

Ch. 3: Phylogeny

Distance Algorithms: UPGMA and Neighbor-Joining

Distance Algorithms

- The UPGMA (Unweighted Pair Group Method with Arithmetic Mean) and Neighbor-Joining Algorithms are used in phylogeny to determine an accurate way of arranging a group of taxa in a phylogenetic tree.
- This allows us to see how closely related different taxa are, as well as how they are connected.
- For both of these algorithms, we begin with a distance matrix in which the numerical phylogenetic difference between various taxa is given. Using these values, we can use the UPGMA and Neighbor-Joining algorithms to find probable relationships between the taxa and construct a phylogenetic tree that reflects this accurately.

“Clustering” Algorithms

- Both UPGMA and Neighbor Joining involve the formation of “clusters” to unite similar groups.
- More specifically, at every iteration of the algorithms, two taxa in the current pool will be merged into a “cluster,” and this cluster will be treated as one taxon in the next iteration.
- So, as the algorithms continue, we will begin to form multiple clusters, as well as clusters of clusters, until we ultimately have one lone cluster that includes all of the taxa we are interested in (at this point, we have our phylogenetic tree).

Ultrametric Trees

- Although both the UPGMA and Neighbor-Joining algorithms are used to construct phylogenetic trees, there is one important distinction between them.
- For UPGMA, we assume that evolution is clocklike. In other words, we assume that, across all branches of the tree, evolution proceeds at a uniform rate. We refer to a tree built in this fashion as ultrametric.
- Because of this assumption, inaccuracies can arise. However, the Neighbor-Joining algorithm mitigates this issue, since it does not assume uniform evolution rates (non-ultrametric).

UPGMA

1. Create a distance matrix for the taxa of interest.
2. Identify the two taxa with the shortest distance between them, then merge them into one OTU (operational taxonomic unit). This new cluster will be used in subsequent calculations instead of the original taxa that make it up.
3. Calculate a new distance matrix, taking into consideration the new OTU previously formed.
4. Repeat these steps until all of the taxa are merged into one cluster.

UPGMA (example)

Take the following distance matrix for 4 taxa (A, B, C, and D):

	A	B	C
B	4		
C	2	6	
D	3	7	5

UPGMA (example)

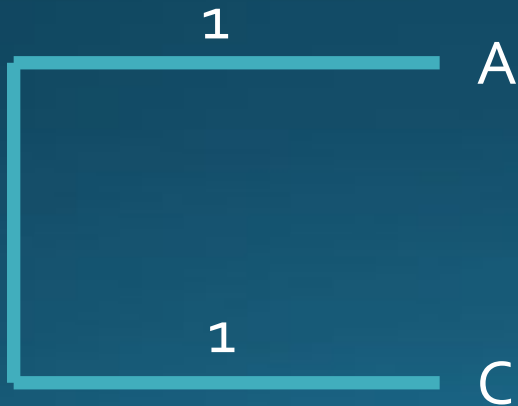
First, we find the smallest value in the table (2, in this case).

	A	B	C
B	4		
C	2	6	
D	3	7	5

Since 2 is the distance between A and C, we merge A and C into one cluster (AC).

UPGMA (example)

The AC cluster that we formed can be represented in a phylogenetic tree as shown below. Since the total distance between A and C is 2, we label the branches of the tree with a value of 1.



UPGMA (example)

Now, we must generate a new distance matrix.

To calculate the distance between each remaining taxon to the new cluster AC, we must sum the distance of a taxon from both A and C then divide this value by 2.

In other words, take the expression below (where x is some taxon remaining in the table, and D is the distance):

$$D_{x,AC} = 0.5 * (D_{x,A} + D_{x,C})$$

UPGMA (example)

Using $D_{x,AC} = 0.5 * (D_{x,A} + D_{x,C})$, we can generate our new distance matrix.

	A	B	C
B	4		
C	2	6	
D	3	7	5



	AC	B
B	5	
D	4	7

$$D_{B,AC} = 0.5 * (D_{B,A} + D_{B,C}) = 0.5 * (4 + 6) = 5$$

$$D_{D,AC} = 0.5 * (D_{D,A} + D_{D,C}) = 0.5 * (3 + 5) = 4$$

UPGMA (example)

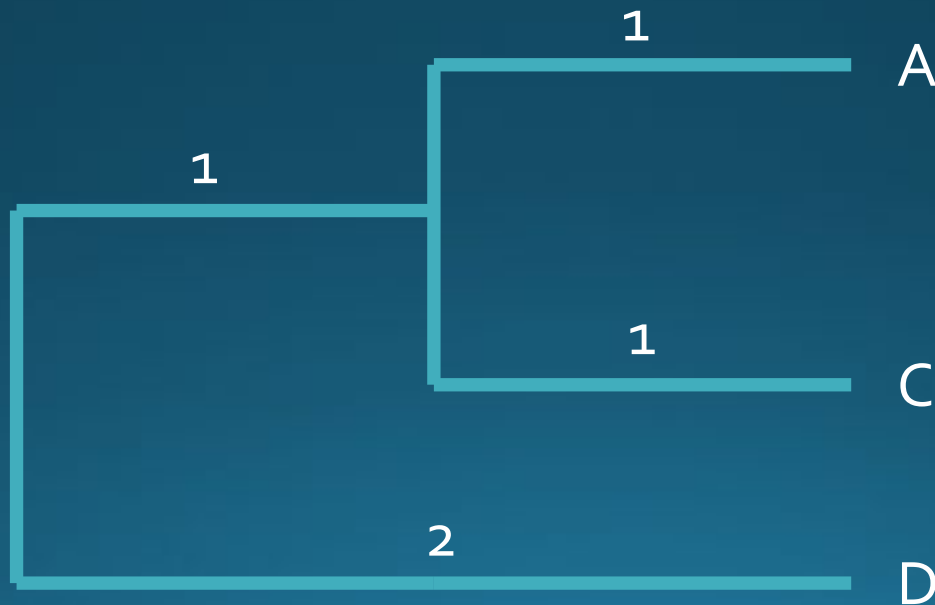
Now, we take the smallest value from the new matrix (4, in this case).

	AC	B
B	5	
D	4	7

Since 4 is the distance between AC and D, we merge AC and D into one cluster (ACD).

UPGMA (example)

Because we formed a new cluster, we must update our phylogenetic tree. We already have A and C in our tree, but we must add taxon D appropriately. Since the total distance between the AC cluster and D is 4, we can represent our ACD cluster as shown below.



UPGMA (example)

Now, we must generate a new distance matrix, using the same formula as before, but this time finding the distance between the ACD cluster and the remaining OTUs (only B, in this case).

	AC	B
B	5	
D	4	7



	ACD
B	6

$$D_{B,ACD} = 0.5 * (D_{B,AC} + D_{B,D}) = 0.5 * (5 + 7) = 6$$

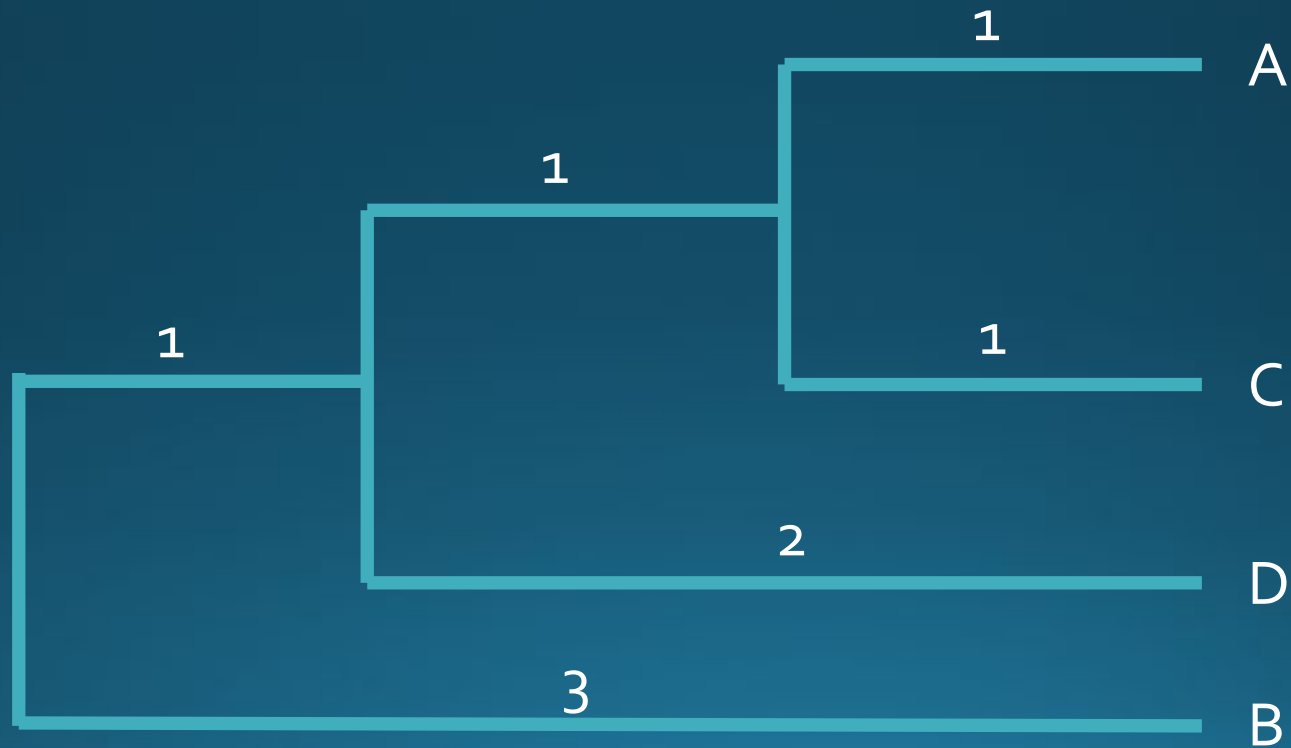
UPGMA (example)

There is only one value left in the matrix, so we add the remaining taxon to the cluster. Now, we have merged all of the original taxa into one cluster (ACDB).

	ACD
B	6

UPGMA (example)

We must update our phylogenetic tree one last time. We already have A, C, and D in our tree, but we must add taxon B appropriately. Since the total distance between the ACD cluster and B is 6, we can represent our ACDB cluster as shown below. We are done.



Neighbor-Joining

1. Create a distance matrix for the taxa of interest.
2. From the beginning, we will assume an ambiguous tree in which all of the taxa are branching out from a single central node. This tree will be amended throughout the algorithm.
3. For each OTU, calculate the following (call it S): take the sum of the distances between that OTU and every other OTU, then divide this value by $N-2$, where N is the total number of OTUs.
4. Determine which pair of OTUs yields the smallest value for the following expression: $M_{ij} = D_{ij} - S_i - S_j$. These OTUs will be merged.

Neighbor-Joining (continued)

5. As was done in UPGMA, join the two taxa corresponding to this minimum value at a node in a subtree (note: the tree in this case will be formed differently than in UPGMA, since it is not ultrametric. This will become clearer when we work through an example).
6. Now that we have decided which OTUs will be at the ends of our subtree, we must determine the length of the branches that they stem from. This can be done as follows (where x is the start node and i, j are the OTUs): $D_{xi} = (1/2) * D_{ij} + (1/2) * (S_i - S_j)$. To calculate D_{xj} , simply swap the S_i and S_j in the final set of parentheses.

Neighbor-Joining (continued)

7. A new distance matrix must now be calculated, replacing the two OTUs that were joined with a node representing a cluster of the original OTUs. In order to do this, we can perform the following (where x is the new node, i and j are the OTUs making up x , and k is one of the remaining OTUs):
$$D_{xk} = (D_{ik} + D_{jk} - D_{ij}) / 2$$
8. Repeat these steps until all of the original OTUs are gathered into one cluster, at which point an accurate tree can be constructed for the taxa of interest.

Neighbor-Joining (example)

Take the following distance matrix for 4 taxa (A, B, C, and D):

	A	B	C
B	5		
C	6	2	
D	3	4	7

Neighbor-Joining (example)

First, we must calculate S values for every OTU.

$$S_A = (5+6+3) / (4-2) = 7$$

$$S_B = (5+2+4) / (4-2) = 5.5$$

$$S_C = (6+2+7) / (4-2) = 7.5$$

$$S_D = (3+4+7) / (4-2) = 7$$

	A	B	C
B	5		
C	6	2	
D	3	4	7

Neighbor-Joining (example)

Now, we must find the minimum value for M_{ij} among all of the OTU pairs.

$$M_{AB} = D_{AB} - S_A - S_B = 5 - 7 - 5.5 = -7.5$$

$$M_{AC} = D_{AC} - S_A - S_C = 6 - 7 - 7.5 = -8.5$$

$$M_{AD} = D_{AD} - S_A - S_D = 3 - 7 - 7 = -11$$

$$M_{BC} = D_{BC} - S_B - S_C = 2 - 5.5 - 7.5 = -11$$

$$M_{BD} = D_{BD} - S_B - S_D = 4 - 5.5 - 7 = -8.5$$

$$M_{CD} = D_{CD} - S_C - S_D = 7 - 7.5 - 7 = -7.5$$

	A	B	C
B	5		
C	6	2	
D	3	4	7

The M values for A,D and B,C are the smallest. We can choose either one. In this case, let's choose A,D and merge them together.

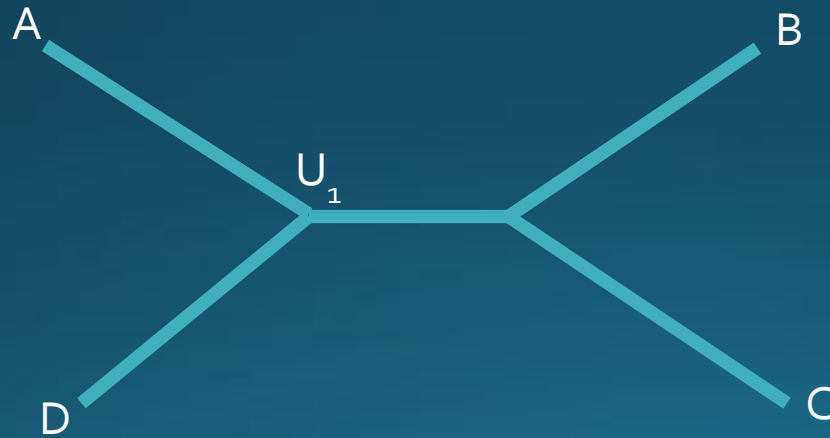
Neighbor-Joining (example)

We must now reflect this merge in our tree. As stated earlier, we will begin with an ambiguous tree with all OTUs branching off from a single node, as shown below.



Neighbor-Joining (example)

We decided that we are going to merge A and D. To do this, we must create another node (call it U_1) that branches off from the center node, and then have A and D branch off from this new node, as shown.

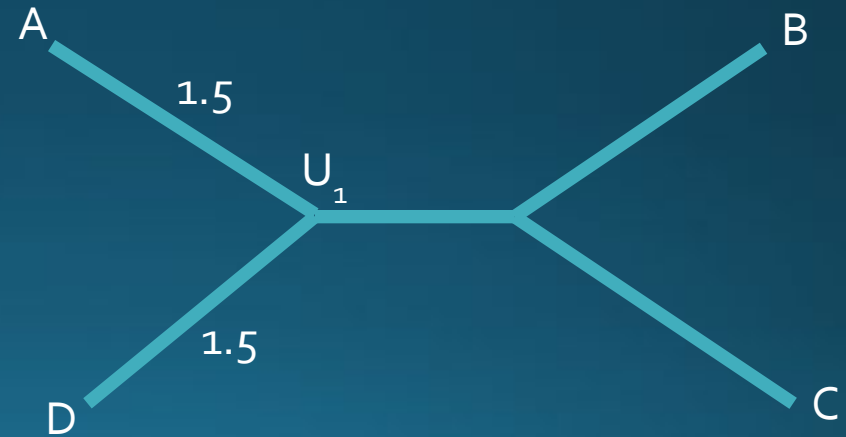


Neighbor-Joining (example)

Now, we have to calculate the distance between U_1 and A and U_1 and D. This value (D_{xi}) is equal to $(1/2) * D_{ij} + (1/2) * (S_i - S_j)$.

$$D_{U_1A} = (1/2) * (3) + (1/2) * (7-7) = 1.5$$

$$D_{U_1D} = (1/2) * (3) + (1/2) * (7-7) = 1.5$$



Neighbor-Joining(example)

Using $D_{xk} = (D_{ik} + D_{jk} - D_{ij}) / 2$, we must generate our new distance matrix.

	A	B	C
B	5		
C	6	2	
D	3	4	7



	AD	B
B	3	
C	5	2

$$D_{AD,B} = (D_{A,B} + D_{D,B} - D_{A,D}) / 2 = (5 + 4 - 3) / 2 = 3$$

$$D_{AD,C} = (D_{A,C} + D_{D,C} - D_{A,D}) / 2 = (6 + 7 - 3) / 2 = 5$$

Neighbor-Joining(example)

Now, we must recalculate S values for our new distance matrix.

$$S_{AD} = (3+5) / (3-2) = 8$$

$$S_B = (3+2) / (3-2) = 5$$

$$S_C = (5+2) / (3-2) = 7$$

	AD	B
B	3	
C	5	2

Neighbor-Joining (example)

Next, we must find the minimum value for M_{ij} among all of the OTU pairs.

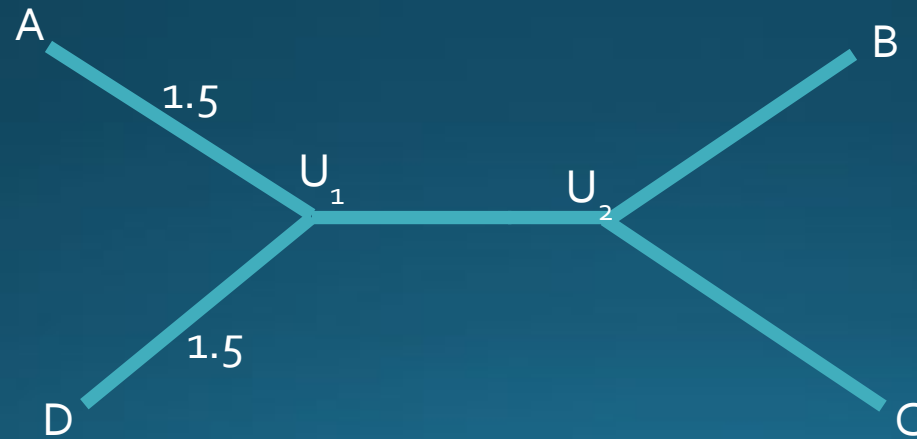
$$M_{ADB} = D_{ADB} - S_{AD} - S_B = 3 - 8 - 5 = -10$$
$$M_{ADC} = D_{ADC} - S_{AD} - S_C = 5 - 8 - 7 = -10$$
$$M_{BC} = D_{BC} - S_B - S_C = 2 - 5 - 7 = -10$$

	AD	B
B	3	
C	5	2

The M values for all of the remaining options are the same. In this case, let's merge B and C together (we could have chosen any one).

Neighbor-Joining (example)

We decided that we are going to merge B and C. To do this, we must create another node (call it U_2) that branches off from the center node, and then have B and C branch off from this new node, as shown.

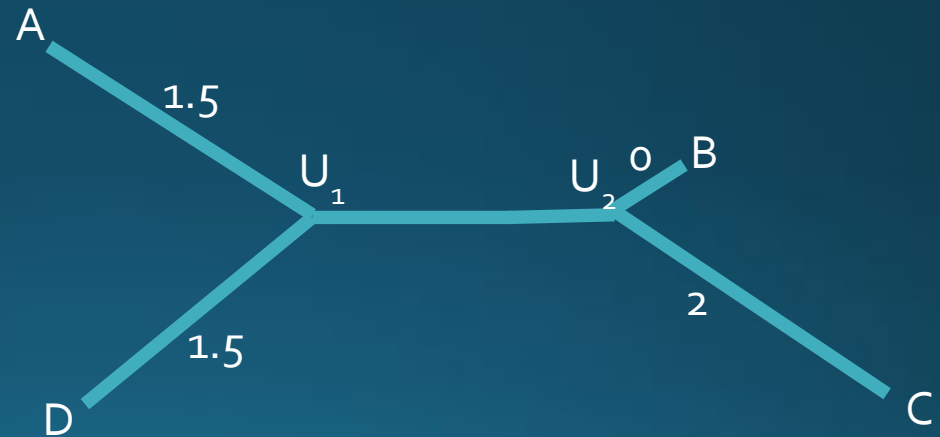


Neighbor-Joining (example)

Now, we have to calculate the distance between U_2 and B and U_2 and C. Again, this value (D_{xi}) is equal to $(1/2) * D_{ij} + (1/2) * (S_i - S_j)$.

$$D_{U_2B} = (1/2) * (2) + (1/2) * (5-7) = 0$$


$$D_{U_2C} = (1/2) * (2) + (1/2) * (7-5) = 2$$



Neighbor-Joining(example)

Using $D_{xk} = (D_{ik} + D_{jk} - D_{ij}) / 2$, we must generate our new distance matrix.

	AD	B
B	3	
C	5	2

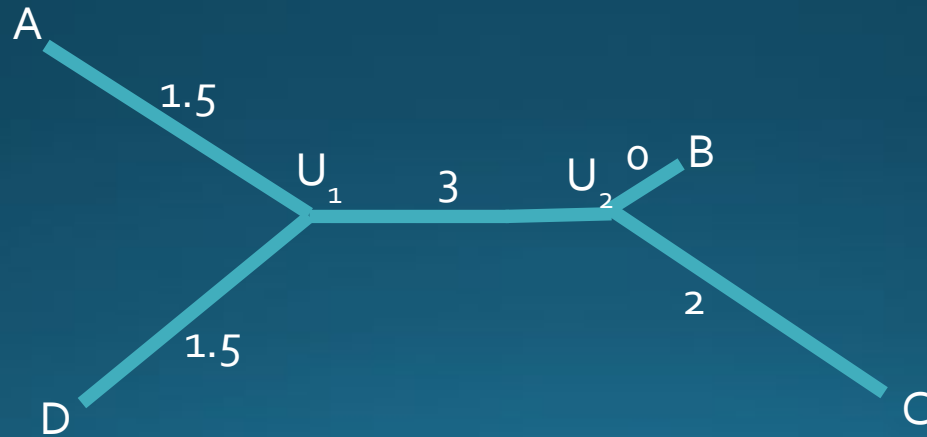


	AD
BC	3

$$D_{BC,AD} = (D_{B,AD} + D_{C,AD} - D_{B,C}) / 2 = (3 + 5 - 2) / 2 = 3$$

Neighbor-Joining (example)

All that's left to do is add the distance 3 between the AD and BC portions of the tree (in other words, between the U_1 and U_2 nodes). We are now done.



More UPGMA and Neighbor-Joining

- For a more in-depth look at the two algorithms we just went over, take a look at the Barton textbook chapter posted to the Assignments page of the course website next to HW₄.