BMI/CS 576 – Day 16

- Today
  - Parsimony
- Next week
  - Genome annotation
    - Markov chains and hidden Markov models

Upcoming events

- Midterm grades back on Friday
- HW3 posted soon (likely tomorrow)
- Mid-semester survey out soon

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

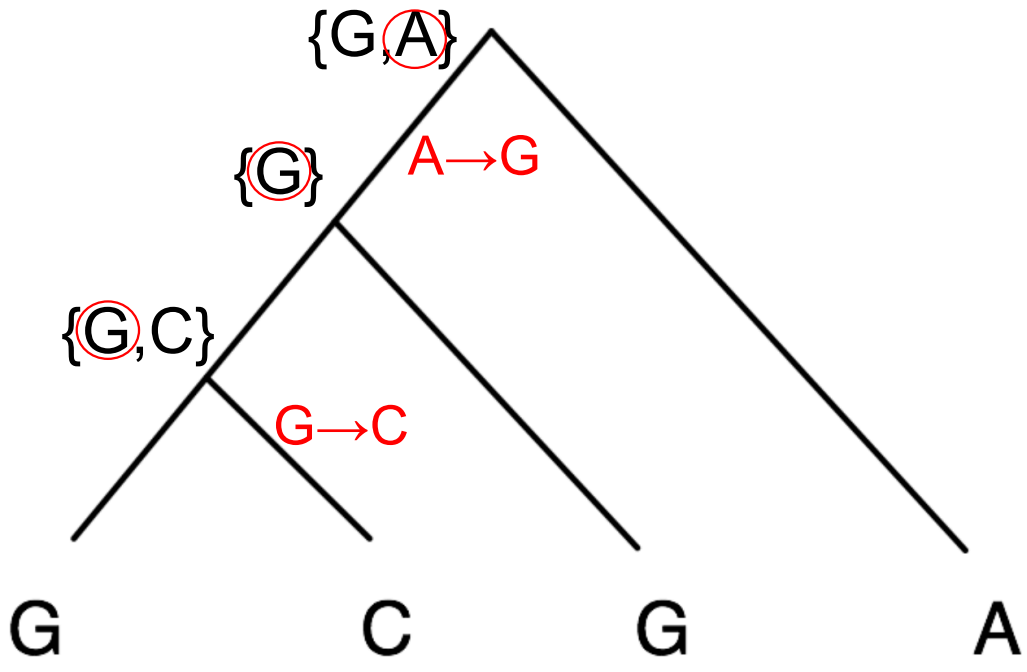
Fitch's Algorithm: Step 2
Select States for Internal Nodes

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

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

Quiz

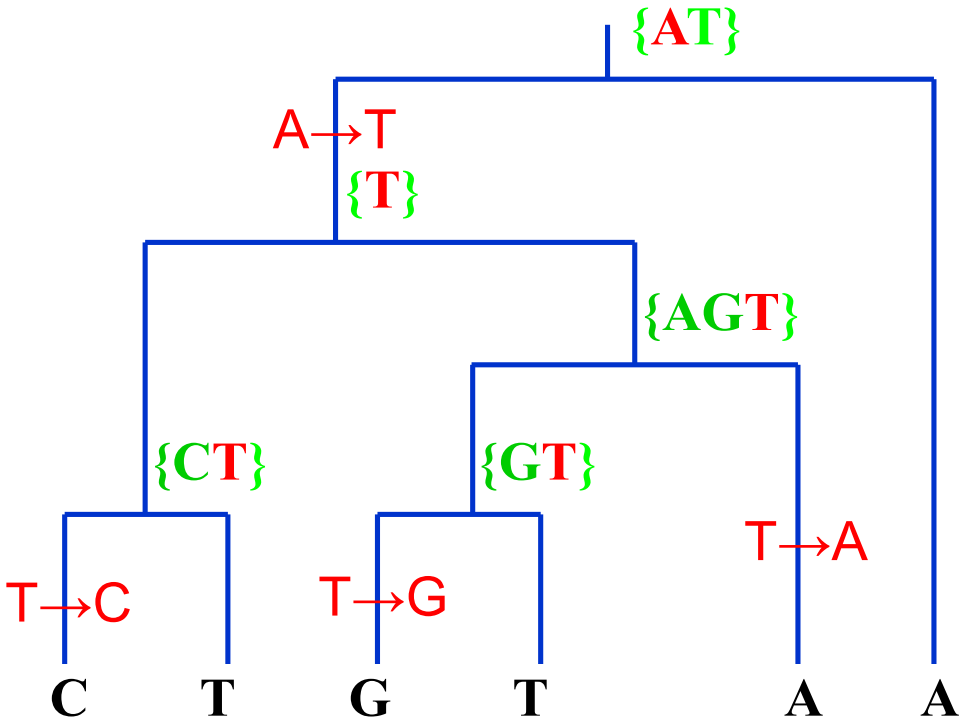
Given this tree, what is the minimum number of changes required to explain these characters?



At least **2** changes are required

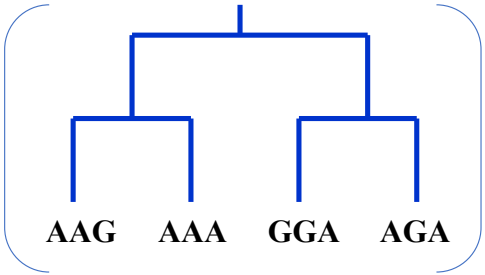
(can also select G at root node to give same number of changes)

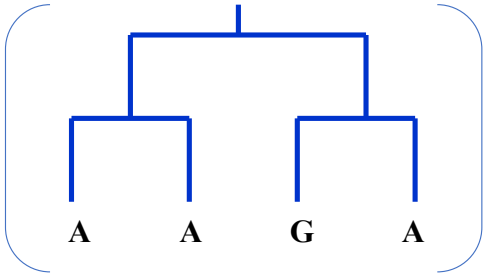
Fitch's Algorithm: Step 2



4 changes to explain characters at leaves in this example

Parsimony scoring with longer sequences at leaves

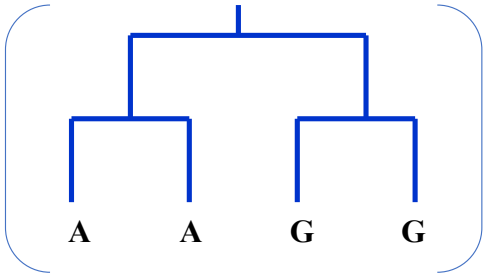
score 

= score 

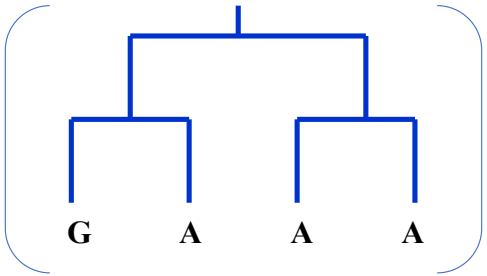
(column 1)

Multiple alignment:

AAG
AAA
GGA
AGA

+ score 

(column 2)

+ score 

(column 3)

Searching through tree space for parsimony methods

- Heuristic greedy search
  - Start with an arbitrary tree
  - Move around tree space via nearest-neighbor interchanges to improve score
  - Stop when no move can improve score
- Branch and bound
  - Grow trees starting from partial tree with three leaves in all possible ways (branch)
  - Stop growing a given partial tree if the lower bound on the score of a full tree that contains it is larger than the best score of a full tree seen thus far (bound)

Searching through tree space for parsimony methods

- Heuristic greedy search
  - Not guaranteed to find optimal solution
  - But generally fast
- Branch and bound
  - Guaranteed to find optimal solution
  - In worst case, exponential time
  - Three variants given in lecture
    - All are guaranteed to find optimal solution
    - Vary in memory, time requirements

NJ in day 15 notebook

- You may get ties in the last two iterations
  - Any choice should get you the same unrooted tree
- Testing code was improved to ensure that branch lengths were checked correctly
  - Try submitting again if your branch lengths were marked as incorrect