1 Reminder

In past lectures we talked about evolutionary trees. We modeled an evolution tree as a rooted binary tree $T$, with tree leaves $x_1, ..., x_n$, internal nodes $x_{n+1}, ..., x_{2(n-1)}$ and tree root $x_{2n-1} = x_r$. The lengths of tree edges are $\tau = \{t_{ij}\}_{(i,j) \in T}$.

In our model the tree has the form of a hidden Markov model - each node, is independent of all other nodes, given it’s parent.

In the previous lecture we presented the ‘Up’ algorithm - which is an adaptation of the forward-backward algorithm for trees. We used the algorithm to calculate the likelihood of our tree leaves, which are the model observations.

In this lecture we will use the same algorithm and show how we can answer different questions regarding the model, calculating posterior probabilities of different tree nodes.

2 Posterior Probabilities Calculation

2.1 Posterior Probability of the Root

The first case we address is calculating the posterior of the tree root.

Recall that for the Up algorithm we calculated for each node $i$ and value $a$ (in our model values are single nucleotides), $U_i[a] = P(x_L | x_i = a)$ where $L_i$ are the leaves of the tree which are descendants of node $i$. We used these calculations to find the likelihood of the leaves - $P(x_1, ..., x_n) = P(x_L) = \sum_a U_r[a] \cdot \pi_a$ where $\pi_a$ is the probability of $a$ in the stationary distribution.

Applying Bayes’ theorem we get an immediate result regarding the root posterior -

$$P(x_r = a | x_L) = \frac{P(x_L | x_r = a) \cdot P(x_r = a)}{P(x_L)} = \frac{U_r[a] \cdot \pi_a}{\sum_b U_r[b] \cdot \pi_b} \quad (1)$$

2.2 Posterior Probability of Internal Nodes

An interesting question to ask is the posterior of a value (for a nucleotide) at any node. To answer this question, we need to generalize the previous formula.

As a first attempt, we look at the sub-tree of $T$ rooted from $i$ and replace the root with node $i$ - $P(x_i = a | x_{L_i}) = \frac{U_i[a] \cdot \pi_a}{\sum U_i[b] \cdot \pi_b}$. This formula doesn’t take into consideration all of our observations, which might change our estimation of the node. This intuition is demonstrated in figure 1.

In order to formulate the intuition from figure 1 and find the correct posterior of $i$ we will look at the original tree, rooted from node $i$. Let node $l$ be $i$’s father, then to get the tree $T$ rooted from node $i$, all we need to do is hang the tree from $i$, making $l$ another child of $i$ and leaving the rest of the nodes the same as they were. We will get a tree that is almost binary and is very similar to the original tree. This tree change is demonstrated in figure 2.
We say that a model is reversible if for all \( a, b \) \( a \rightarrow b \) and \( b \rightarrow a \) are reversible. This means that the probability of each node. We will present the reversibility property, and assume that the evolution trees in our model are reversible.

Reversibility

We say that a model is reversible if for all \( a, b \in \Sigma \) and all \( t_1, t_2 \), it holds that

\[
\pi_a \cdot P(a \rightarrow t_1 + t_2 b) = \pi_b \cdot P(b \rightarrow t_1 + t_2 a)
\]

Where \( P(a \rightarrow t_1 + t_2 b) \) is the probability of a nucleotide \( a \) to transform to nucleotide \( b \) in time \( t_1 + t_2 \).

For two nodes \( a, b \) in a tree \( T \) rooted from node \( c \) (such that \( (c, a), (c, b) \in T \)) let \( t_1 = t_{ca}, t_2 = t_{cb} \), then the length of the path between \( a \) and \( b \) is exactly \( t_1 + t_2 \). Intuitively, reversibility means that we can move \( c \) along \([0, t_1 + t_2] \) - changing the lengths of \( t_1, t_2 \) to any \( t_1', t_2' \) such that \( 0 \leq t_1', t_2' \leq t_1 + t_2 \) and \( t_1' + t_2' = t_1 + t_2 \) without changing the probabilities of the nodes in the tree. In particular, if the tree is reversible then for two adjacent nodes \( i, l \) such that \((l, i) \in T \) we can rotate the tree around the node \( i \), make it the ancestor of \( l \) without any influence on the different probabilities of the nodes.

How can we know whether our tree is reversible? Recall that the matrix of transformation probabilities during time \( t \) is given by \( P(t) = e^{tR} \). Then for any short period of time \( \varepsilon \) we have that \( P(\varepsilon)I + \varepsilon \cdot R \), and for any two different nucleotides \( a, b \) we have that \( P(\varepsilon)_{a,b} = [\varepsilon R]_{a,b} \). If the reversibility property holds, then \( \pi_a \cdot \varepsilon \cdot R_{a,b} = \pi_b \cdot \varepsilon \cdot R_{b,a} \) and \( \frac{\pi_a}{\pi_b} = \frac{R_{a,b}}{R_{b,a}} \). This is also a sufficient condition for a matrix to be reversible, although the proof is left out of the scope of our course.

From now on we will consider only reversible transformation matrices.

Back to Our Model

Now that we know that we can rotate the original tree and treat each node as a root we can write a formula that is similar to the one we saw in the Up algorithm. We remember from the previous lesson that the probability of the leaves given the value of \( x_i \) is:

\[
P(x_L|x_i = a) = \prod_{w:v is a son of i} U_w[a] = U_{ij}[a] \cdot U_{ik}[a] \cdot U_{il}[a]
\]

where

\[
U_{ij}[a] = \sum_b P(x_j = b, x_L|x_i = a) = \sum_b P(x_L|x_j = b) \cdot P(x_j = b|x_i = a)
\]

Just as before, we can use this probability together with Bayes’ theorem and finally write the formula for the posterior probability of \( x_i \):

\[
P(x_i = a|x_L) = \frac{P(x_L|x_i = a) \cdot P(x_i = a)}{P(x_L)} = \prod_{j:j \text{ is a son of } i} U_{ij}[a] \cdot \pi_a
\]

(2)
Running Time

In previous lessons we saw that using DFS’ post-order, we can fill the table $U_i[a]$ in linear time. This formula uses the same table as before, only now we will look at the tree rooted from each internal node. This means that we use both directions of each edge - we need to calculate $U_{ij}[a]$ (as before), but also $U_{ji}[a]$. We can use the same technique as before - starting DFS twice from arbitrary node $i$ will produce the order in which we need to fill the table. From the first DFS we take the back edges (as before, this order guarantees that when filling a cell we will have all cells in it’s calculation already filled). From the second DFS we take the forward edges - which are exactly all tree edges, in the direction we haven’t calculated yet.

The table has $2 \cdot (2n - 2)$ rows, where $2n - 2$ is the number of edges in a tree with $2n - 1$ nodes, and $|\Sigma|$ columns. To fill each cell, we need to make $O(1)$ calculations using the cells of the node’s adjacents which were already filled (and there are at most three adjacents).

The total running time is $O(n)$ for calculating the order in which we fill the table, and $(2n - 2) \cdot |\Sigma| \cdot O(1) = O(n)$ time to fill the table - $O(n) + O(n) \cdot |\Sigma| = O(n)$ since $|\Sigma|$ is constant and much smaller than $n$.

2.3 A-Posterior Probability of Transition

We can now use the table $U_{ij}$ together with Bayes’ rule for other calculations - for two adjacent nodes $i, j$ we can calculate $P(x_i = a, x_j = b | x_L)$. This way we get the probability of any transition between two adjacent nodes. Let $k_1, k_2$ be the children of $i$, and $l_1, l_2$ be the children of $j$. We demonstrate the neighborhood of $i, j$ in $T$ in figure 3. We can calculate the probability of transition from nucleotide $a$ in node $i$ to nucleotide $b$ in node $j$ by:

$$P(x_i = a, x_j = b | x_L) = \frac{P(x_L | x_i = a, x_j = b) \cdot P(x_i = a) \cdot P(x_j = b | x_i = a)}{P(x_L)}$$

$$= \frac{U_{ik_1}[a] \cdot U_{ik_2}[a] \cdot U_{jl_1}[b] \cdot U_{jl_2}[b] \cdot \pi_a \cdot P(a \rightarrow i, j)}{P(x_L)}$$

(3)

Note

By calculating the posterior for all possible values for $i, j$ we can get the probability that the two nodes have identical values, and reestimate the nodes’ distance using Jukes-Cantor model.
Figure 3: \(i,j\) are adjacent nodes, and each one of them has at most two neighbors (other than each other). We can separate the whole tree to three parts - the edge \((i,j)\), the part of the tree on \(i\)'s side and the part of the tree on \(j\)'s side. The only places in common to these parts are the nodes \(i,j\) and thanks to the Markovian assumption - if we fix those, we can calculate the probability of each part independently as done in equation (3).