61 65 70;  //Marker positions
      nma    = all 2;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 2;             //Number of QTL
      qpos   = pd 40 60;      //QTL positions
      nqa    = all 2;         //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = pd 0.1 0.9;    //QTL allele effects
      cld    = mq;      //Complete LD in the first historical generation
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
end_output;
```

Example 12: Migration.

```
/******************************
 **     Global parameters     **
 ******************************/
title = "Example 12 - Migration";
nrep  = 1;                    //Number of replicates
h2    = 0.3;                  //Heritability
qtlh2 = 0.1;                  //QTL heritability
phvar = 1.0;                  //Phenotypic variance

/******************************
 **   Historical population   **
 ******************************/
begin_hp;
   hg_size = 1000 [0]         //Size of the historical generations
             1000 [100];
end_hp;

/******************************
 **        Populations        **
 ******************************/
begin_pop = "line1";
   begin_founder;
      male   [n = 10,  pop = "hp", select = tbv /h];
      female [n = 100, pop = "hp", select = tbv /h];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 10;                  //Number of generations
   md  = rnd;                 //Mating design
   sd  = phen /h;             //Selection design
   begin_popoutput;
        data;
        stat;
        allele_freq /gen 10;
   end_popoutput;
end_pop;

begin_pop = "line2";
   begin_founder;
      male   [n = 10,  pop = "hp", select = tbv /l];
      female [n = 100, pop = "hp", select = tbv /l];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 10;                  //Number of generations
   md  = rnd;                 //Mating design
   sd  = phen /l;             //Selection design
   begin_popoutput;
        data;
        stat;
        allele_freq /gen 10;
        end_popoutput;
end_pop;
```

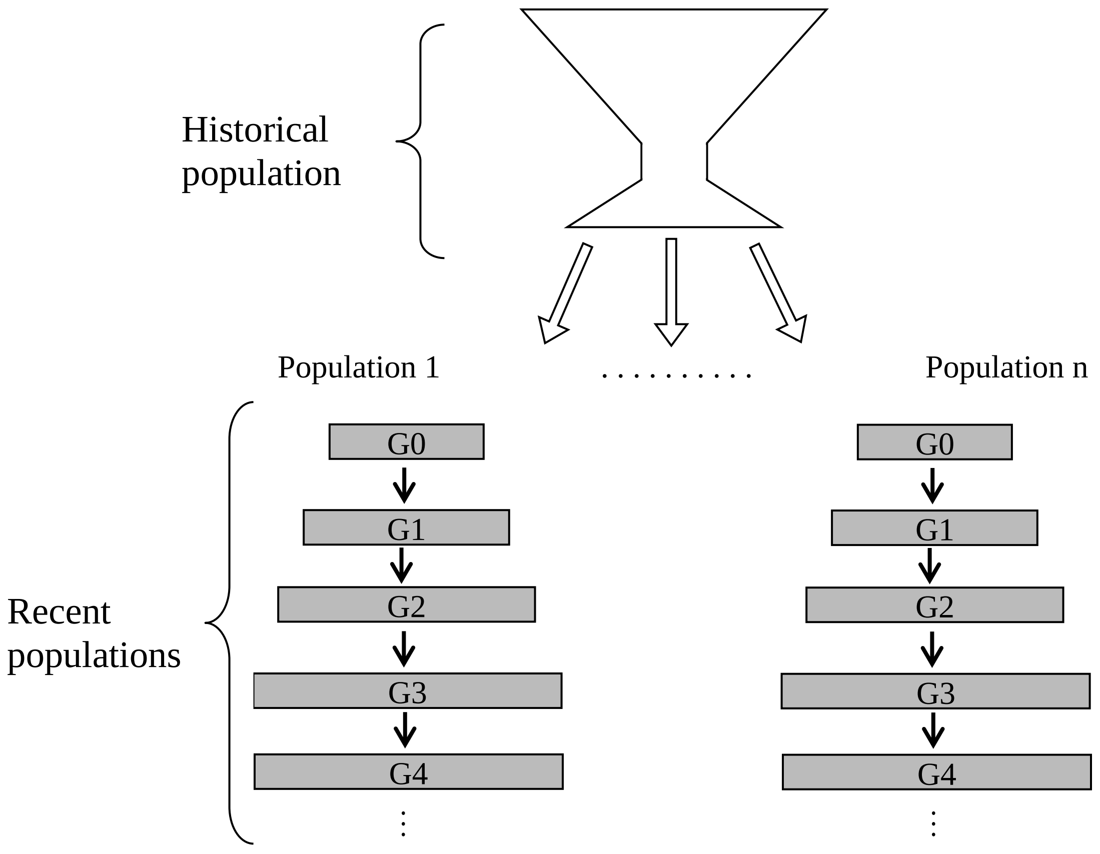
```
//2 males and 10 females from line2 migrate to line1
begin_pop = "line1_c";
   begin_founder;
      male   [n =  8, pop = "line1", gen = 10];
      male   [n =  2, pop = "line2", gen = 10];
      female [n = 90, pop = "line1", gen = 10];
      female [n = 10, pop = "line2", gen = 10];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;            //Proportion of male progeny
   ng  = 5;                   //Number of generations
   md  = rnd;                 //Mating design
   sd  = rnd;                 //Selection design
   begin_popoutput;
        data;
        stat;
        allele_freq /gen 5;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 10;
      chrlen = 100;           //Chromosome length
      nmloci = 101;           //Number of markers
      mpos   = even;          //Marker positions
      nma    = all 4;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 25;            //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = rnd 2 3 4;     //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndg 0.4;      //QTL allele effects
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
end_output;
```

Note that parameters for all examples given above have been chosen arbitrarily and may not represent a realistic situation. Examples are only provided to help users to get a quick start with QMSim.

Example of simulation diagram:



Decay of linkage disequilibrium in a simulated data set using QMSim:

When an infinite population undergoes random mating and random selection, the amount of linkage disequilibrium between two adjacent loci is expected to decay exponentially over generations at rate equal to recombination rate. We have investigated the decay of LD in a simulated population obtained by QMSim. The population consisted of 500 sires and 500 dams in each generation which were mated and selected at random for 100 discrete generations. Different marker densities were considered. In the first generation, markers were forced to be in complete LD with each other (see next page for the parameter file). The following graph (Figure 1) shows the decay of LD between adjacent markers for different recombination rates observed in the simulated data set.

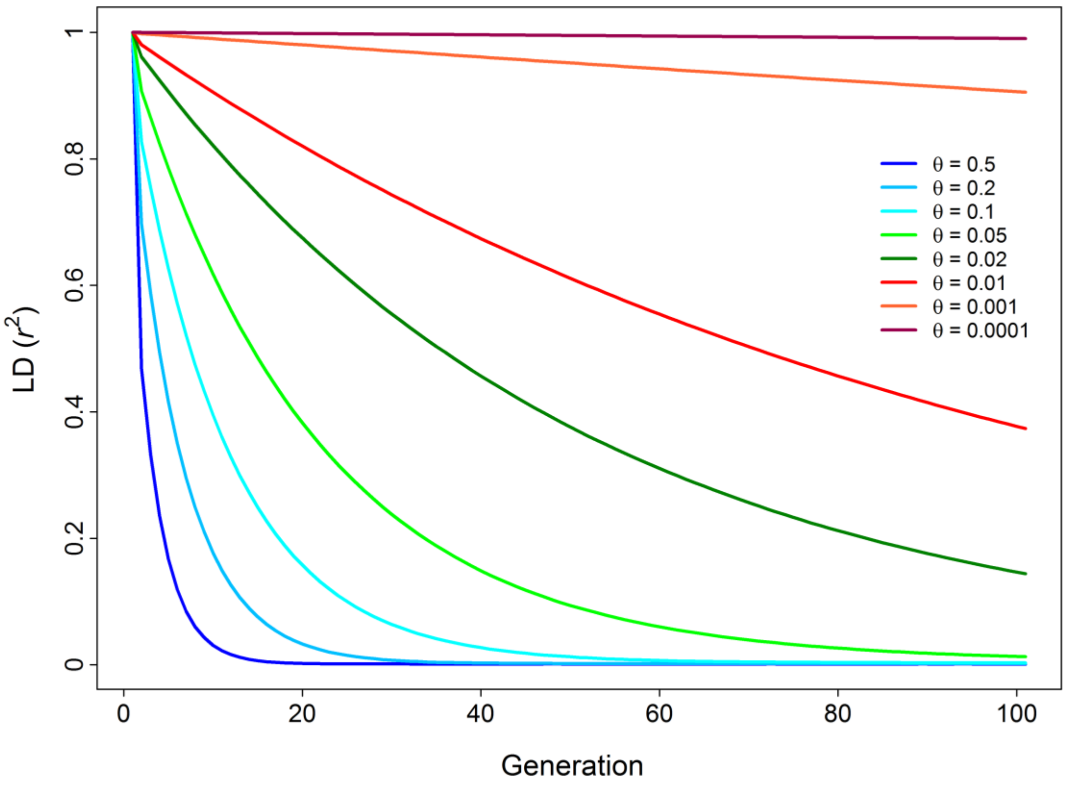


Figure 1. Observed decay of linkage disequilibrium (LD) between adjacent marker pairs for different recombination rates (θ) in a simulated data set using QMSim.

Parameter file for assessing decay of LD using QMSim

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Decay of linkage disequilibrium - marker interval is 1 cM";
nrep  = 1000;                 //Number of replicates
h2    = 0.2;                  //Heritability
qtlh2 = 0.2;                  //QTL heritability
phvar = 1.0;                  //Phenotypic variance
skip_inbreeding;

/*******************************
 **   Historical population   **
 *******************************/
begin_hp;
   hg_size = 1000[0];         //Size of the historical generations
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "Pop1";
   begin_founder;
      male   [n = 500, pop = "hp"];
      female [n = 500, pop = "hp"];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;
   ng  = 100;                 //Number of generations
   md  = rnd;                 //Mating design
   sr  = 1;                   //Replacement ratio for sires
   dr  = 1;                   //Replacement ratio for dams
   sd  = rnd;                 //Selection design
   begin_popoutput;
        ld /maft 0.1;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 1;
      chrlen = 100;           //Chromosome length
      nmloci = 101;           //Number of markers
      mpos   = even;          //Marker positions
      nma    = all 2;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 50;            //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = all 2;         //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndn;          //QTL allele effects
      cld    = m;     //Complete LD in the first historical generation
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   linkage_map;
end_output;
```

Observed inbreeding vs. expected inbreeding:

Inbreeding is an important parameter in the population and evolutionary genetics. The mean population inbreeding coefficient at generation *t* can be predicted as $F_t = 1-(1-\frac{1}{2Ne})^t$, where *Ne* is the initial effective population size. To investigate whether QMSim generates inbreeding properly, observed average inbreeding coefficients for different effective population sizes (20, 50, 100, 200 and 500) against generations were plotted. The observed average inbreeding coefficients were well in agreement with the expected ones. Results are shown in Figure 2. For details of the simulated population structure, see parameter file in below.

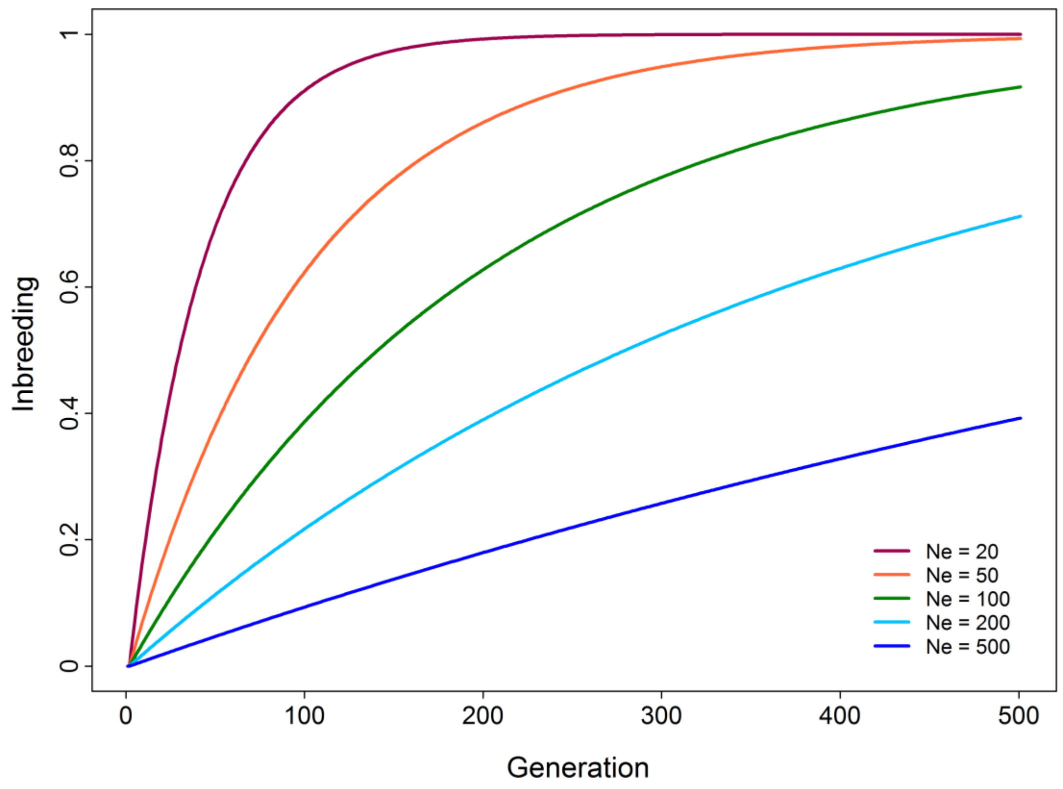


Figure 2. Observed average inbreeding coefficients over generations for different effective population sizes (*Ne*).

Parameter file for assessing inbreeding:

```
/*******************************
 **     Global parameters     **
 ******************************/
title = "Inbreeding and effective population size";
nrep  = 1000;               //Number of replicates
h2    = 0.2;                //Heritability
qtlh2 = 0.2;                //QTL heritability
phvar = 1.0;                //Phenotypic variance
skip_inbreeding;

/*******************************
 **   Historical population   **
 ******************************/
begin_hp;
   hg_size = 100[0];        //Size of the historical generations
end_hp;

/*******************************
 **        Populations        **
 ******************************/
begin_pop = "Pop1";
   begin_founder;
      male   [n = 50, pop = "hp"];
      female [n = 50, pop = "hp"];
   end_founder;
   ls  = 2;                 //Litter size
   pmp = 0.5 /fix;
   ng  = 500;               //Number of generations
   md  = rnd_ug;            //Mating design
   sr  = 1;                 //Replacement ratio for sires
   dr  = 1;                 //Replacement ratio for dams
   sd  = rnd;               //Selection design
   begin_popoutput;
        stat;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 ******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;         //Chromosome length
      nmloci = 166;         //Number of markers
      mpos   = even;        //Marker positions
      nma    = unique;      //Number of marker alleles
      maf    = eql;         //Marker allele frequencies
      nqloci = 20;          //Number of QTL
      qpos   = rnd;         //QTL positions
      nqa    = all 2;       //Number of QTL alleles
      qaf    = eql;         //QTL allele frequencies
      qae    = rndn;        //QTL allele effects
   end_chr;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
end_output;
```

Mutation-drift equilibrium:

Genetic variability is generated by mutation but it is lost randomly over generations through genetic drift. The amount of new variation depends on the mutation rate and loss of variation due to fixation of alleles depends on effective population size.

Let’s assume that alleles are neutral (no selection) and that offspring s in each generation are produced from random union of gametes from N males and M females. Over generations, mutation and genetic drift act in opposite directions. However after certain number of generations the population reaches mutation-drift equilibrium where the population maintains a certain amount of variation. At equilibrium $F_t = F_{t-1} = F_{t-2} = F$. Under infinite-allele mutation model F is approximately $\frac{1}{1+4Neu}$ and proportion of heterozygous loci is 1-F or $H \approx \frac{4Neu}{1+4Neu}$ (Crow and Kimura, 1970), where u is the mutation rate. Under infinite-allele model, mutation creates new unique allele that never existed in the population.

For finite-allele model, mutation does not create novel allele but rather modifies an allele to another allele within the allelic space with the same probability. Recurrent mutation can be reversible. At equilibrium F is approximately $\frac{1+\frac{4Neu}{k-1}}{1+\frac{4Neuk}{k-1}}$, where k is the number of possible alleles.

In the following we assessed the distribution of allele frequencies at mutation-drift equilibrium for neutral alleles under assumption of random union of gametes from N males and M females. Here we have simulated bi-allelic markers mimicking SNP markers, which follow a recurrent mutation model. When $4Neu$ is smaller than 1, close to 1 or larger than 1, at mutation-drift equilibrium a U-shape, uniform or normal distribution of allele frequencies is expected, respectively (Wright, 1931). Six different scenarios were considered. Parameters for these scenarios are shown in Table 1.

Table 1. Parameters for different scenarios of mutation-drift equilibrium.

Parameters	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Ne	100	1,000	1,000	500	1,000	2,000
u	2.5e-4	2.5e-5	2.5e-5	2.5e-4	2.5e-4	2.5e-4
4*Neu*	0.1	0.1	0.1	0.5	1	2
No. of SNP	10,000	10,000	10,000	10,000	10,000	10,000
Initial allele freq.	fixed	fixed	0.5	fixed	fixed	fixed
No. of gen.	1,000	1,000	1,000	1,000	1,000	1,000
	5,000	5,000	2,000	5,000	5,000	5,000
	10,000	10,000	5,000	10,000	10,000	10,000
	20,000	20,000	10,000	20,000	20,000	20,000
		40,000				
		60,000				
No. of replicates	100	100	100	100	100	100

For all scenarios equal numbers of males and females were considered, matings were based on random union of gametes and mutation and counter-mutation rates are assumed to be equal. Results for each scenario are shown in Figures 3 to 7.

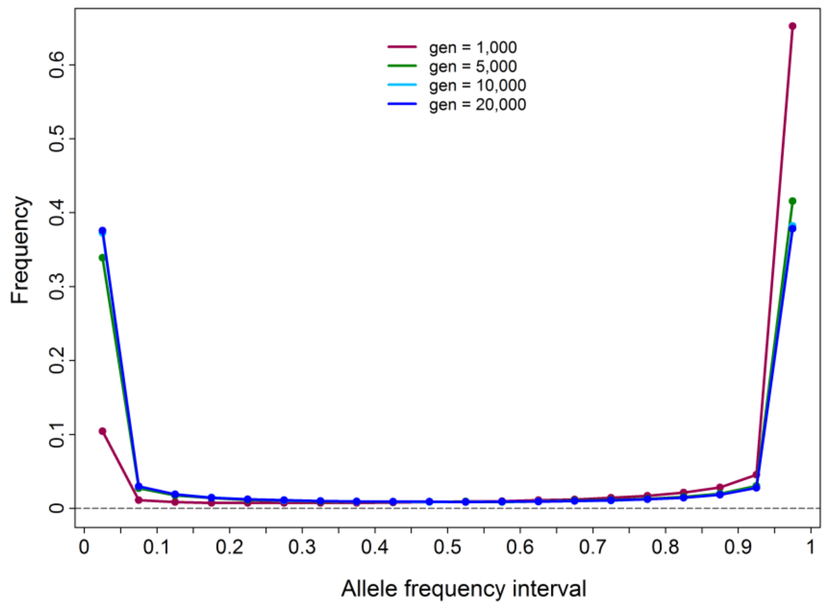


Figure 3. Distribution of allele frequencies for scenario 1.

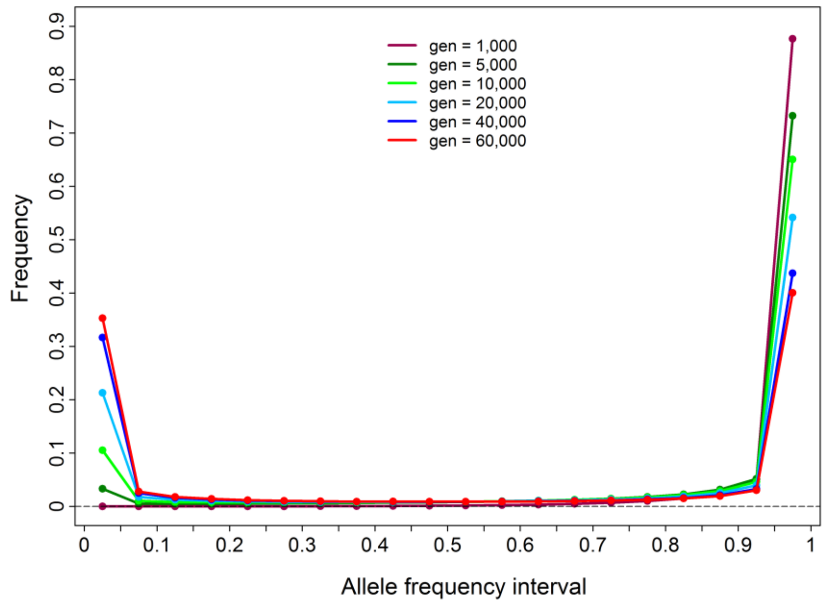


Figure 4. Distribution of allele frequencies for scenario 2.

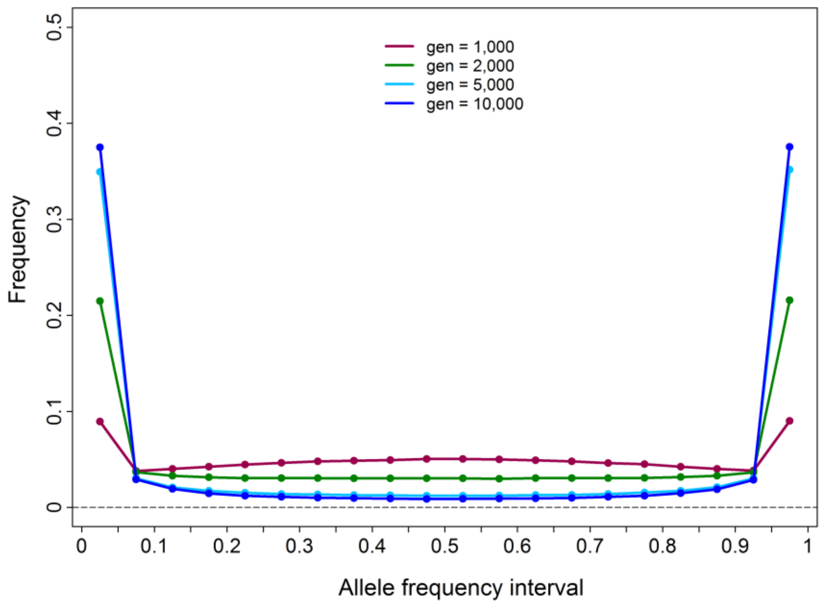


Figure 5. Distribution of allele frequencies for scenario 3.

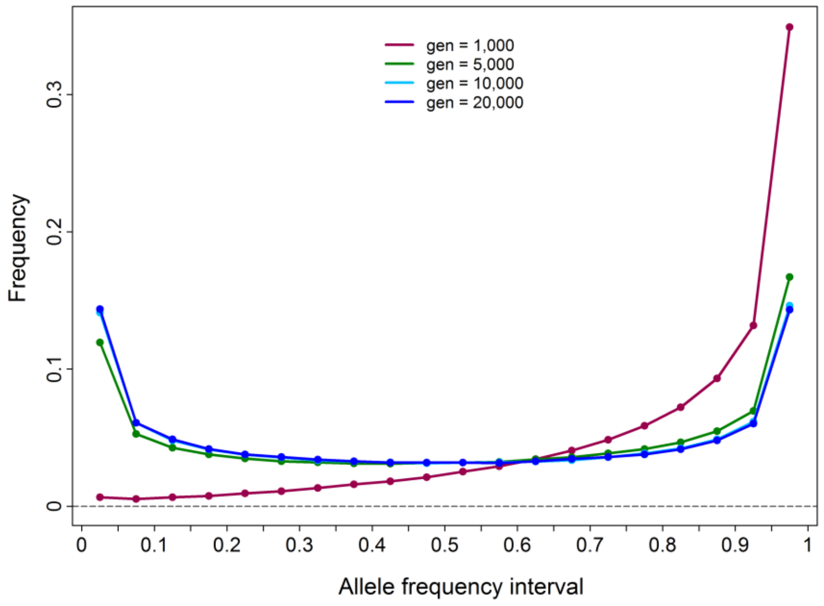


Figure 6. Distribution of allele frequencies for scenario 4.

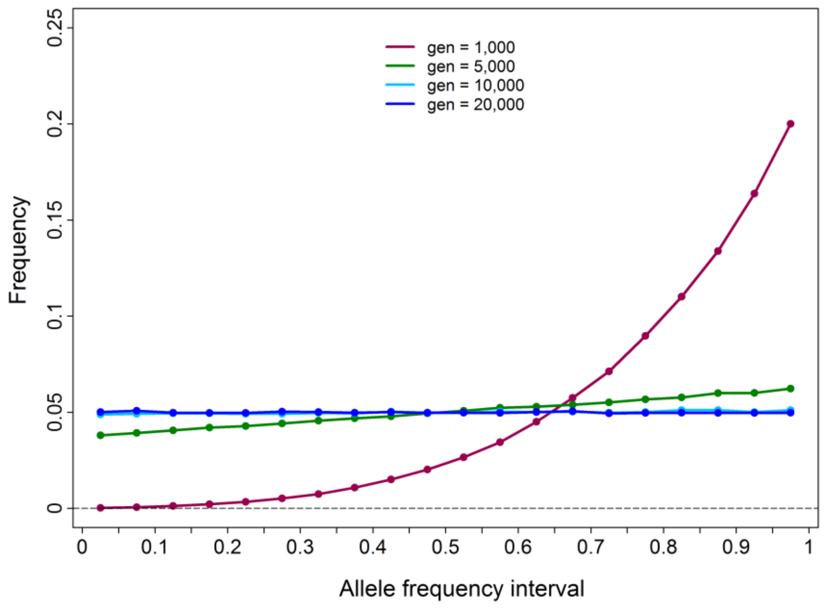


Figure 7. Distribution of allele frequencies for scenario 5.

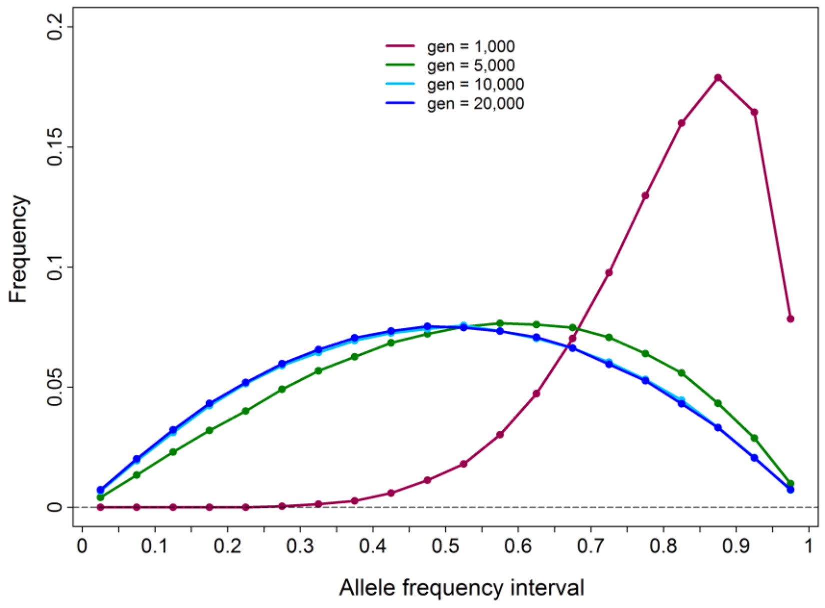


Figure 8. Distribution of allele frequencies for scenario 6.

From graphs 3 and 4 (scenarios 1 and 2) it can be seen that when 4*Neu* is constant but *Ne* differs, allele frequencies approach mutation-drift equilibrium slower with larger *Ne*.

Comparing graph 4 to graph 5 (scenario 2 to scenario 3) shows that when allele frequencies are equal (0.5) in the first generation instead of fixed (1.0), they approach their steady-state frequencies much quicker. Graphs 6, 7 and 8 show the distribution of allele frequencies for different values of 4*Neu*.

Parameter file for assessing mutation-drift equilibrium (scenario 4):

```
/*******************************
 **     Global parameters     **
 *******************************/
title = "Mutation-drift equilibrium";
nrep  = 100;                  //Number of replicates
h2    = 0.2;                  //Heritability
qtlh2 = 0.2;                  //QTL heritability
phvar = 1.0;                  //Phenotypic variance

/*******************************
 **   Historical population   **
 *******************************/
begin_hp;
   hg_size = 500[0]           //Size of the historical generations
             500[1000];
end_hp;

/*******************************
 **        Populations        **
 *******************************/
begin_pop = "Pop1";
   begin_founder;
      male   [n = 250, pop = "hp"];
      female [n = 250, pop = "hp"];
   end_founder;
   ls  = 2;                   //Litter size
   pmp = 0.5 /fix;
   ng  = 0;                   //Number of generations
   md  = rnd;                 //Mating design
   sr  = 1;                   //Replacement ratio for sires
   dr  = 1;                   //Replacement ratio for dams
   sd  = rnd;                 //Selection design
   begin_popoutput;
        data;
   end_popoutput;
end_pop;

/*******************************
 **          Genome           **
 *******************************/
begin_genome;
   begin_chr = 30;
      chrlen = 100;           //Chromosome length
      nmloci = 333;           //Number of markers
      mpos   = even;          //Marker positions
      nma    = all 1;         //Number of marker alleles
      maf    = eql;           //Marker allele frequencies
      nqloci = 25;            //Number of QTL
      qpos   = rnd;           //QTL positions
      nqa    = all 2;         //Number of QTL alleles
      qaf    = eql;           //QTL allele frequencies
      qae    = rndn;          //QTL allele effects
   end_chr;
   mmutr=2.5e-4/recurrent;
   qmutr=2.5e-4;
end_genome;

/*******************************
 **       Output options      **
 *******************************/
begin_output;
   monitor_hp_homo /freq 100;
end_output;
```

Acknowledgments:

The authors gratefully acknowledge financial support from L'Alliance Boviteq Inc. (SEMEX Alliance, Canada), Natural Sciences and Engineering Research Council of Canada (Collaborative Research and Development grant), and the Ontario Centre for Agriculture Genomics (Challenge Fund).

References:

Andersson, L. 2001. Genetic dissection of phenotypic diversity in farm animals. Nature Genetics Reviews 2, 130-138.

Ardlie, K. G., L. Kruglyak and M. Seielstad. 2002. Patterns of linkage disequilibrium in the human genome. Nature Reviews, Genetics 3, 299-309.

Crow, J. F., and M. Kimura, 1970. *An Introduction to Population Genetic Theory*. Harper & Row, New York.

Henderson, C. R. 1975. Best linear unbiased estimation and prediction under a selection model. Biometrics 31, 423-447.

Hoggart, C. J., M. Chadeau-Hyam, T. G. Clark, R. Lampariello, J. C. Whittaker, M. De Iorio and D. J. Balding. 2007. Sequence-level population simulations over large genomic regions. Genetics 177, 1725-1731.

Hudson, R. R. 2002. Generating samples under a Wright-Fisher neutral model. Bioinformatics 18, 337-378.

Li, C. and M. Li. 2007. GWAsimulator: A rapid whole-genome simulation program. Bioinformatics 24, 140-142.

Matsumoto, M. and T. Nishimura. 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Trans. Model. Comput. Simul. 8, 3-30.

Meuwissen, T. H. E., B. J. Hayes and M. E. Goddard. 2001. Prediction of total genetic value using genome-wide dense marker maps. Genetics 157, 1819-1829.

Meuwissen, T. H. E., A. Karlsen, S. Lien, I. Olsaker and M. E. Goddard. 2002. Fine mapping of a quantitative trait locus for twinning rate using combined linkage and linkage disequilibrium mapping. Genetics 161, 373-379.

Pritchard, J. K. and N. A. Rosenberg, 1999. Use of unlinked genetic markers to detect population stratification in association studies. Am. J. Hum. Genet. 65, 220-228.

Schaffner, S. F., C. Foo, S. Gabriel, D. Reich, M. J. Daly and D. Altshuler. 2005. Calibrating a coalescent simulation of human genome sequence variation. Genome Res. 15, 1576-1583.

Sonesson, A. K. and T. H. E. Meuwissen. 2000. Mating schemes for optimum contribution selection with constrained rates of inbreeding. Genetics Selection Evolution 32, 231-248.

Wright, S. 1931. Evolution in Mendelian populations. Genetics 16, 97-159.

License for the Mersenne Twister random number generator

The random number generator incorporated in QMSim is based on a code downloaded from:

http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/MT2002/emt19937ar.html