FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix

Morgan N. Price, Paramvir S. Dehal, Adam P. Arkin

Presented by

Kajori Banerjee

Fast Tree - Motivation

Input : aligned sequences

Output : Phylogenetic Tree with minimum evolution

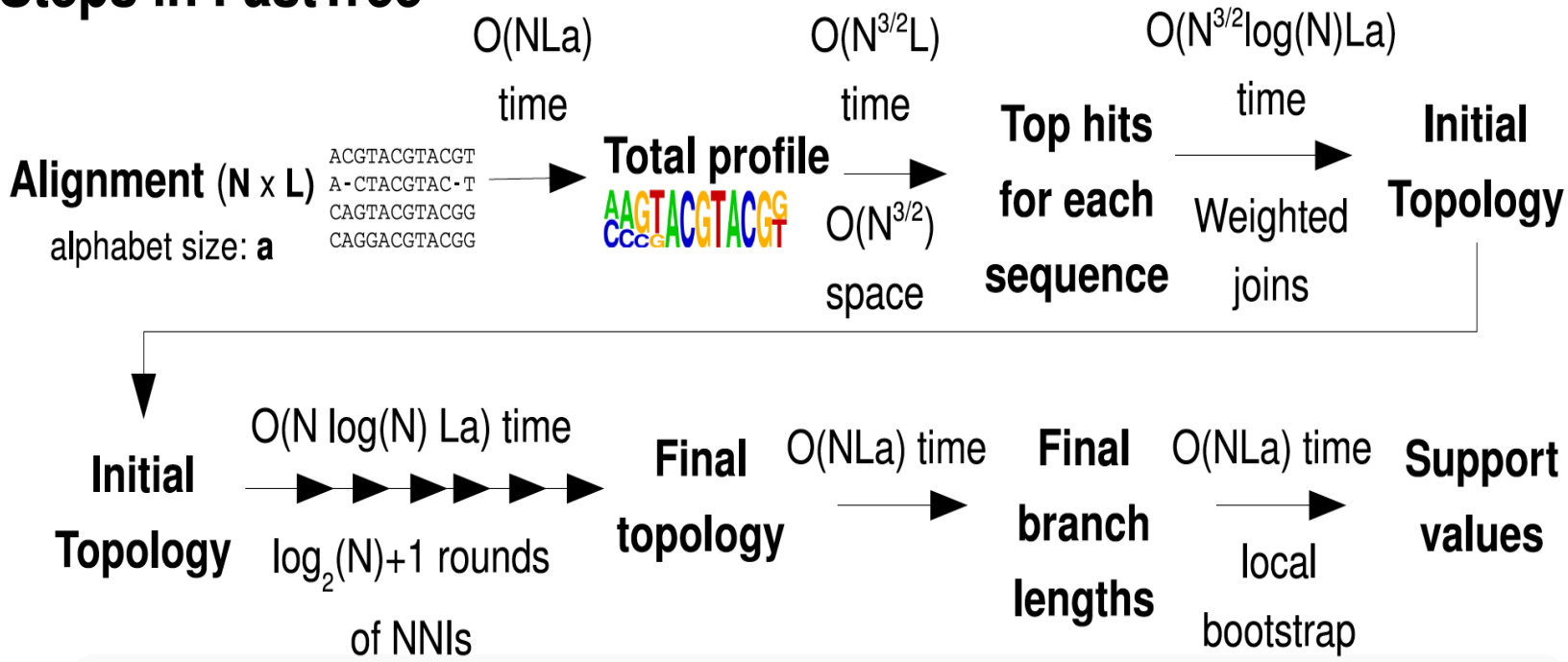
Performance

	Distance based Method	**FastTree**
space	$O(N^2)$	$O(NLa + N\sqrt{N})$
Time	$O(N^2L)$	$O(N\sqrt{N}\log(N)La)$

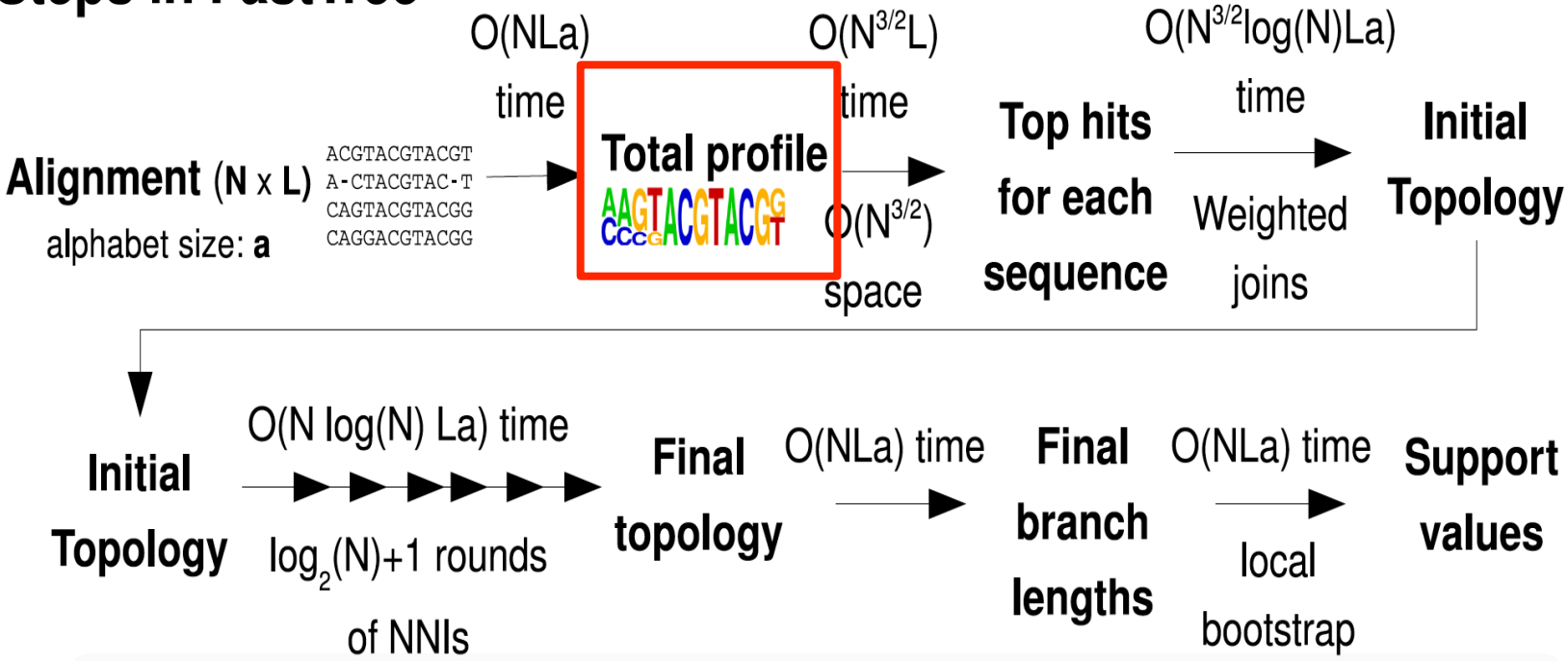
N = Number of sequences
L = Number of sites
a = number of characters

In simulations, FastTree was slightly more accurate than neighbor joining, BIONJ, or FastME.

Steps in FastTree



Steps in FastTree



Distance Computation – Total Profile

1) First, FastTree stores profiles for the internal nodes in the tree.

2) For each position, and the profile of an internal node is the weighted average of its childrens' profile

Example, if we join two leaves i and j , and i has an A at a position and j has a G , then the profile of ij at that position will be 50% A and 50% G (and 0% for other characters).

Space = **O(NLa)** instead of $O(N^2)$ space

Distances Between Profiles

- **Uncorrected distance between two profiles at position l**

$$P_l(A,B) = \sum_\alpha \sum_\beta f_{Al}(\alpha) f_{Bl}(\beta) D(\alpha,\beta)$$

- $f_{Al}(\alpha)$ = Frequencies of the characters α at position l in profile A
- D the dissimilarity matrix

- **The uncorrected distance between the two profiles**

- where w_{Al} is the $P(A,B) = \sum_l w_{Al} w_{Bl} P_l(A,B) / \left(\sum_l w_{Al} w_{Bl}\right)$ t position l

Distances Between Internal Nodes

NJ

$$d_u(AB, C) = \frac{d_u(A, C) + d_u(B, C) - d_u(A, B)}{2}.$$

FastTree

$\Delta(i,j)=\lambda P(i)-(1-\lambda)P(j)$

$$d_u(i, j) = \Delta(i, j) - u(i) - u(j),$$

$\Delta(i,j)$: is the profile distance

$u(i)$ is the "up-distance," or the average distance of the node from its children.

Calculating the Neighbor-Joining Criterion

- Minimizing criterion : $d_u(i, j) - r(i) - r(j)$

$$r(i) \equiv \sum_{k \neq i} d_u(i, k) / (n - 2)$$

$d_u(i, j)$ is the distance between nodes i and j

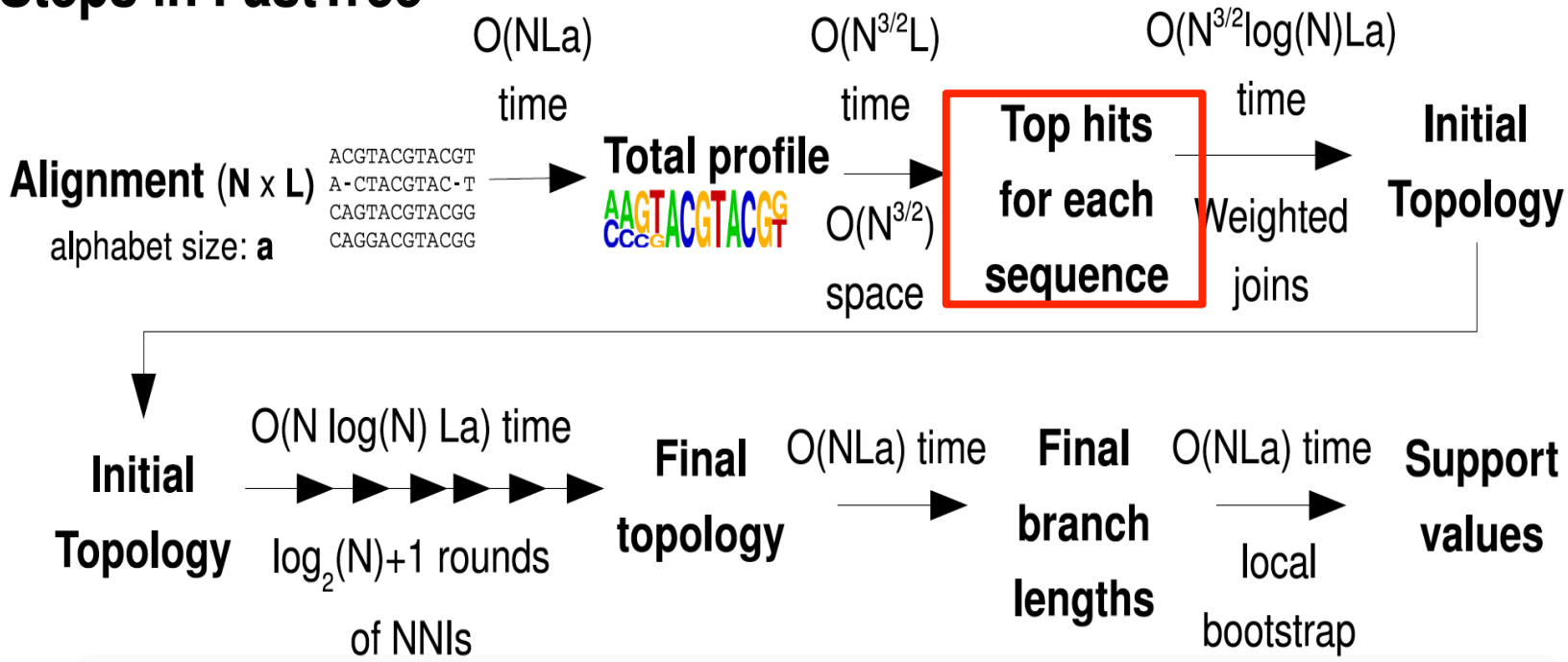
$r(i)$ - can be thought of as the average "out-distance" of i to other active nodes

Selecting the Best Join

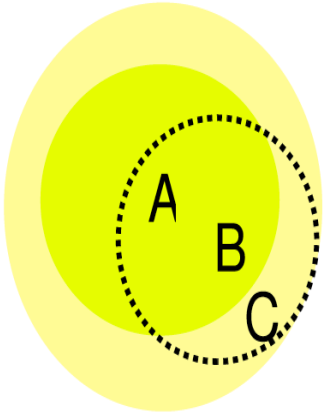
- NJ considers $O(n^2)$ joins at each step
- FastTree considers O(n) joins at each step

- **3 heuristics:**
- **1)**) "top-hits" heuristics
- 2) Before accepting a join do hill climbing to find a better join
- 3) Remember the best join for each node like FastNJ

Steps in FastTree



Top-hits Heuristic



If B is close to A, then the best join for B is also close to A:

$$\text{TopHits}(B) \subset \text{TopHits}(A) \text{ with a larger radius}$$

When we do a join:

$$\text{TopHits}(AB) \subset \text{TopHits}(A) \cup \text{TopHits}(B)$$

Top-hits

- For each node, top-hits list: the nodes that are the closest m neighbors of that node, according to the neighbor-joining criterion.

- If A and B have similar sequences, then the top-hits lists of A and B will largely overlap.

- Then, for each node B within the top m hits of A that does not already have a top-hits list, FastTree estimates the top-hits of B by comparing B to the top 2m hits of A FastTree restricts the top hits heuristic to ensure that a sequence's top hits are only inferred from the top hits of "close enough" neighbor.

Top-hits

- First, after a join, FastTree computes the top-hits list for the new node in O(mLa) time by comparing the node to all entries in the top hits lists of its children.

- Second, after a join, some of the other nodes' top hits may point to an inactive (joined) node. When FastTree encounters these entries, it replaces them with the active ancestor.

- Finally, as the algorithm progresses, the top-hits lists will gradually become shorter, as joined nodes become absent from lists. Thus, Fast-Tree periodically "refreshes" the top hit list by comparing the new node to all other nodes, and also by comparing each of the new node's top hits to each other.

Selecting the Best Join

- **3 heuristics:**
- **1)**) “top-hits” heuristics
- 2) **Before accepting a join do hill climbing to find a better join**
- 3) Remember the best join for each node like FastNJ

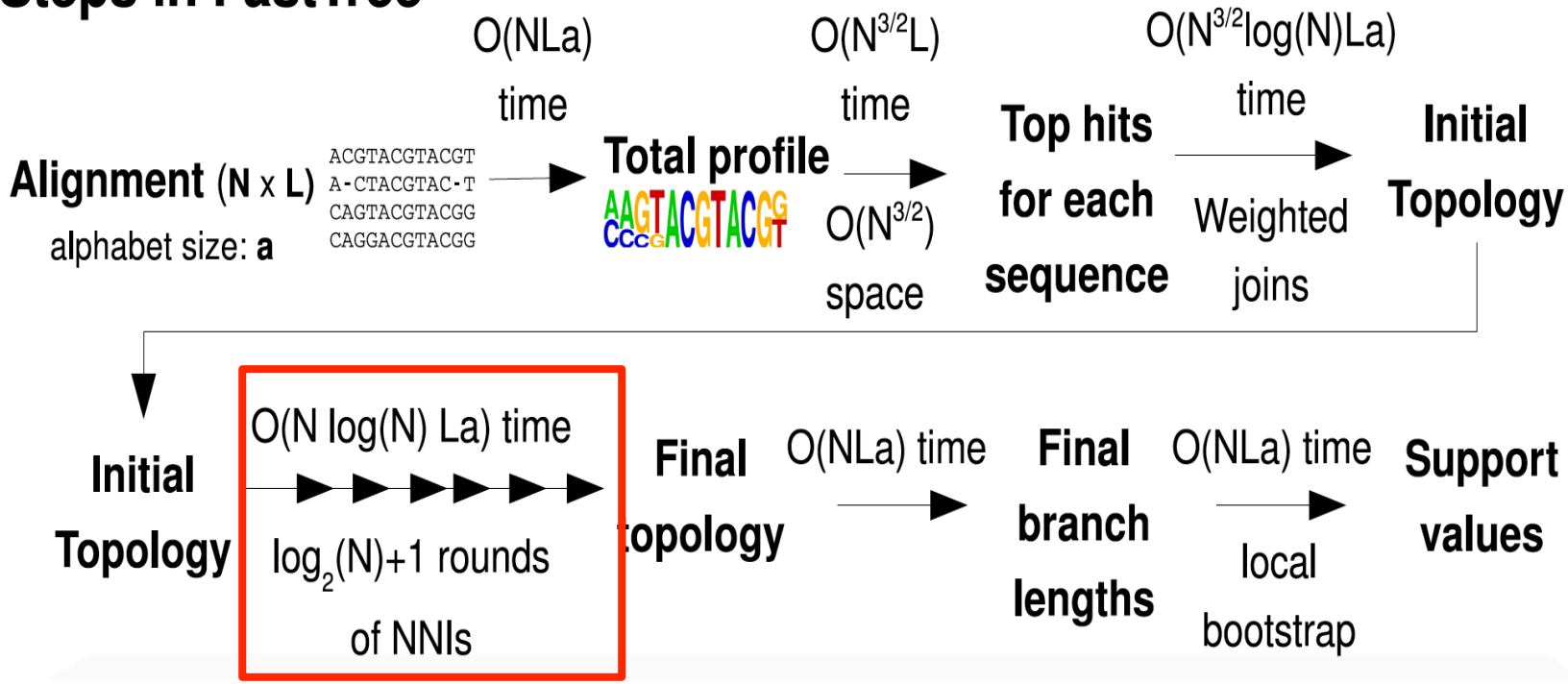
Local hill climbing

- Given a join AB, it considers all joins AC or BD, where C is in top- hits(A) or D is in top-hits(B).

- This can be beneficial because the out-distances change after every join, so the best join for a node can change as well.

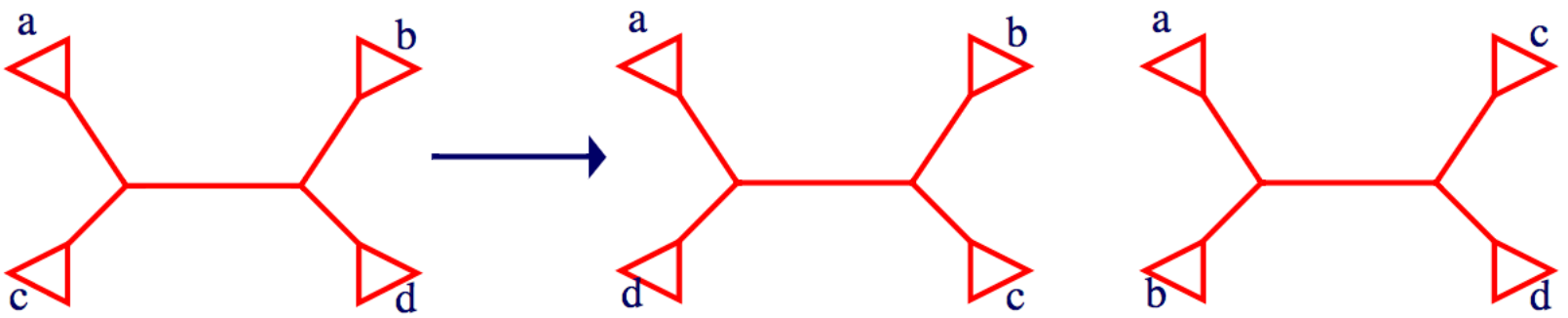
- traditional neighbor joining considers
- O(N^3) possible joins FastTree considers O(N√N logN) possible
- joins.
- heuristics require
- additional O(N√N) memory, raising the total storage requirement for FastTree
- to O(NLa + N√N), which is still much less than O(N^2).

Steps in FastTree



NNI - Nearest Neighbor Interchange.

Start with a tree and consider neighboring trees. If any neighboring tree has fewer changes, take it as current tree. Stop when no improvements



NNI

- The minimum evolution criterion prefers ((A,B), (C,D)) over alternate topologies ((A,C), (B,D)) or ((A,D), (B,C)) if

  $d(A,B)+d(C,D) < d(A,C)+d(B,D)$ and

  $d(A,B) + d(C,D) < d(A,D) + d(B,C)$.

- By default, FastTree does $\log_2(N) + 1$ rounds of NNIs.

- Chose a fixed number of rounds, instead of iterating until no more NNIs occur, to ensure fast completion.

Bootstrap

- Local Bootstrap
- To estimate the support for each split, FastTree resamples the alignment's columns with Knuth's 2002 random number generator.

- FastTree counts the fraction of resamples that support a split over the two potential NNIs around that node, much as it does while using NNIs to improve the topology.

Topological Accuracy in Simulations

Dataset

- 10 and 5,000 sequences.
- Sites : 64-1,009 (median 304)
- Gaps : 9% gaps
- on average, pairs of sequences within these alignments were 33% identical.

Table 1 - Topological accuracy of tree-building methods on simulated protein alignments with gaps.

		Topological Accuracy				
Method	**Distances**	**N=10**	**N=50**	**N=250**	**N=1,250**	**N=5,000**
PhyML	JTT	0.744^{+}	0.771^{+}	0.817^{+}	0.801^{+}	–
FastTree	log-corrected	0.724^{0}	0.763^{0}	0.797^{0}	0.778^{0}	0.763^{0}
FastME	log-corrected	0.716^{-}	0.754^{-}	0.796^{0}	0.777^{0}	0.753^{-}
BIONJ	log-corrected	0.725^{0}	0.754^{-}	0.766^{-}	0.730^{-}	0.723^{-}
BIONJ	JTT	0.701^{-}	0.758^{-}	0.777^{-}	0.737^{-}	0.731^{-}
BIONJ	JTT+Γ	0.567^{-}	0.625^{-}	0.737^{-}	0.697^{-}	–
QuickTree	log-corrected	0.716^{-}	0.746^{-}	0.760^{-}	0.726^{-}	0.716^{-}
QuickTree	%different	0.673^{-}	0.678^{-}	0.699^{-}	0.672^{-}	0.655^{-}
Clearcut	log-corrected	0.682^{-}	0.733^{-}	0.755^{-}	0.723^{-}	0.715^{-}

$^{+}$ Significantly more accurate than FastTree ($P < 0.01$, paired t test)

^{0} Not significantly different from FastTree ($P > 0.01$, paired t test)

$^{-}$ Significantly less accurate than FastTree ($P < 0.01$, paired t test)

Effectiveness of FastTree's Approximations and Heuristics

Dataset :

Alignment with 1,250 proteins and 338 positions

- neighbor-joining phase of FastTree with exhaustive search = 1,551 seconds

- neighbor-joining phase of FastTree with heuristic search = 8 seconds.

Table 2 - The topological accuracy of variants of FastTree on simulated protein alignments with gaps.

Method	Topological Accuracy		
	N=250	**N=1,250**	**N=5,000**
FastTree, Default settings	0.797	0.778	0.763
FastTree + Extra NNI (20 rounds)	0.797	0.778	0.763
FastTree's Neighbor-joining (No NNI)	0.734	0.702	0.698
FastTree, Exhaustive search, No NNI	0.733	0.701	–
BIONJ, uncorrected distances	0.731	0.699	0.694
BIONJ, log-corrected distances	0.766	0.730	0.723

Biological Dataset

- Dataset:
- alignments of 500 randomly selected sequences from large COGs.
- Positions : 65 to 1,009 positions,
- each alignment, the average pair of sequences were 27% identical.

Table 2 - The topological accuracy of variants of FastTree on simulated protein alignments with gaps.

Method	Topological Accuracy		
	N=250	**N=1,250**	**N=5,000**
FastTree, Default settings	0.797	0.778	0.763
FastTree + Extra NNI (20 rounds)	0.797	0.778	0.763
FastTree's Neighbor-joining (No NNI)	0.734	0.702	0.698
FastTree, Exhaustive search, No NNI	0.733	0.701	–
BIONJ, uncorrected distances	0.731	0.699	0.694
BIONJ, log-corrected distances	0.766	0.730	0.723

Table 2 - The topological accuracy of variants of FastTree on simulated protein alignments with gaps.

Method	Topological Accuracy		
	N=250	**N=1,250**	**N=5,000**
FastTree, Default settings	0.797	0.778	0.763
FastTree + Extra NNI (20 rounds)	0.797	0.778	0.763
FastTree's Neighbor-joining (No NNI)	0.734	0.702	0.698
FastTree, Exhaustive search, No NNI	0.733	0.701	–
BIONJ, uncorrected distances	0.731	0.699	0.694
BIONJ, log-corrected distances	0.766	0.730	0.723

Questions