Advanced LAMARC exercise: Module 21, Summer Institute 2014

How to improve a poorly performing run?

We analyzed the Lipoprotein Lipase data yesterday and saw generally poor performance of the algorithm (bad-looking Tracer results). Today we'll try to improve our results while still getting the program to run relatively quickly. Listed below are several ways to improve a bad run; choose a few and try them. At the end of the session we will compare notes and see which methods were successful.

Here is the run that did badly yesterday:

To estimate subpopulation parameters you will need data with more than one population. We provide a file `lpl.xml` with SNP data from three human populations: European Finns from North Karelia, assorted Europeans from Rochester, Minnesota; and African-Americans from Jackson, Mississippi. A shorter version, better for class exercises, is in `lpl_short.xml`. This is a complete lamarc infile to perform a Bayesian analysis with one final chain sampling 7000 parameter values split randomly among all of the available parameters.

Here are some strategies to try:

Based on what you know, select a few of these strategies to try. Keep an eye on the predicted runtimes so that you don't start something you can't finish!

1. Increase the number of steps in each chain (in the Search Strategy menu).
2. Add a heated chain (also in the Search Strategy menu). I recommend a temperature of 1.2 for the hot chain.
3. Restrict the migration model (in the Analysis menu) so that migration rates are symmetrical, or so that some migration pathways are not allowed. For example, it is probably reasonable not to estimate gene flow from the Jackson population (African-American) into the Karelia population (Finnish).
4. Fix one or more of the Theta values at a pre-determined figure so that it will not have to be estimated. A previous study suggested Theta(Jackson)=0.0072, Theta(Karelia)=0.0027, Theta(Rochester)=0.0031. This is done in the Analysis menu.
5. If you were estimating recombination, turn it off (Analysis menu).
6. Change the priors (Search Strategy menu) to make them narrower, so that a smaller range of values will have to be considered. (This may improve the rate at which proposed values are accepted.)
7. Change from a Bayesian to a likelihood run. I don't expect this to help, but in theory it might. If you do this you will have two more things to consider changing:
    - Run more chains (Search Strategy menu).
    - Fine-tune your starting values (Analysis menu).
8. Reduce this three-population case to a two-population case. You will need to feed the raw data files into the converter to make a new, two-population Lamarc input file.
9. If you are ambitious, you can try randomly removing some sequences from the Lamarc input file using a word processor. Be careful to keep the rest of the XML intact.

10. Add population divergence, assuming that Jackson is the oldest population. You will need to re-run *lam_conv* to do this.

After each run, use Tracer (or the Lamarc output file, for a likelihood run) to judge your success. In a Bayesian run, look for:

- ESS greater than 200 for all parameters.
- Traces that level out and stay level for the majority of the run.
- Histograms that have only one peak, and look fairly smooth.
- Histograms that are not scrunched up against one side of the diagram.
- Constancy of results from one run to another (with different random number seeds; compare with your neighbors!)

In a likelihood run, look for:

- Parameter estimates that level out rather than continuing to rise or fall.
- Data lnL values that level out rather than continuing to rise or fall.
- Posterior lnL values in the last chain no greater than 2x the number of parameters.
- Constancy of results from one run to another.

Which strategies seem to improve run success the most? How badly did they slow the program down? Do you think they jeopardized the scientific value of the results?

An additional piece of information that may help: there is believed to be a recombination hotspot in the LPL locus.