

## MIT Open Access Articles

*Random Tree Optimization for the Construction  
of the Most Parsimonious Phylogenetic Trees*

The MIT Faculty has made this article openly available. **Please share**  
how this access benefits you. Your story matters.

**Citation:** Fulu Li, and A. Lippman. "Random tree optimization for the construction of the most parsimonious phylogenetic trees." Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on. 2009. 757-762. ©2009 Institute of Electrical and Electronics Engineers.

**As Published:** <http://dx.doi.org/10.1109/CISS.2009.5054819>

**Publisher:** Institute of Electrical and Electronics Engineers

**Persistent URL:** <http://hdl.handle.net/1721.1/59834>

**Version:** Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

**Terms of Use:** Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



# Random Tree Optimization for the Construction of the Most Parsimonious Phylogenetic Trees

Fulu Li and Andrew Lippman

The Media Laboratory  
Massachusetts Institute of Technology  
Cambridge, MA USA 02139  
Email: [fulu@mit.edu](mailto:fulu@mit.edu)

## Abstract

With the availability of ever-increasing gene sequence data across a large number of species, reconstruction of phylogenetic trees to reveal the evolution relationship among those species becomes more and more important. In this paper, we focus on the construction of the most parsimonious phylogenetic trees given sequence data of a group of species as parsimony is probably the most widely used among all tree building algorithms [4]. The major contribution of this paper is the presentation of a novel algorithm, the random tree optimization (RTO) algorithm based on cross-entropy method [16], for the construction of the most parsimonious phylogenetic trees. We analyze the RTO algorithm in the framework of expectation maximization (EM) and point out the similarities and differences between traditional EM algorithm and the RTO algorithm.

## 1. Introduction

Reconstruction of phylogenetic trees is one of the most fundamental problems in computational biology. A phylogenetic tree reveals the evolutionary relationship among a given set of species. The basic hypothesis is that all organisms on Earth are evolutionary related via a common ancestor. Notably, genes can be diverged by either gene duplication (also known as paralog) or speciation (also known as ortholog). For the construction of phylogenetic trees of species it must be based on orthologs. We also need to emphasize that even the phylogenetic tree that best explains the sequence data of a group of species does not necessarily represent the true phylogenetic tree of the host species due to the processes of gene duplication, loss and lineage sorting [11].

In general, phylogenetic tree construction methods can be classified into four categories: distance-based methods, maximum parsimony methods, maximum likelihood schemes and maximum compatibility methods. Distance-based methods include UPGMA (Unweighted Pair Group Method Using Arithmetic Averages) and neighbor joining algorithms (see [4] for

algorithm details). For distance-based approach, the interspecies distances are defined *a priori*. The problem is to search for a tree which is most consistent with these distances. A typical way of tackling this problem is to minimize the sum of the squares of the difference between the pre-defined pairwise distance and the corresponding predicted pairwise distance from a given tree, e.g., the Least Squares Methods (LSM). It is known that using LSM to find the best tree is NP-complete. For maximum parsimony (MP) method [22], the phylogenetic tree construction performs a site-by-site (each site is normally represented by a character) analysis for the given sequence data. For each tree topology, it calculates the minimum number of mutation events that are required to explain the given sequence data. The total number of character changes are summed over all paths in a given tree topology. The problem is to search for a tree topology that has the minimum number of substitutions, which is considered as the most parsimonious tree to explain the given sequence data.

Existing maximum parsimony methods include Branch-and-Bound, Philip [13], Subtree Pruning and Regrafting (SPR), Nearest Neighbor Interchange (NNI), Tree Bisection and Reconnection (TBR), etc. For maximum likelihood method, it defines the likelihood of a tree as the probability of the observed data conditioned on the given tree and the probabilistic model. The problem is to maximize the likelihood of the tree among all potential tree topologies and branch lengths for the given observed data. Existing ML methods include PHYML [10], etc. Finally, for maximum compatibility (MC) methods [2], it attempts to maximize the number of characters which are compatible with the given tree based on some predefined notion of character compatibility.

In this paper, we focus on the construction of the most parsimonious phylogenetic trees given sequence data of a group of species. As mentioned in [4], parsimony is probably the most widely used among all tree building algorithms. Although small parsimony tree problem (given tree topology and leaf labels, but not internal

node labels, find best internal node labels with the minimum parsimony score for that tree) is polynomial-time solvable for both uniform cost based on character change [5] (Fitch's algorithm) and non-uniform cost [18] (Sankoff's algorithm), large parsimony tree problem (given sequence data, find minimum parsimony score tree with leaves labeled by sequences) has been proven to be NP-hard [3,6]. There are a lot of efforts in trying to find efficient solutions for the construction of the most parsimonious phylogenetic trees. Most of the existing heuristic approaches still struggle with some moderate-size data sets in practice [20]. However, the data sets to be solved have been getting increasingly large in the genomic era of the 21<sup>st</sup> century. Some recent efforts has been focused on the most parsimonious tree problem using integer linear programming [20] but the number of variables and constraints could be exponential in worst cases. The intuition behind the integer linear programming approach is still of the *deterministic* optimization philosophy.

We present a novel random tree optimization algorithm based on the cross-entropy (CE) method [16,17]. The basic idea behind the cross entropy method is to translate the *deterministic* optimization problem into a related *stochastic* one and then use rare event simulation techniques to find the solution. We call the specification of the CE method to the most parsimonious phylogenetic trees (MPPT) problem the random tree optimization (RTO) algorithm. The RTO algorithm operates iteratively by randomly generating improved sample trees based on a common-parent probability matrix (CPPM) until the optimization process converges based on our predefined performance function, i.e., the minimal parsimony score (the total number of changes along all the edges of the tree) to explain the data.

The rest of the paper is organized as follows. The problem formulation is given in Section 2. We present the random tree optimization algorithm for the construction of the most parsimonious phylogenetic tree in Section 3. We analyze RTO in the framework of EM (expectation maximization) in Section 4. We give the conclusion and future directions in Section 5.

## 2. The Problem

Given  $n$  aligned strings of equal length (DNA sequences of different species, etc.), find a (binary) tree  $T$  by labeling its *leaf nodes* with these input strings, and by assigning its *internal nodes* similar strings so as to minimize the parsimony score over all possible trees and all possible labeling of the *internal nodes*.

Parsimony score is the total number of mutation events in which the value of some nucleotide base changes along some edge of the tree to explain the given sequence data. Formally, given a phylogenetic tree  $T$ , its parsimony score can be defined as follows:

$$S(T) = \sum_{(u,v) \in E(T)} |\{j : v_j \neq u_j\}| \quad (1)$$

where  $E(T)$  is the set of edges of the tree  $T$ ,  $(u,v)$  is an edge of the tree  $T$ , and  $u, v$  represent the corresponding strings associated with them.

The problem is to search for a tree topology that has the minimum number of substitutions, which is considered as the most parsimonious tree. One drawback with the parsimony approach is that the "backflips" phenomena such as  $A \rightarrow C \rightarrow A$  during the evolution of two species could not be taken into account by only looking at the given sequences of the two species.

In a phylogenetic tree, the leaf nodes denote the species and the internal nodes represent the hypothetical ancestors. Each edge in a phylogenetic tree reveals the evolutionary relationship between the two end nodes that join the given edge. The edge length indicates the evolutionary distance or evolutionary time between the two end nodes that join the given edge. The root node in a phylogenetic tree represents the ultimate ancestor of the group of species. Normally, we restrict ourselves to full binary trees for phylogenetic trees. Of course, we can add links of zero length when necessary.

Notably, the large parsimony problem (given sequence data, find minimum parsimony score tree with leaves labeled by the given sequences) is NP-complete [4,12,19].

## 3. The RTO Algorithm

The major contribution of this paper is the presentation of the random tree optimization approach based on cross-entropy (CE) method [16] to find the most parsimonious tree given aligned sequences of different species. The basic idea of CE method is to translate the deterministic optimization problem into a corresponding stochastic one and then use rare event simulation techniques to find the optimal solution.

As discussed in [16], CE (cross entropy) methods differs from other well-known random search algorithms for global optimization such as simulated annealing [1,7,15], tabu search [8] and genetic algorithms [9], which are *local* search heuristics and employ the notion of local neighborhood structures. CE method employs *multi-extremal* optimization process based on Kullback-Leibler cross-entropy, importance

sampling, etc. Therefore, CE method represents a *global* random search procedure rather than a *local* one.

### 3.1. Initialization of CPPM

We define pair-wise common parent probability matrix (**CPPM**) as the probability model for the generation of random tree samples. The pair-wise common-parent probability matrix is initialized based on the Jukes-Cantor distance  $-\frac{3}{4} \times \log_e(1 - \frac{4}{3}d)$  for pair-wise DNA strings of different species, where  $d$  is the fraction of sites where nucleotides differ. The Jukes-Cantor model produces a maximum likelihood estimate of the number of nucleotide substitutions between two sequences. We define  $Q = (q_{i,j})_{(n \times n)}$  as the common-parent probability matrix, where  $q_{i,j}$  denotes the probability that node  $i$  and node  $j$  have the common parent node. The sum of each row of the matrix is normalized as one.

Although we use Jukes-Cantor model to generate the pairwise distance matrix, there is no guarantee that those pairwise distances are either ultrametric or additive [4] (in practice, in most cases those pairwise distances are neither ultrametric nor additive). Therefore, most likely the UPGMA and neighbor joining algorithms [4] could not perform well toward the optimal solution without these conditions. We refer interested reader to [4] on the notions of ultrametric and additivity conditions, with which UPGMA and neighbor joining (**NJ**) algorithms perform well respectively (UPGMA with ultrametric condition, NJ with additivity condition).

It is important to note that we choose Jukes-Cantor distance over other alternative distances such as Hamming distance, Poisson distance, etc. It is our understanding that for the problem setting of DNA strings of different species, the choice of Jukes-Cantor distance is the most appropriate and natural [4].

In the case of solving the most parsimonious phylogenetic tree problem using CE method, the key is how to efficiently generate random trees based on the common-parent probability matrix (CPPM) that we introduced and how to efficiently initiate and update the CPPM. Note that the random tree generation algorithm has to be based on the CPPM in order to get improved sample trees after each iteration of the RTO algorithm.

### 3.2. Random Tree Generation

We initialize  $n$  un-parented leaf nodes, each of which is corresponding to the input DNA string of a given species.

The random tree generation algorithm operates as follows: we first *randomly* pick one un-parented node and then *randomly* pick a sibling node for this node based on the pair-wise common-parent probabilities in CPPM among other un-parented nodes, until all nodes find a unique parent node.

For the new row of the newly-formed internal node in the CPPM, we simply take the average of the original values for its two child nodes, which is reasonable if we assume equal mutation rate towards those two child nodes from their common parent node. Formally, we have

$$CPPM[m][i] = \frac{CPPM[r][i] + CPPM[l][i]}{2} \quad (2)$$

where  $m$  is the parent node,  $r$  and  $l$  are the corresponding child nodes,  $1 \leq i \leq 2n-1$  ( $n$  is the number of species).

Similarly, for the new column of the newly-formed internal node in the CPPM, we simply take the average of the original values for its two child nodes, which is reasonable if we assume equal mutation rate towards those two child nodes from their common parent node. Formally, we have

$$CPPM[i][m] = \frac{CPPM[i][r] + CPPM[i][l]}{2} \quad (3)$$

where  $m$  is the parent node,  $r$  and  $l$  are the corresponding child nodes,  $1 \leq i \leq 2n-1$  ( $n$  is the number of species).

Notably, after the normalization process along each row of the CPPM, the symmetry of the CPPM may not hold, i.e.,  $CPPM[i][j]$  may not be equal to  $CPPM[j][i]$ . That is why we need to calculate  $CPPM[m][i]$  (Eq. (2)) and  $CPPM[i][m]$  (Eq. (3)) separately.

After the two un-parented nodes find a new parent node, the corresponding rows and columns for those two nodes will be set as zeros. The rows of the CPPM will be renormalized such that the sum of each row is one.

Regarding the labeling of a newly-formed parent node (internal node) with two child nodes, we use the Union operation for each character position (per nucleotide) along the sequence (similar to the process in Fitch's algorithm), in which we take the *intersection* of the two sets if they have some character in common at the same position, otherwise, multiple choices resulted from the Union of the two sets at the same position are kept if they do not have some character in common. The *beauty* of our proposed random tree generation algorithm is that when a random tree is generated, the

parsimony score of the tree is also figured out at the same time. This is due to the fact that the way we determine the label for the parent node fits well with Fitch's algorithm [5] to calculate the parsimony score of the tree.

This process will continue till there are only two unparented nodes that have to be paired together in the end.

### 3.3. The RTO Algorithm

First, we define the performance function  $F(\text{tree})$  as the parsimony score of a tree, e.g., the total required number of changes along all edges of a tree to explain the data, which is given in Eq. (1). As mentioned earlier, there are two key components in the RTO algorithm: (a) Generation of *random* sample trees; (b) Update of the parameters at each iteration. The update mechanism is supposed to encourage trees with high performance so that the randomization mechanism would lead to trees with even better performance.

At each iteration of the RTO algorithm, we need to calculate the benchmark value of  $\gamma_t$  as follows:

$$\gamma_t = \min\{f : P_{Q_{t-1}}(F(T) \leq f) \geq \rho\}, \quad (4)$$

where  $\rho$  normally takes a value of 0.01 so that the event of obtaining high performance is not too rare,  $F(T)$  stand for the parsimony score of a randomly-generated sample tree, say  $T$ , based on the common-parent probability matrix in the  $(t-1)^{\text{th}}$  round, e.g.,  $Q_{t-1}$ ,  $P_{Q_{t-1}}(A)$  denote the probability of the event  $A$  conditioned on  $Q_{t-1}$ . Essentially,  $\gamma_t$  is the sample  $\rho$ -quantile of the performance of the randomly generated trees in the  $t^{\text{th}}$  round.

There are several choices to set the termination conditions. Normally, If for some  $t \geq l$ , say  $l = 5$ ,

$$\gamma_t = \gamma_{t-1} = \dots = \gamma_{t-l}, \quad (5)$$

then stop the optimization process.

The updated value of  $q_{i,j}$  can be estimated as:

$$q_{i,j}^e = \frac{\sum_{k=1}^M H_{\{F(T_k) \leq \gamma\}} H_{\{T_k \in T_{i,j}\}}}{\sum_{k=1}^M H_{\{F(T_k) \leq \gamma\}}} \quad (6)$$

where  $M$  stands for the number of sample trees,  $H_{\Omega}$  is an indicator function,  $T_{i,j}$  denotes the set of trees in which node  $i$  and node  $j$  have common parent node. While there are solid theoretical justifications for Equation (6), we refer the readers to [16,17], and focus

on the algorithms that were implemented in practice. In order to avoid overly quick convergence to 1s and 0s for the update of  $q_{i,j}$ , which could limit the randomness of the sample trees, normally we use a *smoothed* update procedure in which

$$q_{i,j}^t = \alpha \times q_{i,j}^e + (1 - \alpha) \times q_{i,j}^{t-1} \quad (7)$$

where  $q_{i,j}^{t-1}$  is the value of  $q_{i,j}$  in the previous round and  $q_{i,j}^e$  is the estimated value of  $q_{i,j}$  based on the performance in the previous round according to Equation (6), and  $q_{i,j}^t$  stands for the value of  $q_{i,j}$  for the current round. Empirically, a value of  $\alpha$  between  $0.4 \leq \alpha \leq 0.9$  gives the best results as in [16,17].

In summary, we have RTO Algorithm operation flow as follows based on CE [16,17]:

1. Set  $t = 1$  and set  $Q_0$  according to the initialization of  $q_{i,j}$ .
2. Randomly generate sample trees (typically  $20n^2$  sample trees).
3. Calculate  $\gamma_t$  according to Equation (4).
4. Update  $q_{i,j}$  according to Equation (6) and Equation (7).
5. If for some  $t \geq l$ , say  $l = 5$ , such that  $\gamma_t = \gamma_{t-1} = \dots = \gamma_{t-l}$ , then stop; otherwise, reiterate from step 2.

## 4. RTO in the Framework of EM

In this section we analyze RTO algorithm in the framework of EM (expectation maximization) [23].

Let  $D$  indicate the values of the observed variables, e.g., the sequence data of the species, and let  $T$  denote the hidden data, e.g., the parsimonious phylogenetic trees. Let  $P$  be the probability mass function of the complete data with parameters given by the matrix CPPM. So, we have  $P(D, T | \text{CPPM})$  as the complete data likelihood, which can be thought of as a function of CPPM.

By using the Bayes's rule and the law of total probability, the conditional probability of the hidden data given the observed data and CPPM can be expressed as:

$$\begin{aligned} P(T | D, \text{CPPM}) &= \frac{P(D, T | \text{CPPM})}{P(D | \text{CPPM})} \\ &= \frac{P(D | T, \text{CPPM}) P(T | \text{CPPM})}{\sum_{\hat{T}} P(D | \hat{T}, \text{CPPM}) P(\hat{T} | \text{CPPM})} \end{aligned} \quad (8)$$

where  $\hat{T}$  indicates the estimated tree samples.

The goal is to estimate CPPM. The E-step is given by:

$$E: Q(CPPM) = E_T[\log P(D, T | CPPM)] \quad (9)$$

where  $Q(CPPM)$  is the expected value of the log-likelihood of the complete data.

The expected value of log-likelihood in Eq. (9) can be further expressed as:

$$Q(CPPM) = \sum_T P(T | D, CPPM) \times \log P(D, T | CPPM) \quad (10)$$

In the RTO algorithm, CPPM evolves from  $CPPM_0$  to  $CPPM_1, CPPM_2, \dots, CPPM_{t-1}, CPPM_t$  based on the improved tree samples according to Eq. (6), and Eq. (7). The update process of  $q_{i,j}$ s, the elements of the CPPM, according to Eq. (6) and Eq. (7) minimizes the cross-entropy based on the Kullback-Leibler cross-entropy principle [16] and importance sampling philosophy.

The M-step is thus given by:

$$M: CPPM_t = \arg \max_{CPPM} Q(CPPM) \quad (11)$$

$CPPM_t$  is the value that maximizes (M-step) the conditional expectation (E-step, e.g., Eq. (9)) of the complete data log-likelihood given the observed sequence data under the previous parameter value of  $CPPM_{t-1}$ . Notably, for each set of sample trees, e.g.,  $T$ , there is a likelihood value for CPPM. We can thus calculate an expected value of the likelihood, which depends on the previously assumed value of CPPM as it influenced the probabilities of the sample trees, e.g.,  $T$ .

In traditional EM algorithm, it can be shown that an EM iteration does not decrease the observed data likelihood function. However, there is no guarantee that the sequence converges to a maximum likelihood estimator. In other words, EM is a local search algorithm. In the RTO algorithm based on CE method, a multi-extremal search is employed due to the adoption of Kullback-Leibler cross-entropy [16] and importance sampling techniques. It can be shown [16] that with high probability the observed data likelihood function increases and eventually converges to one so long as the number of samples is large enough and the number of iterations is large enough.

Formally, we have the following property holds with high probability for RTO algorithm based on CE method:

$$\begin{aligned} P(D | CPPM_i) &= \sum_T P(D, T | CPPM_i) \leq P(D | CPPM_{i+t}) \\ &= \sum_T P(D, T | CPPM_{i+t}) \end{aligned} \quad (12)$$

When both  $t$  and the number of samples, e.g.,  $|T|$ , go to infinity, the data likelihood function converges. This property of CE method is proved in [16] based on the stochastic nature of Kullback-Leibler cross-entropy process. We refer interested readers to Appendix A in [23] for details.

## 5. Conclusion and Future Directions

With the availability of ever-increasing gene sequence data across a large number of species, reconstruction of phylogenetic trees to reveal the evolution relationship among those species becomes more and more important. In this paper, we focus on the construction of the most parsimonious trees given sequence data of a group of species as parsimony is probably the most widely used among all tree building algorithms [4]. The major contribution of this paper is the presentation of a novel algorithm, the random tree optimization (RTO) algorithm based on cross-entropy method [16], for the construction of the most parsimonious phylogenetic trees. We analyze the RTO algorithm in the framework of expectation maximization (EM) and point out the similarities and differences between traditional EM algorithm and the RTO algorithm.

We will conduct experimental tests for RTO algorithm with different sets of input data and compare its performance with the benchmark data reported by Rasmussen and Kellis in [14] as our future directions.

## Acknowledgement

The authors would like to thank Matt Rasmussen, James Galagan, Manolis Kellis, Mike Lin and David Sontag for insightful comments and various assistance. Fulu Li would also like to thank the Digital Life consortium at MIT Media Lab for the support.

## Reference

- [1] Aarts E., J. Korst, "Simulated Annealing and Boltzmann Machines", *John Wiley & Sons*, 1989.
- [2] Bonet M., M. Steel, T. Waxnow, and S. Yooseph, "Better Methods for Solving Parsimony and Compatibility", in the proceeding of *ACM RECOMB* '1998.
- [3] Day W. and D. Sankoff, "Computational complexity of inferring phylogenies by compatibility", *Systematic Zoology*, 35:224-229, 1986.
- [4] Durbin R., S. Eddy, S. Krogh, G. Mitchison, "Biological Sequence Analysis", *Cambridge Press*, 2006.

- [5] Fitch W. M., "Toward Defining the Course of Evolution: Minimum Change for Specific Tree Topology," *J. Zoological System*, vol. 20, pp. 406-416, 1971.
- [6] Foulds L. R. and R. L. Graham, "The Steiner problem in phylogeny is NP-complete", *Advances in Applied mathematics*, 3:43-49, 1982.
- [7] Geyer C., E. Thompson, "Annealing Markov chain Monte-Carlo with applications to Ancestral Inference", *Journal of the American Statistical Association*, 1995.
- [8] Glover F., M. Laguna, "Tabu Search", a chapter in *Modern Heuristic Techniques for Combinatorial Optimization*, 1992.
- [9] Goldberg D., "Genetic Algorithms in Search, Optimization and Machine Learning", *Addison Wesley*, 1989.
- [10] Guindon S., O. Gascuel, "PhyML - A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood", *Systematic Biology*, 2003 52(5): 696-704.
- [11] Gupta A., J. Manuch, L. Stacho and C. Zhu, "Small Phylogeny Problem: Character Evolution Trees", *Springer Press*, 2004.
- [12] Olken F., "Phylogenetic Tree Computation Tutorial", Lawrence Berkeley National Lab, Presentation to PGA course, May 2002.
- [13] Philip program, <http://bioweb.pasteur.fr/seqanal/phylogeny/phytip-uk.html>
- [14] Rasmussen M., M. Kellis, "Accurate Gene-tree Reconstruction by Learning Gene-and Species-specific Substitution Rates across Multiple Complete Genomes", *Genome Research*, Nov. 2007.
- [15] Romeijn H., R. Smith, "Simulated Annealing for Constrained Global Optimization", *Journal of Global Optimization*, 1994.
- [16] Rubinstein R., "The Cross-Entropy Method for Combinatorial and Continuous Optimization", *Methodology And Computing in Applied Probability*, 1999.
- [17] Rubinstein R., D. Kroese, "The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning", *Springer*, 2004.
- [18] Sankoff D. "Minimal mutation trees of sequences", *SIAM Journal on Applied Mathematics*, 28:35-42, 1975.
- [19] Shamir R., "Phylogenetics and Phylogenetic Trees", *lecture notes in the class of Algorithms for Molecular Biology*, Dec. 2001.
- [20] Sridhar S., F. Lam, G. Blelloch, R. Ravi, R. Schwartz, "Efficiently Finding the Most Parsimonious Phylogenetic Tree via Linear Programming", in *ISBRA '2007*.
- [21] Stoye J., D. Evers, F. Meyer, "Rose: generating sequence families", *Bioinformatics*, 1998.
- [22] Yang Z., "Phylogenetic Analysis Using Parsimony and Likelihood Methods", in *J. of Molecular Evolution*, 1996.
- [23] Expectation Maximization: [http://en.wikipedia.org/wiki/Expectation-maximization\\_algorithm](http://en.wikipedia.org/wiki/Expectation-maximization_algorithm)