Phylogenetic trees

Parsimony

Outline

- Parsimony approach to phylogenetic tree estimation
- Fitch's algorithm (unweighted parsimony)

Phylogenetic Tree Approaches

- three general types
  - *distance*: find tree that accounts for estimated evolutionary distances
  - *parsimony*: find the tree that requires minimum number of changes to explain the data
  - *probabilistic-model based*: find the tree that maximizes the likelihood of the data

Parsimony Based Approaches

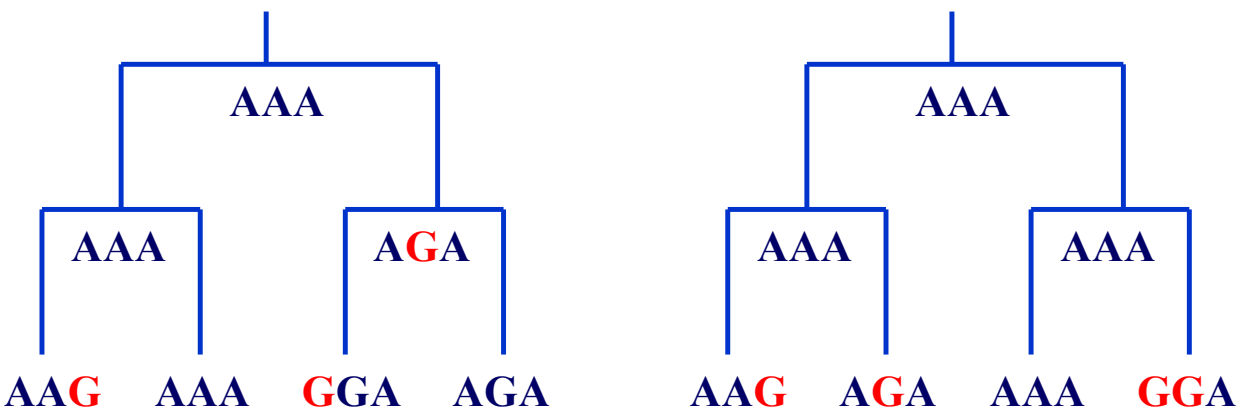
given: character-based data

do: find tree that explains the data with a minimal number of changes

- focus is on finding the right tree topology, not on estimating branch lengths

Parsimony Example

- there are various trees that could explain the phylogeny of the sequences AAG, AAA, GGA, AGA including these two:



- parsimony prefers the first tree because it requires fewer substitution events

Parsimony Based Approaches

- usually these approaches involve two separate components
  1. a procedure to find the minimum number of changes needed to explain the data (for a given tree topology)
  2. a search through the space of trees

direct explanation

Finding Minimum Number of Changes for a Given Tree

- Basic assumptions:
  - any character state (e.g. a particular base or amino acid) can convert to any other character state
  - the "cost" of any single character state change is the same, regardless of the particular states
  - positions are independent
    - Thus, we can compute the minimum number of changes for each position separately

Finding Minimum Number of Changes for a Given Tree

- Brute force method:
  - For each possible assignment of states to the internal nodes
    - Calculate number of changes for that assignment
  - Report the minimum number of changes found

- Runtime: $O(Nk^N)$
  - k = number of possible character states (e.g., 4 for DNA)
  - N = number of leaves

Fitch's Algorithm

Fitch's algorithm [1971] :

1. traverse tree from leaves to root determining set of possible *states* (e.g. nucleotides) for each internal node

2. traverse tree from root to leaves picking ancestral states for internal nodes

Fitch's Algorithm: Step 1
Possible States for Internal Nodes

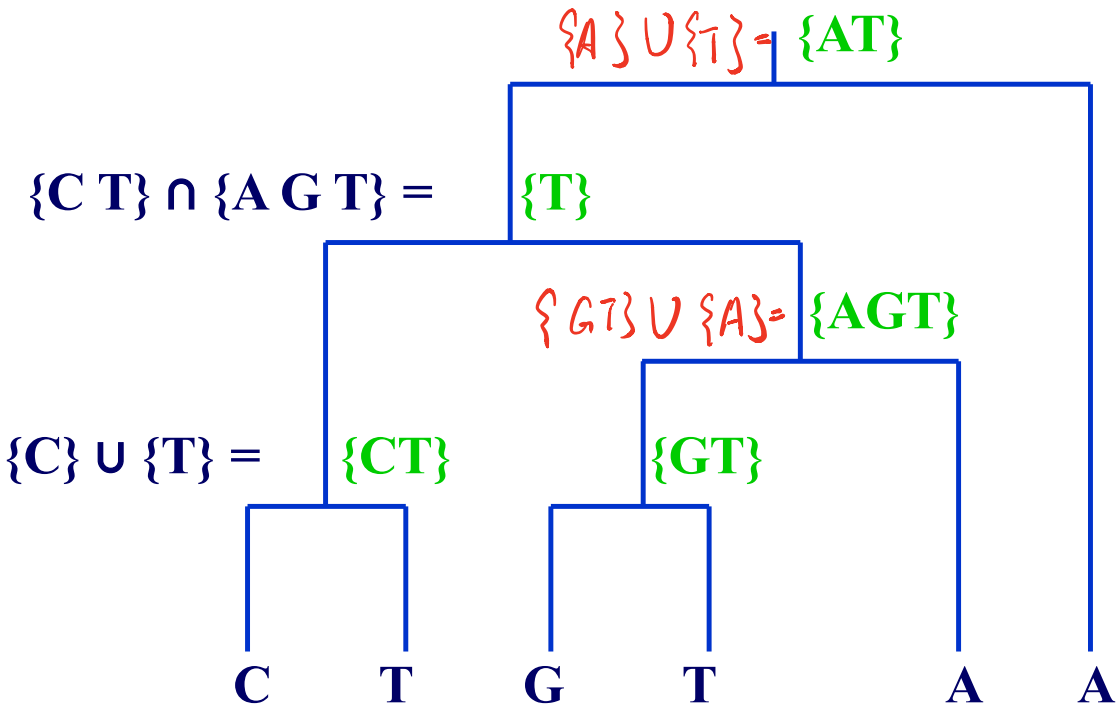
- do a *post-order* (from leaves to root) traversal of tree
- determine possible states R_i of internal node i with children j and k

$$R_i = \begin{cases} R_j \cup R_k, & \text{if } R_j \cap R_k = \varnothing \\ R_j \cap R_k, & \text{otherwise} \end{cases}$$

- this step calculates the number of changes required

  # of changes = # union operations

Fitch's Algorithm: Step 1 Example



Fitch's Algorithm: Step 2
Select States for Internal Nodes

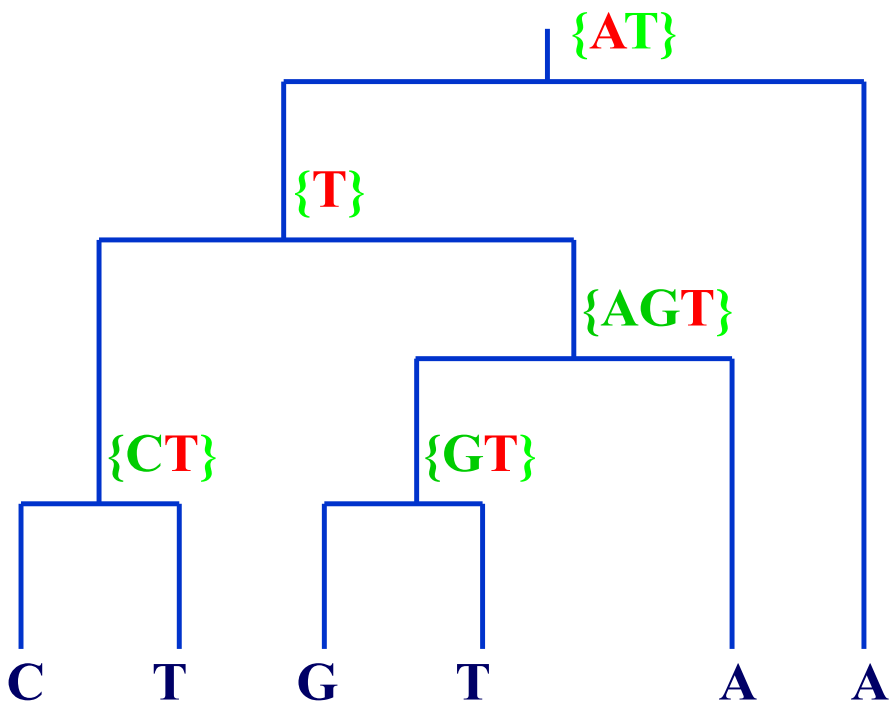
- do a *pre-order* (from root to leaves) traversal of tree
- select state r_j of internal node j with parent i

$$r_j = \begin{Bmatrix} r_i, \text{ if } r_i \in R_j \\ \text{arbitrary state} \in R_j, \text{ otherwise} \end{Bmatrix}$$

i

j

Fitch's Algorithm: Step 2



Summary

- Parsimony approach for tree estimation
  - minimize number of changes along tree required to explain data
- Fitch's algorithm
  - Computes parsimony score of a tree efficiently
  - Two stages
    - Postorder traversal for possible state set computation
    - Preorder traversal for selection of ancestral states