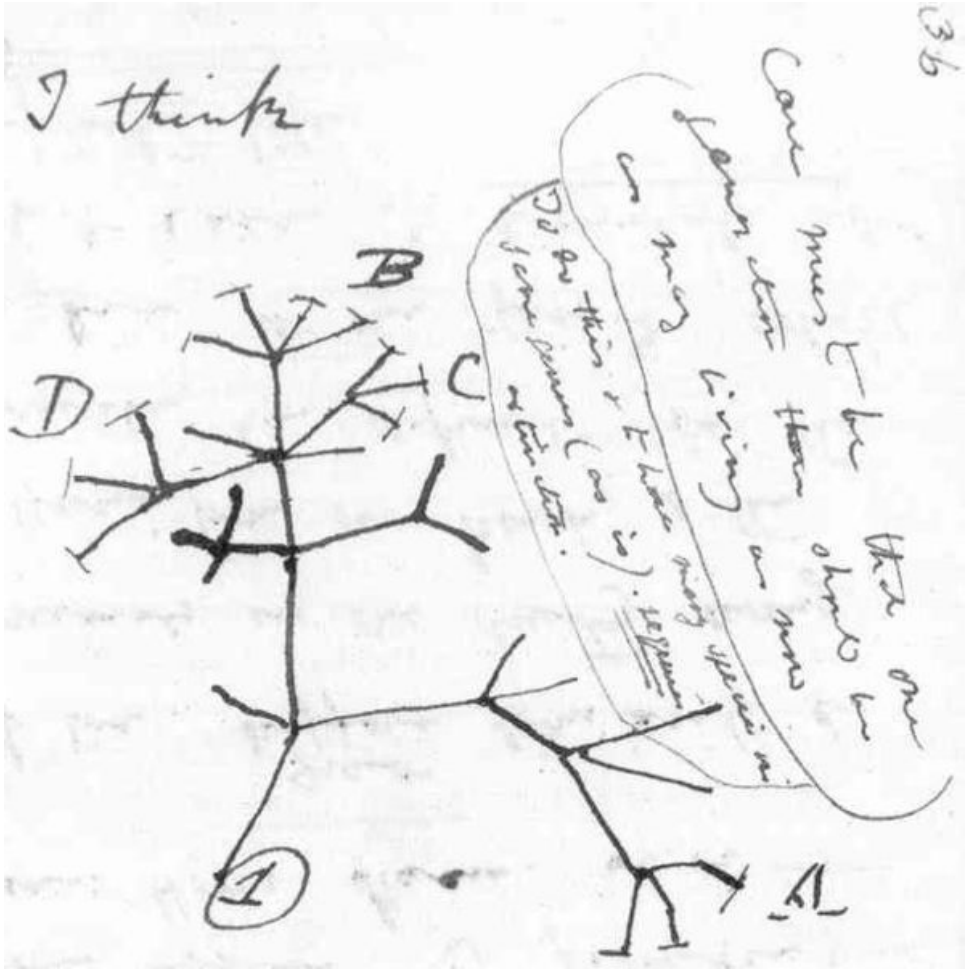
CMSC423: Bioinformatic Algorithms, Databases and Tools

Phylogenetic trees

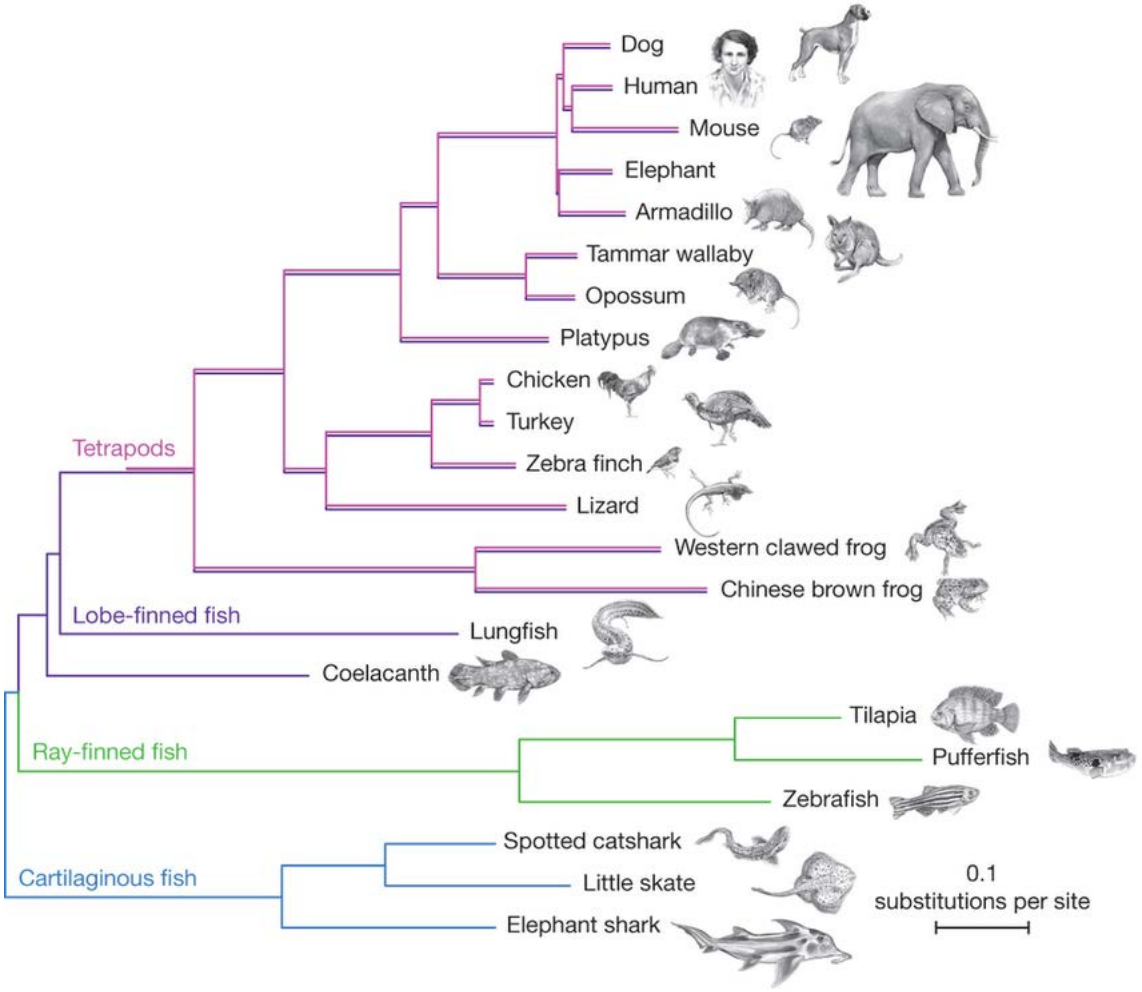
First evolutionary tree…

a.k.a phylogenetic tree

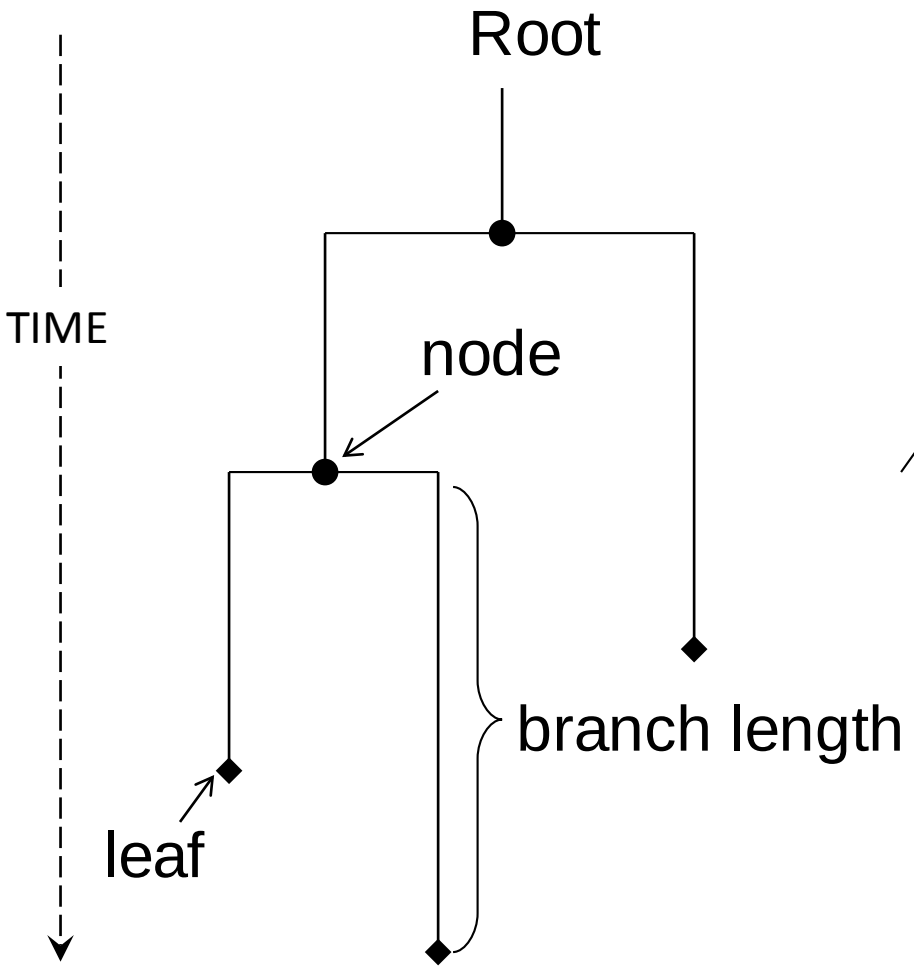


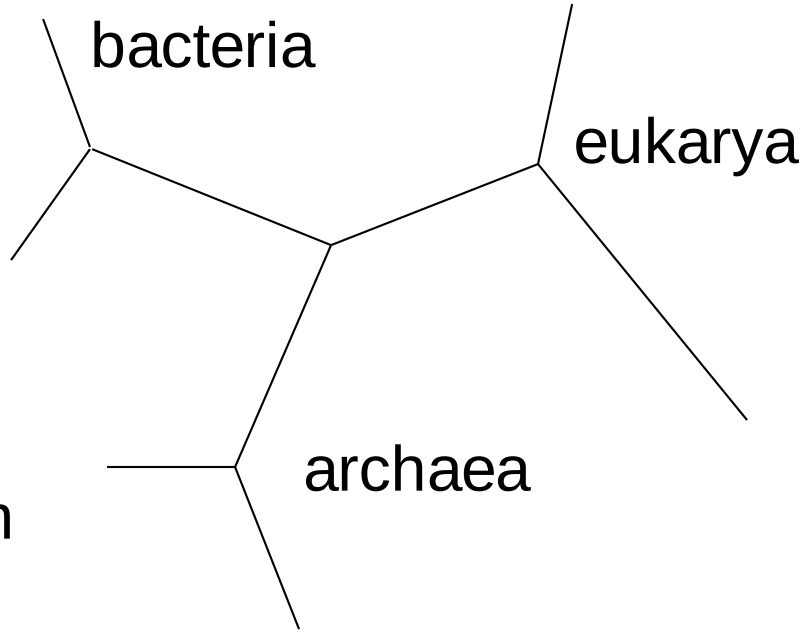
Phylogenetic trees – how evolution works

http://www.tolweb.org/tree/ - the tree of life



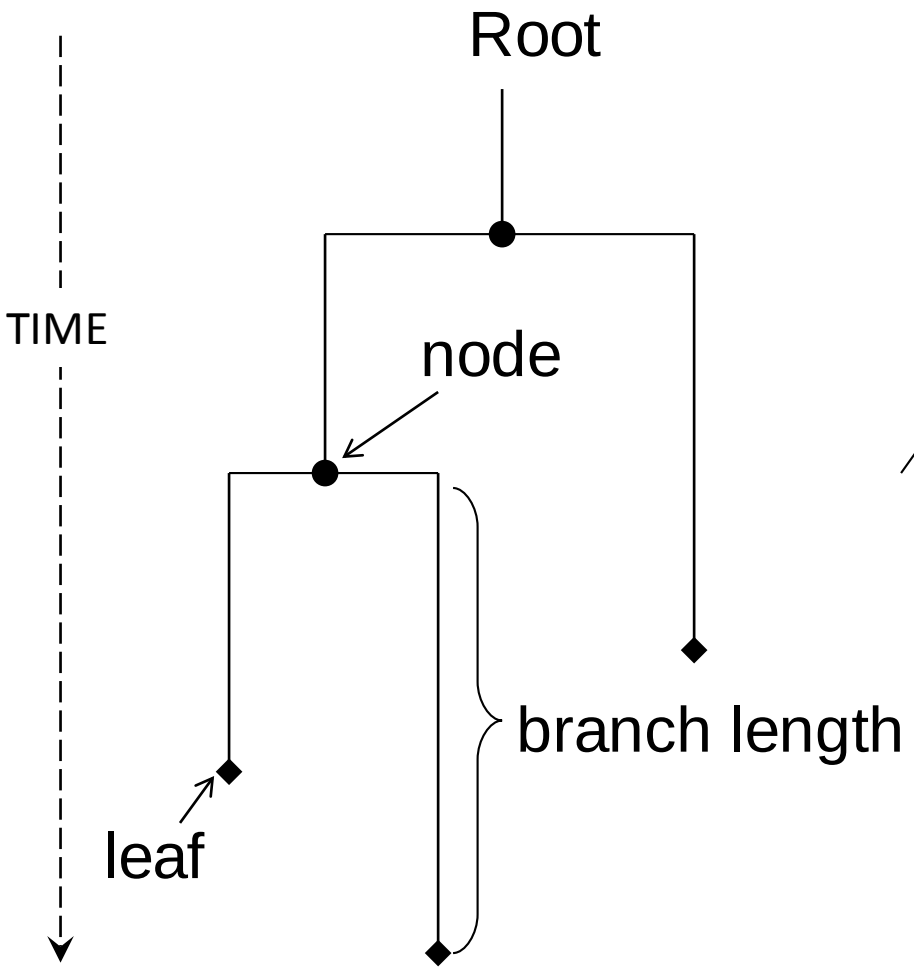
Anatomy of a tree

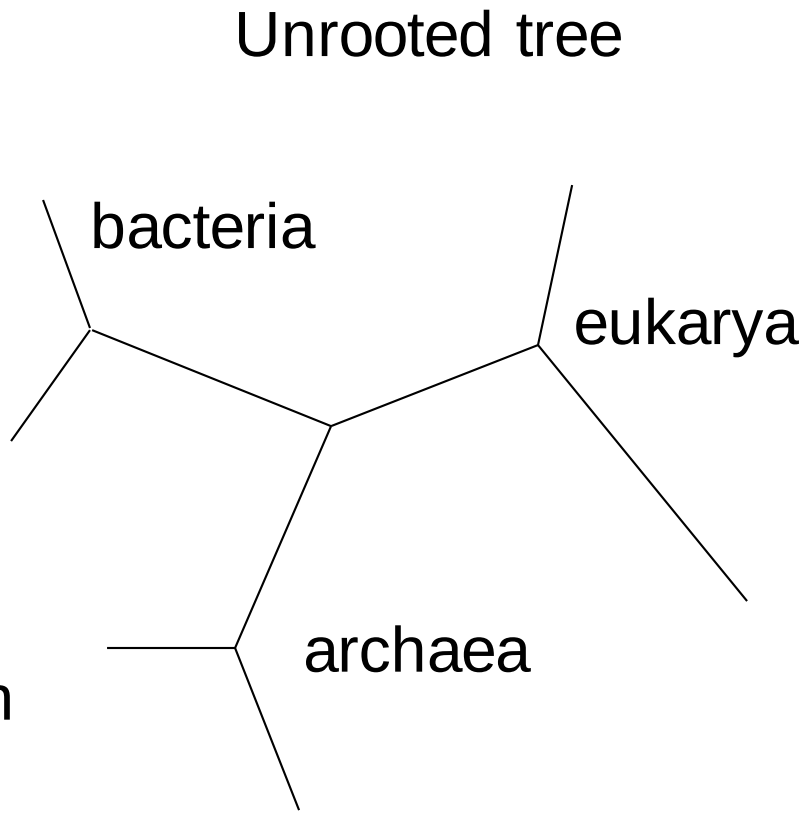




Connected and Acyclic

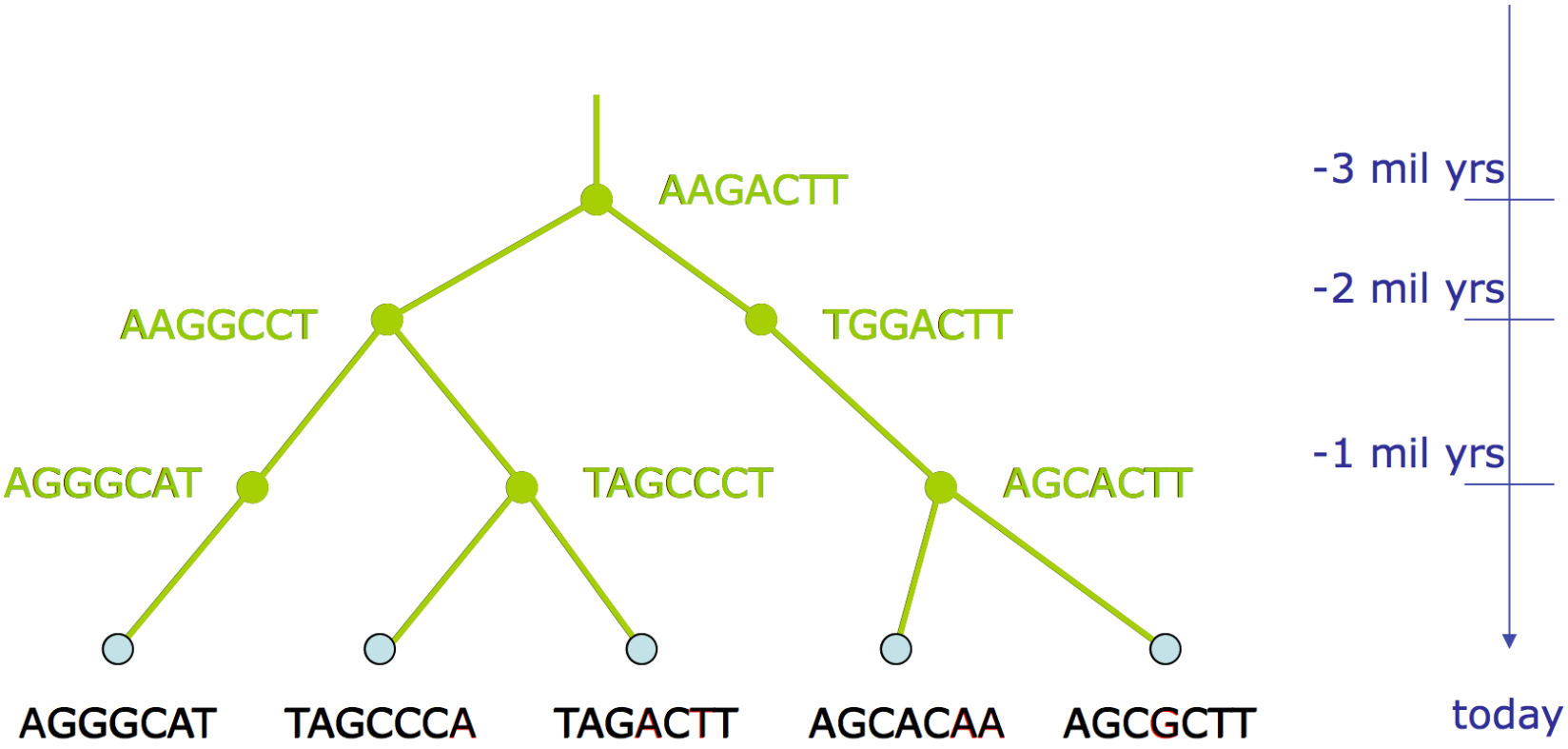
Anatomy of a tree



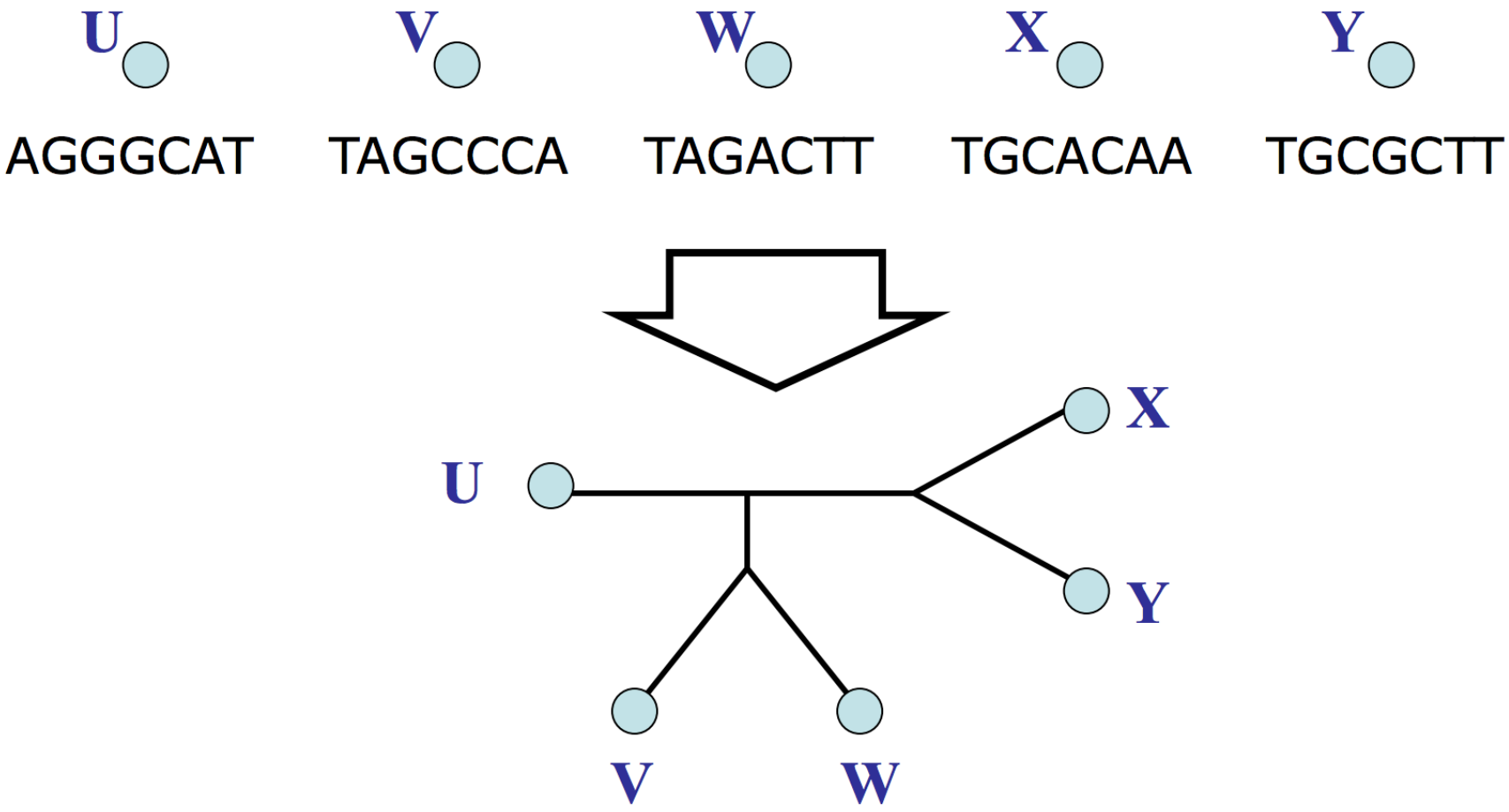


Phylogenetic trees are usually binary (though they don't have to)

DNA sequence evolution



Phylogeny problem



Phylogeny estimation methods

- Distance-based methods
- Maximum parsimony
- Maximum Likelihood
- Bayesian MCMC

Distance-Based Phylogeny Problem

Reconstruct an evolutionary tree fitting a distance matrix

Input: A distance matrix

Output: A tree fitting this distance matrix

Distance matrix

Distance matrix

- Symmetric (for all i,j $D_{i,j} = D_{j,i}$)
- Non-negative
- satisfy triangle's inequality (for all i, j, and k, $D_{i,j} + D_{j,k} >= D_{i,k}$)

Species	Alignment	Distance Matrix			
		Chimp	**Human**	**Seal**	**Whale**
Chimp	ACGTAGGCCT	0	3	6	4
Human	ATGTAAGACT	3	0	7	5
Seal	TCGAGAGCAC	6	7	0	2
Whale	TCGAAAGCAT	4	5	2	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
- It assumes an ultrametric tree in which the distances from the root to every branch tip are equal

Trees as clustering

•UPGMA – Unweighted Pair Group Method using Arithmetic averages

•It assumes an ultrametric tree in which the distances from the root to every branch tip are equal.

–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) where r_i, r_j are correction factors

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

Neighbor joining

- 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

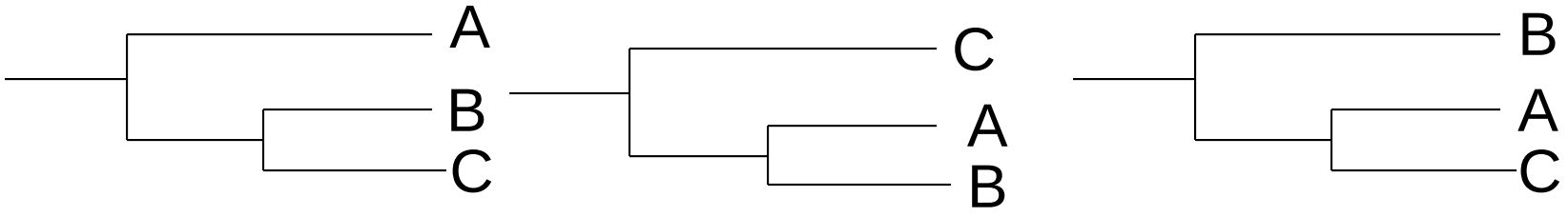
- Note that both UPGMA and NJ assume distance matrix is additive.
- (Recall – Additive matrix is when lengths of all edges along the path between leaves i and j in a tree fitting the matrix D add to $D_{i,j}$)
- Also, NJ can be proven to build the optimal tree!

Character-based phylogeny questions

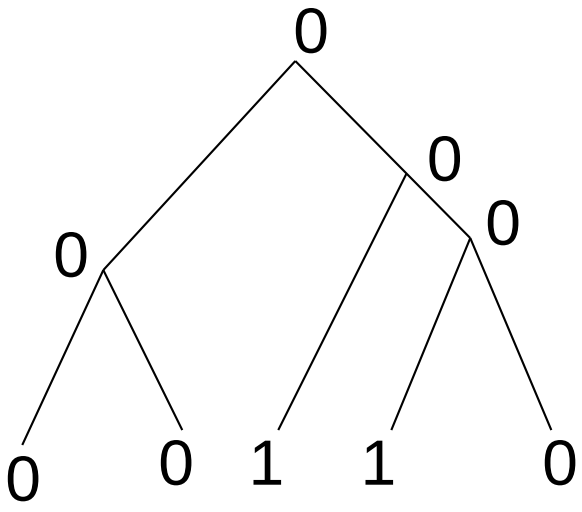
•Given several organisms & a set of features (usually sequence, but also morphological: wing shape/color...)

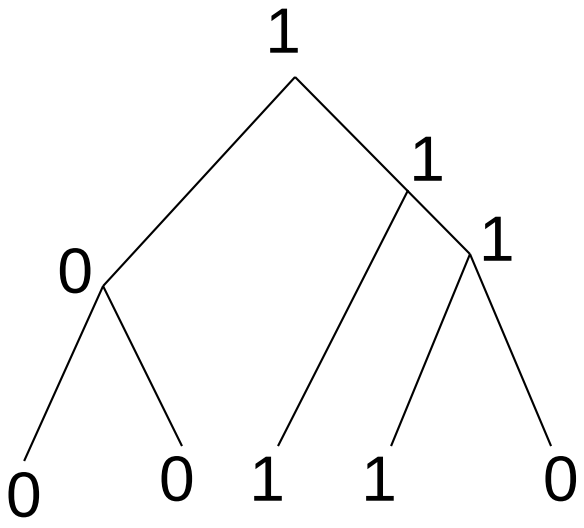
•A. Given a phylogenetic tree – figure out what the ancestors looked like (what are the features of internal nodes)

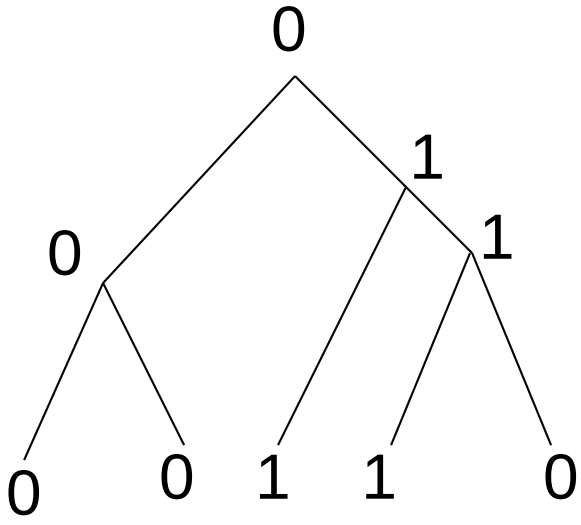
•B. Find the phylogenetic tree that best describes the common evolutionary heritage of the organisms



Example

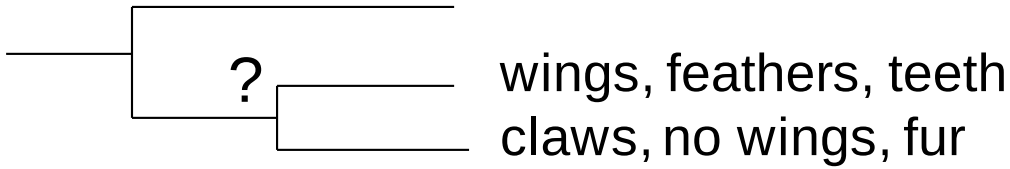




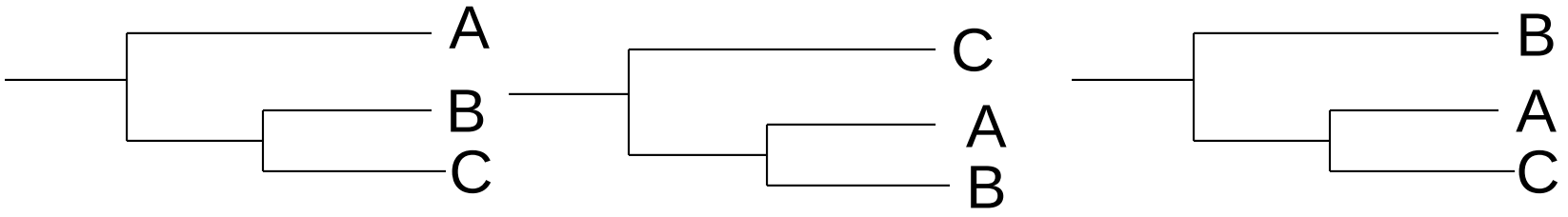


Think about these before next class…

•A. Given a phylogenetic tree – figure out what the ancestors looked like (what are the features of internal nodes)



•B. Find the phylogenetic tree that best describes the common evolutionary heritage of the organisms



Thanks!

Contact:
nidhi@cs.umd.edu

Phylogeny questions

- A. Easy-ish – can be done with dynamic programming
- B. Hard – Many possible trees

$$\frac{(2n-3)!}{2^{n-2}(n-2)!}$$

rooted trees with n leaves

Scoring a tree – Sankoff's algorithm

- Assumption – we try to minimize # of state changes from root to leaves – Parsimony approach
- Small parsimony
  - given a tree where leaves are labeled with m-character strings
  - find labels at internal nodes s.t. # of state transitions is minimzed
- Weighted small parsimony
  - same as parsimony except that state transitions are assigned weights
  - minimize the overall weight of the tree

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

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