Phylogenetic trees

Neighbor joining

Outline

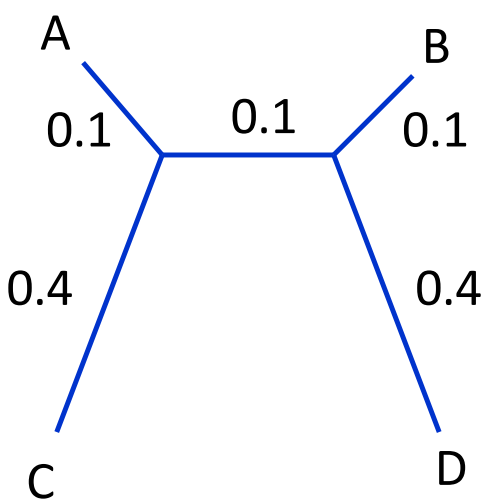
- The Neighbor joining algorithm for estimating a phylogenetic tree from pairwise distances
- Checking for additivity
- Rooting an unrooted tree

Overview of neighbor joining

- Motivation: the assumption about the ultrametric property is too strong
  - Most sequences diverge at different rates
- A more relaxed requirement is that of additivity
  - Distance between a pair of species/nodes is equal to the sum of the branch lengths
- Neighbor joining
  - Guaranteed to succeed if additivity property holds
  - Like UPGMA, consider pairs of nodes at each iteration and joins them
  - Produces unrooted trees

How to select nodes for joining?

- Given all pairwise distances for n taxa
- d_{ij} denote the distance between node i and j
- Should we select node pairs with the smallest d_{ij}?



$$d_{AB} = 0.3$$

$$d_{AC} = 0.5$$

This will give us an incorrect tree

Selecting nodes to join

- Neighbor joining requires us to correct the distance to account for distances from all other nodes.
- The corrected distance is denoted as $\mathrm{D_{ij}}$

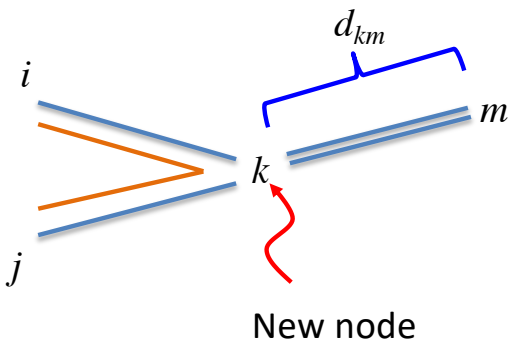
$$D_{ij} = d_{ij} - (r_i + r_j)$$

$$r_i = \frac{1}{L-2} \sum_{1 \leq k \leq L} d_{ik}$$

L: number of leaves

r_i : “Average” distance from all other leaves

Defining the distance to a new node



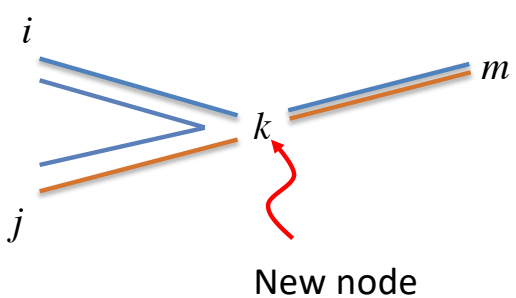
Given d_{ij}, d_{im}, d_{jm}, how to calculate distance of existing node m to new node k?

$$d_{km} = \frac{d_{im} + d_{jm} - d_{ij}}{2}$$

Updating Distances in Neighbor Joining

- can calculate the distance from a leaf to its parent node in the same way

$$d_{ik} = \frac{1}{2}(d_{ij} + d_{im} - d_{jm})$$



$$d_{jk} = d_{ij} - d_{ik}$$

Updating Distances in Neighbor Joining

- we can generalize this so that we take into account the distance to all other leaves

$$d_{ik} = \frac{1}{2}(d_{ij} + r_i - r_j)$$

where

$$r_i = \frac{1}{|L| - 2} \sum_{m \in L} d_{im}$$

and L is the set of leaves

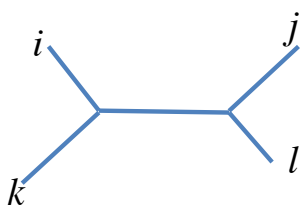
- this is more robust if data aren't strictly additive

Algorithm for NJ

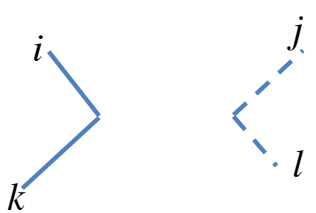
- Inputs
  - Pairwise distance matrix entries: d_{ij}
- Initialization
  - T be set the of leaf nodes
  - $L = T$
  - Compute r_i for all i in L
  - Compute D_{ij}
- Iteration
  - Pick a pair i, j from L such that D_{ij} is smallest
  - Define new node k
  - Compute d_{ik}, d_{jk}, add edge between k and i, and between k to j
  - Compute d_{km} for all other nodes m in L
  - Add k to L, remove i and j from L
  - (Re)compute D_{mn} for all nodes m, n in L
- Terminate
  - If L has two nodes, add the edge between these two.

Can we check for additivity?

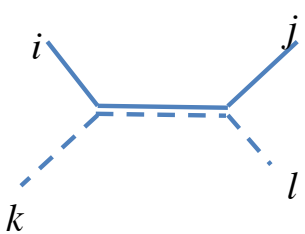
Check for additivity: For four leaves, i, j, k, l and the distances d_{ij}, d_{ik}, d_{il}, d_{jk}, d_{jl}, d_{kl}



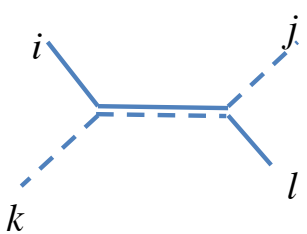
The three sums of two distances



$d_{ik} + d_{jl}$



$d_{ij} + d_{kl}$



$d_{il} + d_{jk}$

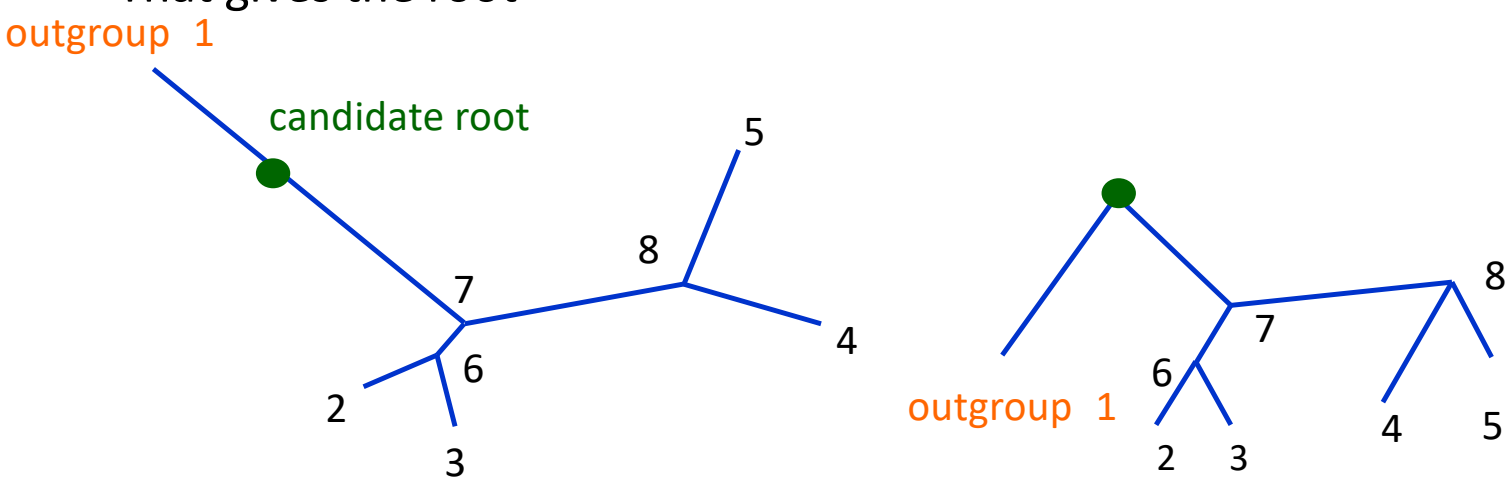
Should be such that two of these are equal, and larger than the third.

Comparing NJ and UPGMA

- Similarities
  - Input is pairwise distance matrix
  - Iteratively join nodes in greedy manner
- UPGMA
  - Rooted tree
  - Assumptions: Additivity and ultrametric (Molecular clock)
- NJ
  - Unrooted tree
  - Assumption: Additivity

Rooting a tree

- An unrooted tree can be converted to a rooted tree using an outgroup species
- Outgroup: a species known to be more distantly related to all the species than each of the species themselves
- Find the branch where the outgroup is selected to be added
- That gives the root



Summary

- Neighbor joining succeeds in reconstructing the correct tree with only the assumption of additivity
- Neighbor joining and UPGMA are both greedy iterative algorithms
- The algorithms differ in their choice of the pair of nodes that are to be joined at each step
- Although neighbor joining outputs an unrooted tree, outgroup information can be used to root it.