# On the Optimality of the Neighbor Joining Algorithm

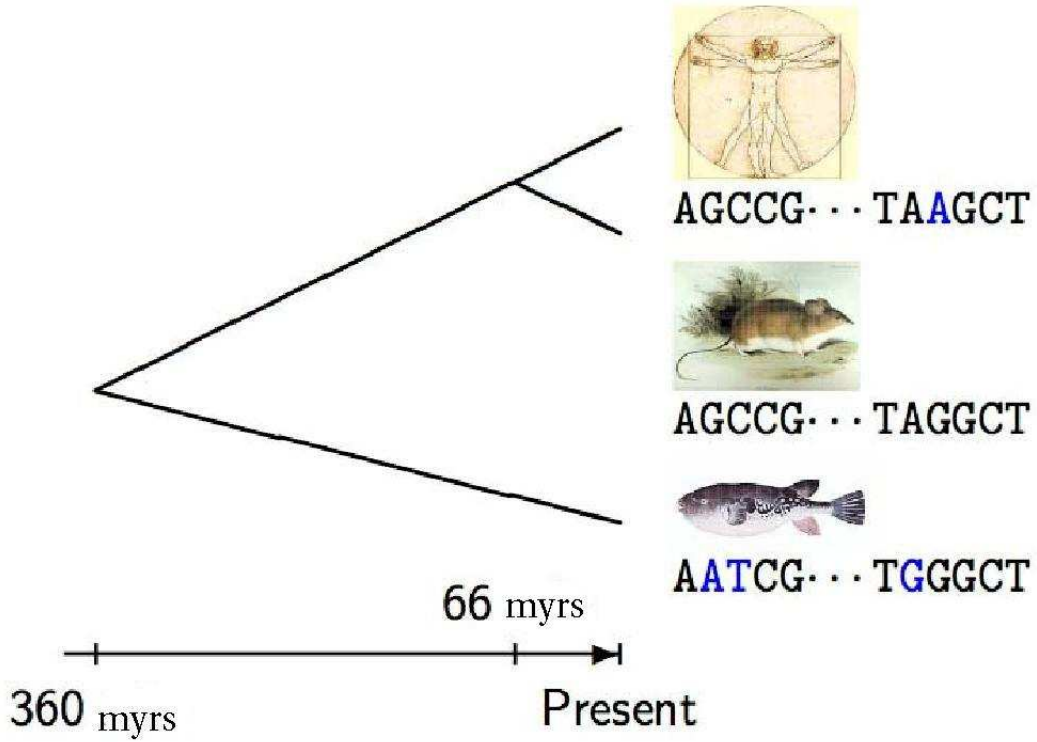
Ruriko Yoshida
Dept. of Statistics University of Kentucky

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

www.ms.uky.edu/~ruriko

# 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.

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)).

# Comparing NJ to BME

- Very recently it has been shown that neighbor joining is a greedy heuristic for finding BME trees. This means that NJ actually belongs to a class of several techniques (such as FastME) which iteratively improve the BME tree length by modifying the tree.

- Specifically, given pairwise distances $D$, neighbor joining starts with a star tree, and then repeatedly picks the cherry which results in the largest decrease in BME tree length $D \cdot W_\tau$ where

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

  for a particular tree topology $\tau$.

- So our motivating question is: How good of a greedy heuristic is NJ for BME? In other words, how often does NJ recover the BME tree?

# Neighbor joining: Fast and consistent

- 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 a cherry in the tree.

- Actually NJ joins nodes $a, b$ which have minimal $Q$-criterion:

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

- 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 NJ 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.

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.....

# Neighbor joining's output is defined by cones

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

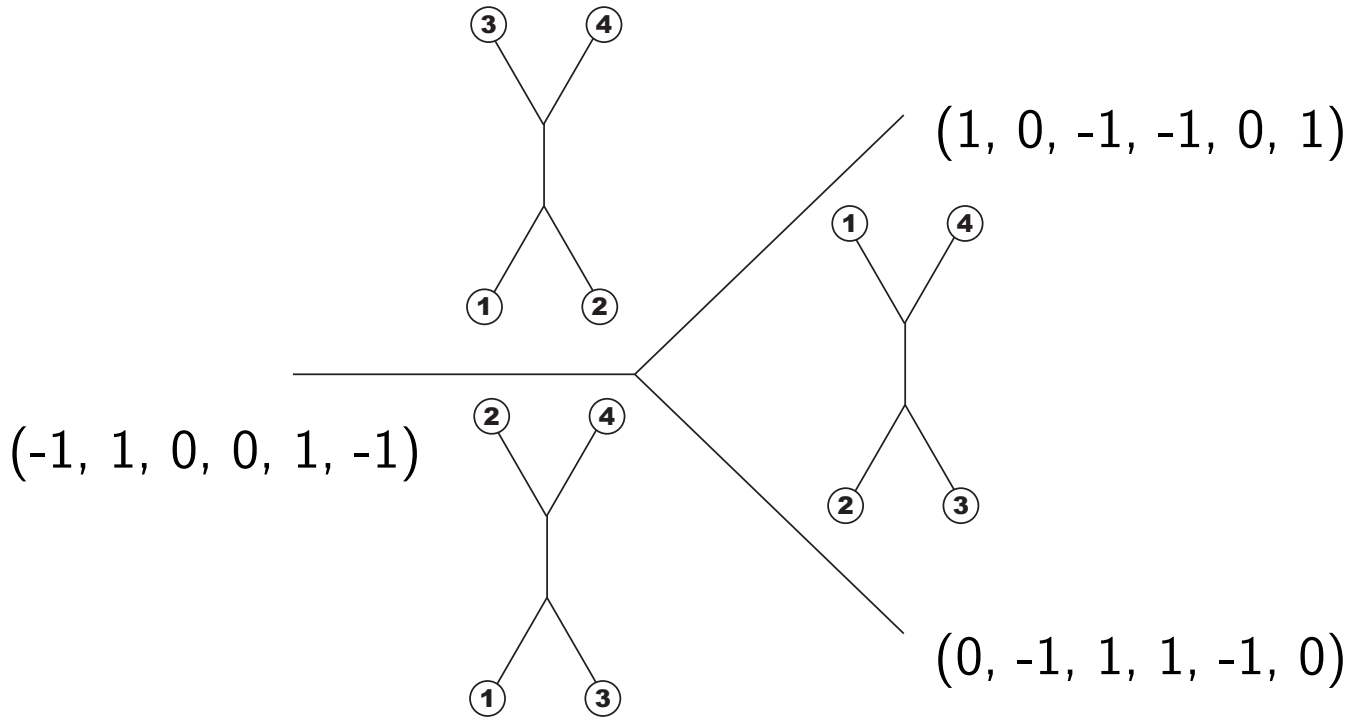
$$d_{ab} - \frac{1}{n-2}(\sum_k d_{ak} + d_{bk}) \leq d_{ij} - \frac{1}{n-2}(\sum_k d_{ik} + d_{jk}), \ \forall i, j$$

  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 $\tau$, 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 $\tau$ iff the pairwise distances $D \in R^{\binom{n}{2}}$ lie in a union of NJ cones.

# Example for $n = 4$



# 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 $\tau$ 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 $\tau_1$ when run on input $D$ or $D'$, but NJ produces a different tree $\tau_2 \neq \tau_1$ when run on the input $(D + D')/2$

# 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 which estimates the total length of a tree, based on:

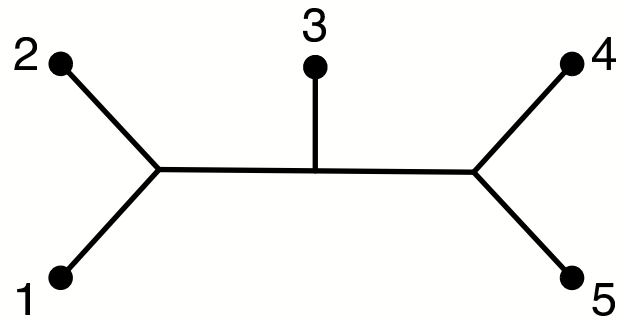
(1) its topology $\tau$,

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

The BME is to find $\tau$ such that $\min_{\tau_t,\, t=1,\cdots,(2n-5)!!} D \cdot W_{\tau_t}$ where

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

# 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

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

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

# 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 $\tau_1, \tau_2$ 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.

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

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

- BME cones are convex.

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

# BME cones and NJ cones

- For each tree topology $\tau$, 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.

- 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 $\epsilon$ 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?

# Thank you....

http://arxiv.org/abs/0710.5142