Lecture 6B – Optimality Criteria: Maximum Likelihood & Minimum Evolution

II. Optimality Criteria – Continued

B. Maximum Likelihood – Chapter 16

ML estimation, in general, is a standard and incredibly useful statistical procedure. There is an excellent introduction to it in the text, on pages 248-251.

1) The approach focuses on calculating the likelihood (L), which is the probability of the data given a particular hypothesis:

$$L = \Pr(\text{data} \mid \text{hypothesis}).$$

Typically, several hypotheses are evaluated, and the one that maximizes the probability of having generated the observed data is the ML estimate.

2) Calculation of these probabilities requires an **explicit model**, so really (although it's rarely done) we should write the expression as such:

$$L = \Pr(\text{data} \mid \text{hypothesis}, \text{model}).$$

Historically, this limited its application in phylogenetics to molecular data, however the application of likelihood to morphology has been increasing and is now pretty common.

3) So, the application of ML as an optimality criterion in phylogeny estimation is as follows:

$$L_{(\tau)} = \Pr(D \mid \tau, m).$$

This is simply the probability of the data (the set of aligned sequences), given the tree, assuming some model of character evolution (much more on this later).

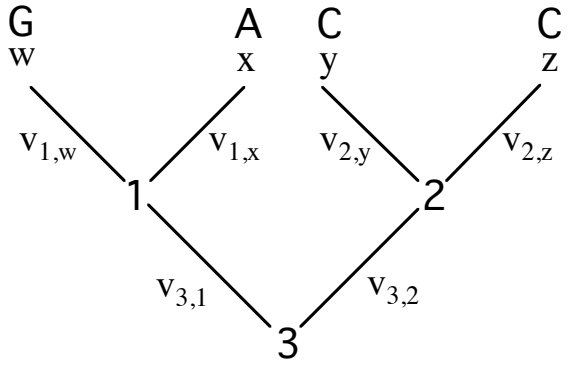
Just as in parsimony, we make the assumption that characters are independent so that we can treat likelihoods for each site separately. If we have s sites, the likelihood of a tree is:

$$L_{(\tau)} = \prod_{i=1}^{s} \Pr(D^i \mid \tau) = \prod_{i=1}^{s} L^i(\tau),$$

where $\Pr(D^i \mid \tau)$ is the probability of site *i* given tree τ. This is the **single-site likelihood** and is analogous to the single-character length in parsimony.

So, just as the parsimony score for a tree across an entire data set is the *sum of the character lengths*, the likelihood of a tree across an entire data set is the **product of the site likelihoods**, and just as in parsimony, we do this because we assume independence of sites.

Let's demonstrate how single-site likelihoods are calculated with the same (arbitrarily rooted) tree we just used to demonstrate parsimony optimizations.



Note that we're now paying attention to the branches that and we will incorporate their lengths, which are labeled $v_{x,y}$. The units are expected number of substitutions per site and they are a function of rate of evolution times the duration of branch (rate x time).

So, to calculate the single-site likelihood for this site, we sum the probabilities for all possible character-state reconstructions. Since there are $n - 1 = 3$ internal nodes (for a rooted tree) and 4 possible character states at each node, there are $4^{n-1} = 4^3 = 64$ possible reconstructions. So:

$$L^i(\tau) = \text{Prob}\left(\begin{array}{ccccc} G & & A & C & C \\ & A & & A & \\ & & A & & \end{array}\right) + \text{Prob}\left(\begin{array}{ccccc} G & & A & C & C \\ & A & & A & \\ & & C & & \end{array}\right)$$

$$+ \text{Prob}\left(\begin{array}{ccccc} G & & A & C & C \\ & A & & A & \\ & & G & & \end{array}\right) \bullet\bullet\bullet + \text{Prob}\left(\begin{array}{ccccc} G & & A & C & C \\ & T & & T & \\ & & T & & \end{array}\right)$$

Or,

$$L^i_{(\tau)} = \sum_r^{4^{n-1}} \Pr(R^i_r \mid \tau)$$

Of course, many reconstructions are extremely unlikely, so they'll contribute very little to the single-site likelihood, but we still consider them as possibilities and account for them.

So now the issue is how one calculates the probabilities of a particular reconstruction.

Let's index reconstruction r (R_r) such that m is the state at node 3,
k is the state at node 1, and
l is the state at node 2.

We know (from our data) that nodes w, x, y, & z (the terminals) have states G, A, C & C, respectively.

$$\Pr(R_r \mid \tau) = \pi_m \times P_{m,k}(v_{3,1}) \times P_{k,\mathrm{G}}(v_{1,w}) \times P_{k,\mathrm{A}}(v_{1,x}) \times P_{m,l}(v_{3,2}) \times P_{l,\mathrm{C}}(v_{2,y}) \times P_{l,\mathrm{C}}(v_{2,z})$$

where:

π_m is the frequency of the nucleotide m. This provides an estimate of the probability of observing state m at the root node.

$P_{i,j}$ is the probability of substitution between states i & j. This is derived the model of sequence evolution that we assume (much more on those in a few weeks). This is in some sense analogous to the step matrix that determines costs for transformations between states in the Sankoff algorithm in parsimony.

So, this can be calculated from all 4^{n-1} reconstructions and summed to calculate the single-site likelihoods.

However, there's a more efficient way to calculate the same value.

$$L^i_{(\tau)} = \sum_m \pi_m \times \left(\sum_k P_{m,k}(v_{3,1}) P_{k,G}(v_{1,w}) P_{k,A}(v_{1,x}) \right) \times \left(\sum_l P_{m,l}(v_{3,2}) P_{l,C}(v_{2,y}) P_{l,C}(v_{2,z}) \right)$$

Each summation is across all four nucleotides.

Notice that the structure of the parentheses conforms to the subtrees that comprise the tree we're evaluating. This is the essence of **Felsenstein's pruning algorithm**, which you may have heard of.

Also note that *branch lengths* are parameters that can be *optimized*. That is, we take an initial set of branch lengths and try to improve them numerically using, say, the Newton-Raphson algorithm. It turns out that this is also very computationally expensive, so different approaches (programs) spend varying amounts of computational resources to do this.

Almost all applications you'll see, rather than deal with a series of products, we typically take the natural logs and sum them:

$$\ln L(\tau) = \sum_{i=1}^{s} \ln L^{i}_{(\tau)}$$

Another level of efficiency can be achieved by realizing that multiple sites may have the same **site pattern**.

```
  1       A G T A C A . . . . . . . . . . . . . . . .
  2       A G T A . . . . . . . . . . . . . . . . . .
  3       A G T A . . . . . . . . . . . . . . . . . .
  .       . . . . . . . . . . . . . . . . . . . . . .
  .       . . . . . . . . . . . . . . . . . . . . . .
  n       A G T A . . . . . . . . . . . . . . . . . .
Pattern   1 2 3 1 . . . . . . . . . . . . . . . . .(a)
```

There are 4^n possible site patterns. Rather than recalculate the single-site likelihood for sites with the same pattern, the frequency of each pattern is stored. Thus,

$$\ln L(\tau) = \sum_{a=1}^{4^n} f_a (\ln L^a(\tau))$$

Yet a third level of efficiency was developed years ago by Kosokovsky-Pond & Muse (2004. Syst. Bio. 53:658). It's called column sorting and takes advantage of our assumption of independence of sites in the matrix (this assumption is made for most

models typically used – more later). This means that the order of sites is arbitrary and that sites can be taken in any order we find convenient.

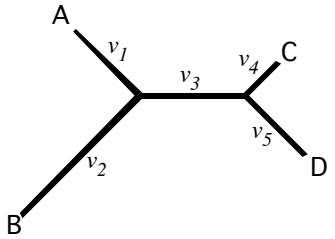
These authors place columns (sites) that are have states distributed similarly near each other in the matrix and retain computations that are redundant. This requires overhead in determining the best order in which to take sites, but that's more than recovered by reducing the redundancy of computation and time-savings range from 15% to 50%. *We'll discuss faster strategies to estimate the likelihood of a tree when we discuss searching tree space*.

C. Minimum Evolution (Chapter 11).

If data are either inherently distance-based, or are converted to a series of pairwise distances, one may choose to evaluate trees under the Minimum-Evolution criterion.

Remember that this matrix has $(n^2\text{-}n)/2$ cells (pair-wise comparisons).

Consider the following unrooted 4-taxon tree:



Note that, again, we're paying attention to branch lengths.

The optimality criterion in the original formulation is the sum of the absolute value of the branch lengths (Kidd & Sgaramella-Zonta, 1971. Amer. J. Hum. Gen. 23:235).

$$ME_{(\tau)} = \sum_{k=1}^{2n-3} | v_k |$$

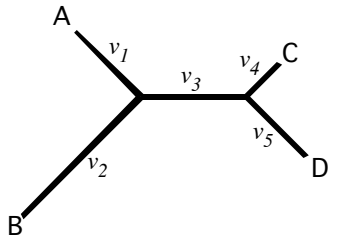
Or, in a subsequent formulation (Rhezetsky and Nei, 1992. Mol.Biol.Evol., 9:945), it is the sum of branch lengths.

$$ME_{(\tau)} = \sum_{k=1}^{2n-3} v_k$$

So, given a matrix of pairwise distances and a tree, how does one find the branch lengths?

Let's think for a minute about the data (pairwise distance matrix). If we can estimate a distance that perfectly captures the actual amount of evolution that has occurred since the two taxa last shared a common ancestor, the distances will exhibit the property of tree additivity, such that on the tree:

$d_{AB} = p_{AB} = v_1 + v_2$,
$d_{AC} = p_{AC} = v_1 + v_3 + v_4$,
$d_{AD} = p_{AD} = v_1 + v_3 + v_5$,
$d_{BC} = p_{BC} = v_2 + v_3 + v_4$,
$d_{BD} = p_{BD} = v_2 + v_3 + v_5$,
$d_{CD} = p_{CD} = v_4 + v_5$



Where p_{xy} is the **patristic distance** between taxa x & y, or the distance calculated from tee.
Another property of additive distances (that Saitou and Nei discussed) is the four-point metric condition.

$$d_{A,B} + d_{C,D} < d_{A,C} + d_{B,D}$$
$$\&$$
$$d_{A,B} + d_{C,D} < d_{A,D} + d_{,B,C}$$

The point is, if we could come up with such error-free distance measures, and therefore analyze perfectly additive distances, there would be a perfect fit between the observed distance matrix and the patristic distance matrix on the true tree.

So, for any combination of branch lengths, we can measure the distortion associated with fitting the distance matrix to the tree:

$$E_{(\tau)} = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \mid d_{ij} - p_{ij} \mid^{\alpha}$$

α is usually 2 (it may be 1 in some cases – if you suspect some distances are really poor estimates but don't know which ones), so this is usually a least-squares method.

w_{ij} allows weighting of the pairwise error terms.
$w_{ij} = 1$ assumes that the errors are identical across all d_{ij}.
$w_{ij} = 1/d_{ij}$ assumes that errors are proportional to d_{ij}.
$w_{ij} = 1/d^2_{ij}$ assumes that errors are proportional to the square root of d_{ij}.
$w_{ij} = 1/\sigma^2_{ij}$ weights by the expected variance in the d_{ij} (which may not be known).

So, optimum branch lengths for a particular tree are found by the least-squares criterion but remember that the Minimum-Evolution criterion operates on the sum of branch lengths.

Another point is that the optimization of branch lengths may actually result in negative branches, which have no biological meaning.

Probably the best way to deal with them is to set them to zero when summing branch lengths to estimate the ME score.

III. Relationships among optimality criteria.

Lots of the controversy in phylogenetics revolves around choice of optimality criteria, and lots of work (including some of my own) has been focussed on performance of the various criteria.

Certainly, these differences are important, but often lost in the debate are the various fundamental conceptual similarities that are common to certain sets of them.

Both MP and ML are character based (whereas ME is not).

Both ME and MP minimize the amount of evolution (i.e., sum of branch lengths).

Both ME and ML rely on an explicit model of sequence evolution.

So certainly, we should expect there to be a broad range of conditions across which the methods perform similarly, and that certainly has been shown to be the case.

Nevertheless, the differences among them are real, and it's incredibly informative to study the conditions under which the methods perform differently.

We'll do that later in the semester.