

Introduction.....	2
Installation.....	2
Starting the program.....	3
Loading interfaces.....	3
Running a simulation .....	3
Simulation Input.....	4
a. The Load/Run section .....	4
i. Processes .....	4
ii. Configuration File.....	4
iii. Output directory.....	4
iv. Output files base name.....	4
b. Configuration Info section .....	4
i. Configuration File Name.....	5
ii. Model Name .....	5
c. The Population section.....	5
i. N0 (Newborns) .....	5
ii. Nb/Nc.....	6
iii. Nb/Ne.....	6
iv. Nb .....	6
vi. Ages .....	6
vii. Female, Male Survival.....	6
viii. Female, Male Fecundity .....	6
ix. Force Skip.....	6
x. Litter .....	6
xi. Reproductive cycles.....	7
xii. Monogamous .....	7
xiii. Probability of male birth.....	7
xiv. Population size.....	7
d. The Genome section.....	7
i. Mutation frequency .....	7
ii. Number of microsatellites.....	8
iii. Number of SNPs .....	8
iv. Starting Msat allele total.....	8
e. The Simulation section.....	8
i. Cull method .....	8
ii. Replicates.....	8

iii. Nb and census adjustment .....	8
iv. Skip breeding probability .....	9
v. Cycles of burn-in .....	9
vi. Start recording at cycle number .....	9
Simulation output .....	9
1. The conf file .....	9
2. The age counts file .....	9
3. The Nb values file .....	9
4. The genepop file .....	10
Running an Nb or Ne Estimation .....	10
Nb/Ne Estimations input .....	11
a. The Load/Run section .....	11
i. Total processes .....	11
ii. The Load genepop files button .....	11
iii. Select output directory .....	11
iv. Output files base name .....	11
i. Minimum allele frequency .....	12

## Introduction

The AgeStructureNb GUI interface offers a user interface to allow easy access to simuPOP-based simulations<sup>1</sup> and the LDNe<sup>2</sup>-based Nb and Ne estimations as implemented in Tiago Antao's python program, AgeStructureNe, 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. The original analyses based on Tiago's code, with their applications to many species, are available in several publications, including [refs to AgeStructureNe - based pubs].

Our program offers a separate interface to perform 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.

## Installation

<sup>1</sup> Bo Peng and Marek Kimmel, "SimuPOP: A Forward-Time Population Genetics Simulation Environment," *Bioinformatics* 21, no. 18 (September 15, 2005): 3686–87, doi:10.1093/bioinformatics/bti584.

<sup>2</sup> Robin S. Waples and Chi Do, "Ldne: A Program for Estimating Effective Population Size from Data on Linkage Disequilibrium," *Molecular Ecology Resources* 8, no. 4 (July 1, 2008): 753–56, doi:10.1111/j.1755-0998.2007.02061.x.

Note that our program uses the beta version 2, the source code generously supplied to us by the authors.

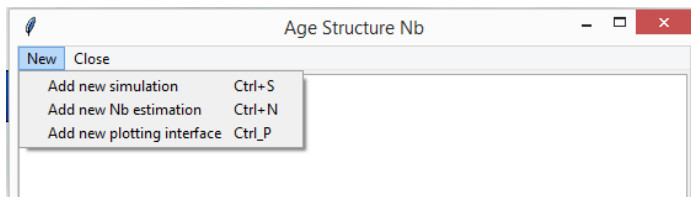
The program can be downloaded from <https://www.github.com/popgengui/negui>. Click on the button labeled “clone or download”. You can keep the program in any directory to which it can be written. For dependency details and other installation advice, see the README.md file provided with the program files.

## Starting the program

The program is launched using a python 2.7 or python 3.6 executable, invoking the negui.py module. Please see the README.md file for details on different ways to start the program.

## 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 (fig. 1). You can load any number of interfaces and run them simultaneously, though running too many at once can tax your computers cpu and/or memory capacity to a standstill.



*Figure 1 Adding an interface*

## Running a simulation

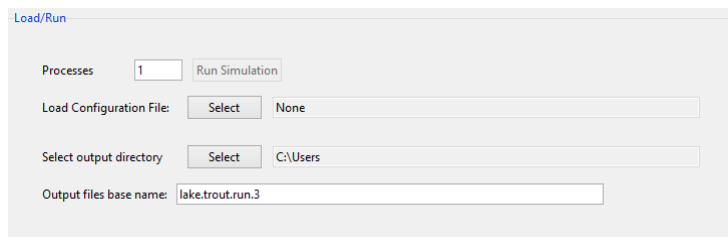
- a. Load a simulation interface with the add menu (Figure 1), and set the parameters with the provided controls. Steps for preparing the interface to run a simulation follow.
- b. Load a configuration file. The initial simulation interface requires the user to load a configuration file (Figure 2). The user can load a configuration file provided with the program, found inside the configuration/simulation directory inside the main program directory. Note that you can open one of these files and change the parameters manually, if you prefer it to setting them in the interface
- c. Adjust the simulation parameters. With a configuration file loaded (Figure 3) you can change the values in the editable controls. These are detailed below in the “Simulation Input” section.

- d. Click the button labeled “run simulation,” and the simulation will start. The button’s text now 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.

## Simulation Input

The simulation interface has 5 subframes that divide the parameters by category.

- a. **The Load/Run section** (Figure 2) of the Simulation interface offers parameters related to input and output files.

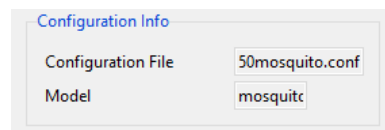


The screenshot shows the 'Load/Run' subframe of the simulation interface. It contains the following controls:

- Processes:** A text input field with the value '1' and a 'Run Simulation' button.
- Load Configuration File:** A 'Select' button and a text input field containing 'None'.
- Select output directory:** A 'Select' button and a text input field containing 'C:\Users'.
- Output files base name:** A text input field containing 'lake.trout.run.3'.

Figure 2 Simulation interface, Load/Run section.

- i. **Processes**, with valid values between 1 and the total number of virtual 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.
  - ii. **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/simulation” path inside the main program folder.
  - iii. **Output directory**. Press the select button next to the label, *Select output directory*, select a folder for the output files written by the simulation
  - iv. **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 (see the “Simulation Output” section below).
- b. **Configuration Info section** (Figure 3). This group simply shows you the input file information and has no settable parameters.



The screenshot shows the 'Configuration Info' subframe of the simulation interface. It contains the following controls:

- Configuration File:** A text input field containing '50mosquito.conf'.
- Model:** A text input field containing 'mosquito'.

Figure 3 Simulation interface, Configuration info

- i. **Configuration File Name.** This gives the file name of the loaded configuration file.
- ii. **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.

**Population**

N0 (Newborns)	32
Nb	206
Nb/Nc	0.986
Nb/Ne	0.267
Nb Tolerance	0.05
Ages	32
Female relative fecundity	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<	
Male relative fecundity	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<	
Force skip	0
Reproductive cycles	100
<input type="checkbox"/> Monogamous	
Litter	None
Probability male birth	0.5
Population size	1505
Female survival	0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85
<	
Male survival	0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85
<	

Figure 4, Simulation interface, population section

- c. **The Population section** (Figure 4) offers many parameter settings that characterize the population's size and fecundity.
  - i. **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, 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.
- ii. **Nb/Nc** is the effective number of breeders in one reproductive cycle divided by the cohort **size**.
- iii. **Nb/Ne** 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 the section “**Nb** estimation input,” below).
- iv. **Nb** is the effective number of breeders in one reproductive cycle.
- v. **Nb Tolerance** is the proportion of the **Nb** parameter by which a new population is added to the simulation after at each reproductive cycle. For example, if the **Nb** is set at 600, and the **Nb Tolerance** is set at 0.02 then populations created at each reproductive cycle (after the burn-in period, explained below), must have an **Nb** value, as calculated using the parentage analysis without parents (PWoP) procedure<sup>3</sup>.
- vi. **Ages** 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 either a life table or configuration file (see the section “Manually editing life tables and configuration files.”
- vii. **Female, Male Survival** are lists whose  $i^{\text{th}}$  item gives the probability of survival for an individual of the  $i^{\text{th}}$  age **category**.
- viii. **Female, Male Fecundity** are lists whose  $i^{\text{th}}$  item gives the probability of reproducing for individuals of the  $i^{\text{th}}$  age **category**.
- ix. **Force Skip** 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).
- x. **Litter**, if not a “None” value, 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 enter this parameter value in a

**Comment [TC1]:** This definition needs checking.

**Comment [TC2]:** More detail about how the sim uses this value

**Comment [TC3]:** Also need more detail about role in sim.

<sup>3</sup> Robin S. Waples and Ryan K. Waples, “Inbreeding Effective Population Size and Parentage Analysis without Parents,” *Molecular Ecology Resources* 11 (March 1, 2011): 162–71, doi:10.1111/j.1755-0998.2010.02942.x.

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  $1, 2, 3, \dots, n$ , 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  $j$ th of her offspring assignments produces the maximum total offspring for the cycle (i.e. **NO** is reached), and  $j < i$ .
- xi. **Reproductive cycles** shows the total number of reproductive cycles that will be simulated.
- xii. **Monogamous**, when checked, tells the simulation to enforce monogamy.
- xiii. **Probability of male birth** is used during reproductive cycles to determine the sex of new individuals. As noted above in the description of the **NO**, it is used to in the **NO** calculation, and so the latter is recalculated when this value is changed. When the **Cull method** is set to *equal\_sex\_ratio*, this parameter is automatically set to 0.5, and it's entry box is disabled.
- xiv. **Population size** shows the number of individuals that will be created in the simulation's initial population. Thereafter the size will change according to the reproductive parameters, notably **NO**.
- d. **The Genome section** (Figure 5) parameters determine the simulated individuals' allelic content.

**Comment [TC4]:** Need to get Brian T's input here, for the details about male probability'

The image shows a window titled 'Genome' with four input fields and their corresponding values:

Parameter	Value
Mutation frequency	0.0
Number of microsatellites	100
Number of SNPs	0
Starting Msat allele total	10

Figure 5 Simulation interface, genome section

- i. **Mutation frequency**, if non-zero, is applied to microsatellites (not to SNPs). It will be used to set the simuPOP simulation StepwiseMutater's rate parameter.

- ii. **Number of microsatellites**, simulated as diploid. Note that in cases in which you specify both  $n$  microsatellites and  $m$  SNPs, in the output genepop file, the first  $n$  loci are the microsatellites and the last  $m$  loci are the SNPs.
- iii. **Number of SNPs**, simulated as diploid.
- iv. **Starting Msat allele total** gives the initial number of microsatellite alleles present in each microsatellite in the initial population. For each microsatellite, the initial genotype frequencies are drawn from the Dirichlet distribution. With **Number of microsatellites** set to 10, for example, each microsatellite will have 10 alleles with frequencies given by the Dirichlet distribution of order 10, with alpha's uniformly set to 1.0. The maximum allowed number of alleles is 100.

The screenshot shows a 'Simulation' window with the following parameters:

- Cull method:** A dropdown menu set to 'survival\_rates'.
- Nb and census adjustment:** A text input field containing '2-100:0.0', with 'Add Value' and 'Trim' buttons next to it.
- Replicates:** A text input field set to '1'.
- Skip breeding probability:** A dropdown menu set to 'None'.
- Cycles of burn-in:** A text input field set to '32'.
- Start recording at cycle:** A text input field set to '1'.

Figure 6, Simulation interface, simulation section

- e. **The Simulation section** (Figure 6) parameters determine several per-cycle behaviours:
- i. **Cull method** indicates one of two possible per-cycle methods whereby individuals are removed from the population.
  - ii. **Replicates** 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.
  - iii. **Nb and census adjustment** 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  $\min \leq 2 \leq \max$ , and  $\text{rate} \geq 0.0$ . No adjustment is made with  $\text{rate} = 0.0$ . For example, to reduce the Nb and the total number of individuals by a tenth at cycle 3, you would edit the entry to read, 3-3:0.1. The adjustments are different, depending whether the rate is less than or greater than 1.0.
    1. If the 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 Population section's description of  $N0$ .

**Comment [TC5]:** This is best explained after a consult with Brian T



2. If the rate is more than 1.0, the target Nb value will be multiplied by the rate, with a resulting recalculation of *N0*. No change will be made to the current census.
- iv. **Skip breeding probability**, if its value is not “None,” 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*<sup>th</sup> percent *p<sub>i</sub>* gives the probability (*p*/100) that a female of age=*i*, is not able to breed in cycle *c*. Like the **Litter** and **Force Skip** parameters, this parameter is not settable in the interface, but can be included in your configuration file.
- v. **Cycles of burn-in**, give an integer *n* in the range  $1 \leq n \leq r$ , with *r* giving the total **Reproductive cycles**. This value tells simulation that the Nb tolerance test (see the **Nb Tolerance** parameter description) should not be performed for the first *n* cycles. The default value for this parameter equals the **number of Ages in the model**.
- vi. **Start recording at cycle number** *c* will result in the genepop file containing only the populations of cycles *c* through *r*,  $r = \text{total Reproductive cycles}$ . 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/or have large populations/loci to simulate.

**Comment [TC6]:** I think Brian T or Biran H would better give the rationale for burn-in == ages.

## Simulation output

When a simulation is complete the message “simulation in progress” will disappear from the interface and the its editable entry boxes will no longer be grayed-out. A completed simulation delivers a genepop file for each replicate, named using the *output base name* parameter shown in the *Load/Run* section of the Simulation Input, the base name extended with a replicate number *n* 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 3<sup>rd</sup> replicate would be named “bulltrout.r3.genepop.” Also, there are three files produced during the first replicate only, all prefixed with the *output base name*. One with the extension “conf,” lists the parameter settings for the simulation (and, hence, for all replicates), another has extension “\_age\_counts\_by\_gen.tsv,” and a third file has extension “\_nb\_values\_calc\_by\_gen.tsv”. Details on the output files follow.

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 simulation similar, but 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 reproductive cycle. The first line in the file gives column headers, the first “generation,” referring to reproductive cycle number, a zero-based count of reproductive cycles, and the rest listing age classes simply as 1,2,3...*t*, *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 P<sub>WoP</sub><sup>3</sup> Nb values calculated during the simulation, and used to compare to the target **Nb** value +/- the **Nb Tolerance** value. The first column gives the zero based reproductive cycle number and the second the P<sub>WoP</sub>-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 genepop file** conforms to the genepop file standards given at [http://genepop.curtin.edu.au/help\\_input.html](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 intermediate file from which it derived its population information. It also gives the value of **Nb/Ne**. If the value is non-zero, it can be loaded automatically 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 "l0." Each consecutive loci, l0, l1, l2...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 microsatellites, and the last S loci will represent the SNPs, with M and S the totals given in the *Genome* section 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...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 as 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 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.

## Running an Nb or Ne Estimation

The screenshot shows a software interface titled "Load/Run". It contains several input fields and buttons:
 

- "Total processes:" with a text box containing "1" and a "Run Nb Estimation" button.
- "Load genepop files" with a "Select" button and a text box showing the path "C:/temp/woodfrog.sim.r1.genepop,C:/temp/woodfrog.sim.r2.genepop".
- "Select output directory" with a "Select" button and a text box showing the path "C:\Users".
- "Output files base name:" with a text box containing "woodfrog.sims".

Figure 7 Nb/Ne estimations interface, Load/Run section

The Nb (and Ne) estimation interface performs and LDNe<sup>2</sup> 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 based on any genepop file conforming to the format used by our simulation-based files. To run estimations:

- a. Load a "new nb interface" with the add menu (Figure 7), and set the parameters with the provided controls.
- b. Load a genepop file as described in the *Simulation output* section.

- c. Adjust the estimation parameters. The parameters are detailed in the *Nb/Ne estimations input* section below.
- 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.

## Nb/Ne Estimations input

The interface is divided into sections, and allows from multiple subsampling schemes of both individuals and loci within the input genepop file pop sections.

- a. **The Load/Run section** (Figure 7) offers an interface to load input and name the output files.
  - i. **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 processes already running, to use most if not all of your available (virtual) processing cores, to speed up the estimation runparameter.
  - ii. **The Load genepop files button** offers a file loading interface to locate and load one or more **genepop file(s)**. Note that when you load multiple genepop files, the parameter settings will be applied to all. In particular, activating an Nb bias adjustment [bookmark] will apply it to all the files, so that only data with which it is compatible should be loaded. This also applies to other parameters, such as population number range [bookmark] and loci number range [bookmark].
  - iii. **Select output directory.** By clicking on the button and choosing your preferred folder you select where the estimation output files will be written. Note that this will also be used as a temporary directory in which intermediate files will be written inside new directories with the “tmp” prefix, ending in random characters. These files will be removed on completion of the simulation. Sometimes, if the estimation run is cancelled or otherwise is interrupted, they will not be removed, but can be manually deleted from your directory.
  - iv. **Output files base name.** The text entered here will become the prefix for the output files [bookmark].
- b. The Genepop Files Loaded section (Figure 8) shows you the names of the loaded genepop files. It is not an editable section.

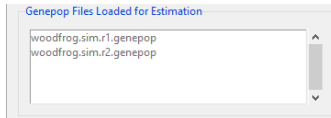


Figure 8, Nb estimation interface, genepop files loaded section

- c. Parameters section gives the main parameters, including the choice of subsampling in pop sections and/or loci.

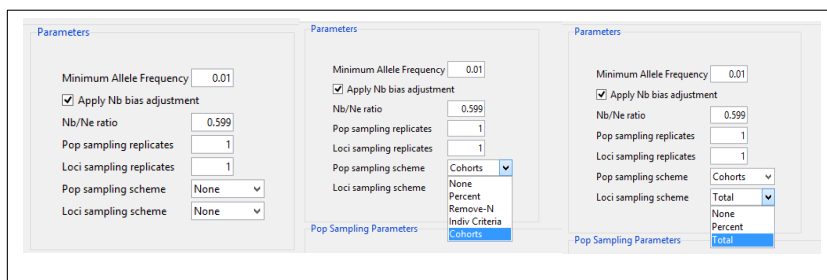


Figure 9, Nb estimations interface, parameters section. The rightmost figure shows the section with no subsampling selected. The middle figure shows the types of subsampling available for pop sections, and the left figure shows those available for the loci.

- i. **Minimum allele frequency** sets the threshold below which the LDNe program will ignore an allele in its LDNe calculation.
- ii. The **Nb bias adjustment** check box allows you to apply a bias adjustment to the estimations, as described in. Note that when you load a **genepop file** generated by the simulation interface, when check the bias adjustment box, 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.
- iii. The **Nb/Ne ratio** is that which will be used for the bias adjustment, when it is checked. A zero value or an un-checked box means no bias adjustment will be done.
- iv. The **Pop sampling replicates** parameter, when set to  $n$ ,  $n > 1$ , allows you to do an estimate on the same population, with the same pop subsampling parameter  $n$  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, or you have more than 1 loci subsampling parameter, or both. Otherwise the

**Comment [TC7]:** Should talk this over with collaborators, to better guide user setting this param.

**Comment [TC8]:** Need reference