Week 10
Phylogenetic Trees

Paul Chew
CS 426
Fall 2003

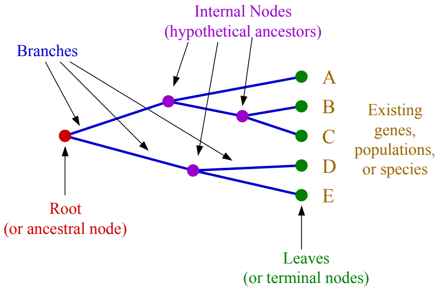
"Tree of Life"

- Through evolution, new species have split off from existing ones
- A key goal of evolutionary biology: reconstruct history of speciation events (i.e., build *phylogenetic trees*)
- Phylogenetic trees have been constructed for years using *morphological* (i.e., physical) features
- Increasing availability of DNA sequence data has led to wider interest in such trees

Fig. 251. Genealogical tree of plant life

This diagram is intended to suggest the common origin of all plant forms, with the constant progressive departure from ancestral types, now in one direction and now in another, like the branching of a tree. *Lower* and *higher* mean nearer to or farther from the original types. The closer together two forms are on a given branch, the more closely related they are considered (cf. Fig. 252)

Tree Terminology

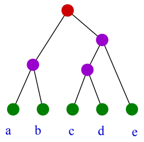


An Algorithm for Phylogenetic Trees

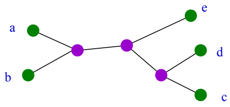
- Input
  - A set of n species
  - A method for computing a score for a labeled tree
- Output
  - The labeled phylogenetic tree with the optimal score
- Algorithm (impractical)
  - Determine score for each possible labeled tree
  - Report labeled tree with best score
- Difficulty: there are too many possible labeled trees
  - For rooted binary trees with n labeled leaves, there are (2n-3)!! distinct trees
  - "!!" is special notation for "like factorial but skip every other number"
  - Example: For 5 leaves there are (7)(5)(3)(1) = 105 distinct rooted trees

Rooted vs. Unrooted Trees

- Rooted Trees
  - A rooted binary tree with n leaves has 2n-2 edges and n-1 internal nodes

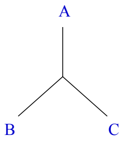


- Unrooted Trees
  - An unrooted binary tree (think of the root and its two edges combining to become a single edge) on n leaves has 2n-3 edges and n-2 internal nodes

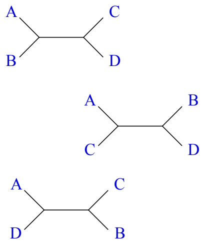


Counting Unrooted Trees (small n)

- If there are 3 labeled leaves then there is just one possible unrooted tree



- If there are 4 labeled leaves there are 3 different unrooted trees



Counting Unrooted Trees (any n)

- Let U(n) be the number of unrooted trees with n labeled leaves
- Given an unrooted tree with n leaves, an extra leaf can be added on any branch to make a tree with (n+1) leaves
- n leaves

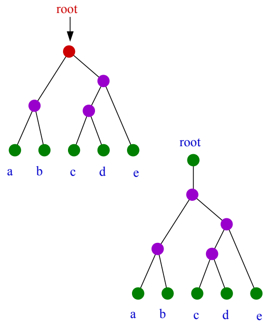
  ⇒ 2n-3 possible branches

  ⇒ U(n+1) = (2n-3)U(n)

  ⇒ U(n) = (2n-5)!!

n	U(n)
3	1
4	3
5	15
6	105
7	945
8	10935
9	135135
10	2027025
...	...
30	3.58×10^{36}

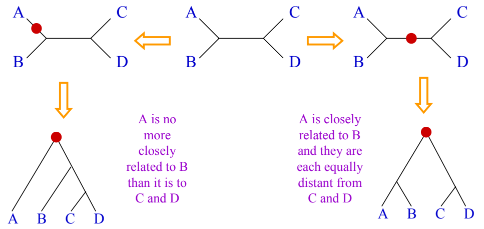
Counting Rooted Trees



- The root is a special node
- If we want to though, we can look at it as just another leaf (labeled *root*)
  - A rooted tree with n leaves corresponds to an unrooted tree with n+1 leaves
  - Thus there are (2n-3)!! rooted trees with n leaves

Usually Want Rooted Trees

- A single unrooted tree can imply different relationships between species depending on the location of the root



A is no more closely related to B than it is to C and D

A is closely related to B and they are each equally distant from C and D

Data for Phylogenetic Trees

- Characters

Species	Characters
A	ACTGTTCGTTCTGA
B	ACCGTTCCTTCTAG
C	CCTGTTGCTTCTGA
D	ACTGTCCCTTCTAG

or

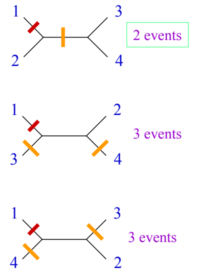
Species	webbed feet	round eggs	beak
A	1	0	2
B	0	1	1
C	1	0	2
D	0	1	0

- Distances

	A	B	C	D
A	--	0.75	0.35	0.27
B	0.75	--	0.85	0.33
C	0.35	0.85	--	0.31
D	0.27	0.33	0.31	--

Parsimony (Character Based)

- The most parsimonious tree is the one that requires the fewest evolutionary events
- Example:
  - 1: AC
  - 2: TC
  - 3: TG
  - 4: TG



The Small Parsimony Problem

- Given a labeled tree, we can determine the most parsimonious assignment of characters to the ancestor nodes
- Note that we need only examine one character at a time (i.e., we determine the solution for position 1, then we work on position 2, etc.)

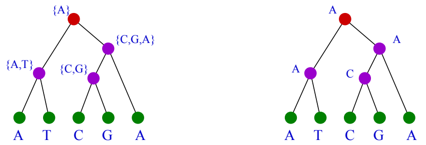
S_N represents a *set* of character values chosen for node N

```
for each node N (using postorder):
    Let L and R be N's children
    if $S_L \cap S_R = \varnothing$: $S_N = S_L \cup S_R$
    else: $S_N = S_L \cap S_R$
```

c_N represents the *single* character value chosen for node N

```
for each node N (using preorder):
    Let P be N's parent
    if $c_P$ in $S_N$: $c_N = c_P$
    else: $c_N$ = any c in $S_N$
```

Small Parsimony Problem Example



The Large Parsimony Problem

- Input
  - A matrix M describing m characters for n species
- Output
  - The most parsimonious phylogenetic tree
- This problem is NP-hard
- Various heuristics are used (with some success)
  - But results are often not known to be optimal
- Can solve *small parsimony problem*: for m characters, each with k possible values, and for n species
  - O(kmn) time to determine character assignment
- We can evaluate a given tree, but we don't know which tree to use!

UPGMA (Distance Based)

- UPGMA (Unweighted Pair Group Method with Arithmetic mean)
- Input is a distance matrix showing distances between species
- Idea is to combine the two "closest" species, then iterate until we reach a single cluster
- Distance between two species clusters C and D is defined as $d(C,D) = [\Sigma_{p\in C}\Sigma_{q\in D}d(p,q)] / |C||D|$
- If clusters D' and D'' are combined to make D then can show d(C,D) = weighted average of d(C,D') and d(C,D'')

UPGMA Algorithm

- Initialization:
  - Assign each species to its own cluster C_i
  - Each such cluster is a tree leaf
- Iteration:
  - Determine i and j so that $d(C_i,C_j)$ is minimal
  - Define a new cluster $C_k = C_i \cup C_j$ with a corresponding node at height $d(C_i,C_j)/2$
  - Update distances to C_k using weighted average
  - Remove C_i and C_j
- Termination:
  - Halt when just a single cluster remains

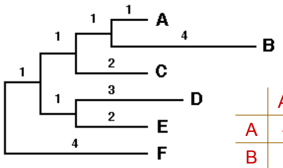
UPGMA Example

	A	B	C	D	E
A	--				
B	63	--			
C	94	79	--		
D	111	96	47	--	
E	67	23	83	100	--

	A	C	D	BE
A	--			
C	94	--		
D	111	47	--	
BE	65	$\boxed{81}$	98	--

UPGMA Can be Fooled

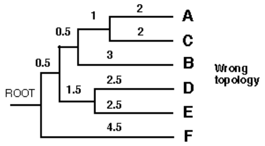
- Example (from http://www.icp.ucl.ac.be/~opperd/private/upgma.html)



	A	B	C	D	E	F
A	--					
B	5	--				
C	4	7	--			
D	7	10	7	--		
E	6	9	6	5	--	
F	8	11	8	9	8	--

Fooling UPGMA (Cont'd)

- The failure of UPGMA here is caused by unequal rates of mutation
- UPGMA is based on the assumption that all species have the same mutation rate

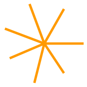


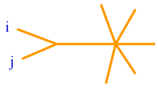
When Does UPGMA Succeed?

- UPGMA always produces an *ultrametric tree*
  - Thus the UPGMA algorithm produces the correct result only when the distance matrix corresponds to an ultrametric tree
  - Since mutation rates are *not* the same for all species, UPGMA will sometimes produce a wrong tree
- A tree is an *ultrametric tree* if the edges can be labeled (with distances) so that all leaves are the same distance from the root
  - In other words, all species must be the same "evolutionary distance" from the root

Neighbor Joining (Distance Based)

- Intuition
  - Start with all species in a simple star-shaped tree
    - Can show cost of this tree is $[\Sigma_{i<j} d(i,j)] / (n-1)$
  - Determine the least-cost tree among all trees with (slightly) better topology
    - Can show cost of this tree is $d(i,j) – u_i – u_j + C$ where u_i is $[\Sigma_{k \neq i} d(i,k)] / (n-2)$ and C is the same for all such trees





Neighbor Joining Algorithm

- For each species, compute $d(i,j) – u_i – u_j$
- Choose the i and the j for which this value is smallest
- Join clusters i and j to form a new cluster (call it n)
- Compute distances to the new cluster n as $d(n,k) = [d(i,k) + d(j,k) – d(i,j)] / 2$
- Delete i and j from the distance table, add the new cluster n, and iterate

An Example:

	A	B	C	D	E	F
A	--					
B	5	--				
C	4	7	--			
D	7	10	7	--		
E	6	9	6	5	--	
F	8	11	8	9	8	--

Least Squares Methods (Distance Based)

- Idea: Find the edge-labeled tree that minimizes the squared error between the *distance in the tree* and the *distance as presented in the input matrix*
  - Each edge label is the "evolutionary distance" along that edge
  - d(i,j) (from the input table) is not necessarily the same as D(i,j) (distance computed by walking along tree edges)
  - Error = $\Sigma_{i<j}(d(i,j) – D(i,j))^2$
- This method has better statistical justification than UPGMA or Neighbor Joining
- Just as for Parsimony
  - Given a tree, there is a reasonable algorithm to find the best labeling for the edges
  - But finding the best tree is NP-hard

Maximum Likelihood (Character Based)

- Idea: Given a tree, we evaluate the probability that *this* tree is produced under the assumption that evolution operates according to model M
  - M represents a model of evolution (e.g., we might use the BLOSUM or PAM matrices to indicate the likelihood of various substitutions)
  - The tree with the highest probability is assumed to be the correct one
- Advantages:
  - Statistically well-justified
  - Relatively robust to sampling error
- Disadvantages:
  - Computationally expensive
  - Result depend on model of evolution

Pros & Cons

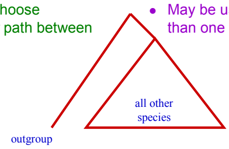
- Character based methods
  - Computationally expensive
  - Can create hypotheses about ancestral characters
- Distance based methods
  - Character data can be converted to distance data, but information is lost
  - Generally faster
- For the most part, methods based on any kind of optimality criteria lead to NP-hard problems

- Character based
  - Parsimony
    - Philosophically appealing (Occam's razor)
    - Sensitive to small input changes
  - Maximum Likelihood
    - Statistically well founded
    - Extremely slow
- Distance based
  - UPGMA
    - Reliable only for closely related species
  - Neighbor Joining
    - Fast; suitable for large datasets
  - Least Squares Methods
    - Statistically justified
    - NP-hard

Rooting an Unrooted Tree

- Most of the methods produce unrooted rather than rooted trees
- One method for finding the root: include an *outgroup*
  - An *outgroup* is species known to have branched off before all the other species (e.g., use a bird as an outgroup for a mammalian tree)
- Another method: Choose midpoint of *longest* path between leaves

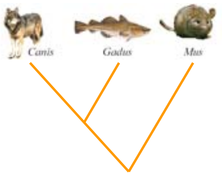
- Choosing an outgroup
  - Don't choose an outgroup that is too distant from your other species (multiple mutations can "erase" information)
  - Don't choose an outgroup that is too close (it may not really be an outgroup)
  - May be useful to use more than one outgroup



What Can Go Wrong?

- Model of evolution may not match real evolution
  - Example: the most parsimonious tree may not be the true tree
  - Example: the most likely tree may not be the true tree
  - Example: distance table may not match true "evolution distances"

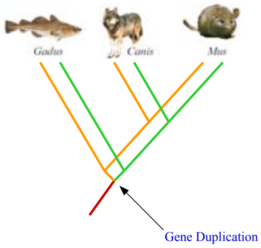
- Even when a phylogenetic tree algorithm is run correctly it is possible to produce a tree like this (example from http://www.daimi.au.dk/~schauser/bioinformatik_E03/lectures_E03/phylogenetic.pdf):



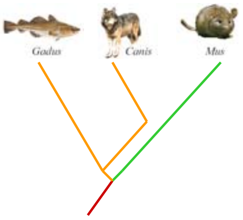
- How can this happen?

Gene Duplication

- Actual history



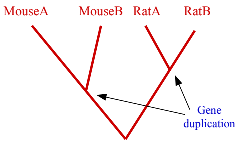
- The "species" we studied



- Our tree was correct, but we were mixing paralogs and orthologs

Orthologs vs. Paralogs

- Two genes are said to be orthologous if they diverged after a speciation event
  - ortho = exact
- Two genes are said to be paralogous if they diverged after a duplication event
  - para = parallel
- One can build trees of paralogs or of orthologs, but don't mix them



- MouseA and MouseB are paralogs
- RatA and RatB are paralogs
- MouseA and RatA are orthologs
- Also orthologs: (MouseA, RatB), (MouseB, RatA), (MouseB, RatB)