Ideas on sampling branch lengths given a tree topology.

The requirement is to find a method that is reasonably fast to generate at random (with a probability density that can be computed) branch lengths for a tree topology that are not too different from an actual posterior distribution for the purposes of implementing an importance sampling method for Bayesian phylogenetics. The approach outlined here requires a method to generate a random distance between two "sequences", where "sequences" is quoted because a sequence will be generalized to a site-wise independent probability distribution. The approach will be to work on subtrees with three leaves at a time, two nodes from a cherry and a neighboring node, generate the three pairwise distances, and then use the neighbor-joining approach to set the branch lengths to the two nodes of the cherry. The cherry is pruned from the tree and the process continues until there are only three nodes remaining. The final three edges are set in a final neighbor-joining step.

**Algorithm to generate branch lengths on an unrooted tree.**— If there are at least 4 taxa in the unrooted tree:

1. Make list of all cherries; (there must be at least two)
2. Pick one cherry at random; label its leaves $x$ and $y$ and let $z$ be the node adjacent to $x$ and $y$.
3. Pick a third leaf by taking a random path beginning at $z$ moving to its single adjacency that is neither $x$ nor $y$, and then continuing the path by selecting adjacent nodes at random with no back-tracking until hitting a leaf. Label this leaf $w$.
4. Generate at random pairwise distances for the three pairs; $d_{xy}$, $d_{xw}$, and $d_{yw}$.
5. Set distances
$$d_{xz} = \frac{d_{xy} + d_{xw} - d_{yw}}{2}, \qquad d_{yz} = \frac{d_{xy} + d_{yw} - d_{xw}}{2}$$
for the corrsponding tree edges. (Fudge this somehow if $d_{xz}$ or $d_{yz}$ turn out to be negative.)
6. Determine a sequence distribution for node $z$ conditional on $d_{xz}$, $d_{yz}$ and the sequences (or distributions) at $x$ and $y$.
7. Prune $x$ and $y$ from the tree and remove the pair from the list of cherries.
8. If $z$ is part of a new cherry, add it to the list of cherries.
9. If there are still at least 4 taxa, go back to step 2.
10. Else, there are only 3 taxa left: These become nodes $x$, $y$, and $w$. Go to step 4. However, also compute $d_{wz} = (d_{xw} + d_{yw} - d_{xy})/2$ and set it to the corresponding tree edge. This completes the generation of edge lengths on the tree.

**Likelihood.**— Let two nodes $x$ and $y$ be connected by an edge $e$ be the roots of subtrees extending away from $e$ with leaf data in the subtrees being $A$ and $B$ and conditional distributions of the sequences at $x$ and $y$ being $a$ and $b$, respectively.

For continuous-time, time-reversible infinitesimal generator $Q$ with stationary distribution $\pi$ and the associated transition matrix $P(t) = \mathrm{e}^{Qt}$ for the distance $t$, the likelihood of $t$ is

$$L(t) = \prod_k \Big( \sum_i \sum_j \pi_i \mathsf{P}\left(A_k \mid i\right) P_{ij}(t) \mathsf{P}\left(B_k \mid j\right) \Big)$$

where $i$ and $j$ vary over the four possible bases and $A_k$ and $B_k$ are the respective subtree data at site $k$. I need to show that this likelihood is proportional to

$$\tilde{L}(t) = \prod_k \Big( \sum_i \sum_j \pi_i a_{ik} P_{ij}(t) b_{jk} \Big)$$

where $a_{ik}$ is the conditional probability that node $x$ has base $i$ given the data in subtree $A$ and $b_{jk}$ is similarly defined. If $t$ has prior density $p(t)$, then the posterior density of $t$ given the sequences at the two nodes, $p(t \mid A, B)$, is proportional to $L(t)p(t)$. The approach here is to form an approximation of $p(t \mid A, B)$ by assuming that $P(t)$ is close enough to the Jukes-Cantor distribution.

**Distance generation.**— Let $x$ be the number of sites that differ among $n$ sites. For distributions $a$ and $b$, compute

$$x = n - \sum_i \sum_j a_{ij} b_{ij}.$$

I'm assuming that $x < 0.75n$ or the MLE for the Jukes-Cantor distance is infinite. The Jukes-Cantor distance is

$$d = -\Big(\frac{3}{4}\Big) \ln \Big(1 - \frac{4x}{3n}\Big)$$

Here, my thought is to generate $W \sim \mathrm{Beta}(\eta 4x/3, \eta(n - 4x/3)$ where $\eta$ is a tuning parameter close to one: a slightly smaller value, such as 0.9(?), will make the tails a bit wider than the target density which can be helpful for more efficient importance sampling.

Here is one example.

```
> require(ggplot2)
> like = function(t,x,n) {
+     theta = (1 - exp(-4*t/3))/4
+     log.like = n*log(0.25) + x*log(theta) + (n-x)*log(1-3*theta)
+     log.like = log.like - max(log.like)
+     f = exp(log.like)
+     return( f / sum(f))
+ }
> u = seq(0.001,0.5,0.001)
> y = like(u,100,500)/0.001
> df1 = data.frame(u,y)
```

```
> eta = 0.6
> w = rbeta(10000,eta*500*4*100/3/500,eta*500*(1 - 4*100/3/500))
> ww = -0.75 * log(1-w)
> d = density(ww)
> df2 = data.frame(x=d$x,y=d$y)
> p1 = ggplot(df1,aes(x=u,y=y)) +
+     geom_line() +
+     geom_line(aes(x=x,y=y),data=df2,color="red",linetype="dashed")
> plot(p1)
```