# Reconstructing the Full Internal State of a Molecular Phylogenetic Tree with Arbitrary, Branch-Dependent Substitution Probabilities

William H. Press

May 1, 2006

## 1 Introduction

This note is about how to get a full statistical reconstruction of the internal-node state probabilities and the Markov transition matrices of a phylogenetic tree, given only data at the leaf nodes (that is, data for extant taxa). Almost all of this is standard textbook stuff—except that I can't find any standard textbook, or even paper in the literature, that gives a clear exposition in modern language. While Reference [1] is a good starting point, it is too elementary. Reader, if you know of a more advanced published treatment, or have written one, please let me know!

By "modern" I of course mean Bayesian. The treatment here includes what is commonly known as *maximum likelihood (ML)* reconstruction. The relationship between ML and Bayesian methods is widely understood.

The data is taken to be a set of aligned positions in multiple genomes, each position being a base pair, or possibly a codon labeled by its amino acid. Missing data is allowed. In this note, we are *given* the topology and taxon node assignment of the phylogenetic tree. That is, the whole vast machinery of *finding* the correct tree is assumed already to have operated. Now, we only want to reconstruct the internal details of the (assumed) Markov model.

We make the usual (simple) assumption that the Markov model acts identically and independently on each aligned position. That is, we here ignore the possibility that substitution rates vary across the given set of nucleotide sites (e.g., as a gamma distribution [2]) and concomitant pitfalls known to be possible in such mixture models (see [3],[4]). However, there are several further common specializations that we *don't* make, instead treating the general case:

1. Our transition matrices (base substitution probabilities) can be different on each edge of the tree.

2. No special form (e.g., GTR, JC69, K80, F81, HKY85, T92, TN93) is assumed for the transition matrices.

3. The transition matrices need not be generated by an infinitesimal generator, nor have positive eigenvalues, nor have matrix logarithms.

## 2 Notation and the Example Tree

Let us try not to get bogged down in a notational morass. Nodes are labeled by uppercase roman letters. Figure 1 shows the example tree that we use throughout this note. When we refer to nodes O, E, F, I, J, K, L, X, Y, or Z, it is understood that we mean nodes related as in the figure. The number of leaf nodes is denoted $M$, so that there are $M-1$ internal nodes (including the root node) and $2M-2$ edges or branches. The number of aligned positions, that is, the number of separate copies of the tree implied by the data set, is denoted $N$.

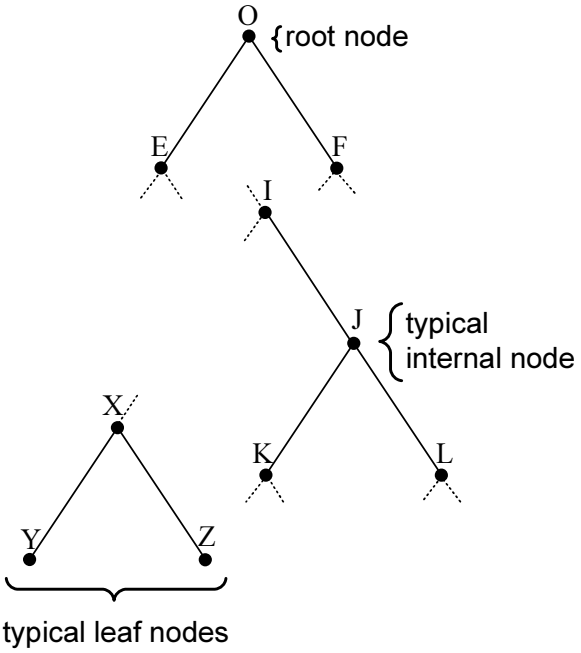


Figure 1: Example tree used throughout this note. O denotes the root node. Y and Z are typical leaf nodes. J is the typical internal node.

By *specific assignment*, we mean a tree that has every node labeled with a specific value, for example, A, C, G, or T for aligned nucleotide positions, or one of 20 amino acids for aligned codons. We use lowercase italic letters to denote the variable that holds the assignment: $b$ is the value at node B, etc. When the value at a leaf node is known from the data, we put a hat over it, so $\hat{y}$ means the specific value measured (for one aligned position) at node Y.

We refer to edges (also called branches) by pairs of roman letters, e.g., OE, OF, IJ, JK, JL, XY, YZ. We always label edges from their rootward to their

leafward nodes.

Transition matrices are labeled by their node, so, for example, $\mathbf{A}^{\mathrm{JK}}$ is the transition matrix from node J to node K. The components of this matrix are then written as $A^{\mathrm{JK}}_{jk}$. These components satisfy the condition for a stochastic (i.e., Markov) matrix,

$$\sum_k A^{\mathrm{JK}}_{jk} = 1 \tag{1}$$

that is, every row sums to unity.

Finally, we adopt a modified Einstein summation convention: Subscript indices are taken to be summed over if they are underlined. So, for example,

$$A^{\mathrm{IJ}}_{i\underline{j}} A^{\mathrm{JK}}_{\underline{j}k} A^{\mathrm{JL}}_{\underline{j}l} \equiv \sum_j A^{\mathrm{IJ}}_{ij} A^{\mathrm{JK}}_{jk} A^{\mathrm{JL}}_{jl} \tag{2}$$

and

$$A^{\mathrm{IJ}}_{i\underline{j}} A^{\mathrm{JK}}_{\underline{jk}} A^{\mathrm{KM}}_{\underline{k}m} \equiv \sum_{j,k} A^{\mathrm{IJ}}_{ij} A^{\mathrm{JK}}_{jk} A^{\mathrm{KM}}_{km} \tag{3}$$

The range of each sum is $0, \ldots, 3$ (for a data set of aligned nucleotides) or $0, \ldots, 19$ (for a data set of aligned amino acids).

Figure 2 shows the example tree, but now labeled by the value at each node, and the transition matrix on each edge.

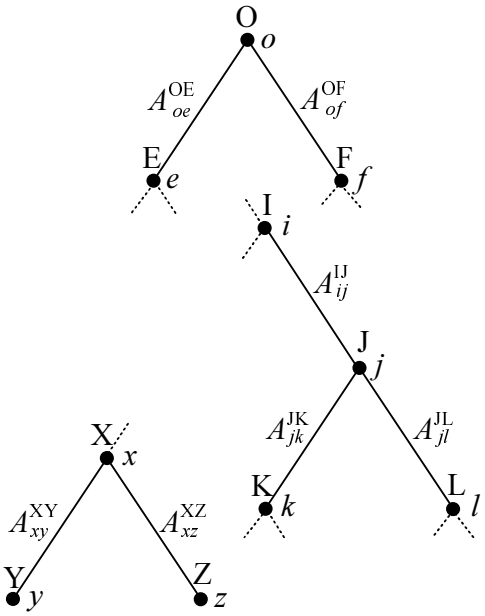


Figure 2: The example tree is here labeled by the assignment of values to nodes (lowercase Italic letters) and by the transition matrices.

# 3 The Bayesian Setup

Consider a single aligned position, and let $D$ be the observed leaf-node data, $D = \{\ldots, \hat{y}, \hat{z}, \ldots\}$. Let $A$ denote all the transition matrices, $A = \{\mathbf{A}^{\mathrm{OE}}, \mathbf{A}^{\mathrm{OF}}, \ldots, \mathbf{A}^{\mathrm{XZ}}, \ldots\}$. Let $T$ denote a particular labeling of the tree, that is, a vector of particular values $T = \{o, e, f, \ldots, i, j, k, l, \ldots, x, y, z, \ldots\}$.

Given $A$ and $D$, what is the probability of a particular labeling $T$? Bayes' rule gives,

$$p(T|D, A) = \frac{p(D|T, A)p(T|A)}{\sum_{T'} p(D|T', A)p(T'|A)} \tag{4}$$

Here $p(T|A)$ is the (prior) probability of the labeling $T$, before we see any data but given the complete model $A$. The sum over $T'$ is shorthand for a sum over $2M - 1$ individual indices, one for each node in the tree.

Now observe that $p(D|T, A)$, the probability of the observed data *given* a complete labeling of the tree, doesn't depend on $A$ and is in fact trivial: If the data $D$ is consistent with the labeling $T$ it is unity, because the nodes in $D$ are a subset of the nodes in $T$; otherwise it is zero:

$$p(D|T, A) = \Delta_{DT} \equiv \cdots \delta_{y\hat{y}}\delta_{z\hat{z}} \ldots \qquad \text{(product over leaf nodes)} \tag{5}$$

If there are missing data (e.g., $\hat{y}$ is not known), then the corresponding delta functions are omitted (e.g., $\delta_{y\hat{y}}$). In either case, we have,

$$p(T|D, A) = \frac{\Delta_{DT}\, p(T|A)}{\sum_{T'} \Delta_{DT'}\, p(T'|A)} \tag{6}$$

The denominator in equations (4) and (6) is important not just as a normalizing factor, but also (we are about to see) in its own right. It is usually called the "total likelihood of the data,"

$$\mathcal{L}(D|A) \equiv \sum_{T} \Delta_{DT}\, p(T|A) \tag{7}$$

Now we are ready to use the information from many different alignment positions, indexed by $\nu = 1, \ldots, N$, to estimate the transition matrices $A$. In other words, we have not just one $D$, but an $N$-vector $\mathbf{D}$ of data sets $D_\nu$, where $N$ is the number of aligned positions. Since the positions are assumed to be independent, the overall probability is a product,

$$\begin{aligned} p(A|\mathbf{D}) &= \prod_{\nu} P(A|D_\nu) \\ &\propto \prod_{\nu} P(D_\nu|A)\pi(A) \\ &\propto \pi(A) \prod_{\nu} \sum_{T'} p(D|T', A)p(T'|A) \\ &\propto \pi(A) \prod_{\nu} \mathcal{L}(D_\nu|A) \end{aligned} \tag{8}$$

Here, the second line uses Bayes' rule, while the third uses the law of total probability. Because we will only be interested in the *relative* probabilities of different models $A$, we can ignore the (implicit) denominators and use proportionality signs. The quantity $\pi(A)$ is the prior on the set of transition matrices. We will take it to be the non-informative (flat) prior,

$$\pi(A) = 1 \qquad \text{(flat prior)} \tag{9}$$

but it could be used to impose *a priori* conditions on $A$, such as the special forms about which we were so dismissive in the introduction, above. One could also use the prior $\pi(A)$ as a regularization parameter, for example favoring $A$'s that are closer to the identity matrix, a kind of parsimony.

When we assume equation (9), we might as well also redefine $p(A|\mathbf{D})$ so as to turn the final proportionality sign in equation (8) into an equality,

$$p(A|\mathbf{D}) = \prod_{\nu} \mathcal{L}(D_\nu|A) \tag{10}$$

We need only keep in mind that, hereafter, $p(A|\mathbf{D})$ is always a relative probability, with an unknown normalizing constant. In fact, since we never need integrate over all $A$'s, the issue never arises.

Up to this point everything is general for a setup that has $N$ copies of an "object", each with state labeled by $T$, of which a subset $D$ is measured. The copies share a common set of model parameters $A$.

# 4 Calculating the Various Probabilities

Going beyond the general setup, we now want to use properties that are specific to our "objects" being trees, as in Figure 2.

The fundamental thing we need to calculate is $p(T|A)$ (cf. equations 6 and 7). As remarked, this is a prior, because it doesn't depend on the data $D$. Since specifying $T$ is equivalent to specifying its nodes $\{o, e, f, \ldots, i, j, k, l, \ldots, x, y, z, \ldots\}$, $p(T|A)$ depends both on our priors for the node values, $\pi_o^{\mathrm{O}}, \pi_e^{\mathrm{E}}, \ldots, \pi_z^{\mathrm{Z}}, \ldots$, and also on the probabilities associated by $A$ to the implied transitions. In other words, a tree can be unlikely either because its node labels are themselves unlikely, or because the transitions between them have low probability. We thus have

$$p(T|A) = \pi_o^{\mathrm{O}} A_{oe}^{\mathrm{OE}} \pi_e^{\mathrm{E}} A_{of}^{\mathrm{OF}} \pi_f^{\mathrm{F}} \cdots A_{ij}^{\mathrm{IJ}} \pi_j^{\mathrm{J}} A_{jk}^{\mathrm{JK}} \pi_k^{\mathrm{K}} A_{jl}^{\mathrm{JL}} \pi_l^{\mathrm{L}} \cdots A_{xy}^{\mathrm{XY}} \pi_y^{\mathrm{Y}} A_{xz}^{\mathrm{XZ}} \pi_z^{\mathrm{Z}} \cdots \tag{11}$$

Note that there are *no sums* in this expression, and that there is exactly one factor for each edge and each node in the tree.

Equation (6) tells us that the relative probabilities of all trees, now *given* the data, have just the same form as equation (11), if we just restrict the indices to agree with the observed data (thus imposing the factor $\Delta_{DT}$). In practice, we

rarely have any informative prior information about non-observed node probabilities like $\pi_k^{\mathrm{K}}$, so we can set these to unity, proportional to the noninformative constant prior. However, we may well have useful priors on leaf notes, e.g., $\pi_y^{\mathrm{Y}}, \pi_z^{\mathrm{Z}}$, since we may know their distribution over many base pairs.

Making this simplification, we now renormalize the result with denominator $\mathcal{L}(D|A)$, because the sum of the (restricted) terms is no longer unity.

$$p(T|D,A) = \pi_o^{\mathrm{O}} A_{oe}^{\mathrm{OE}} A_{of}^{\mathrm{OF}} \cdots A_{ij}^{\mathrm{IJ}} A_{jk}^{\mathrm{JK}} A_{jl}^{\mathrm{JL}} \cdots A_{x\hat{y}}^{\mathrm{XY}} A_{x\hat{z}}^{\mathrm{XZ}} \pi_y^{\mathrm{Y}} \pi_z^{\mathrm{Z}} \cdots \Big/ \mathcal{L}(D|A) \quad (12)$$

The denominator $\mathcal{L}(D|A)$ is the sum of the numerators over all allowed $T$'s (equation 7). We do the sum, by summing over all internal nodes,

$$\mathcal{L}(D|A) = \pi_{\underline{o}}^{\mathrm{O}} A_{\underline{oe}}^{\mathrm{OE}} A_{\underline{of}}^{\mathrm{OF}} \cdots A_{\underline{ij}}^{\mathrm{IJ}} A_{\underline{jk}}^{\mathrm{JK}} A_{\underline{jl}}^{\mathrm{JL}} \cdots A_{\underline{x}\hat{y}}^{\mathrm{XY}} A_{\underline{x}\hat{z}}^{\mathrm{XZ}} \pi_{\hat{y}}^{\mathrm{Y}} \pi_{\hat{z}}^{\mathrm{Z}} \cdots \quad (13)$$

All non-leaf indices are summed over, as well as any leaf indices with missing data.

A practical issue is that the number of terms in the sum is exponentially large (4 or 20 the the power $M-1$, the number of internal nodes). Luckily, as discovered by Felsenstein[7], there is an efficient recursive way to do the sums, exactly analogous to the forward-backward algorithm for evaluating hidden Markov models in time (see [5], [6] for reviews of HMMs). Focus attention on a typical node, J, and make the sum over its index $j$ explicit. Also partition factors into three groups: those that connect to J only through its parent node I, and those that connect to J only through each of its daughter nodes K and L. Each such group evaluates, doing its internal sums, to 4-vector (or 20-vector), labeled by the single index $j$:

$$\begin{aligned} \mathcal{L}(D|A) &= \sum_j (\pi_{\underline{o}}^{\mathrm{O}} A_{\underline{oe}}^{\mathrm{OE}} A_{\underline{of}}^{\mathrm{OF}} \cdots A_{\underline{x}\hat{y}}^{\mathrm{XY}} A_{\underline{x}\hat{z}}^{\mathrm{XZ}} \cdots A_{\underline{ij}}^{\mathrm{IJ}}) \times (A_{j\underline{k}}^{\mathrm{JK}} \cdots) \times (A_{j\underline{l}}^{\mathrm{JL}} \cdots) \\ &\equiv \sum_j d_j^{\mathrm{IJ}} u_j^{\mathrm{JK}} u_j^{\mathrm{JL}} \end{aligned} \quad (14)$$

That is, we denote by $d_j^{\mathrm{IJ}}$ the "down" vector of all (fully summed over) factors that come down to node J through its parent I, and by $u_j^{\mathrm{JK}}$ and $u_j^{\mathrm{JL}}$, respectively, the "up" vectors that come through J's daughter nodes K and L. Every edge has both an up vector and a down vector.

An additional special case for a down vector is

$$d_o^{\mathrm{O}} \equiv \pi_0^{\mathrm{O}} \quad (15)$$

which is to be regarded as the down vector from the (non-existent) parent of node O. At node O we then have

$$\mathcal{L}(D|A) = \sum_o d_o^{\mathrm{O}} u_o^{\mathrm{OE}} u_o^{\mathrm{OF}} \quad (16)$$

entirely analogous to equation (14).

It should be immediately apparent that the up vectors have a simple recurrence, working from the bottom of the tree upwards (see Figure 3). The recurrence starts with leaf nodes, for which

$$u_x^{\mathrm{XY}} = A_{x\hat{y}}^{\mathrm{XY}} \pi_{\hat{y}}^{\mathrm{Y}} \tag{17}$$

when $\hat{y}$ is known (usual case) or

$$u_x^{\mathrm{XY}} = A_{x\underline{y}}^{\mathrm{XY}} \pi_{\underline{y}}^{\mathrm{Y}} \qquad \text{(note the sum)} \tag{18}$$

if $\hat{y}$ is missing data (because, in this case, there is no factor $\delta_{y\hat{y}}$ restricting the sum over all $T$). Now, for all internal nodes,

$$u_i^{\mathrm{IJ}} = A_{i\underline{j}}^{\mathrm{IJ}} u_{\underline{j}}^{\mathrm{JK}} u_{\underline{j}}^{\mathrm{JL}} \tag{19}$$

which is a single sum over 4 (or 20) terms. This is the analog of the "forward pass" in the HMM forward-backward algorithm.

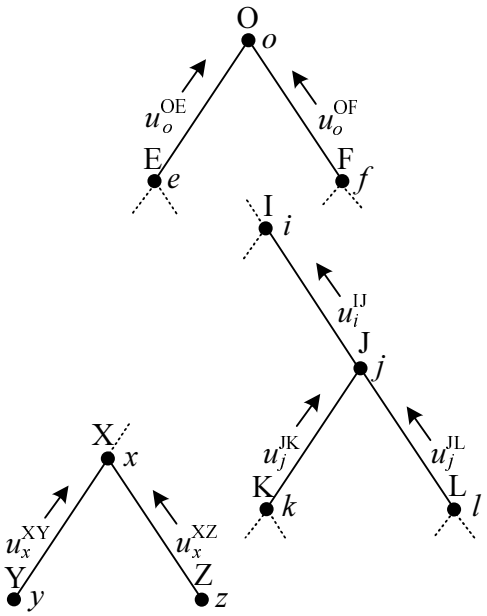


Figure 3: Recursive calculation of the "up vectors" $u$.

Recursing our way up the tree in this manner, and then evaluating equation (16), we obtain $\mathcal{L}(D|A)$. This is the original Felsenstein (1981) [7] "pruning method".

At this point, we could declare victory: Using equations (16) and (10), we can efficiently calculate $p(A|D)$ for any model $A$. If $A$ is expressed in a parameterized form, we could use standard techniques to maximize $A$ over those parameters, obtaining the maximum likelihood (ML) estimate. We would not,

however, be in a position to find the probability distribution of labels (nucleotides or amino acids) on internal nodes, nor to find the best reconstruction of the tree for any single position. To do that, we need not just a forward pass, but also a backward pass, as we now describe. Also, we will be able to use the powerful machinery of Bayesian re-estimation (or, nearly equivalently, the estimation-maximization (EM) method) instead of, or in addition to, the standard maximization techniques.

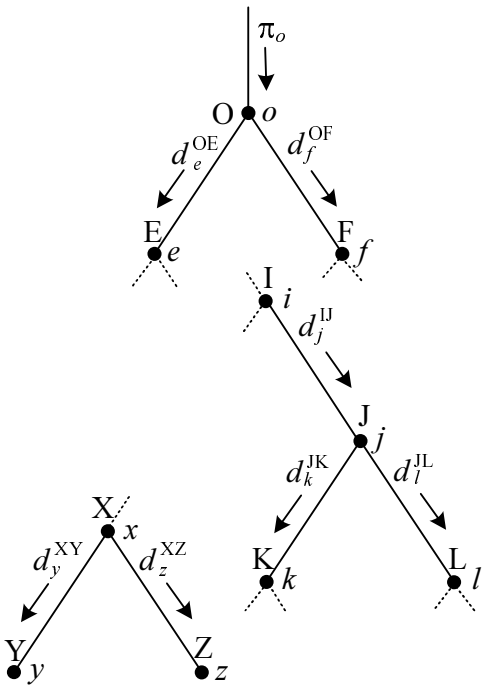


Figure 4: Recursive calculation of the "down vectors" $d$. Not shown is that the up vectors are also needed in the recurrence.

# 5 Downward Recurrence, Node and Node-Pair Probabilities

Starting at the root of the tree with equation (15), we can compute the down vectors on all tree edges. We couldn't have done this before computing all the up vectors, however, because the up vectors enter into the recurrence:

$$d_l^{\mathrm{JL}} = d_{\underline{j}}^{\mathrm{IJ}} u_{\underline{j}}^{\mathrm{JK}} A_{\underline{j}l}^{\mathrm{JL}} \tag{20}$$

a single sum over $j$. This is the analog of the HMM "backward pass".

What does this get us? Not very usefully, we can now compute the same, redundant value $\mathcal{L}(D|A)$ at every node, using equation (14). Much more useful is to compute the probability distribution of labels at a single internal node I. Since this is just a marginal over the original joint distribution of equation (12), we can immediately write (see Figure 5 for the node topology),

$$p(i|D,A) = d_i^{\mathrm{HI}} u_i^{\mathrm{IM}} u_i^{\mathrm{IJ}} / \mathcal{L}(D|A) \qquad \text{(No sum!)} \tag{21}$$

for an interior node, or

$$p(\hat{z}|D,A) = d_{\hat{z}}^{\mathrm{XZ}} \pi_{\hat{z}}^{\mathrm{Z}} / \mathcal{L}(D|A) = 1 \qquad \text{(No sum!)} \tag{22}$$

for a measured leaf node, or

$$p(z|D,A) = d_z^{\mathrm{XZ}} \pi_z^{\mathrm{Z}} / \mathcal{L}(D|A) \qquad \text{(No sum!)} \tag{23}$$

for a leaf node with missing data.

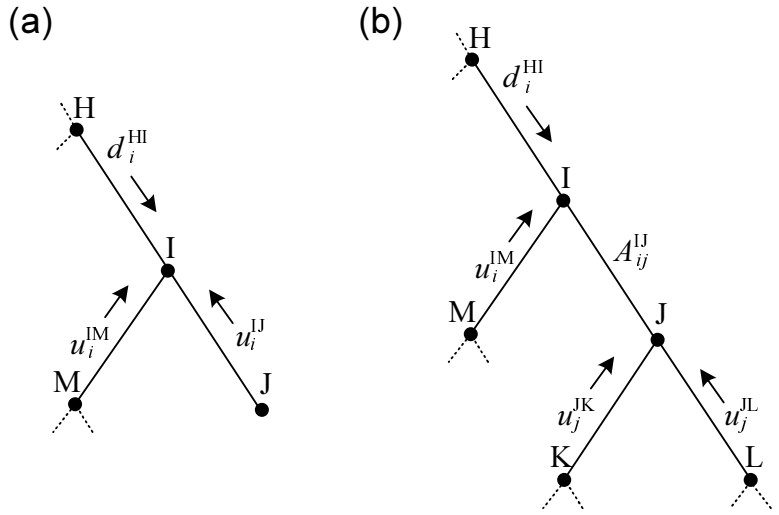


Figure 5: Calculation of (a) the probability distribution at an internal node I, or (b) the joint distribution at two nodes I and J.

In like manner, the joint distribution of two adjacent nodes I and J is the marginal that sums over all internal indices except $i$ and $j$,

$$p(i,j|D,A) = d_i^{\mathrm{HI}} u_i^{\mathrm{IM}} A_{ij}^{\mathrm{IJ}} u_j^{\mathrm{JK}} u_j^{\mathrm{JL}} / \mathcal{L}(D|A) \qquad \text{(No sums!)} \tag{24}$$

with special cases for leaf nodes analogous to those of equation (21).

Note that equations (21) and (24) are the distributions for a single tree, corresponding to a single aligned position. If we want the sample distributions at a single node $i$ or pair $i,j$ over the whole set of positions, we repeat the

calculations $N$ times, using the data for each position, and then average the individual probabilities,

$$p(i|\mathbf{D},A) = \frac{1}{N}\sum_{\nu=1}^{N} p(i|D_\nu,A), \qquad p(i,j|\mathbf{D},A) = \frac{1}{N}\sum_{\nu=1}^{N} p(i,j|D_\nu,A) \tag{25}$$

If we want the expected number of times that the value $i$ in the sample at a specific node (or the values $i,j$ at a specific node pair), then just omit the factor $1/N$,

$$E(\#[i]) = \sum_{\nu=1}^{N} p(i|D_\nu,A), \qquad E(\#[i,j]) = \sum_{\nu=1}^{N} p(i,j|D_\nu,A) \tag{26}$$

Had we declared victory, as was suggested, at the end of §4, there might still be some utility in doing the downward pass of this section. The reason is that the entire dependence of $\mathcal{L}(D|A)$ on $A_{ij}^{\mathrm{IJ}}$ can be isolated in the form

$$\mathcal{L}(D|A) = d_{\underline{i}}^{\mathrm{HI}} u_{\underline{i}}^{\mathrm{IM}} A_{\underline{ij}}^{\mathrm{IJ}} u_{\underline{j}}^{\mathrm{JK}} u_{\underline{j}}^{\mathrm{JL}} \tag{27}$$

As noted by Schadt et al. [8], this gives an easy way to compute the partial derivatives of $P(A|\mathbf{D})$ with respect to every component $A_{ij}^{\mathrm{IJ}}$, which is required by some (e.g., conjugate gradient and quasi-Newton) maximization methods. The routine `dfpmin` in [9] is a good example. On the other hand, there are other maximization methods that don't need derivatives, for example direction set methods such as `powell` in [9].

# 6 Bayesian Re-estimation or EM Iteration

The more powerful use of equation (26) is to re-estimate all the transition matrices, exactly analogous to Baum-Welch re-estimation for HMMs. Since, by definition $A_{ij}^{\mathrm{IJ}}$ is the number of times that we see an $i \to j$ transition per occurrence of $i$, we immediately have,

$$\widehat{A_{ij}^{\mathrm{IJ}}} = \frac{\sum_\nu p(i,j|D_\nu,A)}{\sum_\nu p(i|D_\nu,A)} \tag{28}$$

Just like Baum-Welch, equation (28) can be viewed as either an improved posterior estimate of $A_{ij}^{\mathrm{IJ}}$, or as one step in an EM (expectation-maximization) [10] iteration to maximize $P(A|\mathbf{D})$ over $A$. From either perspective, it can be proved that $P(A|\mathbf{D})$ (equation 10) is increased by the iteration, and that multiple steps will converge to a (local) maximum of $P(A|\mathbf{D})$. In the context of phylogenetic trees, the earliest use of this re-estimation in the general model seems to be Barry and Hartigan (1987) [11], although Felsenstein's original paper [7] gives EM re-estimation formulas for edge lengths alone, using a simple model for $A$.

While it is easy to accept equation (28) on plausibility grounds, let us actually prove that the re-estimation increases $P(A|\mathbf{D})$. (While the proof is analogous to both EM and Baum-Welch proofs, there are enough small differences to

make it worth writing down explicitly.) First, we write $P(A|\mathbf{D})$ as a sum over not just all possible labeling of a single tree, but all possible labelings of all $N$ trees:

$$\begin{aligned} P(A|\mathbf{D}) &= \prod_\nu \mathcal{L}(D_\nu|A) \qquad \text{(equation 10)} \\ &= \prod_\nu \sum_{T_\nu} p(T_\nu|D_\nu, A) \\ &= \sum_{T_*} \prod_\nu p(T_\nu|D_\nu, A) \\ &\equiv \sum_{T_*} p(T_*|\mathbf{D}, A) \end{aligned} \tag{29}$$

Here $T_\nu$ is shorthand for $M-1$ indices that correspond to internal nodes in tree $\nu$ (that is, the $\Delta_{DT}$ factor has already been applied), while $T_*$ is shorthand for the $N(M-1)$ indices that correspond to *all* internal nodes in *all* the trees. The probability $p(T_*|\mathbf{D}, A)$ is thus the probability of a particular full internal-node labeling of all $N$ trees.

Now define an auxiliary function $Q(A, \overline{A})$ by

$$Q(A, \overline{A}) \equiv \sum_{T_*} p(T_*|\mathbf{D}, A) \ \ln\left[p(T_*|\mathbf{D}, \overline{A})\right] \tag{30}$$

(The bar over the $A$ in $\overline{A}$ is not a summation convention. It's just a bar!) We show that an increase in $Q$ obtained by adjusting its second argument, $\overline{A}$, implies at least as large an increase in $p(A|\mathbf{D})$:

$$\begin{aligned} Q(A, \overline{A}) - Q(A, A) &= \sum_{T_*} p(T_*|\mathbf{D}, A) \ln\left[\frac{p(T_*|\mathbf{D}, \overline{A})}{p(T_*|\mathbf{D}, A)}\right] \\ &\le \sum_{T_*} p(T_*|\mathbf{D}, A) \left[\frac{p(T_*|\mathbf{D}, \overline{A})}{p(T_*|\mathbf{D}, A)} - 1\right] \\ &= P(\overline{A}|\mathbf{D}) - P(A|\mathbf{D}) \end{aligned} \tag{31}$$

The inequality $\ln(x) \le x - 1$, with equality holding only for $x = 1$, has been used.

The use of equation (31) that gives the biggest guaranteed increase in $P(A|\mathbf{D})$ is—if we can do it—to jump to the maximum over $\overline{A}$ of $Q(A, \overline{A})$. Since $\overline{A}$ is a stochastic matrix, we must impose its row-sum conditions, equation (1), by Lagrange multipliers, one for each row. (To simplify the notation, we write $A_{ij}$

for $A_{ij}^{\mathrm{IJ}}$.)

$$\begin{aligned}
0 &= \frac{\partial}{\partial \overline{A}_{ij}} \left[ Q(A, \overline{A}) + \lambda_{\underline{i}} \overline{A}_{\underline{ij}} \right] \\
&= \frac{\partial}{\partial \overline{A}_{ij}} \left[ \sum_{T_*} p(T_*|\mathbf{D}, A) \sum_{\nu} \ln p(T_\nu|D_\nu, \overline{A}) + \lambda_{\underline{i}} \overline{A}_{\underline{ij}} \right] \\
&= \left[ \sum_{\nu} \frac{\partial}{\partial \overline{A}_{ij}} \sum_{T_*} p(T_*|\mathbf{D}, A) \ln p(T_\nu|D_\nu, \overline{A}) \right] + \lambda_i \\
&= \left[ \sum_{\nu} \frac{\partial}{\partial \overline{A}_{ij}} \sum_{T_*} p(T_*|\mathbf{D}, A) (\cdots + \ln \overline{A}_{ij} + \cdots) \right] + \lambda_i
\end{aligned} \tag{32}$$

In the last line we have indicated that, if we write out $\ln p(T_\nu|D_\nu, \overline{A})$, there is exactly one additive term, as shown, that depends on the particular component $\overline{A}_{ij}$. Not so easy to indicate notationally is that, because the sum over $\nu$ has been moved out, that particular $i, j$ pair specializes the indices in $T_*$ that correspond to only one of the $N$ sets of $M - 1$ indices for which $T_*$ is shorthand. All the other $N - 1$ sets of indices can be moved rightward to sum their respective terms, giving

$$\begin{aligned}
0 &= \left\{ \sum_{\nu} \left[ \sum_{T_\nu \neq i,j} p(T_\nu|D_\nu, A) \right] \left[ \prod_{\mu \neq \nu} \mathcal{L}(D_\mu|A) \right] \frac{1}{\overline{A}_{ij}} \right\} + \lambda_i \\
&= \left\{ \frac{1}{\overline{A}_{ij}} \sum_{\nu} p(i, j|D_\nu, A) \right\} + \frac{\lambda_i}{p(A|\mathbf{D})}
\end{aligned} \tag{33}$$

where in the second line we have divided by $P(A|\mathbf{D})$ (using equation 10). We can now solve equation (33) for $A_{ij}$. When the $\lambda_i$'s are set to impose the conditions on row sums, the denominator in equation (28) automatically appears, and the result is exactly equation (28).

We should note in passing that *maximizing* $P(A|\mathbf{D})$ is not the only game in town: One may wish to explore the space of posterior estimates of $A$ by Markov Chain Monte Carlo (MCMC) methods applied to $P(A|\mathbf{D})$, yielding not just maximum values, but also uncertainties, Bayes-factor model comparisons, and so forth (see, e.g., [12], [13]). In such a case initial maximization by repeated re-estimation can be used to generate a starting point that minimizes subsequent burn-in time.

# 7 Branch Lengths

The most natural measures of branch length in the context of the general model used in this note are the logdet distance and the related paralinear distance (see, e.g., [14], [15]). We can define these in terms of posterior (or ML) estimates of

$\mathbf{A}^{\mathrm{IJ}}$ and of the posterior distribution of states at nodes I and J, denoted $f_i^{\mathrm{I}}$ and $f_j^{\mathrm{J}}$ by

$$d_{LD}^{\mathrm{IJ}} \equiv -\tfrac{1}{4} \ln \det(\mathbf{A}^{\mathrm{IJ}}) \tag{34}$$

and

$$d_{PL}^{\mathrm{IJ}} \equiv -\tfrac{1}{4} \ln \det \left[ \mathrm{diag}(f^{\mathrm{I}})^{1/2} \, \mathbf{A}^{\mathrm{IJ}} \, \mathrm{diag}(f^{\mathrm{J}})^{-1/2} \right] \tag{35}$$

(Our definition of $d_{LD}$ is slightly nonstandard, but our definition of $d_{PL}$ is equivalent to the standard definition in terms of the data matrix $\mathbf{J}^{\mathrm{IJ}} = \mathrm{diag}(f^{\mathrm{I}})\mathbf{A}^{\mathrm{IJ}}$.) Evidently these two distances are related by

$$d_{PL}^{\mathrm{IJ}} = d_{LD}^{\mathrm{IJ}} - \tfrac{1}{8} \sum_k \ln \frac{f_k^{\mathrm{I}}}{f_k^{\mathrm{J}}} \tag{36}$$

Both distance measures are positive. Paralinear distance is additive over the tree (even over a path that goes up, then down), and is always directly related to the data matrix between the two ends of the path. The data matrix over a path is defined as the joint probability of seeing nucleotide $i$ at one end of the path and $j$ at the other end, its 16 components adding up to unity.

We now assume, for the first time, that the matrix $\mathbf{A}^{\mathrm{IJ}}$ can be generated by an infinitesimal generator $\mathbf{G}^{\mathrm{IJ}}$, that is,

$$\mathbf{A}^{\mathrm{IJ}} = \exp(\mu \mathbf{G}^{\mathrm{IJ}}) \tag{37}$$

with $\mathbf{G}$ having zero row sums, zero or negative diagonal and zero or positive off-diagonal elements. Since $\mathbf{G}$ can absorb any constant factor from $\mu$, it needs a normalization convention. A convenient one is

$$\mathrm{tr}(\mathbf{G}) = -4 \tag{38}$$

Then $\mu$ will be the evolutionary distance measured in mean changes per site from a hypothetical uniform nucleotide distribution. Characterizing the conditions under which such a generator exists is known as the "embedding problem," and goes back to 1937. A sufficient (and usual) case is that $\mathbf{A}$ is diagonalizable and has positive eigenvalues. See [16] for a review.

Since for any matrix

$$\ln[\det(\mathbf{A})] = \mathrm{tr}[\ln(\mathbf{A})] \tag{39}$$

(if the quantities exist), we have

$$d_{LD} = \mu = -\tfrac{1}{4} \ln \det(\mathbf{A}) \tag{40}$$

and

$$\mathbf{G} = \frac{1}{\mu} \ln(\mathbf{A}) \tag{41}$$

So individual branch lengths and individual generators are immediately available from the posterior estimate of $\mathbf{A}^{\mathrm{IJ}}$, if the generator exists.

If we want branch lengths conditioned on the assumption that the generators are identical on some or all branches, then we can iterate as follows:

1. Re-estimate $\mathbf{A}^{\mathrm{IJ}}$ on all branches.
2. From each $\mathbf{A}^{\mathrm{IJ}}$, compute $\mu^{\mathrm{IJ}}$ and $\mathbf{G}^{\mathrm{IJ}}$, using equations (40) and (41).
3. Average the $\mathbf{G}^{\mathrm{IJ}}$'s obtained for different branches to get a consensus generator $\mathbf{G}$.
4. Replace each $\mathbf{A}^{\mathrm{IJ}}$ by $\exp(\mu^{\mathrm{IJ}}\mathbf{G})$.
5. Repeat steps 1–4 to convergence.
6. Compute the branch length $d_{PL}^{\mathrm{IJ}}$ by equation (36).

# References


[1] Felsenstein, J. (2004) *Inferring Phylogenies* (Sunderland, MA: Sinauer Associates).

[2] Yang, Z. (1994) "Maximum Likelihood Phylogenetic Estimation from DNA Sequences with Variable Rates over Sites: Approximate Methods," J. Mol. Evol., 39, 306–314.

[3] Stefankovic, D., and Vigoda, E. (2006) "Pitfalls of Varying Substitution Rates for Phylogenetic Reconstruction," preprint at http://www-static.cc.gatech.edu/~vigoda/StefVig.pdf

[4] Mossel, E., and Vigoda, E. (2005) "Phylogenetic MCMC Algorithms Are Misleading on Mixtures of Trees," Science, 309, 2207-2209.

[5] Poritz, A.B. (1988) "Hidden Markov Models: A Guided Tour," in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing [ICASSP-88] (New York: IEEE Press), vol. 1, pp. 7–13. Online at http://ieeexplore.ieee.org/ xpls/abs_all.jsp?arnumber=196495

[6] Rabiner, L.R. (1989) "A Tutorial on Hidden Markov Models and Selected Applications to Speech Recognition," Proc. IEEE, 77, 257–286.

[7] Felsenstein, J. (1981) "Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach," J. Mol. Evol., 17, 368–376.

[8] Schadt, E.E., Sinsheimer, J.S., and Lange, K. (1998) "Computational Advances in Maximum Likelihood Methods for Molecular Phylogeny," Genome Research, 8, 222-233.

[9] Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (2002) *Numerical Recipes in C++*, 2nd ed. (New York: Cambridge University Press).

[10] Dempster, A., Laird, N., and Rubin, D. (1977) "Maximum likelihood from incomplete data via the EM algorithm," J. Roy. Stat. Soc., Series B, 39, 1–38.

[11] Barry, D., and Hartigan, J.A. (1987) "Statistical Analysis of Hominoid Molecular Evolution," Statistical Sci., 2, 191–210.

[12] Tierney, L. (1994) "Markov Chains for Exploring Posterior Distributions," Ann. Stat., 22, 1701–1762.

[13] Suchard, M.A., Weiss, R.E., and Sinsheimer, J.S. (2001) "Bayesian Selection of Continuous-Time Markov Chain Evolutionary Models," Mol. Biol. Evol., 18, 1001–1013.

[14] Lake, J.A. (1994) "Reconstructing Evolutionary Trees from DNA and Protein Sequences: Paralinear Distances," PNAS, 91, 1455–1459.

[15] Gu, X., and Li, W-H (1996) "Bias-Corrected Paralinear and LogDet Distances and Tests of Molecular Clocks and Phylogenies under Nonstationary Nucleotide Frequences," Mol. Biol. Evol., 13, 1375–1383.

[16] Israel, R.B., Rosenthal, J.S., and Wei, J.Z. (2001) "Finding Generators for Markov Chains via Empirical Transition Matrices, with Applications to Credit Ratings," Mathematical Finance, 11, 245–265.