Phylogenetics

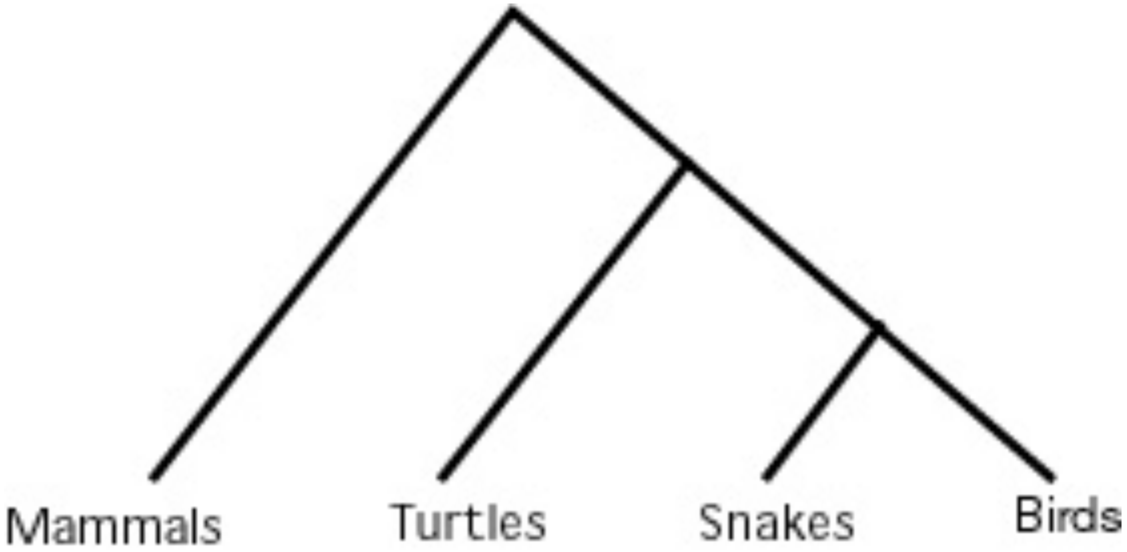
COS551, Fall 2003

Mona Singh

Phylogenetics

- Phylogenetic trees illustrate the evolutionary relationships among groups of organisms, or among a family of related nucleic acid or protein sequences
- E.g., how might have this family been derived during evolution

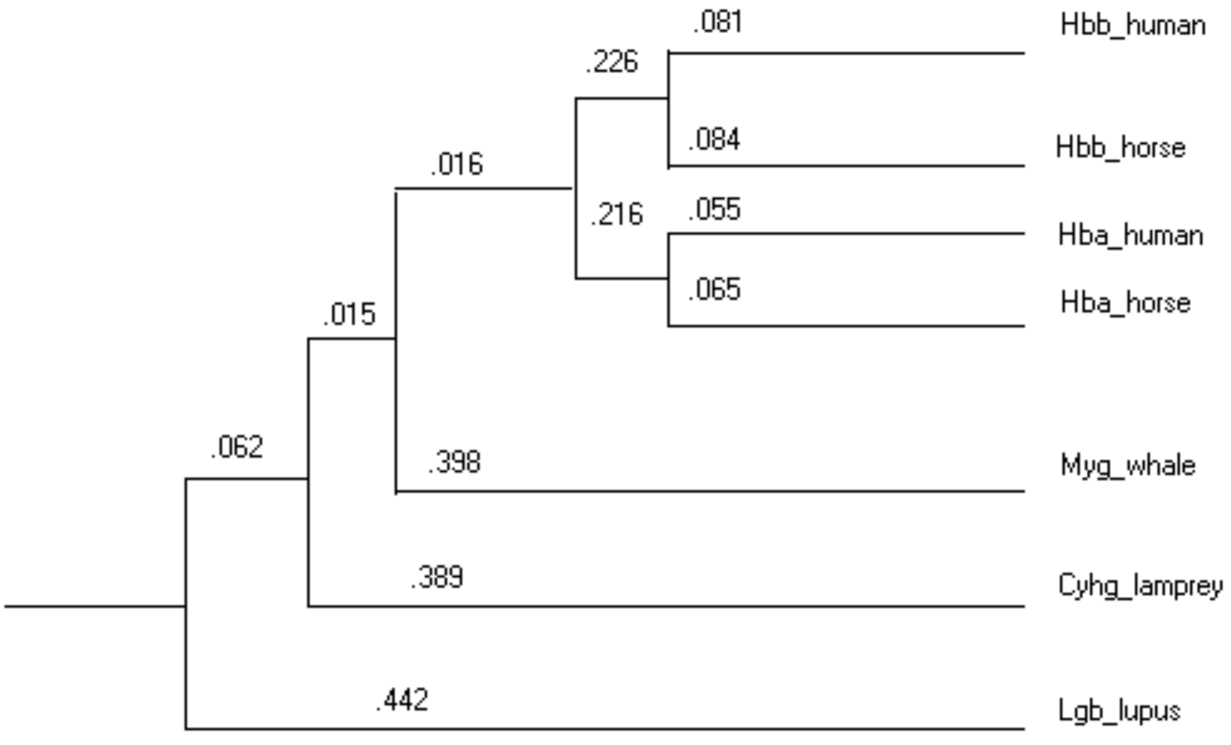
Hypothetical Tree Relating Organisms



Phylogenetic Relationships Among Organisms

- Entrez: www.ncbi.nlm.nih.gov/Taxonomy
- Ribosomal database project: rdp.cme.msu.edu/html/
- Tree of Life: phylogeny.arizona.edu/tree/phylogeny.html

Globin Sequences



Note: Figure not drawn to scale

Phylogeny Applications

- Tree of life: Analyzing changes that have occurred in evolution of different organisms
- Phylogenetic relationships among genes can help predict which ones might have similar functions (e.g., ortholog detection)
- Follow changes occuring in rapidly changing species (e.g., HIV virus)

Phylogeny Packages

- PHYLIP, Phylogenetic inference package
  - evolution.genetics.washington.edu/phylip.html
  - Felsenstein
  - Free!
- PAUP, phylogenetic analysis using parsimony
  - paup.csit.fsu.edu
  - Swofford

What data is used to build trees?

- Traditionally: morphological features (e.g., number of legs, beak shape, etc.)
- Today: Mostly molecular data (e.g., DNA and protein sequences)

Data for Phylogeny

- Can be classified into two categories:
  - Numerical data
    - Distance between objects

      e.g., distance(man, mouse)=500,

      distance(man, chimp)=100

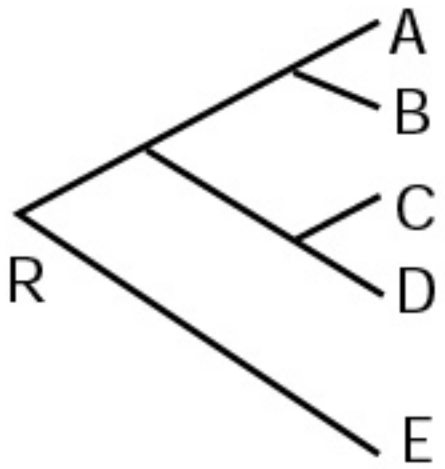
      Usually derived from sequence data
  - Discrete characters
    - Each character has finite number of states

      e.g., number of legs = 1, 2, 4

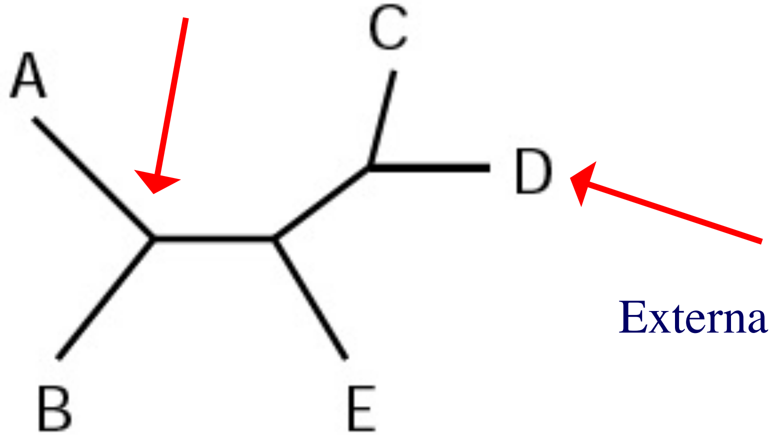
      DNA = {A, C, T, G}

Rooted vs Unrooted Trees

Root



Internal node



External node

Rooted tree

Unrooted tree

Note: Here, each node has three neighboring nodes

Terminology

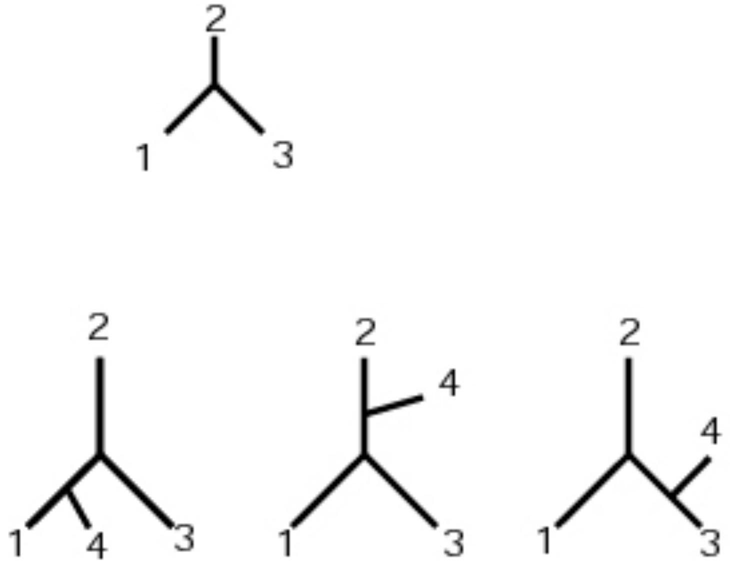
- External nodes: things under comparison; operational taxonomic units (OTUs)
- Internal nodes: ancestral units; hypothetical; goal is to group current day units
- Root: common ancestor of all OTUs under study. Path from root to node defines evolutionary path
- Unrooted: specify relationship but not evolutionary path
  - If have an outgroup (external reason to believe certain OTU branched off first), then can root
- Topology: branching pattern of a tree
- Branch length: amount of difference that occurred along a branch

How to reconstruct trees

- Distance methods: evolutionary distances are computed for all OTUs and build tree where distance between OTUs “matches” these distances
- Maximum parsimony (MP): choose tree that minimizes number of changes required to explain data
- Maximum likelihood (ML): under a model of sequence evolution, find the tree which gives the highest likelihood of the observed data

Number of possible trees

Given *n* OTUs, there are $\prod_{i=3}^{n}(2i-5)$ unrooted trees



OTUs	unrooted trees
3	1
4	3
5	15
10	2,027,025

Number of possible trees

Given *n* OTUs, there are $\prod_{i=3}^{n}(2i-3)$ rooted trees

Bottom Line: an enumeration strategy over all possible trees to find the best one under some criteria is not feasible!

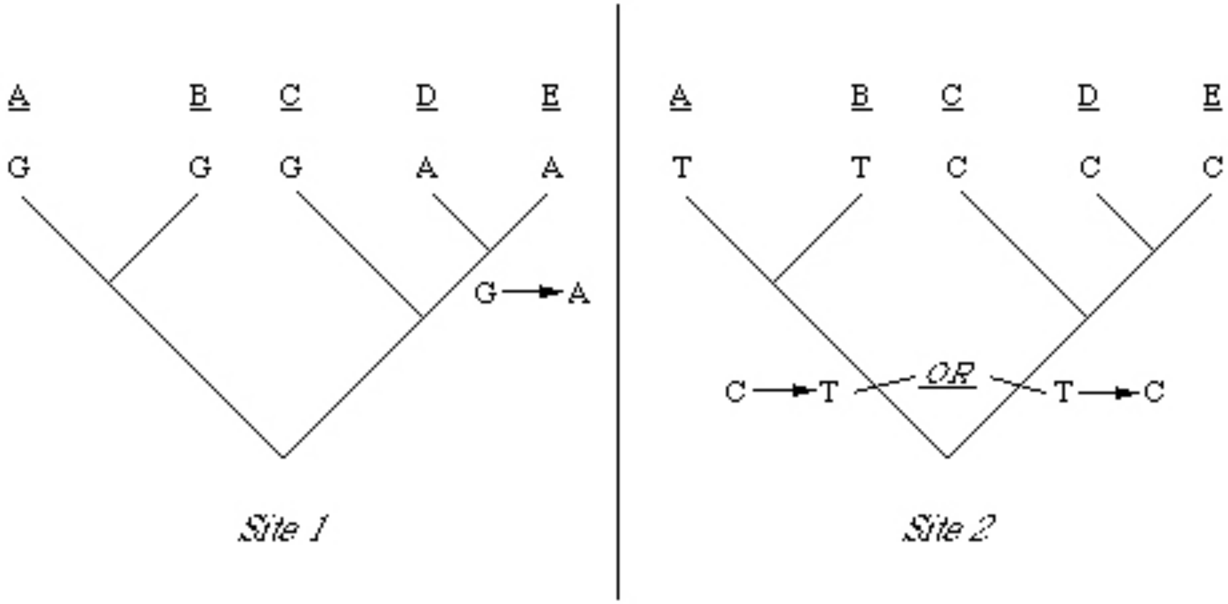
OTUs	Rooted trees
3	3
4	15
5	105
10	34,459,425

Parsimony

Find tree which minimizes number of changes needed to explain data

Ex:

	123456
A	GTCGTA
B	GTCACT
C	GCGGTA
D	ACGACA
E	ACGGAA



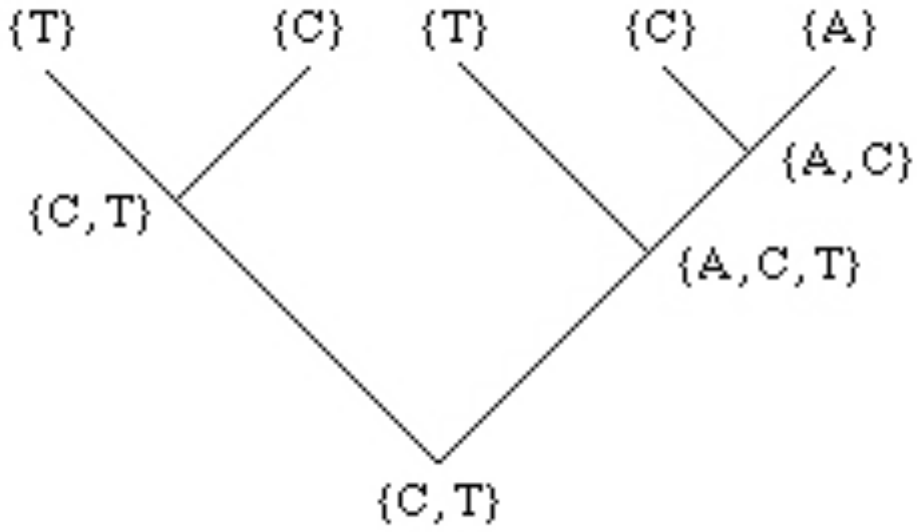
Site 1

Site 2

Parsimony

- For given example tree and alignment, can do this for all sites, and get away with as few as 9 changes
- Changing the tree (either the topology or labeling of leaves) changes the minimum number of changes need
- Two computational problems
  - (Easy) Given a particular tree, how do you find minimum number of changes need to explain data? (Fitch)
  - (Hard) How do you search through all trees?

Parsimony: Fitch's algorithm



Idea: construct set of possible nucleotides for internal nodes, based on possible assignments of children

Parsimony: Fitch's algorithm

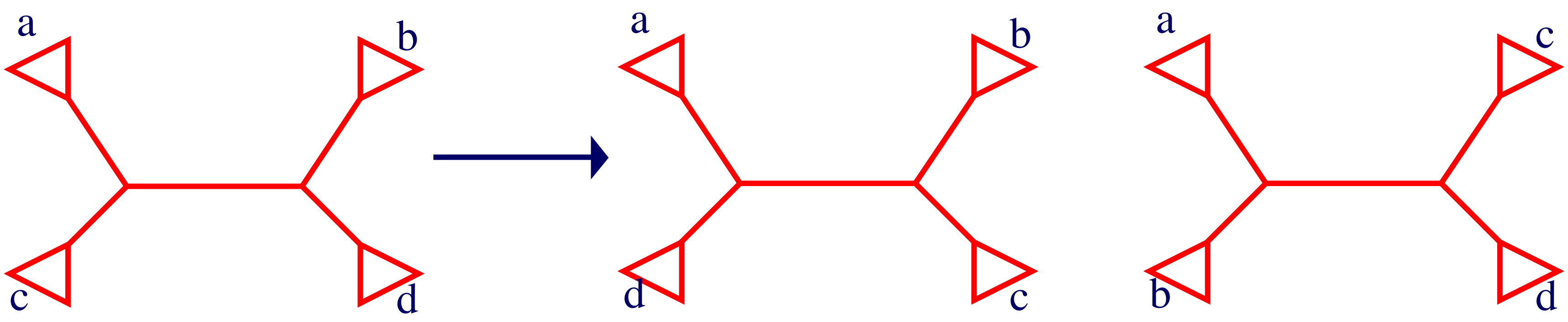
- For each site:
  - Each leaf is labeled with set containing observed nucleotide at that position
  - For each internal node *i* with children *j* and *k* with labels S_j and S_k

$$S_i = \begin{cases} S_j \cup S_k \text{ if } S_j \cap S_k \text{ is empty} \\ S_j \cap S_k \text{ otherwise} \end{cases}$$

- Total # changes necessary for a site is # of union operations

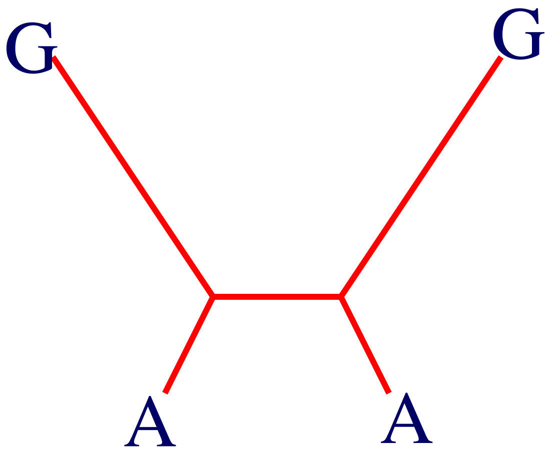
Parsimony

- How do you search through all trees?
  - Enumerate all trees (too many…)
  - Can use techniques to try to limit the search space (e.g., branch and bound)
  - or use heuristics (many possibilities)
    - E.g., nearest neighbor interchange. Start with a tree and consider neighboring trees. If any neighboring tree has fewer changes, take it as current tree. Stop when no improvements

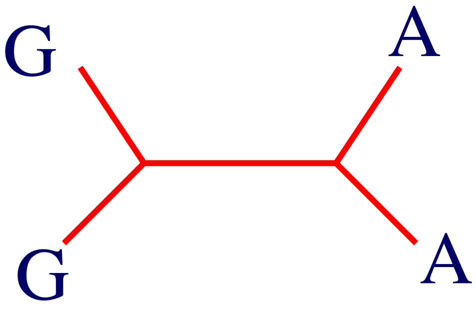


Parsimony weaknesses

Parsimony analysis implicitly assumes that rate of change along branches are similar



Real tree: two long branches
where G has turned to A independently



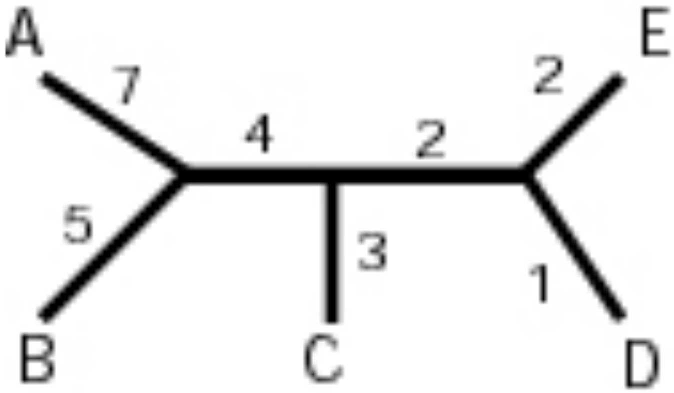
Inferred tree

Distance Methods

- Input: given an n x n matrix M where $M_{ij}>=0$ and M_{ij} is the distance between objects i and j
- Goal: Build an edge-weighted tree where each leaf (external node) corresponds to one object of M and so that distances measured on the tree between leaves i and j correspond to M_{ij}

Distance Methods

	A	B	C	D	E
A	0				
B	12	0			
C	14	12	0		
D	14	12	6	0	
E	15	13	7	3	0



A tree exactly fitting the matrix does not always exist.

Distance Method Criteria

- Try to find the tree with distances d_{ij} which "best fits" the distance data M_{ij}
- Different possibilities for "best"
  - Cavalli-Sforza criterion: minimize

    $$\sum_{i,j}(M_{ij} - d_{ij})^2$$

  - Fitch-Margoliash criterion: minimize

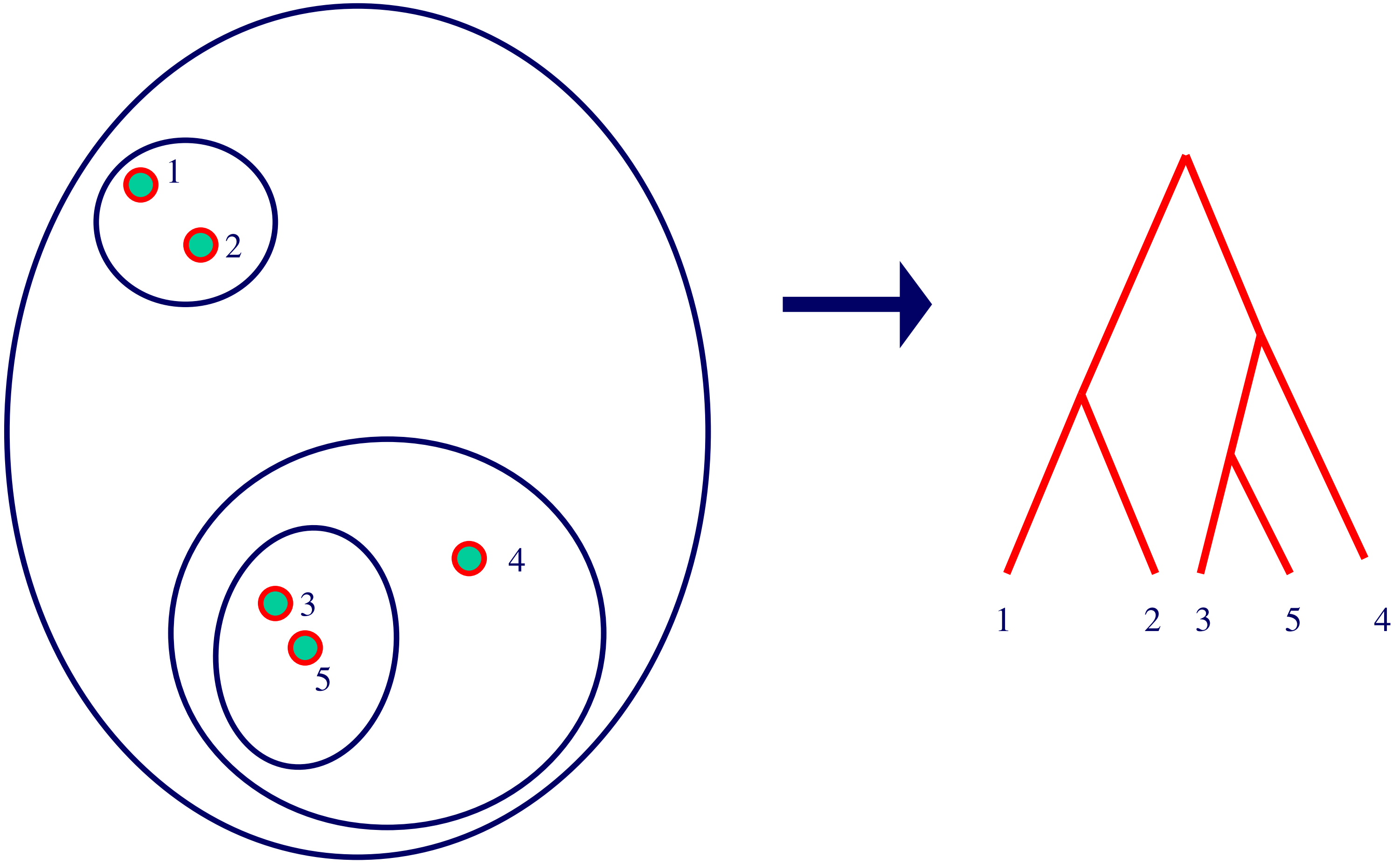
    $$\sum_{i,j} \frac{(M_{ij}-d_{ij})^2}{M_{ij}^2}$$

- Unfortunately, both lead to computationally intractable problems (e.g., enumerating)

Distance Method Heuristic: UPGMA

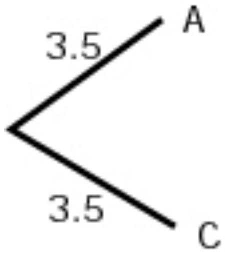
- UPGMA (Unweighted group method with arithmetic mean)
  - Sequential clustering algorithm
  - Start with things most similar
    - Build a composite OTU
  - Distances to this OTU are computed as arithmetic means
  - From new group of OTUs, pick pair with highest similarity etc.
- Average-linkage clustering

UPGMA: Visually



UPGMA Example

	A	B	C	D
A	0			
B	8	0		
C	7	9	0	
D	12	14	11	0

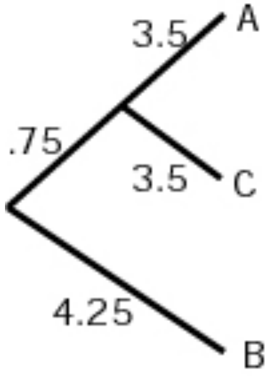


$$M_{B(AC)} = (M_{BA} + M_{BC})/2 = (8+9)/2=8.5$$

$$M_{D(AC)} = (M_{DA} + M_{DC})/2= (12+11)/2=11.5$$

UPGMA Example

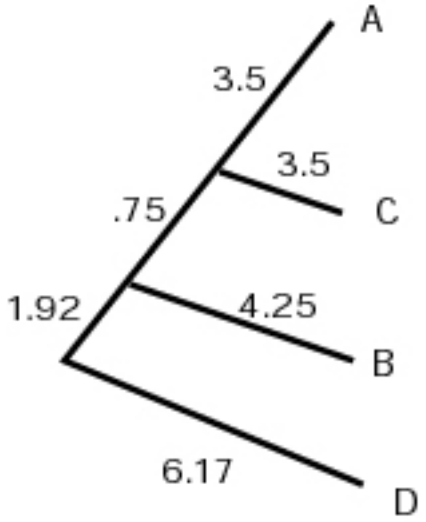
	AC	B	D
AC	0		
B	8.5	0	
D	11.5	14	0



$$M_{(ABC)D} = (M_{AD} + M_{BD} + M_{CD})/3 = (12+14+11)/3$$

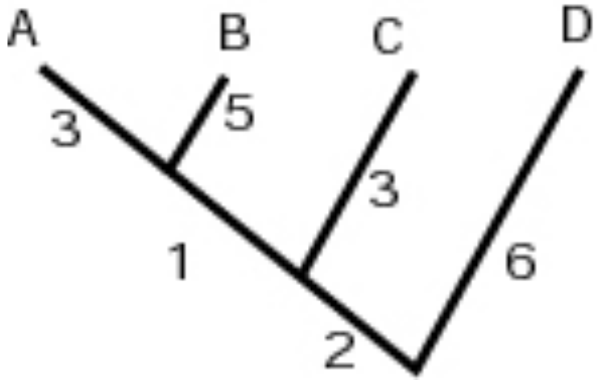
UPGMA: Example

	ABC	D
ABC	0	
D	12.33	0



UPGMA weaknesses

	A	B	C	D
A	0			
B	8	0		
C	7	9	0	
D	12	14	11	0



In fact, exact fitting tree exists !

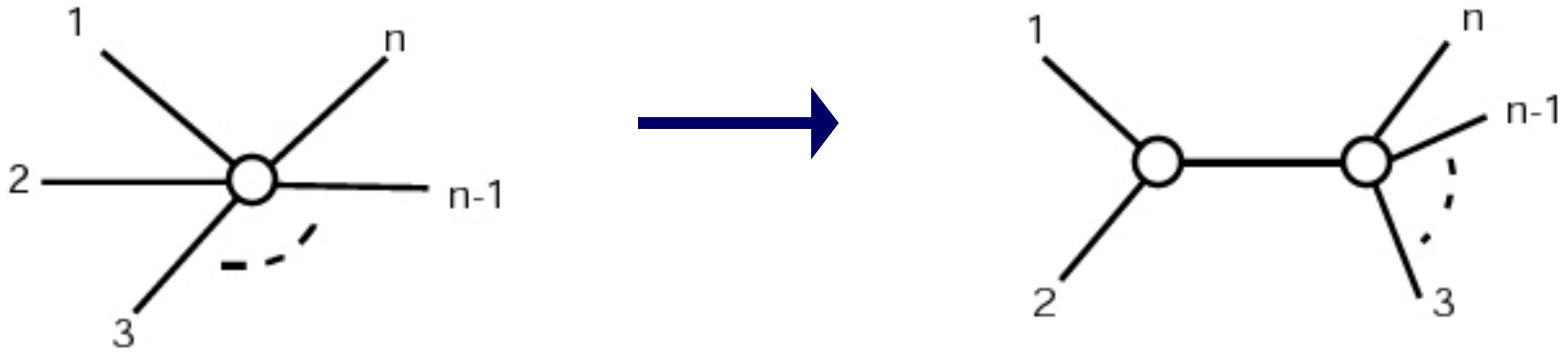
UPGMA weaknesses

- UPGMA assumes that the rates of evolution are the same among different lineages
- In general, should not use this method for phylogenetic tree reconstruction (unless believe assumption)
- Produces a rooted tree
- As a general clustering method (as we discussed in an earlier lecture), it is better…

Distance Method: Neighbor Joining

- Most widely-used distance based method for phylogenetic reconstruction
- UPGMA illustrated that it is not enough to just pick closest neighbors
- Idea here: take into account averaged distances to other leaves as well
- Produces an unrooted tree

Neighbor Joining (NJ)



Start off with star tree; pull out pairs at a time

NJ Algorithm

Step 1: Let $u_i = \sum_k \frac{M_{ik}}{n-2}$

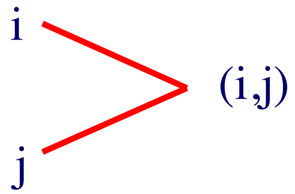
- (Almost) “average” distance to other nodes

Step 2: Choose *i* and *j* for which $M_{ij} - u_i - u_j$ is smallest

- Look for nodes that are close to each other, and far from everything else
- Turns out minimizing this is minimizing sum of branch lengths

NJ algorithm

Step 3: Define a new cluster (*i*, *j*), with a corresponding node in the tree



Distance from *i* and *j* to node (*i*,*j*):

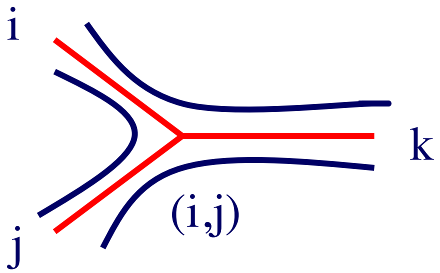
$$d_{i,\,(i,j)} = 0.5(M_{ij} + u_i - u_j)$$
$$d_{j,\,(i,j)} = 0.5(M_{ij} + u_j - u_i)$$

Default: split distance but if on average one is further away, make it longer

NJ Algorithm

Step 4: Compute distance between new cluster and all other clusters:

$$M_{(ij)k} = \frac{M_{ik}+M_{jk} - M_{ij}}{2}$$



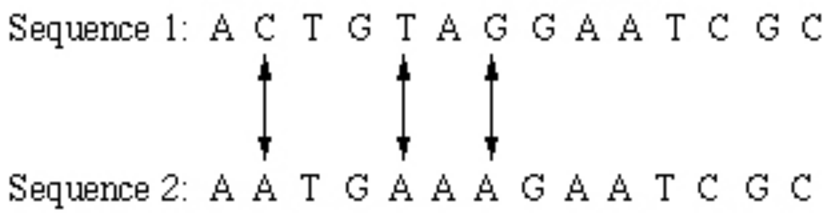
Step 5: Delete *i* and *j* from matrix and replace by *(i, j)*

Step 6: Continue until only 2 leaves remain

NJ Performance

- Works well in practice
- If there is a tree that fits the matrix, it will find it
- Can sometimes get trees with negative length edges (!)

Computing Distances Between Sequences



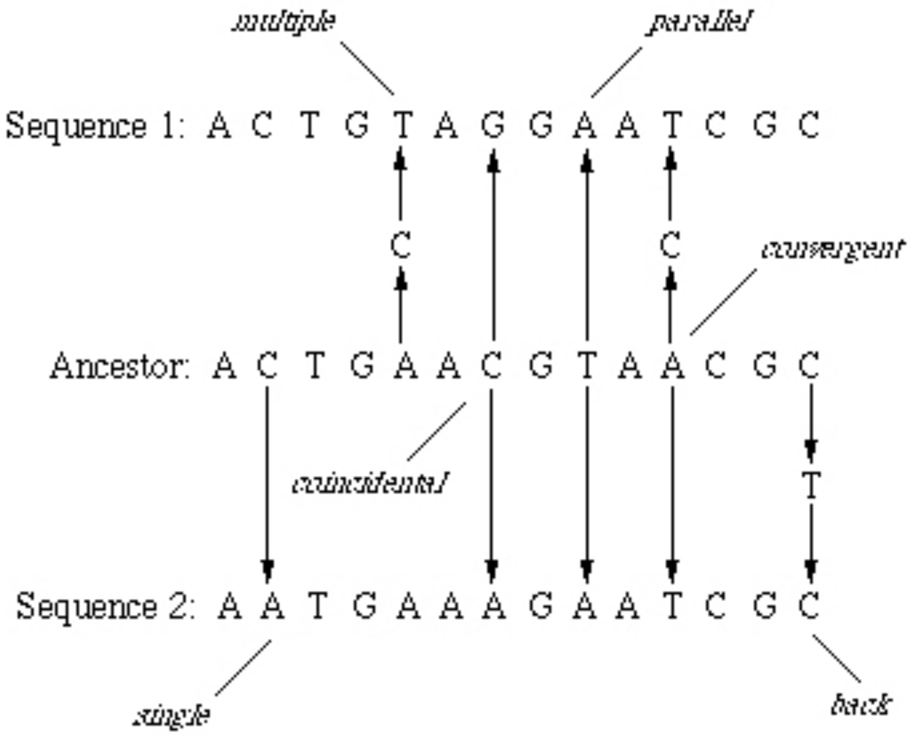
Sequence 1: A C T G T A G G A A T C G C

Sequence 2: A A T G A A A G A A T C G C

Could compute fraction of mismatches between two sequences; however, this is an underestimate of actual distance

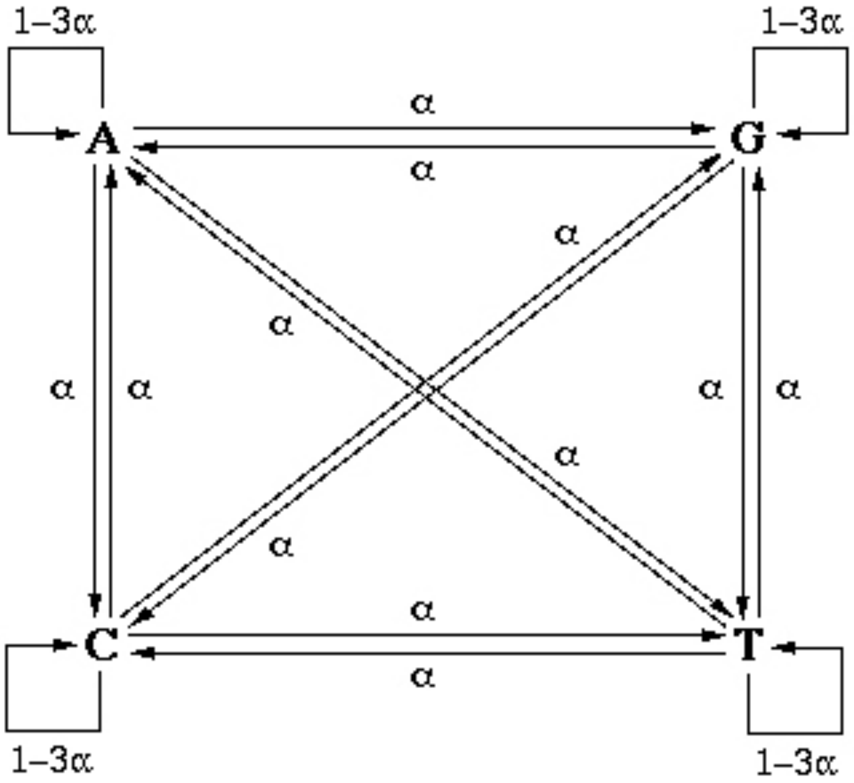
Computing Distances Between Sequences

E.g., many underlying substitutions possible


multiple
parallel
Sequence 1: A C T G T A G G A A T C G C
C
C
convergent
Ancestor: A C T G A A C G T A A C G C
coincidental
T
Sequence 2: A A T G A A A G A A T C G C
single
back

Use models of substitution to correct these values

Computing Distances Between Sequences



Jukes & Cantor model

-Each position in DNA sequence is independent

-Each position can mutates with same probability to any another base

Correction to observed substitution rate (see notes):

$$-0.75(\ln(1 - \tfrac{4}{3}(\frac{\text{observed \# differences}}{\text{length}}))$$

Ex: Computing Distances Between Sequences

- Alignment of two DNA sequences
  - Length of alignment (non gapped positions): 100
  - Number of differences: 25
- Naïve distance calculation = 25/100 = ¼
- Correction

$$-0.75(\ln(1-\tfrac{4}{3}(1/4)) = -.75\ln(\tfrac{2}{3}) = .304$$

- Other models for DNA, also protein (e.g., PAM)

Maximum Likelihood

- Given a probabilistic model for nucleotide (or protein) substitution (e.g., Jukes & Cantor), pick the tree that has highest probability of generating observed data
  - I.e., Given data *D* and model *M*, find tree *T* such that $Pr(D|T, M)$ is maximized
- Models gives values $p_{ij}(t)$, the probability of going from nucleotide *i* to *j* in time *t*

Maximum Likelihood

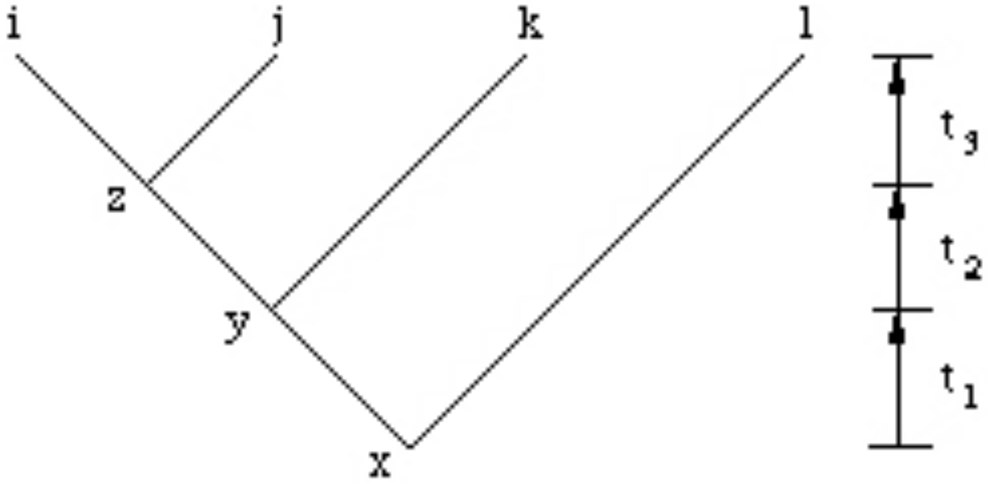
- Makes 2 independence assumptions
  - Different sites evolve independently
  - Diverged sequences (or species) evolve independently after diverging
- If D_i is data for *i*th site

$$Pr(D|T,M) = \prod_i Pr(D_i|T,M)$$

Maximum Likelihood

How to calculate *$Pr(D_i|T,M)$* ?

$p_{xy}(t)$ ~ prob of going from *x* to *y* in time *t*



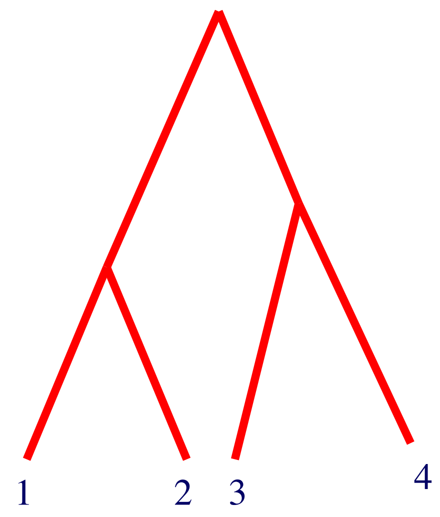
$$Pr(i,j,k,l|T,M) =$$
$$\sum_x \sum_y \sum_z pr(x)(p_{xl} \cdot (t_1+t_2+t_3) \cdot p_{xy}(t_1)$$
$$\cdot\, p_{yk}(t_2+t_3) \cdot p_{yz}(t_2) \cdot p_{zi}(t_3) \cdot p_{zj}(t_3))$$

Maximum Likelihood

- Given tree topology and branch lengths, can efficiently calculate *Pr(D|T, M)* using dynamic programming
  - I.e., don't have to enumerate over all internal states
- Finding best maximum likelihood tree is expensive
  - Must consider all topologies
  - Find best edge lengths for each topology
    - Idea: use some search procedure, e.g., EM, to optimize these lengths

Assessing Reliability: Bootstrap

Say we've inferred the following tree



Would like to get confidence levels that 1 & 2 belong together, and 3&4 belong together

Assessing Reliability: Bootstrap

Say we’re given following alignment:

```
    12345678
1   GCAGTACT
2   GTAGTACT
3   ACAATACC
4   ACAACACT
```

We’ll create a pseudosample by choosing sites randomly until N sites are chosen (N is length of alignment)

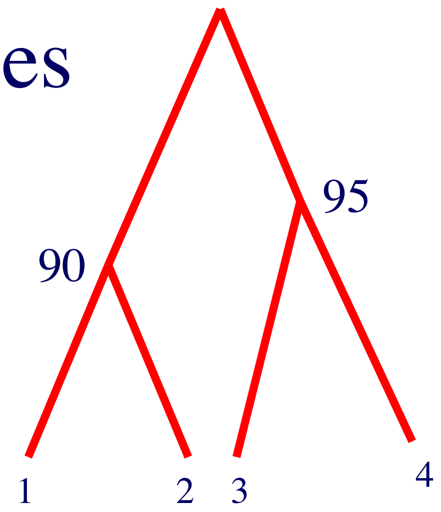
Assessing Reliability: Bootstrap

Say chose 6th, 1st, 6th, 8th, …

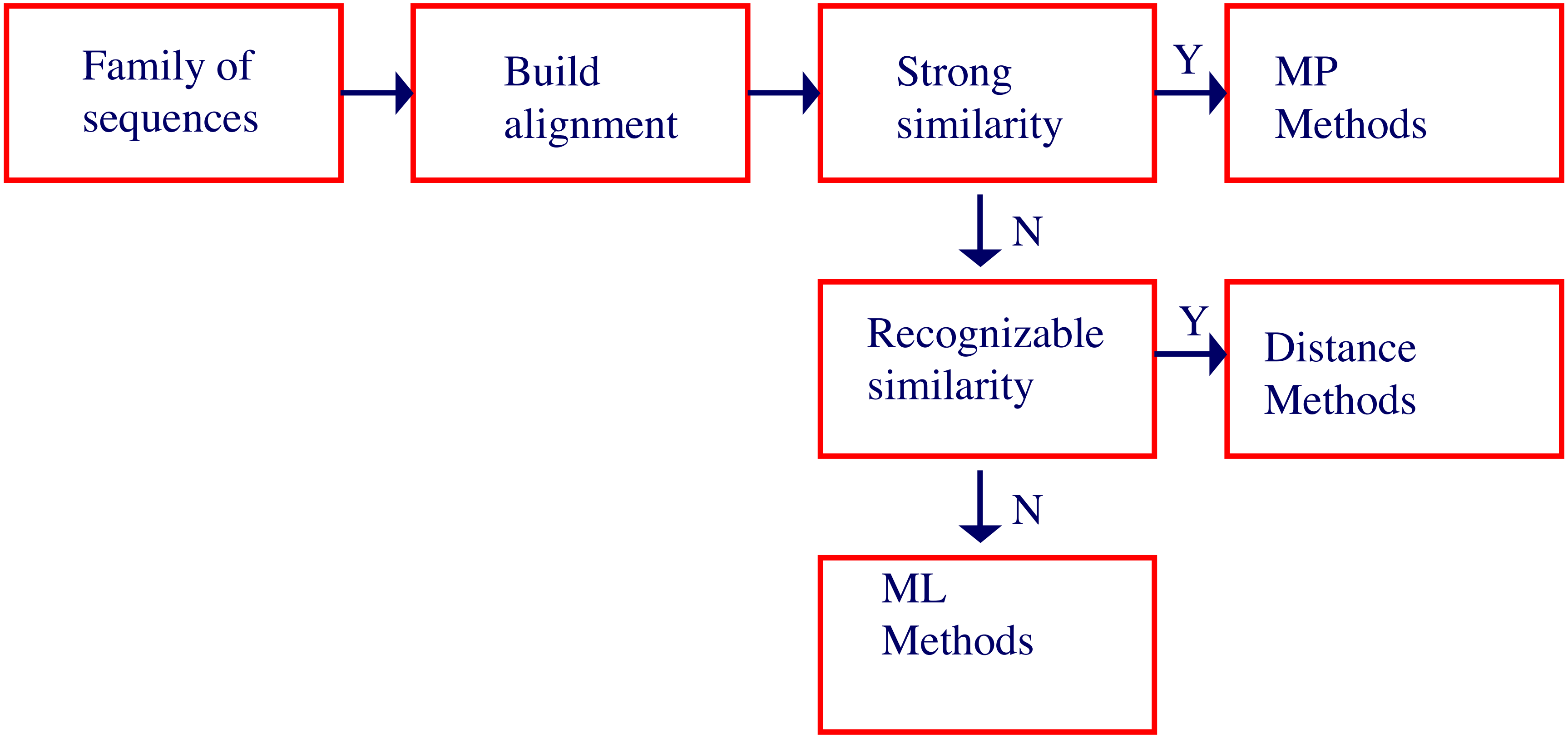
```
    12345678         6168  …
1   GCAGTACT         AGAT  …
2   GTAGTACT   →     AGAT  …
3   ACAATACC         AAAC  …
4   ACAACACT         AAAT  …
```

Assessing Reliability: Bootstrap

- Use pseudosample to construct tree
- Repeat many times
- Confidence of (1) and (2) together is fraction of times they appear together in trees generated from pseudosamples



Phylogeny Flowchart



(Mount, *Bioinformatics*)

Difference in Methods

- Maximum-likelihood and parsimony methods have models of evolution
- Distance methods do not necessarily
  - Useful aspect in some circumstances
    - E.g., trees built based on whole genomes, presence or absence of genes
- Religious wars over which methods to use
  - Most people now believe ML based methods are best: most sensitive at large evolutionary distances – but also most time-consuming & depend on specific model of evolution used
- Most commonly used packages contain software for all three methods: may want to use more than 1 to have confidence in built tree

Phylip

- Parsimony
  - DNApenny or Protpars
- Distance
  - Compute distance measure using DNAdist or Protdist
  - Neighbor (can use NJ or UPGMA)
- ML
  - DNAml