GenRis User Manual

Large genetic diversity and low number of observed identical multi-locus genotypes (MLG) in population genetic studies of aquatic microalgae create challenges when interpreting the proportion of MLGs in natural algal blooms. This computer model was therefore created to simulate the composition of microalgal populations after a defined period of exponential growth. We simulated the probability of picking identical MLGs from this population as a function of initial genotypic diversity, intraspecific differences in growth rates and sample size. The number of distinct genotypes in each population can then be extrapolated from the model outputs and observed proportion of identical MLGs in respective population genetic studies.

Requirements

GenRis was authored and tested in Python 3.7.6. Other versions of Python may work, but have not been tested.
The Python modules numpy, matplotlib, pandas and seaborn are required to run this simulation. You can install them from a unix terminal or in python using e.g. pip or conda.
See examples at https://numpy.org/install/

Running *GenRis*

To run this simulation, you first need to download the python script GenRis.py and and move it to a folder, where you want to create the output files. Afterwards, you have to specify this folder as the working directory in your unix or python terminal/console (depending on how you want to run the script) using the command ‘cd’.

```
cd 'path/to/folder/with/GenRis.py'
```

To run the simulation in python, use the following commands:

```
from GenRis import genris
```

```
genris(numgenos = t, days = u, growthrate = v, sd = w, samplesize = x, mincell = y, maxcell = z)
```

Alternatively, if you want to run the script from a unix terminal, you can use:

```
python3 -c 'from GenRis import genris; genris(numgenos = t, days = u, growthrate = v, sd = w, samplesize = x, mincell = y, maxcell = z)'
```

The GenRiS.py file needs to be stored in the same folder, where you want to create your output files. The simulation permutates through a range of different population sizes (number of genotypes = numgenos), sample sizes (number of cells that will be picked) and standard deviations of the normally distributed growth rate function (sd). It calculates the probability of picking identical MLGs for all combinations of two out of these three variables (numgenos, samplesize, sd). These two variables of interest have to contain a range of values, while all other variables can only be defined by one number. Ranges have to be expressed as lists (e.g. numgenos = [200, 400, 600, 800, 1000, 1500, 2000, 3000]), while discrete values can be filled in directly (e.g. samplesize = 150). Larger sd will result in larger differences between growth rates of individual genotypes.

The growth rate should be measured per day. Each genotype will initially be represented by a randomly picked number of cells from the range defined by mincell and maxcell. If you assume that

each genotype is represented by only one cell at the beginning of the growth seasons (e.g. when all cells germinate from sexually produced resting stages), you can specify mincell=1 and maxcell=1. In other microalgal species, it might be more realistic to assume one to a few hundred cells per genotype in the population (e.g. mincell=1, maxcell=200). The variable "days" refers to the time period of interest, in which the algal population grows exponentially to form a bloom.
The simulation will create a table and a heatmap showing the mean probability of picking identical MLGs across the range of your selected variables (numgenos, samplesize or sd).

If you're interested in estimating the genotypic richness for a specific observed proportion of identical MLGs, you have to add the size of your sample (in which you counted the identical MLGs) to the end of the variable list (e.g. sample = 100). The variable list for this run has to contain ranges of numgenos and samplesize, which includes the size of your actual sample! This will create an output called "Probs_for_Sample.csv" containing the probability of picking identical MLGs in a sample with your specified size for the selected numbers of genotypes (numgenos) from 1000 repeated picking events. This table will be the input for the python script "CurveFitting.py".

Examples:

```
from GenRis import genris

genris(numgenos = [200, 400, 600, 800, 1000, 1500, 2000, 3000, 5000, 10000], days = 60, growthrate = 0.2, sd = 0.025, samplesize = [50, 100, 150, 200, 300, 400, 500], mincell = 1, maxcell = 1, sample = 150)
```

```
from GenRis import genris

genris(numgenos = 2000, days = 60, growthrate = 0.2, sd = [0.021, 0.023, 0.025, 0.027, 0.029], samplesize = [50, 100, 150, 200, 300, 400, 500], mincell=1, maxcell = 200)
```

CurveFitting User Manual

Fitting a curve through the mean probabilities of picking identical MLGs in a range of population sizes and a set sample size to estimate genotypic richness based on an observed proportion of identical MLGs.

Requirements

CurveFitting was authored and tested in Python 3.7.6. Other versions of Python may work, but have not been tested.
The Python modules numpy, matplotlib, scipy and pandas are required to run this simulation.

Running *CurveFitting*

Download the python script CurveFitting.py and set the working directory as done for *GenRis*. You can fit a curve to the output from *GenRis* and estimate the genotypic richness for a specific population in python with the following commands:

```
from CurveFitting import curve_fitting

curve_fitting(input = "Probs_for_Sample.csv", prop_iMLG = x)
```

Alternatively, if you want to run the script from a unix terminal, you can use:

```
python3 -c 'from CurveFitting import curve_fitting; curve_fitting(input = "Probs_for_Sample.csv", prop_clones = x)'
```

The CurveFitting.py file needs to be stored in the same folder where the "Probs_for_Sample.csv" file from the *GenRis* simulation is saved and where you want to create your output files. The observed proportion of identical MLGs in your sample (number of clones / number of isolates) has to be indicated as prop_iMLG and needs to be larger than 0. For this simulation, the number of clones is defined as the differences between the sample size (number of isolates) and the observed number of distinct multi-locus genotypes.

This script will plot fitteded curves through the mean probabilities and σ. It will also provide the R^2 value for the fit of the mean curve, the average number of expected genotypes in your sample, and the lower and upper percentiles (σ) to this estimate.

Example:

```
from CurveFitting import curve_fitting

curve_fitting(input = "Probs_for_Sample.csv", prop_iMLG = 0.153)
```