1.4

Phylogenetic trees

Phylogenetic trees

(1) Phylogeny = relationship between species
- Phylogenetic tree: derivation of evolutionary relationship
- Genome sequences can be used to estimate phylogeny

(2) Genetic phylogeny not coincident with species phylogeny
- Because of events like gene duplication
- Orthologues = genes diverged through speciation
- Paralogues = genes diverged through e.g., gene duplication

(3) Tree has nodes and edges
- Edges have a distance that indicates the amount of change between species/sequences
- Edge length does not necessarily correspond exactly to evolutionary time periods (different change rates)

UPGMA clustering

(1) Computes binary tree from set of leafs and distances
- Building pairs of nearest nodes or node clusters
- Assume that distance to all leaves is the same (constant molecular clock)

(2) Initialization
- Assign each sequence i to its cluster C_i
- Define a leaf at height 0 for each sequence i

(3) Iteration while there is more than one cluster
- Find two clusters with minimal distance (average between all possible pairs)
- Join the clusters as C_k and calculate its distance to all others
- Place the new node k at the the height of half the cluster distance

Distance measures

(1) Based on alignment of sequences
- Fraction f of positions that differ
- A random alignment gives about f=0.75

(2) More realistic estimate
- Jukes-Cantor distance = $-0.75 \log (1 - 4f/3)$
- Approaches infinity as f goes towards 0.75

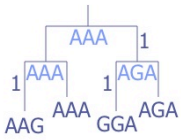
(3) UPGMA assumes additive distance
- Distance between any leaves is sum of paths conneting them
- Automatically constructed by the algorithm

Parsimony

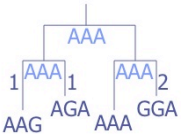
(1) Build a tree that minimizes the number of substitutions

- Enumerate all possible trees (exponential)
- Generate trees heuristically (until good enough)

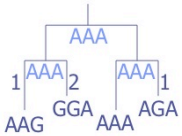
Example:
AAG
AAA
GGA
AGA



cost = 3



cost = 4



Estimating cost of tree

(1) Traditional parsimony

- Cost of one replaced letter is 1

(2) Walk recursively down the tree

- Keep minimal costs C and list of minimal-cost residues R_k at each node
- Start with the root node k = 2n – 1 and C = 0

(3) Recursion for R_k and C

- If k is a leaf:
  R_k = {assigned sequence at k}
- Otherwise:
  Compute R_i and R_j for the daughter nodes i and j
  If $R_i \cap R_j$ is empty:
  $R_k = R_i \cup R_j$;
  increment C;
  Else:
  $R_k = R_i \cap R_j$

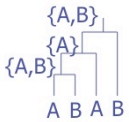
Traceback procedure

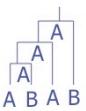
(1) To assign possible residues to each node

- Pick one of the minimal-cost root assignments
- Go down the tree
- Pick either the same assignment for the daughter nodes if possible
- Otherwise pick any of the minimal-cost assignments of this node

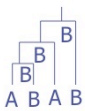
(2) Not all possible assignments can be recovered

- An additional cost down the tree can be recovered higher up
- Can be solved by keeping a list of residues at each node that have a cost of 1 more than the minimum





Recoverable



Non-recoverable, same cost