

Introduction.....	1
Installation.....	2
Starting the program.....	2
Loading interfaces.....	2
Running a simulation	2
Simulation Input.....	3
a. The Load/Run section	3
b. Configuration Info section	3
c. The Population section.....	4
i. N0 (Newborns).....	4
ii. Nb/Nc	5
iii. Nb/Ne.....	5
iv. Nb	5
vi. Ages	5
vii. Female,Male Survival.....	5
viii. Female,Male Fecundity	5
ix. Force Skip.....	5
x. Litter	5

Introduction

The AgeStructureNb GUI interface offers a user interface to allow easy access to simuPOP-based simulations¹ and the LDNe²-based Nb and Ne estimations 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. 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.

¹ 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.

² 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.

Installation

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.

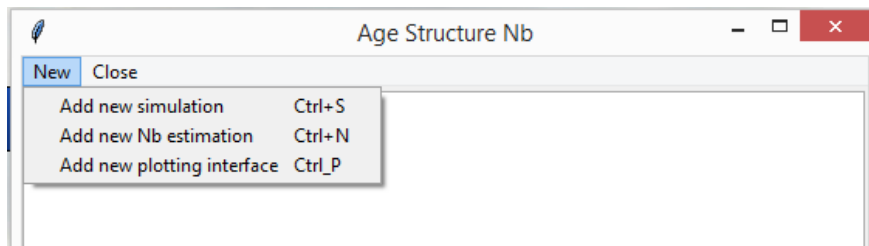


Figure 1 Adding an interface

Running a simulation

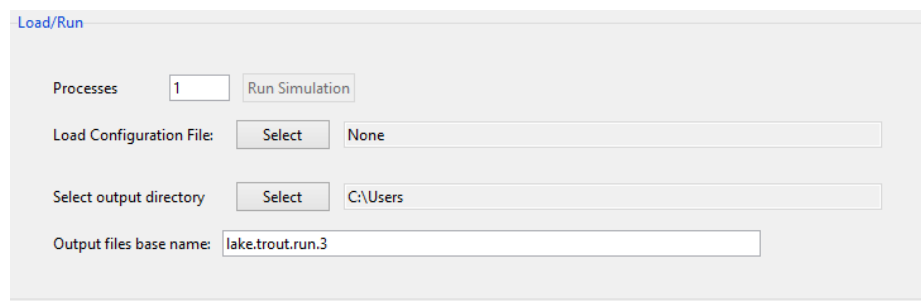
- Load a simulation interface with the add menu (fig. 1), and set the parameters with the provided controls. Steps for preparing the interface to run a simulation follow.
- Load a configuration file. The initial simulation interface requires the user to load a configuration file (fig. 2). The user can 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
- Adjust the simulation parameters. With a configuration file loaded (fig. 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 inter face has 5 subframes that divide the parameters by category.

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

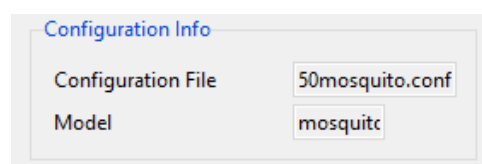


The screenshot shows the 'Load/Run' section of the simulation interface. It contains the following elements:

- Processes:** A text input field with the value '1' and a 'Run Simulation' button.
- Load Configuration File:** A 'Select' button and a text field containing 'None'.
- Select output directory:** A 'Select' button and a text 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 greather 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** (fig. 3). This group simply shows you the input file information and has no settable parameters.



The screenshot shows the 'Configuration Info' section of the simulation interface. It contains the following elements:

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

Figure 3 Simulation interface, Configuration info

- i. *Configuration File*. This gives the file name of the loaded configuration file.
- ii. *Model*. This gives the name of the model parameterized by the configuration file. In our example configuration files, the model name is usually a species'

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
<	

common name.

- c. **The Population section** (fig. 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, survival rates, 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.

Figure 4, Simulation interface, population section

- b. Update, $\text{current_female} = \text{current_female} \times \text{female_survival at age_val}$.
 - c. Update $\text{cumulative_proportion} = \text{cumulative_proportion} + \text{current_male_proportion}$.
 - d. Update $\text{cumulative_proportion} = \text{cumulative_proportion} + \text{current_female_proportion}$.
- 6. Set $N_0 = N_c / \text{cumulative_proportion}$, 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) procedure³.
- 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 *Fecundity* and *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^{th} item gives the probability of survival for an individual of the i^{th} age category.
- viii. **Female, Male Fecundity** are lists whose i^{th} item gives the probability of reproducing for individuals of the i^{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 configuration file. This should be a list, and can have one of 2 valid configurations:

³ 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.

- 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 random, but weighted proportionally according to the values at each index.
 - iv. Female f_a is then the female of the male/female pair for the next i offspring (i.e. these steps are skipped for the next i matings, 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 offspring for the cycle (i.e. NO is reached), and $j < i$.