Distance based phylogeny reconstruction

Usman Roshan

Neighbor joining

1. Additive and *O(n^3)* time
2. Initialization: same as UPGMA
3. For each species compute

$$u_i = \frac{\sum_{k=1,k\neq i}^{n} d_{ik}}{n-2}$$

4. Select *i* and *j* for which $d_{ij} - (u_i + u_j)$ is minimum
5. Make them neighbors in the tree by adding new node *(ij),* and set tree distance from *(ij)* to *i* and *j* as

$$d_{i,(ij)} = \frac{1}{2}d_{i,j} + \frac{1}{2}(u_i - u_j) \qquad d_{j,(ij)} = \frac{1}{2}d_{i,j} + \frac{1}{2}(u_j - u_i)$$

Neighbor joining

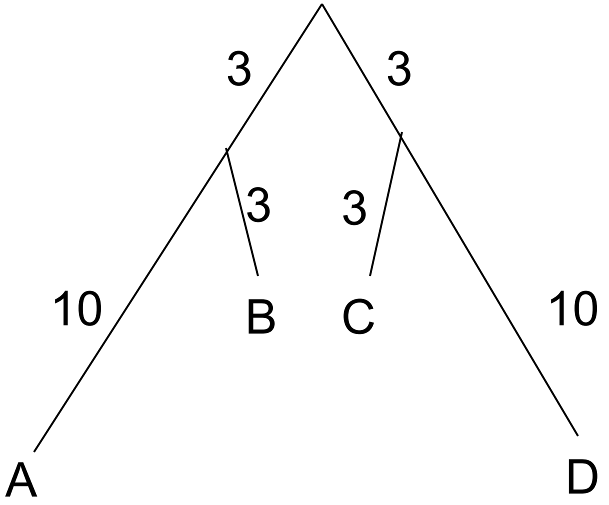
6. Update distance matrix *d*: for all clusters *k* do the following

$$d_{(ij),k} = \frac{d_{ik} + d_{jk} - d_{ij}}{2}$$

7. Delete columns and rows for *i* and *j* in *d* and add new ones corresponding to cluster *(ij)* with distances as computed above
8. Go to 3 until two nodes/clusters are left

NJ

NJ constructs the correct tree for additive matrices



↔

	A	B	C	D
A		13	19	26
B			12	19
C				13
D				

Neighbor joining implementation in Python

- Instead of -1 to denote unusable entries of d we will use ‘X’. For this we make changes to
  - Find_closest_pair
  - Update_distance_matrix
  - Make_little_d

Neighbor joining implementation in Python

- We also need to modify the functions below to reflect neighbor joining updates
  - Find_closest_pair
  - Update_distance_matrix
  - Distance