quantiNEMO 2

release 0.9.0

CONFIDENTIAL

User Manual

June 13, 2017

authors
Samuel Neuenschwander
samuel.neuenschwander@unil.ch

Jérôme Goudet
jerome.goudet@unil.ch

website
http://www.unil.ch/popgen/softwares/quantinemo

CONFIDENTIAL

© 2014 Samuel Neuenschwander

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies. Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the sections entitled Copying and GNU General Public License are included exactly as in the original, and provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one. Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Free Software Foundation.

Contents

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

Chapter 1

Introduction

CONFIDENTIAL

1.1 Scope

quantiNEMO is an individual-based, genetically explicit stochastic simulation program. It was developed to investigate the effects of selection, mutation, recombination, and drift on quantitative traits with varying architectures in structured populations connected by migration and located in a heterogeneous habitat. quantiNEMO is highly flexible at various levels: population, selection, trait(s) architecture, genetic map for QTL and/or markers, environment, demography, mating system, etc.

1.2 Availability

The website http://www.unil.ch/popgen/softwares/quantinemo includes executables for Windows, Linux and Mac, the source code, a detailed user's manual, and also a syntax highlighting definition to edit the settings file with the shareware TextPad (http://www.textpad.com). All downloads are freely available under the terms of the GNU General Public License.

1.3 Technical

quantiNEMO is a console program, and is coded in standard C++ using an object oriented approach. This allows compiling quantiNEMO on any computer platform which supports standard C++ compilation. There is no limit on the number of populations, individuals, genes, etc that quantiNEMO can handle, apart from the available hardware capacities (CPU and memory). quantiNEMO was optimized for high computation efficiency in particular for large simulations on clusters. quantiNEMO is built on the evolutionary and population genetics programming framework NEMO (Guillaume and Rougemont, 2006), with well developed demographic models. The demographic models of NEMO were kept, although several of these functionalities were re-coded, respectively adapted to the new functionalities of quantiNEMO.

CONFIDENTIAL

1.4 License

quantiNEMO is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. quantiNEMO is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with quantiNEMO. If not, see http://www.gnu.org/licenses/.

1.5 Citation

quantiNEMO has been published in the journal *Bioinformatics*. If you are using quantiNEMO for your research please cite it as
Neuenschwander S, Hospital F, Guillaume F, Goudet J, 2008. QuantiNEMO: an individual-based program to simulate quantitative traits with explicit genetic architecture in a dynamic metapopulation. **Bioinformatics** 24, 1552-1553.

1.6 Acknowledgments

We are grateful to Ricardo Kanitz, Sylvain Antoniazza, Yves Rousselle, Patrick Meirmans, Christine Grossen, Claire Mouton, and Olivier Blaser, Alex von Ungern-Sternberg. These persons helped us to improve quantiNEMO by reporting bugs and/or great discussions.

1.7 Main features

quantiNEMO consists in several simulation components which may be easily extended. The simulation components with their corresponding parameters are described in more detail in the rest of this manual.

CONFIDENTIAL

Quantitative traits
: quantiNEMO allows the simulation of one to multiple quantitative traits each having its own specifications. Each trait is defined by one to many loci each with up to 256 alleles. The allelic effects at each locus can be drawn from a normal distribution or can be set explicitly. Mutations are implemented with several models. The trait determinism can be purely additive, or include dominance and/or epistatic interactions among loci. Environmental effects can also be set in different ways.

Neutral markers
: quantiNEMO also allows the simulation of neutral markers, such as microsatellites or SNPs with different mutation models (K Allele, Stepwise). Different types of neutral markers and/or quantitative trait loci can be combined within the same simulation.

Genetic map
: quantiNEMO has an underlying genetic map, which may consist of several chromosomes. This allows an explicit positioning of all types of loci on the map (quantitative trait loci (QTL) and neutral markers).

Metapopulation
: quantiNEMO allows simulating realistic population dynamics. Population sizes may vary in space and in time. The user can choose between several preset migration models (island, 1-D stepping-stone,

2-D stepping-stone), or specify the full migration matrix. The migration pattern can change over time, allowing to investigate scenarios of population fragmentation.

Lifecycle

Each individual undergoes a single life cycle (non-overlapping generations). The life-cycle is fixed (in contrast to NEMO (Guillaume and Rougemont, 2006)) and starts with breeding and reproduction. Several mating systems are available: random mating, selfing or cloning for hermaphrodites; promiscuity, monogamy or polygyny for dÃŤoecious (gonochoric) species. Selection acts by default on the reproductive fitness of individuals, however the current version of quantiNEMO supports also selection at other life cycle stages. After reproduction, juveniles may disperse to other populations, and then population size is possibly regulated. Environmental stochasticity can also be introduced, where populations may go extinct due to an external factor, independent of population size or genetic constitution of the population.

Selection

Several modes of selection are available. In stabilizing selection modes, a specific optimum and selection intensity may be defined for each population and quantitative trait (Bürger, 2000). In directional selection modes, the strength and direction of selection may vary for each trait and population. By defining a fitness landscape it is possible to represent any selection pressure. Furthermore the selective pressures can change over time. Last, selection can be soft or hard (Wallace, 1968).

Initial settings

quantiNEMO is highly flexible in the setting of the initial simulation conditions. The initial population sizes may be set for each population and sex (parameters patch_ini_size, patch_ini_size_fem, or patch_ini_size_mal). Furthermore, the initial allele frequencies for each population may be either maximal polymorph or monomorph (parameters quanti_ini_allele_model and ntrl_ini_allele_model), or may be set explicitly for each population, locus, and allele (parameters quanti_allelic_file and ntrl_allelic_file).

CONFIDENTIAL

1.8 Input and output

Input

quantiNEMO is launched using a settings file. The settings file is a text file with flexible and easy to understand structure. The information is specified in a parameter-argument scheme, where the order of the parameters does not matter. The file can be edited with any text editor, and comments may be added for better readability. We provide also a syntax-highlighting definition for better readability.

Output

quantiNEMO provides summary statistics for the different simulation components, including genetic variance estimates, quantitative trait analysis (e.g. Q_{ST}), and F-statistics for all types of loci (neutral and QTL). Most of the summary statistics are available for juveniles and adults. The summary statistics can be computed for any generation during the simulation. quantiNEMO can also produce files with the raw genetic data. The genotypes at all loci can be dumped to file in the FSTAT (Goudet, 1995) or the Arlequin format (Excoffier, 2010). Phenotypes, as well as the additive, dominance, and epistatic effect values can be written to a file and then analyzed with any population or quantitative genetic software, e.g. to get patterns of differentiation, study linkage disequilibrium, or scan for QTLs. We are currently developing an R package to carry out the most common analyses from a simulation.

CONFIDENTIAL

Chapter 2

Getting started

CONFIDENTIAL

This chapter explains how to use quantiNEMO starting from the basics.

2.1 Installation

Executables of quantiNEMO for several operating systems can be downloaded from the web site http://www.unil.ch/popgen/softwares/quantinemo. The executables are standalones, meaning that quantiNEMO does not require an installation. After downloading the compressed file with the executable corresponding to your operating system simply extract it to a folder of your choice. The compressed file includes the executable for your operating system, the user manual, and also an example settings file.

2.2 Launching quantiNEMO

There are different ways to launch a simulation with quantiNEMO. Normally a simulation is defined using a settings file and the settings file is then passed to quantiNEMO, but it is also possible to specify the simulation parameters directly as arguments passed to the program. These two approaches may also be combined:

no argument

If no argument is passed to quantiNEMO during launching (double-clicking on the executable or by launching the executable via a console (e.g. `quantinemo.exe` for Windows)) then quantiNEMO will automatically use the settings file named `quantinemo.ini` residing next to the executable if present. If such a settings file (with the given name) is not present, then quantiNEMO will prompt you for the name of a settings file until a valid name is entered (relative and absolute paths are accepted) or "exit" is passed as file name (without quotes).

file name

The settings file name may be passed as a parameter to the executable when quantiNEMO is launched. This can be done using a console window:

```
> quantinemo.exe settings.ini
```

Depending on the operating system a '`./`' in front of the executable is sometimes needed to specify that the executable is located in the current directory (Linux):

```
> ./quantinemo settings.ini
```

parameters

quantiNEMO supports the Unix way of passing parameters to a program. Single character parameters require a '−' as prefix, longer parameter names need a '−−' as prefix. Parameters are accepted with or without arguments. Single character parameters may be concatenated if they don't have an argument. quantiNEMO accepts arguments with spaces if the spaces are within brackets or quotes (e.g. {}, (), or ""). The order of the arguments is irrelevant.

```
> quantinemo.exe -v
> quantinemo.exe --version
> quantinemo.exe --generations 200
```

Using this syntax it is possible to pass any simulation parameter to quantiNEMO. A simulation may be defined using a combination of settings file and parameters directly passed to the program. If a parameter is specified in both ways, then the parameter passed as argument is used. quantiNEMO considers any argument not belonging to a parameter to be the name of a settings file. To be sure that the settings file

CONFIDENTIAL

name is correctly passed, one may also use the parameter $--settings$ followed by the settings file name. Arguments of he parameters may be single values, words, matrices, temporal parameters or text within quotes. Thus any parameter may be specified as in the settings file except that the argument may not be written over several lines and the batch mode (see section 3.13) is also not functional.

```
> quantinemo.exe quantinemo.ini
> quantinemo.exe --settings quantinemo.ini
> quantinemo.exe quantinemo.ini --patch_capacity 10
> quantinemo.exe --patch_capacity {10 10} quantinemo.ini
```

The first and the last two commands and are equivalent.

CONFIDENTIAL

Chapter 3

Settings file

CONFIDENTIAL

This section describes the format of the settings file. The settings file is a text file with in general one parameter per line in a key-value scheme. For example

```
patch_capacity 1000
```

sets the parameter patch_capacity to the value of 1000. The order of appearance of the parameters In the settings file does not matter. However, a particular parameter should appear only once in the settings file. If a parameter appears several times in the file only the last instance is considered.

3.1 Default value

Most of the parameters have default values. The default value of a parameter is taken into account when either the parameter is not listed in the settings file or its argument is missing. The default values are listed behind the parameter name in this manual and are specified by (default:). The default values are the most common setting of the parameter. the default values allow to keep the settings file short and clear.

3.2 Comments

quantiNEMO allows to add comments in the settings file (and all other input files). There are two different types of comments:

Single line comments
A simple hash character '#' defines the start of a single line comment. The hash character and the remaining text of the line are ignored by quantiNEMO.

Block comments
Block comments may start and end at any place in the file. This allows to comment out multiple lines at once or only a part of a line. A block comment starts with the characters '#/' and ends with the characters '/#'. The starting and ending characters and the text between them are ignored.

CONFIDENTIAL

3.3 Line end

In general a parameter (the key and its argument) has to be written on a single line (except for matrices and temporal parameters). However, using a backslash '\' it is possible to bypass the end of a line and to write an argument on several lines. Note, that after the backslash any text on the line is removed.

3.4 Parameter types

There are different types of arguments that a parameter may take. In the following the argument type is specified within square brackets. Example:

```
stat_log_time [integer]
```

Integer
Integers are whole-numbers, i.e. a dot-less number. The following forms are equivalent: 1000 or 1e3.

Decimal

Decimals are floating-point numbers. The following forms are equivalent: 0.0001, .0001 or 1e-4.

String

Strings are text arguments. If the string contains spaces the argument has to be enclosed within quotation marks "...". When a string is enclosed by quotation marks it may be written over several lines. Example of a string with a space:

```
folder "first simulation"
```

Matrix

Matrices allow to pass several numbers (integer or decimal) to a parameter. This may be necessary to specify carrying capacities (see section 1D Matrix), or to pass a dispersal matrix (see section 2D Matrix) to quantiNEMO. Matrices are enclosed between curly brackets '{ }', and numbers are separated by at least a space. Matrices may be written on several lines and may also contain comments. There is no *a priori* restriction on the size of the matrix.

CONFIDENTIAL

1D Matrix

A one dimensional matrix (vector) consists of data in a single dimension. There are three different ways to write a one dimensional matrix, which are all equivalent:

```
patch_number 4
patch_capacity {  20   10   20   10  }
patch_capacity { {20   10   20   10} }
patch_capacity { {20} {10} {20} {10} }
```

2D Matrix

Some parameters need a second dimension for their argument. A second dimension is obtained by enclosing the inner rows of the matrix again within curly brackets '{ }'.

```
patch_number 4
disp_rate { {0.2  0.0  0.0  0.8}
            {0.4  0.2  0.0  0.4}
            {0.4  0.4  0.2  0.0}
            {0.0  0.4  0.4  0.2} }
```

This example shows the pairwise dispersal matrix for 4 patches (4x4). Each row specifies a source patch from which emigrants emigrate. Each column specifies the target patch receiving the immigrants. The diagonal of the matrix specifies the proportion of individuals remaining in the natal patch.

Matrix length adjustment

Usually matrices are defined in whole, i.e. the number of carrying capacities in the 1D matrix example above meets the number of patches (4 patches). However, if there is a repeating pattern in the matrix, it is also possible to define only the repetition. In this case the matrix will be repeated as needed. For instance the 1D matrix above could also be written in one of the following ways as there is a repetition in it:

```
patch_number 4
patch_capacity {20  10}
```

Note, that rows and/or columns are repeated as needed. If the number of columns (or rows) is not an entire subset a warning will be returned and the simulation will adjust the matrix as:

```
patch_number 5
patch_capacity {20  10}
```

In this example we have 5 patches, however the carrying capacities are only specified for 2 patches. As 2 is not an entire subset of 5 a warning will be returned and the patches will have the following carrying capacities: 20, 10, 20, 10, 20. If all values of a matrix are identical one may leave out the brackets. In the following example all three declarations result in the same simulation:

```
patch_number 4
patch_capacity {20  20  20  20}
patch_capacity {20}
patch_capacity  20
```

Unbalanced matrices

Usually a matrix is balanced, i.e. each row has the same number of columns. However, some parameters (such as the parameter `quanti_loci_positions`) allow to have unbalanced matrices, i.e. rows do not necessarily contain the same number of columns.

CONFIDENTIAL

Other parameters (such as the parameter `quanti_loci_positions`) allow to skip a row, i.e. some rows do not contain any data. A row can be skipped by explicitly indicating the rank of the row just after the beginning of the row followed by a colon ":". The ranking starts with 1. If a row has no explicit rank it is assumed to follow the preceding row. Here is an example for the genetic map of a quantitative trait:

```
quanti_loci 11
quanti_loci_positions { {1: 10}
                        {3: 20  40  60  80  100}
                        {   20  40  60  80  100} }
```

In this example the quantitative trait is defined by 11 loci located on three out of four chromosomes. The first chromosome contains a single locus at position 10 cM. No locus is located on the second chromosome. The third and fourth chromosome have the same structure: 5 loci are located on each of the two chromosomes, and the distance between adjacent loci is 20 cM.

CONFIDENTIAL

3.5 Temporal parameters

An important feature of quantiNEMO is that some parameters may change over time during a simulation. Such parameters are indicated as "temporal" in this manual. Temporal arguments are enclosed within two parentheses '(...)'. For each change of the argument over time a pair consisting of a generation index and a corresponding argument is needed. The values of a pair are separated by at least one space. Pairs are separated by a coma ',' or by a semi-colon ';'. The first value of a pair specifies the time, i.e. before which generation the change happens. The second value is the new argument. A parameter may change as often as any generation. A simulation starts at generation 1. Therefore the first change has to have the time value 1 otherwise the parameter cannot be initialized leading quantiNEMO to return an error. A temporal argument may be written over several lines.

```
patch_capacity (1    100,
                50   200,
                100  500)
```

Here, the carrying capacity is 100 for the first 49 generations, 200 from generation 50 on, and 500 from generation 100 on.

3.6 Keywords

quantiNEMO supports keywords in the settings file. A keyword may be defined using the word 'set', followed by the keyword and the argument. The keyword can be used in parameter arguments and quantiNEMO will replace all keywords by their defined arguments. A keyword supports any argument which makes sense for the parameter where the keyword is used, including macros and temporal parameters.

```
set NUM_GEN runif(1, 100, 1000)
generations NUM_GEN
stat_log_time NUM_GEN
```

In this example the keyword 'NUM_GEN' is defined as a random macro. Consequently both parameters **generations** and **stat_log_time** are set to the same value. In other words, this settigns file allows to easily define randomly the number of generations to simulate between 100 and 1000 generations and to compute the statistics at the last generation.

CONFIDENTIAL

3.7 External files

In general arguments are written directly after the parameter name on a single line. However, arguments may be sometimes large (e.g. big matrices, temporal parameters, ...) leading to poorly readable settings file. Using external files for large arguments allows to keep the settings file clear and well readable. An external file may be used for any parameter. Instead of writing the argument directly after the parameter name, the name of an external file is written after the parameter name. In order for quantiNEMO to recognize the argument as a file name the prefix '$' has to be added before the file name. The external file must contain the argument of the parameter. Only a single argument per external file is possible, however the same external file may be used for several parameters. The format of the argument in the external file follows the same rules as in the settings file, except that line

breaks are ignored, i.e the character '\' between lines is not necessary. Here is an example:

settings file:

```
disp_rate $dispersal_file.txt
```

external file named "dispersal_file.txt":

```
# dispersal rates
{ {0.2  0.0  0.0  0.4  0.4}
  {0.4  0.2  0.0  0.0  0.4}
  {0.4  0.4  0.2  0.0  0.0}
  {0.0  0.4  0.4  0.2  0.0}
  {0.0  0.0  0.4  0.4  0.2} }
```

3.8 Parameters to quantiNEMO

Parameters are normally defined in a settings file which is then passed to the command line. All parameters and there arguments may also be passed directly to the command line. Command line options provide a convenient mechanism to override commonly altered parameter values. Parameters and their argument passed to the command line override the arguments defined in the settings file. (see section 2.2).

CONFIDENTIAL

3.9 Macros

quantiNEMO supports several macros allowing to write more easily the settings file. In principle the macros are interpreted and replaced before quantiNEMO reads the settings file. Several macros allow to draw random deviates from a given distribution. Note, that the random deviates are only drawn once, i.e. all replicates have thus the same numbers. The names and parameters of all macros follow as good as possible the equivalent functions in the statistical package R. To understand or control the outcome of a macro the log file (see parameter **logfile_type**) may be consulted. This log file may either contain the macros as they were specified or the interpreted macros.

option "Time"

Adding this option to most of the macro allows generating temporal arguments. If this option is used with a macro allowing to draw random deviates it is possible to simulate temporal fluctuations. In principle the macro draws for the given time interval random numbers following the specifications and generates a corresponding temporal argument with *nb* intervals. The option *Time* has to follow directly the macro name, e.g. *rnormTime* for the macro *rnorm* and the arguments of the macro have to be extended at the end by two values which are the start and the end time, e.g. *rnorm(nb, mean, sd, timeFrom, timeTo)*. In this example the number of random deviates (*nb*) specifies the number of regular temporal changes between *timeFrom* and *timeTo*. The time intervals are generated internally as $seq(timeFrom, timeTo, nb)$. An example of such a temporal macro is shown with the macro *seq* below.

seq($from, to, steps$**)**
seqTime($from, to, steps, timeFrom, timeTo$**)**

With this macro it is possible to specify a sequence of data points. *seq* works similar as *seq* in the statistical package R. The macro needs three arguments separated by a comma: $from$ is the first data point of the sequence, *to* is the last data point of the sequence and *steps* specifies the number of data points including the edges (equivalent to *length.out* in R). $from$ and *to* may be single values or matrices while *steps* has to be a number. If the matrices of $from$ and *to* do not have identical dimensions the matrices are adjusted to each other. Example of *seq*:

```
patch_capacity {seq(100, 1000, 10)}
patch_capacity {100 200 300 400 500 600 700 800 900 1000}
```

Both specifications of the carrying capacities are identical, i.e. the macro *seq* translates its arguments to the lower specification.
The command *seq* can also be used for temporal sequences:

```
patch_capacity (seqTime(200, 1000, 5, 1, 5))
patch_capacity (1 200, 2 400, 3 600, 4 800, 5 1000)
```

```
patch_stab_sel_optima (seqTime({{1 10}},{{10}{1}}, 10, 1 10))
patch_stab_sel_optima (1 {{1 9}{1 9}},
                       2 {{2 9}{1 8}},
                       3 {{3 9}{1 7}},
                       4 {{4 9}{1 6}},
```

CONFIDENTIAL

```
5 {{5 9}{1 5}},
6 {{6 9}{1 4}},
7 {{7 9}{1 3}},
8 {{8 9}{1 2}},
9 {{9 9}{1 1}})
```

Both specifications of the linear increase of the carrying capacities and selection optima, respectively, over time are identical. Again the macro *seqTime* translates its arguments into the lower specification.

seq2D($xlim, ylim, patchID1, patchID2, value1, value2, steps$))
This macro allows to generate a two dimensional cline of values. For example this allows generating a cline of carrying capacities across a two dimensional landscape. Since the macro does not know the other parameter arguments all necessary information has to be passed to the macro, i.e. some information may be redundant in the settings file. In principle the cline is defined by the values at two patches and the number of steps between these two patches to interpolate. The changes of numbers are vertical to the line between the two patches. $xlim$ and $ylim$ are the matrix dimensions, $patchID1$ and $patchID2$ are the patch-IDs for which the values will be defined and the corresponding values are $value1$ and $value2$, respectively. *step* is the number of changes between these two patches, following the macro *seq*.

seq2Db($xlim, ylim, patchID1, patchID2, value1, value2, steps$))
This macro is similar to the previous one $seq2D$ but the interpolation for the cline follows another rule: The value of any patch is defined by the ratio of the distances from this patch to the two specified patches.

rep($text, number$)
With this macro it is possible to specify a repetition of the text. *rep* works similar to *rep* in the statistical package R. The macro needs two arguments separated by a comma: the first one is the text to be repeated, and the second argument specifies the number of repetitions. Example:

```
patch_ini_size {1000 rep(0, 9)}
patch_ini_size {1000 0 0 0 0 0 0 0 0 0}
```

Both specifications of the initial population sizes are identical, i.e. the macro *rep* translates its arguments to the lower specification. In this

CONFIDENTIAL

example only the first patch is populated at the start of the simulation, allowing to simulate a colonization scenario.

runif(nb)
runif(nb, min, max)
runifTime($nb, timeFrom, timeTo$)
runifTime($nb, min, max, timeFrom, timeTo$)

Random numbers may be drawn from a uniform distribution. nb specifies the number of random deviates to draw. min and max specify the range of the distribution. If min and max are not specified the random deviates are drawn within 0 and 1. The returned values are always decimal numbers. To obtain entire random deviates of an uniform distribution the macro has to be set within the macro *ceil*.

rnorm($nb, mean, sd$)
rnormTime($nb, mean, sd, timrFrom, timeTo$)

Random numbers may be drawn from a normal distribution, where $mean$ is the mean of the normal distribution, sd the standard deviation of the normal distribution, and nb specifies the number of random deviates to draw. $mean$ and sd may be single values or matrices, while nb has to be a number.

rnorm($nb, mean, sd, min, max$)
rnormTime($nb, mean, sd, min, max, timrFrom, timeTo$)

Same as above but random numbers are drawn from a truncated normal distribution, with min as lower bound and ma as upper bound. min and max have to be single numbers.

rlnorm($nb, meanlog, sdlog$)
rlnormTime($nb, meanlog, sdlog, timeFrom, timeTo$)

Random numbers may be drawn from a log normal distribution, where $meanlog$ is the mean of the log normal distribution on the log scale ($log(mean)$), $sdlog$ the standard deviation of the log normal distribution on the log scale ($log(sd)$), and nb specifies the number of random deviates to draw. $meanlog$ and $sdlog$ may be single values or matrices, while nb has to be a number. Note that the parameters to pass to the function are not the $mean$ and $standarddeviation$ of the underlying distribution. The statistics of the underlying distribution $mean$ and sd may be obtained as follows:

$$mean = e^{meanlog + \frac{sdlog^2}{2}}$$

CONFIDENTIAL

$$sd = \sqrt{(e^{sdlog^2} - 1)e^{2meanlog+sdlog^2}}$$

If these equations are transformed it allows to compute the input parameters *meanlog* and *sdlog* for the macro if the statistics of the underlying distribution are known (i.e. *mean* and *sd*):

$$meanlog = log\,(mean) - \frac{1}{2} log\left(\left(\frac{sd}{mean}\right)^2 + 1\right)$$

$$sdlog = \sqrt{log\left(\left(\frac{sd}{mean}\right)^2 + 1\right)}$$

rlnorm(*nb*, *meanlog*, *sdlog*, *min*, *max*)
rlnormTime(*nb*, *meanlog*, *sdlog*, *min*, *max*, *timeFrom*, *timeTo*)
Same as above but random numbers are drawn from a truncated log normal distribution, with *min* as lower bound and *ma* as upper bound. *min* and *max* have to be single numbers.

rgamma(*nb*, *shape*, *scale*)
rgammaTime(*nb*, *shape*, *scale*, *timeFrom*, *timeTo*)
Random numbers may be drawn from a gamma distribution, where *shape* is the shape of the gamma distribution, *scale* the scaling factor, and *nb* specifies the number of random deviates to draw. *shape* and *scale* may be single values or matrices, while *nb* has to be a number.

rbeta(*nb*, *alpha*, *beta*)
rbeta(*nb*, *alpha*, *beta*, *from*, *to*)
rbetaTime(*nb*, *alpha*, *beta*, *timeFrom*, *timeTo*)
rbetaTime(*nb*, *alpha*, *beta*, *from*, *to*, *timeFrom*, *timeTo*)
Random numbers may be drawn from a beta distribution, where *alpha* and *beta* define the shape of the beta distribution, *min* and *max* specify the range of the beta distribution, and *nb* specifies the number of random deviates to draw. If *min* and *max* are not specified the random deviates are drawn in the standard beta distribution ranging from 0 to 1. *alpha*, *beta*, *min*, and *to* may be single values or matrices, while *nb* has to be a number.

rpois(*nb*, *mean*)

CONFIDENTIAL

rpoisTime(*nb*, *mean*, *timeFrom*, *timeTo*)
Random numbers may be drawn from a discrete poisson distribution, where *mean* is the mean of the poisson distribution, and *nb* specifies the number of random deviates to draw. Note, since the poisson distribution is discrete the macros *rpois*() and *rpoisI*() are identical. *mean* may be a single value or a matrix, while *nb* has to be a number.

rbinom(*nb*, *size*, *prob*)
rbinomTime(*nb*, *size*, *prob*, *timeFrom*, *timeTo*)
Random numbers may be drawn from a discrete binomial distribution, where *size* is the number of trials, *prob* is the probability of success on each trial, and *nb* specifies the number of random deviates to draw. *size* and *prob* may be single values or matrices, while *nb* has to be a number.

rsample(*nb*, *with_replacement*?, *elem*1, *elem*2, *elem*3, ...)
rsampleTime(*nb*, *with_replacement*?, *elem*1, *elem*2, *elem*3, ..., *timeFrom*, *timeTo*)

This macro allows to sample within the given elements (numbers or text supported). *nb* specifies the number of elements to output. The second parameter specifies if the numbers are randomly drawn with replacement (1) or not (0). Any following element separated by a comma is considered as an element allowed to be sampled. Note, that without replacement the number of possible elements to draw may not exceed the number of input elements.

round(*elements*)
This macro rounds all numbers within the brackets to entire numbers.

ceil(*elements*)
This macro ceils all numbers within the brackets to entire numbers, i.e. to the next higher entire number.

floor(*elements*)
This macro floors all numbers within the brackets to entire numbers, i.e. to the next lower entire number.

trunc(*elements*)
This macro truncates the numbers at zero, i.e. all negative numbers are set to zero.

CONFIDENTIAL

equation(...)
: This macro allows to solve simple equations from left to right. Supported are arithmetics on single values, matrices, or a combination of them. It supports the arithmetics plus, minus, product and division. Values and arithmetic signs have to be separated by any space.

In principle macros may be combined within each other or may be concatenated:

```
patch_capacity {rnorm(1000, 1000, 6)}
# {1032.52 968.315 1006.58 1009.23 974.836 976.087}

patch_capacity {round(rnorm(1000, 1000, 6))}
# {972 1001 998 998 973 1003}

patch_capacity {rep(round(rnorm(1000, 1000, 3), 2))}
# {942 1055 1035  942 1055 1035}
```

In all three examples the mean carrying capacity is 1000, however the capacity of the individual patches varies following a normal distribution with variance 1000. By default the random generator draws double deviates as shown in the first example. Note, that this argument will be accepted by quantiNEMO, however since this parameter expects entire numbers quantiNEMO will round the numbers down to the next smaller entire number. By adding the *I* for *integer* to the distribution name the double values are rounded to the next entire number as shown by the second example. The third example illustrates how the macros may be encapsulated with each other resulting in two patches having always the same capacity.

CONFIDENTIAL

3.10 Output files

3.10.1 Types of output

quantiNEMO can generate the following types of outputs:

summary statistics
: quantiNEMO provides summary statistics for the different simulation components, including genetic variance estimates, quantitative trait

analysis (e.g. h^2, V_A, genetic diversity), and F-statistics for all types of loci (neutral and QTL). Most of the summary statistics are available for juveniles and adults. The summary statistics can be computed for any generation during the simulation. The summary statistics can be saved in two ways: Either they are stored for each replicate separately and/or the summary statistics are averaged across replicates.

raw data

quantiNEMO can also produce files with the raw genetic and phenotypic data. The genotypes at all loci can be dumped to file in the FSTAT (Goudet, 1995) or Arlequin format (Excoffier, 2010). Phenotypes, as well as the additive, dominance, and epistatic effect values can be written to a file and then analyzed with any population or quantitative genetic software, e.g. to get patterns of differentiation, study linkage disequilibrium, or scan for QTL. Genotypes and phenotypes can be saved for any generation during the simulation.

log files

quantiNEMO also generates log files allowing to reconstruct performed simulations. There are two types of log files. The first log file records the simulations performed with quantiNEMO and stores some general information. This log file is stored in the folder of the executable and allows to reconstruct the chronology of performed simulations and their main features. The other log file contains the used parameters, is generated for each simulation separately, and is stored in the simulation folder. This file is in principle a copy of the used settings file and contains the starting time and the duration time of the simulation. It contains also the seed (see parameter seed) used to initialize the simulation. This file can be used as settings file to exactly repeat the performed simulation. Note, that due to the seed in the file the random generator will be initialized in the same way leading to the exact same values in the output.

CONFIDENTIAL

3.10.2 Naming convention

All files of a simulation are stored in a unique folder (see parameter folder). This simulation folder may contain a substructure. The name of the output files are based on the base name given by the parameter filename. Depending

on the type of output different extensions are added to the base name. To avoid that recurring outputs overwrite previous outputs a counter is added to the file name between base name and extension. There are two types of counters: the generation counter and the replication counter. A counter is only added if there is a risk of overwriting. For example the replication counter is only added if several replications are performed. The generation counter starts with "_g" and the replication counter with "_r". These characters are followed by the number of the generation and replication, respectively. Note, that generations and replications start at 1. The number has as many digits as are needed to represent the highest number in the simulation:

```
simulation_g0001_r01.dat
simulation_g0002_r01.dat
...
simulation_g5000_r10.dat
```

CONFIDENTIAL

3.11 Minimal settings file

Most of the parameters have default values. This allows to make short and clear settings files. This section describes the parameters needed for a minimal settings file and describes the simulated model, respectively how the default values are set.

The following two parameters are needed in every settings file:

```
generations     500
patch_capacity  1000
```

This minimal settings file allows to perform a simulation with a single population consisting of 1000 hermaphrodites. The population evolves under neutral random mating for 500 generations keeping the population size constant at carrying capacity. No genetic data are simulated, since no quantitative traits or neutral markers are specified. The simulation generates no output apart from the log file. Although this simulation works it makes no sense, since no output is generated. This can be improved by specifying quantiNEMO to compute summary statistics on the demography:

```
generations     500
patch_capacity  1000
stat            {adlt.demo}
```

This settings file results in exactly the same simulation as the previous one, but now three additional files are generated. The first file lists the names of the computed summary statistics ("simulation_legend.txt"), while the two other files are almost identical containing the summary statistics for each generation. One of these latter files contains the summary statistics for each generation and replicate separately ("simulation.txt"), while the other one shows the summary statistics averaged across replicates ("simulation_mean.txt").

To simulate genetic data one has to add either a quantitative trait or a neutral marker to the simulation. This is done by specifying the number of loci to simulate (for quantitative traits and neutral markers separately). To force quantiNEMO to compute some summary statistics on quantitative traits, respectively on neutral markers one has also to specify corresponding summary statistics. For example:

```
generations      500
patch_capacity   1000
stat             {adlt.demo
                  quanti
                  n.adlt.fstat}
quanti_loci      1
ntrl_loci        1
```

CONFIDENTIAL

In this example now selection acts on the reproduction stage (soft selection). The fitness is computed on the simulated quantitative trait. The trait consists of a single locus with up to 255 alleles. Allelic effects are normally distributed with a variance of 1. The phenotype of the quantitative trait is determined by pure additive effects of the alleles. Stabilizing selection acts on the phenotype with an optimum of 0 and a variance of 1. A single neutral marker locus is simulated with up to 255 alleles. Both the quantitative trait locus and the neutral marker locus do not mutate. for both markers some common summary statistics are computed.

3.12 Simulation example

This section describes a more realistic simulation example and describes how the output is stored. The settings file of this example named "quantiNemo2_example.ini" is included in the compressed folders of the downloads of quantiNEMO:

```
generations                 100

# metapopulation
patch_number                10
patch_capacity              1000
dispersal_rate              0.01

# mating
selection_level             0
mating_system               0

# selection
patch_stab_sel_optima       10
patch_stab_sel_intensity    1

# quantitative trait
quanti_loci                 5
quanti_all                  255
quanti_mutation_model       0
quanti_mutation_rate        1e-4
quanti_save_genotype        2
quanti_genot_logtime        10
quanti_genot_dir            quanti_genotype
quanti_save_phenotype       2
quanti_phenot_logtime       10
quanti_phenot_dir           quanti_phenotype

# neutral marker
ntrl_loci                   5
ntrl_all                    10
ntrl_mutation_model         0
ntrl_mutation_rate          1e-4

ntrl_save_genotype          2
ntrl_genot_logtime          10
ntrl_genot_dir              ntrl_genotype

# statistics
stat                        {n.adlt.fstat}
stat_save                   1
stat_log_time               10
stat_dir                    stats
```

CONFIDENTIAL

A simulation based on this settings file named "quantiNemo2_example.ini"

produces the following output to your terminal window:

```
************************************************

                quantiNEMO
************************************************
*     Release: 1.5.0 [May 28 2009; 09:58:32]     *
*    Copyright (C) 2008 Samuel Neuenschwander    *
* http://www.unil.ch/popgen/softwares/quantinemo *
************************************************

Reading settings file '../quantiNEMO.ini' ...
Reading settings file '../quantiNEMO.ini' done (29 parameters)

SETTINGS
  Simulation:
    10 generations
    1 replicates

  Loaded traits:
    Quantitative trait: 5 loci; 255 alleles; stabilizing selection
    Neutral marker type: 5 loci; 10 alleles

  Life cycle sequence:
    1. breed
    2. save_stats
    3. save_files
    4. aging
    5. disperse

  Metapopulation:
    10 populations
    Migration model: island
    Mating system: random mating (hermaphrodite)

  Genetic map:
    10 independend loci


SIMULATION
    replicate 1/1 [00:00:14] 100/100

------ SIMULATION done (CPU time: 00:00:15s) ------

quantiNEMO terminated successfully!
```

CONFIDENTIAL

This output informs you that this simulation was parameterized by the settings file "quantiNemo2_example.ini". The simulation consisted of one quantitative trait and one type of neutral markers. The simulated life cycle is indicated and shows that summary statistics and genotypes or phenotypes are output. Only one replicate of 100 generations was performed. For each replicate the elapsed time (hh:mm:ss) is printed out to the console and at the end of the simulations also the total elapsed time. The output of this simulation was stored in the following structure relative to the executable:

```
simulation_2008-01-01_00-00-00/     # automatically named folder
    simulation.log                  # log file
    quanti_phenotype/               # phenotype folder
        simulation_g010.phe         # phenotypes of generation 10
        simulation_g020.phe
        simulation_g030.phe
        simulation_g040.phe
        simulation_g050.phe
        simulation_g060.phe
        simulation_g070.phe
        simulation_g080.phe
        simulation_g090.phe
        simulation_g100.phe
    quanti_genotype/                # genotype folder (quantitative)
        simulation_g010.dat         # genotypes at generation 10
        simulation_g020.dat
        simulation_g030.dat
        simulation_g040.dat
        simulation_g050.dat
        simulation_g060.dat
        simulation_g070.dat
        simulation_g080.dat
        simulation_g090.dat
        simulation_g100.dat
    ntrl_genotype/                  # genotype folder (neutral)
        simulation_g010.dat         # genotype at generation 10
        simulation_g020.dat
        simulation_g030.dat
        simulation_g040.dat
        simulation_g050.dat
        simulation_g060.dat
        simulation_g070.dat
        simulation_g080.dat
        simulation_g090.dat
        simulation_g100.dat
```

CONFIDENTIAL

```
    stats/                          # statistic files
        simulation.txt              # statistics for each replicate
        simulation_mean.txt         # statistics across replicates
        simulation_legend.txt       # statistic legends
```

3.13 Batch mode

quantiNEMO allows to perform multiple simulations by executing a single command. There are two ways to perform such batch simulations. Note, that these two types may not be mixed, i.e. be used at the same time.

3.13.1 Multiple settings files

A normal simulation can be launched by passing the settings file name as parameter to the executable. It is also possible to pass several settings file names to the executable. In this case a simulation for each settings file is executed consecutively:

```
> quantinemo.exe sim1.ini sim2.ini
```

In this example quantiNEMO is launched with two settings files (`sim1.ini` and `sim2.ini`). The simulations will be executed one after the other leading to the following console output:

```
...
Reading settings file 'sim1.ini' ...
Reading settings file 'sim1.ini' done (29 parameters)

Reading settings file 'sim2.ini' ...
Reading settings file 'sim2.ini' done (29 parameters)

------ SIMULATION 1/2 ------

...

------ SIMULATION 1/2 done (CPU time: 00:00:15s) ------


------ SIMULATION 2/2 ------
```

CONFIDENTIAL

```
...
------ SIMULATION 2/2 done (CPU time: 00:00:16s) ------
```

3.13.2 Sequential parameters

A batch simulation may also be launched by a single settings file if so-called sequential parameters are used. Sequential parameters are any parameter with not only one but several arguments. Note, that a sequential parameter is not the same as a temporal parameter (see section 3.5).

```
patch_capacity 5 10 20
```

In this example patch_capacity is a sequential parameter with three arguments. If patch_capacity is the only sequential parameter three consecutive simulations will be launched with identical parameter arguments, except for the parameter patch_capacity which will be set to 5 for the first simulation, to 10 for the second simulation, and to 20 for the third simulation.

CONFIDENTIAL

If several parameters are sequential parameters all combinations of the sequential parameters will be simulated. Example:

```
patch_number    10 50
patch_capacity 5 10 20
```

There are two sequential parameters in this example. This will result in 6 consecutive simulations with the following parameters:

	patch_number	patch_capacity
1. simulation	10	5
2. simulation	10	10
3. simulation	10	20
4. simulation	50	5
5. simulation	50	10
6. simulation	50	20

To prevent the output from overwriting the preceding simulation a unique base file name is given to each simulation. This unique base file name consists

of the the base file name (parameter filename) plus a suffix which includes the rank of the simulation. If in the example above the parameter filename was set to "mysim" the base name for each simulation would be as follows:

	filename
1. simulation	mysim−1
2. simulation	mysim−2
3. simulation	mysim−3
4. simulation	mysim−4
5. simulation	mysim−5
6. simulation	mysim−6

However, the base file name can also be individualized by the user using expansion characters '%' in the base file name. The expansion characters allow to incorporate the changing argument value (of the sequential parameters) in the base file name. For this the expansion character '%' has to be followed by a number which corresponds to the rank of the sequential parameter. Note, that the rank corresponds to the alphabetical order of the sequential parameter names, starting with 1 for the first sequential parameter. The expansion characters and the rank numbers are then automatically replaced by the corresponding arguemnt values used in the simulation. Expansion characters can be applied to all types of arguments, however arguments of matrices and temporal parameters are not replaced by the argument value, but by the rank number of the argument (how they appear in the settings file after the parameter name) due to their big size. So each sequential parameter can be addressed by its rank allowing to build separate filenames. If not all sequential parameters are addressed by the filename, i.e if the base name is not unique for each simulation, the rank of the simualtion is added as suffix to the filename to avoid overwritting the output. Note, that after the rank number (which may consists of several digits) a character must follow which cannot be interpreted as a number by quantiNEMO. For example the first sequential parameter has to be called as "name_%1_4K" and not as "name_%14K". If the filename for the example above was set to "sim_%2pop_%1ind" the following base names would be generated (alphabetically patch_capacity comes before pach_number):

	filename
1. simulation	sim_10pop−5ind
2. simulation	sim_10pop−10ind
3. simulation	sim_10pop−20ind
1. simulation	sim_50pop−5ind

CONFIDENTIAL

```
5. simulation    sim_50pop-10ind
6. simulation    sim_50pop-20ind
```

CONFIDENTIAL

Chapter 4

Life Cycle

CONFIDENTIAL

quantiNEMO is a discrete generation-based simulator. This means that a single individual undergoes only once the life cycle and that the generations are not overlapping. Depending on the parameterization in the settings file some events may be skipped. The life cycle has a fixed order of events. A simulation starts with "Breeding", i.e. only adults are present at the initialization of the simulation. The life cycle is repeated for each generation, i.e. the life cycle event "Breeding" follows the life cycle event "Extinction":

1 : Breeding. Adults mate and may produce offspring. Selection acts at this stage.

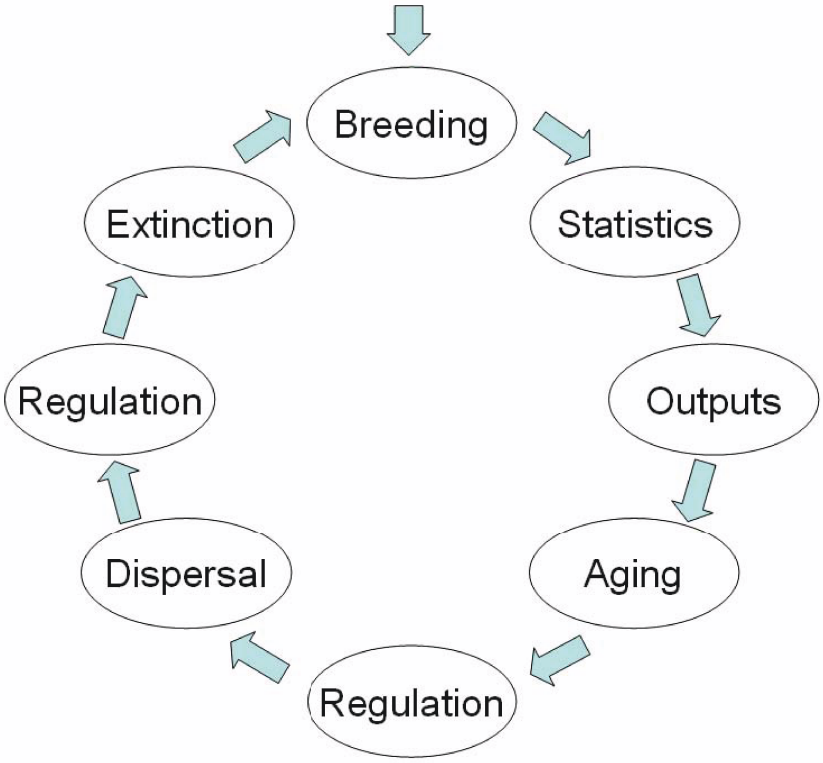
2 : Statistics. Adults and juveniles are present. It is the stage were summary statistics may be computed for adults and juveniles.

3 : Outputs. Adults and juveniles are still present. Genotypes and/or phenotypes may be dumped to files for adults and/or juveniles.

4 : Aging. It is the event where the adults are removed. Only the juveniles remain in the model.

5 : Regulation. Before dispersal some patches may be overcrowded. This regulation stage allows to control the population sizes.

6 : Dispersal. Juveniles may migrate to other patches and become adults.



7 : Regulation. After dispersal some patches may be overcrowded. This regulation stage allows to control the population sizes.

8 : Extinction. Due to stochastic events populations may go extinct.

In the following the life cycle events and their options are described in details:

4.1 Breeding

This stage performs mating and reproduction following the selected mating system. The implemented reproduction model in quantiNEMO consists of two steps: First for each patch the number of offspring to be produced is defined. This number of offspring depends on the parameter mating_nb_offspring_model and if selection acts at this stage also on the level of selection (see parameter selection_level). In a second step parents are randomly assigned for each offspring (parents are coming form the same patch).

This assignment of the parents to the offspring depends on the mating system (parameter mating_system) and on the fitness of the parents if selection acts. Thereby, adults with a higher fitness have on average a higher reproductive success. By default selection acts at the reproductive success. Adults are not removed at this stage (adults are removed in the event aging). The following parameters allow to parametrize this life cycle event:

mating_system [0-6] (default: 0)

Five general mating systems are implemented in quantiNEMO. The assignment of the parents to the offspring is random depending on the fitness of the local parents (if no selection acts all individuals have a fitness of 1). Thereby, adults with a higher fitness have on average a higher reproductive success):

0 : random mating (hermaphrodite). For each offspring two hermaphrodite parents are randomly assigned. With probability $1/N$ these two hermaphrodites are identical which leads to selfing. Females are used to simulate hermaphrodites.

1 : selfing (hermaphrodite). For each offspring a hermaphrodite is randomly assigned to self fertilize. The parameter mating_proportion allows to set the proportion of outcrosses. quantiNEMO controls that the proportion of outcrosses is met, i.e. that outcrossing does not result by chance ($1/N$) in selfing. Females are used to simulate hermaphrodites.

2 : cloning (hermaphrodite). This mating system is identical to the model selfing but without any recombination. The genotype of the offspring is identical to the genotype of the mother (only females are simulated) except for changes due to mutations. The parameter mating_proportion allows to set the proportion of sexual reproduction (random mating). Females are used to simulate hermaphrodites.

3 : random mating (promiscuity). This is random mating with two sexes. For each offspring a father and a mother are randomly assigned.

4 : polygyny. Depending on the parameter mating_males only one (default) or several males per patch may reproduce. This fixed

CONFIDENTIAL

number of reproductive males are selected randomly depending on their fitnesses, i.e. the reproductive males have on average a higher fitness. Then for each offspring a mother and one of these reproductive males are randomly assigned depending on their fitnesses. Thus reproductive males with higher fitnesses have a higher reproductive success among the reproductive males. If no selection acts (parameter selection_level set to 3) the reproductive males are randomly chosen (all males have the same probability) and each male has the same probability to father an offspring. The parameter mating_proportion allows to set the proportion of random matings between any male and female, i.e. also males of the non-reproductive group may get the chance to reproduce.

5 : monogamy. For each female a male is randomly assigned to be its partner for all offspring. If there are less females than males present in the patch, not all males will mate. In contrast, if there are more females than males present in the patch, males may belong to several mating pairs. For each offspring a parent pair is randomly assigned depending on the fitness of the female (if no selection acts the parent pairs have the same probability to be selected). Thus parent pairs, where the females have a higher fitness have on average a higher reproductive success. The parameter mating_proportion allows to set the proportion of random matings.

6 : no mating/reproduction. In this case no mating or reproduction occurs, i.e. this life cycle event is skipped. This option allows to use quantiNEMO as a statistical package. The input data may be passed as initial genotype files (ntrl or quanti) in the FSTAT format (see sections 9.3 and 10.3). The number of generations has to be set to one since a simulation without any mating and reproduction does not make sense.

Models and their specific parameters

CONFIDENTIAL

model	additional parameters
0 : random mating (herma.)	fem_sex_allocation
1 : selfing (herma.)	mating_proportion / fem_sex_allocation
2 : cloning (herma.)	mating_proportion
3 : random mating (prom.)	sex_ratio
4 : polygyny	sex_ratio / mating_proportion / mating_males
5 : monogamy	sex_ratio / mating_proportion
6 : no mating/reproduction	

mating_proportion [decimal] (temporal/default: 1)

This parameter allows to specify the ratio of a special mating system (selfing, cloning, polygyny, or monogamy) in relation to random mating. A value of 1 (default) means that only the special mating occurs and a value of 0 means that only random mating occurs. For example if we want to simulate a plant with a selfing rate of 90% we have to set the parameter **mating_system** to 1 (selfing) and this parameter **mating_proportion** to 0.9. These settings will lead to 90% selfing and 10% random mating. Note, that quantiNEMO controls that the ratio is met, i.e. that selfing does not occur by chance (probability would be $1/N$) when random mating should occur.

mating_males [integer] (default: 1)

This parameter sets the number of males that will be available for mating within each patch. The parameter will only be used if the mating system is polygyny (parameter mating_system set to 4). The range of values is between 1 (a single male mates) and the carrying capacity of the males (all males may mate).

sex_ratio [decimal] (default: 1)

This parameter allows to specify the ratio of males to females of the offspring in a patch. If hermaphrodites are simulated (parameter mating_system is set to 0, 1 or 2) the sex ratio is not considered, respectively set to 0 (females are used to simulate hermaphrodites).

mating_nb_offspring_model [0-9] (default: 0)

This parameter specifies how the total number of offspring (N_{Off}) is determined. Depending on the selection level (see parameter selection_level) the total number of offspring is computed at the patch (soft and hard selection) or metapopulation (metapopulation selection) level.

CONFIDENTIAL

In the latter case the number of offspring of the entire metapopulation is then distributed among the patches based on the mean fitnesses of the populations. In case of hard selection (parameter selection_level set to 2) this parameter specifies the total number of offspring per patch assuming a maximal fitness of 1 for all adults.

0 : carrying capacity.

$$N_{Off} = K$$

The total number of offspring (N_{Off}) is set to the carrying capacity (K, parameter patch_capacity) of the patch or the metapopulation, respectively.

1 : keep number.

$$N_{Off} = N$$

The total number of offspring (N_{Off}) corresponds to the number of adults (N), i.e. the number of individuals is kept constant. Note that a regulation of the patch densities after dispersal can lead to an unwanted continuing reduction of the entire metapopulation size.

2 : fecundity.

$$N_{Off} = Poisson(N_F f)$$

The number of offspring (N_{Off}) depends on the mean fecundity of the females (f) defined by the parameter mean_fecundity.

3 : fecundity simple.

$$N_{Off} = round(N_F f)$$

A simplified version of point 2 assuming that the fecundity is always the same (no fluctuations). This simplification speeds up computation and is acceptable for a wide range of simulation problems.

4 : fecundity binomial.

$$N_{Off} = floor(N_F f) + Binomial(N_F f - floor(N_F f), 1)$$

Similar to point 3, but the rounding is replaced by random rounding with probability equal to the decimal part of the number, i.e. drawing a random number in a binomial distribution with probability equal to the decimal part of the number.

CONFIDENTIAL

5 : fecundity limited.

$$N_{Off} = Poisson(N_F f)$$

$$if(N_{Off} > K)N_{Off} = K$$

Same as point 2, but if the new population size exceeds carrying capacity the population size is down-regulated to carrying capacity.

6 : fecundity simple & limited.

$$N_{Off} = round(N_F f)$$

$$if(N_{Off} > K)N_{Off} = K$$

Same as point 3, but if the new population size exceeds carrying capacity the population size is down-regulated to carrying capacity.

7 : fecundity binomial & limited.

$$N_{Off} = floor(N_F f) + Binomial(N_F f - floor(N_F f), 1)$$

$$if(N_{Off} > K)N_{Off} = K$$

Same as point 4, but if the new population size exceeds carrying capacity the population size is down-regulated to carrying capacity.

8 : logistic regulation.

$$N_{Off} = \frac{NK(1+r)}{N(1+r) - N + K}$$

The total number of offspring (N_{Off}) is logistically regulated following the discrete-time function of Beverton and Holt (1957) and depends therefore on the carrying capacity (K) and on the parameter growth_rate (r).

9 : stochastic logistic regulation.

$$N_{Off} = Poisson(\frac{NK(1+r)}{N(1+r) - N + K})$$

Same as point 8, but the computation of the total number of offspring has a stochastic component.

CONFIDENTIAL

Models and their specific parameters

model	additional parameters
carrying capacity (0)	
keep number (1)	
fecundity (2-7)	mean_fecundity
logistic regulation (8-9)	growth_rate

mean_fecundity [decimal] (temporal)

This parameter specifies the mean female fecundity. The parameter is mandatory (and only used) if the fecundity of the female specifies the number of offspring (i.e. parameter mating_nb_offspring_model set to 2 or 3).

growth_rate [decimal] (temporal)

This parameter is mandatory (and only used) if logistic regulation is used to specify the number of offspring (i.e. parameter mating_nb_offspring_model set to 4 or 5). It specifies the growth rate r of the population using the discrete-time function of Beverton and Holt (1957). Note, that an r of 0 implies constant population size.

sex_ratio_threshold [decimal] (default: "")

By default the sex of an individual is randomly assigned depending on the specified sex ratio (see parameter sex_ratio). If the parameter sex_ratio_threshold is set the sex of an offspring is determined by the phenotype of the first quantitative trait of the offspring. The argument specifies the threshold above which an individual becomes a male. If this parameter is set the parameter sex_ratio is not considered. Note, that when using this parameter the phenotype has to be defined at the offspring stage, thus the parameter quanti_environmental_proportion has to be set to 0. Using this parameter it is possible to simulate sex chromosomes (see chapter 12).

fem_sex_allocation [decimal] (default: 0.5)

This parameter allows defining the female sex allocation proportion to sperm/pollen and ovules, respectively, when hermaphrodites are simulated. The value may range between 0 (only sperm/pollen allocation) and 1 (only ovule allocation). The parameter also allows that the female sex allocation may evolve over time if the female sex allocation is defined by a quantitative trait. For this either the phenotype (Z)

CONFIDENTIAL

or the genotypic value (G) of the quantitative trait may be used. The argument in this case has to be the index of the quantitative trait and has to have the prefix 'G' (genotypic value) or 'Z' (phenotype, e.g fem_se_allocation Z1, if the phenotype of the first quantitative trait specifies the female sex allocation). Note, that if this parameter is used in combination with population growth defined by the female fecundity (parameter mating_nb_offspring_model set to 2 or 3) then the number of females per patch is defined by the sum of the female sex allocation proportions within the patch.

4.2 Statistics

After breeding has taken place it is possible to record summary statistics specified by the parameter stat. Most of the summary statistics can be recorded for offspring and/or adults and for females and/or males. It is also possible to set the frequency (parameter stat_log_time) of the recording. At the end of a simulation the summary statistics are written to a text file. There is the choice to print the summary statistics individually per replicate and/or summed up across replicates (mean and variance across replicates). In this latter case an additional statistic named *alive.rpl* will be added which contains the number of alive replicates, i.e. the number of simulations where the populations did not get extinct. If no summary statistics are computed this event is skipped. Note, that the computation of some summary statistics may be time consuming.

CONFIDENTIAL

stat_save [0-6] (default: 0)

This parameter specifies if the summary statistics should be computed and how they should be dumped to file. The summary statistics may be dumped to file for each specified generation and replicate separately (file "generic_name_stats.txt"), or summary statistics may be summed up across replicates by their mean (file "generic_name_stats.txt") and their variance (file "generic_name_var.txt").

- **0 : All.** Output includes all types of summary statistic (files "generic_name_stats.txt", "generic_name_mean.txt", and "generic_name_var.txt").
- **1 : Detailed.** Output includes only the file containing the summary statistics for each replicate separately (file "generic_name_stats.txt").

Since in this case (in contrast to all other options) it is not necessary to save the statistics over all replicates, the statistics are written to file when they are computed. This means that the internal database to store the statistics is not used, consequently the memory used by quantiNEMO does not increase with each generation and replicate.

2 : Summed up. Output includes the files containing the summary statistics summed up by their mean and variance across replicates (files "generic_name_mean.txt", and "generic_name_var.txt").

3 : Mean. Output includes only the file containing the summary statistics summed up by their mean across replicates (file "generic_name_mean.txt").

4 : Variance. Output includes only the file containing the summary statistics summed up by their variance across replicates (file "generic_name_var.txt").

5 : Median. Output includes only the file containing the summary statistics summed up by their median across replicates (file "generic_name_median.txt").

6 : None. No summary statistics are written. The life cycle event "Statistics" is skipped.

Whenever the summary statistics are output (parameter stat_save not set to 6) a file named "generic_name_legend.txt" containing a small description of the summary statistics is also generated.

stat_log_time [integer] (temporal/default: 1)

This is the time interval at which summary statistics are recorded. The interval must range between 1 and the number of generations. Since the parameter may change over time (temporal parameter) the summary statistics may be computed for any generation:

```
stat_log_time (1 1, 10 10, 100 100)
```

In this example for the first 9 generations the summary statistics are computed every generation, from the tenth until generation 99 they are computed at every tenth generation, and from the generation 100 every hundred generation.

stat_dir [string] (default: "")

This parameter is used to specify a subdirectory within the simulation folder (parameter folder) where the summary statistic files will

CONFIDENTIAL

be stored. If the parameter is not set, the files will be stored in the simulation folder.

stat_filename [string] (default: "")
This parameter is used to specify an individual base filename for the statistics. If not specified the generic base filename will be used (see parameter filename).

stat [string/matrix]
This parameter allows to specify the summary statistics to be computed. The arguments are key words standing for one or multiple summary statistics. If multiple key words are passed they have to be written as a matrix within brackets separated by space. The available key words and their corresponding summary statistics are listed in the corresponding simulation component section in this manual. Summary statistics about the demography may be found in section 7.5, summary statistics about quantitative traits may be found in section 9.7, and summary statistics about neutral markers may be found in section 10.5. If no arguments are specified this event will be skipped. If summary statistics are specified, which cannot be computed since the corresponding component is missing (e.g. F-statistics of neutral loci can only be computed if neutral markers are simulated) a warning will be given.

```
stat {n.fstat
      # q.fstat
      quanti
      adlt.demo}
```

In this example the summary statistics for the key words *n.fstat*, *quanti*, and *adlt.demo* are considered by quantiNEMO, while the key word *q.fstat* is commented out.

param [string/matrix]
This parameter allows to specify the parameter arguments to output together with the statistics. Note, the output is listed only in the detailed statistic file.

stat_NaN [string] (default: "NaN")
This parameter allows to specify a place holder used for the output

CONFIDENTIAL

of the summary statistics for statistics which are not computable. By default the place holder is *NaN*. This default value is correctly read by the statistical package R when a such a file is imported.

sample_all_or_nothing [0,1] (default: 0)
This parameter allows to specify when statistics should be computed (only available for the coalescence simulations (see parameter coalescence)):

0 : when possible. In this case statistics and outputs are generated whenever it is possible, i.e. whenever a patch is enough populated that a given statistic may be computed.

1 : all or nothing. In this case statistics and output are generated only if the entire sampling schema may be applied, i.e. if the patches are enough populated that the specified sampling may be applied. If this is not the case a NaN or a zero depending on the statistic is outputted for all statistics. Note that in such a case the genetic part of the coalescence simulations are omitted, thus the simulation may be much quicker although useless. A specified sampling may not be applied if (and only if) not all specified patches (parameter sampled_patches defined as a matrix) may be sampled or the population size is inferior to the sampling size (parameter patch_sample_size defined in entire numbers (absolute)).

CONFIDENTIAL

4.3 Output

quantiNEMO can also produce files with the raw genetic and phenotypic data. Genotypes of neutral markers and quantitative traits can be dumped to file in the FSTAT (Goudet, 1995) or Arlequin format (Excoffier, 2010). For quantitative traits, the phenotypes may also be written to a file for any generation, sex or age. The selection of the various outputs is done in the corresponding simulation components (neutral genotype see section 10.4, quantitative trait genotype see section 9.6.1, and quantitative trait phenotype see section 9.6.3). If no output is desired this event is skipped. After each output a script may be launched to process the output.

4.4 Aging

This life cycle event simply removes the adults present.

4.5 Regulation offspring

This event performs population regulation before dispersal, i.e. at the offspring stage. This life cycle event allows to regulate the population sizes down to carrying capacity. In fact the regulation should only be used if there is no regulation at the reproduction stage, i.e. if the parameter mating_nb_offspring_model is set to 1 (keep number), 2 (fecundity), or 3 (fecundity stochastic). If the parameter selection_position is set to 2 then selection acts at this stage. In this case the following parameter is ignored and offspring regulation happens following the parameter selection_level.

regulation_model_offspring [0,1] (default: 0)

- **0 : no regulation.** No population size regulation takes place at the offspring stage, i.e. overcrowding can occur.
- **1 : random regulation.** For each patch quantiNEMO regulates the population size down to its carrying capacity if the population size exceeds carrying capacity. Individuals are thereby randomly removed. No regulation takes place if the population size is lower than carrying capacity.

CONFIDENTIAL

4.6 Dispersal

This life cycle event allows the exchange of individuals between populations. The dispersal rates may vary among patches, sexes and time. Several dispersal models are available (see parameter dispersal_model). It is also possible to specify a dispersal matrix, which will have precedence over other dispersal parameters. After dispersal, individuals become adults. By default (if none of the following parameters are set) individuals do not disperse among patches.

dispersal_rate
dispersal_rate_fem
dispersal_rate_mal [decimal/matrix] (temporal/default: 0)

These parameters allow to set the emigration rate (if the argument is between 0 and 1 (1 excluded), or the number of emigrants (if the argument is an integer). If the argument is a single value the dispersal model used depends on the other parameters of this section. But it is also possible to specify the dispersal rate explicitly between each pair of patches (for both directions) if the argument is a matrix. A dispersal matrix has precedence over all other dispersal settings. The matrix must be `patch_number` x `patch_number` in dimensions. Each d_{ij} element of this matrix is the dispersal probability from patch i to patch j, where i specifies the row and j the column of the matrix. Consequently the values in a row must sum up to 1 (if the row do do not sum up to 1 the proportion of residents (diagonal) will be adjusted and a warning returned). The dispersal rates can either be specified for both sexes in general or for each sex separately. If the dispersal rates are sex specific the dispersal rates for both sexes have to be specified and they have to be in the same format (matrix or a single dispersal rate). Sex specific dispersal rates have precedence over a general dispersal rate. Note, that the dispersal matrix has to be fully specified, i.e. the matrix is not adjusted to the number of patches as for other parameters.

dispersal_model [0-4] (default: 0)

The following dispersal models can be specified, if the dispersal rate is a single rate:

0 : Migrant-pool Island model. If the dispersal rate is m and the number of patches is n_p, the probability to disperse to any $n_p - 1$ non-natal patch is $\frac{m}{n_p-1}$ while the probability to stay at home is $1 - m$.

1 : Propagule-pool Island model. In that modified version of the Island model a proportion of emigrants from a patch (parameter dispersal_propagule_prob, φ) disperse to the same non-natal patch. This propagule patch varies among patches and is reassigned at each generation. Each offspring of a patch has a probability $m\varphi$ to migrate to this propagule patch. With probability $\frac{m(1-\varphi)}{n_p-2}$, it will disperse to any patch but its natal or propagule patch. With a probability of $1 - m$ it will stay at home.

CONFIDENTIAL

2 : 1D Stepping-Stone model. In the one dimensional Stepping Stone model patches are placed on a line and migrants can only move to one of the two adjacent patches. If the dispersal rate is m, the probability to disperse to one of the adjacent patches is $m/2$ while the probability to stay at home is $1 - m$. The parameter `dispersal_border_model` allows to specify how to treat the border patches.

3 : 2D Stepping-Stone model. In the two dimensional Stepping-Stone model patches are placed on a grid (or lattice) and migrants can move to 4 or 8 adjacent patches (set by the dispersal_lattice_range parameter below). If the dispersal rate is m, the probability to disperse to one of the adjacent patches is $m/4$ or $m/8$ depending on the the parameter dispersal_lattice_range, while the probability to stay at home is $1 - m$. The parameter dispersal_border_model allows to specify how to treat the border patches and the parameter dispersal_lattice_dims allows to specify the dimensions of the grid.

4 : 2D long range dispersal. A two dimensional lattice is used and dispersal follows a geometric distribution to simulate long range dispersal. The slope of the geometric distribution is given by the parameter **dispersal_long_range_coef**. Note that this model is fully parametrized by the parameter **dispersal_long_range_coef** and does not require the parameter **dispersal_rate**, i.e. this parameter is not considered (expect if it is a matrix, as in this case the dispersal matrix is used to define the migration). For example a value of 0.3 defines the geometric slope and at the same time the emigration rate which is in this case 0.3.

Models and their specific parameters

model	additional parameters
0 : Migrant-pool Island	
1 : Propagule-pool Island	dispersal_propagule_prob
2 : 1D Stepping-Stone	dispersal_border_model
3 : 2D Stepping-Stone	dispersal_border_model / dispersal_lattice_range / dispersal_lattice_dims
4 : 2D long range	dispersal_long_range_coef

dispersal_lattice_range [0,1] (default: 0)

CONFIDENTIAL

This parameter sets the number of neighboring patches used for dispersal. The dispersal probabilities to these adjacent patches are $m/4$ in the first case and $m/8$ in the second. This parameter is only used in the 2D Stepping-Stone model (parameter dispersal_model set to 3).

0 : 4 neighbors. 4 adjacent patches (up, down, left and right)

1 : 8 neighbors. 8 adjacent patches (as before plus the diagonals)

dispersal_border_model [0-2] (default: 0)

This parameter specifies how the patches at a border of the Stepping Stone model should be treated:

0 : Circle/Torus. In the 1D Stepping-Stone model the first and last patches are connected to each other by migration leading to a circle. In the 2D Stepping-Stone model individuals of an edge patch may migrate to the other side leading to a torus (donut world). This means that there are no edges, eliminating any such effects.

1 : Reflective boundaries. The borders are reflective. Dispersers from the border patches cannot move beyond the border. Border cells have thus less cells connected to them and their dispersal probabilities to the adjacent patches are higher (e.g. m for the 1D Stepping-Stone model, $m/3$ (corners $m/2$) for the 2D Stepping-Stone model with four adjacent cells, and $m/5$ (corners $m/3$) for the 2D Stepping-Stone model with eight adjacent cells). No dispersers are lost.

2 : Absorbing boundaries. Dispersers of the border patches are lost if they choose to move beyond the border. The dispersal probabilities of a border patch are not modified.

dispersal_lattice_dims [matrix]

This parameter allows to specify the length and width of the 2D Stepping-Stone lattice (only used when dispersal_model is set to 3). The argument is an integer matrix with two values. The first value stands for the number of rows, and the second value for the number of columns. The product of the two values results in the number of patches and thus must match the parameter patch_number. If the parameter is not set quantiNEMO assumes that the 2D Stepping-Stone lattice is quadratic. If this is not possible due to the number of patches an error is returned.

CONFIDENTIAL

dispersal_propagule_prob [decimal] (temporal/default: 1)
This parameter is only used for the Propagule-pool Island model (parameter dispersal_model set to 1). It specifies the probability that a migrant will move to the propagule-assigned patch, i.e. this is also the proportion of emigrants of a patch which migrate to the same non-natal patch. A probability of 1 means that all emigrants migrate to the same non-natal patch, while a value of 0 means that all emigrants migrate to any patch, but the natal and the propagule patch.

dispersal_long_range_coef
dispersal_long_range_coef_fem
dispersal_long_range_coef_mal [decimal] (temporal/default: 0)
These parameters allow to specify the geometric distribution for the long range dispersal in a two dimensional habitat (parameter **dispersal_model** set to 4) for both sexes together or for each sex separately. The value specifies the slope of the geometric kernel of dispersal in the unit of patches. At the same time the dispersal rate is specified, thus using this dispersal model the parameter **dispersal_rate** will be ignored, expect if the argument of this parameter **dispersal_rate** is a matrix, as in this case the dispersal matrix is used and not the dispersal kernel. Note that a value of 0.3 specifies the slope of the geometric kernel, but also specifies the emigration rate which is 0.3.

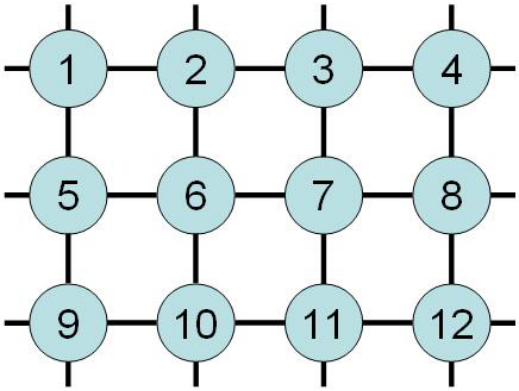
CONFIDENTIAL

Example
The following example defines a metapopulation of 12 patches arranged in a two dimensional grid shown below:

```
patch_number          12
dispersal_model       3
dispersal_lattice_dims {4 3}
dispersal_rate        0.1$
```

4.6.1 Density dependent dispersal rate

By default the dispersal rate is not influenced by the population size. The following parameters allow to define a generalized logistic function (Richards, 1959) to specify a relationship between the dispersal rate and the population density (population size divided by the carrying capacity). The generalized



logistic function defines a factor (f) which is then multiplied with the dispersal rate (parameter dispersal_rate):

$$f = min + \frac{max - min}{(1 + s * e^{r(D_{r_{Max}} - D)})^{1/s}}$$

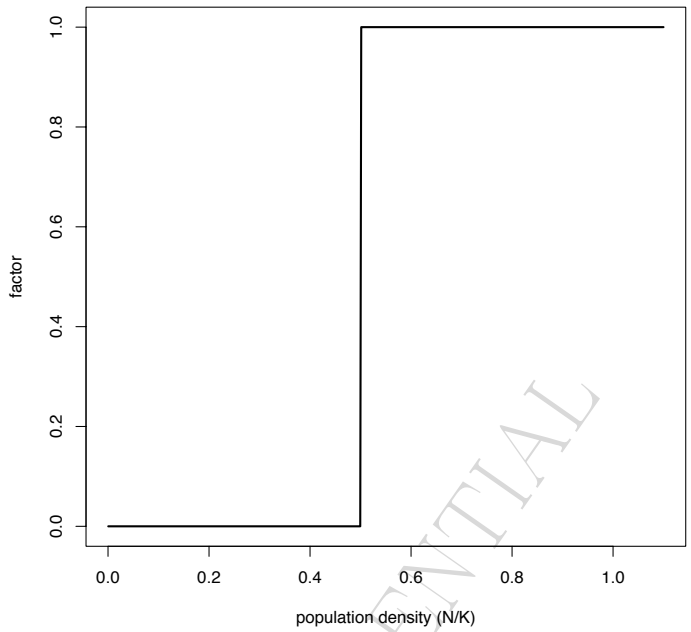
Where min is the lower asymptote (parameter dispersal_min), max is the upper asymptote (parameter dispersal_max), r is the growth rate (parameter dispersal_growth_rate), $D_{r_{Max}}$ is the population density with the maximal slope (parameter dispersal_max_growth), D is the current population density (N/K), and s defines the symmetry of the curve (parameter dispersal_symmetry).

In this figure the growth rate of the curve has a very high value of $1e^4$ (default value of parameter dispersal_growth_rate). This results in a sharp change of the factor from the lower asymptote (parameter dispersal_k_min) to the upper asymptote (parameter dispersal_k_max): Below the threshold (parameter dispersal_max_growth), i.e. when the population density is low no migration occurs since the factor is zero. If the population size is larger than the half of the carrying capacity the patch sends emigrants at the set dispersal rate (parameter dispersal_rate) since the factor is 1.

dispersal_k_min [decimal] (temporal/default: 0)
This parameter specifies the lower asymptote of the slope.

dispersal_k_max [decimal] (temporal/default: 1)
This parameter specifies the upper asymptote of the slope.

CONFIDENTIAL



CONFIDENTIAL

dispersal_k_max_growth [decimal] (temporal/default: -1)
This parameter specifies the population density with the maximal change.

dispersal_k_growth_rate [decimal] (temporal/default: $1e^4$**)**
This parameter specifies the slope of the curve. The high default value of $1e^4$ implies that the slope is more or less vertically, i.e. that the change between lower and higher asymptote is instantaneously. Thus the parameter dispersal_k_max_growth serves as threshold of the population density below which the factor is set to the value of the parameter dispersal_k_min. Above the threshold the factor is set to the value of the parameter dispersal_k_max.

dispersal_k_symmetry [decimal] (temporal/default: 1)
This parameter specifies the symmetry of the curve. The default value of 1 results in a symmetric curve.

4.7 Regulation adults

This event performs population regulation after dispersal, i.e. at the adult stage. Due to asymmetric dispersal some patches may be overcrowded. This life cycle event allows to regulate the population sizes down to carrying capacity. In fact the regulation should only be used if there is no regulation at the reproduction stage, i.e. if the parameter mating_nb_offspring_model is set to 1 (keep number), 2 (fecundity), or 3 (fecundity stochastic). If the parameter selection_position is set to 3 then selection acts at this stage. In this case the following parameter is ignored and offspring regulation happens following the parameter selection_level.

regulation_model_adults [0,1] (default: 0)

- **0 : no regulation.** No population size regulation takes place at the adults stage, i.e. overcrowding can occur.
- **1 : random regulation.** For each patch quantiNEMO regulates the population size down to its carrying capacity if the population size exceeds carrying capacity. Individuals are thereby randomly removed. No regulation takes place if the population size is lower than carrying capacity.

CONFIDENTIAL

4.8 Extinction

This event allows to randomly wipe out populations entirely or partially. The probability that a population goes extinct is specified by the parameter extinction_rate and the parameter extinction_rate_survival specifies how much a population is affected. If the extinction rate is zero this event is skipped.

extinction_rate [decimal/matrix] (temporal/default: 0)
Extinction probability of a patch at each generation. Can be specified for each patch separately.

extinction_rate_survival
extinction_rate_survival_fem

extinction_rate_survival_mal [decimal/matrix] (temporal/default: 0)

These parameters specify how a population is affected when it is hit by an extinction event. By default (value 0) the entire population is wiped out. If the parameter is not 0 then at an extinction event individuals will survive or re-colonize immediately the freed patch depending on the parameter extinction_model. The effect on the population may be defined absolutely if the value is 1 or larger (e.g. 5 individuals survive), relatively if the value is between 0 and 1 (e.g. 10% of the individuals survive), or a combination of relatively and absolutely when specified patch specific using a matrix. The survival rate may be specified for each sex separately or in common. Note that in this later case the values will be used for each sex separately, e.g. if the parameter extinction_rate_survival is set to 0.4, then 40% of the females and 40% of the males will survive if an extinction event happens and if set to 100 and two sexes are simulated 50 females and 50 males will survive.

extinction_model [0-1] (default: 0)

This parameter specifies the model used for the survival/re-colonization (parameter extinction_rate_survival) after an extinction event.

0 : survival. At an extinction the individuals remaining in the focal patch are local individuals, i.e. survivors.

1 : re-colonization island. At an extinction event the population is entirely wiped out, but immediately re-colonized by individuals form other patches following an island model.

CONFIDENTIAL

Chapter 5

Simulation

CONFIDENTIAL

This section describes general parameters of a simulation:

generations [integer]
Number of generations to perform per replicate. This parameter is mandatory, and has no default.

replicates [integer] (default: 1)
Number of replicates to perform per simulation. Replicates are identical simulations (in terms of parametrization), but their outcome may differ due to stochastic events.

working_directory [string/integer] (default: 0)
This parameter allows defining the working directory, the directory where the simulations will be stored in. By default the working directory is the current working directory specified by the OS. The following options are available:

- **0 : current working directory.** The output is stored in the current working directory specified by the OS (default). Normally this is the directory from where the simulation is run.
- **1 : settings file directory.** The output is stored next to the settings file (e.g. *quantinemo.ini*).
- **2 : executable directory.** The output is stored next to the executable.

string : explicitly defined directory. The working directory can also be explicitly defined. If the specified path is relative it is relative to the current working directory given by the OS, but the path can also be absolutely specified.

folder [string] (default: "simulation_yyyy-mm-dd_hh-mm-ss")
This parameter specifies the folder for the output. The default argument of this parameter differs from the rules, as the default folder name is dynamic, i.e. comprises the start time and date of the simulation. The default folder name is "simulation_" with the date and time as suffix in the format "yyyy-mm-dd_hh-mm-ss" (Year-Month-Day_Hour-Minute-Second). This dynamic default folder name allows to store each simulation separately, avoiding that previous outputs are overwritten. If the parameter is set, the passed argument will be used as folder name (i.e. without any addition of the time). In this case it may happen that the new output wants to overwrite previous outputs. The parameter overwrite (see below) allows to specify the rules for overwriting other outputs. If the output should be stored directly in the working directory this parameter has to be listed in the settings file followed by an empty argument "" (for this special parameter an empty argument is not identical to a missing one).

overwrite [integer] (default: 0)
This parameter specifies how present output is treated if there is a danger of overwriting:

0 : no. If the output of the running simulation will overwrite present files, quantiNEMO will ask the user how to proceed (*Do you want to overwrite all the files that use it ? (y(es)/n(o)/s(kip)):*), and will suspend the simulation until the user responds to the question.

1 : yes. Present output is overwritten without asking.

filename [string] (default: "simulation")
This name will be used as the base filename for all outputs, if they are not specified individually. The output file extensions are added to this base filename. If a file is written on a replicate-periodic basis, the replicate number will be added between the base name and the extension, so that the same file is not overwritten periodically. The same is true concerning generation-periodic files (see section 3.10).

CONFIDENTIAL

The base name may include the special expansion character '%' used to build filenames when sequential parameters are present in the input parameter file. See the discussion on sequential parameters in section 3.13.2.

logfile [string] (default: "quantinemo.log")

This is the file name (including the extension) for the log file in which the simulation logs are recorded. This log file is stored next to the executable and records the main information of each simulation, such as the elapsed time for the simulation and the replicates and the time of the start and end of a simulation. By default, quantiNEMO will save all this information in a file named "quantinemo.log".

logfile_type [0-2] (default: 0)

For each simulation a log file is created containing the settings of the simulation. The file is created for each simulation and is stored in the simulation folder (parameter folder). The name of the log file is composed of the base name (parameter filename) and the suffix ".log". Some of the log files are identical to the settings file, i.e. show the macros as they are writen. Other log files have these macros interpreted, i.e. replaced by the values. These latter log files may help to better understand and control the macros:

0 : as input. The file contains the same parameters as the settings file, plus the parameter seed.

1 : minimal. The file contains a minimal set of parameters still able to perform the same simulation. All parameters of the settings file which were set to the default value are not reported.

2 : maximal. The file contains all parameters, including the ones not set in the settings file. For each parameter the type, the default value, if it is a temporal parameter, and a possible range limit is added as comment.

seed [integer/matrix] (default: "")

This parameter specifies the seed which will be used to initialize the random number generator. It is possible to pass a single number as seed in the range of 0 to 4,294,967,295 depending on the computer system (corresponding to an unsigned long in C++). To increase the

CONFIDENTIAL

number of possible seeds it is possible to pass an array of seeds (up to 624) to quantiNEMO if specified as a one dimensional matrix. If the seed is not set the random generator is initialized by the time, i.e. a matrix with two seeds is used. Thereby the first number is the time in seconds since 1.1.1970 (function *time(NULL)* in C++), and the second number is the number of clock ticks elapsed since quantiNEMO started (function *clock()* in C++). Note, no macros are allowed in the seed argument.
If the simulation is multi-threaded, an individual random generator engine is initialized for each thread allowing reproducing the simulation. The first random generator engine is initialized with the passed seed. The subsequent engines are initialized with the same seed, but the last number always incremented by one.

```
seed {2354135 20234510 30345120 456300 50363450}
```

stat_NaN [string] (default: NaN)
This parameter allows specifying the string used in the statistic output for statistics which can not be computed.

threads [integer] (default: 1)
This parameter allows specifying the number of threads to use for multi-threading. Multi-threading is implemented at the level of simulations and replicates. If the number of specified threads exceeds the number of available threads or if the specified thread number is zero, then all available threads of the machine are used.

threaded_lce [binary/integer] (default: 0)
This parameter allows defining which life cycles events should be multi-threaded, if remaining cores are available. By default multi-threading occurs at the simulation and replicate level. In addition multi-threading is implemented in the breeding and dispersal life cycle events. Since multi-threading has a significant overhead in life cycle events it is superior only in computationally intensive life cycle events. If a life cycle event is computationally intensive or not depends entirely on the specified simulation. The argument is of binary form allowing for each life cycle event to define if it should be multi-threaded (1) or not (0) starting with the breeding life cycle event. The argument itself may be directly passed as binary number or as a converted decimal number:

CONFIDENTIAL

00000000 or 0. No life cycle event is multi-threaded.

00000001 or 1. Breeding is multi-threaded.

00100000 or 32. Dispersal is multi-threaded.

00100001 or 33. Breeding and dispersal are multi-threaded.

random_per_replicate [0-1] (default: 0)
This parameter allows specifying how to treat random numbers with multiple replicates:

0 : same. All replicates share the same random number.

1 : individual. All replicates have an individual random number.

all_combinations [0-1] (default: 0)
This parameter allows specifying how multiple sequential parameters are treated:

0 : order. Simulations with the different orders will be simulated (Sim 1: all first arguments; sim 2: all second arguments, ...). If the number of sequential arguments differs, then the y are extended as a vector in R.

1 : full. All possible combinations will be simulated.

preexec_script [string] (default: "")
This parameter allows specifying a script which will be executed before the simulation. The settings file name is passed as unique parameter to the script.

postexec_script [string] (default: "")
This parameter allows specifying a script which will be executed after the simulation. The settings file name is passed as unique parameter to the script.

CONFIDENTIAL

Chapter 6

Coalescence

CONFIDENTIAL

QuantiNEMO has been developed to perform forward in time simulations at the individual level. This allows to simulate realistically highly complex quantitative traits under selection. The drawback of this type of simulation is the huge demand of memory and long computation times. In contrast population-based backward in time simulations based on the coalescence theory are much more efficient in terms of computation time and memory usage, but do not allow to simulate realistic quantitative traits under selection. In order to take advantage of both types of simulations, we have added a layer of coalescence to quantiNemo2. Having both individual and population-based simulations allow for the user to switch easily the simulation mode depending on the type of simulation, keeping all the rest of the simulation/demography identical. This allows for example to explore a demographic history using efficient population-based simulations and then to make the simulations more realistic by switching to individual based simulations. Technically the population-based simulations simulate forward in time the evolution of the entire populations. During this simulation the population sizes and immigration rates for all patches are stored in an internal database. In a second step the database is used to simulate backward in time corresponding coalescence trees on which in a third step the mutations are sprinkled to generate the genetic polymorphism. Most of the parameters of qantiNemo may be used in both simulation modes, but not all. Thus it is not possible to simulate any quantitative traits and thus no selection using population-based simulations. It is also important to note that population sizes in coalescence simulations are effective population sizes, whereas in individual-based simulation census

population sizes. In addition at the current stage the coalescence simulations are limited to hermaphrodite individuals and unlinked neutral markers. In this chapter the coalescence specific parameters are described.

coalescence [0-2] (default: 0)

This parameter allows to switch between forward-in-time simulation of individuals and backward-in-time simulations of populations (coalescence):

0 : individual based. Forward-in-time simulation of individuals. That is the default mode where quantitative traits and selection may be simulated.

1 : population based. Backward-in-time simulations of populations using a coalescence approach. Only able to generate neutral markers.

2 : population based (memory optimized) Same as 1 but temporal database is optimized for minimal memory consumption. At the moment this implementation is not optimized for computation time.

coalescence_model_threshold [decimal] (default: 0.1)

The probability p that a single coalescence event per generation and patch occurs depends on the number of traced lineages n and the population size N (in diploid numbers):

$$p = \frac{n(n-1)}{4N}.$$

This approximation is valid for small numbers of traced lineages compared to the population size ($n << N$). If n in relation to N gets bigger the probability that more than one coalescence event occurs increases and thus the method above is not anymore accurate. Therefore quantiNEMO allows to set a threshold (this parameters) for the probability of a single coalescence even p above which the method used to simulate the coalescence events is changed. In this latter case the coalescence process is simulated realistically, i.e. for each traced lineage a parental gene is drawn randomly. All lineages with an identical parental gene are then coalesced. This method is always valid, however much slower

CONFIDENTIAL

in terms of computation time then the previously mentioned one. By default the value of this parameter, i.e. the threshold, is set to 0.1 which is a acceptable threshold. Note that the probability that a single coalescence event occurs may exceed extensively 1. A value of 0 implies that the coalescent process is always realistically simulated. And ratios above 0 are used to specify the threshold below which a singe coalescence event is assumed and above which the coalescence process is simulated realistically. A parameter value of 1e6 is arbitrarily used to tell quantiNEMO that only single coalescence events should be assumed.

divergence_time [integer] (default: 0)

A coalescence simulation lasts until the most recent common ancestor (MRCA) is reached. Depending on the simulated demography the MRCA may be found within the simulated time, but it can well be that the MRCA of all sampled lineages lays before the start of the demographic simulation. In this case, going backward in time the coalescence simulation is continued behind the demographic simulation assuming no migration between the remaining patches. At the divergence time (parameter **divergence_time** all remaining lineages are then merged to a single patch of specified size (parameter **divergence_pop_size** below). By default the value of the parameter **divergence_time** is 0. If the divergence time is set to less than the number of simulated generations (as the default) the divergence time will be readjusted to the number of generations, thus all remaining lineages are merged to a single patch, when the onset of the demographic simulation is reached.

CONFIDENTIAL

divergence_pop_size [integer] (default: 0)

This parameter allows to define the final population size after the divergence time. Check the parameter **divergence_time** for a description of it. If the argument is 0 (default value) the population size is adjusted to the current total number of individuals. If the specified population size is smaller than the number of traced lineages, then the population size is increased to the smallest possible population size still capable to carry the traced lineages.

6.1 Output

The coalescence simulations allow to output the coalescence trees and also the times to the MRCA.

6.1.1 Trees

The coalescence trees may be dumped to file in the NEXUS format. The branches of the tree may be either scaled by the coalescence times, or by the number of mutations. Several programs allowing to visualize NEXUS files such as for example FigTree developed by Andrew Rambaut www.tree.bio.ed.ac.uk/software/figtree. The NEXUS file has the extension **.tree**.

CONFIDENTIAL

coalescence_save_tree [0-2] (default: 0)
This parameter specifies the output of the coalescence trees.

0 : None. No output is generated.

1 : scaled by coalescence time. The trees are outputted in the NEXUS format and branches are scaled by the coalescence times.

2 : scaled by number of mutations. The topology of the trees is the same as before, but the branches are now scaled by the number of mutations.

An example of a tree file for 3 loci and and 3 sampled diploid individuals:

```
#NEXUS

[Treefile generated by quantiNEMO
    quantiNEMO v1.5.0[May 20 2010; 09:14:19]
    File created the 20-05-2010 09:14:27
    Branches are scaled by the coalescence time
]

Begin trees;

Translate
    1 1a,
    2 1b,
```

```
      3 2a,
      4 2b,
      5 3a,
      6 3b;

  tree LOCUS_1 = ((5: 114, 4: 114): 2, ((1: 59, 6: 59): 2, (3: 13, 2: 13): 48)
  tree LOCUS_2 = (1: 72, ((6: 33, 5: 33): 22, ((2: 3, 4: 3): 2, 3: 5): 50): 17
  tree LOCUS_3 = (((1: 9, 2: 9): 13, 6: 22): 209, ((4: 0, 5: 0): 180, 3: 180):

  End;
```

coalescence_tree_dir [string] (default: "")
This parameter allows to specify the subdirectory where the tree files are stored. This directory has to be specified relative to the simulation folder (parameter folder), and may also contain subdirectories. If not specified (default) the output is stored in the simulation folder (parameter folder).

coalescence_tree_filename [string] (default: "")
This parameter is used to specify an individual base filename for the coalescence trees. If not specified the generic base filename will be used (see parameter filename).

coalescence_tree_script [string] (default: "")
It is possible to launch a script just after the tree file is generated. The argument of the parameter is the file name of the script. The name of the tree file is passed as unique parameter to the script.

CONFIDENTIAL

6.1.2 MRCA

The times to the MRCA may be dumped to file as a simple list. The file has the extension **.mrca**.

coalescence_save_mrca [0-1] (default: 0)
This parameter specifies whether the times to the MRCA are outputted.

0 : None. No output is generated.

1 : Output. The times to the MRCA are outputted. mutations.

An example of such a file for 3 loci is:

```
112
339
813
```

coalescence_mrca_dir [string] (default: "")
This parameter allows to specify the subdirectory where the MRCA files are stored. This directory has to be specified relative to the simulation folder (parameter folder), and may also contain subdirectories. If not specified (default) the output is stored in the simulation folder (parameter folder).

coalescence_mrca_filename [string] (default: "")
This parameter is used to specify an individual base filename for the coalescence MRCA. If not specified the generic base filename will be used (see parameter filename).

coalescence_mrca_script [string] (default: "")
It is possible to launch a script just after the MRCA file is generated. The argument of the parameter is the file name of the script. The name of the MRCA file is passed as unique parameter to the script.

CONFIDENTIAL

6.1.3 Lineages

For the coalescence simulation it is possible to dump the current populated patches, including their population sizes and number of traces lineages to a file for any given generation. The file is generated for each locus separately and has has the ending **_lX.lin**, where X is the locus index. The format is as follows: Each line contains a single generation, i.e. time slice. The line starts with the generation index relative to the start of the simulation. Thus since thes coalescence simulations are proceeded backward in time the generations decrease from top to down of the file. Positive numbers are time slices where a demographic simulation is available and negative numbers for time slices before (forward in time) the demographic simulation. The generation number is followed by the total number of currently traced lineages. Then each populated patch is listed using three numbers, the patch index, the current population size, and the current number of traced lineages. Since over time

some patches may be colonized or freed the number of columns per line may change between generations.

coalescence_save_lineages [0-1] (default: 0)
This parameter specifies whether the patch stages are outputted.

0 : None. No output is generated.

1 : Output. The patch stages are outputted.

coalescence_lineages_logtime [integer] (temporal/default: 1)
This parameter allows to specify at which generations DURING the period of the demographic simulation the patch stages are dumped to file.

coalescence_lineages_logtime2 [integer] (temporal/default: 1)
This parameter allows to specify at which generations BEFORE (forward in time) the demographic simulation the patch stages are dumped to file.

coalescence_save_lineages_dir [string] (default: "")
This parameter allows to specify the subdirectory where the lineages files are stored. This directory has to be specified relative to the simulation folder (parameter folder), and may also contain subdirectories. If not specified (default) the output is stored in the simulation folder (parameter folder).

coalescence_save_lineages_filename [string] (default: "")
This parameter is used to specify an individual base filename for the lineages files. If not specified the generic base filename will be used (see parameter filename).

coalescence_lineages_script [string] (default: "")
It is possible to launch a script just after the patch stage file is generated. The argument of the parameter is the file name of the script. The name of the patch stage file is passed as unique parameter to the script.

CONFIDENTIAL

6.1.4 Population sizes

For the coalescence simulation it is possible to dump the current populated sizes to file. The format of the file is the generation time followed by the population sizes for the patches of choice.

coalescence_save_pop_sizes [0-1] (default: 0)
This parameter specifies whether the population sizes are outputted.

0 : None. No output is generated.

1 : Output. The population sizes are outputted.

coalescence_pop_sizes_logtime [integer] (temporal/default: 1)
This parameter allows to specify at which generations the population sizes are dumped to file.

coalescence_save_pop_sizes_dir [string] (default: "")
This parameter allows to specify the subdirectory where the population sizes files are stored. This directory has to be specified relative to the simulation folder (parameter folder), and may also contain subdirectories. If not specified (default) the output is stored in the simulation folder (parameter folder).

coalescence_save_pop_sizes_filename [string] (default: "")
This parameter is used to specify an individual base filename for the population sizes files. If not specified the generic base filename will be used (see parameter filename).

coalescence_pop_sizes_of_patch [integer/matrix] (default: "")
This parameter allows defining the patches for which the population sizes should be outputted using a matrix with the patch IDs. By default the population sizes of all patches is outputted.

coalescence_pop_sizes_script [string] (default: "")
It is possible to launch a script just after the population sizes file is generated. The argument of the parameter is the file name of the script. The name of the population size file is passed as unique parameter to the script.

CONFIDENTIAL

6.2 Summary statistics

The summary statistics listed in the table below are available for coalescence simulations. The column Stat name contains the name of the summary statistic used to specify which summary statistics are computed (parameter stat, for details see section 4.2). These names appear also in the output file. The column Description contains a short description of the summary statistic.

The summary statistic name (column Stat name) may be used to specify the summary statistic to be computed (e.g. mrca.mean). Similar summary statistics (within a thematic group) may be obtained at once using the name within square brackets after the group title (e.g. mrca). Using this group statistic name all summary statistics of the thematic group marked with a star (*) will be computed.

CONFIDENTIAL

Table 6.1: Summary statistics available for coalescence simulations

Stat name	Description
MRCA [mrca]	
mrca.mean	mean time to the MRCA*
mrca.var	variance of the time to the MRCA*

Table 6.1: Summary statistics available for coalescence simulations continued

Chapter 7

Metapopulation

CONFIDENTIAL

7.1 Dimensions

This section describes general parameters about the metapopulation:

patch_number [integer] (default: 1)
This parameter specifies the number of patches in the metapopulation.

patch_capacity
patch_capacity_fem
patch_capacity_mal [integer/matrix] (temporal)
These parameters allow to set the carrying capacities of the patches. The carrying capacity of a patch is the maximal number of individuals that a patch may support. The carrying capacities may vary among sexes, patches, and time. The carrying capacities have to be specified either for each sex separately (parameters patch_capacity_fem and patch_capacity_mal), or for both sexes together (parameter patch_capacity). In the first case both sex specific parameters have to be set if two sexes are simulated. In the latter case the carrying capacities for females and males are assumed to be identical, i.e. the carrying capacity of females and males is *patch_capacity*/2. If hermaphrodites are simulated the parameters patch_capacity and patch_capacity_fem are identical. In case all three parameters are set, only the sex specific parameters will be used as they are more informative. To set the carrying capacities individually for each patch a matrix is needed. The matrix is adjusted to

the number of patches if necessary (see section 3.4). If all patches have the same carrying capacities a single number as argument is sufficient to specify the carrying capacities. If the initial population sizes (parameter patch_ini_size) are not specified the simulation will start with all the populations set at carrying capacity. Note, using a two dimensional matrix with two columns allows addressing directly a patch and thus specifying carrying capacity explicitly for this patch (e.g. *{{patchID value}{patchID value}...}*), while all not specified patches will have a carrying capacity of 0.

Examples

The following example defines a metapopulation of 5 patches each with a carrying capacity of 100 individuals:

```
patch_number    5
patch_capacity  100
```

If the carrying capacities vary between patches the following definition with a matrix may be used:

```
patch_number    5
patch_capacity  {100 200 300 400 500}
```

In the following example, however the matrix of the parameter patch_capacity will be adjusted by quantiNEMO to meet the number of patches. This will result in 10 patches with the carrying capacities 100, 200, 300, 400, 500, 100, 200, 300, 400, and 500.

```
patch_number    10
patch_capacity  {100 200 300 400 500}
```

If the carrying capacities are defined for each sex separately both sex specific parameters must be present if two sexes are simulated:

```
patch_number        5
patch_capacity_fem  {100 200 300 400 500}
patch_capacity_mal  {200 400 600 800 1000}
```

In the following example the carrying capacity is specified with too many parameters. As the sex specific parameters are more informative, they have precedence over the global setting of carrying capacities (1000) which will therefore be ignored:

CONFIDENTIAL

```
patch_number          5
patch_capacity        1000
patch_capacity_fem    {100 200 300 400 500}
patch_capacity_mal    {200 400 600 800 1000}
```

patch_friction
patch_friction_fem
patch_friction_mal [decimal/matrix] (temporal/default: 1)

These parameters allow to set the friction of the patches. The friction of a patch influences the dispersal. The friction is currently a factor multiplied with the dispersal rate (see parameter **dispersal_rate**), thus a friction of 1 means that the resulting emigration rate corresponds to the specified dispersal rate. A friction below 1 reduces the effective emigration rate (somehow contra intuitive...) until 0 where no emigration at all occurs. A friction above one favours the emigration rate. Note, using a two dimensional matrix with two columns allows addressing directly a patch and thus specifying friction explicitly for this patch (e.g. *{{patchID value}{patchID value}...}*), while all not specified patches will have a default friction of 1.

CONFIDENTIAL

7.2 Initialization

This section allows to specify how the metapopulation is initialized. The population size of the patches may be set in three different ways. Also the genotypes of the individuals of the initial populations may be set differently. By default, i.e. if no special parameters of this section are defined, the initial population size of each patch corresponds to its carrying capacity (see parameter **patch_capacity** above). If the initial population sizes deviate from the carrying capacities the parameter **patch_ini_size** allows to set the initial population size for each patch separately. Finally, it is possible to define the genotypes of each individual explicitly using an FSTAT file (Goudet, 1995). By doing this also the initial population sizes are defined. For details please have a look at the parameters **quanti_ini_genotypes** and **ntrl_ini_genotypes** in their appropriate chapter about quantitative traits (chapter 9) and neutral markers (chapter 10), respectively. If a genotype file is present the initial population sizes are defined using this file. Note,

that it is possible to resume a simulation using the genotype files.

patch_ini_size
patch_ini_size_fem
patch_ini_size_mal [integer/matrix]

These parameters allow to set the initial population sizes of the patches, i.e. the populations sizes present at the beginning of the simulation. These parameters are optional. If none of these parameters is set, i.e. the initial population sizes are not set, the simulation will start with all the population sizes set at carrying capacity. The initial population sizes may vary among sexes, and patches. The initial population sizes have to be specified either for each sex separately (parameters patch_ini_size_fem and patch_ini_size_mal), or for both sexes together (parameter patch_ini_size). In the first case both sex specific parameters have to be set if two sexes are simulated. In the latter case the initial population sizes for females and males are assumed to be identical, i.e. the initial population size of females and males is $patch\_ini\_size/2$. If hermaphrodites are simulated the parameters patch_ini_size and patch_ini_size_fem are identical. In case all three parameters are set the sex specific parameters will be used as they are more informative. To set the initial population sizes individually for each patch a matrix is needed. The matrix is adjusted to the number of patches if necessary (see section 3.4). If all patches have the same initial population sizes a single number as argument is sufficient to specify the initial population sizes. Note, using a two dimensional matrix with two columns allows addressing directly a patch and thus specifying the initial population size for this patch (e.g. *{{patchID value}{patchID value}...}*), while all not specified patches will have an initial population size of 0.

CONFIDENTIAL

7.3 Selection pressure

Currently there are two ways to define the selection pressure. The parameter selection_pressure_definition allows to switch among them. Phenotypes for quantitative traits (see section 9) may be under selection. Selection pressures may vary among quantitative traits, sexes, patches, and time. To specify the

selection pressure individually for quantitative traits and patches, matrices may be used. They are adjusted to the number of quantitative traits and to the number of patches if needed (see section 3.4). If a parameter does not change among quantitative traits and patches, a single value may be used as argument. Selection pressures have to be specified either for each sex separately (parameters with the suffix "_fem" for females and "_mal" for males), or for both sexes together (parameters without a suffix). In the first case both sex specific parameters have to be set if two sexes are simulated. In the latter case the selection pressure of females and males are assumed to be identical. Each row of the matrix corresponds to a quantitative trait, each column to a patch:

```
{ {patch_1 patch_2 . . . patch_n}    # trait 1
  {patch_1 patch_2 . . . patch_n}    # trait 2
  ...
  {patch_1 patch_2 . . . patch_n} }  # trait m
```

quantiNEMO supports three types of selection, stabilizing selection, directional selection and selection based on a fitness landscape. The type of selection may vary among quantitative traits, but not among patches and sexes. However, selection pressure of the same type of selection may vary among patches, quantitative traits, sexes, and may also change through time. This flexibility allows for example to simulate a dynamic environment such as for global warming.

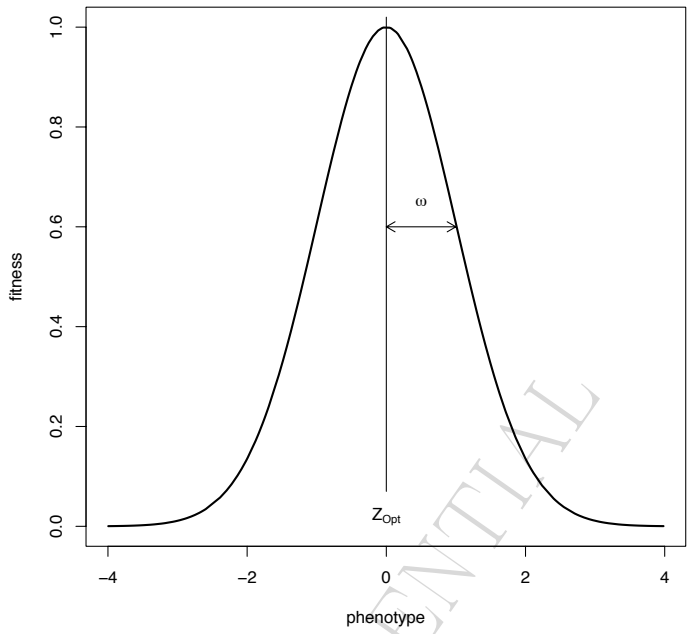
CONFIDENTIAL

7.3.1 Stabilizing selection

Stabilizing selection may act on the phenotype (P) of quantitative traits. The selection pressure is defined by the parameters patch_stab_sel_optima (Z_{Opt}) and patch_stab_sel_intensity (ω). The fitness (W) of a quantitative trait is computed using the following standard Gaussian function for stabilizing selection:

$$W = e^{\left(-\frac{\left(P-Z_{Opt}\right)^2}{2\omega^2}\right)}$$

patch_stab_sel_optima



CONFIDENTIAL

patch_stab_sel_optima_fem
patch_stab_sel_optima_mal [decimal/matrix] (temporal/default: 0)
These parameters allow to set the selection optimum z_{Opt} for each patch and quantitative trait.

patch_stab_sel_intensity
patch_stab_sel_intensity_fem
patch_stab_sel_intensity_mal [decimal/matrix] (temporal/default: 1)
These parameters allow to set the selection intensity ω for each patch and quantitative trait. A small value results in a strong selection pressure, whereas a large value results in a weak selection pressure.

patch_stab_sel_optima_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the selection optimum varies between generations (e.g. annual

fluctuations of the mean temperature). By default the local selection optimum does not vary.

patch_stab_sel_intensity_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the selection intensity varies at each generation (e.g. annual fluctuations of the mean temperature). By default the local selection intensity does not vary.

Example

```
patch_number              2
quanti_nb_trait           3
patch_stab_sel_optima     {{-0.1   0.1}
                           { 0.2   0.2}
                           {-0.3   0.3}}
patch_stab_sel_intensity  1
```

In this example the environment consist of two patches with varying selection pressures. Three quantitative traits are simulated. The first trait has a selection optimum at -0.1 in patch 1 and at 0.1 in patch 2. The selection optimum of the second trait is the same in both patches (0.2). The third trait has an optimum at -0.3 in patch 1 and at 0.3 in patch 2. The intensity of the selection is identical for all three traits and in both patches.

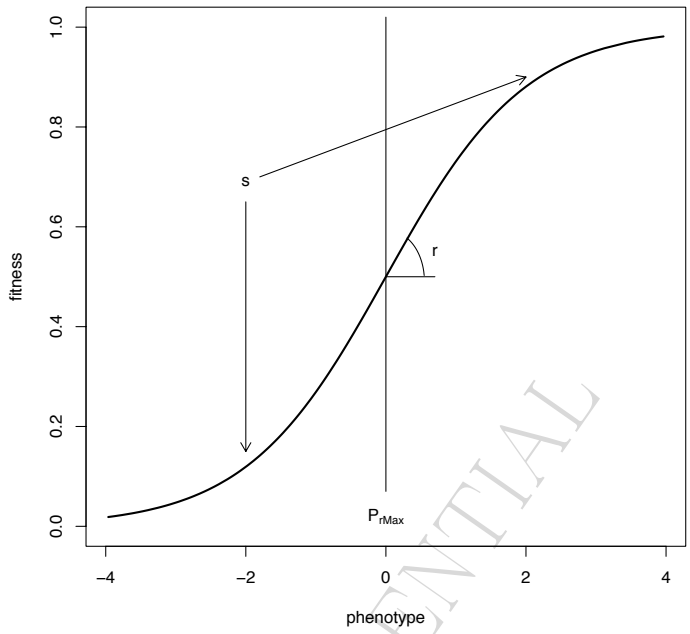
CONFIDENTIAL

7.3.2 Directional selection

Directional selection may act on quantitative traits. The fitness (W) of a quantitative trait is computed using the following generalized logistic function (Richards, 1959):

$$W = min + \frac{max - min}{(1 + s * e^{r(P_{r_{Max}} - P)})^{1/s}}$$

Where min is the lower asymptote (parameter patch_dir_sel_min), max is the upper asymptote (parameter patch_dir_sel_max), r is the growth rate (parameter patch_dir_sel_growth_rate), $P_{r_{Max}}$ is the phenotype with the



maximal slope (parameter patch_dir_sel_max_growth), and s defines the symmetry of the curve (parameter patch_dir_sel_symmetry; the curve is symmetric by default (value 1)).

CONFIDENTIAL

patch_dir_sel_min
patch_dir_sel_min_fem
patch_dir_sel_min_mal [decimal/matrix] (temporal/default: 0)
: These parameters allow to set the lower asymptote of the selection curve for each patch and quantitative trait.

patch_dir_sel_max
patch_dir_sel_max_fem
patch_dir_sel_max_mal [decimal/matrix] (temporal/default: 1)
: These parameters allow to set the upper asymptote of the selection curve for each patch and quantitative trait.

patch_dir_sel_growth_rate
patch_dir_sel_growth_rate_fem

patch_dir_sel_growth_rate_mal [decimal/matrix] (temporal/default: 1)
These parameters allow to set the slope of the selection curve for each patch and quantitative trait. If the argument is positive larger phenotypes have a higher fitness, while if negative smaller phenotypes have a higher fitness.

patch_dir_sel_max_growth
patch_dir_sel_max_growth_fem
patch_dir_sel_max_growth_mal [decimal/matrix] (temporal/default: 0)
These parameters allow to set the phenotype with the maximal growth.

patch_dir_sel_symmetry
patch_dir_sel_symmetry_fem
patch_dir_sel_symmetry_mal [decimal/matrix] (temporal/default: 1)
These parameters allow to set the symmetry of the curve. The default value of 1 results in a symmetric slope.

patch_dir_sel_min_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the lower asymptote of the selection curve varies at each generation (e.g. annual fluctuations of the mean temperature). By default the lower asymptote of the selection curve does not vary.

patch_dir_sel_max_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the upper asymptote of the selection curve varies at each generation (e.g. annual fluctuations of the mean temperature). By default the upper asymptote of the selection curve does not vary.

patch_dir_sel_growth_rate_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the selection slope varies at each generation (e.g. annual fluctuations of the mean temperature). By default the local selection slope does not vary.

CONFIDENTIAL

patch_dir_sel_max_growth_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the phenotype with maximal growth varies at each generation (e.g. annual fluctuations of the mean temperature). By default the local phenotype with maximal growth does not vary.

patch_dir_sel_symmetry_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the symmetry of the curve varies at each generation (e.g. annual fluctuations of the mean temperature). By default the symmetry of the slope does not vary.

Example

```
patch_dir_sel_growth_rate 1
patch_dir_sel_max_growth  0
patch_dir_sel_symmetry    1
```

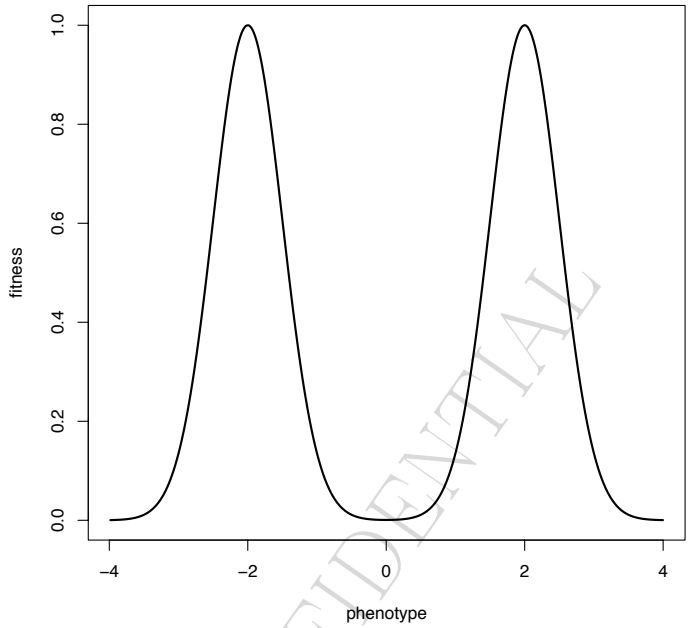
In this example the selection pressure for all patches and quantitative traits are identical and set to the default values. The specified directional selection pressure favors larger phenotypes (parameter patch_dir_sel_growth_rate is positive). This means that individuals with larger phenotypes have on average higher fitnesses and thus higher reproductive successes.

CONFIDENTIAL

7.3.3 Fitness landscape

Specifying a fitness landscape allows to define any selection pressure for a quantitative trait. A fitness landscape is in principle defined by a 2 dimensional matrix containing phenotype values and their corresponding fitness values. There is no limit to the number of phenotype-fitness value pairs. For practical reasons the two vectors phenotype values (parameter patch_phenotype_landscape) and fitness values (parameter patch_phenotype_landscape) are separately passed to quantiNEMO. The fitness value of any phenotype within the specified range of phenotypes is linearly interpolated and if the phenotype lies outside of the specified range of phenotypes the resulting fitness value will result in the fitness value of the smallest or the largest

phenotype, respectively. By default all phenotypes will result in a fitness of 1.



CONFIDENTIAL

The fitness landscape is defined by two tightly linked parameters:

patch_phenotype_landscape
patch_phenotype_landscape_fem
patch_phenotype_landscape_mal [decimal/matrix] (temporal/default: 0)

These parameters allow to specify an array of phenotypes for which the corresponding fitness is defined using the parameter patch_fitness_landscape. The phenotype values are either specified by a single value (similar to the default value) specifying a unique phenotype for all patches and quantitative traits together (all phenotypes will result in the same fitness), or by a one dimensional array (1D matrix) specifying the same fitness landscape for all patches and quantitative traits together, or by a 3D matrix specifying the fitness landscape separately for each patch, and/or quantitative trait. Since the two parameters patch_fitness_phenotype

and patch_fitness_fitness are tightly linked, i.e. specify the phenotypes and their corresponding fitness values, the architecture of the two parameters has to be the same, i.e. has the same number of values for a specific quantitative trait and patch.

patch_fitness_landscape
patch_fitness_landscape_fem
patch_fitness_landscape_mal [decimal/matrix] (temporal/default: 1)

These parameters allow to specify the fitnesses corresponding to the phenotypes defined with the parameter patch_phenotype_landscape, in the same order as the phenotypes are defined in the parameter patch_fitness_phenotype. The fitness values are either specified by a single value (similar to the default value) specifying a unique fitness for all patches and quantitative traits together, or by a one dimensional array (1D matrix) specifying the same fitness landscape for all patches and quantitative traits together, or by a 3D matrix specifying the fitness landscape separately for each patch, and/or quantitative trait. Since the two parameters patch_fitness_phenotype and patch_fitness_fitness are tightly linked, i.e. specify the phenotypes and their corresponding fitness values, the architecture of the two parameters has to be the same, i.e. has the same number of values for a specific quantitative trait and patch.

CONFIDENTIAL

Example

```
patch_phenotype_landscape {-4.0 -3.2 -2.4 -1.6 -0.8  0.0
                            0.8  1.6  2.4  3.2  4.0}
patch_fitness_landscape   {0.00 0.05 0.72 0.72 0.05 0.00
                           0.05 0.72 0.72 0.05 0.00}
```

In this example the selection pressure for all patches and quantitative traits are identical. The specified fitness landscape corresponds to the above shown figure, however the resolution of the specification is reduced in the sown example. The binomial landscape is the sum of two stabilizing selections with optima set to -2 and 2 and an intensity of 0.5.

7.3.4 Selection coefficient

quantiNemo2 supports the simulation of bi-allelic loci where the selection pressure is defined by a selection coefficient. In this simplified, but widespread model, the genotype is directly mapped to a fitness:

Genotype	Fitness	Fitness extended
AA	1	F
Aa	1 - hs	F - hs
aa	1 - s	F - s

Where s is the selection coefficient (parameter patch_coef_sel), h the dominance factor (parameter quanti_dominance_mean) with the model (parameter quanti_dominance_model) set to 1 (h-model, has to be set explicitly!), and F is the fitness factor for the wild type genotype AA set by parameter patch_coef_sel_AA. This latter parameter is normally not used therefore it is listed in the column âĂIJFitness extendedâĂİ. Using this selection model with several loci it might be useful to set multiplicative effects among loci of the quantitative trait using the parameter quanti_multiplicative_model.

CONFIDENTIAL

patch_coef_sel
patch_coef_sel_fem
patch_coef_sel_mal [decimal/matrix] (temporal/default: 0)
: These parameters allow to set the selection coefficient s for each patch and quantitative trait/locus. The default value of 0 will result in equivalent fitness for both alleles, and thus no selection acts.

patch_coef_sel_AA
patch_coef_sel_AA_fem
patch_coef_sel_AA_mal [decimal/matrix] (temporal/default: 1)
: These parameters allow to set the fitness for the wild type genotype AA for each patch and quantitative trait/locus. By default it is 1.

7.3.5 Multiple traits with varying types of selection

A special case arises if multiple quantitative traits are simulated with varying types of selection. The problem is how to specify the individual selection pressures. This is a more technical problem which I would like to illustrate

here. First, quantiNEMO investigates which types of selection will be simulated based on the settings file. Then, quantiNEMO sets for each simulated type of selection the corresponding parameters assuming one selection type after the other that all quantitative traits have the same type of selection: assuming for example first that all quantitative traits are under stabilizing selection, then in a second step assuming that all quantitative traits are under directional selection. This detail is important to understand since this makes it clear how a matrix is treated, i.e. how a matrix is expanded if needed. Of, course finally the selection pressure parameters are only set where needed, i.e. if the type of selection for a given quantitative trait requires the parameter. In other words the matrix of a selection pressure has to have the number of rows of the total number of traits, and not only of the traits with the corresponding selection pressure (this is controlled by quantiNEMO returning a warning if not met). If multiple quantitative traits are simulated with varying selection pressures it is handy to use the row indicator for the rows of the matrix. Example:

```
patch_number 3
quanti_nb_trait 5
quanti_selection_model 0
quanti_selection_model_3 1
patch_stab_sel_optima      {{1: 1 2 3}{2: 2 3 4}}
patch_dir_sel_growth_rate {{3: 4 5 6}{4: 7 8 7}{5: 2 3 2}}
```

In this example the first two quantitative traits are under stabilizing selection, whereas the three last quantitative traits are under directional selection. Using the row indicator it is possible to set the selection pressure directly for the required quantitative trait. However the following parameterization is equivalent to the upper one:

```
patch_number 3
quanti_nb_trait 5
quanti_selection_model 0
quanti_selection_model_3 1
patch_stab_sel_optima      {{1 2 3}{2 3 4}{9 9 9}{9 9 9}{9 9 9}}
patch_dir_sel_growth_rate {{9 9 9}{9 9 9}{4 5 6}{7 8 7}{2 3 2}}
```

In the example above the rows consisting of the number nine are read but then not taken into account, since the rows do not correspond to the correct quantitative traits. Caution if you are using matrix expansions since this may lead to unwanted configurations as shown below:

CONFIDENTIAL

```
patch_number 3
quanti_nb_trait 5
quanti_selection_model 0
quanti_selection_model_3 1
patch_stab_sel_optima      {{1 2 3}}
patch_dir_sel_growth_rate {{4 5 6}{2 3 7}{2 8 3}}
```

In this example the growth rate are as follows:

```
1. trait: stabilizing selection optima: {1 2 3}
2. trait: stabilizing selection optima: {1 2 3}
3. trait: directional selection growth rate: {2 8 3}
4. trait: directional selection growth rate: {4 5 6}
5. trait: directional selection growth rate: {2 3 7}
```

Maybe this behavior was desired, but it could also well be that one anticipated the following specification which is wrong:

```
1. trait: stabilizing selection optima: {1 2 3}
2. trait: stabilizing selection optima: {1 2 3}
3. trait: directional selection growth rate: {4 5 6}
4. trait: directional selection growth rate: {2 3 7}
5. trait: directional selection growth rate: {2 8 3}
```

Why is this not the case? Assume that all quantitative traits are under directional selection. Thus the matrix of the growth rate has to be repeated leading to the following full matrix: {{4 5 6}{2 3 7}{2 8 3}{4 5 6}{2 3 7}}. This matrix expansion leads to a warning indicating that the number of rows is not an entire subset of the number of quantitative traits. Based on this matrix it is now obvious that the growth rate of the third quantitative trait (thus the first trait under directional selection) has the growth rates {2 8 3} and not {4 5 6}.

7.4 Sampling

By default the outputs generated by quantiNEMO (summary statistics, genotypes, phenotypes, and genotypic values) are computed/outputted considering all individuals and popualtions. This chapter describes parameters allowing to constrain the considered populations and/or individuals. The parameter **sampled_patches** allows to make a pre-selection on the patches,

CONFIDENTIAL

whereas the parameter **patch_sample_size** allows to define for each patch and/or sex the sampled number of individuals. It is therefore possible to specify the samples just using the parameter **patch_sample_size**.

sampled_patches [integer/matrix] (default: 0)

The parameter **sampled_patches** allows to define a sampling schema for patches, where ALL individuals are sampled. There are different methods to define the sampled patches:

single number. If the argument is a single number the patches to sample are drawn randomly. In this case the passed number specifies the total number of patches to sample. Patches are drawn randomly for each replicate, but remain the same during a simulation. A special case is 0 as argument (default value). In this case all patches are sampled.

matrix. Using a one dimensional matrix as argument allows to define explicitly the patches to sample. Please note that for this parameter a matrix has to be defined fully, i.e. all patches to sample have to be listed in the matrix and a single number is not expanded to a matrix. Note, that in contrast to the normal matrix behaviour the two arguments *20* and *{20}* are not identical. While the former one is considered as a single number defining the total number of sampled patches, the second one is a matrix and defines the patch to be sampled, i.e. patch 20.

patch_sample_size
patch_sample_size_fem
patch_sample_size_mal [decimal/matrix] (temporal/default: NaN)

These parameters allows to define for each patch and/or sex the sample size or sample proportion. Since the sampling schema may change over time the summary statistics are computed and outputted for all populations sampled at any time during the simulation even if at a given time the sample size is zero. These parameters here overwrite any settings specified by the parameter **sampled_patches**.If the sampling is sex specific both sex specific parameters have to be set. The sampling for a single patch may be defined in the following ways:

CONFIDENTIAL

proportion. A decimal number (number between 0 and 1, exclusive 1) may be used to define the sample size relatively to the population size.

absolute. The sample size may be defined in an absolute number (1 or larger). If the absolute sample size exceeds the population size the entire population is sampled.

NaN. This is the default argument and allows to sample all individuals of a population.

The sampling schema for all patches may be defined as follows:

single number. All patches have the same sampling schema, i.e. sampling proportion or sampling number.

1D matrix. The sample sizes/proportions of each patch may be set using a 1D matrix. The setting of this matrix behaves as the matrix for the parameter **patch_capacity** or **patch_ini_size**, i.e. the matrix size is adjusted if needed. Within a matrix the individual patch sample sizes may vary between relative, absolute, or *NaN* definitions.

2D matrix. A 2D matrix allows to specify directly the sample sizes/proportions for a given number of patches. The numbers of columns has to be 2, where the first column contains the patch ID and the second column the corresponding sampling number/proportion. Within a matrix the individual patch sample sizes may vary between relative, absolute, or *NaN* definitions. All not specified patches will not be sampled, i.e. have a sample size of 0.

CONFIDENTIAL

7.5 Summary statistics

The summary statistics listed in the table below are available for the demography of the populations. The column Stat name contains the name of the summary statistic used to specify which summary statistics are computed (parameter stat, for details see section 4.2). These names appear also in the output file. The column Description contains a short description of the summary statistics.

Some of the summary statistics are available for adults and offspring (indicated by (adlt/off)). To obtain a certain summary statistic for adults the prefix adlt. has to be added to the summary statistic name (e.g. adlt.allnb), respectively the prefix off. to obtain the summary statistic for offspring (e.g. off.allnb).

Some of the summary statistics are computed for each patch separately. These statistics are characterized by a suffix "_p" to the Stat name and by the words *(computed for each patch)* in the description of the statistic. Other summary statistics are computed for pairwise combinations of patches. These statistics are characterized by a suffix "_pair" to the Stat name and by the words *(all pairwise combinations computed)* in the description of the statistic. Note, that depending on the number of patches such a statistic may lead to a huge amount of output and a higher computation time. The names of such summary statistics in the output have the suffix "_pX" and "_pX-Y", respectively. Where X and Y are the index of the patches (starting with 1).

The summary statistic name (column Stat name) may be used to specify the summary statistic to be computed (e.g. adlt.nbInd). Similar summary statistics (within a thematic group) may be obtained at once using the name within square brackets after the group title (e.g. adlt.demo). Using this group statistic name all summary statistics of the thematic group marked with a star (*) will be computed.

All summary statistics (and also all other outputs) are computed for the specified samples. By default all individuals of all populations are sampled. The chapter 7.4 describes how to specify a sampling schema. The only summary statistics available across all individuals and populations (at any time) describe the demography and are listed below containing an extension *Tot*.

Table 7.1: Summary statistics available for the demographic structure

CONFIDENTIAL

Stat name	Description
Demography [(adlt/off).demo]	
(adlt/off).nbInd	total number of individuals in the metapopulation*
(adlt/off).nbFem	total number of females in the metapopulation*
(adlt/off).nbMal	total number of males in the metapopulation*

Table 7.1 continued on next page

CONFIDENTIAL

Stat name	Description
(adlt/off).meanInd	mean number of individuals per inhabited patch*
(adlt/off).meanFem	mean number of females per inhabited patch*
(adlt/off).meanMal	mean number of males per inhabited patch*
(adlt/off).sexRatio	sex ratio ($\frac{males}{females}$)*
(adlt/off).nbPops	number of inhabited patches*
(adlt/off).nbInd_p	number of individuals in patch i
(adlt/off).nbFem_p	number of females in patch i
(adlt/off).nbMal_p	number of males in patch i
Demography of all individuals: In contrast to all other stats and outputs the following stats consider all individuals and patches and not just the sampled ones. [(adlt/off).demoTot]	
(adlt/off).nbIndTot	total number of individuals in the metapopulation*
(adlt/off).nbFemTot	total number of females in the metapopulation*
(adlt/off).nbMalTot	total number of males in the metapopulation*
(adlt/off).meanIndTot	mean number of individuals per inhabited patch*
(adlt/off).meanFemTot	mean number of females per inhabited patch*
(adlt/off).meanMalTot	mean number of males per inhabited patch*
(adlt/off).sexRatioTot	sex ratio of individuals ($\frac{males}{females}$)*
(adlt/off).nbPopsTot	number of inhabited patches*
(adlt/off).nbIndTot_p	number of individuals in patch i
(adlt/off).nbFemTot_p	number of females in patch i
(adlt/off).nbMalTot_p	number of males in patch i
Patch extinction [ext.rate]	
ext.rate	proportion of extinct patches in the metapopulation*
Fecundity [fecundity, available only for adults]	
fem.meanFec	mean realized female fecundity*
fem.varFec	mean variance of realized female fecundity*
mal.meanFec	mean realized male fecundity*
mal.varFec	mean variance of realized male fecundity*
Kinship [(adlt/off).kinship]	
(adlt/off).fsib	mean proportion of full-sib*
(adlt/off).phsib	mean proportion of paternal half-sib*
(adlt/off).mhsib	mean proportion of maternal half-sib*
(adlt/off).nsib	mean proportion of non-sib*

Table 7.1 continued on next page

CONFIDENTIAL

Stat name	Description
(adlt/off).self	mean proportion of selfed offspring*
Migration [migration, available only for adults]	
emigrants	mean number of emigrants per patch*
immigrants	mean number of immigrants per patch*
residents	mean number of residents per patch*
immigrate	mean effective immigration rate per patch* ($\frac{immigrants}{immigrants+residents}$)
colonisers	mean number of colonizers per extinct patch*
colon.rate	mean effective colonization rate of extinct patches*
Fitness [fitness, available only for adults]	
VwW	variance of the fitness of adults within patches*
VwB	variance of the fitness of adults between patches*
meanW_p	mean fitness of adults in patch i (computed for each patch)
varW_p	variance of the fitness of adults in patch i (computed for each patch)
Random number initialization [seed]	
seed	outputs all used seeds (not really a statistic)*

Table 7.1: Summary statistics available for the demographic structure continued

CONFIDENTIAL

Chapter 8

Selection

CONFIDENTIAL

In the current version of quantiNEMO selection is not any more limited to the reproduction stage, where selection affected the reproductive success. Now it is possible to have selection at almost any life cycle stage. Note however, that selection may only act at a single stage at once. In quantiNEMO selection acts on the phenotypes of quantitative traits (see section 9 for how to specify quantitative traits). The fitness of an individual is then determined by the interaction of the phenotype(s) with the selection pressure(s) of the environment (see section 7.3 for how to specify the selection pressure of a patch and quantitative trait).With the following parameters it is possible to define the stage where selection acts and the level of selection.

selection_level [0-3] (default: 0)

This parameter specifies if and how selection acts at the reproduction stage. Selection acts on the phenotype of the quantitative traits (see section 9 for more details).

0 : soft selection. $nbOffs_p = N_p$

The fitness of an individual is relative to the mean fitness of the population of its patch (soft selection at the patch level). This means that the population size does not depend on the mean fitness of the population and that selection does act locally at the patch level, i.e. patches do not interact (Wallace, 1975). No populations may therefore go extinct due to their maladaptation.

1 : metapopulation selection (soft $\Leftrightarrow$ metapop). $nbOffs_p = (W_p/W_m)N_m$

The fitness of an individual is relative to the mean fitness of the entire metapopulation (soft selection at the metapopulation level). This means that the size of the entire metapopulation does not depend on the mean fitness. However, the population size of each patch depends on the mean fitness of its population (Ravigne et al., 2004). This implies that a well adapted population grows while the population size of a less adapted population declines. Individual populations may go extinct due to their maladaptation, however not the entire metapopulation. This option allows to simulate any degree of selection between soft and metapopulation selection using the parameter selection_level_coef.

2 : hard selection (soft ⇔ hard). $nbOff_p = \bar{W}_p * N_p$

The fitness of an individual is absolute. If selection acts at reproduction the number of offspring to produce is scaled by the mean fitness of the population (the maximum number of offspring to produce (with a mean fitness of 1) is defined by the parameter mating_nb_offspring). If selection acts during regulation then the fitness of an individual is identical to its survival probability. Only with hard selection the entire metapopulation may go extinct if the populations are maladapted. This option allows to simulate any degree of selection between soft and hard selection using the parameter selection_level_coef.

3 : hard selection (metapop ⇔ hard).

This model is the same as model 2, however less efficient in terms of computation load. In contrast to the previous model this model allows to simulate any degree of selection between metapopulation and hard selection using the parameter selection_level_coef.

Shown equations: The shown equation are correct for selection during reproduction and deviate slightly if selection acts at another stage. $nbOff_p$ is the number of offspring to be produced in patch p, N_p and N_m are the total number of offspring defined by the parameter mating_nb_offspring_model of patch p and metapopulation m, respectively, W_p and W_m are the sum of fitnesses of patch p and metapopulation m, respectively, and $\bar{W}_p$ is the mean fitness of patch p.

Here is an example of the different selection levels in case of selection at the reproductive success. Both populations have a carrying capacity

CONFIDENTIAL

(K) of 1000 individuals, the parameter mating_nb_model is set to 0, and the mean fitnesses of the populations ($\bar{W}$) are 0.8 and 0.4, respectively:

selection level	population 1 $K = 1000$ $\bar{W} = 0.6$	population 2 $K = 1000$ $\bar{W} = 0.2$	total
soft	1000	1000	2000
metapopulation	1500	500	2000
hard	600	200	800

selection_level_coef [decimal] (temporal/default: 1)

This parameter allows to change continuously between the selection levels soft, metapopulation and hard. This parameter is not taken into account if the parameter selection_level is set to 0 (soft selection). If the parameter is set to 1 then this parameter allows to simulate any degree of selection between soft and metapopulation selection, if set to 2 then any degree of selection between soft and hard selection, and if set to 3 any degree of selection between metapopulation and hard selection. Thereby a coefficient of 0 stands for the first mentioned selection level (e.g. soft selection) and a coefficient of 1 stands for the second mentioned selection level (e.g. hard selection).

selection_position [0-4] (default: 0)

This parameter specifies when and how selection acts. By default selection acts at reproduction where the fecundity of an individual depends on its fitness.

0 : reproductive success. Selection acts at the reproduction stage (see section 4.1), where the number of offspring of an individual depends on its fitness.

1 : reproductive success special. Selection acts at the reproduction stage (see section 4.1). In contrast to the previous option mating is here random and selection acts just after fertilization. Thus in contrast to the previous option the fitness of the offspring is considered and not the fitness of the adults. The resulting number of offspring is the same as in the previous option.

2 : offspring survival. Selection acts at the survival probability of the offspring, i.e. before dispersal (see section 4.5).

CONFIDENTIAL

3 : adult survival. Selection acts at the survival probability of the adults, i.e. after dispersal (see section 4.7).

4 : no selection. This option allows to perform simulations without any selection. This is also the case if no quantitative traits under selection are simulated (see section 9.5).

patch_mean_fitness [0-2] (default: 0)

This parameter specifies how the mean fitness of a patch is defined. By default all all individuals are considered.

0 : all. The fitness of all individuals (females and males) is considered (default).

1 : only females Just the fitness of the females is considered.

2 : only males Just the fitness of the males is considered.

CONFIDENTIAL

Chapter 9

Quantitative traits

CONFIDENTIAL

quantiNEMO allows the simulation of multiple quantitative traits each having its own specifications. Each quantitative trait is defined by one to many loci each with up to 256 alleles. The allelic effects at each locus can be drawn from a normal distribution or can be set explicitly. Mutations are implemented with several models. The trait determinism can be purely additive, or include dominance and/or epistatic interactions among loci. Environmental effects can also be set in different ways:

$$P = G + E$$

$$G_{11'22'...ii'...} = \epsilon_{11'22'...ii'...} + \sum_{i=1}^{nbLoci} a_i + a_{i'} + k_{ii'}|a_{i'} - a_i|$$

Where P is the phenotype of the trait, G the genotypic value, and E the environmental contribution to the trait. $G_{11'22'...ii'...}$ is the genotypic value of genotype $11'22'...ii'...$ (1 and $1'$ are the alleles of locus 1, 2 and $2'$ are the alleles of locus 2, ...), a_i is the effect of the first allele of locus i, $a_{i'}$ is the effect of the second allele of locus i, $k_{ii'}$ is the dominance value between allele i and i', and $\epsilon_{11'22'...ii'...}$ is the epistatic value of genotype $11'22'...ii'...$.

Each quantitative trait can have its own selection pressure, which may be stabilizing or directional selection, or the trait can be neutral. The selection pressure may vary in time and space. A quantitative trait is at least specified by the number of loci and number of alleles. In this minimal definition the genotypic value of the quantitative trait is purely additive and the traits

does not undergo any mutation. By default only additive genetic effects are simulated and each new population is initiated by assigning random allelic values within the range `[1, quanti_all]` to each locus thus assuring a very large initial variance.

9.1 Architecture

quanti_loci [integer]
This parameter specifies the number of loci defining the quantitative trait. This parameter is mandatory for the simulation of a quantitative trait.

quanti_all [1-256/matrix] (default: 255)
This parameter specifies the maximal possible number of alleles per locus. Using a matrix it is possible to specify this number for each locus of a trait separately.

CONFIDENTIAL

9.1.1 Allelic effects

Each allele has an allelic effect, its contribution to the genotype. There are two possibilities to define these effects. Either they are defined explicitly for each allele using a separate file (see parameter quanti_allelic_file), or they can be defined by specifying the variance of the normal distribution of the allelic effects (see parameter quanti_allelic_var). If the allelic effects are defined in both ways the explicitly defined effects are used. Note, that all effects have to be defined in the same way, i.e. explicitly or by their distribution.

quanti_allelic_file [string] (default: "")
This parameter allows to pass the name of a file containing allelic informations, such as the allelic effects, the mutation frequency, and/or the initial frequency. The information passed by this file has precedence over other settings, however the number of alleles and loci has to be in line with the parameters quanti_loci and quanti_all. The information can be set globally for all loci, if they have the same specifications, or for each locus separately. The allelic file has the following format:

```
#Allelic file
#################
[FILE_INFO]{
  col_locus 1
  col_allele 2
  col_allelic_value 3
  col_mut_freq 4
  col_ini_freq 5
}

#locus  allele  value   mut_freq  ini_freq
1       1       -1.0    0.2       {0  0.2}
1       2       -0.5    0.2       {0  0.2}
1       3       0.0     0.2       {1  0.2}
1       4       0.5     0.2       {0  0.2}
1       5       1.0     0.2       {0  0.2}
2       1       -2      0.2       {0  0.2}
2       2       -1      0.2       {0  0.2}
2       3       0       0.2       {1  0.2}
2       4       1       0.2       {0  0.2}
2       5       2       0.2       {0  0.2}
```

CONFIDENTIAL

The file has to start with a file information box [FILE_INFO]{...}. This box contains informations about the structure of the following table and thus allowing flexibility in the format of the table. For example the order of the columns in the table may vary, or some columns may be ignored. The file information box starts with the key word [FILE_INFO]. This key word is followed by brackets "{...}" within which the user has to specify the contents of the columns to be considered by quantiNEMO. Each column is specified by a pair consisting of a key word (e.g. col_locus followed by the column number (the ordering starts with 1). Each column definition has to be on a new line. The following column keywords are available:

col_locus This keyword specifies the column containing the locus index. If this column is not declared in the file information box, quantiNEMO will use the same settings for all loci. In this case the number of lines of the table must be equal to (quanti_all). If the keyword col_locus is declared the number of rows of the table must be equal to the number of alleles times the number of loci (quanti_loci * quanti_all).

col_allele This keyword specifies the column containing the allele index. This column is mandatory. The index of the allele goes from 1 to quanti_all.

col_allelic_value This keyword specifies the column containing the allelic effects. If this column is not set the allelic effects will be drawn randomly from a normal distribution with the variance defined by the parameter quanti_all_effect_var.

col_mut_freq This keyword specifies the column containing the probability to mutate to this allele given that a mutation occurs. This probability is only valid with the RMM mutation model (parameter quanti_mutation_model set to 0).

col_ini_freq This keyword specifies the column containing the initial frequencies of the alleles. This column allows to explicitly set the allele frequencies at the start of a simulation. The frequencies can be set for each patch separately in which case the column consists in fact of one dimensional matrices, with each value of the matrix corresponding to one population. The lines of the matrix are enclosed in braces. In the example above, individuals of the first population are initially fixed for the allele 3 at the first locus as well as at the second locus. In the second population all alleles have the same initial frequency of 0.2. Note, that the matrix is adjusted in length if the number of populations does not correspond to the length of the matrix. If this column is not given the initial allele frequencies are set globally depending on the parameter quanti_ini_allele_model.

Note, that as in all input files for quantiNEMO it is possible to add comments (also in the file information box) using the hash character: '#' or '#/ ...any text... /#'.

quanti_allelic_var [decimal/matrix] (default: 1)

This parameter allows to specify the variance of the normal distribution from where the allelic effects are drawn randomly. The normal distribution is centered around 0 (mean phenotype is by definition 0). By using a matrix it is possible to define different variances of the normal distributions for each locus, i.e. specifying different contributions of the QTLs to the trait. This parameter is taken into account only if the allelic effects are not defined explicitly by the allelic file.

CONFIDENTIAL

9.1.2 Dominance effects

By default alleles are considered to be purely additive. Dominance effects can be defined in different ways. Either the dominance effect of specific pairs of alleles are defined explicitly using a separate file (see parameter quanti_dominance_file), or by defining the normal distribution from where the dominance effects are randomly drawn (see parameters quanti_dominance_mean and quanti_dominance_var). An explicitly defined dominance effect has precedence over the general settings. By default, the parameters quanti_dominance_mean and quanti_dominance_var are set to zero resulting in a purely additive genotypic value of a locus. There are also two methods to define dominance:

quanti_dominance_model [0,1] (default: 0)

This parameter allows to choose among two models to define dominance effects.

0 : method k. $G_i = a_i + a_{i'} + k_{ii'}|a_i - a_{i'}|$

1 : method h. $G_i = 2[(1 - h_{ii'})a_i + h_{ii'}a_{i'}]$

Where G_i is the genotypic value of locus i, a_i and $a_{i'}$ are the effects of the two alleles at locus i, whereas the effect of a_i is smaller than the effect of $a_{i'}$. $k_{ii'}$ and $h_{ii'}$ are the dominance values between allele i and i' for method 1 and method 2, respectively. k and h can have the following effects:

Effect	method k	method h
overdominance	$<$ âĹŠ1	< 0
smaller allele a_i is dominant	âĹŠ1	0
recessive: purely additive	0	0.5
larger allele $a_{i'}$ is dominant	1	1
underdominance	> 1	> 1

quanti_dominance_file [string] (default: "")

This parameter allows to pass the name of a file containing the dominance effects and/or the fitness factor (see section 9.1.6). The information can be set globally for all loci, if they have the same specifications, or for each locus separately. For all not specified allele pairs other settings or the default values are applied. The dominance file has a similar format as the allelic file:

CONFIDENTIAL

```
#Dominance file
########################
[FILE_INFO]{
  col_locus           1
  col_allele1         2
  col_allele2         3
  col_dominance       4
  col_fitness_factor  5
}

#locus  allele1  allele2  dominanc  fitness
1       1        1        99999     1
1       1        2        0.298     1
1       1        3        0.435     1
1       1        4        0.224     1
1       2        2        99999     1
1       2        3        0.104     1
1       2        4        0.974     1
1       3        3        99999     1
1       3        4        0.808     1
1       4        4        99999     0
```

CONFIDENTIAL

The file has to start with a file information box [FILE_INFO]{...}. This box contains informations about the structure of the following table and thus allowing flexibility in the format of the table. For example the order of the columns in the table may vary, or some columns may be ignored. The file information box starts with the key word [FILE_INFO]. This key word is followed by brackets "{...}" within which the user has to specify the contents of the columns to be considered by quantiNEMO. Each column is specified by a pair consisting of a key word (e.g. col_locus followed by the column number (the ordering starts with 1). Each column definition has to be on a new line. The following column keywords are available:

col_locus This keyword specifies the column containing the locus index. If this column is not declared in the file information box, quantiNEMO will use the same settings (dominance effect and/or fitness factor) for all loci.

col_allele1 This keyword specifies the column containing the index of the allele with the smaller allelic effect. This column is mandatory. The index of the allele goes from 1 to quanti_all.

col_allele2 This keyword specifies the column containing the index of the allele with the larger allelic effect. This column is mandatory. The index of the allele goes from 1 to quanti_all.

col_dominance This keyword specifies the column containing the dominance effects. For any pair of alleles for which the dominance effect is not specified a dominance effect will be drawn from the normal distribution defined by the parameters quanti_dominance_mean and quanti_dominance_var.

col_fitness_factor This keyword specifies the column containing the fitness factors. For more informations on the fitness factor see please section 9.1.6. For any pair of alleles for which the fitness factor is not specified explicitly the fitness factor be taken either from the parameter quanti_sel_fitness_factor or quanti_fitness_factor_homozygote.

Note, that as in all input files for quantiNEMO it is possible to add comments (also in the file information box) using the hash character: '#' or '#/ ...any text... /#'. If you define explicitly the dominance effect and the fitness factor, but for a given pair of alleles you want only to define one of the parameters you may use the number *99999* as placeholder. quantiNEMO will treat this number as not set. In the above example this has been used to show that a monomorph locus is not valid (However a fitness factor may be set for a monomorph locus). Please note that in this special case (monomorph locus) one could have set any number as placeholder for the dominance effect, since the value would never been used.

CONFIDENTIAL

quanti_dominance_mean [decimal] (default: 0)

This parameter allows to specify the mean of the normal distribution from where the dominance effects are randomly drawn. Note that a_1 has the smaller allelic effect than a_2. This parameter is only taken into account if the dominance effects are not defined explicitly by the dominance file.

quanti_dominance_var [decimal] (default: 0)

This parameter allows to specify the variance of the normal distribution from where the dominance effects are randomly draw. This parameter is taken into account only if the dominance effects are not defined explicitly by the dominance file.

9.1.3 Epistatic effects

It is possible to simulate epistatic effects between alleles at different loci. The following equation is used to compute the genotype:

$$G_{11'22'...ii'...} = \epsilon_{11'22'...ii'...} + \sum_{i=1}^{nbLoci} G_i$$

Where $G_{11'22'...ii'...}$ is the genotypic value of genotype $11'22'...ii'...$ (1 and $1'$ are the alleles of locus 1, 2 and $2'$ the alleles of locus 2, ...), G_i is the genotypic effect of locus i (additive and dominance effects), and $\epsilon_{11'22'...ii'...}$ is the epistatic effect of genotype $11'22'...ii'...$ (unique for each multilocus genotype). There are two possibilities to define these epistatic effects. Either they are defined explicitly for each genotype using a separate file (see parameter quanti_epistatic_file), or by defining the variance of the normal distribution from where the epistatic effects are randomly drawn (see parameter quanti_epistatic_var). Using the parameter quanti_epistatic_file it is also possible to define the genotypic effects directly. If the epistatic effects are defined in both ways the explicitly defined effects are used. Note, that all effects have to be defined in the same way, i.e. explicitly or by their distribution. By default, the parameter quanti_epistatic_var is set to zero resulting in simulations without any epistatic effects.

CONFIDENTIAL

quanti_epistatic_file [string] (default: "")

This parameter allows to pass the name of a file containing the epistatic effects and/or the fitness factor (see section 9.1.6). The number of defined genotypes has to be in line with the parameters quanti_loci and quanti_all if epistatic or genotypic values are passed. If only fitness factors are defined it is enough to list the desired genotypes with their fitness factors. In this case all not specified genotypes have a fitness factor of 1. The epistatic file has a similar format as the allelic file:

```
#Epistatic file
#######################
[FILE_INFO]{
  col_genotype 1
  col_epistatic_value 2
  # col_genotypic_value 3
  col_fitness_factor 4
```

```
}

#genotype          epistaticVal   genotypicVal   fitness
{0101 0101}        0.242984       1.87009        1
{0101 0102}        0.580787       0.834811       99999
...
{5050 5048}        0.264001       1.24981        0
{5050 5049}        0.118982       -0.55701       1
{5050 5050}        0.071359       -2.26644       1
```

The file has to start with a file information box [FILE_INFO]{...}. This box contains informations about the structure of the following table and thus allowing flexibility in the format of the table. For example the order of the columns in the table may vary, or some columns may be ignored. The file information box starts with the key word [FILE_INFO]. This key word is followed by brackets "{...}" within which the user has to specify the contents of the columns to be considered by quantiNEMO. Each column is specified by a pair consisting of a key word (e.g. col_locus followed by the column number (the ordering starts with 1). Each column definition has to be on a new line. The following column keywords are available:

col_genotype This keyword specifies the column containing the genotype. This column is mandatory. The genotype is enclosed by brackets "{...}" and the genotype itself has to be in the FSTAT format (Goudet, 1995). The two alleles of a locus are written consecutively without any space. For all alleles the same number of digits are needed. Two different loci are separated by a space. The above example consists of two loci, each with 50 alleles. Each allele is written using two digits.

col_epistatic_value This keyword specifies the column containing the epistatic effects. If this column is set the epistatic effect is added to the genotypic effect computed as described above.

col_genotypic_value This keyword specifies the column containing directly the genotypic effects. If this column is set, the genotypic effect of each genotype is set directly without taking into account allelic, dominance, and/or epistatic effects. If both keywords col_epistatic_value and col_genotypic_value are specified

CONFIDENTIAL

in the information box, only the column col_genotypic_value is considered.

col_fitness_factor This keyword specifies the column containing the fitness factors. For more informations on the fitness factor see please section 9.1.6. For each genotype the fitness factor is not specified explicitly the fitness factor will be taken either from dominance file (parameter quanti_dominance_file) or from the global parameters quanti_sel_fitness_factor and quanti_fitness_factor_homozygote. If a genotype is listed to define the genotypic or epistatic value, but you don't want to define explicitly the fitness factor for this genotype you may use the number *99999* as placeholder. quantiNEMO will treat this number as not set. In the above example this has been used to show that a fitness factor for a given genotype is not set.

Note, that as in all input files for quantiNEMO it is possible to add comments (also in the file information box) using the hash character: '#' or '#/ ...any text... /#'.

quanti_epistatic_var [decimal] (default: 0)
This parameter allows to specify the variance of the normal distribution from where the epistatic effects are drawn randomly. The normal distribution is centered around 0. This parameter is only taken into account if the epistatic effects are not defined explicitly by the epistatic file.

CONFIDENTIAL

9.1.4 Pleiotropy

quantiNEMO supports pleiotropy. Pleiotropic loci are defined using the genetic map (see section 12). If the position of a QTL (parameter genome) is attributed to several loci (parameter quanti_locus_index), then this is a pleiotropic QTL and thus the involved quantitative traits pleiotrop. Mutation model and mutation rate are defined by the quantitative trait. For pleiotropic loci this has to be defined for each trait and the mutation model and mutation rate have to be identical. The allelic effect for the different traits of a pleiotropic locus mutate simultaneously, but the occurring change of the allelic effect is uncorrelated among the effects.

Example:

```
quanti_nb_trait 2
quanti_loci 3
quanti_genome {0 0.1 0.2}
quanti_locus_index_1 {1 2}
quanti_lovus_index_2 {2 3}
```

In this example the second QTL is pleiotrop characterizing both quantitative traits. In contrast the first locus just characterizes the first trait and the last locus the last trait.

9.1.5 Environment

CONFIDENTIAL

The environment may also contribute to the phenotype. There are several possibilities to set the contribution of the environment to the phenotype globally or for each patch separately. By default (without any specification of the following parameters) the environment has no effect on the phenotype of the quantitative trait. Either the contribution of the environment to the phenotype is defined directly by the variance of the environmental effect (model 0), by the narrow-sense heritability h^2 (model 1 and 2), or by the broad-sense heritability H^2 (model 3 and 4). The heritability is later translated into a corresponding environmental variance V_E:

$$h^2: \quad V_E = \frac{1-h^2}{h^2} * V_A$$

$$H^2: \quad V_E = \frac{1-H^2}{H^2} * V_G$$

Where V_A is the additive genetic variance computed following (Lynch and Walsh, 1998, p85-87), and V_G the genetic variance.

quanti_environmental_model [0-4] (default: 0)

This parameter specifies how the environmental variance is defined. The following models are available:

0 : set V_E directly. The variance of the environment is set directly by the parameter quanti_heritability, which is in this case not the heritability, but the environmental variance.

1 : V_E defined by the narrow-sense heritability (V_E constant). The variance of the environment (V_E) is set at the beginning of a simulation (generation 1) and is based on the narrow-sense heritability (h^2, parameter quanti_heritability) and the additive genetic variance (V_A at generation 1). Note, that in this case the environmental variance remains constant over time, but not the heritability.

2 : V_E defined by the narrow-sense heritability (h^2 constant). This is the same as model 1, but the environmental variance is readjusted at each generation. Thus the narrow-sense heritability remains constant over time, but not the environmental variance.

3 : V_E defined by the broad-sense heritability (V_E constant). The variance of the environment (V_E) is set at the beginning of a simulation (generation 1) and is based on the broad-sense heritability (H^2, parameter quanti_heritability) and the genetic variance (V_G at generation 1). Note, that in this case the environmental variance remains constant over time, but not the heritability.

4 : V_E defined by the broad-sense heritability (H^2 constant). This is the same as model 3, but the environmental variance is readjusted at each generation. Thus the broad-sense heritability remains constant over time, but not the environmental variance.

quanti_heritability [decimal/matrix] (default: 0)

This parameter depends on the environmental model chosen (parameter quanti_environmental_model): If V_E is directly set (model 0) this parameter is the environmental variance. For the environmental model 1 and 2 this parameter is the narrow-sense heritability (h^2):

$$h^2 = V_A/V_P$$

For the environmental model 3 and 4 this parameter is the broad-sense heritability (H^2):

$$H^2 = V_G/V_P$$

Note, that a heritability (narrow and broad sense) of 0 (no genetic component) makes no sense for models 1 to 4 in this simulation framework. Therefore a value of 0 is not accepted by quantiNEMO for models 1 to 4, resulting in an error message. If the parameter quanti_environmental_model

CONFIDENTIAL

is set to 0 the parameter quanti_heritability is no longer the heritability, but the environmental variance directly, and can be set to 0 (default). Using a matrix as argument it is possible to set the environmental variance, respectively heritability for each patch separately.

quanti_environmental_proportion [decimal] (default: 1)

This parameter specifies which environment affects the phenotype of the quantitative trait: the natal or the current (at the adult stage) environment? The argument specifies the relative weight of the current patch effect on the phenotype: if the value is 1 (default value) only the environmental variance of the current patch affects the phenotype while if the value is 0 only the environmental variance of the natal patch affects the phenotype.

quanti_va_model [0-2] (default: 0)

This parameter specifies how the additive genetic variance (V_A) of a quantitative trait is computed. The additive genetic variance of a trait is used in several statistics. For instance it is used to set the environmental variance of the patches if this variance is specified by the narrow-sense heritability ($h2$, parameter quanti_environmental_model set to 2 or 3, see also parameter quanti_heritability), the additive genetic variance may also be directly output as summary statistic (stat option $q.varA$), or indirectly in the population differentiation measurement Q_{ST} (stat option $q.qst$, $q.qst.f$, $q.qst_paair$, and $q.qst.f_paair$). The additive genetic variance (V_A) is computed following Lynch and Walsh (1998, p85-87). For a quantitative trait determined by a single locus this is:

$$V_A = 2 \sum_{i=1}^{nbAllele} p_i \alpha_i \alpha_i^*$$

Where p_i is the allele frequency of allele i, α_i is the additive effect of allele i, and α_i^* is the average excess of allele i. Note, that we show here the computation for a single locus for simplicity reasons. In quantiNEMO a full version for traits determined by multiple loci is implemented. However, the implementation does not take into account linkage between loci, thus the additive genetic variance is underestimated when loci are linked.

CONFIDENTIAL

0 : for any case. If this model is chosen the additive effect (α_i) and the average excess (α_i^*) are computed. While the average excess is simple to compute, the computation of the additive effect requires a time consuming least-square regression. This formula is valid for any mating system, but its computation is rather slow.

Note, that the least-square regression has not always a solution thus the additive effects α, are not always computable. In this case the locus in question is skipped from the analysis and a warning is returned:

```
***WARNING*** Va could not be correctly estimated (1. time
 at generation 20, see manual parameter 'quanti_va_model')
```

This warning is returned the first 10 times the problem occurs and then every hundredth time. It is up to the user to decide whether the problem occurs to often or if these "missing points" are acceptable. If the environmental variance is set by the narrow-sense heritability (parameter quanti_environmental_model set to 1 or 2) and the additive genetic variance cannot be computed quantiNEMO uses the additive genetic variance computed using the algorithm for random matings (this parameter set to 1). Note, that

1 : limited to random mating. In case of random mating the additive effect (α_i) and the average excess (α_i^*) are identical. Therefore the time consuming computation of the additive effect can be omitted. Note, that even when the mating system is set to random mating (parameter mating_system set to 0 or 2) mating is not randomly if selection acts, as the choice of the parents depends on their fitnesses. Its up to the user to decide whether this quick way to compute the additive genetic variance is appropriate or not.

1 : $V_A = V_G$. This model assumes that the additive genetic variance V_A is identical or may be approximated by the genetic variance V_G.

CONFIDENTIAL

9.1.6 Fitness factor

The fitness factor allows to translate a genotype directly into a fitness value (without taking the phenotype and the selection pressure into account). A fitness factor may be set either at the quantitative trait genotype level for each quantitative trait genotype separately (see parameter quanti_epistatic_file), or at the locus level for each combination of alleles at a locus separately (see parameter quanti_dominance_file), or globally for homozygote and/or heterozygote loci (see parameters quanti_fitness_factor_homozygote and quanti_fitness_factor_heteroz
The resulting fitness factor is determined in the above mentioned order, first quantitative trait genotype specific settings are considered, followed by locus specific settings and global parameters. Thus the default fitness factor is 1 for all cases (default value of the parameters quanti_fitness_factor_homozygote and quanti_fitness_factor_heterozygote is 1). A fitness factor smaller than 1 will result in a negative effect of this locus/genotype on the individual fitness and if the fitness factor is larger than 1 it will result in a positive effect on the individual fitness. This feature of quantiNEMO allows among others to simulate recessive deleterious alleles, heterozygote or homozygote deficits, or incompatibilities of alleles for example to simulate hybridization or speciation. It is possible to combine the fitnesses obtained directly from the genotype (fitness factor) and the one obtained through selection on the phenotype (section 9.5), although this may not really make sense. Thus if only the selection based on the fitness factor is desired the parameter quanti_selection_model has to be set to 0 (default value), i.e. the quantitative trait will be neutral in terms of environmental selection. The fitness factor f_t of a quantitative trait t is defined as the product of the fitness factors of the underlying loci f_l

$$f_t = \prod_{l}^{loci} f_l$$

.

or is set directly using the the parameter quanti_epistatic_file. The fitness of a quantitative trait is the product of the fitness factor and the fitness based on the selection pressure (see section 9.5) and the final fitness of an individual is the product of all trait fitnesses

COPYRIGHT

$$W = \prod_{t}^{traits} f_t * W_t$$

.

Where f_t is the fitness factor of trait t, W_t is the fitness based on the selection pressure of trait t (see section 9.5), and W is the total fitness of the individual.

The fitness factors may be set in different ways as described above. To define a fitness factor explicitly the epistatic file (see parameter quanti_epistatic_file) and/or the dominance file (see parameter quanti_dominance_file) may be used. To define the fitness factor globally the following parameters may be used:

quanti_fitness_factor_heterozygote [decimal/matrix] (default: 1)
This parameter allows to specify the fitness factor for heterozygote loci, i.e. if the two alleles at a given locus are different. Using a matrix it is possible to define different fitness factors for each locus. By default the fitness factor for heterozygote loci is 1, resulting in no effect on the fitness.

quanti_fitness_factor_homozygote [decimal/matrix] (default: 1)
This parameter allows to specify the fitness factor for homozygote loci, i.e. if the two alleles at a given locus are identical. Using a matrix it is possible to define different fitness factors for each locus. By default the fitness factor for homozygote loci is 1, resulting in no effect on the fitness.

fitness_factor_zero_lethal [0-1] (default: 0)
This parameter allows to define how a fitness of zero (0.0) is treated within the fitness factor (and only here). A value of 0 (default) means that an individual with fitness factor 0 has a fitness of 0, but is not lethal. Under specific circumstances such an individual may even reproduce. In contrast if this parameter is set to 1 an individual with fitness factor 0 dies just after breeding.

CONFIDENTIAL

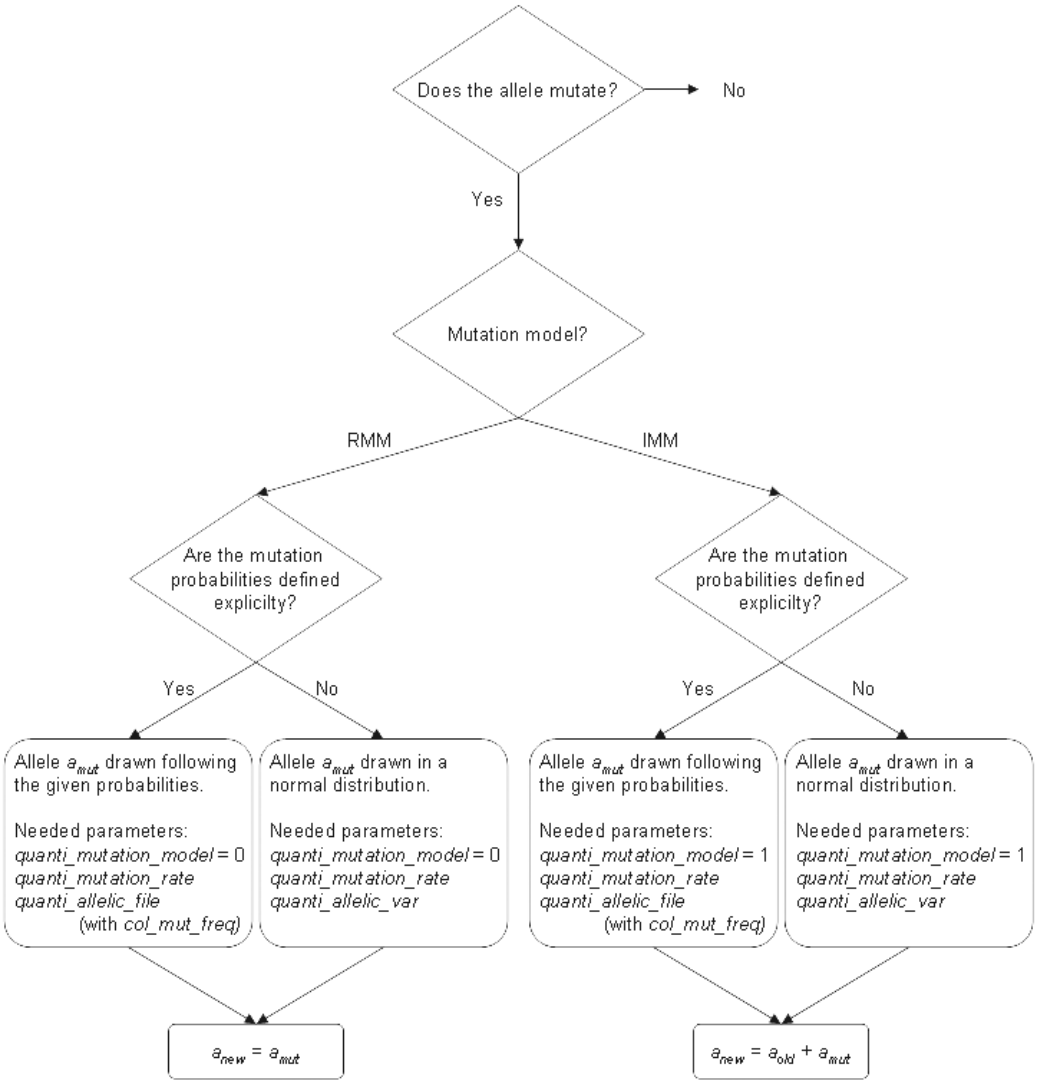


Figure 9.1: Schematic representation of the mutation process for a specific allele

9.2 Mutation

Mutation rates may be defined for each locus individually by explicitly defining the individual mutation rates (parameter quanti_mutation_rate) or by defining the gamma distribution from which the individual mutation rates are drawn (parameters quanti_mutation_rate and quanti_mutation_var). Depending on the mutation model (parameter quanti_mutation_model) the mutant effect is the effect of the drawn allele (model 0), or the effect of the drawn allele is added to the current allelic effect to get the mutant effect (model 1). Using the allelic file (see section 9.1.1) it is possible to specify for each allele its effect and the probability to mutate to this allele given that there is a mutation. A minimal definition for mutations requires the setting of a common mutation rate (parameter quanti_mutation_rate has a single value). In this case all loci have the same mutation rate and the mutation model is RMM.

CONFIDENTIAL

quanti_mutation_model [0-3] (default: 0)

This parameter allows to specify the mutation model. The following mutation models are available:

0 : RMM (Random Mutation Model)

At a given mutation a new allele is drawn randomly.

$$a_{new} = a_{mut}$$

Where a_{new} is the effect of the new allele, and a_{mut} is the effect of the drawn allele. The probability to mutate to a certain allele depends on its probability to mutate to it given that there is a mutation. These probabilities and also the effects of the alleles can explicitly be set by the allelic file (see section 9.1.1). If this is not the case quantiNEMO allocates the allelic effects and the probabilities to mutate to the alleles given that there is a mutation automatically: The effects of the alleles are regularly spaced between -6σ and 6σ, if there are more than 5 alleles, and the range is down regulated if the number of alleles is lower. Note, that in this later case the variance of the allelic effects in a population may not meet the specified variance. Where σ is the square root of the variance of the normal distribution from where the allelic effects are

drawn randomly (see parameter quanti_allelic_var). The probability to mutate to a given allele given that there is a mutation is identical to the frequency of this allele defined by the distribution of the allelic effects (see parameter quanti_allelic_var), i.e. follow a normal distribution.

1 : IMM (Increment Mutation Model)
At a given mutation an allele effect is drawn randomly and is added to the current allelic effect.

$$a_{new} = a_{old} + a_{mut}$$

Where a_{new} is the effect of the new allele, a_{mut} is the effect of the drawn allele, and a_{old} is the effect of the old allele, before the mutation. The probability to mutate to a certain allelic effect depends on its probability to mutate to it given that there is a mutation. These probabilities and also the effects of the alleles can explicitly be set by the allelic file (see section 9.1.1). If this is not the case quantiNEMO allocates the allelic effects and the probabilities to mutate to the alleles given that there is a mutation automatically: The effects of the alleles are regularly spaced between -20σ and 20σ, where σ is the square root of the variance of the normal distribution from where the allelic effects are drawn randomly (see parameter quanti_allelic_var). The probability to mutate to a given allelic effect given that there is a mutation is identical to its allele frequency, i.e. follow a normal distribution. This mutation model requires that the allelic effects are regularly spaced around zero, that the number of alleles is odd (thus the allele with index $\lfloor$quanti_all$/2\rfloor$ is zero), and that there are at least 51 possible alleles (parameter quanti_all).

2 : KAM (K-Allele Model)
At each mutation the existing allele is randomly exchanged by another allele within the range of alleles (i.e. $[1; quanti\_all]$) not taking into account any allelic effect. By default the probability to mutate to any allele is the same. However, if the mutation probabilities are specified explicitly by the allelic file (see section 9.1.1) the probability to mutate to a certain allele depends on the specified probability to mutate to this allele.

COPYRIGHT

3 : SSM (Single Step Mutation)
In contrast to the K-Allele-Model the new allele depends on the current allele. When a mutation occurs the current allele is replaced by one of its neighboring alleles (concerning the allele index). For example, if the allele with the index 12 mutates, it changes either to the allele with the index 13, or to the allele with the index 11. The boundaries are reflexive, i.e. the allele index can not exceed the range of alleles (i.e. $[1; ntrl\_all]$). In line with the Increment Mutation Model (IMM) for quantitative traits it is possible to specify mutation probabilities of the "steps to mutate" using the allelic file (see section 10.1). In this case the number of alleles (ntrl_all) has to be odd (as for the IMM). Then similar to the IMM the probabilities of the step to mutate has to be defined using the allelic file, where the step size is measured from the middle allele (concerning its index, i.e. $ntrl\_all/2$). The middle allele (with index $ntrl\_all/2$) has to have a mutation probability of 0 as a mutation of step zero is not a mutation. This allows generating any mutation pattern, as for instance the Generalized stepwise mutation model (Estoup et al., 2002).

Note, that for all mutation models the number of alleles have to be odd if the allelic effects and the probabilities to mutate to the alleles given that there is a mutation are set automatically. In this case also a warning will be drawn if the number of alleles is below 200, informing that this number of alleles may not well represent the normal distribution of the allelic effects.

CONFIDENTIAL

quanti_mutation_rate [decimal/matrix] (temporal/default: 0)
This parameter specifies the mutation rate per locus and generation. If the argument is a single value the mutation rate for all loci is the same. By passing a matrix of mutation rates it is possible to set the mutation rate for each single locus individually. By default no mutations occur.

mutation_trial [integer] (default: 1e4)
This parameter allows setting the number of trials a randomly drawn allele tries to mutate to another allele and if not successfully the allele does not change, i.e. no mutation occurs. The parameter only applies to the RMM mutation model (parameter quanti_mutation_model set

to 0) where the probability to mutate to any other allele may be small (see also parameter quanti_allelic_file).

9.3 Initial genotypes

There are several methods to set the allele frequencies or even the genotypes of the individuals at the start of a simulation (initialisation). The genotypes of the individuals may be set using an FSTAT file (Goudet, 1995) (parameter **quanti_ini_genotypes**). If such a FSTAT file is not present the genotypes are randomly drawn following the explicitly set allele frequencies in the allelic file (see parameter **quanti_allelic_file** and especially column keyword **col_ini_freq**). If the allele frequencies are not set explicitly in the allelic file the initialization is performed following the parameter **quanti_ini_allele_model**.

CONFIDENTIAL

quanti_ini_genotypes [string] (default: "")

This parameter allows to specify a name of an FSTAT file (Goudet, 1995) containing the initial genotypes of the individuals for each population. If such a file is present the initialization of the metapopulation is done solely by this file ignoring the parameters **quanti_ini_allele_model** and **patch_ini_size**. A single FSTAT file is needed for all quantitative traits together containing the total number of loci of all quantitative traits together. Note, that quantiNEMO allows to output an appropriate file for any generation (see parameter **quanti_save_genotype**). This allows to resume a simulation, to generate tailored initial conditions, or to continue a simulation with modified settings. If the parameter **quanti_save_genotype** is set to 2 an extended FSTAT file is generated. Also this file may be used to initialize the metapopulation. In this case quantiNEMO overtakes the supplement information provided by the file, especially the sex and age of the individual, the index of the individual, its mother and father. Thus the supplement information allows to resume a simulation without the loss of the pedigree. Note, that for an entire resume of a simulation also the genotypes of the neutral markers have to be set (see parameter **ntrl_save_genotype**). In this case the number of individuals per patch and the individuals supplement information (sex, age, individuals id, mothers id, fathers id

must be in agreement with each other.

quanti_ini_allele_model [0,1] (default: 0)
If the genotypes or allele frequencies are not already defined in another way, the initialization of the genotypes may be either polymorphic, where the probability of each allele is identically or monomorphic, where all populations are fixed for a single allele.

0 : polymorph. The populations are maximally polymorph in respect to allele frequencies at the start of a simulation.

1 : monomorph. The populations are monomorph in respect to allele frequencies at the start of a simulation. All individuals are fixed for a single allele, which is the "middle" allele, i.e. the allele with the index $\lfloor \mathsf{ntrl\_all}/2 \rfloor$.

9.4 Selection pressure

Currently there are two ways to define the selection pressure. The parameter selection_pressure_definition allows to switch among them. Selection pressures may vary among quantitative traits, sexes, patches, and time. To specify the selection pressure individually for quantitative traits and patches, matrices may be used. They are adjusted to the number of patches if needed (see section 3.4). Selection pressures have to be specified either for each sex separately (parameters with the suffix "_fem" for females and "_mal" for males), or for both sexes together (parameters without a suffix). In the first case both sex specific parameters have to be set if two sexes are simulated. In the latter case the selection pressure of females and males are assumed to be identical.

quantiNEMO supports three types of selection, stabilizing selection, directional selection and selection based on a fitness landscape. The type of selection may vary among quantitative traits, but not among patches and sexes. However, selection pressure of the same type of selection may vary among patches, quantitative traits, sexes, and may also change through time. This flexibility allows for example to simulate a dynamic environment such as for global warming.

CONFIDENTIAL

selection_pressure_definition [0;1] (default: 0)
This parameter specifies how the selection pressure is defined.

0 : patch. The selection pressure is defined at the patch level as described in section 7.3. The advantage is that across quantitative traits the matrix expansion may be used.

1 : quanti. The selection pressure is defined a the quantitative trait level as described in this section. The advantage is that the parameters may be extended by the postix "_1", "_2". This allows to define the selection pressure separately for each quantitative trait or to group the selection pressure among groups.

9.4.1 Stabilizing selection

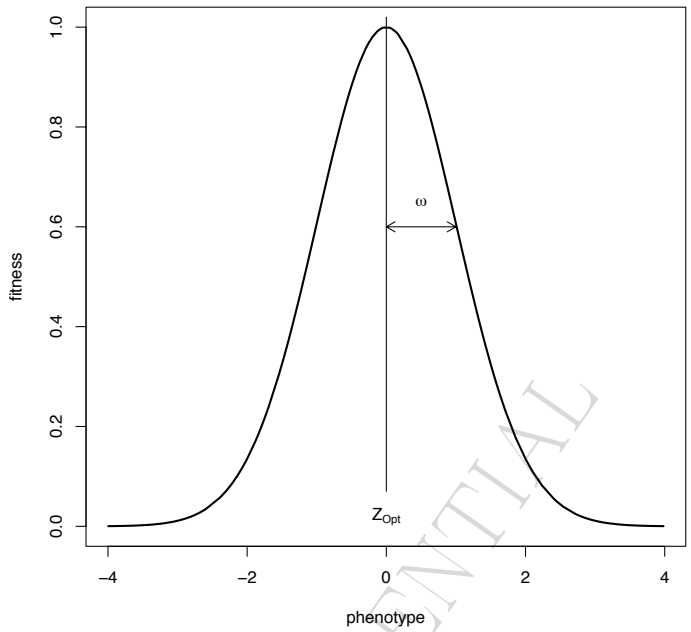
Stabilizing selection may act on the phenotype (P) of quantitative traits. The selection pressure is defined by the parameters quanti_stab_sel_optima (Z_{Opt}) and quanti_stab_sel_intensity (ω). The fitness (W) of a quantitative trait is computed using the following standard Gaussian function for stabilizing selection:

$$W = e^{\left(-\frac{\left(P-Z_{Opt}\right)^2}{2\omega^2}\right)}$$

CONFIDENTIAL

quanti_stab_sel_optima
quanti_stab_sel_optima_fem
quanti_stab_sel_optima_mal [decimal/matrix] (temporal/default: 0)
These parameters allow to set the selection optimum z_{Opt} for each patch for a given quantitative trait.

quanti_stab_sel_intensity
quanti_stab_sel_intensity_fem
quanti_stab_sel_intensity_mal [decimal/matrix] (temporal/default: 1)
These parameters allow to set the selection intensity ω for each patch for a given quantitative trait. Contraintuitively a small value results in a strong selection pressure, whereas a large value results in a weak selection pressure.



CONFIDENTIAL

patch_stab_sel_optima_var [decimal/matrix] (temporal/default: 0)

This parameter specifies the variance of the normal distribution by which the selection optimum varies between generations (e.g. annual fluctuations of the mean temperature). By default the local selection optimum does not vary.

patch_stab_sel_intensity_var [decimal/matrix] (temporal/default: 0)

This parameter specifies the variance of the normal distribution by which the selection intensity varies at each generation (e.g. annual fluctuations of the mean temperature). By default the local selection intensity does not vary.

Example

```
patch_number            2
quanti_nb_trait         3
```

```
quanti_stab_sel_optima_1 {-0.1  0.1}
quanti_stab_sel_optima_2 { 0.2  0.2}
quanti_stab_sel_optima_3 {-0.3  0.3}
quanti_stab_sel_intensity 1
```

In this example the environment consist of two patches with varying selection pressures. Three quantitative traits are simulated. The first trait has a selection optimum at -0.1 in patch 1 and at 0.1 in patch 2. The selection optimum of the second trait is the same in both patches (0.2). The third trait has an optimum at -0.3 in patch 1 and at 0.3 in patch 2. The intensity of the selection is identical for all three traits and in both patches.

9.4.2 Directional selection

Directional selection may act on quantitative traits. The fitness (W) of a quantitative trait is computed using the following generalized logistic function (Richards, 1959):

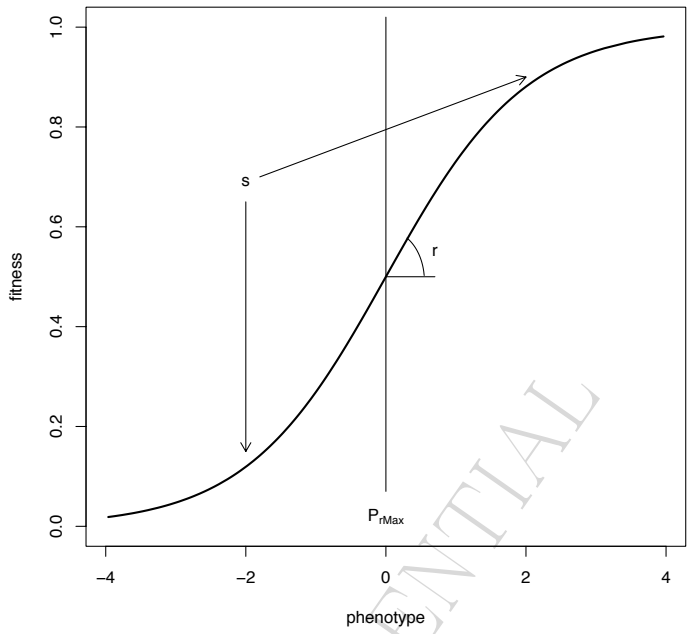
$$W = min + \frac{max - min}{(1 + s * e^{r(P_{r_{Max}} - P)})^{1/s}}$$

Where min is the lower asymptote (parameter quanti_dir_sel_min), max is the upper asymptote (parameter quanti_dir_sel_max), r is the growth rate (parameter quanti_dir_sel_growth_rate), $P_{r_{Max}}$ is the phenotype with the maximal slope (parameter quanti_dir_sel_max_growth), and s defines the symmetry of the curve (parameter quanti_dir_sel_symmetry; the curve is symmetric by default (value 1)).

quanti_dir_sel_min
quanti_dir_sel_min_fem
quanti_dir_sel_min_mal [decimal/matrix] (temporal/default: 0)
: These parameters allow to set the lower asymptote of the selection curve for each patch for a given quantitative trait.

quanti_dir_sel_max
quanti_dir_sel_max_fem
quanti_dir_sel_max_mal [decimal/matrix] (temporal/default: 1)
: These parameters allow to set the upper asymptote of the selection curve for each patch for a given quantitative trait.

CONFIDENTIAL



CONFIDENTIAL

quanti_dir_sel_growth_rate
quanti_dir_sel_growth_rate_fem
quanti_dir_sel_growth_rate_mal [decimal/matrix] (temporal/default: 1)

These parameters allow to set the slope of the selection curve for each patch for a given quantitative trait. If the argument is positive larger phenotypes have a higher fitness, while if negative smaller phenotypes have a higher fitness.

quanti_dir_sel_max_growth
quanti_dir_sel_max_growth_fem
quanti_dir_sel_max_growth_mal [decimal/matrix] (temporal/default: 0)

These parameters allow to set the phenotype with the maximal growth.

quanti_dir_sel_symmetry
quanti_dir_sel_symmetry_fem
quanti_dir_sel_symmetry_mal [decimal/matrix] (temporal/default: 1)

These parameters allow to set the symmetry of the curve. The default value of 1 results in a symmetric slope.

patch_dir_sel_min_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the lower asymptote of the selection curve varies at each generation (e.g. annual fluctuations of the mean temperature). By default the lower asymptote of the selection curve does not vary.

patch_dir_sel_max_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the upper asymptote of the selection curve varies at each generation (e.g. annual fluctuations of the mean temperature). By default the upper asymptote of the selection curve does not vary.

patch_dir_sel_growth_rate_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the selection slope varies at each generation (e.g. annual fluctuations of the mean temperature). By default the local selection slope does not vary.

patch_dir_sel_max_growth_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the phenotype with maximal growth varies at each generation (e.g. annual fluctuations of the mean temperature). By default the local phenotype with maximal growth does not vary.

patch_dir_sel_symmetry_var [decimal/matrix] (temporal/default: 0)
This parameter specifies the variance of the normal distribution by which the symmetry of the curve varies at each generation (e.g. annual fluctuations of the mean temperature). By default the symmetry of the slope does not vary.

Example

```
quanti_dir_sel_growth_rate 1
quanti_dir_sel_max_growth  0
```

CONFIDENTIAL

```
quanti_dir_sel_symmetry     1
```

In this example the selection pressure for all patches and quantitative traits are identical and set to the default values. The specified directional selection pressure favours larger phenotypes (parameter quanti_dir_sel_growth_rate is positive). This means that individuals with larger phenotypes have on average higher fitnesses and thus higher reproductive successes.

9.4.3 Fitness landscape

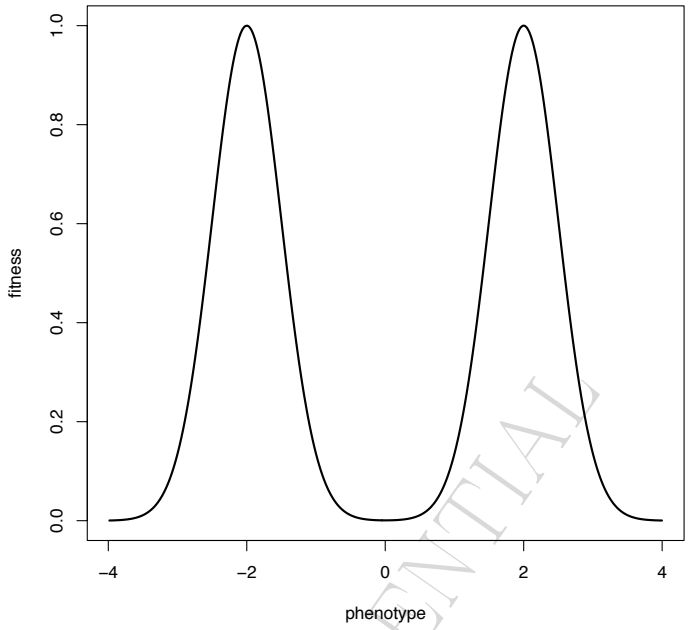
Specifying a fitness landscape allows to define any selection pressure for a given quantitative trait. A fitness landscape is in principle defined by a 2 dimensional matrix containing phenotype values and their corresponding fitness values. There is no limit to the number of phenotype-fitness value pairs. For practical reasons the two vectors phenotype values (parameter quanti_phenotype_landscape) and fitness values (parameter quanti_phenotype_landscape) are separately passed to quantiNEMO. The fitness value of any phenotype within the specified range of phenotypes is linearly interpolated and if the phenotype lies outside of the specified range of phenotypes the resulting fitness value will result in the fitness value of the smallest and largest phenotype, respectively. By default all phenotypes will result in a fitness of 1.

The fitness landscape is defined by two tightly linked parameters:

quanti_phenotype_landscape
quanti_phenotype_landscape_fem
quanti_phenotype_landscape_mal [decimal/matrix] (temporal/default: 0)

These parameters allow to specify an array of phenotypes for which the corresponding fitness is defined using the parameter quanti_fitness_landscape. The phenotype values are either specified by a single value (similar to the default value) specifying a unique phenotype for all patches and quantitative traits together (all phenotypes will result in the same fitness), or by a one dimensional array (1D matrix) specifying the same fitness landscape for all patches and a given quantitative traits together, or by a 2D matrix specifying the fitness landscape separately for each patch for a given quantitative trait. Since the two parameters quanti_fitness_phenotype and quanti_fitness_fitness are tightly linked,

CONFIDENTIAL



CONFIDENTIAL

i.e. specify the phenotypes and their corresponding fitness values, the architecture of the two parameters have to be identical, i.e. show up the same number of values for a patch.

quanti_fitness_landscape
quanti_fitness_landscape_fem
quanti_fitness_landscape_mal [decimal/matrix] (temporal/default: 1)

These parameters allow to specify the fitnesses corresponding to the phenotypes defined with the parameter quanti_phenotype_landscape, in the same order as the phenotypes are defined in the parameter quanti_fitness_phenotype. The fitness values are either specified by a single value (similar to the default value) specifying a unique fitness for all patches and quantitative traits together, or by a one dimensional array (1D matrix) specifying the same fitness landscape for all patches for a given quantitative traits together, or by a 2D matrix specifying the fitness landscape separately for each patch of a given quantitative trait. Since the two parameters quanti_fitness_phenotype

and quanti_fitness_fitness are tightly linked, i.e. specify the phenotypes and their corresponding fitness values, the architecture of the two parameters has to be the identical, i.e. show up the same number of values for a patch at a given quantitative trait.

Example

```
quanti_phenotype_landscape {-4.0 -3.2 -2.4 -1.6 -0.8  0.0
                             0.8  1.6  2.4  3.2  4.0}
quanti_fitness_landscape   {0.00 0.05 0.72 0.72 0.05 0.00
                            0.05 0.72 0.72 0.05 0.00}
```

In this example the selection pressure for all patches and quantitative traits are identical. The specified fitness landscape corresponds to the above shown figure, however the resolution of the specification is reduced in the sown example. The binomial landscape is the sum of two stabilizing selections with optima set to -2 and 2 and an intensity of 0.5.

CONFIDENTIAL

9.4.4 Selection coefficient

quantiNemo2 supports the simulation of bi-allelic loci where the selection pressure is defined by a selection coefficient. In this simplified, but widespread model, the genotype is directly mapped to a fitness:

Genotype	Fitness	Fitness extended
AA	1	F
Aa	1 - hs	F - hs
aa	1 - s	F - s

Where s is the selection coefficient (parameter quanti_coef_sel), h the dominance factor (parameter quanti_dominance_mean) with the model (parameter quanti_dominance_model) set to 1 (h-model, has to be set explicitly!), and F is the fitness factor for the wild type genotype AA set by parameter quanti_coef_sel_AA. This latter parameter is normally not used therefore it is listed in the column "Fitness extended".

quanti_coef_sel
quanti_coef_sel_fem

quanti_coef_sel_mal [decimal/matrix] (temporal/default: 0)
These parameters allow to set the selection coefficient s for each patch and quantitative trait/locus. The default value of 0 will result in equivalent fitness for both alleles, and thus no selection acts.

quanti_coef_sel_AA
quanti_coef_sel_AA_fem
quanti_coef_sel_AA_mal [decimal/matrix] (temporal/default: 1)
These parameters allow to set the fitness for the wild type genotype AA for each patch and quantitative trait/locus. By default it is 1.

9.5 Selection models

CONFIDENTIAL

The fitness of an indivudal is defined by the interaction of the pheotypes of individuals with their environments. The shape of the environmental selection pressure may vary. Each quantitative trait has its individual architecture, and may be selected for a different selection pressure. Selection types may differ between quantitative traits. The selection pressure has to be defined at the patch level (see section 7.3) or at the quantitative trait level (see section 9.4) depending on personal choice (see parameter **selection_pressure_definition**). If several quantitative traits are simulated the fitness of an individual is the product of the quantitative trait's fitnesses:

$$W = \prod_{t}^{traits} f_t * W_t$$

Where W is the total fitness of the individual, f_t is the fitness factor of quantitative trait t (see section 9.1.6), and W_t is the fitness of quantitative trait t.

quanti_selection_model [0-4] (default: 0)
This parameter allows defining the selection model for each quantitative trait. By default no selection acts on a quantitative trait

0 : neutral quantitative trait.
The quantitative trait is not under selection. The fitness of this quantitative trait is set to 1 not taking into account the phenotype:

$$W_t = 1$$

1 : stabilizing selection

Stabilizing selection acts on the quantitative trait. The fitness W of quantitative trait t is computed using the following standard Gaussian function for stabilizing selection:

$$W_t = e^{\left(-\frac{\left(P-Z_{Opt}\right)^2}{2\omega^2}\right)}$$

Z_{Opt} is the selection optimum of the current trait and patch (parameter patch_stab_sel_optima), P is the phenotypic value of the individual, and ω is the intensity of the selection (parameter patch_stab_sel_intensity).

2 : directional selection.

Directional selection acts on the quantitative trait. A generalized logistic curve (Richards, 1959) is implemented in quantiNEMO to characterize the directional selection pressure:

$$W_t = \left(1 + s * e^{r(P_{r_{Max}}-P)}\right)^{-1/s}$$

Where r is the growth rate (parameter patch_dir_sel_growth_rate), $P_{r_{Max}}$ is the phenotype with the maximal slope (parameter patch_dir_sel_max_growth), and s defines the symmetry of the slope (parameter patch_dir_sel_symmetry; slope is symmetric by default (value 0.5)).

3 : fitness landscape.

The selection pressure is defined by a fitness landscape. The fitness value of any phenotype may be specified. The fitness value of a phenotype within the specified range of phenotypes is linearly interpolated, Phenotype outside the range of specified phenotypes will result in the fitness value of the smallest phenotype, respectively in the fitness value of the largest phenotype.

4 : selection coefficient.

The selection pressure of a bi-allelic locus is defined by a selection coefficient. In this simplified, but widespread model, the genotype is directly mapped to a fitness. Note that for this type of selection a quantitative trait may only contain a single bi-allelic locus, without explicitly specified allelic effects. If environment variance shapes the fitness this has to be defined directly by the variance

CONFIDENTIAL

(and not by any heritability).

Genotype	Fitness	Fitness extended
AA	1	F
Aa	1 - hs	F - hs
aa	1 - s	F - s

Where s is the selection coefficient (parameter patch_sel_coef), h the dominance factor (parameter quanti_dominance_mean) with the model (parameter quanti_dominance_model) set to 1 (h-model, has to be set explicitly!), and F is the fitness factor for the wild type genotype AA set by parameter patch_sel_coef_AA. This latter parameter is normally not used therefore it is listed in the column "Fitness extended".

9.6 Output

Apart from the genotypes and the phenotypes it is also possible to obtain certain summary statistics, the allelic, the dominance, and the epistatic effects. For each type of output it is possible to define individually the time interval at which the output is generated, which may also vary over time.

9.6.1 Genotype

The genotype of the sampled individuals and popualtions may periodically be dumped to files. The output files will be stored in the folder given by the parameter quanti_genot_dir and will have the name of the base file name (see parameter filename in section 5). The extension is ".dat". A counter for the generation (e.g. _g05) and the replicate (e.g. _r4) is inserted before the extension. An example of such a file name is "simulation_g05_r4.dat". Note, that such a genotype file may be used to start a new simulation (see parameter **quanti_ini_genotypes**). By default all individuals of all populations are sampled. The chapter 7.4 describes how to specify a sampling schema.

quanti_save_genotype [0-2] (default: 0)

This parameter specifies the output of the quantitative genotype at the QTLs.

CONFIDENTIAL

0 : None. No output is generated.

1 : FSTAT. Genotypes are outputted in the FSTAT format (Goudet, 1995).

2 : FSTAT extended. Same as point 1, but the file contain the following six additional columns: the age class (1 = offspring, 2 = adult), the sex (0 = male; 1 = female), the ID of the individual, the ID of the mother, the ID of the father, and the fitness of the individual. The ID is a unique identifier for each individual of a simulation in the format "345_23", meaning that this is the 345th individual born in patch 23. The IDs of the individual, the mother and the father allow to extract pedigree informations, if the output is stored for each generation, and also to investigate the migration behavior of the individual and its parents.

3 : Arlequin. Genotypes are outputted in Arlequin format (Excoffier, 2010).

4 : Arlequin extended. Same as point 3, but with additional commented individual information as in point 2.

5 : PLINK. Outputs the **quantitative** genotypes with the phenotypes of the quantitative traits (Purcell et al., 2007). The standard two files .ped and .map are created. The 6th column (phenotype) of the .ped file contains the fitness of the individual. If quantitative traits are simulated, their phenotypes are listed in an alternate phenotype file (.pheno). The alleles of all bi-allelic loci (quanti_nb_allele set to 2) are listed. The .map file is generated for each replicate. The .ped and .pheno files are generated for each specified time point and outputs of successive time points are concatenated to a single file, allowing to obtain entire or parts of pedigrees. The filename of such a concatenated file contains the time stamp of the first entry. Note that only successive individuals list their parentIDs. To generate an entire pedigree the entire populations have to be sampled.

6 : PLINK extended. Same as point 5, but all **quantitative** genotypes are listed, including the multi-allele loci.

An example of such a file (with quanti_save_genotype set to 2):

```
5 4 20 2
```

CONFIDENTIAL

```
t1_l1
t1_l2
t1_l3
t1_l4
1 1415 1019 2002 0820 1 1 10_1 1_1 0_1 0.345
1 0814 0219 2002 2020 1 1 11_1 8_1 2_4 0.334
1 0808 0217 1902 0820 1 1 12_1 5_3 5_1 0.123
...
5 1004 0917 1404 1007 1 1 16_5 9_5 3_2 0.999
5 2017 1010 2013 1812 1 0 17_5 3_2 9_2 1.000
5 2017 1008 2013 1811 1 1 11_4 8_2 9_2 0.678
```

The first line contains the number of patches (5 patches here), the number of loci (4), the highest possible allele index (20), and the number of digits used to write each allele (2). The next four lines contain the locus names. The following lines contain the individual's info, one individual per line. The first number is the patch number of the individual, followed by the genotype. Each column represents a locus, and the first half of the locus (first 2 digits) represents the first allele index, while the second half of the locus (last 2 digits) the second allele at the given locus. As, in this example, we are using two digits per allele, the first two digits of a locus genotype number are the first allele (e.g. allele 14 for the first allele of the first locus of the first individual) while the two next digits are the second allele (e.g. allele 15 for the second allele of the first locus of the first individual). Each line ends with six columns consisting supplementary information on the individual (see above) if the parameter quanti_save_genotype is set to 2.

quanti_genot_dir [string] (default: "")
This parameter allows to specify the subdirectory where the genotypes are stored. This directory has to be specified relative to the simulation folder (parameter folder), and may also contain subdirectories. If not specified (default) the output is stored in the simulation folder (parameter folder).

quanti_genot_filename [string] (default: "")
This parameter is used to specify an individual base filename for the genotypes. If not specified the generic base filename will be used (see parameter filename).

quanti_genot_logtime [integer] (temporal/default: 1)

CONFIDENTIAL

This parameter specifies the time interval of the genotype output. Since the parameter may change over time the output may be generated at any generation.

quanti_genot_script [string] (default: "")
It is possible to launch a script just after the genotype file is generated. The argument of the parameter is the file name of the script. The name of the genotype file is passed as unique parameter to the script.

quanti_genot_sex [0-2] (default: 0)
This parameter allows to choose which sex is output.

0 : Both. Output includes both sexes.

1 : Females. Output includes only female genotypes.

2 : Males. Output includes only male genotypes.

quanti_genot_age [0-2] (default: 0)
This parameter allows to choose which age is output.

0 : Adults. Output includes only adult genotypes.

1 : Juveniles. Output includes only juvenile genotypes.

2 : Both. Output includes juveniles and adults genotypes.

CONFIDENTIAL

9.6.2 Genotypic value

Similar to the genotypes the genotypic values may be periodically dumped to files. The output files will be stored in the folder given by the parameter quanti_geno_value_dir and will have the name of the base file name (see parameter filename in section 5). The extension is ".gen". A counter for the generation (e.g. _g05) and the replicate (e.g. _r4) is inserted before the extension. An example of such a file name is "simulation_g05r4.gen". All summary statistics (and also all other outputs) are computed for the specified samples. By default all individuals of all populations are sampled. The chapter 7.4 describes how to specify a sampling schema.

quanti_save_geno_value [0-2] (default: 0)
This parameter specifies the output of the phenotype.

0 : None. No output is generated.

1 : Standard. The output contains the phenotypes in the standard FSTAT-like format (Goudet, 1995).

2 : Extended. Same as point 1, but the file contain the following six additional columns: the age class (1 = offspring, 2 = adult), the sex (0 = male; 1 = female), the ID of the individual, the ID of the mother, the ID of the father, and the fitness of the individual. The ID is a unique identifier for each individual of a simulation in the format "345_23", meaning that this is the 345th individual born in patch 23. The IDs of the individual, the mother and the father allow to extract pedigree informations, if the output is stored for each generation, and also to investigate the migration behavior of the individual and its parents.

An example of such a file (quanti_save_geno_value is set to 2):

```
2 5
genotypic_value_trait-1
genotypic_value_trait-2
genotypic_value_trait-3
genotypic_value_trait-4
genotypic_value_trait-5
1 0.0493 -3.203 -2.441 0.0683 -3.199 2 1 10_1 1_1 0_1 0.345
1 0.4924 -3.803 -0.869 -2.002 -2.594 2 1 11_1 8_1 2_2 0.334
1 2.2342 -2.931 -0.725 -0.750 -0.698 2 1 12_1 5_2 5_1 0.123
...
2 0.8623 0.6525 -0.857 1.7483 -4.194 2 1 16_2 9_2 3_2 0.999
2 1.7752 -2.223 -3.117 0.3409 -2.003 2 1 17_2 3_2 9_2 1.000
2 0.2081 -2.803 -0.146 -0.456 -5.137 2 1 11_1 8_1 9_1 0.678
```

The first line contains the number of patches (2 patches here), and the number of traits (5). The next five lines contain the five trait names. The following lines contain the individual's info, one individual per line. The first number is the patch number of the individual, followed by the genotypic value for each trait. Each line ends with six columns consisting supplementary information on the individual (see above) if the parameter quanti_save_geno_value is set to 2.

quanti_geno_value_dir [string] (default: "")

This parameter allows to specify the subdirectory where genotypic values are stored. This directory has to be specified relative to the sim-

CONFIDENTIAL

ulation folder (parameter folder), and may also contain subdirectories. If not specified (default) the output is stored in the simulation folder (parameter folder).

quanti_geno_value_filename [string] (default: "")
This parameter is used to specify an individual base filename for the genotypic values. If not specified the generic base filename will be used (see parameter filename).

quanti_geno_value_logtime [integer] (temporal/default: 1)
This parameter specifies the time interval of the genotypic value output. Since the parameter may change over time the output may be generated at any generation.

quanti_geno_value_script [string] (default: "")
It is possible to launch a script just after the genotypic value file is generated. The argument of the parameter is the file name of the script. The name of the genotypic value file is passed as unique parameter to the script.

quanti_geno_value_sex [0-2] (default: 0)
This parameter allows to choose which sex is output.

0 : Both. Output includes both sexes.

1 : Females. Output includes only female genotypic values.

2 : Males. Output includes only male genotypic values.

quanti_geno_value_age [0-2] (default: 0)
This parameter allows to choose which age is output.

0 : Adults. Output includes only adult genotypic values.

1 : Juveniles. Output includes only juvenile genotypic values.

2 : Both. Output includes juveniles and adults genotypic values.

9.6.3 Phenotypic value

Similar to the genotypes the phenotypic values of the adults may be periodically dumped to files. The phenotype of juveniles cannot be output, as

CONFIDENTIAL

the phenotype is only computed when selection acts, and this is at the reproduction stage, i.e. when individuals are adults. The output files will be stored in the folder given by the parameter quanti_phenot_dir and will have the name of the base file name (see parameter filename in section 5). The extension is ".phe". A counter for the generation (e.g. _g05) and the replicate (e.g. _r4) is inserted before the extension. An example of such a file name is "simulation_g05r4.phe". All summary statistics (and also all other outputs) are computed for the specified samples. By default all individuals of all populations are sampled. The chapter 7.4 describes how to specify a sampling schema.

quanti_save_phenotype [0-2] (default: 0)

This parameter specifies the output of the phenotype.

0 : None. No output is generated.

1 : Standard. The output contains the phenotypes in the standard FSTAT-like format (Goudet, 1995).

2 : Extended. Same as point 1, but the file contain the following six additional columns: the age class (1 = offspring, 2 = adult), the sex (0 = male; 1 = female), the ID of the individual, the ID of the mother, the ID of the father, and the fitness of the individual. The ID is a unique identifier for each individual of a simulation in the format "345_23", meaning that this is the 345th individual born in patch 23. The IDs of the individual, the mother and the father allow to extract pedigree informations, if the output is stored for each generation, and also to investigate the migration behavior of the individual and its parents.

An example of such a file (quanti_save_phenotype is set to 2):

```
2 5
phenotypic_value_trait-1
phenotypic_value_trait-2
phenotypic_value_trait-3
phenotypic_value_trait-4
phenotypic_value_trait-5
1 0.0493 -3.203 -2.441 0.0683 -3.199 2 1 10_1 1_1 0_1 0.345
1 0.4924 -3.803 -0.869 -2.002 -2.594 2 1 11_1 8_1 2_2 0.334
1 2.2342 -2.931 -0.725 -0.750 -0.698 2 1 12_1 5_2 5_1 0.123
...
```

CONFIDENTIAL

```
2 0.8623 0.6525 -0.857 1.7483 -4.194 2 1 16_2 9_2 3_2 0.999
2 1.7752 -2.223 -3.117 0.3409 -2.003 2 1 17_2 3_2 9_2 1.000
2 0.2081 -2.803 -0.146 -0.456 -5.137 2 1 11_1 8_1 9_1 0.678
```

The first line contains the number of patches (2 patches here), and the number of traits (5). The next five lines contain the five trait names. The following lines contain the individual's info, one individual per line. The first number is the patch number of the individual, followed by the phenotype value for each trait. Each line ends with six columns consisting supplementary information on the individual (see above) if the parameter quanti_save_phenotype is set to 2.

quanti_phenot_dir [string] (default: "")
This parameter allows to specify the subdirectory where phenotypes are stored. This directory has to be specified relative to the simulation folder (parameter folder), and may also contain subdirectories. If not specified (default) the output is stored in the simulation folder (parameter folder).

quanti_phenot_filename [string] (default: "")
This parameter is used to specify an individual base filename for the phenotypes. If not specified the generic base filename will be used (see parameter filename).

quanti_phenot_logtime [integer] (temporal/default: 1)
This parameter specifies the time interval of the phenotype output. Since the parameter may change over time the output may be generated at any generation.

quanti_phenot_script [string] (default: "")
It is possible to launch a script just after the phenotype file is generated. The argument of the parameter is the file name of the script. The name of the phenotype file is passed as unique parameter to the script.

quanti_phenot_sex [0-2] (default: 0)
This parameter allows to choose which sex is output.

0 : Both. Output includes both sexes.

1 : Females. Output includes only female phenotypes.

2 : Males. Output includes only male phenotypes.

CONFIDENTIAL

9.6.4 Architecture

Depending on the definition of the architecture of the quantitative trait it is possible to obtain the allelic file, the dominance file, and/or the epistatic file. All three files follow the structure of the homonymous input files, except that for the input all possible combinations have to be present, while in the output only the used combinations are output. Therefore the output files of allelic, dominance, and epistatic cannot always be used as input files. The files have the names "allelic_values.txt", "dominance_values.txt, and "epistatic_values.txt" and are stored in the simulation folder (parameter folder). If several replicates are simulated the files are generated for each replicate and a replicate counter (e.g. _r4) is inserted before the extension.

quanti_output [0,1] (default: 0)

0 : None. The files are not generated.

1 : Output. The allelic, the dominance, and/or the epistatic files are stored in the simulation folder (parameter folder), if they were used during the simulation.

CONFIDENTIAL

9.7 Summary statistics

The summary statistics listed in the table below are available for quantitative traits. The column Stat name contains the name of the summary statistic used to specify which summary statistics are computed (parameter stat, for details see section 4.2). These names appear also in the output file. The column Description contains a short description of the summary statistic.

Some of the summary statistics are available for adults and offspring (indicated by (adlt/off)). To obtain a certain summary statistic for adults the prefix adlt. has to be added to the summary statistic name (e.g. adlt.allnb), respectively the prefix off. to obtain the summary statistic for offspring (e.g. off.allnb).

Some of the summary statistics are computed for each patch separately. These statistics are characterized by a suffix "_p" to the Stat name and

by the words *(computed for each patch)* in the description of the statistic. Other summary statistics are computed for pairwise combinations of patches. These statistics are characterized by a suffix "_pair" to the Stat name and by the words *(all pairwise combinations computed)* in the description of the statistic. Note, that depending on the number of patches such a statistic may lead to a huge amount of output and a higher computation time. The names of such summary statistics in the output have the suffix "_pX" and "_pX-Y", respectively. Where X and Y are the index of the patches (starting with 1).

The summary statistics are computed by default for every quantitative trait. If several quantitative traits are simulated the postfix "_tT" is added to the summary statistic name in the output file, where T is the index of the trait. It is possible to compute statistics just for specified quantitative traits. This can be specified by the index of the trait inserted just after the *(*q). For example the stat option *(*q.adlt.fst) computes the Fst for all quantitative traits while the stat option *(*q2.adlt.fst) computes the Fst just for the second quantitative trait.

The summary statistic name (column Stat name) may be used to specify the summary statistic to be computed (e.g. q.varA). Similar summary statistics (within a thematic group) may be obtained at once using the name within square brackets after the group title (e.g. quanti). Using this group statistic name all summary statistics of the thematic group marked with a star (*) will be computed.

All summary statistics (and also all other outputs) are computed for the specified samples. By default all individuals of all populations are sampled. The chapter 7.4 describes how to specify a sampling schema.

CONFIDENTIAL

Table 9.1: Summary statistics available for quantitative traits

Stat name	Description
Quantitative trait statistics	[quanti, available only for adults]
q.VgW	genetic variance within patches*
q.VgB	genetic variance between patches*
q.VpW	phenotypic variance within patches*
q.VpB	phenotypic variance between patches*

Table 9.1 continued on next page

CONFIDENTIAL

Stat name	Description
q.VaW	additive genetic variance within patches
q.qst	Q_{ST}
q.qst.f	Q_{ST} corrected for inbreeding following Bonnin et al. (1996). Inbreeding coefficient F computed following Nei and Chesser (1983)
q.qst_pair	Q_{ST} between patch i and j (all pairwise combinations computed)
q.qst.f_pair	Q_{ST} between patch i and j (all pairwise combinations computed) corrected for inbreeding
q.varA_p	additive genetic variance of patch i following Lynch and Walsh (1998, p85-87) (computed for each patch)
q.meanG_p	genetic mean of patch i (computed for each patch)
q.varG_p	genetic variance of patch i (computed for each patch)
q.meanP_p	phenotypic mean of patch i (computed for each patch)
q.varP_p	phenotypic variance of patch i (computed for each patch)
Genotype coancestry [q.(adlt/off).coa]	
q.(adlt/off).theta	mean within patch coancestry*
q.(adlt/off).alpha	mean between patch coancestry*
q.(adlt/off).thetaFF	mean within patch, within females coancestry*
q.(adlt/off).thetaMM	mean within patch, within males coancestry*
q.(adlt/off).thetaFM	mean within patch, between sexes coancestry*
q.(adlt/off).coa.fsib	mean coancestry within full-siblings*
q.(adlt/off).coa.phsib	mean coancestry within paternal half-siblings*
q.(adlt/off).coa.mhsib	mean coancestry within maternal half-siblings*
q.(adlt/off).coa.nsib	mean coancestry within non-siblings*
q.(adlt/off).theta_p	mean coancestry within patch i (computed for each patch)
q.(adlt/off).alpha_pair	mean coancestry between patch i and j (all pairwise combinations computed)
Genetic diversity [q.(adlt/off).gendiv]	
number of alleles:	
q.(adlt/off).nbAll	mean across patches and loci*
q.(adlt/off).nbAll_p	mean across loci (computed for each patch)
q.(adlt/off).nbAll_l	mean across patches (computed for each locus)

Table 9.1 continued on next page

CONFIDENTIAL

Stat name	Description
q.(adlt/off).nbAll_p_l	(computed for each patch and locus)
q.(adlt/off).nbAll.tot	mean total across loci*
q.(adlt/off).nbAll.tot_l	total (computed for each locus)
number of fixed loci:	
q.(adlt/off).nbFixLoc	mean across patches and loci*
q.(adlt/off).nbFixLoc_p	mean across loci (computed for each patch)
q.(adlt/off).nbFixLoc_l	mean across patches (computed for each locus)
q.(adlt/off).nbFixLoc_p_l	(computed for each patch and locus)
q.(adlt/off).nbFixLoc.tot	mean total across loci*
q.(adlt/off).nbFixLoc.tot_l	total (computed for each locus)
allele frequencies (caution: any potential allele is outputted!):	
q.(adlt/off).a.freq	local (computed for each patch, locus and allele)
q.(adlt/off).a.freq.global	global (computed for each locus and allele)
locus gentoype frequencies (caution: any potential allele combination is outputted!):	
q.(adlt/off).l.freq	local (computed for each patch, locus genotype)
q.(adlt/off).l.freq.global	global (computed for each locus genotype)
observed heterozygosity following Nei and Chesser (1983):	
q.(adlt/off).ho	*
q.(adlt/off).ho_p	(computed for each patch)
q.(adlt/off).ho_l	(computed for each locus)
q.(adlt/off).ho_p_l	(computed for each patch and locus)
expected heterozygosity following Nei and Chesser (1983):	
q.(adlt/off).hs	*
q.(adlt/off).hs_p	(computed for each patch)
q.(adlt/off).hs_l	(computed for each locus)
q.(adlt/off).hs_p_l	(computed for each patch and locus)
q.(adlt/off).ht	total*
q.(adlt/off).ht_l	total (computed for each locus)
expected heterozygosity ($H = 1 - \sum p^2$):	
q.(adlt/off).hs.p2	
q.(adlt/off).hs.p2_p	(computed for each patch)
q.(adlt/off).hs.p2_l	(computed for each locus)
q.(adlt/off).hs.p2_p_l	(computed for each patch and locus)

Table 9.1 continued on next page

CONFIDENTIAL

Stat name	Description
q.(adlt/off).ht.p2	total*
q.(adlt/off).ht.p2_l	total (computed for each locus)
allelic richness following El Mousadik and Petit (1996):	
(rarefaction is based on the smallest sample size)	
q.(adlt/off).rs	
q.(adlt/off).rs_p	(computed for each patch)
q.(adlt/off).rs_l	(computed for each locus)
q.(adlt/off).rs_p_l	(computed for each patch and locus)
q.(adlt/off).rt	total
q.(adlt/off).rt_l	total (computed for each locus)
allelic range (difference between min and max allele):	
q.(adlt/off).r	
q.(adlt/off).r_p	(computed for each patch)
q.(adlt/off).r_l	(computed for each locus)
q.(adlt/off).r_p_l	(computed for each patch and locus)
q.(adlt/off).r.tot	total
q.(adlt/off).r.tot_l	total (computed for each locus)
garza-williamsons statistic following Garza and Williamson (2001):	
(modification: $gw = \sum(nb.allele)/\sum(1 + range)$)	
q.(adlt/off).gw	
q.(adlt/off).gw_p	(computed for each patch)
q.(adlt/off).gw_l	(computed for each locus)
q.(adlt/off).gw_p_l	(computed for each patch and locus)
q.(adlt/off).gw.tot	total
q.(adlt/off).gw.tot_l	total (computed for each locus)
F-statistics following Nei and Chesser (1983) [q.(adlt/off).fstat]	
q.(adlt/off).fst	global F_{ST}*
q.(adlt/off).fst_l	global F_{ST} (computed for each locus)
q.(adlt/off).fst_pair	pairwise F_{ST} between patch i and j (all pairwise combinations computed)
q.(adlt/off).fst_pair_l	pairwise F_{ST} between patch i and j (all pairwise combinations computed for each locus separately)

Table 9.1 continued on next page

CONFIDENTIAL

Stat name	Description
q.(adlt/off).fis	global F_{IS}*
q.(adlt/off).fis_l	global F_{IS} (computed for each locus)
q.(adlt/off).fit	global F_{IT}*
q.(adlt/off).fit_l	global F_{IT} (computed for each locus)
F-statistics following Weir and Cockerham (1984) [q.(adlt/off).fstat.wc]	
q.(adlt/off).fst.wc	global F_{ST}*
q.(adlt/off).fst.wc_l	global F_{ST} (computed for each locus)
q.(adlt/off).fst.wc_pair	pairwise F_{ST} between patch i and j (all pairwise combinations computed)
q.(adlt/off).fst.wc_pair_l	pairwise F_{ST} between patch i and j (all pairwise combinations computed for each locus separately)
q.(adlt/off).fis.wc	global F_{IS}*
q.(adlt/off).fis.wc_l	global F_{IS} (computed for each locus)
q.(adlt/off).fit.wc	global F_{IT}*
q.(adlt/off).fit.wc_l	global F_{IT} (computed for each locus)
Linkage disequilibirum reviewed in Devlin and Risch (1995)	
q.(adlt/off).Dprime	global *Dprime* (computed globally for each pair of loci)
q.(adlt/off).Dprime_pair	$Dprime_p$ (computed for each patch each pair of loci)
q.(adlt/off).Dstar	global *Dstar* (computed globally for each pair of loci)
q.(adlt/off).Dstar_pair	$Dstar_p$ (computed for each patch each pair of loci)
q.(adlt/off).R2	global $R2$ (computed globally for each pair of loci)
q.(adlt/off).R2_pair	$R2$ (computed for each patch and each pair of loci)
q.(adlt/off).Chi2	global $Chi2$ (computed globally for each pair of loci)
q.(adlt/off).Chi2_pair	$Chi2$ (computed for each patch and each pair of loci)

Table 9.1: Summary statistics available for quantitative traits continued

CONFIDENTIAL

Chapter 10

Neutral markers

CONFIDENTIAL

quantiNEMO also allows the simulation of neutral markers, such as microsatellites or SNPs with different mutation models (K allele and Stepwise). Different types of neutral markers can be combined within the same simulation. The initial allele frequencies can be defined for each population separately.

10.1 Architecture

ntrl_loci [integer]
This parameter specifies the number of neutral marker loci per individual. This parameter is mandatory for the simulation of a neutral marker.

ntrl_all [1-256/matrix] (default: 255)
This parameter specifies the maximal possible number of alleles per locus. Using a matrix it is possible to specify this number for each locus of a trait separately.

ntrl_allelic_file [string] (default: "")
This parameter allows to pass the name of a file containing allele informations, such as the initial allele frequencies, and/or the mutation probability to an allele. The number of alleles and loci has to be in line with the parameters ntrl_loci and ntrl_all. The information can

be set globally for all loci, if they have the same specifications, or for each locus separately. The allelic file for neutral markers has the same format as the allelic file for quantitative traits, but does not allow to specify the allelic effects:

```
#Allelic file
#################
[FILE_INFO]{
  col_locus 1
  col_allele 2
  col_mut_freq 3
  col_ini_freq 4
}

#locus   allele    mut_freq   ini_freq
1        1         0.20       {0   0.2}
1        2         0.25       {0   0.2}
1        3         0.20       {1   0.2}
1        4         0.20       {0   0.2}
1        5         0.20       {0   0.2}
2        1         0.20       {0   0.2}
2        2         0.25       {0   0.2}
2        3         0.20       {1   0.2}
2        4         0.20       {0   0.2}
2        5         0.20       {0   0.2}
```

CONFIDENTIAL

The file has to start with a file information box [FILE_INFO]{...}. This box contains the information of the structure of the following table allowing a flexible structure of the table. For example the order of the columns in the table may vary, or columns may be ignored. The file information box starts with the key word [FILE_INFO] and the information is enclosed by brackets "{...}". Line by line the index of the columns to be read have to be declared. Thereby a keyword for the specific setting is followed by the column index (the ordering starts with 1). The following column keywords are available:

col_locus This keyword specifies the column containing the locus index. If this column is not declared in the file information box, quantiNEMO will use the same settings for all loci. In this case the length of the table must meet the number of alleles (ntrl_all). If this keyword is declared the length of the table must meet the number of alleles times the number of loci (ntrl_loci * ntrl_all).

col_allele This keyword specifies the column containing the allele index. This column is mandatory. The index of the allele goes from 1 to ntrl_loci.

col_mut_freq This keyword specifies the column containing the mutation probabilities, i.e. the probability to mutate to this allele when a mutation occurs. The behavior of this mutation probability depends on the mutation model (see parameter (ntrl_mutation_model).

col_ini_freq This keyword specifies the column containing the initial frequencies of the alleles. This column allows to explicitly set the allele frequencies at the start of a simulation. The frequencies can be set for each patch separately using a matrix. In the example above, individuals of the first population are initially fixed for the allele 3 at the first locus as well as at the second locus. In the second population all alleles have the same initial frequency of 0.2. Note, that the matrix is adjusted in length if the number of populations does not correspond to the length of the matrix. If this column is not given the initial allele frequencies are set globally depending on the parameter ntrl_ini_allele_model.

Note, that as in all input files for quantiNEMO it is possible to define comments (also in the file information box) using the hash character: '#' or '#/ ...any text... /#'.

CONFIDENTIAL

10.2 Mutation

Mutation rates may be defined for each locus individually by explicitly defining the individual mutation rates (parameter ntrl_mutation_rate) or by defining the gamma distribution from which the individual mutation rates are drawn (parameters ntrl_mutation_rate and ntrl_mutation_var). There are two mutation models available. Using the allelic file for neutral markers (see section 10.1) it is possible to specify the probability to mutate to a certain allele explicitly for all alleles. Depending on the mutation model the new allele is the drawn one (KAM), or its index distance from the allele index in the middle is added to the current allele index (SSM). A minimal definition for mutations requires the setting of a common mutation rate (parameter ntrl_mutation_rate has a single value). In this case all loci have the same

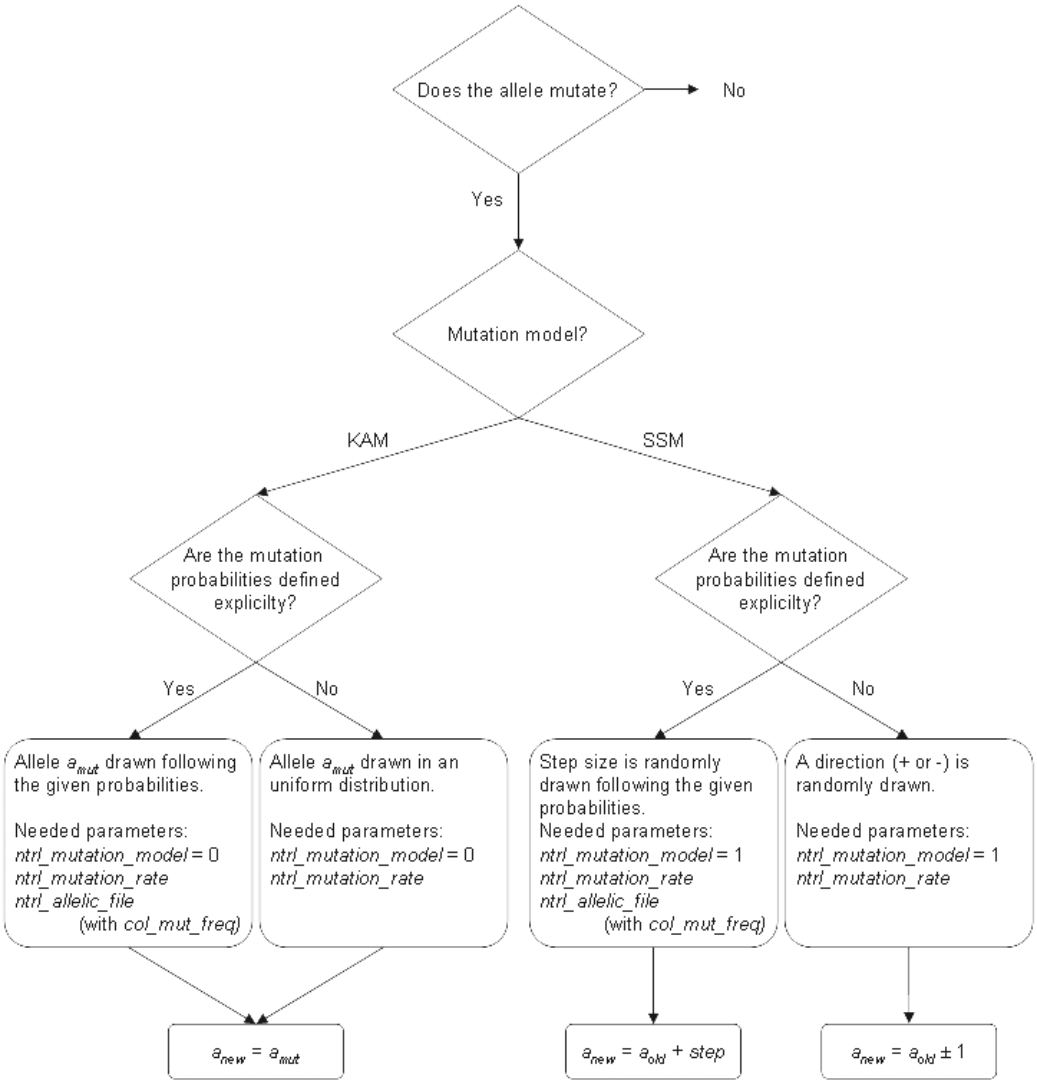


Figure 10.1: Schematic representation of the mutation process for a specific allele

mutation rate and the mutation model is KAM.

ntrl_mutation_model [0,1] (default: 0)
This parameter allows to specify the mutation model. The following mutation models are available:

0 : KAM (K-Allele Model)
At each mutation the existing allele is randomly exchanged by another allele within the range of alleles (i.e. $[1; ntrl\_all]$). By default the probability to mutate to any allele is the same. However, if the mutation probabilities are specified explicitly by the allelic file (see section 10.1) the probability to mutate to a certain allele depends on the specified probability to mutate to this allele.

1 : SSM (Single Step Mutation)
In contrast to the K-Allele-Model the new allele depends on the current allele. When a mutation occurs the current allele is replaced by one of its neighboring alleles (concerning the allele index). For example, if the allele with the index 12 mutates, it changes either to the allele with the index 13, or to the allele with the index 11. The boundaries are reflexive, i.e. the allele index can not exceed the range of alleles (i.e. $[1; ntrl\_all]$). In line with the Increment Mutation Model (IMM) for quantitative traits it is possible to specify mutation probabilities of the "steps to mutate" using the allelic file (see section 10.1). In this case the number of alleles (ntrl_all) has to be odd (as for the IMM). Then similar to the IMM the probabilities of the step to mutate has to be defined using the allelic file, where the step size is measured from the middle allele (concerning its index, i.e. $ntrl\_all/2$). The middle allele (with index $ntrl\_all/2$) has to have a mutation probability of 0 as a mutation of step zero is not a mutation. This allows generating any mutation pattern, as for instance the Generalized stepwise mutation model (Estoup et al., 2002).

ntrl_mutation_rate [decimal/matrix] (temporal/default: 0)
This parameter specifies the mutation rate per locus and generation. If the argument is a single value the mutation rate for all loci is the same. By passing a matrix of mutation rates it is possible to set the mutation rate for each single locus individually. By default no mutations occur.

CONFIDENTIAL

10.3 Initial genotpyes

There are several methods to set the allele frequencies or even the genotypes of the individuals at the start of a simulation (initialisation). The genotypes of the individuals may be set using an FSTAT file (Goudet, 1995) (parameter **ntrl_ini_genotypes**). If such a FSTAT file is not present the genotypes are randomly drawn following the explicitly set allele frequencies in the allelic file (see parameter **ntrl_allelic_file** and especially column keyword **col_ini_freq**). If the allele frequencies are not set explicitly in the allelic file the initialization is performed following the parameter **ntrl_ini_allele_model**.

ntrl_ini_genotypes [string] (default: "")
: This parameter allows to specify a name of an FSTAT file (Goudet, 1995) containing the initial genotypes of the individuals for each population. If such a file is present the initialization of the metapopulation is done solely by this file ignoring the parameters **ntrl_ini_allele_model** and **patch_ini_size**. A single FSTAT file is needed for all types of neutral markers together containing the total number of loci of all neutral markers together. Note, that quantiNEMO allows to output an appropriate file for any generation (see parameter **ntrl_save_genotype**). This allows to resume a simulation, to generate tailored initial conditions, or to continue a simulation with modified settings. If the parameter **ntrl_save_genotype** is set to 2 an extended FSTAT file is generated. Also this file may be used to initialize the metapopulation. In this case quantiNEMO overtakes the supplement information provided by the file, especially the sex and age of the individual, the index of the individual, its mother and father. Thus the supplement information allows to resume a simulation without the loss of the pedigree. Note, that for an entire resume of a simulation also the genotypes of the quantitative traits have to be set (see parameter **quanti_save_genotype**). In this case the number of individuals per patch and the individuals supplement information (sex, age, individuals id, mothers id, fathers id must be in agreement with each other.

ntrl_ini_allele_model [0,1] (default: 0)
: If the genotypes or allele frequencies are not already defined in another way, the initialization of the genotypes may be either polymor-

CONFIDENTIAL

phic, where the probability of each allele is identically or monomorphic, where all populations are fixed for a single allele.

0 : polymorph. The populations are maximally polymorph in respect to allele frequencies at the start of a simulation.

1 : monomorph. The populations are monomorph in respect to allele frequencies at the start of a simulation. All individuals are fixed for a single allele, which is the "middle" allele, i.e. the allele with the index $\lfloor$ntrl_all/2$\rfloor$.

10.4 Genotype output

The neutral genotype of all sampled individuals and populations may periodically be dumped to files similar to the genotype of quantitative traits. The output files will be stored in the folder given by the parameter ntrl_genot_dir and will have the name of the base file name (see parameter filename in section 5). The extension is ".dat". A counter for the generation (e.g. _g05) and the replicate (e.g. _r4) is inserted before the extension. By default all individuals of all populations are sampled. The chapter 7.4 describes how to specify a sampling schema. An example of such a file name is "simulation_g05_r4.dat". Note, that such a genotype file may be used to start a new simulation (see parameter **ntrl_ini_genotypes**).

CONFIDENTIAL

ntrl_save_genotype [0-5] (default: 0)
This parameter specifies the output of the neutral genotype.

0 : None. No output is generated.

1 : FSTAT. Genotypes are outputted in the FSTAT format (Goudet, 1995).

2 : FSTAT extended. Same as point 1, but the file contain the following six additional columns: the age class (1 = offspring, 2 = adult), the sex (0 = male; 1 = female), the ID of the individual, the ID of the mother, the ID of the father, and the fitness of the individual. The ID is a unique identifier for each individual of a simulation in the format "345_23", meaning that this is the 345th individual born in patch 23. The IDs of the individual, the

mother and the father allow to extract pedigree informations, if the output is stored for each generation, and also to investigate the migration behavior of the individual and its parents.

3 : Arlequin. Genotypes are outputted in Arlequin format (Excoffier, 2010).

4 : Arlequin extended. Same as point 3, but with additional commented individual information as in point 2.

5 : PLINK. Outputs the **neutral** genotypes with the phenotypes of the quantitative traits (Purcell et al., 2007). The standard two files .ped and .map are created. The 6th column (phenotype) of the .ped file contains the fitness of the individual. If quantitative traits are simulated, their phenotypes are listed in an alternate phenotype file (.pheno). The alleles of all bi-allelic loci (ntrl_nb_allele set to 2) are listed. The .map file is generated for each replicate. The .ped and .pheno files are generated for each specified time point and outputs of successive time points are concatenated to a single file, allowing to obtain entire or parts of pedigrees. The filename of such a concatenated file contains the time stamp of the first entry. Note that only successive individuals list their parentIDs. To generate an entire pedigree the entire populations have to be sampled.

6 : PLINK extended. Same as point 5, but all **neutral** genotypes are listed, including the multi-allele loci.

CONFIDENTIAL

An example of such a file (ntrl_save_genotype is set to 2):

```
5 4 20 2
t1_l1
t1_l2
t1_l3
t1_l4
1 1415 1019 2002 0820 1 1 10_1 1_1 0_1 0.345
1 0814 0219 2002 2020 1 1 11_1 8_1 2_4 0.334
1 0808 0217 1902 0820 1 1 12_1 5_3 5_1 0.123
...
5 1004 0917 1404 1007 1 1 16_5 9_5 3_2 0.999
5 2017 1010 2013 1812 1 0 17_5 3_2 9_2 1.000
5 2017 1008 2013 1811 1 1 11_4 8_2 9_2 0.678
```

The first line contains the number of patches (5 patches here), the number of loci (5), the highest possible allele index (20), and the number of digits used to write each allele (2). The next four lines contain the locus names. The following lines contain the individual's info, one individual per line. The first number is the patch number of the individual, followed by the genotype. Each column represents a locus, and the first half of the locus represent the first allele index, while the second half of the locus the second allele at the given locus. In this example, we are using two digit per allele, the first two digits of a locus genotype number are the index of the first allele (e.g. allele 14 for the first allele of the first locus of the first individual) while the two next digits are the index of the second allele (e.g. allele 15 for the second allele of the first locus of the first individual). Each line ends with six columns consisting supplementary information on the individual (see above) if the parameter ntrl_save_genotype is set to 2.

ntrl_genot_dir [string] (default: "")
This parameter allows to specify the subdirectory where the genotypes are stored. If not specified (default) the output is stored in the simulation folder (parameter folder).

ntrl_genot_filename [string] (default: "")
This parameter is used to specify an individual base filename for the genotypes. If not specified the generic base filename will be used (see parameter filename).

ntrl_genot_logtime [integer] (temporal/default: 1)
This parameter specifies the interval of the genotype output. Since the parameter may change over time the output may be generated at any generation.

ntrl_genot_script [string] (default: "")
It is possible to launch a script just after the genotype file is generated. The argument of the parameter is the file name of the script. The name of the genotype file is passed as unique parameter to the script.

ntrl_genot_sex [0-2] (default: 0)
This parameter allows to choose which sex is output.

0 : Both. Output includes both sexes.

CONFIDENTIAL

1 : Females. Output includes only female genotypes.

2 : Males. Output includes only male genotypes.

ntrl_genot_age [0-2] (default: 0)
This parameter allows to choose which age is output.

0 : Adults. Output includes only adult genotypes.

1 : Juveniles. Output includes only juvenile genotypes.

2 : Both. Output includes juveniles and adults genotypes.

10.5 Summary statistics

CONFIDENTIAL

The summary statistics listed in the table below are available for neutral markers. The column Stat name contains the name of the summary statistic used to specify which summary statistics are computed (parameter stat, for details see section 4.2). These names appear also in the output file. The column Description contains a short description of the summary statistic.

Some of the summary statistics are available for adults and offspring (indicated by (adlt/off)). To obtain a certain summary statistic for adults the prefix adlt. has to be added to the summary statistic name (e.g. adlt.allnb), respectively the prefix off. to obtain the summary statistic for offspring (e.g. off.allnb).

Some of the summary statistics are computed for each patch separately. These statistics are characterized by a suffix "_p" to the Stat name and by the words *(computed for each patch)* in the description of the statistic. Other summary statistics are computed for pairwise combinations of patches. These statistics are characterized by a suffix "_pair" to the Stat name and by the words *(all pairwise combinations computed)* in the description of the statistic. Note, that depending on the number of patches such a statistic may lead to a huge amount of output and a higher computation time. The names of such summary statistics in the output have the suffix "_pX" and "_pX-Y", respectively. Where X and Y are the index of the patches (starting with 1).

The summary statistics are computed by default for every type of neutral marker. If several neutral marker types are simulated the postfix "_tT" is

added to the summary statistic name in the output file, where T is the index of the neutral marker type. It is possible to compute statistics just for specified types of neutral markers. This can be specified by the index of the type inserted just after the (n). For example the stat option (n.adlt.fst) computes the Fst for all types of neutral markers while the stat option (n2.adlt.fst) computes the Fst just for the second neutral marker type.

The summary statistic name (column Stat name) may be used to specify the summary statistic to be computed (e.g. adlt.fst). Similar summary statistics (within a thematic group) may be obtained at once using the name within square brackets after the group title (e.g. adlt.fstat). Using this group statistic name all summary statistics of the thematic group marked with a star (*) will be computed.

All summary statistics (and also all other outputs) are computed for the specified samples. By default all individuals of all populations are sampled. The chapter 7.4 describes how to specify a sampling schema.

CONFIDENTIAL

Table 10.1: Summary statistics available for neutral markers

Stat name	Description
Genotype coancestry [n.(adlt/off).coa]	
n.(adlt/off).theta	mean within patch coancestry*
n.(adlt/off).alpha	mean between patch coancestry*
n.(adlt/off).thetaFF	mean within patch, within females coancestry*
n.(adlt/off).thetaMM	mean within patch, within males coancestry*
n.(adlt/off).thetaFM	mean within patch, between sexes coancestry*
n.(adlt/off).coa.fsib	mean coancestry within full-siblings*
n.(adlt/off).coa.phsib	mean coancestry within paternal half-siblings*
n.(adlt/off).coa.mhsib	mean coancestry within maternal half-siblings*
n.(adlt/off).coa.nsib	mean coancestry within non-siblings*
n.(adlt/off).theta_p	mean coancestry within patch i (computed for each patch)
n.(adlt/off).alpha_pair	mean coancestry between patch i and j (all pairwise combinations computed)
Genetic diversity [n.(adlt/off).gendiv]	
number of alleles:	

Table 10.1 continued on next page

Stat name	Description
n.(adlt/off).nbAll	mean across patches and loci*
n.(adlt/off).nbAll_p	mean across loci (computed for each patch)
n.(adlt/off).nbAll_l	mean across patches (computed for each locus)
n.(adlt/off).nbAll_p_l	(computed for each patch and locus)
n.(adlt/off).nbAll.tot	mean total across loci*
n.(adlt/off).nbAll.tot_l	total (computed for each locus)
number of fixed loci:	
n.(adlt/off).nbFixLoc	mean across patches and loci*
n.(adlt/off).nbFixLoc_p	mean across loci (computed for each patch)
n.(adlt/off).nbFixLoc_l	mean across patches (computed for each locus)
n.(adlt/off).nbFixLoc_p_l	(computed for each patch and locus)
n.(adlt/off).nbFixLoc.tot	mean total across loci*
n.(adlt/off).nbFixLoc.tot_l	total (computed for each locus)
allele frequencies (caution: any potential allele is outputted!):	
n.(adlt/off).a.freq	local (computed for each patch, locus and allele)
n.(adlt/off).a.freq.global	global (computed for each locus and allele)
locus gentoype frequencies (caution: any potential allele combination is outputted!):	
n.(adlt/off).l.freq	local (computed for each patch, locus genotype)
n.(adlt/off).l.freq.global	global (computed for each locus genotype)
observed heterozygosity following Nei and Chesser (1983):	
n.(adlt/off).ho	*
n.(adlt/off).ho_p	(computed for each patch)
n.(adlt/off).ho_l	(computed for each locus)
n.(adlt/off).ho_p_l	(computed for each patch and locus)
expected heterozygosity following Nei and Chesser (1983):	
n.(adlt/off).hs	*
n.(adlt/off).hs_p	(computed for each patch)
n.(adlt/off).hs_l	(computed for each locus)
n.(adlt/off).hs_p_l	(computed for each patch and locus)
n.(adlt/off).ht	total*
n.(adlt/off).ht_l	total (computed for each locus)
expected heterozygosity ($H = 1 - \sum p^2$):	
n.(adlt/off).hs.p2	

Table 10.1 continued on next page

CONFIDENTIAL

Stat name	Description
n.(adlt/off).hs.p2_p	(computed for each patch)
n.(adlt/off).hs.p2_l	(computed for each locus)
n.(adlt/off).hs.p2_p_l	(computed for each patch and locus)
n.(adlt/off).ht.p2	total*
n.(adlt/off).ht.p2_l	total (computed for each locus)
allelic richness following El Mousadik and Petit (1996):	
(rarefaction is based on the smallest sample size)	
n.(adlt/off).rs	
n.(adlt/off).rs_p	(computed for each patch)
n.(adlt/off).rs_l	(computed for each locus)
n.(adlt/off).rs_p_l	(computed for each patch and locus)
n.(adlt/off).rt	total
n.(adlt/off).rt_l	total (computed for each locus)
allelic range (difference between min and max allele):	
n.(adlt/off).r	
n.(adlt/off).r_p	(computed for each patch)
n.(adlt/off).r_l	(computed for each locus)
n.(adlt/off).r_p_l	(computed for each patch and locus)
n.(adlt/off).r.tot	total
n.(adlt/off).r.tot_l	total (computed for each locus)
garza-williamsons statistic following Garza and Williamson (2001):	
(modification: $gw = \sum(nb.allele)/\sum(1 + range)$)	
n.(adlt/off).gw	
n.(adlt/off).gw_p	(computed for each patch)
n.(adlt/off).gw_l	(computed for each locus)
n.(adlt/off).gw_p_l	(computed for each patch and locus)
n.(adlt/off).gw.tot	total
n.(adlt/off).gw.tot_l	total (computed for each locus)
F-statistics following Nei and Chesser (1983) [n.(adlt/off).fstat]	
n.(adlt/off).fst	global F_{ST}*
n.(adlt/off).fst_l	global F_{ST} (computed for each locus)
n.(adlt/off).fst_pair	pairwise F_{ST} between patch i and j (all pairwise combinations computed)

Table 10.1 continued on next page

CONFIDENTIAL

Stat name	Description
n.(adlt/off).fst_pair_l	pairwise F_{ST} between patch i and j (all pairwise combinations computed for each locus separately)
n.(adlt/off).fis	global F_{IS}*
n.(adlt/off).fis_l	global F_{IS} (computed for each locus)
n.(adlt/off).fit	global F_{IT}*
n.(adlt/off).fit_l	global F_{IT} (computed for each locus)
F-statistics following Weir and Cockerham (1984) [n.(adlt/off).fstat.wc]	
n.(adlt/off).fst.wc	global F_{ST}*
n.(adlt/off).fst.wc_l	global F_{ST} (computed for each locus)
n.(adlt/off).fst.wc_pair	pairwise F_{ST} between patch i and j (all pairwise combinations computed)
n.(adlt/off).fst.wc_pair_l	pairwise F_{ST} between patch i and j (all pairwise combinations computed for each locus separately)
n.(adlt/off).fis.wc	global F_{IS}*
n.(adlt/off).fis.wc_l	global F_{IS} (computed for each locus)
n.(adlt/off).fit.wc	global F_{IT}*
n.(adlt/off).fit.wc_l	global F_{IT} (computed for each locus)
Linkage disequilibirum reviewed in Devlin and Risch (1995)	
n.(adlt/off).Dprime	global *Dprime* (computed globally for each pair of loci)
n.(adlt/off).Dprime_pair	$Dprime_p$ (computed for each patch each pair of loci)
n.(adlt/off).Dstar	global *Dstar* (computed globally for each pair of loci)
n.(adlt/off).Dstar_pair	$Dstar_p$ (computed for each patch each pair of loci)
n.(adlt/off).R2	global $R2$ (computed globally for each pair of loci)
n.(adlt/off).R2_pair	$R2$ (computed for each patch and each pair of loci)
n.(adlt/off).Chi2	global $Chi2$ (computed globally for each pair of loci)
n.(adlt/off).Chi2_pair	$Chi2$ (computed for each patch and each pair of loci)

Table 10.1: Summary statistics available for neutral markers continued

CONFIDENTIAL

Chapter 11

Multiple traits

CONFIDENTIAL

quantiNEMO allows to simulate multiple traits (quantitative and/or neutral) simultaneously. Each trait has its own architecture and quantitative traits may be under different selection pressures.

(quanti/ntrl)_nb_trait [integer] (default: 1)
This parameter defines the number of traits.

Each trait may have its own specifications, but it is also possible to specify parameters for some traits together, named grouping. If several traits are used it is possible to address a certain trait by its number. For instance to specify a parameter for the fifth trait one has to append a `"_5"` to the parameter name. In contrast if for the fifth trait no parameter with the suffix `"_5"` is passed quantiNEMO checks if the parameter is passed for the fourth trait (suffix `"_4"`). If this is also not the case quantiNEMO checks if the parameter is passed for the third trait (suffix `"_3"`), and so forth until a parameter is found. Note, that a parameter without a suffix is the same as the parameter with the suffix `"_1"`. This behavior of quantiNEMO allows specifying parameters for a group of traits. Sometimes this grouping is not desired. It can be suppressed by setting 'NOT_SET' as an argument. In this case this parameter and all parameters inheriting this parameter will use the default argument. An example for quantitative traits may make it clearer:

```
quanti_nb_trait        12

quanti_loci            5
```

```
quanti_loci_7           10

quanti_all_1            10
quanti_all_4            20
quanti_all_7            10
quanti_all_10           20

quanti_allelic_file_1  file_1.txt
quanti_allelic_file_4  file_2.txt
quanti_allelic_file_7 NOT_SET

quanti_mutation_rate 0.0001
```

In this example we simulate 12 quantitative traits. Traits 1 to 3 consist of 5 loci with up to 10 alleles, traits 4 to 6 consist of 5 loci with up to 20 alleles, traits 7 to 9 consist of 10 loci with up to 10 alleles, and traits 10 to 12 consist of 10 loci with up to 20 alleles. All traits have the same mutation rate of 0.0001. Allele characteristics for traits 1 to 3 are specified in file âĂIJfile_1.txtâĂİ, allele characteristics for the traits 4 to 6 in file âĂIJfile_2.txtâĂİ. Since 'NOT_SET' is set for the parameter quanti_allelic_file_7 all allele characteristics for traits 7 to 12 are set automatically without any file.

quantiNEMO allows simulating pleiotrophy for quantitative traits. For more details please have a look at the chapter **??**.

CONFIDENTIAL

Chapter 12

Genetic map

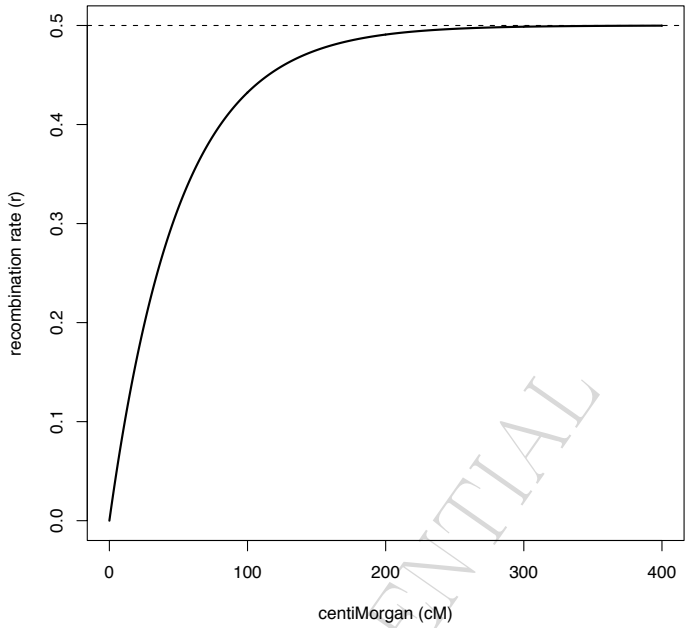
CONFIDENTIAL

quantiNEMO has an underlying genetic map, which may consist of several chromosomes. This allows explicitly positioning all types of loci on the map (quantitative trait loci (QTL) and neutral markers). The unit of the genetic map is centi Morgans (Haldane, 1919, cM). This means that between two loci separated by 100cM on average a single recombiantion event per meiosis is expected. The relation between centiMorgan (m) and recombination rate (r) is given as

$$r = \frac{1 - e^{-2m/100}}{2}$$

.

To simulate linked loci two types of parameters have to be set: One to specify the locus positions on the genome (in cM) and one for each trait (QTL or neutral) to attribute the loci of the trait to these locus positions. If a locus of a trait is not attributed to a position then quantiNemo2 assumes that this locus is unlinked to any other locus. If a position of a locus is assigned several times to a locus (only valid for quantitative traits) then this is a pleiotropic QTL and thus all related quantitative traits pleiotrop (see section 9.1.4). A genome is identical for everybody, however sexes may differ in the rate of recombinations, e.g. hotspots of recombiantion, or degenerated Y-chromosome (see also parameter **sex_ratio_threshold**). quantiNemo2 accounts for this by allowing specifying different genetic maps (in cM) for each sex using the suffixes **_fem** and **_mal**. Note, that in this case the order of loci must be identical, only distances between loci may differ.



CONFIDENTIAL

genome
genome_fem
genome_mal [matrix] (default: "")

These parameters allow specifying the positions of loci in centi Morgans for all traits in common. Brackets separate different chromosomes and within a chromosome the positions of loci are defined cumulatively, i.e. the position of the locus is defined as the length to the start of the chromosome. Note, that chromosomes may contain different numbers of loci. It is possible to skip chromosomes if they don't contain loci (see section 3.4). If the genetic map is sex specific both parameters have to be set. If hermaphrodites are simulated the parameters **genome** and **genome_fem** are identical. If all three parameters are set, only the sex specific parameters will be used. The genome may only be specified in general (i.e. parameter **genome**) or for each type of trait separately (i.e. parameters **quanti_genome** and **ntrl_genome**).

(quanti/ntrl)_genome

(quanti/ntrl)_genome_fem
(quanti/ntrl)_genome_mal [matrix] (default: "")

Similar to the parameters above these parameters allow specfiying the positions of loci in centi Morgans. In contrast to the parameters above these parameters allow specifiying the positions for each type of trait separately.

Example:

```
genome { {1: 10}
         {3: 10 40 60 80 100}
         {   10 40 60 80 100} }
```

In the example above, the genetic map consists of 11 loci located on three of at least four chromosomes. The first chromosome contains a single locus at position 10 cM. No loci are located on the second chromosome. The third and fourth chromosomes have 5 loci at positions 10, 40, 60, 80, and 100 cM.

(quanti/ntrl)_locus_index [matrix] (default: "")

This parameter allows assigning each locus of a trait (quanti or ntrl) to a locus position on the genome (see above). The parameter has to be set for each trait separately and the assignment of the locus to a locus position on the genome is made based on the index of the locus on the entier genome (starting with 1). Note, that the indexing is ignoring any chromosomal structure. Note also, that the indexing depends on the corresponding genome, i.e. if it is specified in general (parameter **genome**) or trait type specific (e.g. parameter **quanti_genome**). Note further, that this parameter is not sex specific, since the order of the loci is identical for both sexes.

Example 1 (trait type sepcific):

```
quanti_nb_trait       2
quanti_loci           3
quanti_genome         {0.1 0.2 0.3 0.4 0.5}
quanti_locus_index_1  {1 2 5}
quanti_locus_index_2  {3 4 5}
ntrl_nb_trait         2
ntrl_genome           {0.1 0.6 0.7 0.8 0.9 1.0}
ntrl_locus_index_1    {1 2 5}
```

CONFIDENTIAL

```
ntrl_locus_index_2    {3 4 6}
```

Example 2 (general):

```
genome                {0.1 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0}
quanti_nb_trait       2
quanti_loci           3
quanti_locus_index_1  {1 3 6}
quanti_locus_index_2  {4 5 6}
ntrl_nb_trait         2
ntrl_locus_index_1    {2 7 10}
ntrl_locus_index_2    {8 9 11}
```

Both examples are identical, the genome is defined once for each type of trait separately (Example 1), and once in general (Example 2). In the example we have 5 QTLs and 6 neutral markers. The last QTL (at 0.5 cM) is pleiotroph coding for both quantitative traits, and thus quantitative traits 1 and 2 are pleiotroph. At 0.1 cM we have a QTL and a neutral marker without any recombiantion between them.

recombination_factor
recombination_factor_fem
recombination_factor_mal [decimal/matrix] (temporal/default: 1)

These parameters allow setting a factor for the recombination rate. This factor is then multiplied with the positions of the loci on the chromosomes allowing easily stretching or shrinking a chromosome, keeping the relation between the loci intact. These parameters allow simulating the evolution of the recombination rate. To achieve this it is possible to use the phenotype (Z) or the genotypic value (G) of a quantitative trait as recombination factor. Thereby the key character G and Z are followed directly by the quantitative trait index.

Example:

```
quanti_genome          {1 4}{1 10}
recombination_factor_mal {2 0.5}
recombination_factor_fem {Z1 Z2}
```

In this example the genome consists of two chromosomes containing each 2 loci. A recombination factor is set separately for females and

CONFIDENTIAL

males. The recombination rate for males is fix and modifies the positions of the first chromosome by a factor of two (increasing the recombination rate among the loci) and by a factor of 0.5 for the second chromosome (reducing the recombination rate among the loci). In contrast the recombination factor for females is variable and is given by the phenotype of the first and second quantitative trait for the first and second chromosome, respectively.

CONFIDENTIAL

Bibliography

Beverton, R. J. H. and Holt, S. J. (1957), On the dynamics of exploited fish populations, Technical report, U.K. Ministry of Agriculture and Fisherie.

Bonnin, I., Prosperi, J. M. and Olivieri, I. (1996), 'Genetic markers and quantitative genetic variation in medicago truncatula (leguminosae): A comparative analysis of population structure', *Genetics* **143**, 1795–1805.

Bürger, R. (2000), *The Mathematical Theory of Selection, Recombination, and Mutation*, Wiley, Cichester, UK.

Devlin, B. and Risch, N. (1995), 'A comparison of linkage disequilibrium measures for fine-scale mapping', *Genomics* **29**, 311–322.

El Mousadik, A. and Petit, R. J. (1996), 'High level of genetic differentiation for allelic richness among populations of the argan tree [argania spinosa (l.) skeels] endemic to morocco', *Theoretical and Applied Genetics* **92**, 832–839.

Estoup, A., Jarne, P. and Cornuet, J. M. (2002), 'Homoplasy and mutation model at microsatellite loci and their consequences for population genetics analysis', *Molecular Ecology* **11**(9), 1591–1604.

Excoffier, L. L. H. (2010), 'Arlequin suite ver 3.5: A new series of programs to perform population genetics analyses under linux and windows', *Molecular Ecology Resources* **10**(3), 564–567.

Garza, J. C. and Williamson, E. G. (2001), 'Detection of reduction in population size using data from microsatellite loci', *Molecular Ecology* **10**, 305–318.

Goudet, J. (1995), 'Fstat (version 1.2): A computer program to calculate f-statistics', *Journal of Heredity* **86**(6), 485–486.

CONFIDENTIAL

Guillaume, F. and Rougemont, J. (2006), 'Nemo: an evolutionary and population genetics programming framework', *Bioinformatics* **22**(20), 2556–2557.

Haldane, J. B. S. (1919), 'The combination of linkage values, and the calculation of distances between loci of linked factors. journal of genetics', *Journal of Genetics* **8**, 299–309.

Lynch, M. and Walsh, B. (1998), *Genetics and analysis of quantitative traits*, Publisher Sinauer Associates Inc.

Nei, M. and Chesser, R. K. (1983), 'Estimation of fixation indexes and gene diversities', *Annals of Human Genetics* **47**(Jul), 253–259.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M., Bender, D., Maller, J., Sklar, P., de Bakker, P., Daly, M. and Sham, P. (2007), 'Plink: a toolset for whole-genome association and population-based linkage analysis', *American Journal of Human Genetics* **81**, 559–575.

Ravigne, V., Olivieri, I. and Dieckmann, U. (2004), 'Implications of habitat choice for protected polymorphisms', *Evolutionary Ecology Research* **6**(1), 125–145.

Richards, F. (1959), 'A flexible growth function for empirical use', *J. Exp. Bot* **10**, 290–300.

Wallace, B. (1968), Polymorphism, population size, and genetic load, *in* R. C. Lewontin, ed., 'Population biology and evolution', Syracuse University Press, Syracuse, N. Y., pp. 87–108.

Wallace, B. (1975), 'Hard and soft selection revisited', *Evolution* **29**, 465–473.

Weir, B. S. and Cockerham, C. C. (1984), 'Estimating f-statistics for the analysis of population structure', *Evolution* **38**, 1358 – 1370.

CONFIDENTIAL

Index

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL

CONFIDENTIAL