CISC 636 Computational Biology & Bioinformatics (Fall 2016)

Phylogenetic Trees (II)

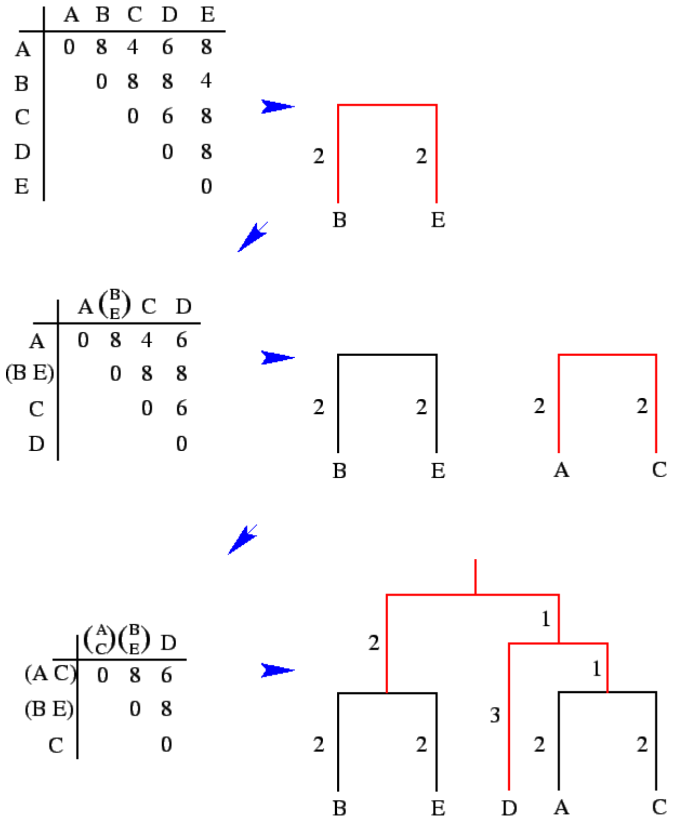
Distance-based methods

UPGMA – unweighted pair group method using arithmetic averages

Distance between two clusters C_i and C_j:

$$d_{ij} = (1/|C_i||C_j|) \sum_{p\in C_i,\, q \in C_j} d_{pq}.$$

Note: it is NOT always possible to interpret pairwise sequence similarity scores as metric distance.



	A	B	C	D	E
A	0	8	4	6	8
B		0	8	8	4
C			0	6	8
D				0	8
E					0

	A	(B E)	C	D
A	0	8	4	6
(B E)		0	8	8
C			0	6
D				0

	(A C)	(B E)	D
(A C)	0	8	6
(B E)		0	8
C			0

Figure: Construction of an ultrametric tree

Algorithm: UPGMA

Initialization:

- Assign each sequence i to its own cluster C_i
- Define one leaf of T for each sequence, and place at height zero

Iteration:

- Determine the two clusters i, j for which d_{ij} is minimal.
- Define a new cluster k by $C_k = C_i \cup C_j$, and define d_{km} for all m
- Define a node k with daughter noes i and j, and place it at height $d_{ij} / 2$.
- Add k to the current clusters and remove i and j.

Termination:

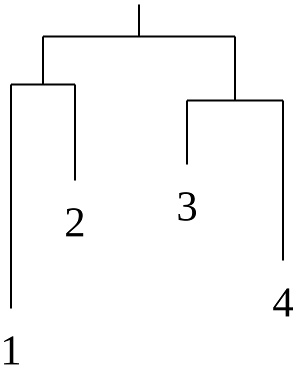
- When only two clusters i, j remain, place the root at height $d_{ij}/ 2$.

Ultrametric: for any triplet (x_i, x_j, x_k), distances d_{ij}, d_{jk}, d_{ki} are either all equal or two are equal and the remaining is smaller.

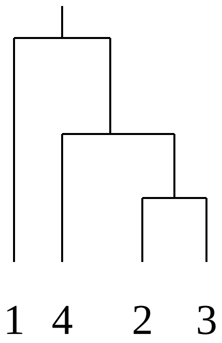
Molecular clock: two siblings evolve at the same constant rate.

Such requirements are often not satisfied, and UPGMA trees then will be not correct.

For example,



Actual tree



Tree reconstructed incorrectly using UPGMA

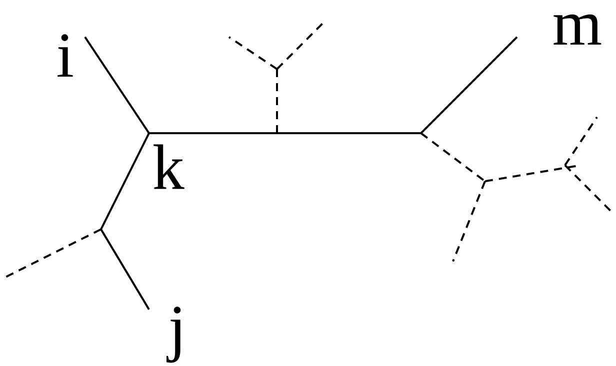
Neighbor-joining:

- Distances are additive.
- Given a pair of leaves, determine if they are neighboring leaves (not necessarily with shortest distance)
- Once we merge a pair of neighboring leaves, how do we compute the distance between this pair (as a whole, called k) and another leaf, called m?

$$\mathbf{\tfrac{1}{2}\,(d_{im} + d_{jm} - d_{ij})}$$

$$= \tfrac{1}{2}\,(d_{ik} + d_{km} + d_{jk} + d_{km} - d_{ik} - d_{jk})$$

$$= \tfrac{1}{2}\,(d_{km} + d_{km}) = d_{km}.$$



Without a tree, how can we know that if two leaves are neighbor (when neighbors do not mean shortest distance)?

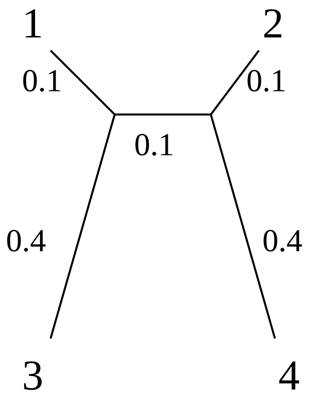
Theorem (Saitou & Nei, 1987**)**: For each leaf i, define r_i as

$$r_i = (1/(|L|-2)) \sum_{k \in L} d_{ik},$$

where L stands for the set of leaves.

Then a pair of leaves i and j will be neighboring leaves if $D_{ij} = d_{ij} - (r_i + r_j)$ is minimal.

Example:



$d_{12} = 0.3$	$D_{12} = -1.1$
$d_{13} = 0.5$	$D_{13} = -1.2$
$d_{14} = 0.6$	$D_{14} = -1.1$
$d_{23} = 0.6$	$D_{23} = -1.1$
$d_{24} = 0.5$	$D_{24} = -1.2$
$d_{34} = 0.9$	$D_{34} = -1.1$

$r_1 = 0.7$

$r_2 = 0.7$

$r_3 = 1.0$

$r_4 = 1.0$

Neighbor joining will generate unrooted trees.

Initialization:

define T to be the set of leaf nodes, one for each given sequence, and put L = T

Iteration:

- Pick a pair i, j in L for which D_{ij} is minimal

- Define a new node k and set $d_{km} = ½(d_{im} + d_{jm} – d_{ij})$ for all m in L.

- Add k to T with edges of lengths $d_{ik} = ½ (d_{ij} +r_i – r_j)$, $d_{jk} = d_{ij} – d_{ik}$.

- Remove i and j from L and add k.

Termination:

When L consists of two leaves I and j, add the remaining edge between i and j, with length d_{ij}.

Pros and Cons of distance-based methods

- Easy to implement, and fast to run
- Robust to minor sequence errors
- Distance-based phylogenetic trees do not generate ancestral sequences
- Definition of “distance” may be problematic