Critical assessment of phylogenetic algorithms

Gergana Vandova BIOC218 2014

Phylogenetic analyses are important part of the biological research, the idea behind which is that species are related through a history of common descent. A main goal of phylogenetics is to reconstruct how a group of species evolved in time by building a tree that would trace that evolution. Constructing such a tree would provide the opportunity to observe speciation events, horizontal gene transfer, recombination, gene duplication, and gene losses. These events are crucial for reconstructing the history of species, diversification of organisms, and the emergence of new cellular functions. Constructing a tree would also be beneficial to understand the forces and constraints of evolution on the group of sequences of interest. Knowing how a set of sequences are related could also be applied for translational research, such as drug discovery.

Unfortunately, there is no way of knowing whether the built tree is the true tree. This problem is addressed by developing a model of DNA evolution, which would explain the most important features of evolution of the taxa of interest. There are several models of DNA evolution (Figure 1), each one of which describes the rate at which one DNA base is replaced by another in the course of evolution. The most common models are the following:

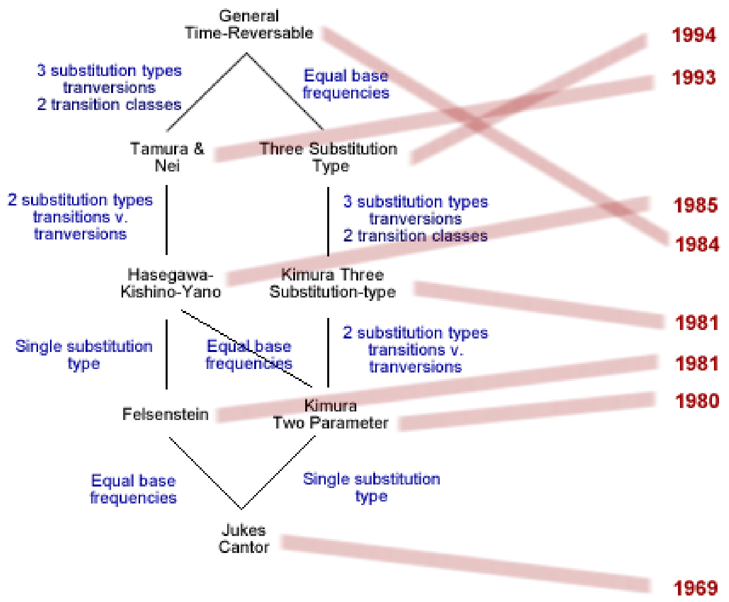


Figure 1. Models of DNA evolution. Figure from http://authors.library.caltech.edu/5456/1/hrst.mit.edu/hrs/evolution/public/models/sequence.html

1) Jukes-Cantor (JC69) model [1] – assumes that all four bases occur with equal frequencies and changes from one base to another occur with the same probability for all bases;
2) Kimura (K80) model [2] - distinguishes between transitions (purines to purines, pirimidines to pirimidines) and transversions (any base to any base);
3) Felsenstein (F81) model [3] – a modification of Jukes-Cantor model, in which the base frequencies vary;
4) Hasegawa, Kishino and Yano (HKY85) model [4] – a combination of Kimura and Felsenstein models – different frequency of occurrence of all bases and different rates of substitutions – one for transitions and one for transversions;

5) Tamura (T92) model [5] – an extension of the Kimura model by adding a GC content bias;
6) Tamura and Nei (TN93) model [6] – assumes different rates for the two transitions and the same rate for the two transversions;
7) GTR: Generalised time-reversible model [7] – it is the most general model, as it utilizes different frequencies of each base and different rates of substitutions.

The chosen model of evolution could be a simple Jukes-Cantor model, or dependence between different sites in the sequences could be assumed. Then a tree reconstruction method that is known to work well for sequences that have evolved under the chosen model is used. There is a trade-off between the simplicity of the model used and the reliability of the data. The more simple the model, less data is needed to construct the tree, less computationally-expensive it is, but less reliable the reconstructed tree is.

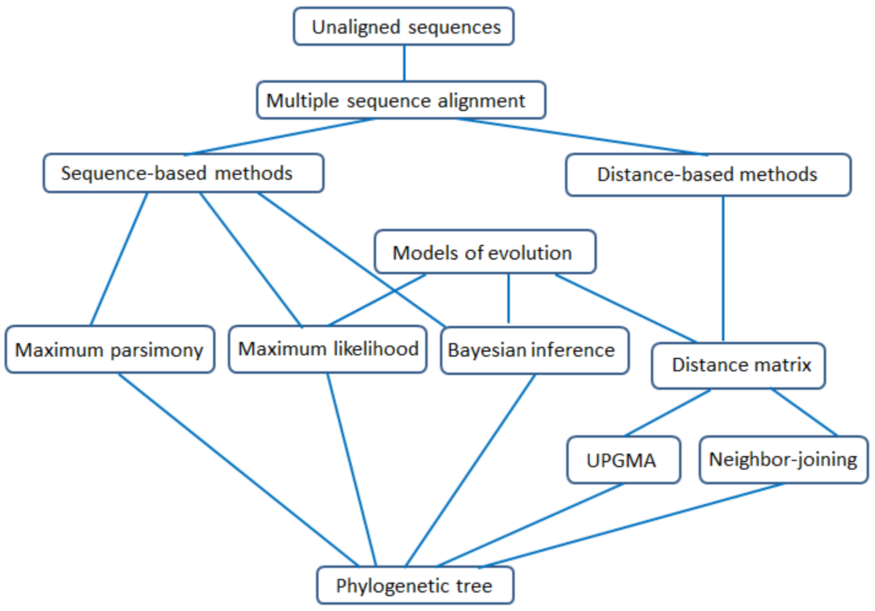


Figure 2. Overview of algorithms for phylogenetic tree reconstruction.

There are two main classes of algorithms used for phylogenetic tree reconstruction: sequence-based and distance-based (Figure 2). The sequence-based methods usually construct a phylogenetic tree T that optimally explains a given multiple sequence alignment M. The distance-based methods construct a phylogenetic tree T from a given distance matrix D. The sequence-based methods are divided in the following algorithms:

1) Maximum parsimony [8] – this method finds the phylogenetic tree that will explain the given data with the least number of observable mutations;
2) Maximum likelihood [9] – this method will output a tree that will explain the data with the highest likelihood given a specified model of evolution;
3) Bayesian method [10] – this method computes the posterior distribution of trees based on a given data, a specified model of evolution, and a presumed prior distribution of phylogenetic trees.

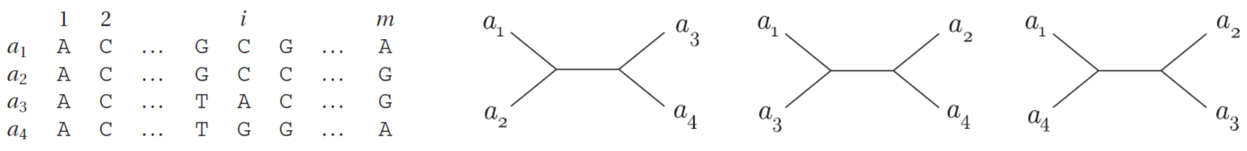
Maximum parsimony

The maximum parsimony method utilizes a preference for the least complicated explanation of the data. This method finds the phylogenetic tree that will explain the given data with the least number of evolutionary events (observable mutations). The maximum parsimony method consists of constructing all possible trees and then evaluating how well (with the least number of mutations) each one of them explains the data.

Pseudo-code for maximum parsimony:

```
1. Create multiple sequence alignment (MSA) of the sequences of interest to define the columns of
aligned characters
2. For each column position i:
        For each tree T:
                Count the number of mutations required for T to explain i
3. The best tree is the tree with the minimum number of mutations over all column positions
```

For a multiple sequence alignment of 4 sequences there are three possible phylogenetic trees (Figure 3).



	1	2			i			m
a_1	A	C	…	G	C	G	…	A
a_2	A	C	…	G	C	C	…	G
a_3	A	C	…	T	A	C	…	G
a_4	A	C	…	T	G	G	…	A

Figure 3. Multiple sequence alignment of four sequences (left) and the three possible trees that could be generated from it (right) Figure adapted from [20].

For column i of the MSA the possible tree topologies are the following (Figure 4):

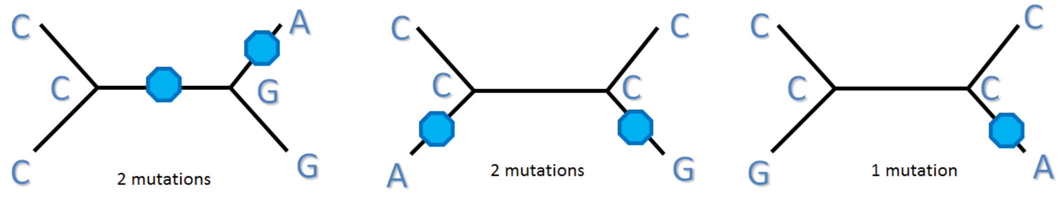


Figure 4. Possible topologies for column i of the given MSA in Figure 3.

The best tree for column i is the last one, as only one mutation is needed (from A to C) to explain the multiple sequence alignment at that column.

This evaluation is done for all columns and the tree with the minimum number of evolutionary events that explain the multiple sequence alignment is constructed.

Advantages:

1) Simple method, as the simplest explanation of the data is considered the correct one;
2) Does not assume model of evolution

Disadvantages:

1) Computationally expensive for large number of sequences because of the exhaustive enumeration of trees – the maximum parsimony problem is NP-complete.
2) The mutation rate may differ in the different columns of the MSA (some sites of the protein of interest might be under stronger selective pressure than others), which is not accounted for in the algorithm
3) The high variability columns would dominate the tree evaluation, since there would be a lot of mutations to be explained.
4) The algorithm underestimates branch lengths, as it does not account for multiple substitutions or reversed mutations at the same site.

There are many available software tools that have implemented the maximum parsimony method, such as MEGA [11], Mesquite [12], and PAUP [13], to list a few.

Maximum likelihood

The idea behind this method is to find the tree that is more likely, given a specified model of sequence evolution. The input is again a multiple sequence alignment and this method would construct all possible trees and evaluate how well each one of them explains the given data. The maximum likelihood

$L(T) = P(M|T, \mathcal{M})$,

of a tree T is defined as the probability of the multiple sequence alignment M given the constructed tree T under the model of evolution $\mathcal{M}$.

The model of evolution would explain how sequences along the edge of the tree evolve. If the simple Jukes –Cantor model is applied, then the probability of occurrence of any base is equal to 0.25 at each position of the sequence and each position of the sequence can mutate with the same rate. More general models are usually implemented, in which the frequencies of occurring of the bases is different and the rate of mutation is different for the two possible transitions.

Pseudo-code for maximum likelihood:

```
1. Do MSA of the sequences of interest to define the columns of aligned characters
2. For each column position i:
        For each tree T:
                For each labeling of internal nodes:
                        Compute that likelihood of that pattern of internal nodes
3. The best tree is the tree with the maximum probability over all column positions
```

For a multiple sequence alignment of 4 sequences there are three possible phylogenetic trees (Figure 3). For column i of the MSA these are the possible tree topologies:

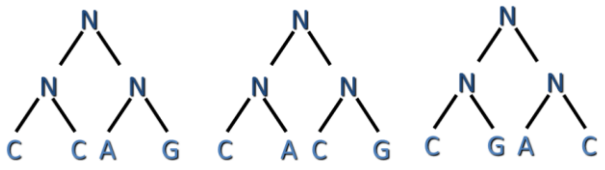


Figure 5. Possible topologies for column i of the given MSA in Figure 3.

For the first topology there are $4^3 = 64$ different labelings of the internal nodes:

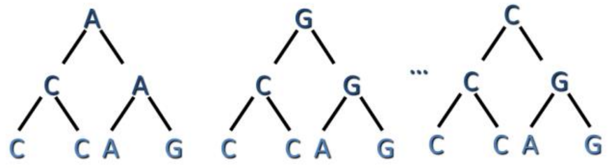


Figure 6. Possible labelings of the internal nodes of the tree with the first topology from Figure 5.

For each of these labelings, the likelihood of that pattern is calculated, given the model of evolution. For the first labeling in Figure 6, the probability equals:

Probability = $P(A) * P(C\text{->}C)^2 * P(A\text{->}A)^2 * P(A\text{->}G) * P(A\text{->}C)$

Thus, the likelihood that a tree T generated the MSA at position i, is calculated by summing the probabilities of all possible labelings for that tree:

$$L(i, T) = \mathrm{P}(M_i | T, \omega, \mathcal{M}) = \sum_{\alpha} \mathrm{P}(M_i, \alpha | T, \omega, \mathcal{M}) = \sum_{\alpha} \left(\mathrm{P}(\alpha(\rho)) \prod_{e=\{v,w\}} \mathrm{P}_e(\alpha(v), \alpha(w)) \right)$$

where α are all possible labelings of that tree, P(α(ρ)) is the probability of generating the state α(ρ) at the root and P_e(α(v), α(w)) is the probability of observing the two states α(v) and α(w) at the two ends of an edge e = {v, w}.

The likelihood that a tree T generated the whole multiple sequence alignment M is calculated by multiplying the likelihoods for all positions of M:

$$L(T) = L(1, T) \cdot L(2, T) \cdot \dots \cdot L(m, T) = \prod_{i=1}^{m} L(i, T)$$

Advantages:

1) Knowledge and assumptions about the process that generated the data can be implemented;
2) Can handle diverse sequences, because it can consider multiple mutations in the model;
3) Different mutation rates in the different sites of the sequences can be assigned;
4) Can have sub-tree-specific models to account for example for recent vs long-term evolution.

Disadvantages:

The exhaustive enumeration of trees and internal nodes is very computationally expensive method, especially for large number of sequences – finding an optimal phylogenetic tree is NP-complete.

There are many available software tools that have implemented the maximum likelihood method, as here are listed several of them: MEGA [11], Mesquite [12], PhyML [13], RaxML [14].

Bayesian methods

The Bayesian inference of phylogeny is based on a distribution of a quantity called the posterior probability of a phylogenetic tree. This posterior probability can be calculated by the Bayes's theorem:

$$P(T \mid M) = \frac{P(M \mid T) \times P(T)}{P(M)}$$

where the posterior probability distribution of a tree given the data (P(T|M) is calculated by taking the prior probability of a phylogenetic tree (P(T)) and the likelihood of the data given the tree (P(M|T).

The posterior probability is also interpreted as the probability that the tree is correct. The prior probability distribution of phylogenetic trees is the probability of the distribution of all possible trees without any prior knowledge of the data.

Initially, all possible trees are considered to be equally probable (Figure 7). Then the posterior probability is proportional to the likelihood and the maximum likelihood tree is the one with maximum posterior probability. A subset of phylogenetic trees is chosen for the subsequent analysis to generate a consensus phylogenetic tree.

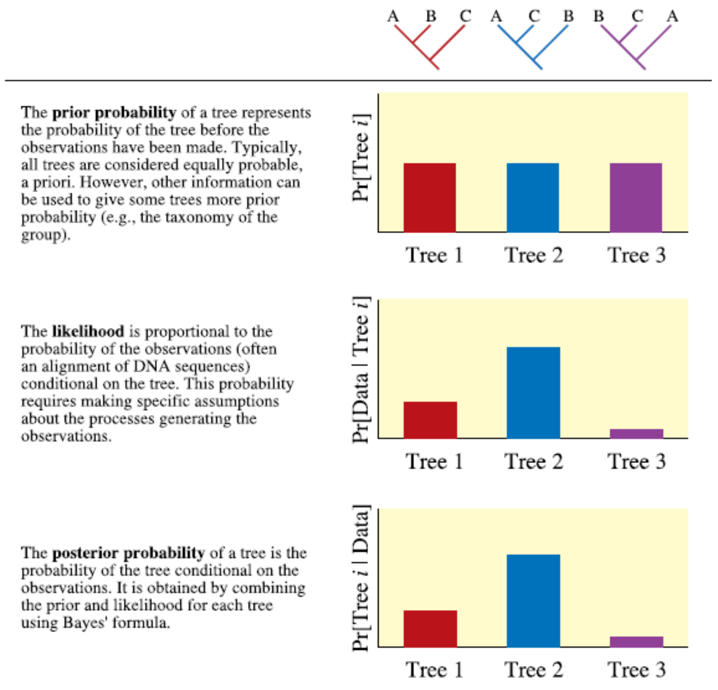


Figure 7. Bayesian inference algorithm. Figure from [10]

The calculation of the posterior probability includes a summation over all possible tree topologies and, for each tree, integration over all possible branch lengths and substitution model parameters, which is very difficult to calculate. An approximation for this calculation is implemented using the Markov chain Monte Carlo (MCMC) approach [15]. The MCMC approach constructs a chain of phylogenetic trees. At each iteration, a modification to the tree is proposed and a probabilistic approach is used to either accept or reject the new modification. If T_i is the phylogenetic tree at the i^{th} position of the MCMC chain and if the T' is the modified tree, then the decision of whether to accept the modification is based on the ratio of the posterior probabilities of the two trees, R:

$$R = \frac{P(T' \mid M)}{P(T_i \mid M)} = \frac{P(M \mid T')P(T')P(M)}{P(M \mid T_i)P(T_i)P(M)} = \frac{P(M \mid T')P(T')}{P(M \mid T_i)P(T_i)}$$

The modified tree T' is accepted with probability min (1, R). This means that, if the posterior probability of T' is higher than the posterior probability of T_i, T' is accepted. If the posterior probability of T' is lower, T' is accepted with a random probability between 0 and 1. If T' is accepted T_{i+1} is set to T'. The goal is to produce an MCMC chain of binary phylogenetic trees, which stationary distribution approximates the posterior probability distribution of the trees.

In order for this method to work properly, there are several conditions that have to be met:

1) The tree modifications should be small enough, so that a sudden drop in posterior probability can be avoided;
2) The tree modifications should be large enough, so that a sufficient parameter space is explored for a reasonable computational time.
3) The MCMC chain starts at a random location at the parameter space with low posterior probability. Thus, it needs sufficient time to start sampling from the posterior probability.
4) The final set of output trees is obtained by sparsely sampling the MCMC chain and thus removing the auto-correlation between very similar trees.

Advantages:

1) It provides a distribution of phylogenetic trees instead of one optimal tree
2) The algorithm is much faster than maximum likelihood with bootstrap resampling [16].

Disadvantages:

1) Bayesian inference is still very slow and not applicable for large datasets.

2) Another major challenge with this method is how to determine whether the algorithm has converged, i.e. the posterior probability distribution is well approximated. One way to judge about convergence is to monitor the likelihood of each new tree added to the chain and if it does not increase, then convergence must have been reached. Another option is to start from several MCMC chains and to iterate the algorithm until all of them are sampling from the same posterior distribution.

There are many online tools that have implemented Bayesian inference, as one of the most commonly used one is MrBayes [17].

The second main group of phylogenetic reconstruction methods is the distance-based methods. To apply these distance approaches, first a distance matrix has to be computed and then a phylogenetic tree T is built from this distance matrix D.

The **advantages** of distance-based methods over the sequence based are several:

1) They can be used for different types of data and there is no need to develop model of evolution;
2) They are faster and can be applied to thousands of taxa for reasonable time.

The input is a multiple sequence alignment. In the simplest approach a Hamming distance H (a, b) is used, which is defined as the proportion of positions two aligned sequences a and b differ. The Hamming distance between two aligned sequences is an underestimation of their true evolutionary distance (the average number of mutations at a certain site over the elapsed time), as reversed mutations and multiple mutations at the same site are not accounted for. To correct for the underestimation, a correction formula based on a chosen model of evolution is used.

If T is the phylogenetic tree for a set of sequences, then the tree distance between any two sequences equals the sum of all lengths of all the edges on the unique tree path from the two leafs labeled with the two sequences. The goal of the distance-based methods is to construct a phylogenetic tree T, for which the tree distances approximate the distances in the distance function. The tree length of T is defined as the sum of the lengths over all edges of the tree. Tree-like distance have the property to be additive, i.e. they can be calculated by adding the edge lengths along the paths of a tree. This property can be determined using the *Four-point condition*: for each four sequences x, y, z, and w, if the following equation

$d(w, x)+d(y, z) \leq \max\{d(w, y)+d(x, z),d(w, z)+d(x, y)\}$ is true, then the tree is additive.

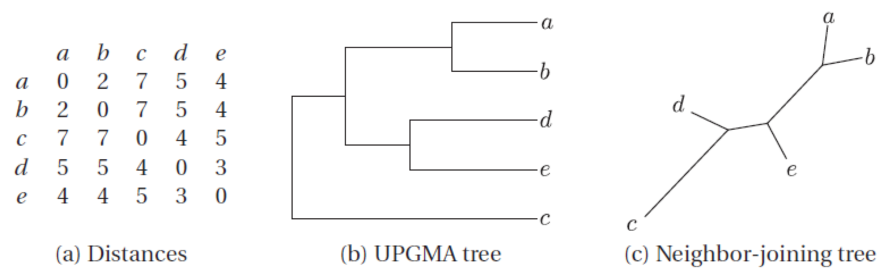
If the *molecular clock theory* is correct (that is, for any protein, accepted mutations in the amino acid sequence occur at a constant rate over all sites of the sequences), then the phylogenetic tree is called *ultrametric* and the distance matrix used to obtain that tree - ultrametric. For an ultrametric tree, all of the leaves of the tree are on the same distance from the root of the tree. To determine whether a distance matrix is ultrametric, the *Three-point condition* has to be fulfilled: for every three sequences x, y, and z, either all distances between the three pairs of taxa are equal ($d(x, y) = d(y, z) = d(x, z)$, or two of the distances are equal and are larger than the third ($d(x, y) = d(y, z) > d(y, z)$.

In this review, I will describe two distance-based methods for building phylogenetic trees:

1) UPGMA [18]
2) Neighbor-joining method [19]

UPGMA

UPGMA, or *Unweighted pair group methods using arithmetic averages* is a distance-based method that as an input takes a multiple sequence alignment and using a distance matrix outputs a rooted phylogenetic tree. The main principle behind this algorithm is that it starts by clustering pairs of leaves, then pairs of clustered leaves, and so on, and thus assembling the tree from bottom-up.



	a	b	c	d	e
a	0	2	7	5	4
b	2	0	7	5	4
c	7	7	0	4	5
d	5	5	4	0	3
e	4	4	5	3	0

Figure 8. a) distance matrix D, b) the corresponding UPGMA tree, and c) Neighbor-joining tree. Figure from [20].

A distance matrix D depicts all the distances d(x, y) between any two pairs of sequences x, y and initially each sequence is placed at its own cluster with a node of height 0. The distance between two disjoined clusters C_i and C_j, d(i, j) = d(C_i, C_j) as the average distance between pairs of sequences from each cluster:

$$d(i, j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(x, y).$$

If C_k unites the two clusters C_i and C_j, to calculate the distance between the newly united cluster C_k and any other cluster C_l, the following formula is used:

$$d(k, l) = \frac{d(i, l)|C_i| + d(j, l)|C_j|}{|C_i| + |C_j|}$$

The new cluster C_k has a node at height d(i, j)/2 that connects the two nodes of the clusters C_i and C_j. At each step the algorithm merges the two closest clusters until only one cluster remains and it has a complexity of $O(N^3)$. All the leaves of a phylogenetic tree built by the UPGMA algorithm have the same distance to the root of the tree.

The major **disadvantage** of UPGMA algorithm is that it assumes a molecular clock. This means that the distance matrix used to compute the phylogenetic tree should be ultrametric. In this case, the closest nodes are also neighbors. In the above example, the distance matrix is not ultrametric which implies that the distances obtained from the UPGMA tree are not equal to the distances from the distance matrix. Thus, in the general case, two clusters may be separated by a short distance but without being true neighbors. The next algorithm surpasses this problem.

Neighbor-joining method

The neighbor-joining (NJ) method takes as an input a multiple sequence alignment and similar to UPGMA uses a distance matrix and agglomerative clustering to construct an unrooted phylogenetic tree. The NJ algorithm constructs a phylogenetic tree if the distance matrix is additive. To surpass the problem mentioned above, it calculates the average distance of each cluster to all other clusters to compensate for very large distances.

C_i and C_j are to be joined in a new cluster C_k if the neighbor-joining matrix is minimized:

$$N(C_i, C_j) = d(C_i, C_j) - (r_i + r_j)$$

Where

$$r_i = \frac{1}{|\mathcal{C}| - 2} \sum_{C \in \mathcal{C}} d(C_i, C)$$

And C is the set of current clusters. To calculate the distance between Ck and any other cluster Cm:

$$d(C_i, C_m) = d(C_i, C_k) + d(C_k, C_m)$$

$$d(C_j, C_m) = d(C_j, C_k) + d(C_k, C_m)$$

$$d(C_i, C_j) = d(C_i, C_k) + d(C_j, C_k)$$

Then:

$$d(C_i, C_m) + d(C_j, C_m) = d(C_i, C_k) + d(C_k, C_m) + d(C_j, C_k) + d(C_k, C_m)$$

$$= d(C_i, C_j) + 2d(C_k, C_m)$$

Thus, the distance between Ck and Cm is:

$$d(C_k, C_m) = \frac{1}{2}\left(d(C_i, C_m) + d(C_j, C_m) - d(C_i, C_j)\right)$$

(Figure 9 a), b)). The length of the new edge connecting the nodes i and k is:

$$d(C_i, C_k) = \tfrac{1}{2}(d(C_i, C_j) + r_i - r_j)$$

(Figure 9c).

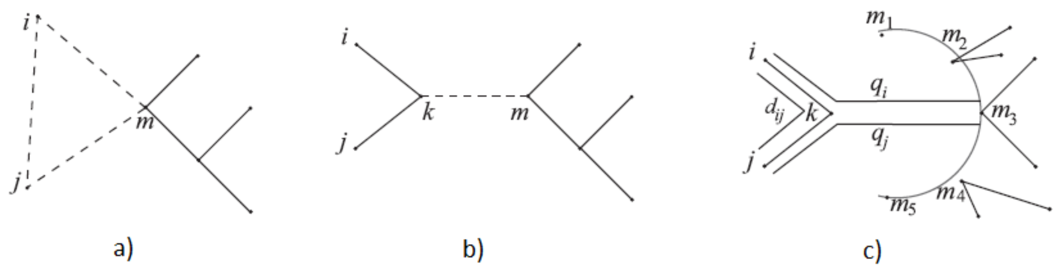


Figure 9. a) The dashed lines represent the distances between clusters Ci, Cj, and Cm before Ci and Cj were merged; b) the dashed line represents the distance between clusters Ck and Cm after merging Ci and Cj into Ck; c) The length between i and k equals to one half of the average distance of i to all current clusters Cm (qi) plus the distance from Ci to Cj, minus the average distance from Cj to all Cm clusters (qj). Figure from [20].

Disadvantages:

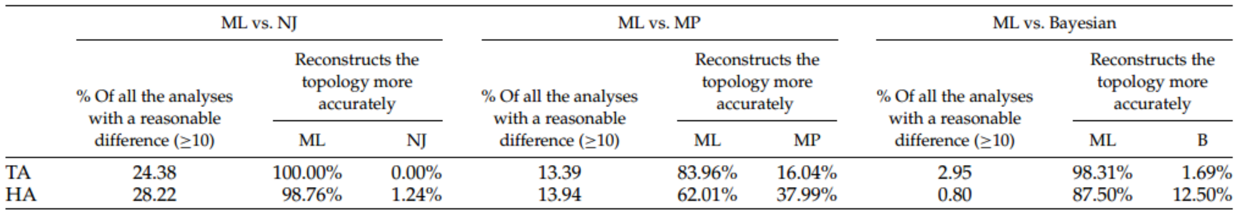
Distance matrixes in reality are rarely ultrametric or additive and thus the trees build by either UPGMA or NJ algorithm are not correct. Nevertheless, both algorithms are used with the hope that the tree distance deviations would not greatly affect the topology of the computed tree.

The most commonly used tool that has implemented both distance-based methods UPGMA and NJ is ClustalW [21].

Comparing the performance of different methods

There are several important features that should be considered when choosing a tree-building algorithm: accuracy, reproducibility, local and global tree topology, and speed. There are many studies that compared different methods, one of which [22] compared topological tree accuracy between

Neighbor-Joining (NJ), maximum parsimony (MP), maximum likelihood (ML), and Bayesian (B). The results of this study are summarized in the table below:

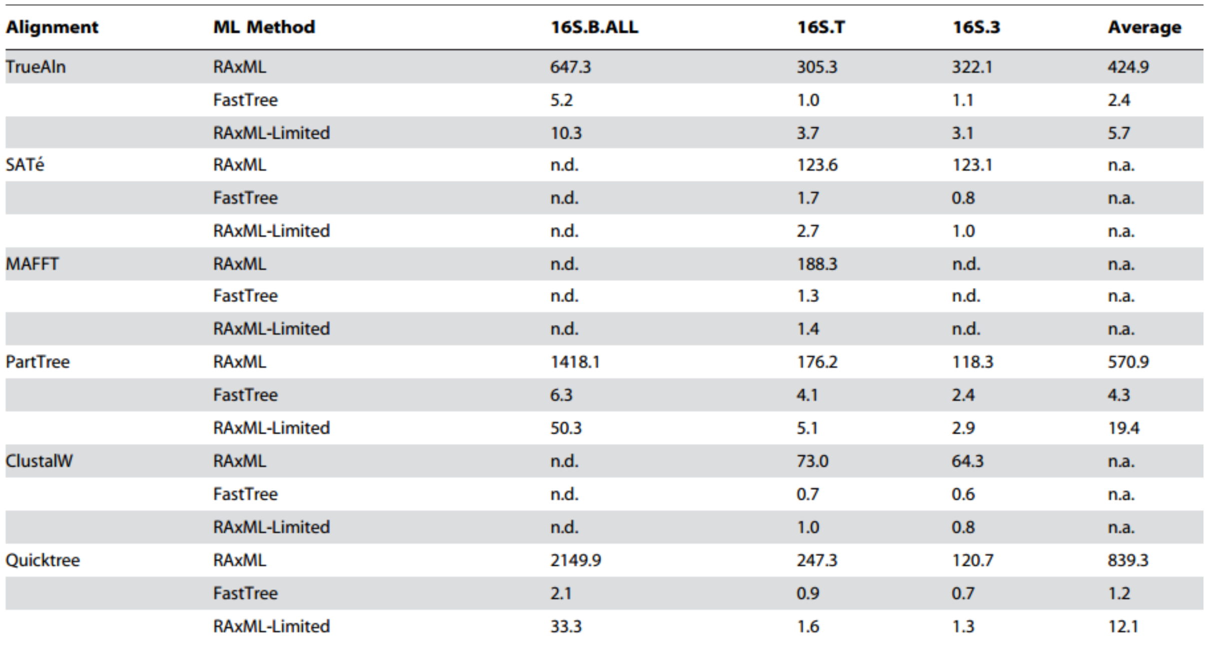


	ML vs. NJ			ML vs. MP			ML vs. Bayesian		
	% Of all the analyses with a reasonable difference (≥10)	Reconstructs the topology more accurately		% Of all the analyses with a reasonable difference (≥10)	Reconstructs the topology more accurately		% Of all the analyses with a reasonable difference (≥10)	Reconstructs the topology more accurately	
		ML	NJ		ML	MP		ML	B
TA	24.38	100.00%	0.00%	13.39	83.96%	16.04%	2.95	98.31%	1.69%
HA	28.22	98.76%	1.24%	13.94	62.01%	37.99%	0.80	87.50%	12.50%

Table 1. Comparison of topological accuracy of ML versus the other three methods. TA = True Alignment; HA = Hypothesized Alignment.

These analyses demonstrate that the best performing algorithms are ML and B, as ML slightly outperforms B. However, when there is a difference, ML is better at reconstructing the correct tree topology.

In addition to accuracy, another important characteristic of all tree-building algorithms is their speed. For large alignments, the most scalable maximum likelihood methods is RAxML [14] but it still would require weeks when used on datasets of thousands of sequences (for example analysis of about 28,000 sequences would require approximately a month of CPU time [23]. A faster method for constructing tree with good ML scores is FastTree [24]. FastTree utilizes the ''minimum-evolution'' principle, as it looks for a tree topology that minimizes the sum of the branch lengths. FastTree uses a heuristic variant of neighbor joining to generate an initial rough topology of the tree and then uses nearest-neighbor interchanges to refine the topology. The theoretical running time is O(N sqrt(N) log(N)La), where N is the number of sequences, L is the length of the MSA, and a is the number of characters in the used alphabet. FastTree and RAxML produce comparably accurate topologies (RAxML still outperforms FastTree on the more accurate alignments, whereas FastTree produces more accurate trees on the less accurate alignments). As far as computational time is concerned, FastTree outperforms RaxML dramatically (Table 2) [25].



Alignment	ML Method	16S.B.ALL	16S.T	16S.3	Average
TrueAln	RAxML	647.3	305.3	322.1	424.9
	FastTree	5.2	1.0	1.1	2.4
	RAxML-Limited	10.3	3.7	3.1	5.7
SATé	RAxML	n.d.	123.6	123.1	n.a.
	FastTree	n.d.	1.7	0.8	n.a.
	RAxML-Limited	n.d.	2.7	1.0	n.a.
MAFFT	RAxML	n.d.	188.3	n.d.	n.a.
	FastTree	n.d.	1.3	n.d.	n.a.
	RAxML-Limited	n.d.	1.4	n.d.	n.a.
PartTree	RAxML	1418.1	176.2	118.3	570.9
	FastTree	6.3	4.1	2.4	4.3
	RAxML-Limited	50.3	5.1	2.9	19.4
ClustalW	RAxML	n.d.	73.0	64.3	n.a.
	FastTree	n.d.	0.7	0.6	n.a.
	RAxML-Limited	n.d.	1.0	0.8	n.a.
Quicktree	RAxML	2149.9	247.3	120.7	839.3
	FastTree	2.1	0.9	0.7	1.2
	RAxML-Limited	33.3	1.6	1.3	12.1

Table 2. Runtime (h) of ML methods on alignments of three largest biological datasets.

In conclusion, there are many different algorithms for phylogenetic tree reconstruction and all of them are implemented in numerous user-friendly tools. Although each method has its advantages and disadvantages, choosing which one to use is very data- and problem-specific. Although maximum-likelihood and Bayesian methods seem to be the most accurate, one might choose to use a heuristic algorithm that provides the necessary speed if thousands of sequences are analyzed. A common practice is to use several different methods and then compare the results and see if they are in agreement. Although there is always a tradeoff between accuracy and speed, future algorithm optimizations should be focused on boosting both.

References:

1. Jukes, T.H. and Cantor, C.R. Evolution of Protein Molecules. *New York: Academic Press:* 21-132 (1969).
2. Kimura, M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution* **16** (2): 111–120 (1980).
3. Felsenstein, J. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution* **17** (6): 368–376 (1981).
4. Hasegawa, M., Kishino, H., Yano, T. Dating of human-ape splitting by a molecular clock of mitochondrial DNA". *Journal of Molecular Evolution* **22** (2): 160–174 (1985).
5. Tamura, K. Estimation of the number of nucleotide substitutions when there are strong transition-transversion and G+C content biases. *Molecular Biology and Evolution* **9** (4): 678–687 (1992).
6. Tamura, K. and Nei, M. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees". *Molecular Biology and Evolution* **10** (3): 512–526 (1993).
7. Tavaré, S. Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences. *Lectures on Mathematics in the Life Sciences* (American Mathematical Society) **17**: 57–86 (1986).
8. Edwards, A.W.F. and L. L. Cavalli-Sforza, L.L. The reconstruction of evolution. *Annals of Human Genetics*, **27**:105–106 (1963).
9. Felsenstein, J. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol.* **17**(6):368-76 (1981).
10. Huelsenbeck, J.P., Ronquist, F., Nielsen, R., Bollback, J.P. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science* **294**:2310-2314 (2001).
11. Tamura, K., Stecher, G., Peterson, D., Filipski, A., Kumar, S. MEGA6: Molecular Evolutionary Genetics Analysis version 6.0. *Mol Biol Evol* **30**(12):2725-9 (2013).
12. Maddison, W.P. and Maddison, D.R. Mesquite: a modular system for evolutionary analysis. Version 2.75, http://mesquiteproject.org (2011).
13. Guindon, S., Dufayard, J.F., Lefort, V., Anisimova, M., Hordijk, W., Gascuel, O. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst Biol*. **59**(3):307-21 (2010).
14. Stamatakis, A. RaxML Version 8: A tool for phylogenetic analysis and post-analyysis of large phylogenies. *Bioinformatics*, open access (2014).
15. Hastings, W.K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika,* **57**:97–109 (1970).
16. Larget, B., and Simon, D. Markov Chain Monte Carlo Algorithms for the Bayesian Analysis of
17. Phylogenetic Trees. *Mol. Biol. Evol.* **16**, 750 (1999).

18. Ronquist, F. and Huelsenbeck, J.P. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Boinformatis Applications note* **19**(12):1572–1574 (2003).
19. Sokal, R. and Michener, C. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin* **38**: 1409–1438 (1958).
20. Saitou, N. and Nei, M. The Neighbor-Joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. and Evol.*, **4**:406–425 (1987).
21. Huson, D.H., Rupp, R., Scornavacca, C. Phylogenetic Networks: Concepts, Algorithms and Applications, *Cambridge University Press*.
22. Larkin, M.A., Blackshields, G., Brown, N.P., Chenna, R., McGettigan, P.A., McWilliam, H., Valentin, F., Wallace, I.M., Wilm, A., Lopez, R., Thompson, J.D., Gibson, T.J. and Higgins, D.G. *Bioinformatics* **23**(21): 2947-2948 (2007).
23. Ogden, T.H., and Rosenberg, M.S. Multiple sequence alignment accuracy and phylogenetic inference. *Systematic Biology* **55**(2):314–328 (2006).
24. Liu, K., Linder, C.R., Warnow, T. Multiple sequence alignment: a major challenge to large-scale phylogenetics. *PLoS Currents*: Tree of Life 2: RRN1198 (2010).
25. Price, M., Dehal, P., Arkin, A. FastTree 2 – approximately maximum-likelihood trees for large alignments. *PLoS One* **5**: e9490 (2010).
26. Liu, K., Linder, C.R., Warnow, T. RAxML and FastTree: Comparing Two Methods for Large- Scale Maximum Likelihood Phylogeny Estimation*. PLoS One,* **6**(11): e27731 (2011).