**TOHORĀ CULTURAL SHOCK CODE: REASONING BEHIND EACH CHOICE**

By Marina Klemm, Casual Research Assistant (Sep2023 – Jan2024)

Contact: [marinaklemm@gmail.com](mailto:marinaklemm@gmail.com)

## **DOCUMENTS**

The final code *whalePatch.py* runs with a full bottleneck simulation, with successfully loaded haplotypes, nitrogen and carbon isotopes, mother and father IDs, while also respecting reproductive age and life expectancy. The final obstacle was the fact that simuPOP did not know how to handle the population when the offspring had less individuals than the parental generation, which was the opposite of what was needed in a bottleneck situation. After [quite a bit of back and forth with Bo Peng](https://github.com/BoPeng/simuPOP/issues/114), he created a patch for me and I was finally able to install it. So, to be able to run the simulation, the latest version of simuPOP has to be installed first. The files *snp\_1.txt* and *mtDNA\_1.txt* also have to be in the same directory where simuPOP runs.

Although I was able to run the whole simulation on Windows using Anaconda + Spyder, this new patch proved really hard to install. I ended up having to use Ubuntu on a virtual machine and only then I was able to make it work. Since this was not straightforward for me, the code I used follows in a separate document, *installPatch.txt*.

## **RESULTS**

I edited the code to create a simulation and run it for 74 generations as a “burn in” time. Then, I had the code print out a csv file for each generation from the 75th (the last one before the bottleneck) to the 82nd generation (the contemporary one). I used one generation = 25 years.

In a first population dump before running the simulation, I could verify that the 57 initial haplotypes were present and loaded in the population. Then, in population 75, 2 haplotypes had been lost. Gradually there were less haplotypes in each generation, with the final one having a loss of 37 haplotypes in total, only presenting 20.

Another check I did was to track a random 8 year old whale from generation 82 (individual id 73930) back in time until generation 75, where it was just born. I could see that its mother and father stayed the same in each generation, while its age decreased as I went back in time. One thing to note is that when it was 8 years old, in the latest generation printed out, it produced an offspring of 62 whales (since it was in the mother id of all these 62 whales). While that does not sound realistic, I’m not sure how the program could otherwise compute that quick change in the number of individuals from one generation to the other. An alternative is to reduce the number of years in each generation and have more of them, but that also requires a bit of fine tuning in selecting the number of individuals in each generation. All the population output files are named *dump\_gen\_75.csv*, *dump\_gen\_76.csv,* etc, and the individual I tracked follows highlighted in the csv files.

## **METHODS**

* **Bottleneck**

I used some functions from Juhana Kammonen’s code to select the population size in each generation. After checking, each sim.dump(pop) has the right number of individuals expected for said generation. The models I tried before, e.g. instant change or linear growth models, are in *removedModelsWhale.py*.

* + **Trajectory:**

Trajectory seems to be the most finicky part of this code. There are many ways for it to break:

(1) At first, I thought it was limited by my notebook’s computing capabilities;

(2) Then, I assumed that the bottleneck was so aggressive that it affected the population in a way that did not allow the code to run;

(3) Now, I understood that it’s just a matter of the allele frequencies not being able to fit the endFreq based on the beginFreq. It is frustrating because both values are required and I was not able to start it with a list of haplotypes. Based on the cookbook, trajectory in simuPOP was created to simulate human diseases and it require too many adjustments to work for us. For now, I removed it from our code but the chunks used before are in *removedChunksWhale.py*.

* + **Nitrogen and Carbon isotopic values:**

Initialized from the original code, using mean, variance and deviant proportion (10%) for N and C separately. Seem to be properly coded in each printed out population, within the expected ranges for both of them.

* + **Haplotype frequencies:**

The last chromosome accounts for the different mitochondrial haplotypes; note that the last loci is always zero, since it is, by definition, haploid data. The *freq\_mtDNA.txt* file prints out the frequency of the 20 remaining haplotypes.

* + **Mating Scheme**:

Within the pop.evolve function, a few mating schemes were added, and each one requires a weight to be added to them. From the [simuPOP manual](https://simupop.sourceforge.net/manual_release/build/userGuide_ch6_sec1.html?highlight=weight), the negative weights are processed before the positive, which justifies the choice of weight = -1 for the clone mating (since it refers to the mitochondrial haplotype).

In that same chunk of code, 6 virtual subpopulations were chosen out of the 8 present (adult male, juvenile female, etc) to account for the two subpopulations that were deceased and, therefore, not included.

## **REFERENCES**

Juhana Kammonen kindly sent me the code that his group used to simulate the Finnish human population evolution, which was useful to simulate its bottleneck. He asked us to cite his most recent work in a future publication using simuPOP. Reference:

Sundell et al (2012) - Retracing Prehistoric Population Events in Finland Using Simulation. *CAA2012 Proceedings of the 40th Conference in Computer Applications and Quantitative Methods in Archaeology, Southampton, United Kingdom, 26-30 March 2012.*