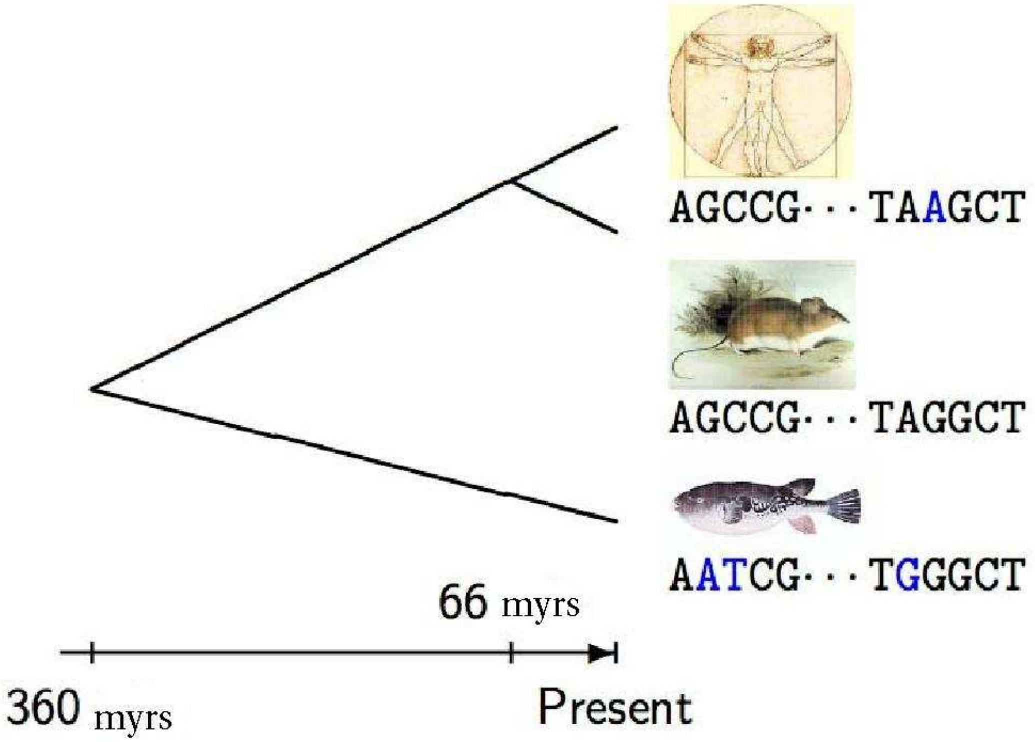
On the Optimality of the Neighbor Joining Algorithm

Ruriko Yoshida
Dept. of Statistics University of Kentucky

Joint work with K. Eickmeyer, P. Huggins, and L. Pachter

Phylogeny

Phylogenetic trees describe the evolutionary relations among groups of organisms.



Why we care?

- We can analyze changes that have occurred in evolution of different species.

- Phylogenetic relations among different species help predict which species might have similar functions.

- We can predict changes occurring in rapidly changing species, such as HIV virus.

Constructing trees from sequence data

“Ten years ago most biologists would have agreed that all organisms evolved from a single ancestral cell that lived 3.5 billion or more years ago. More recent results, however, indicate that this family tree of life is far more complicated than was believed and may not have had a single root at all.” (W. Ford Doolittle, (June 2000) *Scientific American*).

Since the proliferation of Darwinian evolutionary biology, many scientists have sought a coherent explanation from the evolution of life and have tried to reconstruct phylogenetic trees.

Methods to reconstruct a phylogenetic tree from DNA sequences include:

- **The maximum likelihood estimation (MLE) methods**: These describe evolution in terms of a discrete-state continuous-time Markov process. The substitution rate matrix can be estimated using the **expectation maximization (EM) algorithm**. (for eg. Dempster, Laird, and Rubin (1977), Felsenstein (1981)).

- **The Balanced Minimum Evolution (BME) method**: This is a **distance based method** and weighted Least Square method (the principle of Least Squares is a general method for estimating unknown parameters values so that error is minimized). It finds a closest additive metric from the given non-additive distance matrix with the smallest branch lengths (more biologically makes sense).

However

The MLE methods: An exhaustive search for the ML phylogenetic tree is computationally prohibitive for large data sets.

The BME method: This is an NP hard algorithm in terms of the number of taxa (Farach, Kannan, Warnow (1996), Rzhetsky and Nei (1993), Desper and Gascuel (2004)).

But there is a polynomial time algorithm to estimate the BME tree.

Neighbor-joining (NJ) method: This is the most popular distance based method. It computes a tree from all pair-wise distances obtained easily. (Saito and Nei (1987), Studier and Keppler (1988)).

Fact: The NJ algorithm is a greedy algorithm to find the BME tree (Gascuel and Steel (2006)).

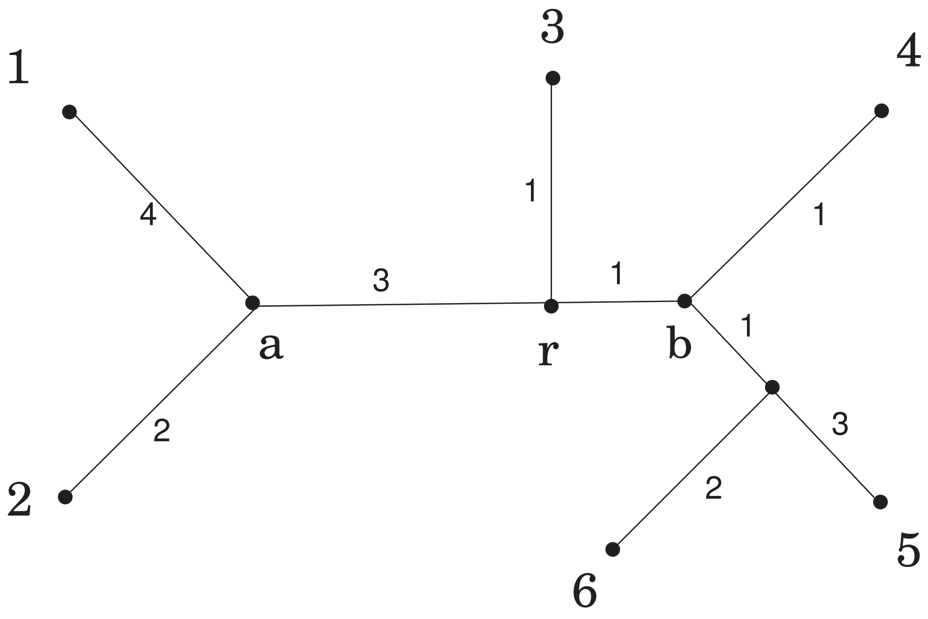
From this point of view, NJ is "optimal" whenever the NJ algorithm outputs the tree which minimizes the BME criterion.

Goal: We want to study the optimality of the NJ algorithm, (i.e., want to study how often the NJ returns the BME tree).

Plan: I will discuss NJ method today closely and then I will discuss BME method in the next lecture.

Distance Matrix

A **distance matrix** for a tree T is a matrix D whose entry D_{ij} stands for the mutation distance between i and j.



Distance Matrix

	1	2	3	4	5	6
1	0	6	8	9	12	11
2	6	0	6	7	10	9
3	8	6	0	3	6	5
4	9	7	3	0	5	4
5	12	10	6	5	0	5
6	11	9	5	4	5	0

Table 1: Distance matrix D for the example.

Definitions

Def. A distance matrix D is a **metric** iff D satisfies:

- Symmetric: $D_{ij} = D_{ji}$ and $D_{ii} = 0$.
- Triangle Inequality: $D_{ik} + D_{jk} \geq D_{ij}$.

Def. D is an **additive metric** iff there exists a tree T s.t.

- Every edge has a positive weight and every leaf is labeled by a distinct species in the given set.
- For every pair of i, j, $D_{ij} =$ the sum of the edge weights along the path from i to j.

Also we call such T an **additive tree**.

The NJ method

Def. We call a pair of two distinct leaves $\{i, j\}$ a **cherry** if there is exactly one intermediate node on the unique path between i and j.

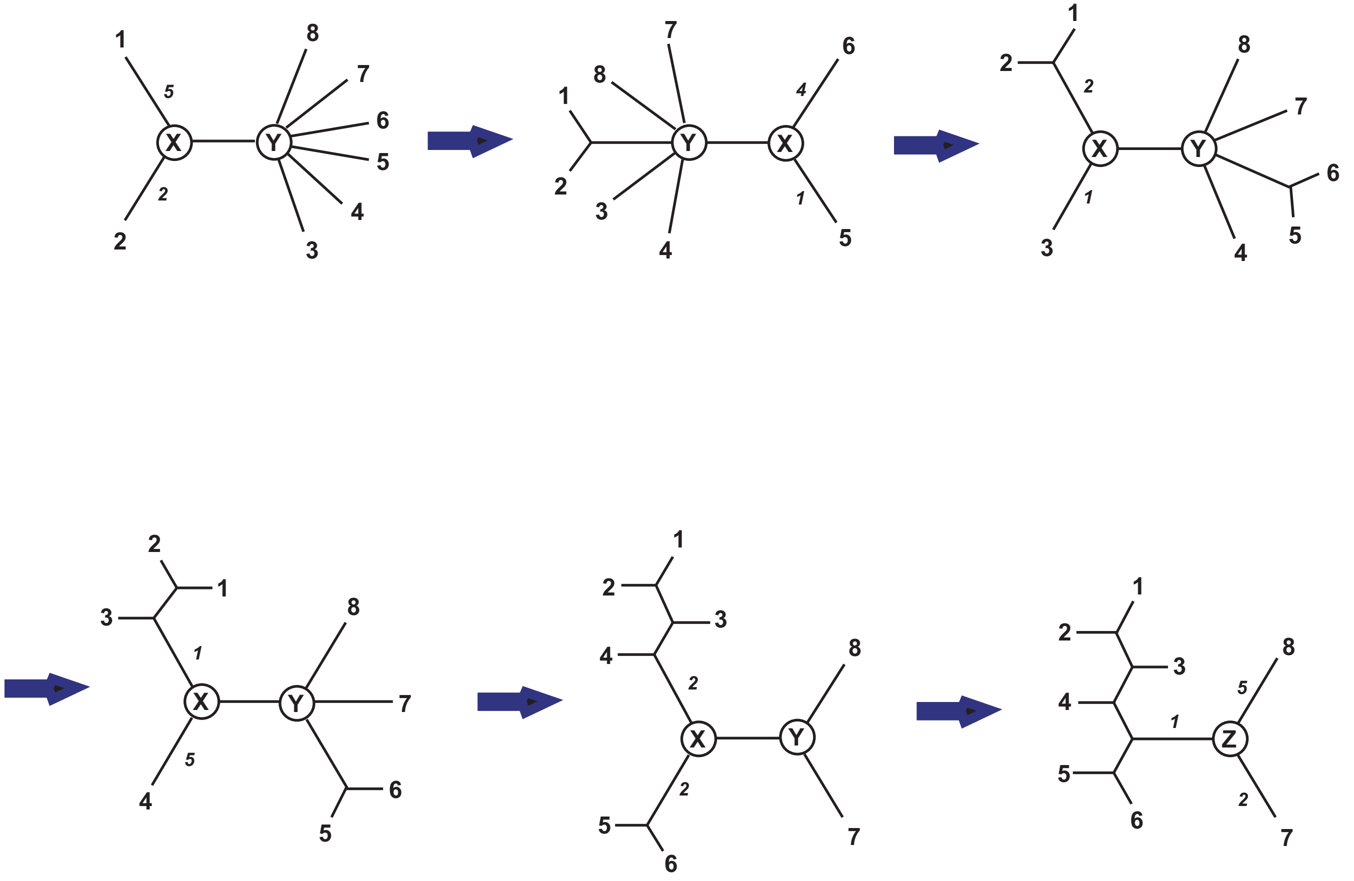
Thm. (Q-criterion) [Saitou-Nei, 1987 and Studier-Keppler, 1988]

Let $Q \in \mathbb{R}^{n\times n}$ such that $Q_{ij} = (n-2)D_{ij} - (r_i + r_j)$, where $r_i := \sum_{k=1}^{n} D_{ik}$. $\{i^*, j^*\}$ is a cherry in T if $Q_{i^*j^*}$ is a minimum for all $i \neq j$.

Neighbor Joining Method:

Input. A tree matric D. **Output.** An additive tree T.
Idea. Initialize a star-like tree. Then find a cherry $\{i, j\}$ and compute branch length from the interior node x to i and from x to j. Repeat this process recursively until we find all cherries.

Neighbor Joining Method



The NJ is consistent, i.e., it returns the additive tree if the input distance matrix is tree metric.

However, we usually estimate all pairwise distances via MLE. Usually these distance matrices are not tree metric.

The NJ returns a tree topology which induces a tree metric that is hopefully close to the input.

Question: For which distance matrices will the NJ return a particular tree topology?

We look at the algorithm closely.....

Q-criterion

Go back to the previous example....

$Q =$

0	-20	-16	-15	-16	-11
-20	0	-14	-13	-14	-9
-16	-14	0	-9	-10	-5
-15	-13	-9	0	-9	-4
-16	-14	-10	-9	0	-5
-11	-9	-5	-4	-5	0

The Q-criterion

For $n = 4$ by symmetry we have

$$\begin{array}{rcl} Q_{12} & = & -D_{13} - D_{14} - D_{23} - D_{24} \\ Q_{13} & = & -D_{12} - D_{14} - D_{23} - D_{34} \\ Q_{23} & = & -D_{12} - D_{13} - D_{24} - D_{34} \\ Q_{14} & = & -D_{12} - D_{13} - D_{24} - D_{34} \\ Q_{24} & = & -D_{12} - D_{14} - D_{23} - D_{34} \\ Q_{34} & = & -D_{13} - D_{14} - D_{23} - D_{24} \end{array}$$

$$\begin{pmatrix} Q_{12} \\ Q_{13} \\ Q_{23} \\ Q_{14} \\ Q_{24} \\ Q_{34} \end{pmatrix} = \begin{pmatrix} 0 & -1 & -1 & -1 & -1 & 0 \\ -1 & 0 & -1 & -1 & 0 & -1 \\ -1 & -1 & 0 & 0 & -1 & -1 \\ -1 & -1 & 0 & 0 & -1 & -1 \\ -1 & 0 & -1 & -1 & 0 & -1 \\ 0 & -1 & -1 & -1 & -1 & 0 \end{pmatrix} \begin{pmatrix} D_{12} \\ D_{13} \\ D_{23} \\ D_{14} \\ D_{24} \\ D_{34} \end{pmatrix}.$$

In general...

Let $m = \binom{n}{2}$. Let $\mathbf{d} \in \mathbb{R}^m$ be a vector representation of D and $\mathbf{q} \in \mathbb{R}^m$ be a vector representation of Q. The Q-criterion is obtained from the input data by a linear transformation:

$$\mathbf{q} = \mathbf{A^{(n)}d},$$

where the entries of the matrix $A^{(n)}$ are given by

$$\mathbf{A^{(n)}_{ab}} = \mathbf{A^{(n)}_{ij,kl}} = \begin{cases} n-4 & \text{if } a = b, \\ -1 & \text{if } a \neq b \text{ and } \{i,j\} \cap \{k,l\} \neq \emptyset, \\ 0 & \text{else}, \end{cases} .$$

The Q-criterion:

$$\text{find smallest } \mathbf{q_a} \text{ for } \mathbf{a = 1, \cdots, m} \text{ such that } \mathbf{q = A^{(n)}d}.$$

The first step in cherry picking

Note that the Q-criterion is a linear programming problem:

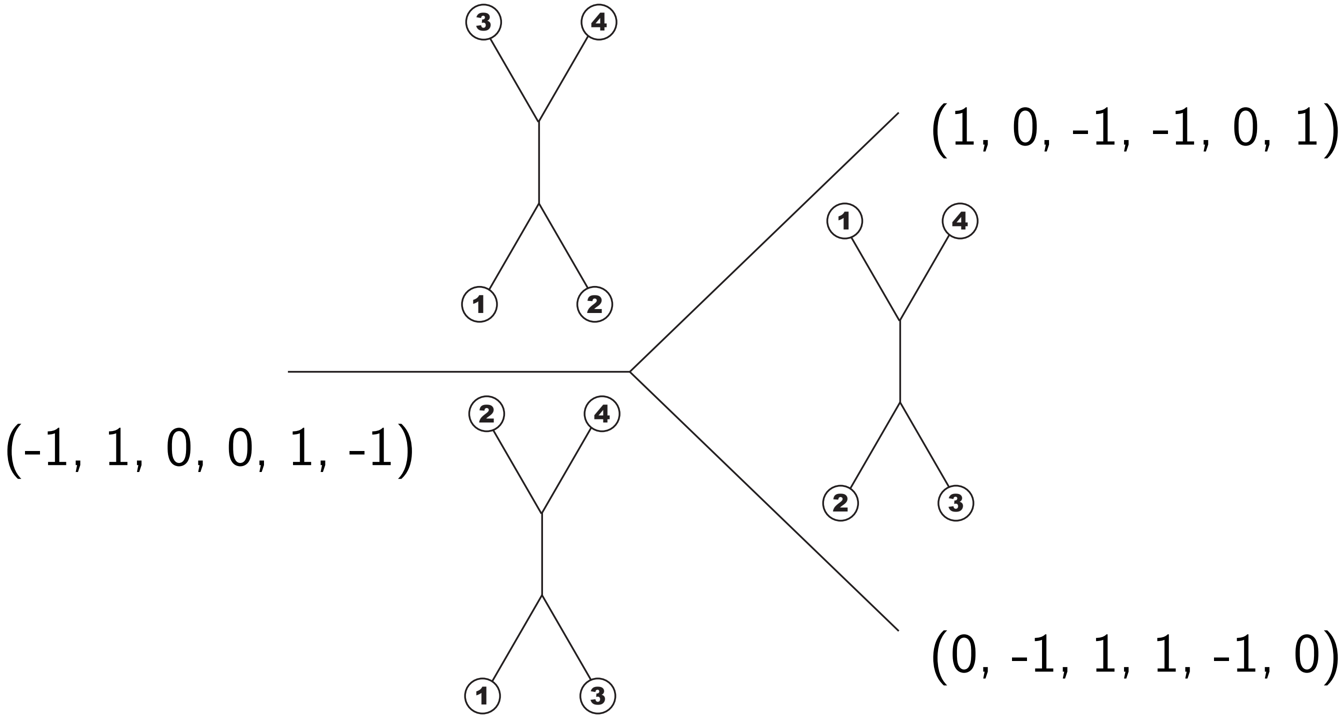
$$\min \mathbf{d} \cdot \mathbf{x} \text{ such that } \mathbf{x} \in \text{conv}\{\mathbf{Ae_1}, \dots, \mathbf{Ae_m}\}.$$

Therefore the set cd_i of all *parameter* vectors $\mathbf{d}$ for which the NJ algorithm will select cherry i in the first step is the normal cone at Ae_i to the polytope

$$\mathbf{P_n^{nj}} := \mathrm{conv}\{\mathbf{Ae_1}, \dots, \mathbf{Ae_m}\}.$$

The **shifting lemma** implies that the affine dimension of the polytope P_n^{nj} is at most $m - n$.

Example for $n = 4$



Reducing the number of taxa

Suppose out of our n taxa $\{1, \ldots, n\}$, the first cherry to be picked is the $\binom{n}{2}$th cherry $\{n-1, n\}$, which we view as the new node number $n-1$.

The reduced pairwise distance matrix is one row and one column shorter than the original one. Explicitly,

$$\mathbf{d'_i} = \begin{cases} d_i & \text{for } 1 \leq i \leq \binom{n-2}{2} \\ \frac{1}{2}(d_i + d_{i+(n-2)} - d_m) & \text{for } \binom{n-2}{2} + 1 \leq i \leq \binom{n-1}{2} \end{cases}$$

We see that the reduced distance matrix depends linearly on the original one:

$$\mathbf{d}' = \mathbf{R}\mathbf{d},$$

with $R = (r_{ij}) \in \mathbb{R}^{(m-n+1)\times m}$, where

$$r_{ij} = \begin{cases} 1 & \text{for } 1 \leq i = j \leq \binom{n-2}{2} \\ 1/2 & \text{for } \binom{n-2}{2} + 1 \leq i \leq \binom{n-1}{2}, j = i \\ 1/2 & \text{for } \binom{n-2}{2} + 1 \leq i \leq \binom{n-1}{2}, j = i + n - 1 \\ -1/2 & \text{for } \binom{n-2}{2} + 1 \leq i \leq \binom{n-1}{2}, j = m \\ 0 & \text{else} \end{cases}$$

The process of picking cherries is repeated until there are only three taxa left, which are then joined to a single new node.

Note: Each tree topology is determined by a polyhedral cone (i.e., we add more constraints to the normal cone at a vertex of P_n^{nj}). We call these cones **NJ cones**.

The cone $C_{45,3}$

Since we can apply a permutation $\sigma \in S_5$ on taxa, without loss of generality, we suppose that the first cherry to be picked is the cherry with leaves 4 and 5. This is true for all input vectors $\mathbf{d}$ which satisfy

$$(\mathbf{h_{10,i}}, \mathbf{d}) \geq \mathbf{0} \text{ for } \mathbf{i} = \mathbf{1}, \ldots, \mathbf{9},$$

where the vector

$$\mathbf{h_{ij}^{(n)}} := -\mathbf{A}^{(\mathbf{n})}(\mathbf{e_i} - \mathbf{e_j}).$$

Then, the set of all input vectors $\mathbf{d}$ for which the first picked cherry is 4-5 and the second one is 1-2:

$$C_{45,3} :=$$
$$\{\mathbf{d} \mid (\mathbf{h}_{10,i}, \mathbf{d}) \geq 0 \text{ for } i = 1, \ldots, 9, \text{ and } (\mathbf{r}_1 - \mathbf{r}_2, \mathbf{d}) \geq 0, (\mathbf{r}_1 - \mathbf{r}_3, \mathbf{d}) \geq 0\}$$

where $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ are the first three rows of $-A^{(4)}R^{(5)}$.

The NJ cones

For $n = 5$, there is only one unlabeled tree and there are 15 labeled trees. There are 30 cones in the 5-dimension (i.e. there are two cones per a lebeled tree).

- They do not form a fan.
- The union of cones $C_{12,3}$ and $C_{45,3}$ does not form a convex body (i.e. the union of two cones for one tree topology does not form a convex cone).

Example

$$D_1 = \begin{pmatrix} 0 & 0.056244 & 0.168744 & 0.506257 & 0.056256 \\ 0.056244 & 0 & 0.168755 & 0.056256 & 0.506245 \\ 0.168744 & 0.168755 & 0 & 0.056244 & 0.056256 \\ 0.506257 & 0.056256 & 0.056244 & 0 & 0.168744 \\ 0.056256 & 0.506245 & 0.056256 & 0.168744 & 0 \end{pmatrix} \text{ and}$$

$$D_2 = \begin{pmatrix} 0 & 0.168694 & 0.056194 & 0.506306 & 0.112556 \\ 0.168694 & 0 & 0.056307 & 0.056307 & 0.562445 \\ 0.056194 & 0.056307 & 0 & 0.168694 & 0.225056 \\ 0.506306 & 0.056307 & 0.168694 & 0 & 0.112444 \\ 0.112556 & 0.562445 & 0.225056 & 0.112444 & 0 \end{pmatrix}.$$

NJ algorithm returns the tree in Figure 1 from D_1 and D_2.

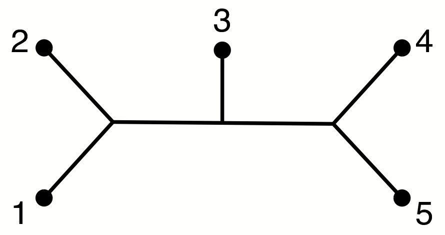


Figure 1: A tree with five leaves.

However, the NJ returns a different tree topology with $(D_1 + D_2)/2$.

For $n = 6$

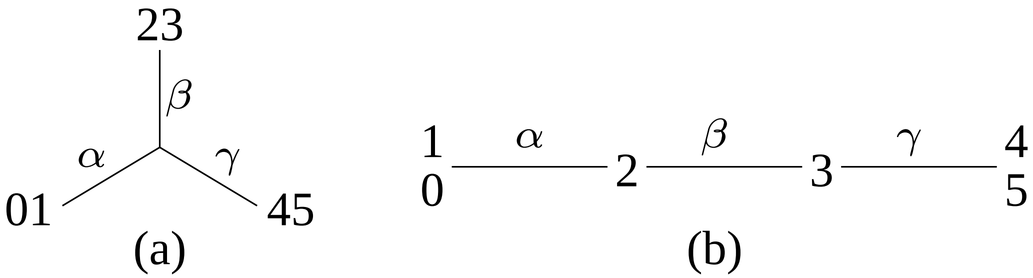


Figure 2: The two possible topologies for trees with six leaves, with edges connecting to leaves shrunk to zero.

There are three different classes of cones which cannot be mapped onto each other by the group action, C_I, C_{II}, C_{III}.

- **Type I**: $a, b, c, d, e, f \to a, b, c, d, (ef) \to a, b, (cd), (ef), \to$ Fig. 2(a)

- **Type II**: $a, b, c, d, e, f \to a, b, c, d, (ef) \to a, b, (cd), (ef)$ $\to cd - a - b - ef$ (like Fig 2(b), but different labels)

- **Type III**: $a, b, c, d, e, f, \to a, b, c, d, (ef) \to a, b, c, (d(ef))$ $\to ab - c - d - ef$ (exactly as in Fig 2(b))

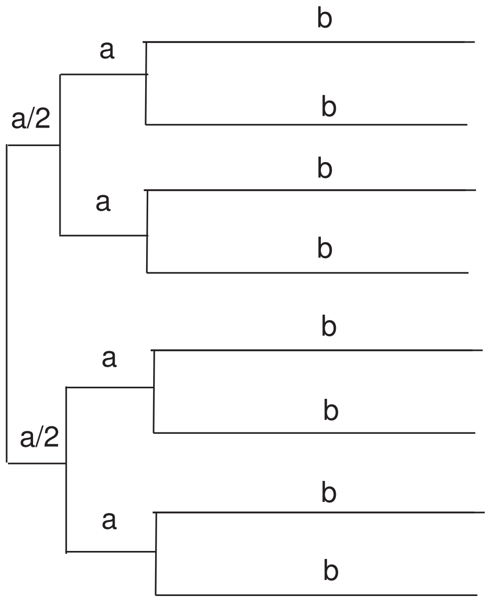
	C_{I}	C_{II}	C_{III}
stabilizer	$\langle(12),(34),(56)\rangle$	$\langle(12),(56)\rangle$	$\langle(12),(56)\rangle$
size of stabilizer	8	4	4
number of cones	90	180	180
cones giving same labeled topology	6	2	2

Simulation Results

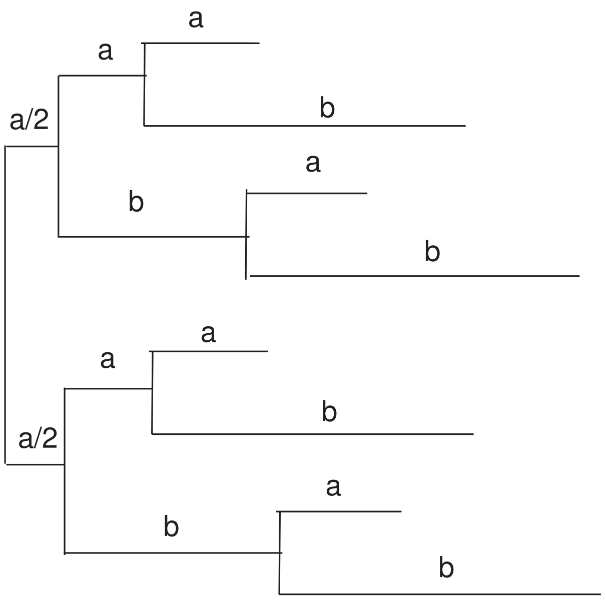
With the Juke Cantor and Kimura 2 parameter models.

Consider two tree models...

Modeled from Strimmer and von Haeseler.



T1



T2

We generate $10,000$ replications at the edge length ratio, a/b $=$ 0.03/0.42 for sequences of length 500BP with the Jukes-Cantor and Kimura 2 parameter models via a software `evolver` from `PAML` package.

For each set of 5 sequences, we compute first pairwise distances via the heuristic MLE method using a software `fastDNAml`. To compute cones, we used `MAPLE` and `polymake`.

We say an input vector (distance matrix) is **correctly classified** if the vector locates in one of the cones where the vector representation of the tree metric (noiseless input) lies. We say an input vector is **incorrectly classified** if the vector locates in the complement of the cones where the vector representation of the tree metric lies.

For distance matrices which are correctly classified by the NJ algorithm, we compute the minimum distance to any cone giving a different tree topology.

Distances of correctly classified vectors from closest misclassified vector

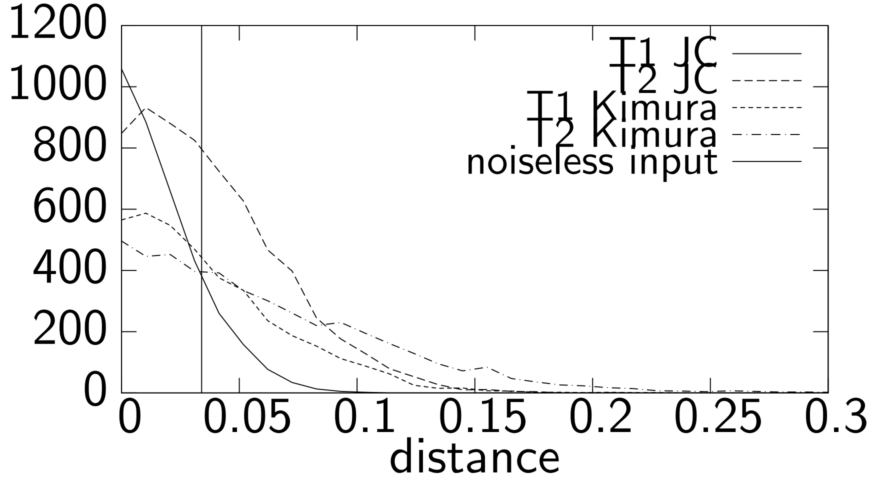


Figure 3: Distances of correctly classified input vectors from the closest correctly classified vector.

Mean and variance of the distances of correctly classified vectors from the nearest misclassified vector.

	JC		**Kimura2**	
	T1	T2	T1	T2
# of cases	3,581	6,441	3,795	4,467
Mean	0.0221	0.0421	0.0415	0.0629
Variance	$2.996 \cdot 10^{-4}$	$9.032 \cdot 10^{-4}$	$1.034 \cdot 10^{-3}$	$2.471 \cdot 10^{-3}$

For input vectors to which the NJ algorithm answers with a tree topology different from the correct tree topology, we compute the distances to the two cones for which the correct answer is given and take the minimum of the two. The bigger this distance is, the further we are off.

Distances of misclassified input vectors from closest correctly classified vector

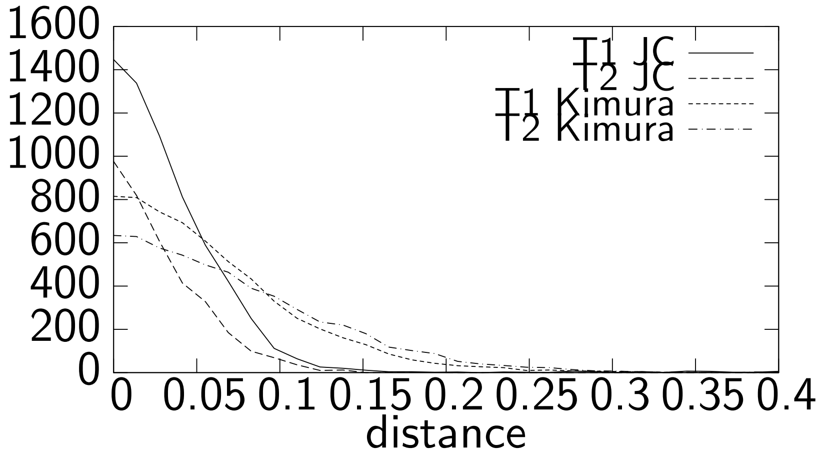


Figure 4: Distances of correctly incorrectly classified input vectors from the closest incorrectly classified vector.

Mean and variance of the distances of misclassified vectors to the nearest correctly classified vector.

	JC		**Kimura2**	
	T1	T2	T1	T2
# of cases	6,419	3,559	6,205	5,533
Mean	0.0594	0.0331	0.0951	0.0761
Variance	0.0203	$7.39 \cdot 10^{-4}$	0.0411	$3.481 \cdot 10^{-3}$

Thank you....