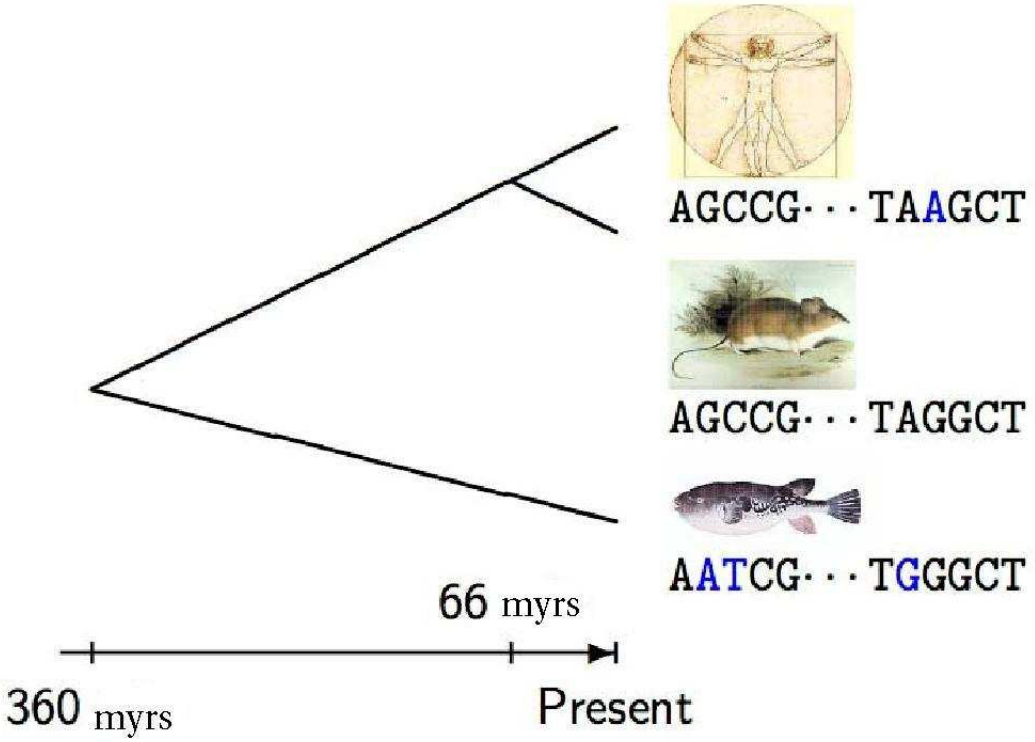
On the Optimality of the Neighbor Joining Algorithm

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.



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)).

RECALL

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

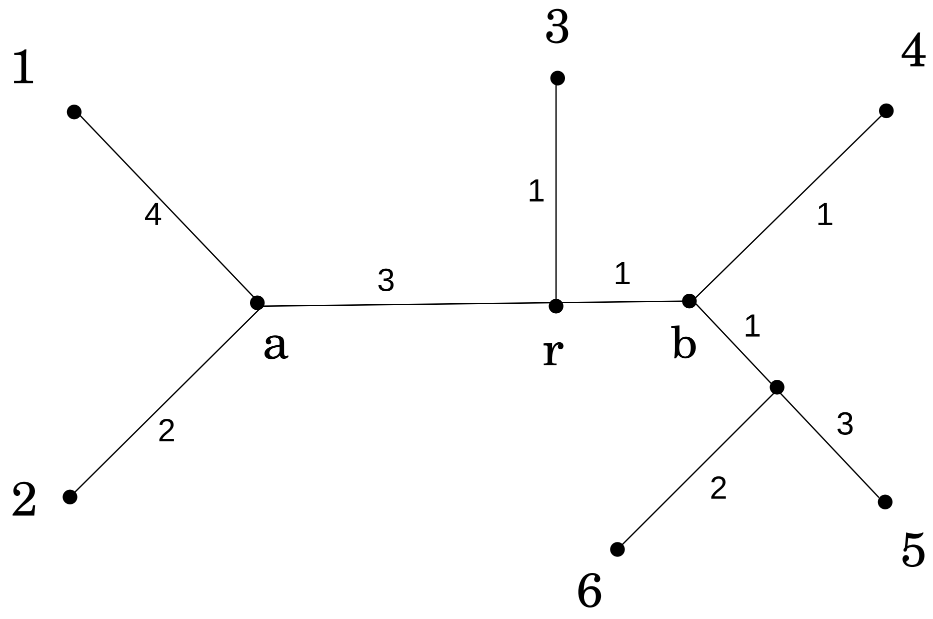
From this point of view, NJ is “optimal” whenever the NJ algorithm outputs the tree which minimizes the BME criterion.

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: We will discuss the BME method closely today.

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.

NJ method

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 $i \neq j$.

$$Q = \begin{array}{|c|c|c|c|c|c|}\hline 0 & -20 & -16 & -15 & -16 & -11 \\ \hline -20 & 0 & -14 & -13 & -14 & -9 \\ \hline -16 & -14 & 0 & -9 & -10 & -5 \\ \hline -15 & -13 & -9 & 0 & -9 & -4 \\ \hline -16 & -14 & -10 & -9 & 0 & -5 \\ \hline -11 & -9 & -5 & -4 & -5 & 0 \\ \hline \end{array}$$

Thus we pick $\{1, 2\}$.

Computing edge lengths and removing a node

We use the formula: Let $\{i, j\}$ be a cherry and x is the unique interior node on the path.

$$D(i,x) = \frac{1}{2 * (n-2)} \left[(n-2)D(i,j) + \sum_{k \neq i,j} D(i,k) - \sum_{k \neq i,j} D(j,k) \right]$$

and

$$D(j,x) = D(i,j) - D(i,x).$$

Distance from x to another node k:

$$D(x,k) = \frac{1}{2}[D(i,k) + D(j,k) - D(i,j)].$$

Balanced Minimum Evolution

The BME is also a distance based method.

This is a weighted LS method to find the closest tree metric such that the total branch lengths of the tree is the smallest.

It is based on Pauplin's formula, $\Delta_D(\tau)$, which estimates the total length of a tree, based on: [Pauplin 2000 J Mol Evol 51]

(1) its topology τ,

(2) an estimated distance matrix $D = (D_{ij})$.

The BME is to find τ such that

$$\min_{\tau_t,\, t=1,\cdots(2n-5)!!} \Delta_D(\tau_t).$$

Pauplin's formula

Pauplin's formula is defined as:

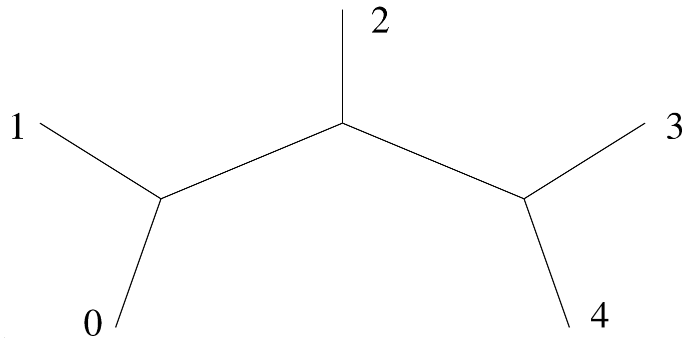
$$\Delta_D(\tau) = \sum_{i<j} W_{ij}(\tau) D_{ij},$$

where

$$W_{ij}(\tau) = (2)^{(1 - \#\text{ of branches between } i \text{ and } j)}$$

for a particular tree topology τ.

Example



For the tree topology above, we have

$$W(\tau) = (1/2, 1/4, 1/4, 1/8, 1/8, 1/4, 1/8, 1/8, 1/4, 1/2).$$

Note that Pauplin's formula can be seen as a linear programming such that

$$\min_{x \in P_n^{ME}} \mathbf{d} \cdot x$$

such that

$$P_n^{ME} = \mathrm{conv}\{W_{\tau_1}, \cdots, W_{\tau_{(2n-5)!!}}\}.$$

We call P_n^{ME} a **BME polytope**.

Thus, the set of all $\mathbf{d}$ such that the topology τ_t is minimal is the normal cone at a vertex W_{τ_t}. We call this cone **BME cone** for a topology τ_t.

For $n = 4$, the BME polytope is a triangle in 6 dimensional space.

Combinatorics of the BME polytopes

For up to $n = 7$ taxa, we computed BME polytopes and studied their structure.

n	dimension of BME polytope	f-vector
4	2	(3,3)
5	5	(15, 105, 250, 210, 52)
6	9	(105, 5460, ?, ?, ?, 90262)
7	14	(945, 445410, ?, ?, ?, ?, ?)

For $n = 5, 6$, the number of edges is $\binom{n}{2}$, so all pairs of bifurcating tree topologies τ_1, τ_2 on $n \leq 6$ taxa can be cooptimal for BME, which we found surprising.

But for $n = 7$, there is one combinatorial type of non-edge.

The non-edges of the BME polytope for $n = 7$ taxa

The non-edges are all between pairs of 7-leaf 3-cherry trees which are related by the depicted pair of modifications.

Note there are two ways to perform each modification, so there are $2 \cdot 2 = 4$ non-edges for each such tree. Recall there are 945 bifurcating trees though, so the vast majority of trees on the $n = 7$ BME polytope are connected by edges.

Edges and non-edges of the BME polytope

- We still do not understand which pairs of trees will form edges on the BME polytope.

- If we did understand the edges, then we might be able to devise a competitive alternative to **FastME** that improves trees by walking along edges on the BME polytope, rather than performing nearest-neighbor interchange (NNI) moves.

- Edge-walking is called the **simplex algorithm** in linear programming, and it works very well in practice.

Balanced minimum evolution cones

- For each bifurcating tree topology τ, the **BME cone** of τ is the set of all choices of pairwise distances $D = (d_{ij})$ for which τ minimizes the dot-product $D \cdot W_\tau$.

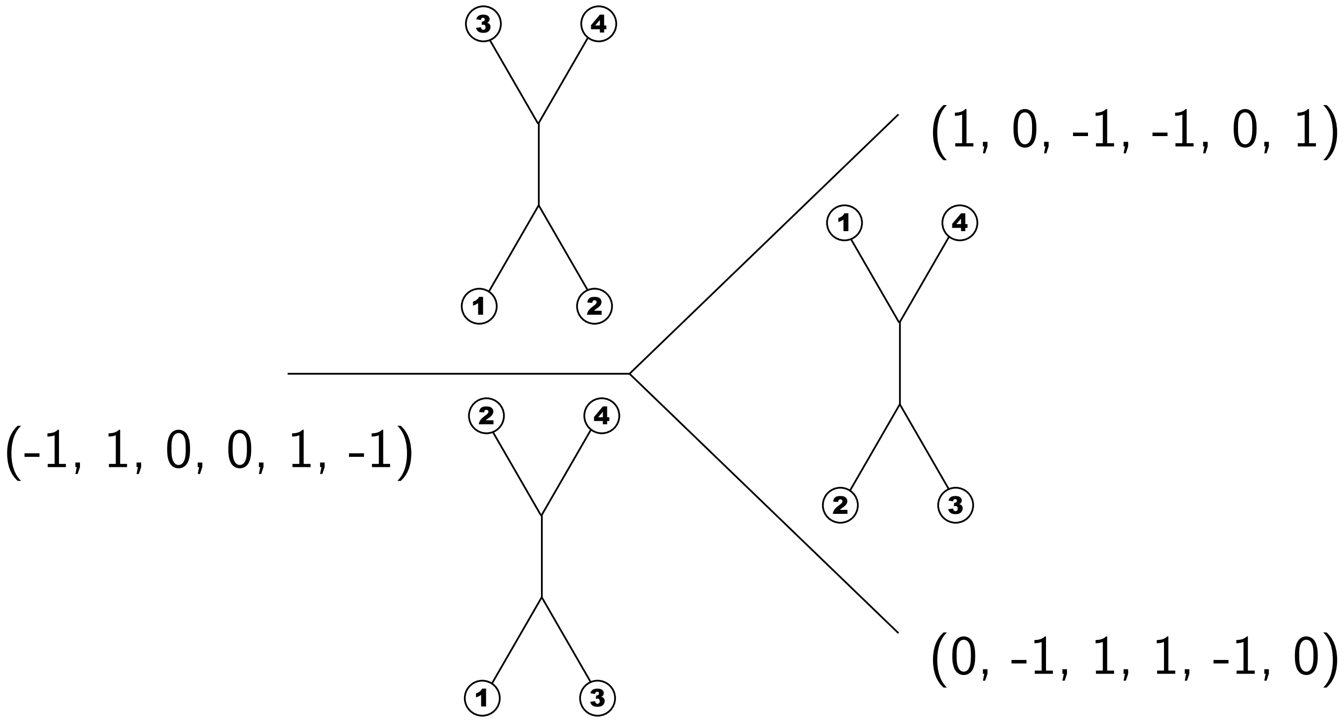
- The edges of the BME polytope emanating from the vertex W_τ determine the facets (flat sides) of the BME cone of τ. The facets of the BME polytope that contain W_τ determine the extreme rays of the BME cone of τ. (This is a perfect example of **duality.**)

- BME cones are convex. (This is a **normal fan**).

- **Thus the BME method (unlike neighbor joining) is convex**: If the BME method outputs tree topology τ for two inputs D, D', then BME will also output τ on the input $(D + D')/2$.

Example

For $n = 4$, each NJ cone equals to one of the BME cone.



(1, 0, -1, -1, 0, 1)

(-1, 1, 0, 0, 1, -1)

(0, -1, 1, 1, -1, 0)

Comparing NJ and BME cones

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.

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

Neighbor joining: Fast and consistent

- Neighbor joining is a popular algorithm for phylogenetic reconstruction.

- Input is pairwise distances $D = (d_{ij})$, presumed to arise as a perturbed tree metric. Output is a tree topology which induces a tree metric that is hopefully close to D.

- Intuition: Find two nodes which are 'close,' and join them as siblings (a 'cherry') in the tree.

- Actually neighbor joining joins nodes a, b which have minimal Q-value:

$$Q_{ab} = d_{ab} - \frac{1}{n-2}(\sum_k d_{ak} + d_{bk})$$

Neighbor joining: Fast and consistent

- Nodes a, b are then replaced by a single new node z which is the root of the cherry (a, b), and distances d_{zk} are defined as $d_{zk} = d_{ak} + d_{bk} - 2d_{ab}$. Then neighbor joining is applied recursively on the remaining nodes, until a tree is obtained.

- Using Q-values instead of the original distances compensates for short internal edges.

- In fact neighbor joining based on Q-values is consistent: Given a tree metric $D = D_T$ as input, NJ will correctly output tree T.

Neighbor joining cones

- Notice that all Q-values are linear in the distances, so picking a cherry (a, b) in the tree means that the distances satisfy linear inequalities:

$$d_{ab} - \frac{1}{n-2}(\sum_k d_{ak} + d_{bk}) \;\leq\; d_{ij} - \frac{1}{n-2}(\sum_k d_{ik} + d_{jk}), \;\; \forall i, j$$

- Also, after picking cherry (a, b) and replacing it with a new node z, the new distances d_{zk} are linear in the old distances: $d_{zk} = d_{ak} + d_{bk} - 2d_{ab}$.

- Thus NJ will output a particular tree topology T, and pick cherries in a particular order, iff the original distances d_{ij} satisfy certain linear inequalities. The inequalities define a cone (apex 0) in $R^{\binom{n}{2}}$, which we call a NJ cone.

- So NJ will output a particular tree topology T iff the pairwise distances $D \in R^{\binom{n}{2}}$ lie in a union of NJ cones.

Issues with neighbor joining

- Neighbor joining is fast and consistent, but it isn't based on a model of speciation.

- Until recently, it hasn't been very clear what NJ is optimizing — if anything at all.

- Neighbor joining outputs a tree topology T iff the data lies in a union of cones. Unions of cones need not be convex.

- In fact neighbor joining is not convex: There are distance matrices D, D', such that NJ produces the same tree T_1 when run on input D or D', but NJ produces a different tree $T_2 \neq T_1$ when run on the input $(D + D')/2$

BME cones and NJ cones

- For each tree topology τ, we take the ratio the NJ cones and the BME cone by comparing the sperical volumes of intersections between the NJ cones and the unit sphare and between the BME cone and the unit sphare.

- A key requirement is the measurement of volumes of spherical polytopes in high dimension, which we obtain using a combination of traditional Monte Carlo methods and polyhedral algorithms.

- Our analysis reveals new insights into the performance of the NJ and BME algorithms for phylogenetic reconstruction.

Comparing NJ and BME cones

- As a supplement to our forthcoming paper, we are creating a catalog of frequencies of all possible types of pairs of NJ and BME trees, for up to 8 (or perhaps even more) taxa.

- Quick summary stats: Overall agreement between NJ and BME topologies is
$100\%, 98\%, 90\%, 80\%, 65\%$ for $n = 4, 5, 6, 7, 8$ taxa.

- For $n \geq 7$ taxa, the ability of NJ to recover a BME caterpillar tree decreases much more quickly than for other BME tree topologies.

Future work

- We conjecture that the caterpillar tree is the most difficult BME tree for NJ to reproduce, as the number of taxa grows. Is this true? Why?

- In general, how does NJ's performance as a greedy BME heuristic depending on the topology of the BME tree?

- Rather than compare NJ and BME under a Gaussian distribution on $R^{\binom{n}{2}}$, one could use other distributions — namely $D = D_0 + \epsilon$, where D_0 are the true distances, and ϵ is either Gaussian or distributed according to the WLS in BME. This might still lead to some tractable and interesting computational geometry.

- Is there a combinatorial criterion (or at least sufficient conditions) for when two tree topologies form an edge on the BME polytope? Can this be used as a better way to move through tree space?

Advertisement...

The Sepcial Session on Advances in Algebraic Statistics

Organized by Sonja Petrović and RY

2010 AMS Spring Southeastern Sectional Meeting

Lexington, KY, March 27–28, 2010 (Saturday – Sunday)

`http://www.ams.org/amsmtgs/2162_program_ss2.html#title`

Thank you....