Neighbor Joining Algorithm of Phylogenetic tree construction

Neighbor joining is another distance-based method for tree construction. It uses clustering approach to construct the tree. It differs with UPGMA that it does not necessarily produce ultrametric tree (branch length can vary in size from common ancestor). This method was give by Saitou and Nei in 1987. The well-known Clustal package uses this approach to construct phylogenetic tree. Neighbor joining is a special case of star decomposition. In contrast to cluster analysis, neighbor joining keeps track of nodes on a tree rather than taxa or clusters of taxa. For the construction of the tree, we need a distance matrix derived from the pair wise hamming distances of the sequences under examination.

Working Example:

Suppose we have to construct the NJ tree using following distance matrix. We will start with a star topology of tree where all taxon will originate from a common center.

	A	B	C	D	E
A	**0**				
B	**4**	**0**			
C	**5**	**7**	**0**		
D	**2**	**6**	**3**	**0**	
E	**3**	**8**	**4**	**6**	**0**

We will construct a modified distance matrix is constructed in which the separation between each pair of nodes is adjusted based on their average divergence from all other nodes.

We calculate the net divergence r (i) for each OTU from all other OTUs

$r_{(A)} = 4 + 5 + 2 + 3 = 14$
$r_{(B)} = 4 + 7 + 6 + 8 = 25$
$r_{(C)} = 5 + 7 + 3 + 4 = 19$
$r_{(D)} = 2 + 6 + 3 + 6 = 17$
$r_{(E)} = 3 + 8 + 4 + 6 = 21$

Now we calculate a new distance matrix using for each pair of OUTs the formula:

$M_{(ij)} = d_{(ij)} - [r_{(i)} + r_{(j)}] / (N-2)$ or in the case of the pair A,B:

$M_{(AB)} = d_{(AB)} - [(r_{(A)} + r_{(B)}] / (N-2) = 4 - [14 + 25] / 3 = -9$

$M_{(AC)} = d_{(AC)} - [(r_{(A)} + r_{(C)}] / (N-2) = 5 - [14 + 19] / 3 = -6$

$M_{(AD)} = d_{(AD)} - [(r_{(A)} + r_{(D)}] / (N-2) = 2 - [14 + 17] / 3 = -8.3$

$M_{(AE)} = d_{(AE)} - [(r_{(A)} + r_{(E)}] / (N-2) = 3 - [14 + 21] / 3 = -8.7$

We will calculate $M_{(ij)}$ for all other pairs and construct the following matrix

	A	B	C	D	E
A	0				
B	- 9	0			
C	-6	-7.7	0		
D	-8.3	-8	-9	0	
E	-8.7	-7.3	-11.3	-6.7	0

/ 3 = - 6.7

$M_{(BC)} = d_{(BC)} - [(r_{(B)} + r_{(C)}] / (N-2) = 7 – [25 + 19] / 3 = -7.7$

$M_{(BD)} = d_{(BD)} - [(r_{(B)} + r_{(D)}] / (N-2) = 6 – [25 + 17] / 3 = - 8$

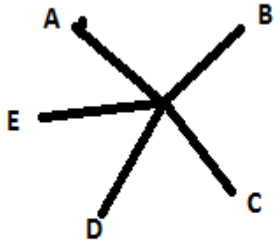
$M_{(BE)} = d_{(BE)} - [(r_{(B)} + r_{(E)}] / (N-2) = 8 – [25 + 17] / 3 = -7.3$

$M_{(CD)} = d_{(CD)} - [(r_{(C)} + r_{(D)}] / (N-2) = 3 – [25 + 17] / 3 = -9$

$M_{(CE)} = d_{(CE)} - [(r_{(C)} + r_{(E)}] / (N-2) = 4 – [25 + 17] / 3 = - 11.3$

$M_{(DE)} = d_{(DE)} - [(r_{(D)} + r_{(E)}] / (N-2) = 6 – [25 + 17] / 3 = - 6.7$

Now we choose as neighbors those two OTUs for which Mij is the smallest, which is between C & E 11.3, we will join these two taxa together. Now from our start topology we have to merge A and E together.



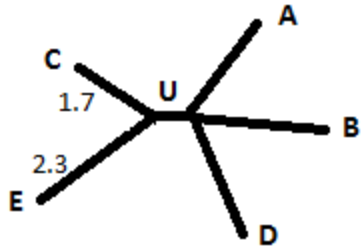
We will join C & E with a common ancestor U. Now we calculate the branch length from the internal node U to the external OTUs A and E with following formula.

$S_{(CU)} = d_{(CE)} / 2 + [r_{(C)} - r_{(E)}] / 2(N-2)$

$S_{(CU)} = 4/2 + [19 – 21] / 2 *3 = 2 + \{ [-2] /6 \} = 1.7$

$S_{(EU)} = d_{(CE)} - S_{(CU)} = 4 – 1.7 = 2.3$

Now our tree topology will change as following



now we define new distances from U to each other terminal node

$d_{(AU)} = [d_{(AC)} + d_{(AE)} - d_{(CE)}] / 2 = [5 + 3 – 4] / 2 = 2$

$d_{(BU)} = [d_{(BC)} + d_{(BE)} - d_{(CE)}] / 2 = [7 + 8 – 4] /2 = 5.5$

$d_{(DU)} = [d_{(DC)} + d_{(DE)} - d_{(CE)}] / 2 = [3 + 6 – 4] /2 = 2.5$

The new distance matrix will be as follows. We have to calculate net divergence with this matrix

	U	A	B	D
U	0			
A	2	0		
B	5.5	4	0	
D	2.5	2	6	0

$r_{(U)} = 2 + 5.5 + 2.5 = 10$

$r_{(A)} = 2 + 4 + 2 = 8$

$r_{(B)} = 5.5 + 4 + 6 = 15.5$

$r_{(D)} = 2.5 + 2 + 6 = 10.5$

We will create a new matrix M(ij) as done above with given formula

$M_{(ij)} = d_{(ij)} - [r_{(i)} + r_{(j)}] / (N-2)$, here value of N will be 4 – 2 = 2

	U	A	B	D
U	0			
A	-7	0		
B	-7.25	-7.75	0	
D	-7.75	-7.25	-7	0

M(UA) = 2 – [10 + 8] /2 = -7

M(UB) = 5.5 – [10 + 15.5] /2 = -7.25

M(UD) = 2.5 – [10 + 10.5] /2 = -7.75

M(AB) = 4 – [8 + 15.5] /2 = -7.75

M(AD) = 2 – [8 + 10.5] / 2 = -7.25

M(BD) = 6 – [15.5 + 10.5] /2 = - 7

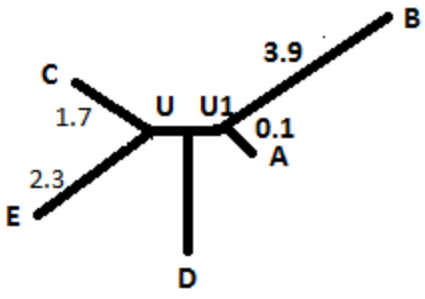
Now we will choose and merge the OUT having lowest M(ij) value which is 7.75 between UD & AB. Lets join AB this time and make this U1

We will merge BD and with common node U1

$S_{(AU1)} = d_{(AB)} / 2 + [r_{(A)} - r_{(B)}] / 2(N-2) = 4/2 + \{[8 - 15.5] / 2 * 2\} = 2 + \{[-7.5] /4\} = 0.1$

$S_{(BU1)} = d_{(AB)} - d_{(AU1)} = 4 - 0.1 = 3.9$

We will rearranged our tree with merging of BC under U1 with calculated distance



Now we define new distances from U1 to each other terminal node

$d_{(UU1)} = [d_{(UA)} + d_{(UB)} - d_{(AB)}] / 2 = [2 + 5.5 - 4] / 2 = 1.75$

$d_{(DU)} = [d_{(DC)} + d_{(DE)} - d_{(CE)}] / 2 = [3 + 6 - 4] /2 = 2.5$

$d_{(DU1)} = [d_{(DA)} + d_{(DB)} - d_{(AB)}] / 2 = [2 + 6 - 4] /2 = 2$

The new distance matrix will be as follows. We have to calculate net divergence with this matrix

	U	U1	D
U	0		
U1	1.7	0	
D	2.5	2	0

$r_{(U)} = 1.7 + 2.5 = 4.2$

$r_{(U1)} = 1.7 + 2 = 3.7$

$r_{(D)} = 2.5 + 2 = 4.5$

We will create a new matrix M(ij) as done above with given formula

$M_{(ij)} = d_{(ij)} - [r_{(i)} + r_{(j)}] / (N-2)$, here value of N will be 3 – 2 = 1

M(UU1) = 1.7 – [4.2 + 3.7] /1 = -6.2

M(UD) = 2.5 – [4.2 + 4.5] /1 = -6.2

M(U1D) = 2 – [3.7 + 4.5] /1 = -6.2

We will create a matrix M_{ij}. Because all the values are same, we can join any one of them

	U	U1	D
U	0		
U1	-6.2	0	
D	-6.2	-6.2	0

Lets join U with D and make it U2

$S_{(DU2)} = d_{(UD)} / 2 + [r_{(D)} - r_{(U)}] / 2(N-2) = 2.5 /2 + \{ [4.5 – 4.2] / 2 * 1\} = 1.25 + \{ 0 .15\} = 1.4$

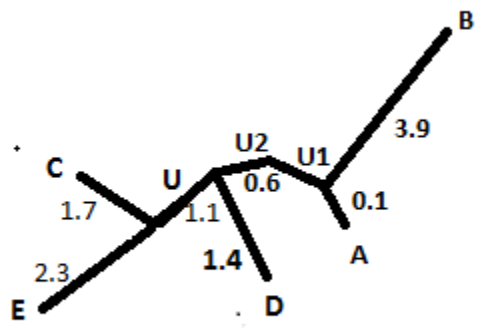
$S_{(UU2)} = d_{(DU)} - S_{(DU2)} = 2.5 – 1.4 = 1.1$

We will stop at this stage because matrix will reduce to 2 X 2 after this

	U1	U2
U1	0	
U2	0.6	0

The distance between U1 & U2 will be

$d_{(U1U2)} = [d_{(U1D)} + d_{(U1U)} - d_{(DU)}] / 2 = [2 + 1.7 – 2.5] / 2 = 0.6$



So this will be our final tree

Advantages and disadvantages of the neighbor-joining method

- **Advantages**
    - It is very fast and thus suited for large datasets and for bootstrap analysis.
    - It permits lineages with largely different branch lengths.
    - It permits correction for multiple substitutions.
    - It create scaled tree

Disadvantages

- The sequence information is reduced
- Neighbor joining gives only one possible tree