Creating Phylogenetic Trees with the Metropolis Algorithm

David Larson

October 31, 2001

1 Introduction

Phylogenetic trees attempt to describe evolutionary history of DNA, RNA, or protein sequences. This paper assumes that several sequences have been found and properly aligned. The metropolis algorithm then provides an intuitive but computationally slow method of creating a phylogenetic tree.

Most of the information in this paper is taken from chapter 8 of *Biological Sequence Analysis* by Durbin et. al.[1]. This paper simplifies that chapter.

2 The Metropolis Algorithm

The metropolis algorithm is a method of optimizing a system. It minimizes some energy function E of the system, as professor Chrisopher Lobb at UMCP explained to me. This algorithm has two requirements:

1. There must be a function E that describes the energy of the system.
2. There must be some method of (randomly) perturbing the system.

The algorithm consists of repeatedly perturbing the system. After each perturbation, the algorithm determines whether or not to keep the perturbation. If the energy has decreased, the perturbation is kept. If the energy has increased, the perturbation is kept with a probability that decreases with increasing energy change. Lobb gave the probability as $e^{-\Delta E/\tau}$ where τ is a constant (analogous to the temperature of the system).

The algorithm will randomly search the system for states of low energy, and it will spend more time in those states. If it runs for a while, it should reach a state of low energy.

In the case of a phylogenetic tree, the system is the tree itself. This is a binary tree: the root is the "earliest" point on the tree and the leaves, which represent the sequences, are the "most recent" points on the tree. The tree has a topology T that describes its branches, and a set of lengths $t_\bullet = \{t_i\}$ that describe the lengths of the branches. Length represents evolutionary time.

3 Finding the Probability of a Tree

The metropolis algorithm is used here to find the tree with the highest probability of being correct, given the original aligned sequences, $x^\bullet = \{x^i\}$. This means the energy of the tree must be something like the negative of the probability that the tree could yield those sequences, or $P(T, t_\bullet | x^\bullet)$.

To find this probability, take the simplest model of sequence evolution: characters of the sequence change randomly at a rate of α characters per unit time. For more realistic and detailed evolutionary models, see Durbin et. al. [1].

This assumption allows for the explicit calculation of the probability $P(x^i, x^j, t)$ that a sequence x^i will mutate into a sequence x^j in a given amount of time t.

Now if one arbitrarily picks a set of sequences $y^\bullet = \{y^j\}$ and puts them at the branch points (interior nodes) of the tree, then one can find a probability $P(x^\bullet | T, t_\bullet, y^\bullet)$ for all the mutations described by segments of the tree between two nodes. The probability $P(x^\bullet | T, t_\bullet)$ of the original sequences given the tree is given by the sum over *all possible* sets of interior node sequences.

$$P(x^\bullet | T, t_\bullet) = \sum_{y^\bullet} P(x^\bullet | T, t_\bullet, y^\bullet)$$

Felsenstein's algorithm is a recursive procedure for calculating this [1].

Now Bayes' rule [1] states that:

$$P(T, t_\bullet | x^\bullet) = \frac{P(x^\bullet | T, t_\bullet) P(T, t_\bullet)}{P(x^\bullet)}$$

If we assume $P(T, t_\bullet)$ and $P(x^\bullet)$ are simply known constants due to flat distributions of trees and sequences, then $P(T, t_\bullet | x^\bullet)$ can finally be calculated. This is the probability that the tree is correct, given the original sequences. One can calculate the energy of the tree by

$$E(T, t_\bullet, x^\bullet) = -P(T, t_\bullet | x^\bullet)$$

The metropolis algorithm discussed by Durbin et. al. uses a slightly different criterion for keeping a perturbation than that given by Lobb. If the perturbation is unfavorable, and $E_{final} > E_{initial}$, Durbin keeps it with probability $E_{initial}/E_{final}$, instead of Lobb's $e^{(E_{initial}/\tau)}/e^{(E_{final}/\tau)}$.

4 Perturbing the Tree

Perturbing the system changes the structure of the tree. For the metropolis algorithm to work well, the changes should only change part of the tree at a time. This corresponds to smaller steps through the configuration space, making it easier to settle in low energy wells.

Durbin et. al. suggest two types of changes to make to the tree topology T: profile changes, and branch switching [1]. See his figures included at the end of this paper. Profile changes modify the topology of the tree, while branch switching does not. Branch switching

changes the order of the leaves, to allow leaves that are originally far apart to become neighbors.

The lengths $t_\bullet$ must also be perturbed. Presumably, they only change slightly in a single perturbation, to make for smaller steps in configuration space.

5 Conclusion

This algorithm literally creates the most likely phylogenetic tree. It maximizes the probability (over all binary trees) that evolution occurred in the manner shown by the tree, given the sequences available $x^\bullet$.

The main problem is the excessive amount of time required for the algorithm. Even calculating the energy E of one tree cannot even be done in polynomial time. The number of calculations increases exponentially with the number of sequences x^i. The algorithm also needs many perturbations to be sure that it has reached an energy minimum. This metropolis algorithm does not show much promise for becoming a fast, efficient method of creating phylogenetic trees.

References

[1] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. *Cambridge.*

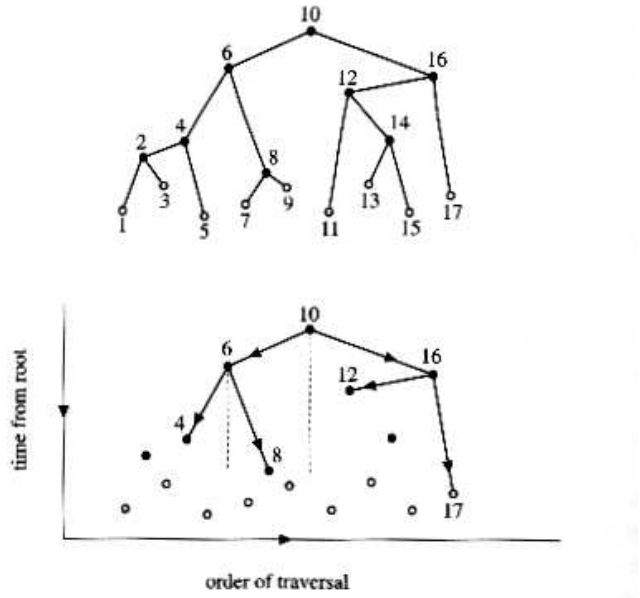

Figure 8.7 *Above: an example of a tree with its nodes numbered in the order of the traversal profile. Below: Reconstruction of the tree from the traversal profile.*

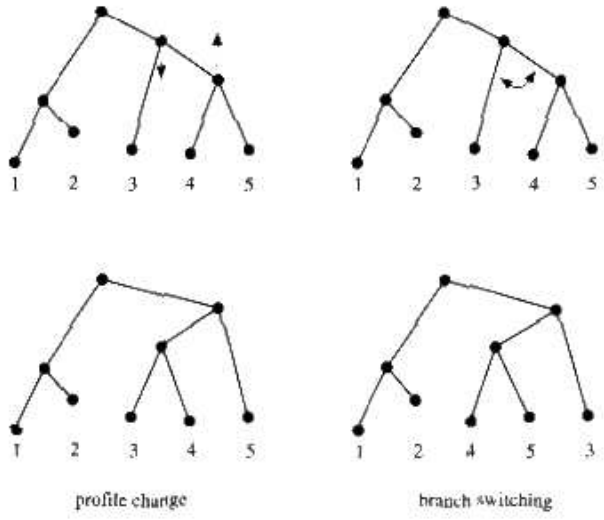

Figure 8.8 *The two parts of the proposal mechanism are changes in the height of the nodes in the profile (left), and reordering of the leaves by switching branches (right). The former can produce changes in the topology, as shown here. The latter does not do this; it just rearranges the existing topology. However, the change of order of the leaves allows new topologies to be reached through further steps of the first type.*