AgeStrucNe User Manual

October 26, 2018

Version: 1.0

Authors: Ted Cosart and Brian Hand

Contact: agestrucne@gmail.com

List of Figures

List of Tables

AgeStrucNe

0.1   Introduction

The AgeStrucNe GUI interface offers a user interface to allow easy access to simuPOP-based simulations [1] and the LDNe-based Nb and Ne estimations, using version 2 of the LDNe program [3]. The program integrates this functionality as implemented in Tiago Antao's python program, AgeStrucureNe, available at `https://github.com/tiagoantao/AgeStructureNe.git`.

We also offer an interface for plotting Nb and Ne estimations, and regressions based on the estimations. Our program offers a separate interface for each of three functions: population simulation, Nb and Ne estimation, and estimate visualization. The genepop file output from a simulation can be loaded into an Nb estimation interface, and in turn, the output from an Nb estimation can be loaded into a visualization interface. The Nb estimation interface can also use any genepop file for input.

For program questions, please contact us through the program's email account at agestrucne@gmail.com.

0.2   Installation

Details about our program's dependancies and installation are in our github project's README.md file, inserted here for convenience, and also available at `https://github.com/popgengui/agestrucne/blob/data/README.md`:

0.2.1   README file

0.3   License

Our license information follows, and is also available at `https://github.com/popgengui/agestrucne/blob/data/LICENSE`.

0.3.1   License file

.

0.4   Starting the program

The program is launched at a terminal using a python 2.7, 3.5, or 3.6 executable, invoking the "agestrucne" executable. Please see the README.md file for instructions on installing the program.
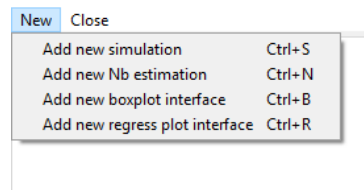
Figure 0.1: Adding an interface

Once you have downloaded the "data" branch of our program's github repository, you can, for example, open a terminal, change your current directory to the agestrucne folder supplied by the download, and load a configuration file into the Simulation interface (Section 0.5).

0.5 Loading interfaces

To load one or more of the three interfaces for performing simulations, Nb/Ne estimations, or plotting programs, from the main menu click on the "New" menu (Figure 0.1). You can load any number of interfaces and run them simultaneously, though caution is warranted, since a computer can be taxed to a standstill by the over-allocation of CPUs and/or memory.

0.6 Running a simulation

0.6.1 Load an interface

Use the add menu (Figure0.1) to load a new simulation interface. Steps for preparing the interface to run a simulation follow.

0.6.2 Load a configuration file.

The initial simulation interface requires the user to load a configuration file (Figure 0.2). If you have not already downloaded our collection of configuration files, you can get them at our programs github repository, downloading our data branch at `https://github.com/popgengui/agestrucne/tree/data`. The files are inside the "configuration_files" subdirectory, inside the main program directory, "ageststrucne." You can also load your own configuration (see our provided configuration files for a formatted example and also see the section "Manually configuration files"). Note that you can also open these files and change the parameters manually, if you prefer it to setting them in the interface

0.6.3 Adjust simulation parameters

With a configuration file loaded (Section 0.6.2) you can change the values in the editable controls. These are detailed below in the "Simulation Input" section.
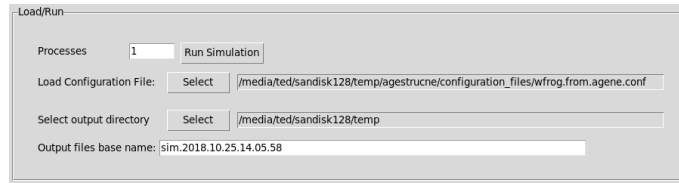
Figure 0.2: Simulation interface, Load/Run section.

0.6.4 Start the simulation

Clicking "run simulation," (Figure 0.2) starts the simulation with the loaded parameters. The button's text changes to say "cancel simulation," and next to it a new label notes that a simulation is in progress. While the simulation is in progress, the parameter controls are disabled.

0.6.5 Running a simulation from a terminal

The module pgdrivesimulation.py provides a command line interface to run a simulation. At a Linux or DOS terminal, you can invoke the command with the form,

$$\langle\text{python}\rangle\ \langle\text{pgdrivesimulation.py}\rangle\ \langle\text{options}\rangle$$

Where ⟨python⟩ is either the python3 or python2.7 executable, ⟨pgdrivesimulation.py⟩ is the module name, which may also include the full path the to the module (see the program installation instructions in our README.md file, distributed in our github repostory's main directory), and ⟨options⟩ is a list of parameters specified using the option flags. You can see the list of option flags by invoking the command with out any arguments. To see the details for each argument, execute:

$$\langle\text{python}\rangle\ \langle\text{pgdrivesimulation.py}\rangle\ -\text{h}$$

Note that there are both required and optional arguments. The latter are only a select number of those offered in the GUI interface (Section 0.7).

0.7 Simulation input

The simulation interface is divided into controls inside sub-frames, based on category.

0.7.1 The Load/Run section

This section offers parameters related to input and output files (Figure 0.2).

0.7.1.1 Processes

Valid values are between 1 and the total number of available (logical) cores in your computer. Multiple processes are only useful if you have set the "Replicates" parameter (see the Simulation subframe details below) to a value greater than one.
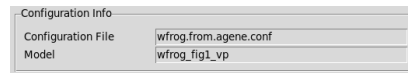
Figure 0.3: Simulation interface, Configuraton info.

0.7.1.2 Configuration File

Press the "Select" button next to the label, "Load Configuration File" to load a configuration file into the interface. We have included configuration files for many species. These can be found in the "configuration_files" subdirectory inside the main program folder.

0.7.1.3 Output directory

Press the select button next to the label, Select output directory, select a folder for the output files written by the simulation

0.7.1.4 Output files base name

You can type in a base name for the simulation output files. The simulation will prepend this to the *.genepop, *.conf, *_age_totals.tsv and *_nb_values.tsv output files (Section 0.9).

0.7.2 Configuration Info section

This parameter group simply shows you the input file information and has no settable parameters (Figure 0.3).

0.7.2.1 Configuration file name

This gives the file name of the loaded configuration file.

0.7.2.2 Model name

This gives the name of the model parameterized by the configuration file. In our example configuration files, the model name is usually a species' common name. This parameter was used when the program depended on separate life table and configuration files. Currently the configuration files contain all of the life table information needed for the simulations, so that the model name can be any string you find suitable, to indicate, for example, that a given configuration file has parameters similar to another with the same name.

0.7.3 The Population section

offers many parameter settings that characterize the population's size and fecundity (Figure 0.4).

0.7.3.1 N0 (Newborns)

. This gives the number of newborns added at each simulated reproductive cycle. This value is not editable directly, but is calculated using several values,

Figure 0.4: Simulation interface, population section

all of which are editable. These including Nb, Nb/Nc , Female, Male Survival, and the probability of male birth. The N0 is recalculated whenever any of these values changes, using the following procedure:

1. Assign an Nc value, as Nb divided by Nb/Nc.

2. Assign a current_male_proportion equal to the Probability of male birth.

3. Assign a current_female_proportion equal to 1 – the Probability of male birth.

4. Assign a cumulative_proportion=1.

5. For each age value age_val giving a male and female survival rate:

   a Update, current_male_proportion=current_male_proportion x male_survival at age_val.

   b Update, current_female=current_female x female_survival at age_val.

   c Update cumulative_proportion =cumulative_proportion + current_male_proportion.

   d Update cumulative_proportion=cumulative_proportion + current_female_proportion.

6. Set N0=Nc/cumulative_proportion, rounding it to the nearest integer.

0.7.3.2   Nb/Nc

This is the effective number of breeders in one reproductive cycle divided by the census size.

0.7.3.3   Nb/Ne

This is the ratio of the effective number of breeders in one reproductive cycle to the effective population size per generation. This value is not used in the simulation itself, but is written to the output genepop file, and can be used in the Nb estimation interface to make a bias correction in the LDNe estimation of Nb (see section 0.11.1.6).

0.7.3.4   Nb

This is the target effective number of breeders in the simulated population.

0.7.3.5   Nb Tolerance

This determines the threshold for allowable "true" Nb values (see section 0.7.3.4) for simulated populations calculated using the parentage analysis without parents (PWoP) procedure ([4]). For example, if the Nb is set at 600, and the Nb Tolerance is set at 0.02, allowable Nb values would be in the range from 588-612 for simulated populations using PWoP.

0.7.3.6   Ages

This value gives the number of age classes for the population to be simulated. Note that this is disabled, and that the length of the lists for Female, Male Fecundity and Female, Male Survival values (see below) are set to length Ages minus one for the former and Ages minus two for the latter. The age value and changes in these lists, therefore, need to be edited in a configuration file (Section 0.8).

0.7.3.7   Female and Male Survival

These are lists whose i^{th} value gives the probability of survival for an individual of the ith age category.

0.7.3.8   Female and Male Fecundity

These are lists whose i^{th} item gives the probability of reproducing for individuals of the ith age category.

0.7.3.9   Force Skip

This value gives a probability, for each non-zero value, f_a, in the female fecundity list, that during a given reproductive cycle r the value will be replaced with zero. Such replacement means that females belonging to the age class a, given by f_a, for cycle r, are infertile. This parameter is set (assigned a non-zero value) in only a few of the configuration files we copied from the AgeStructureNe program, and we have not made it editable in our interface. In your own custom configuration files you can set it to any value 0 through 100 (the value shown in the interface will be the file's value divided by 100).

0.7.3.10   Proportional Litter Sizes

If not a "None" value, it will be a list of integers, affecting litter sizes. Note that we do not allow interface editing of these parameters, but note that, as above for the Force Skip setting, you can inter this parameter value in a configuration file. This should be a list, and can have one of 2 valid configurations:

a The list can have a single value l, and $l < 0$, then at each reproductive cycle the maximum possible number of offspring available to each reproducing female is given by $l * -1$.

b Otherwise, the list should have (positive) integers. In this case these integers proportionally allot litter sizes, as given by their indices in the list. In particular, at each reproductive cycle, as a female is chosen to mate:

 i An age, a, is chosen randomly.

 ii A female f_a is chosen randomly from the females of age a.

 iii A list index i (i.e. one of 1,2,3, ... n , where n is the number of items in the litter list), is selected by weighted probability, proportionally according to the ratio of each list value to the sum of the list values.

 iv Female f_a is then the mother of the next i offspring (i.e. the female selection steps are skipped for the next i pairings, since f_a is the female of the pair). Thus, she will parent the next i offspring, unless the jth of her offspring assignments produces the maximum total offspring for the cycle (i.e. N0 is reached), and $j < i$.

0.7.3.11   Monogamous

When this box is checked, monogamy is enforced.

0.7.3.12   Probability of male birth

This value is used during reproductive cycles to determine the sex of new individuals. As noted above in the description of the N0 (see 0.7.3.1), it is used also in the N0 calculation, and so the latter is recalculated when this value is changed. When the Cull method (0.7.5.1) is set to equal_sex_ratio, this parameter is automatically set to 0.5, and its entry box is disabled.

0.7.3.13   Population size

This value determines the number of individuals that will be created in the simulation's initial population. In subsequent cycles, the size will change according to the reproductive parameters, notably N0 (0.7.3.1).

0.7.4   The Genome section

Parameters in the Genome section determine the simulated individuals' allelic content (Figure 0.5).

Figure 0.5: Simulation interface, genome section

0.7.4.1 Use Loci File

Check this box and click on Loci File to locate a file with 3 or 4 comma-delimited fields, without spaces, that include (in order), loci_name, chromosome_name, and position. You can also add a 4th column that gives an allele frequency f, with $0.0 \leq f \leq 1.0$ such that the biallelic SNP loci will be initialized with frequencies f and $f - 1$

The program expects a floating point or integer value for the position, and that each loci name occurs only once in the file. If multiple loci share the same position on the same chromosome, then, so that all will be included in the simulation, the position of the ith loci with the identical position p will be, $p + (i - 1)(1^{-13})$.

With the entries in this file, giving N loci on M chromosomes, the simulation's genome is initialized accordingly.

When the Use Loci file box is checked and a file is specified, the program ignores the values for the other parameters that specify the number of chromosomes (Section 0.7.4.3), number of snps (Section 0.7.4.5), number of microsats (Section 0.7.4.4), and starting msat allele totals (Section 0.7.4.6). Also, the simulation will generate a chromosome loci file (Section 0.9) that can be used in an LDNe2 run to limit the alleles used in the estimation to those from loci pairs, l_i, l_j, on chromosomes c_k, c_l, such that $c_k \neq c_l$ (Section 0.11.1.2).

0.7.4.2 Mutation frequency

If non-zero, this value is applied to microsatellites (not to SNPs). It will be used to set the simuPOP simulation StepwiseMutater's rate parameter.

0.7.4.3 Number of chromosomes

When set to zero, the simulation assigns each loci to a separate chromosome. When set to a positive and non-zero integer M, the simulation such that the N loci (SNPs plus microsats) are assigned evenly over the M chromosomes. Compliance with simuPOP's loci-adjacency implemenation means that the first N/M loci will be assigned to chromsome 0, the second N/M to chromosome 1, and so on.

When the number of chromosomes don't evenly divide the loci, the remainder, $r = N \ mod \ M$, is distrubuted such that $(N/M) + 1$ loci are assigned to chromosomes $0, 1, 2 \ldots r - 1$.

Whenever this paramter's value is non-zero the simulation will generate a chromosome loci file (Section 0.9) that can be used in an LDNe2 run to limit

the alleles used in the estmation to those of loci pairs, l_i, l_j, on chromosomes c_k, c_l, such that $c_k \neq c_l$ (Section 0.11.1.2).

While simuPOP accepts any numeric values for loci positions, when this parameter is non-zero, our program will assign, on each chromosome, loci positions spaced evenly using 100 as the total units on the chromorome, in order to emulate 100-centimorgan chromosomes.

Note that while you can run the simulation using both m microsats and s SNPs, as noted elsewhere (Section 0.7.4.4), the microsats will always be the first m, and SNPs the last s loci. If you subsequently want to evaluate LDNe for microsats only (or SNPs only), you may unintentionally impose high physical linkage in your loci sample for LDNe. For example, if you use few chromosomes and few microsats in a simulation, and then run LDNe2, limiting the loci to only the microsats (Section 0.11.3.1), while also using the chromosome-loci file output by the simulation (Section 0.11.1.2), you may be severely limiting the loci pairs available to calculate LDNe.

0.7.4.4   Number of Microsatellites

Microsatellites are simulated under diploidy. Note that in cases in which you specify both m microsatellites and s SNPs, in the output genepop file, the first m loci are the microsatellites and the last s loci are the SNPs.

0.7.4.5   Number of SNPs

SNPs are simulated as bi-allelic, under diploidy. Frequencies are initialized, targeting the value for initital heterozygosity and using a truncated normal distribution. See Section 0.7.5.4 for details.

0.7.4.6   Starting Msat allele total

This value gives the initial number of microsatellite alleles a for each microsatellite in the initial population. For each microsatellite, the initial genotype frequencies are drawn from the Dirichlet distribution, using a heuristic that draws a distribution of n allele frequencies by trial, by adjusting the Dirichlet shape parameter until the expected heterozygosity is within a tolerance, currently set to 0.001, of the value for the parameter "Microsat het init" (Section 0.7.5.5).

0.7.5   The Simulation section

These parameters determine several per-cycle behaviours (Figure **??**).

0.7.5.1   Cull method

Cull method indicates one of two possible per-cycle methods of removing individual from the population.

1. Survival rates. Individuals of age greater than zero are removed from the population by comparing a random number against the probability for that individuals age and sex, in the survival list (Section 0.7.3.7).

Figure 0.6: Simulation interface, simulation section

2. Maintain distribution. For each cohort with age $a, a > 0$, (i.e. excepting newborns), divide the individuals by sex, s, into two lists. From each list with t_s individuals, randomly cull n_s individuals from each list where

$$n_s = floor(p_s), \text{ where } p_s = (t_s(1 - survival_s[a]))$$

and $survival_s[a]$ is the survival rate (Section 0.7.3.7) for individuals of age a and sex s. One more individual from each is culled by the probability given by the fractional part of p_s.

0.7.5.2 Filter recorded pops by heterozygosity

When checked, the genepop file output will be restricted to the populations as filtered using the Heterozygosity filter parameters. See section 0.7.5.3.

0.7.5.3 Het filter parameters

If the Het filter checkbox is checked, apply a filter to each pop of the form m, x, t, where,

- Mean expected heterozygosity is calculated as

$$mean(H_{L_1}, H_{L_2}, H_{L_3} \ldots H_{L_N})$$

for N loci, and

$$H_{L_i} = 1 - \sum_{j=1}^{n} freq(a_j)^2$$

and $freq(a_j)$ is the frequency of the jth of the n alleles of loci L_i.

- m is the minimum mean expected heterozygosity,

- x is the maximum mean het, and

- t is the total number of populations to record.

The output genepop file will then record only populations whose mean Het falls inside the range. Further, it will stop the simulation as soon as one of the following is met:

- t populations are recorded.

- The current population's mean Het is less than m and at least one of the following is true:

  - The simulation includes no microsatellites.

  - The mutation frequency is zero (Section 0.7.4.2).

- The last cycle (Section 0.7.5.11) has completed.

Note that if you set the filter to accept a narrow band of values, for example, a setting of 0.299,0.301,1 the simulation may never realize a population within the filter range. In such cases you should can set the replicate parameter (Section 0.7.5.7) to, say, 5 to 10, in order to ensure that at least one of your runs produces a population (as written to the output genepop file).

0.7.5.4   SNP het initialization

The value here determines a mean expected heterozygosity (H_e) for the N SNPs in the initial population as totaled by either the Number of SNPs parameter (Section 0.7.4.5) or as input using a loci information file (Section 0.7.4.1) using the calculation as given in Section 0.7.5.3). The SNP frequencies are drawn from a truncated normal distribution (using the python scipy.stats.truncnorm function, limiting sampling values to the interval $[0.0, 0.5]$) whose mean is is given by solving for f, in $H_e/2 = -f^2 + f$, and whose standard deviation is derived heuristically, so that the mean expected heterozygosity calculated using the frequencies is within a tolerance value t of the value set for initial heterozygosity. The program initially tries with $t = 0.01$. If this fails, it increases to $t = 0.02$. If the latter fails, the program reports an error and stops. The default value for the intital H_e is 0.5. Note that we are currently developing an alternative method for SNP frequency initialization, as we've found the distribution of frequencies generated under this method are generally tightly clustered around the mean, and so SNPs tend to have similar allele frequencies.

0.7.5.5   Msat het initialization

The value here determines an expected heterozygosity (H_e) for all Microsatellites in the initial population, so that it also gives the mean expected heterozygosity for the SNP set. The value will determine a set of Dirichlet-distributed allele frequencies, with the shape parameter chosen heuristically to produce through trials a set of frequencies that generate an expected heterozygosity within 0.001 of the target value. The default value for the intital H_e is 0.8. The valid range for for this parameter is $0.0 < H_e \leq 0.85$.

0.7.5.6 Nb and census adjustment

This parameter offers one or more specifications that will change the target Nb and the number of individuals in the population by a fixed rate and at a range of cycles (one or more). Entries are of the form $min - max : rate$, specifying a change in Nb and census size applied at cycle numbers min through max. The values conform to $min <= 2 <= max$, and $rate >= 0.0$. No adjustment is made with $rate = 0.0$. For example, to reduce the Nb and the total number of individuals by a tenth at cycle 3 (remaining in effect for the remaining cycles, unless another adjustment is added to the list), you would edit the entry to read, 3-3:0.1. The adjustments are different, depending whether $rate$ is less than or greater than 1.0:

- If $rate$ is less than 1.0, the target Nb value, and each age class in the current census is reduced by the proportion given by $rate$. Note that the change in Nb will result in a change to N0 as described above the in section 0.7.3.1.

- If $rate > 1.0$, the target Nb value will be multiplied by $rate$, with a resulting recalculation of N0. No change will be made to the current census.

0.7.5.7 Replicates

This value sets the number of independent simulations run with the current parameter set. These can be run in parallel if you specify more than one process in the Processes parameter (Section 0.7.1.1).

0.7.5.8 Skip breeding probability

If this value is not set to "None," it should be a list of percentages. It effects the number of available females of a given age at a given cycle number c. The $i^{th} percent p_i$ gives the probability $(p/100)$ that a female of $age = i$, is not able to breed in cycle c. Like the Litter (Section 0.7.3.10) and Force Skip parameters (Section 0.7.3.9), this parameter is not settable in the interface, but can be included in your configuration file.

0.7.5.9 Cycles of burn-in

This integer n is valid in the range $1 \leq n \leq r$, with r giving the total reproductive cycles (Section 0.7.5.11). This value tells the simulation that the Nb tolerance test (Section 0.7.3.5) should not be performed for the first n cycles. The default value for this parameter equals twice the number of Ages in the model.

0.7.5.10 Start recording at cycle

This integer value c will result in the genepop file containing only the populations of cycles c through r, where r = total Reproductive cycles . This can greatly reduce the size of the output genepop file, when you are interested only in the last $r - c$ cycles, but want to simulate many cycles before recording, and when you have large populations and many loci to simulate.

0.7.5.11   Stop recording at cycle

This integer c tells the simulation to stop after recording (and end) after the cth cycle. (total cycles includes the burn-in cycles).

0.7.5.12   Tolerance tries

Choices for this parameter are 100, 1000, and 10,00. For each reproductive cycle, this value is the maximum number of tries allowed for a population's PWoP calculated Nb value to be within the range given by parameter Nb Tolerance (Section 0.7.3.5). Set at 100, this parameter can cause simulation interuption failures when the parameterization of the population's size, fecundity, survival values, etc., result in PWoP caluclated distributions of Nb values with a high enough variance that that 100 tries is insufficient. On the other hand, a setting of 10,000, useful when the distribution of possible PWoP values has high variance, will, when the the simulation is parameterized such that PWoP Nb calculations cannot meet the tolerance setting, cause a very lenghtly but ultimately interrupted simulation.

0.8   Manually editing configuration files

Our program provides a collection of simulation configuration files available at https://github.com/popgengui/agestrucne/tree/data. You can directly edit a *conf file, which can sometimes be more convenient than using the GUI interface to change parameter values. If you open one of the supplied configuration files in the subdirectory, "configuration_files," you'll see the parameter=value pairs, each below a section header inside square brackets. Some of the parameter names differ from their functional equivalents in the interface (see Table 0.1).

0.9   Simulation output

When a simulation is complete the message "simulation in progress" will disappear from the interface and editable entry boxes will no longer be grayed-out. A completed simulation delivers a genepop file for each replicate, named using the Output files base name parameter shown in the The Load/Run section of the simulation input (Section 0.7). The base name is extended with a replicate number and a "genepop" extension, so that, for example, if your simulation output base name is bulltrout, and you specified 3 replicates, the output file for the 3rd replicate would be named "bulltrout.r3.genepop". Also, there are several produced during the first replicate only, all prefixed with the output base name. The output file with extension "conf" lists the parameter settings for the simulation (and, hence, for all replicates). Other ouput files include extension "_age_counts_by_gen.tsv,", and extension "_nb_values_calc_by_gen.tsv". When the simulation includes a heterozygosity filter (Section 0.7.5.2) the output will include a file with extension "_het_value_by_cycle_number.tsv". When the simulation is run with a non-zero "Number of chromosomes" (Section 0.7.4.3), the output includes a file with extension, "_loci_and_chromosome.tsv". Details on the output files follow.

Short name in file	Interface Name	Link to description
N0	N0 (Newborns)	(Section 0.7.3.1)
Nb	Nb	Set to "None"value(Section 0.7.3.4)
NbNc	Nb/Nc	(Section 0.7.3.2)
NbNe	Nb/Ne	(Section 0.7.3.3)
NbVar	Nb tolerance	(Section 0.7.3.5)
ages	Ages	(Section 0.7.3.6)
cull_method	Cull method	(Section 0.7.5.1)
dataDir	Ignored	Ignored
doNegBinom	Not documented	Use "False." Feature not documented.
do_het_filter	Filter recorded pops by heterozygosity	(Section 0.7.5.2)
fecundityFemale	Female relative fecundity	(Section 0.7.3.8)
fecundityMale	Male relative fecundity	(Section 0.7.3.8)
forceSkip	Force skip	(Section 0.7.3.9)
gammaAFemale	Not documented	Ignored when neg. binom. is False
gammaAMale	Not documented	Ignored when neg. binom. is False
gammaBFemale	Not documented	Ignored when neg. binom. is False
gammaBMale	Not documented	Ignored when neg. binom. is False
gens	Stop recording at cycle	(Section 0.7.5.11)
het_filter	Het filter parameters	(Section 0.7.5.3)
het_init_snp	SNP het init	(Section 0.7.5.4)
het_init_msat	Microsat het init	(Section 0.7.5.5)
isMonog	Monogamous	(Section 0.7.3.11)
lbd	Ignored	Ignored
litter	Proportional Litter Sizes	(Section 0.7.3.10)
maleProb	Probability male birth	(Section 0.7.3.12)
model_name	Model	(Section 0.7.2.2)
mutFreq	Mutation frequency	(Section 0.7.4.2)
nbadjustment	Nb and census adjustment	(Section 0.7.5.6)
numChroms	Number of chromosomes	(Section 0.7.4.3)
numMSats	Number of microsatellites	(Section 0.7.4.4)
numSNPs	Number of SNPs	(Section 0.7.4.5)
popSize	Population size	(Section 0.7.3.13)
reps	Replicates	(Section 0.7.5.7)
recomination_intensity	Recombination intensity	(Section ??)
skip	Skip breeding probability	(Section 0.7.5.8)
startAlleles	Starting Msat allele total	(Section 0.7.4.6)
startLambda	Cycles of burn-in	(Section 0.7.5.9)
startSave	Start recording at cycle	(Section 0.7.5.10)
survivalFemale	Female survival	(Section 0.7.3.7)
survivalMale	Male survival	(Section 0.7.3.7)
use_loci_file	Use Loci File	(Section 0.7.4.1)

Table 0.1: Short parameter names as found in configuration files, and their descriptions

1. The conf file shows the parameter settings used in the simulation (except the number of replicates, which it always sets to one). This file can be loaded into another instance of the Simulation Input (see the Load/Run parameter) and another simulation with matching parameters can be run. Conveniently, if it represents many customized settings on a former configuration file, small changes to it can be made to run a slightly adjusted simulation, without having to re-enter all of the settings used to create it.

2. The age counts file is a table with tab-delimited fields that gives a count of total individuals for each age class, for each (one-indexed) reproductive

cycle. The first line in the file gives column headers, the first "gener-ation," referring to reproductive cycle number, a count of reproductive cycles, and the rest listing age classes simply as $1, 2, 3 \cdots t$, where t = total age classes. This file is created only for the first simulation replicate.

3. The Nb values file is a table with tab-delimited fields giving the PWoP Nb values calculated during the simulation, and used to compare to the target Nb value $+/-$ the Nb Tolerance value. The first column gives the one-indexed reproductive cycle number and the second the PWoP-based Nb value that passed the tolerance test, and represents the accepted population for that cycle. This file is created only for the first simulation replicate.

4. The heterozygosity-by-cycle output file is created only when the simula-tion includes a heterozygosity filter (Section 0.7.5.2). The two comma-delimited fields give the reproductive cycle number (also referred to as the "pop" number, and the mean expected heterozygosity (Section 0.7.5.3) of the population at that cycle number. The entries are restricted to those pops that are also recorded in the genepop file (i.e. that passed the filter parameters (Section 0.7.5.3).

5. The loci-and-chromosome output file is created only when the value for the "Number of chromosomes" parameter (Section 0.7.4.3) is non-zero. The two, tab-delimited values in each line give, first, a chromosome name, in the form of an integer such that N chromosomes are named $0, 1, 2 \cdots N - 1$, and, second, a loci name that matches one of those in the first section of the output genepop file.

6. The genepop file conforms to the genepop file standards given at `http://genepop.curtin.edu.au/help_input.html`. The header line notes the name of the *.gen file it came from, which simply names an interme-diate file from which it derived it's population information. It also gives the value of Nb/Ne. If the value is non-zero, it can be loaded automat-ically into the Nb/Ne estimation interface (see the Parameters section ?? of the Nb/Ne Estimation interface description. The second line of the genepop file gives the name of the first loci, which is simply its ordi-nal, "l_0." Each consecutive loci, $l_0, l_1, l_2 \ldots l_{L-1}$ (where L gives the total number of microsatellites plus the total number of SNPs) is listed on a separate line. Note that the first M loci will represent the microsatel-lites, and the last S loci will represent the SNPs, with M and S the totals given in the genome parameters (Section 0.7.4) of the Simulation Input. Thereafter the file consists of separate "pop" sections, each representing a reproductive cycle. The first n cycles (as numbered $1, 2, 3 \ldots n$) will not be in the file if the Start at cycle number parameter is set to $n + 1$. The population for each cycle is listed, in order of cycle number. Each is demarked by a line with "pop" as its sole entry. Individuals, one to a line, follow each "pop" entry. Each individual has an ID with multiple fields delimited by a semicolon, giving, individual id number;sex (1 = male, 2 = female);id of father;id of mother;age class. These are followed by a comma, and then a space-delimited set of alleles for each locus named in the lines 2– total number of loci. Note that these allele entries represent
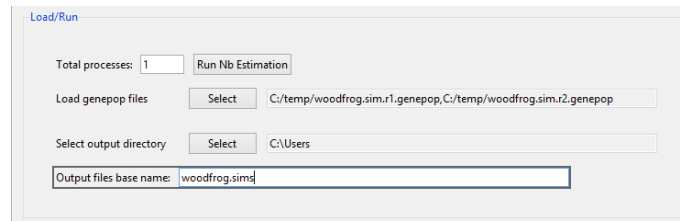
Figure 0.7: Nb/Ne estimation interface, Load/Run section

diploidy, and use 3-digit allele numbering so that, for each loci, allele one is named by the first 3 digits, and allele 2 by the last 3.

0.10   Running an Nb or Ne Estimation

The Nb (and Ne) estimation interface performs an LD based Nb or Ne estimation from genepop file input as supplied by the user. While it was developed in concert with the simuPOP-based simulation output from our program's interface, it will perform estimations on any genepop file input. To run estimations:

a Load a new Nb interface with the add menu's "Add new Nb estimation" option (Figure 0.1) and set the parameters with the provided controls. For details see Section 0.11

b Load a genepop file..

c Adjust the estimation parameters. The parameters are detailed in Section 0.11

d Click the button labeled "Run Nb Estimation", and the computations will start. The button's text now changes to say "cancel simulation," and next to it a new label will note that "estimations in progress." As in the other interfaces, while the estimations are in progress, the parameter controls are disabled.

0.10.1   Running an Nb or Ne estimation from the command line

The module pgdriveneestimator.py offers a command line interface for running LDNe2 estimations. At a Linux or DOS terminal you can type a command of the form,

$$<\text{python}> \ <\text{pgdriveneestimator.py}> \ <\text{options}>$$

where

- ⟨python⟩ is the python3 or python2.7 executable,

- ⟨pgdriveneestimator.py⟩ is either the module name pgdriveneestimator.py, if your terminal already knows the path to the module, which should be the case with most installations (see the README.md listed above in

Section 0.2.1 and available in our github repository's main directory), or the full path to the module (in the distributions main directory), and

- ⟨options⟩ is a list of parameters given using the option flags. To see the list of options, invoke the command without any options.

To see a detailed list of options, invoke the command with,

$$<\text{python}> \ <\text{pgdriveeestimator.py}> \ -h$$

0.11   Nb/Ne Estimations input

The interface provides for multiple subsampling schemes of both individuals and loci within the input genepop file pop sections. The sub-sections of the interface follow.

0.11.1   The Load/Run section

This section (Figure 0.7) offers an interface to load input and name the output files.

0.11.1.1   Total processes

The program will run estimations on the individual "pop" sections in parallel using the number of processes set here. It is usually advisable, unless your computer has many process already running, to use most if not all of your available (virtual) processing cores, to speed up the estimation run parameter. The program defaults to using half of the available cpu's (or virtual cores) available on your machine.

0.11.1.2   Use Chrom Loci File

Check this box, and click on "Chrom Loci File", to locate a file that has two tab-delimited columns, chromosome name and loci name. Each loci name in each genepop file loaded should be present in this file, so that LDNe2 will have a chromosome to associate with all loci.

The LDNe2 program will use this file to compute LDNe using only the alleles of loci pairs l_i, l_j on chromosomes c_k, c_l, such that $c_k \neq c_l$.

Note that the command line module pgdrivesimulation.py offers the user use a second parameter related to this file name, and that will instruct LDNe2 to compute using only the loci pairs such that $c_k = c_l$. This option is not currently available from the graphical interface.

In the absense of a file name (or when the Use Chrom Loci File box is unchecked), LDNe2 will use all loci pairs to compute LDNe.

0.11.1.3   Minimum allele frequency

This value sets the threshold below which the LDNe program will ignore an allele in its LDNe calculation. The interface defaults to 0.05, as a value that is commonly used, and which reduces upward bias [3]

0.11.1.4 Monagamy

When this box is checked, the LDNe2 parameter "mating" is set to monogamy. If the box is unchecked the parameter is set to "random mating."

0.11.1.5 Nb bias adjustment check box

Checking this allows you to apply a bias adjustment to the estimations [3]. See Section 0.11.1.6.

0.11.1.6 Nb/Ne ratio

This parameter is the value used for the bias adjustment, when it is checked. A zero value or an un-checked box means no bias adjustment will be done. Note that when you load a genepop file generated by the simulation interface, and you check the box labeled "applyNb bias adjustment", the program will load the Nb/Ne value as set in the simulation interface. You can accept it or enter another value. If no value is available in the genepop file, then you will need to enter a non-zero value to make any bias adjustment.

0.11.1.7 Pop sampling replicates

When set to an integer n, for each estimate, for each pop sampling parameter, the estimate is repeated n times. While you can set this to any value for any subsampling scheme, note that it is sensible only when your subsampling parameter involves a random sample of individuals (Section 0.11.1.9) more than 1 loci subsampling parameter subsampling schemes (Section 0.11.1.10), or both. If there is no random subsampling, the replicates will be performed, with identical results.

0.11.1.8 Loci sampling replicates.

When set to n, for each pop sampling replicate, for each loci subsampling parameter value, the estimate is repeated n times. As with the Pop sampling replicates parameter, if the loci subsampling scheme has no randomized subsample (Section 0.11.1.10), then the estimates will be identical.

0.11.1.9 Pop sampling scheme

This drop-down box offers the following subsampling schemes (Figure **??**).

1. None. This scheme uses all individuals with the pop section for the Nb or Ne estimation, unless the value for Indiv max per pop reduces the pop size to m randomly selected individuals. If the pop section has fewer individuals than the value given by Indiv min per pop , the pop section is skipped, with a message written to the Messages file (Figure 0.8a).

2. Percent. When you select this scheme the pop sampling interface (Section 0.11.2) shows a "percent" box with two buttons below it (Figure 0.8b), "Add Value," and "Trim." You can edit any box currently in the list. Clicking "Add Value" will append a box to the list, its default value taken from its nearest neighbor. For each percentage p in the list, each

pop section will be reduced to p percent of its total individuals, unless its
census is not in the pop number range (Sections 0.11.2.1 and 0.11.2.2),
in which case the population will be skipped, and a message written to
the messages file (Section 0.12.1). For each subsample the individuals
are randomly selected. Note that any multiple loci subsampling values
and/or Pop sampling replicates and Loci sampling replicates replicates
will result in an estimate for each percentage value, repeated for each of
the loci subsampling and replicate values.

3. Remove-N. This scheme also offers an editable list, which behaves as
   described for the Percent scheme (Figure 0.8c). For this scheme, when
   you enter an integer N, the pop subsample will be that given by its total
   minus N, the removed N individuals randomly selected except in the case
   of $N = 1$, in which case each of the $remove - 1$ cases will be estimated
   (i.e. there will be t estimations for a pop section with t individuals. Pop
   sections are skipped if the total individuals in the population are not in
   the pop number range (see Sections 0.11.2.1, and 0.11.2.2). Note, as with
   the Percent scheme, estimates for each N value will be repeated for loci
   sample values, pop sampling replicates, and Loci sampling replicates.

4. Cohorts Percent. Because this scheme selects individuals by age, it re-
   quires input genepop files produced by the Simulation output (Section
   0.9). If your input is empirical data and you wish to calculate Nb, it is
   assumed that each pop in the genepop file is a single cohort and all indi-
   viduals should be used in the Nb calculation. When Cohorts is selected
   the interface shows, besides its usual pop number and min/max pop size
   parameters, an entry labelled "Indiv max age" (Figure 0.8d). For each
   population, individuals outside the ages given by [0.0, max age] are ex-
   cluded. However, because the simulations produce genepop files with no
   individuals aged less than 1.0, the interval is effectively [1.0, max age].
   For each pop, and for each percentage value in the list, subsampling steps
   are:

   a. For each age value in $[0.0, maxage]$, count the total individuals t_a in
      the age group a.

   b. Find the smallest of the age group totals, $t_{smallest}$.

   c. Randomly subsample $t_{smallest}$ individuals from each age group to get
      a total sample size s.

   d. s is in the range [Indiv min per pop, Indiv max per pop] then, for each
      percentage p in the percentage list, randomly select p percent of the
      collected individuals. If s is outside the range an error occurs and the
      analysis is terminated.

5. Cohorts Count. This scheme behaves very similarly to the Cohorts Per-
   cent (above, 4), with nearly identical parameters (Figure 0.8e). However,
   instead of sampling percentages, for each sample value c_i in the list of
   sampling values, c_i individuals will be randomly selected from each age
   class. If for any pop section, any of the age classes has a count of indi-
   viduals less than c_i, an error is raised and the estimations terminate.

6. Individual Criteria. This scheme allows you to select a range of cohorts by contiguous age group. Like the cohorts schemes, it requires your input to be genepop files produced by the Simulation output. When Individual Criteria is selected the interface shows, besides it's the pop number and min/max pop size parameters, two entries for a minimum and maximum age (Figure 0.8f). The program pools all individuals within the minimum and maximum age range (inclusive), and calculates the estimation using the pool. If the total of pooled individuals is outside the range given by [Indiv min per pop Indiv max per pop], an error is thrown and the analysis is terminated.

0.11.1.10 Loci sampling scheme

1. None. All loci will be used in the estimations, from the ith to the jth, i and j given by the Loci number start and Loci number end (Figure 0.9a). If the Max Loci count m is less than the total loci, then m loci will be randomly selected.

2. Percent. For each percentage p listed in the "percentages" boxes (Figure 0.9b), an estimation is calculated using a random selection of p percent of the loci from those within the range given by Loci number start to Loci number end.

3. Total. For each total t listed in the "totals" boxes (Figure 0.9c), an estimation is calculated using a random of t loci from those within the range given by Loci number start to Loci number end.

0.11.2 Pop sampling parameters section

In this section (Figure 0.8), you set the pop section sampling parameters, which are presented according to the scheme selected in the Pop sampling scheme parameter.

0.11.2.1 Pop number start

When this is set to integer n, the estimates will skip pop section numbers (as ordered in the genepop file) in the range [1, n − 1].

0.11.2.2 Pop number end

If the genepop file has t total pop sections, then, when this parameter is set to integer n, the estimates will skip pop section numbers in the range $[n + 1, t]$.

0.11.2.3 Indiv min per pop

For the None, Percent, and Remove-N sampling schemes, a pop section must contain at least this many individuals, or it will be skipped. Skipped populations will be noted in the Messages file. For the Cohorts and Individual Criteria sampling schemes, the total cohort sample must meet or exceed this minimum, or an error occurs and the estimation run is terminated.
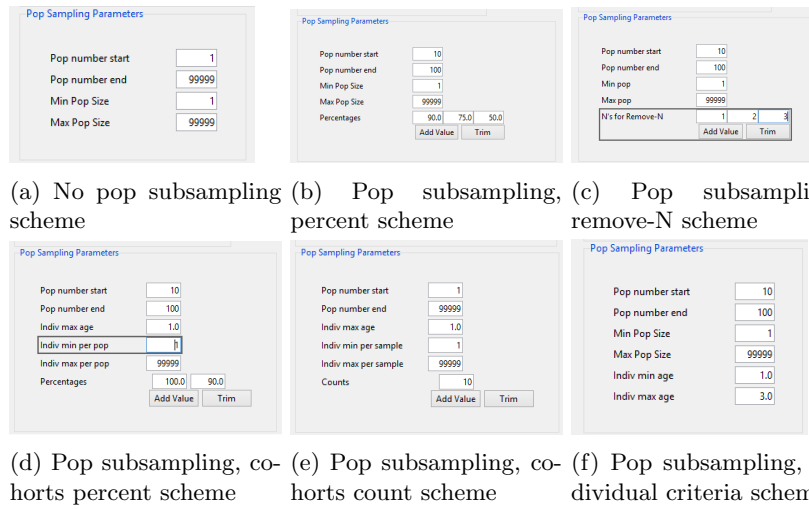
(a) No pop subsampling scheme

(b) Pop subsampling, percent scheme

(c) Pop subsampling, remove-N scheme

(d) Pop subsampling, cohorts percent scheme

(e) Pop subsampling, cohorts count scheme

(f) Pop subsampling, individual criteria scheme

Figure 0.8: Nb/Ne estimations interface, pop subsampling parameters

(a) No loci subsampling scheme

(b) Loci subsampling, percent scheme
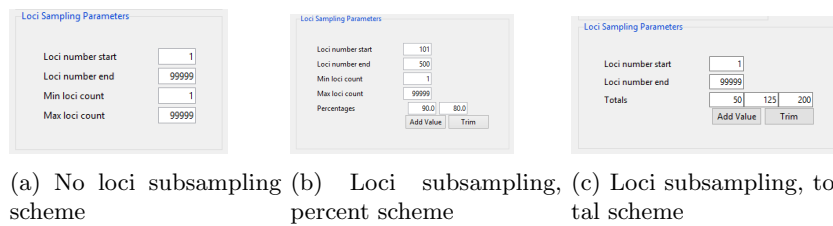
(c) Loci subsampling, total scheme

Figure 0.9: Nb/Ne estimations interface, loci subsampling parameters

0.11.2.4 Indiv max per pop

For the None sampling scheme, if total individuals in a pop section exceed this value i_max, than i_max individuals will be randomly selected from the pop. For the Percent and Remove-N schemes, pop sections with individuals totaling more than this value will be skipped, with a message written to the Messages file. For the Cohorts scheme, if sample size exceeds this value, then an error occurs and the estimation run is terminated; for the Individual Criteria scheme, the sample is reduced to the value by random selection of individuals.

0.11.2.5 Scheme-specific parameters

See the descriptions in Section 0.11.1.9.

0.11.3 Loci sampling parameters section

In this part of the interface you can limit the loci used to determine the Nb or Ne estimation.

0.11.3.1 Loci number start

Loci used in the estimation will be drawn from the ith to the jth Loci as ordered in the genepop file. This value gives the ith. Note that this range allows you, for example, to run a simulation in which the first m loci are microsatellites and the last s loci are SNPs, and then use the same genepop file in two different estimations, one based on microsatellites using loci range 1 to m, and the second based on SNPs using range loci $m + 1$ to $m + s$.

0.11.3.2 Loci number end

This value gives the jth loci in the range as described for Loci number start.

0.11.3.3 Min Loci count

This value sets a minimum for the total loci to be used in the estimation. If the loci sample in the genepop file is less than this value, the program generates an error message and the run is terminated.

0.11.3.4 Max Loci count

This sets a maximum value m on the number of loci to be used in the estimation. If the range gives a larger total, then m loci will be randomly selected from those in the range, Loci number start to Loci number end.

0.11.3.5 Scheme specific parameters

See the descriptions in Section 0.11.1.10.

0.12 Nb estimation output

0.12.1 Messages file

With extension *.msgs, this file shows the parameter settings used in the estimations, and also logs error messages.

0.12.2 Estimates table

With extension *.tsv, this file gives tab-delimited quantities associated with the Nb or Ne estimations (Table 0.2)

0.13 Visualization Interfaces

The program offers 2 interfaces that plot the Ne/Nb estimations as output by the program's LDNe output (Section 0.12).

0.13.1 Boxplot Interface

This interface is available by clicking on the "Add" menu in the main menu (Figure 0.1), then clicking "Add new boxplot interface." After using the Button labeled "Select to load a *tsv file (Section 0.12), you can group and filter input

Output file column	Description
original_file	Source genepop (gp) file from which the pop sections are derived.
pop	The ith reproductive cycle as ordered in the source gp file.
census	The total individuals in the pop.
indiv_count	The total individuals used in the estimation.
sample_value	The parameter, if any, used in the pop subsampling (e.g. percent).
replicate_number	The pop replicate number.
loci_sample_value	The parameter value, if any, used in the loci subsamlint (e.g. percent).
loci_replicate_number	The loci replicate number.
min_allele_freq	An LDNe2 parameter [no citation yet for LDNe2], the min allele frequency required to include an allele in the estimation.
est_type	Refers to the estimation type, currently only implemented fo "ld", that is, the estimation is based on the LDNe method.
est_ne	The Ne or Nb estimation [no citation.
95ci_low	The lower of a 95% confidence interval for the estimate, based on the jackknife method described in [no citation yet for LDNe2].
95ci_high	The upper 95% confidence interval for the estimate, based on the jackknife method described in [no citation yet for LDNe2]
overall_rsquared	A weighted mean of the allelic pairwise estimators $\hat{r}_\Delta$ (see [no citation yet for LDNe2]).
expected_rsquared	The expected value of the estimator $\hat{r}_\Delta$ (see [no citation yet for LDNe2]).
indep_comparisons	The number of independent comparisons used to determine the estimate (see [no citation yet for LDNe2]).
harmon_mean_samp_size	Sample size to calculate the LDNe estimate ([no citation yet for LDNe2]).
alt_ci_low	Lower, parametric 95% confidence interval ([no citation yet for LDNe2]).
alt_ci_high	Upper, parametric 95% confidence interval ([no citation yet for LDNe2]).
nbne	If not "None," then this is the Nb/Ne ratio used in a bias adjustment for the estimate (see [2]).
ne_est_adj	The estimate as bias-adjusted (see [2]). If no bias adjustment was calculated, then this value will be identical to that in the est_ne column.
95ci_low_adj	The 95ci_low value as bias-adjusted (see [2]). If no bias adjustment was calculated, then this value will be identical to that in the 95ci_low column.
95ci_high_adj	The 95ci_high value as bias-adjused (see [2]). If no bias adjustment was calculated, then this value will be identical to that in the 95ci_high column.
mean_het	The mean expected heterozygosity of the population, calculated as defined in Section 0.7.5.3. The calculation employs all individuals in the pop (no subsampling is applied), and only the loci with the range given by the start loci number parameter (Section 0.11.3.1) and the end loci number parameter (Section 0.11.3.2).

Table 0.2: Nb or Ne estimation output values

values to get a boxplot of the LDNe Nb/Ne estimations according to the groups

and filters (Figure 0.10a).

0.13.1.1 Grouping.

The interface offers four Group-by fields allowing you to group the data using up to four value sets according to the input fields as given in the Ne/Nb estimation output (See Table 0.2):

- genepop file from original_file column in the Ne/Nb estimation output tsv file.

- pop number, from the pop column.

- total indiv. sampled, from the indiv_count column.

- pop subsample value, from the sample_value column.

- pop sampling replicate, from the replicate_number column.

- loci subsample value, from the loci_sample_value column.

- loci sampling replicate, from the loci_repicate_number column.

0.13.1.2 Filtering.

Filtering allows you to specify one of the set of values for several input fields, so that output entries without that value will be excluded from the plot data. For example, if you have output from a run in which you set a percentage subsampling scheme for pops (Section 0.11.1.9), and you sampled by 50, 80, and 100% , you can select one of the percentages, and all entries used to make the boxplot will be those with the selected pop subsampling value.

0.13.1.3 Y-axis field.

You can select the values for the y-axis field using the "Y-axis values" drop-down box. The following are the availbale output fields in the LDNe Nb/Ne estimation output (See Table 0.2):

- LDNe estimation, from the est_ne column.

- low 95% CI, from the 95ci_low column.

- high 95% CI, from the 95ci_high column.

- adjusted LDNe estimation, from the ne_est_adj column.

0.13.1.4 Y-axis value limits.

You can restrict the range of Y values plotted using the sliders or their associated entry boxes. Note that in all cases values of Infinity, and "NA," which can be present in the ouput fields, have been excluded.

(a) The boxplot interface, grouping on two input fields, pop (reproductive cycle number) and pop sampling replicate number.
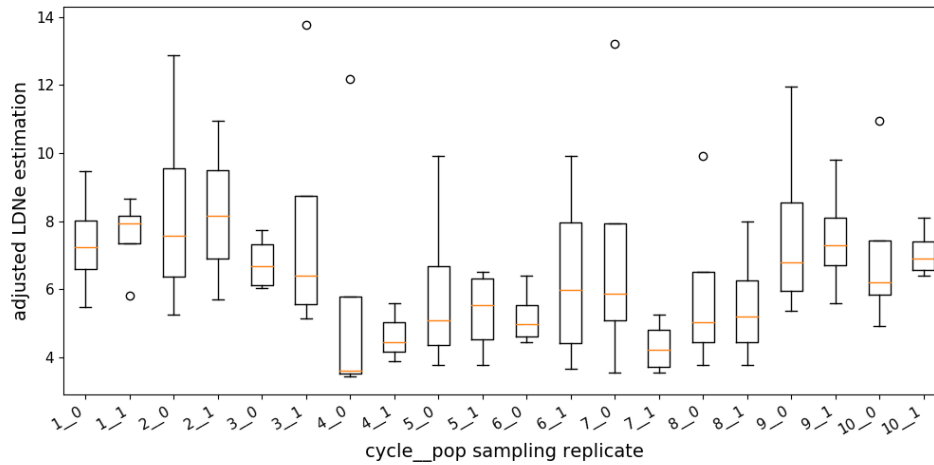
(b) The resulting boxplot.

Figure 0.10: The LDNe Ne/Nb boxplot interface.

0.13.1.5   Axis and tick label font size adjustment

You can adjust the font size for the axis and the tick labels using either the slidling scale controls or the text boxes. These can be especially helpful when the size of tic labels causes axis labels to be pushed out of the frame.

0.13.1.6   The plot.

The plots produced (Figure 0.10b) result from a call to the maplotlib "boxplot" command , with its "x" parameter set to the grouped/filtered data, and the "labels argument" set to show the grouped field values (separated by double underscores). Otherwise, arguments to boxplot are not specified, and so use default arguments. You can use the "Save" button in the "Save Plot" subframe to save the plot to file. A *png image format will result if you use any file name, except when you add ".pdf," which will then be the format of the saved file image. png and pdf are the only formats used.

0.13.2   Regression Interface

This interface is available by clicking on the "Add" menu in main menu (Figure 0.1), then clicking "Add new regress plot interface." After loading a LDNe Nb/Ne estimation output file (Section 0.12.2), you can plot regression lines (Figures 0.11 and 0.12).

0.13.2.1   Filtering.

These selections offer the same filtering as do those for the boxplot (Section 0.13.1.2). In this case, restricting an input field to a single value reduces the number of lines regressed, such as regressing over pop values, but only for one of multiple genepop files (Figure 0.11a), or regressing over a series of genepop files, but only for the estimations associated with pop sampling replicate 2 (Figure 0.12a).

0.13.2.2   Y-axis field

The "Y axis value" drop-down box allows you to select which set of LDNe output values are used as the response variable in the regression. See the description for the boxplots in section 0.13.1.3.

0.13.2.3   X axis variable

You can regress using either the pop number (i.e. the values in the pop column in the output tsv file, see Table 0.2), or the genepop file (i.e. the file names in the "original_file" column in the output tsv file, stripped of their directory paths. When you use file names as x-axis values the program will assign numeric values to the file names in one of two ways.

- Genepop file names are numbers. If your genepop files are named using all-numbers, they will become the numerical values used along the x-axis.

- Genepop file names are not all-numbers. In this case the files will be assigned integers $1, 2, 3 \ldots N$ according to a natural sorting algorithm,

(a) Settings, filtering set to select one of 3 pop subsampling values, 0.7.
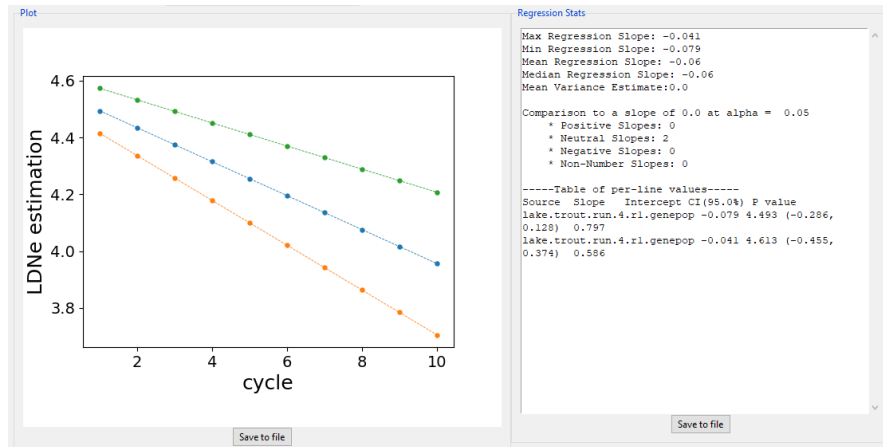
(b) The resulting plot.

Figure 0.11: The Nb/Ne estimation regression interface with cycle number as the x-axis variable.

that will sort non-number parts of the name alphabetically, but will sort number-parts numerically. This means, for example, that files with form myfile_1.txt, myfile_11.txt, and myfile_2.txt will be sorted in order, and assigned x-values 1,2,3:

1. myfile_1.txt

2. myfile_2.txt

3. myfile_11.txt

This genepop-file based numbering was created to accommodate regression over a series of genepop files with a single pop entry, with each file representing, for example, an annual sample of a real population, and each file named text giving the population name and a number representing the year, or an otherwise ordinal number.

(a) Settings, showing selection of a single pop subsample value, and leaving the "pop sampling replicate setting" to all, in this case plotting data for 4 replicates.
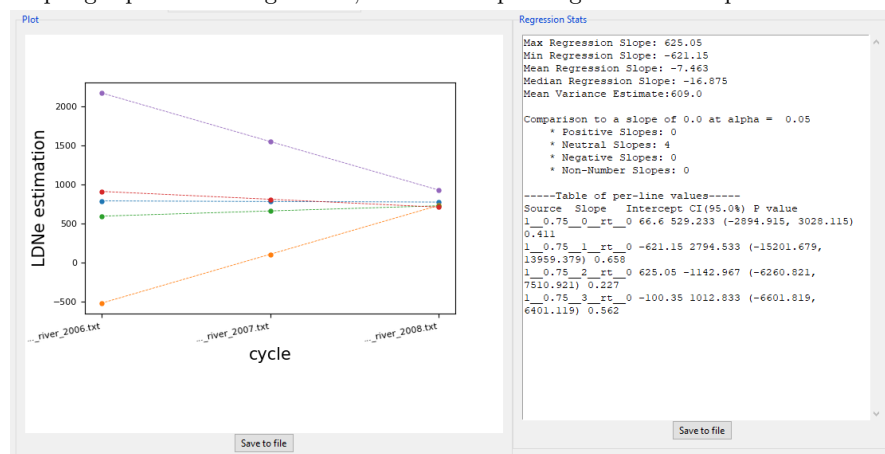
(b) The resulting plot. Note that there is a line for each of the 4 replicates, for the single pop subsampling value of 0.75.

Figure 0.12: The Nb/Ne estimation regression interface with file sort ordination as the x-axis variable.

0.13.2.4 Min cycle and Max cycle text boxes

You can limit the cycle range over which the regression is calculated by entering a minimum and maximum cycle number in these boxes.

0.13.2.5 Axis and tick label font size adjustment

You can adjust the font size for the axis and the tick labels using either the slidling scale controls or the text boxes. This will be helpfull as changing tic axis values can push labels out of frame.

0.13.2.6 The plot

The plot shows a regression line for each combination of input fields, as well as a line based on an expected slope.

You can use the "Save" button to save the current plot to file, either as a png (the default image format when you use a name with no extension, or an extension other than ".pdf" which is the sole alternative format.

0.13.2.7   The regression stats text box

This text box shows regression statistics and quantities. You can save this text
to file using the "Save" button.

Bibliography

[1] Bo Peng and Marek Kimmel. simuPOP: a forward-time population genetics simulation environment. *Bioinformatics*, 21(18):3686–3687, September 2005.

[2] Robin S. Waples, Tiago Antao, and Gordon Luikart. Effects of Overlapping Generations on Linkage Disequilibrium Estimates of Effective Population Size. *Genetics*, 197(2):769–780, June 2014.

[3] Robin S. Waples and Chi Do. ldne: a program for estimating effective population size from data on linkage disequilibrium. *Molecular Ecology Resources*, 8(4):753–756, July 2008.

[4] Robin S. Waples and Ryan K. Waples. Inbreeding effective population size and parentage analysis without parents. *Molecular Ecology Resources*, 11:162–171, March 2011.