GenRis User Manual

Simulating the probability of picking clones and unique genotypes in phytoplankton populations to estimate Genotype Richness.

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, os, random, pandas and seaborn are required to run this simulation. You can install them from a unix terminal using the command:

```
python -m pip install numpy
```

Or within python using:

```
pip install numpy
```

Running *GenRis*

To run this simulation in python, you first need to set the working directory where the GenRis.py file is located and where the output files will be generated. You can use the python module os to check and set your current working directory.

```
import os

os.getcwd()

os.chdir('path/to/folder/with/GenRiS.py')
```

To run the simulation, 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), samplesizes (number of cells that will be picked) and standard deviations of the normally distributed growth rate function (sd). Larger sd will result in larger differences between growth rates of individual genotypes. 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.
Only two variables with ranges of values are allowed per run!

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. The variable "days" refers to the time period of interest, in which the algal population grows exponentially.
The simulation will create a table and a heatmap showing the mean probability of picking clones across the range of your selected variables (numgenos, samplesize or sd).

If you're interested in estimating the genotype richness for a specific observed proportion of clones, you have to add the size of your sample (in which you counted the clones) 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 clones 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 clones in a range of population sizes and a set sample size to estimate genotype richness based on an observed proportion of clones.

<u>Requirements</u>
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, os and pandas are required to run this simulation. Install these modules as explained above.

<u>Running *CurveFitting*</u>
Set the working directory as done for *GenRis*. You can fit a curve to the output from *GenRis* and estimate the genotype richness for a specific population in python with the following commands:

```
from CurveFitting import curve_fitting
```

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

Alternatively, if you want to run the script from a 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 clones in your sample (number of clones / number of isolates) has to be indicated as prop_clones. 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 create a plot of the curves fitting 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 confidence intervals (σ) to this estimate.

Example:

```
from CurveFitting import curve_fitting
```

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