PhyloMath Lecture 1

by Paul O. Lewis, 22 January 2004

Simulation of a single sequence under the JC model

We drew 10 uniform random numbers to simulate a nucleotide sequence 10 sites long. The JC model specifies that each base has a relative frequency of 0.25, so the following table was used to decide which base to insert for each random number drawn:

Base	Range		
A	0.00	to	0.25
C	0.25	to	0.50
G	0.50	to	0.75
T	0.75	to	1.00

Below is a table showing the random numbers drawn and the data generated (not the exact values presented in class, I made up new numbers):

Random number	Base chosen
0.644	G
0.783	T
0.752	T
0.717	G
0.757	T
0.307	C
0.079	A
0.471	C
0.093	A
0.998	T

Likelihood under the JC model using the "JC" sequence

The likelihood for the sequence simulated in the previous section (`GTTGTCACAT`) under the JC model is just

$$L_{JC} = (\tfrac{1}{4})(\tfrac{1}{4})(\tfrac{1}{4})(\tfrac{1}{4})(\tfrac{1}{4})(\tfrac{1}{4})(\tfrac{1}{4})(\tfrac{1}{4})(\tfrac{1}{4})(\tfrac{1}{4}) = (\tfrac{1}{4})^{10} = 9.536743164 \times 10^{-7}$$

The likelihood is the same numerical value for every possible sequence under the JC model. The question was raised: "Why wouldn't the JC model specify a smaller likelihood for the sequence GGGGGGGGGG than for the one we simulated; it seems much less probable to have 10 Gs in a row than to have a sequence with at least one representative of each of the four bases." The answer is that it really *is* just as probable to see `GGGGGGGGGG` as any other *particular* sequence, including `GTTGTCACAT`. The reason this is not intuitive is that in our minds we (mistakenly) think of sequences as falling into two groups: "monomorphic" sequences like `GGGGGGGGGG` and "polymorphic" sequences like `GTTGTCACAT`. Our minds correctly figure out that the probability of simulating one of the four monomorphic sequences is small compared to the probability of simulating one of the polymorphic sequences (of which there are $4^{10} - 4!$). The chance of simulating the *particular* sequence `GTTGTCACAT` is, however, identical to the chance of simulating the *particular* sequence `GGGGGGGGGG`.

Simulation of a single sequence under the F81 model

We next drew 10 uniform random numbers to simulate a second nucleotide sequence 10 sites long, this time using the F81 model, which allows the relative nucleotide frequencies to be unequal. We decided (arbitrarily) to let $\pi_A = 0.1$, $\pi_C = 0.2$, $\pi_G = 0.3$, and $\pi_T = 0.4$. The following table was used to decide which base to insert for each random number drawn:

Base	Range		
A	0.0	to	0.1
C	0.1	to	0.3
G	0.3	to	0.6
T	0.6	to	1.0

Below is a table showing the random numbers drawn and the data generated (again, these are not the exact values presented in class):

Random number	Base chosen
0.511	G
0.632	T
0.601	T
0.739	T
0.627	T
0.766	T
0.125	C
0.480	G
0.599	G
0.978	T

Likelihood under the F81 model using the "F81" sequence

The **empirical base frequencies** for a sequence are computed as simple proportions, using the number of each base observed in the sequence, divided by the total number of sites. The empirical frequency is 0.0 for base A (no As were observed), 0.1 for base C (1 C was observed out of 10 sites), 0.3 for G, and 0.6 for T. Below, the likelihood under the F81 model is computed for this sequence using the empirical nucleotide composition:

$$L_{F81} = (0.3)(0.6)(0.6)(0.6)(0.6)(0.6)(0.1)(0.3)(0.3)(0.6) = 1.259712 \times 10^{-4}$$

This can be written as a general formula as follows:

$$L_{F81} = \pi_A^{n_A} \pi_C^{n_C} \pi_G^{n_G} \pi_T^{n_T}$$

Here, π_i is the relative frequency of base i, and n_i is the number of times base i was seen in the sequence. Plugging in the empirical values 0.0, 0.1, 0.3, and 0.6 for π_A, π_C, π_G, and π_T, respectively, we get the same answer as before (1.259712×10^{-4}). Plugging in the **true** frequencies, we get

$$L_{F81} = \pi_A^{n_A} \pi_C^{n_C} \pi_G^{n_G A} \pi_T^{n_T} = (0.1)^0 (0.2)^1 (0.3)^3 (0.4)^6 = 2.21184 \times 10^{-5}$$

Thus, the probability of the sequence GTTTTTCGGT under the F81 model is higher using the empirical frequencies than it is using the true frequencies. This will always be the case, because if the data consist of just one sequence, the empirical base frequencies are the *maximum likelihood estimates* of the relative nucleotide frequencies. This means that one cannot make the likelihood any higher using any other combination of relative frequencies, including the true combination.

Likelihood under both models using the "JC" sequence

The difference between the JC model and the F81 model lies in the fact that the F81 model allows any combination of base frequencies, whereas the JC model forces all the base frequencies to be equal: i.e. $\pi_A = \pi_C = \pi_G = \pi_T = 0.25$. The likelihood computed under a certain model tells us how well the model fits the data. We might reasonably expect the JC model to fit the "JC" sequence better than the F81 model, and vice versa, but there is so little data in this case (10 sites) that this result is certainly not guaranteed. Also, note that it is impossible for the JC model to beat the F81 model in this game, because the F81 model can be made to equal the JC model by stipulating that the base frequencies are all equal. So the only way for JC to win is to prevent F81 from setting its base frequency parameters to anything it chooses.

We have already computed the likelihood for the "JC" sequence (GTTGTCACAT) under the JC model: $9.536743164\times 10^{-7}$. The likelihood under the F81 model for the same sequence (using the empirical frequencies) is:

$$L_{F81} = \pi_A^{n_A}\pi_C^{n_C}\pi_G^{n_GA}\pi_T^{n_T} = (0.2)^2(0.2)^2(0.2)^2(0.4)^4 = 1.6384\times 10^{-6}$$

The probability of the data (i.e. the likelihood) is higher under the F81 model, even though the data were generated using the JC model.

Exercise 1: compute the likelihood for the "JC" sequence under the F81 model using the true frequencies (that is, 0.25 for all four bases)

The F81 likelihood should be identical to the JC likelihood in this case, namely $9.536743164\times 10^{-7}$.

Exercise 2: compute the likelihood for the "JC" sequence under the F81 model assuming $\pi_A = 0.4$, $\pi_C = 0.3$, $\pi_G = 0.2$, and $\pi_T = 0.1$?

These frequencies are pretty far from the true ones, so we might expect the F81 likelihood to be less (i.e. worse) than the JC likelihood in this case. The answer I got for this was 5.76×10^{-8}, which is indeed quite a bit worse than the JC likelihood.

Likelihood under both models using the "F81" sequence

Doing the same exercise for the "F81" sequence (GTTTTTCGGT), we again get $9.536743164\times 10^{-7}$ for the JC model (all sequences of length 10 have the same likelihood under the JC model) and 2.21184×10^{-5} for the F81 model (using the true frequencies). So this time, the model actually used to simulate the sequence is indeed the best-fitting model. The fit of the F81 model is even better if the empirical frequencies are used: 1.259712×10^{-4}. Using the empirical frequencies "tunes" the F81 model to fit the data as well as possible. In this case, the sequence GTTTTTCGGT is 132 times more probable under the F81 model than it is under the JC model (0.0001259712 is 132 times larger than 0.0000009536743164).

Logarithms

Logarithms are useful for representing likelihoods especially for large amounts of data. The probability that an A will be seen at the first position of a sequence and a T will be seen at the second position is the probability of a coincidence. The more players (i.e. sites) involved in the coincidence, the smaller will be the probability of that coincidence. There is nothing unusual about this: just think of the probability of seeing three of your friends while sitting at the stoplight at Four Corners. All of your friends (and you) probably have perfectly good reasons for being at Four Corners at some time during the day, but the coincidence is always going to be much less probable than any individual component. With sequences, the probabilities of coincidences involving thousands of sites are very tiny numbers indeed. It doesn't mean these numbers are not useful, but they can become so small that computers cannot distinguish them from zero. Logarithms allow such tiny numbers to be manipulated accurately.

Common logarithms are defined to be the power of 10 that is needed to represent some number. For example, the common logarithm of the value x (written $\log(x)$) is the value y such that $10^y = x$. This means that the following is a true statement:

$$x = 10^{\log(x)}$$

This is a very useful tautology. Natural logarithms are much more common in the sciences, and behave just like common logarithms but use the base e instead of the base 10. Thus

$$x = e^{\ln(x)}$$

The value e is a constant approximately equal to 2.718281828. You can use your calculator to get the value of e by entering the number 1 and pressing the e^x key.

Logs of products

The (natural) log of a product is the sum of the logs of the individual terms in the product. Thus, logs turn products into sums. For example, for any two real numbers a and b,

$$\ln(ab) = \ln(a) + \ln(b)$$

For example,

$$\begin{aligned} \ln((2.2)(4.4)) &= \ln(9.68) = 2.270061901 \\ \ln(2.2) &= 0.78845736 \\ \ln(4.4) &= 1.481604541 \\ \ln(2.2) + \ln(4.4) &= 2.270061901 \end{aligned}$$

Why is this true? Starting with ab, replace a and b with their equivalents using the "tautology" and simplify the resulting expression using the rules of exponents:

$$\begin{aligned} ab &= \left(e^{\ln(a)}\right)\left(e^{\ln(b)}\right) \\ &= e^{\ln(a)+\ln(b)} \end{aligned}$$

The last step just involves invoking the tautology once more, realizing that the power to which e on the right side is raised (i.e. the quantity $\ln(a) + \ln(b)$) must be (by definition) the natural logarithm of ab. In other words, if

$$ab = e^{\ln(ab)}$$

then $\ln(ab)$ must be equal to $\ln(a) + \ln(b)$ (because they both equal ab).

Logs of quotients

The same process can be used to derive a rule for quotients. Starting with the quotient a/b,

$$\frac{a}{b} = \frac{e^{\ln(a)}}{e^{\ln(b)}} = e^{\ln(a)-\ln(b)}$$

Thus, $\ln(a/b)$ must equal $\ln(a) - \ln(b)$.

Log-likelihoods

The log-likelihood (commonly abbreviated lnL) is the natural logarithm of the likelihood. For the JC likelihood of one sequence 10 sites in length,

$$\ln L = \ln\left((0.25)^{10}\right) = \ln\left[\left(e^{\ln(0.25)}\right)^{10}\right] = \ln\left(e^{10\ln(0.25)}\right) = 10\ln(0.25) = -13.86294361$$

Note that you can recover the now-familiar $L_{JC} = 9.536743164 \times 10^{-7}$ by entering $\ln L = -13.86294361$ into your calculator and pressing the e^x button. The exponential function is the inverse of the natural log function, and the two cancel each other out, leaving just the likelihood L: $e^{\ln L} = L$.

A general formula for the log-likelihood of a sequence under the F81 model would be:

$$\ln L = n_A \ln(\pi_A) + n_C \ln(\pi_C) + n_G \ln(\pi_G) + n_T \ln(\pi_T) = \sum_{i \in \{A,C,G,T\}} n_i \ln(\pi_i)$$

Simulation of two sequences separated by time t and evolving at rate α

We finished by beginning a discussion on evolving one sequence from another sequence. Two more quantities must be introduced before the simulation can be done. Assume the time t is 10 and the substitution rate α is 0.01. The product αt is thus 0.1. This product represents the expected number of substitutions. Thus, for $\alpha t = 0.1$, we expect one substitution to occur for every 10 sites. For the JC model, the expected number of substitutions is actually $3\alpha t$, but *not for the reason I proposed in class.* The real reason is because there are three stochastic substitution processes going on simultaneously (the base in the ancestral sequence can change to any one of the three other bases), and the expected number of substitutions is the same for all three (i.e. αt). Adding the expected number of substitutions from all three processes together gives you the $3\alpha t$.

Starting with the "JC" sequence created at the beginning of this lecture (`GTTGTCACAT`), simulating the evolution of a second sequence involves **transition equations**, which for the JC model are:

$$\begin{aligned} P_{ii}(t) &= \tfrac{1}{4} + \tfrac{3}{4}e^{-4\alpha t} \\ P_{ij}(t) &= \tfrac{1}{4} - \tfrac{1}{4}e^{-4\alpha t} \end{aligned}$$

These equations tell us that the probability of a site ending up with a *different* base than it started with over a time $t = 10$ when it is evolving at a substitution rate $\alpha = 0.01$ is $0.25 - 0.25e^{-0.4} = 0.25 - (0.25)(0.670320046) = 0.08242$. The probability of ending up in the same base as it started with is just $1 - (3)(0.08242) = 0.75274$. Calculating it the hard way, you get the same answer: $0.25 + 0.75e^{-0.4} = 0.25 + (0.75)(0.670320046) = 0.75274$.

For the first site, the starting base is G, and to simulate the evolution of this site, we would set up a table for deciding how to process the uniform random numbers as follows:

Base	Range		
A	0.0	to	0.08242
C	0.08242	to	0.16484
G	0.16484	to	0.91758
T	0.91758	to	1.0

Note that this table is designed specifically for a starting base of G (the probability of ending with a G is the largest slice of the pie in this case, a little over 75%). If we were simulating the second site, which starts with a T, the table would look like this instead (with the largest slice of the pie going to T as the base at the far end of the branch):

Base	Range		
A	0.0	to	0.08242
C	0.08242	to	0.16484
G	0.16484	to	0.24726
T	0.24726	to	1.0

It is of course important to establish these rules *before you begin drawing random numbers*! Otherwise, you might be sorely tempted to cheat and start inserting your favorite base too many times. This process is exactly how programs such as SeqGen simulate sequence data. Programs like SeqGen allow one to use a variety of models, of course, but this just involves using different transition probability equations, which differ among models. Simulating data on a tree involves simply repeating this process for each branch in the tree, each time starting with the sequence you were left with in the last step.

Here is a second sequence I simulated starting with an ancestral sequence consisting of 10 Gs. The simplicity of the ancestral sequence meant that I could use the first table above for all 10 sites.

Random number	Base chosen
0.670	G
0.333	G
0.173	G
0.733	G
0.524	G
0.890	G
0.400	G
0.156	**C**
0.591	G
0.708	G

Only one site (8) ended up with a different base (C); all the other sites did not change over the branch. The expected number of substitutions was $3\alpha t = 0.3$. The transition equations take account of hidden substitutions, so the fact that we only detected one substitution when three substitutions were expected out of 10 sites should not worry us. If one of the other sites had experienced a substitution from G to say A and back again, the ending base would still be G and we would not be able to detect these additional two substitutions. I will show you another method of simulating data next time that allows us to see these hidden substitutions. This alternative method forms the basis of the "character mapping" approach advocated by Rasmus Nielsen in several recent papers (Huelsenbeck et al., 2003; Nielsen, 2002).

Literature Cited

- Huelsenbeck, J. P., R. Nielsen, and Bollback, J. P. 2003. Stochastic mapping of morphological characters. *Systematic Biology* **52**(2): 131–158.

- Nielsen, R. 2002. Mapping mutations on phylogenies. *Systematic Biology* **51**(5): 729–739.