Demographic inference with dadi

In this exercise, you will use dadi to fit demographic models to subsets of the 1000 Genomes data. First, we'll work together on fitting the single YRI population. Then you'll work more independently to fit models to pairs of populations. The data we'll fit are from synonymous mutations in coding regions, a reasonable (but imperfect) proxy for neutrally evolving sites.

In a new terminal window, change to the `dadiExercise/dadi` directory and edit the script I've provided for you: `nano fitYRI.py`. You'll be modifying this script as we work together.

In another terminal window, change to the `dadi` directory, and start iPython with `ipython --pylab`. In the iPython window, you can run your script at any time with `%run fitYRI.py`. (Be sure to save your script when you change it!)

1. First, quickly read through the script, especially the comments. We're fitting data from the Yoruba population in Nigeria. Notice that I've commented out large bits of code. This is because we'll begin with the standard neutral model (SNM) and work our way up to more complex models.

   Run the script, using `%run` in iPython. In the plot, the top panel compares your data (in blue) to the model (in red). The bottom panel shows the residuals, a standardized measure of how much the model deviates from the data. In the perfect case, these would be uncorrelated and roughly normally distributed, with standard deviation 1.

   Does this model show systematic differences from the data? If so, what are the differences?

2. Next, we'll consider a more complex model that allows an instantaneous size change some time in the past. To implement this model, comment out the lines pertaining to the SNM (lines 21&22). Then uncomment line 27, to set the demographic function `func` to be dadi's built-in two-epoch model. Lastly, uncomment lines 57-64 to set up a parameter optimization.

   Run your script in iPython. What maximum-likelihood parameters do you infer, and what is the maximum likelihood? Look at the plot. Has the fit improved qualitatively? Are there still regions of the frequency spectrum that the model fits poorly?

3. You probably saw that the remaining disagreement between model and data is in high frequency alleles. This is a sign of misidentification of the ancestral state (most likely due to mutation on the branch to the outgroup). We could solve this problem by folding the spectrum or by statistically correcting the spectrum for the expected amount of misidentification. Here, we're going to add a parameter to our model to account for misidentification. This is an example of a custom dadi model.

   To continue your analysis, first comment out line 27 to turn off the original two-epoch model, then uncomment lines 34-45 to enable the misidentification model.

   We'll also need to change the optimization code to account for the additional parameter. First append `0` to the lower`_bound` array and `1` to the `upper_bound` array. Also append a guess for the misidentification parameter to the first argument of `perturb_params`. Try a starting guess of `0.1`.

   Run your script. (You may need to run it a few times until the misidentification parameter converges to something non-zero.) Look at the plot. Has the fit improved qualitatively? How do the time and extent of the size change compare with your previous fit? How does the likelihood compare?

4. Your best-fit parameters should be a growth of roughly a factor υ=3.1 which occurred T=0.16 time units ago. Now let's convert those to non-genetic units.

   a. In dadi, all parameters are scaled by the effective size of the ancestral population. To find that effective size, we use the relationship $\theta = 4N_e\mu L$, where μ is the per-base mutation rate, and L is length of sequence from which the SNPs came. So $N_e = \theta/4\mu L$. For this data $L \sim 25$ Mb. A reasonable estimate for the human mutation rate is $\mu = 2\times10^{-8}$ per base per generation. Using your estimate for θ, what is your estimate for N_e?

   b. Our estimate for the contemporary effective population size is then $\upsilon\times N_e$. What is your estimate for this population size?

   c. The units of T are $2N_e$ generations, so to convert to years, we multiply T by $2N_e$ and the generation time. A reasonable estimate for the human generation time is 25 years. What is your estimate for the time of the size change?

5. Lastly, we want to estimate the uncertainties of our parameters. The most robust way to do this is bootstrapping. To do that, we would resample the data many times and repeat the fitting procedure many times. This is very computationally expensive.

   An efficient approximation is to use derivatives of the likelihood function, but this isn't valid for the composite likelihood that dadi calculates with linked data. A useful approach is to calculate the Godambe Information Matrix and estimate uncertainties through that. This relies on bootstrapping the data, but the computations can be made very efficient in dadi[1].

   Computationally, we can do this for your model by uncommenting lines 85-93 in the script.

   Run your script. What uncertainties do you infer for your model parameters? Are they large are small relative to the parameter values themselves?

Now we'll turn to fitting two-population models. Close your `fitYRI.py` script and open `fitTwoPop.py` in nano. The script is set up with a demographic model in which the two populations split then grow exponentially, potentially with migration between them. Notice the `fixed_params` option to the optimization function, which is an easy way to hold some parameters constant.

6. Run the script in iPython. As I gave it to you, the script only fits the divergence time between the two populations. What time do you infer? What is the likelihood? Look at the residual plot. Red indicates that the model predicts too many SNPs in that frequency class, blue means too few. You should see that the model struggles to reproduce the private SNPs in each population, and it overpredicts the amount of shared polymorphism.

7. Experiment with allowing more parameters to vary. Which gives the greatest improvement in likelihood? How much better is the likelihood if all parameters are allowed to vary? (Note that when more parameters are optimized, you may need to increase the number of optimization iterations to reach convergence, using the `maxiter` argument. Also remember to run several optimizations per case, to gain confidence that you've actually found the maximum likelihood.)

[1] For details, see AJ Coffman, P Hsieh, S Gravel, RN Gutenkunst "Computationally efficient

8. OPTIONAL: In the directory `YRI_3_pop_fs_syn` there are spectra for many other sets of populations that you can experiment with. Pick a spectrum, and modify the `from_file` and `marginalize` commands to pick the populations you want to work with. How well can the IM model fit those populations? (The population ids are detailed at http://www.1000genomes.org/category/frequently-asked-questions/population.)