1 Recap and Definitions

Last lecture we defined the task of constructing a phylogenetic tree. Our target is to reconstruct a tree, representing the inferred evolutionary relationships among various DNA sequences, given the distances between each pair of sequences. Each node in the tree represents a DNA sequence, and the length of a path from node $i$ to node $j$ on the tree represents the distance between sequence $i$ and sequence $j$ according to some sequence distance metric. The distances between each pair of sequences is represented in a distance matrix $D$, such that:

$$D_{i,j} = d(i, j)$$

Where $d$ is a distance metric between 2 sequences. A metric is a pair $(X, d)$ such that $X$ is a group of points and $d$ is a distance function, $d: X^2 \rightarrow \mathbb{R}^+$, such that $d$ is reflexive, symmetric and maintains the triangle inequality.

We continue with a few definitions:

- Ultrametricity:
  A distance matrix $d$ is ultrametric if and only if for each triplet of indices (leaves in the matching tree) $i, j, k$ there is a permutation such that without loss of generality:

$$d_{i,k} = d_{j,k} \geq d_{i,j}$$

In terms of evolutionary sequences, this property states that any pair of leaves will be in an identical distance from their least common ancestor, as we expect the time past since any 2 species had split from their least common ancestor will be the same.

- Additivity:
  A distance matrix $d$ is additive if and only if it has a matching tree representation $T$ with branch lengths $\{l\}$ such that for every pair of leaves $i, j$:

$$d_{i,j} = \sum_{l \in \text{path in } T \text{ from } i \text{ to } j} l$$

In phylogenetic trees the meaning of the distance between 2 sequences is the time it took for one sequence to evolve from the other. Therefore, the distance between 2 sequences should match the sum of distances along their path.

- The 4 points property:
  A tree $T$ with a distance matrix $d$ satisfies the 4 points property if for each 4 nodes in $T$ there is a permutation $i, j, k, l$ such that:
In terms of evolutionary sequences the 4 points property states that we expect 2 sequences that evolved from a single sequence to be more similar to each other (that is, closer in the phylogenetic tree) than to other pairs of sequences.

The 4 points property holds for any additive distance matrix.

2 Neighbor Joining Algorithm

Last lecture we saw that the UPGMA algorithm relies on the ultrametric property of the given tree. Therefore, if the tree is not ultrametric, applying the UPGMA algorithm on the distance matrix results in a tree with a different distance matrix.

To address that issue we will use the 4 points property. Instead of joining the closest 2 points in each iteration, as in the UPGMA algorithm, we will join the 2 points that are closest to each other and furthest from other couples.

We will define a similarity measure:

\[ Q_{i,j} = r_i + r_j - d_{i,j} \]

Where \( d_{i,j} \) is the distance between node \( i \) and node \( j \), and \( r_i \) is the average distance of node \( i \) from all other nodes (with a small change in the denominator), such that:

\[ r_i = \sum_{k \in \text{all nodes in } T, k \neq i} \frac{d_{i,k}}{(n-2)} \]

Where \( n \) is the total number of nodes.

2.1 Theorem:

Let \( i, j \) be such that:

\[ i^*, j^* = \arg\max Q_{i,j} \]

\( i \) and \( j \) are neighbors in the additive tree of the distance matrix \( Q \).
Algorithm 1 Neighbor Joining

Init: Set $A = \{n\}$
For each $i$ initialize: $r_i = \sum_{k \in \text{all nodes in } T, k \neq i} \frac{d_{i,k}}{n-2}$
For each $i,j$ initialize: $Q_{i,j} = r_i + r_j - d_{i,j}$
For $L = 1,...,n-1$ do:
Find $i,j$ such that: $i,j = \argmax_{k,l \in A} Q_{k,l}$
Create a new node $k$ to be such that: $k = \text{merge}(i,j)$
For each node $m$ in tree $T, m \notin \{i,j\}$, do: $d_{m,k} = \frac{1}{2}(d_{i,m} + d_{j,m} - d_{i,j})$
Update: $d_{i,k} = \frac{1}{2}(d_{i,j} + r_i - r_j), d_{j,k} = \frac{1}{2}(d_{i,j} + r_j - r_i)$
Update: $A = \{k\} \cup A \setminus \{i,j\}$
End loop

Algorithm input: A distance matrix $d$, where for each sequence pair $i,j$:

$$D_{i,j} = d(i,j)$$

Algorithm output: An updated distance matrix $d$, with all of the original sequences and some additional nodes that represent the inner nodes of the constructed tree.

The algorithm performs correctly for additive matrices, without the requirement of ultrametricity. That is, given a tree with an additive distance matrix that is not necessarily ultrametric, applying the Neighbor Joining algorithm on its distance matrix results in the original tree.