CMSC423: Bioinformatic Algorithms, Databases and Tools
Lecture 12

Phylogenetic trees

Phylogenetic tree display

Phylogenetic analysis

Sankoff's algorithm

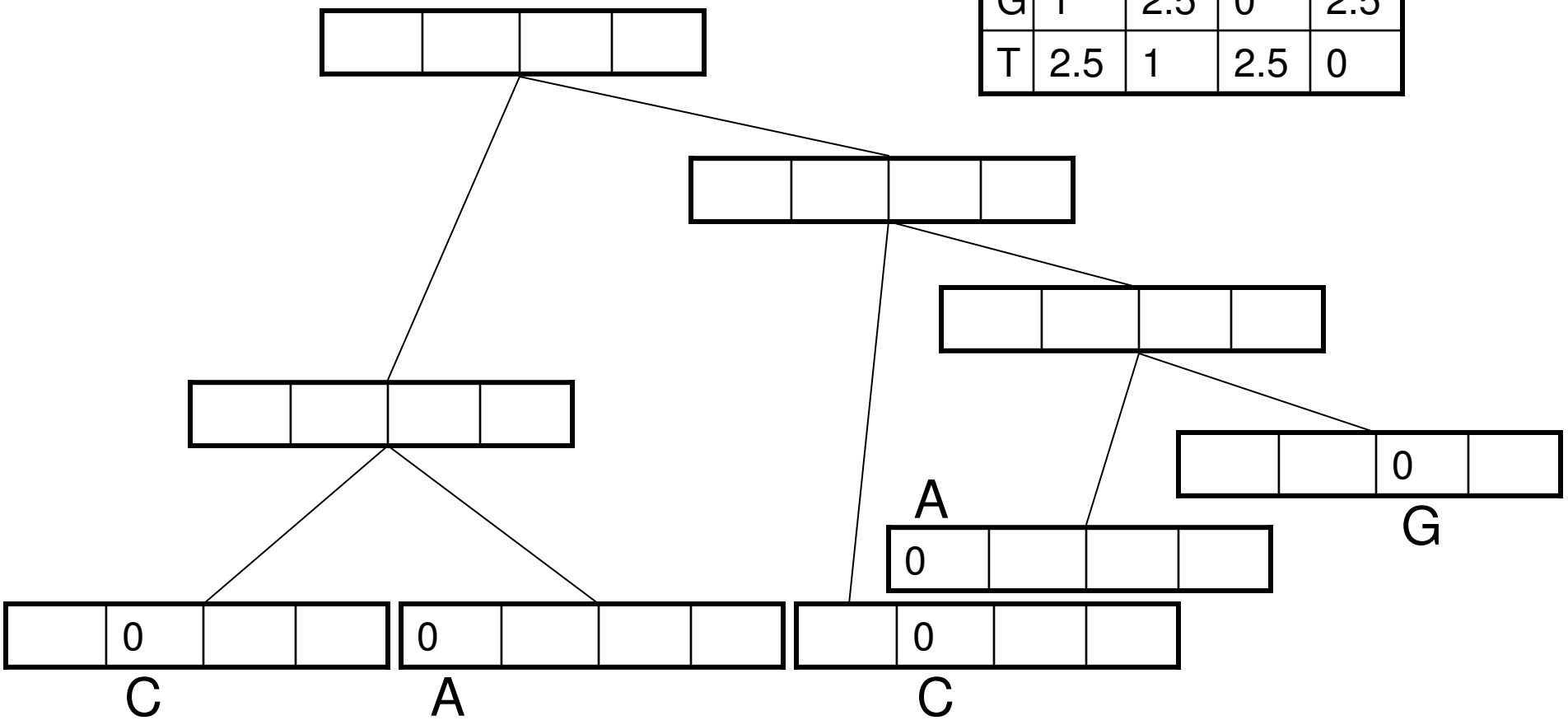
- At each node v in the tree store s(v,t) – best parsimony score for subtree rooted at v if character stored at v is t
- Traverse the tree in post-order and update s(v,t) as follows
  - assume node v has children u and w
  - $s(v,t) = \min_i \{s(u,i) + score(i,t)\} + \min_j \{s(w,j) + score(j,t)\}$
- Character at root will be the one that maximizes s(root, t)

- Note – this solves the weighted version. For unweighted set score (i,i) = 0, score(i,j) = 1 for any i,j

Sankoff's algorithm - example

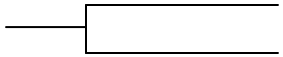
$$s(v,t) = \min_i \{s(u,i) + score(i,t)\} + \min_j \{s(w,j) + score(j,t)\}$$

	A	C	G	T
A	0	2.5	1	2.5
C	2.5	0	2.5	1
G	1	2.5	0	2.5
T	2.5	1	2.5	0



Trees as clustering

- Start with a distance matrix – distance (e.g. alignment distance) between any two sequences (leaves)
- Intuitively – want to cluster together the most similar sequences
- UPGMA – Unweighted Pair Group Method using Arithmetic averages
  - Build pairwise distance matrix (e.g. from a multiple alignment)
  - Pick pair of sequences that are closest to each other and cluster them – create internal node that has the sequences as children
  - Repeat, including newly created internal nodes in the distance matrix



  - Key element – must be able to quickly compute distance between clusters (internal nodes) – weighted distance

$$D(cl_1, cl_2) = \frac{1}{|cl_1||cl_2|} \sum_{p \in cl_1, q \in cl_2} D(p, q)$$

Trees as clustering

- Note that UPGMA does not estimate branch lengths – they are all assumed equal
- Neighbor-joining
  - distance between two sequences is not sufficient – must also know how each sequence compares to every other sequence
  - NJdist(i,j) = D(i,j) – (r_i + r_j) -r_i, r_j correction factors

$$r_i = \frac{1}{m-2}\sum_k D(i,k)$$

- Pick two nodes with NJdist(i,j) minimal
  - Create parent k s.t.
  - D(k, m) = 0.5 (D(i,m) + D(j,m) – D(i,j)) for every other node m
  - D(i, k) = 0.5 (D(i,j) + r_i – r_j) - length of branch between i &k
  - D(j, k) = 0.5 (D(i,j) + r_j – r_i) – length of branch between j & k

Trees as clustering

- Note that both UPGMA and NJ assume distance matrix is additive: $D(i,j) + D(j,k) = D(i,k)$ - usually not true but close
- Also, NJ can be proven to build the optimal tree!

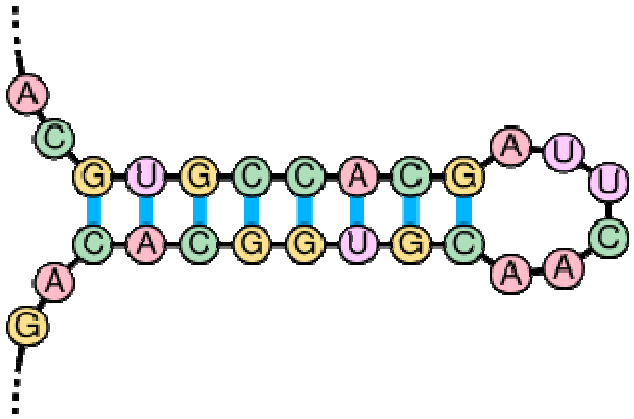
- But, simple alignment distance is not a good metric

Maximum likelihood

- For every branch S->T of length t, compute P(T|S,t) – likelihood that sequence S could have evolved in time t into sequence T
- Find tree that maximizes the likelihood

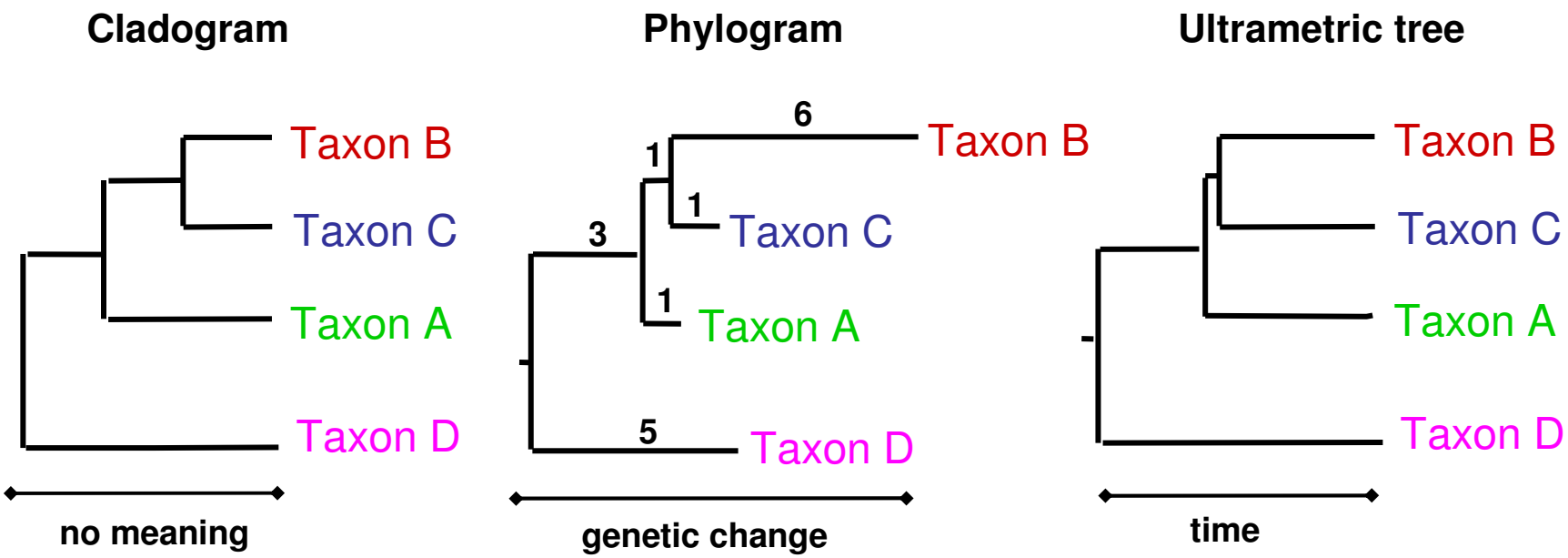
- Note that likelihood of a tree can be computed with an algorithm similar to Sankoffs
- However, no simple way to find a tree given the sequences – most approaches use heuristic search techniques
- Often, start with NJ tree – then "tweak" it to improve likelihood

From multiple alignment to tree




```
GUGCCACGAUUCA-A---CGUGG-CAC
GUGCC-CGAGGCAUAGGCCG-G-UCAC
GUUCCACG-U--U--G-CCGUGG-AAC
GUGCC-GGAUU--UGCAGCC-GG-CAC
```

Three types of trees



All show the same evolutionary relationships, or branching orders, between the taxa.

from www.albany.edu/faculty/cs812/StewartTalk2.ppt