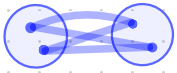
distance-based phylogeny construction algorithms: UPGMA & Neighbor-Joinin

UPGMA ~ "unweighted pair group method using arithmetic averages"

- OTU = operational taxonomic unit
- major assumption for UPGMA: universal biological clock → same mutational/evolutionary rate for all species
- input: • set of species: $\{A_1, A_2, \dots, A_n\}$
  • pairwise distance matrix (D_{ij}) containing distances btwn species (i.e. alignment scores, Jukes-Cantor dist, etc)

⇒ Consider 2 clusters of points (or species) S_i and S_j. We want compute d_{ij}: the distance between the 2 clusters

$$d_{ij} = \frac{1}{|S_i| \cdot |S_j|} \sum_{A \in S_i} \sum_{A' \in S_j} d_{AA'}$$ = average distance between all pairs of sequences from each cluster

ex|  → d_{ij} = avg of the 4 distances

S_i S_j

so now does the algorithm use this distance calculation d_{ij}?

⇒ suppose S_ℓ is the union of S_i and S_j. S_m is another cluster. what's the distance btwn S_ℓ and S_m?

$$d_{\ell m} = \frac{d_{im}|S_i| + d_{jm}|S_j|}{|S_i| + |S_j|}$$ → now did we get this from our distance formula?

$$d_{\ell m} = \frac{1}{|S_i \cup S_j| \cdot |S_m|} \sum_{A \in S_i} \sum_{A' \in S_m} d_{AA'} = \frac{1}{|S_i + S_j| \cdot |S_m|} \left(\sum_{A \in S_i} \sum_{A' \in S_m} d_{AA'} + \sum_{A \in S_j} \sum_{A' \in S_m} d_{AA'} \right) = \frac{1}{|S_i + S_j| \cdot |S_m|} \cdot \frac{|S_i|}{|S_i|} \sum_{A \in S_i} \sum_{A' \in S_m} d_{AA'} + \frac{1}{|S_i + S_j||S_m|} \cdot \frac{|S_j|}{|S_j|} \sum_{A \in S_j} \sum_{A' \in S_m} d_{AA'}$$

$$= \frac{|S_i|}{|S_i + S_j|} \cdot \frac{1}{|S_m| \cdot |S_i|} \sum_{A \in S_i} \sum_{A' \in S_m} d_{AA'} + \frac{|S_j|}{|S_i + S_j|} \cdot \frac{1}{|S_m| \cdot |S_j|} \sum_{A \in S_j} \sum_{A' \in S_m} d_{AA'}$$

$$= \frac{|S_i|}{|S_i + S_j|} \cdot d_{im} + \frac{|S_j|}{|S_i + S_j|} d_{jm} = \boxed{\frac{d_{im}|S_i| + d_{jm}|S_j|}{|S_i + S_j|}}$$

the algorithm:

input: set of sequences $\{A_1, \dots, A_n\}$ and a pairwise distance matrix
output: phylogenetic tree of $A_1, \dots, A_n$

initialization:
- define a cluster for each sequence: $S_k = \{A_k\}$, $1 \le k \le n$
- define a leaf of T for each input sequence A

iteration: # keep doing this step until you have 2 clusters remaining
- Find 2 clusters S_i and S_j for which d_{ij} is minimal
- define a new cluster $S_\ell = S_i \cup S_j$
- compute $d_{\ell m}$ for every other cluster m using
- define a new node ℓ with daughter nodes i and j. Place ℓ at height $\frac{d_{ij}}{2}$
- add ℓ to T and remove i and j

termination: # when there are only 2 clusters remaining (S_i and S_j)
- place root at height $\frac{d_{ij}}{2}$

♥ now, let's turn our attention to the perfect ppt on the website ♥