# Phylogeny Tree Algorithms

Jianlin Cheng, PhD

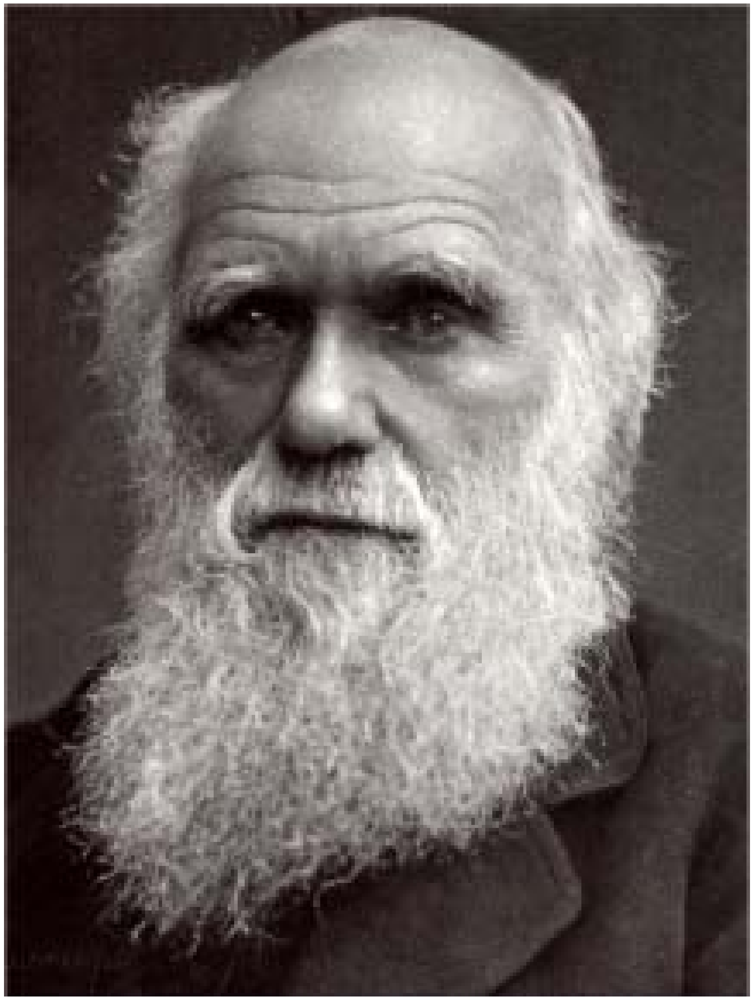
School of Electrical Engineering and Computer Science
University of Central Florida

2006

# Evolution

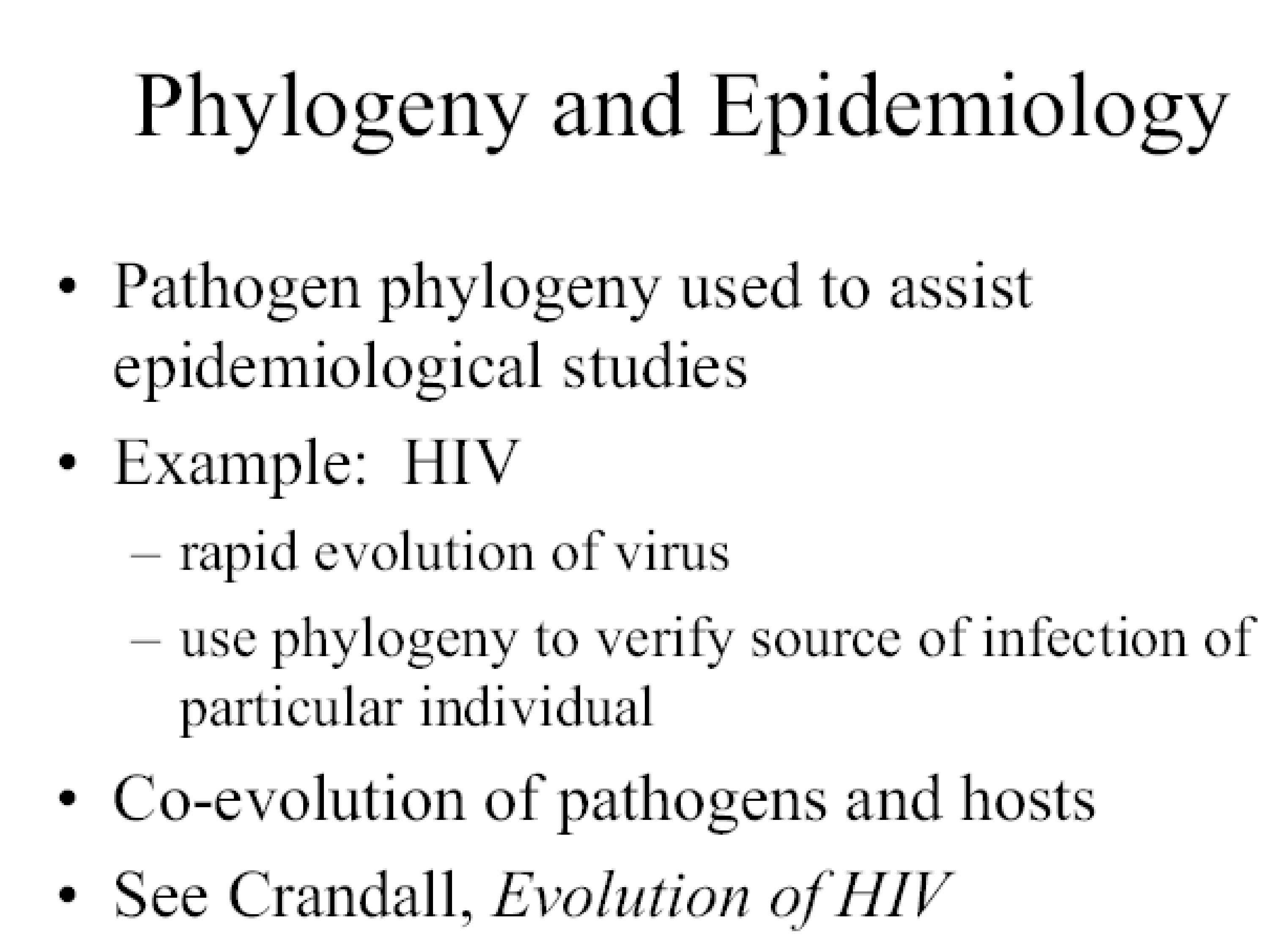
Evolution of new organisms is driven by

- **Diversity**
  - Different individuals carry different variants of the same basic blue print
- **Mutations**
  - The DNA sequence can be changed due to single base changes, deletion/insertion of DNA segments, etc.
- **Selection bias**



# Motivation

- To understand lineage of various species (evolutionary history)
- To understand how various functions evolve
- To inform multiple alignments
- To map virus strains (vaccine construction)
- To identify what is most conserved / important in some class of sequences



# Phylogeny and Epidemiology

- Pathogen phylogeny used to assist epidemiological studies
- Example:  HIV
  - rapid evolution of virus
  - use phylogeny to verify source of infection of particular individual
- Co-evolution of pathogens and hosts
- See Crandall, *Evolution of HIV*

Frank Olken, 2002

# Historical Note

- Until mid 1950’s phylogenies were constructed by experts based on their opinion (subjective criteria)

- Since then, focus on **objective** criteria for constructing phylogenetic trees
  - Thousands of articles in the last decades

- Important for many aspects of biology
  - Classification
  - Understanding biological mechanisms

# Morphological vs. Molecular

- Classical phylogenetic analysis: **morphological** features: number of legs, lengths of legs, etc.

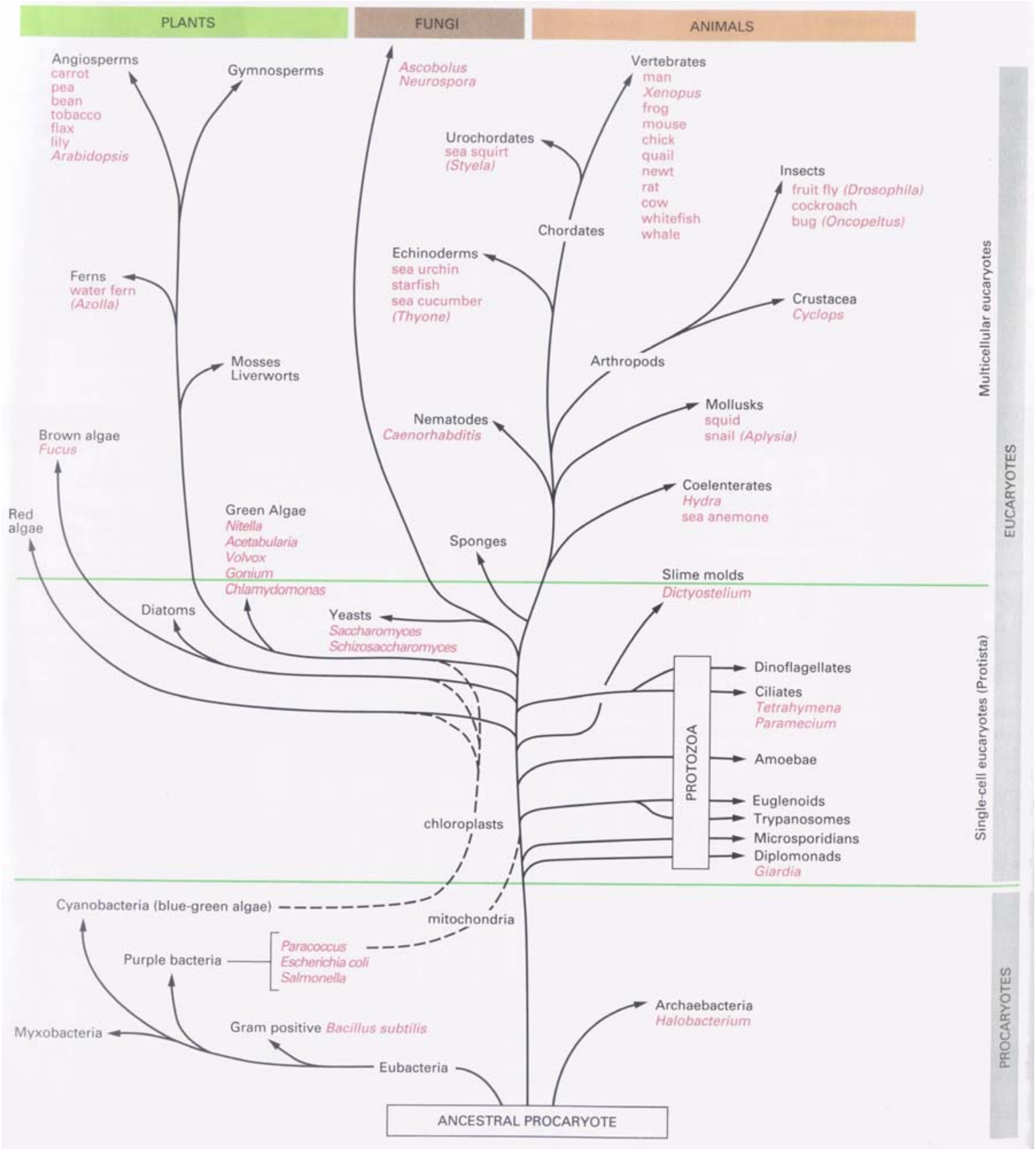
- Modern biological methods allow to use **molecular** features
  - Gene sequences
  - Protein sequences

- Analysis based on homologous sequences (e.g., globins) in different species

# Phylogeny Tree Basics

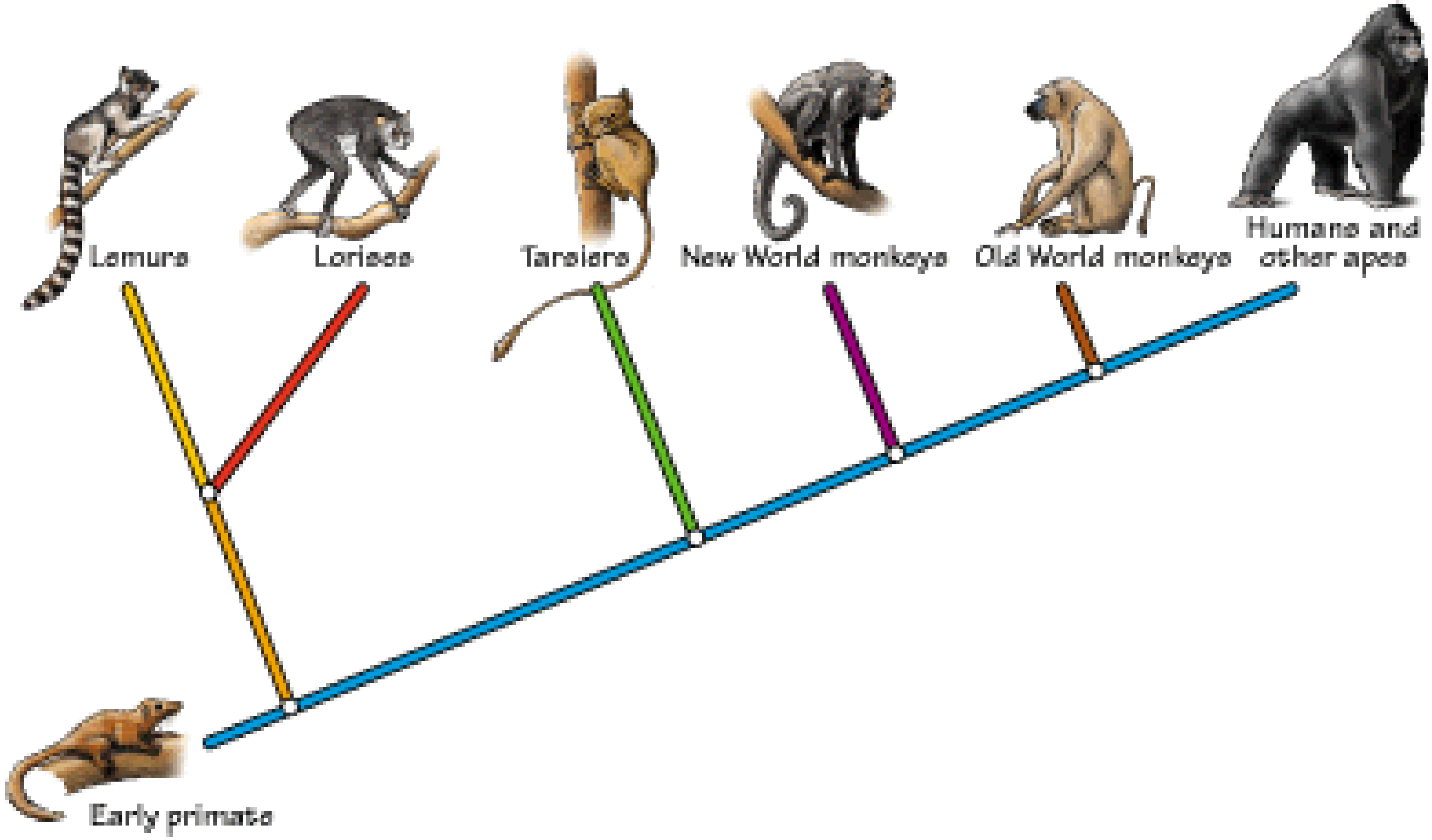
- Leaves represent things (genes, individuals, strains, species) being compared. Term taxon (taxa plural) is used to refer to this.
- Internal nodes are hypothetical ancestral units
- In a rooted tree, path from root represents an evolutionary path (root represents the common ancestor)
- An unrooted tree specifies relationships among things, but no evolutionary path.

# Tree of Life

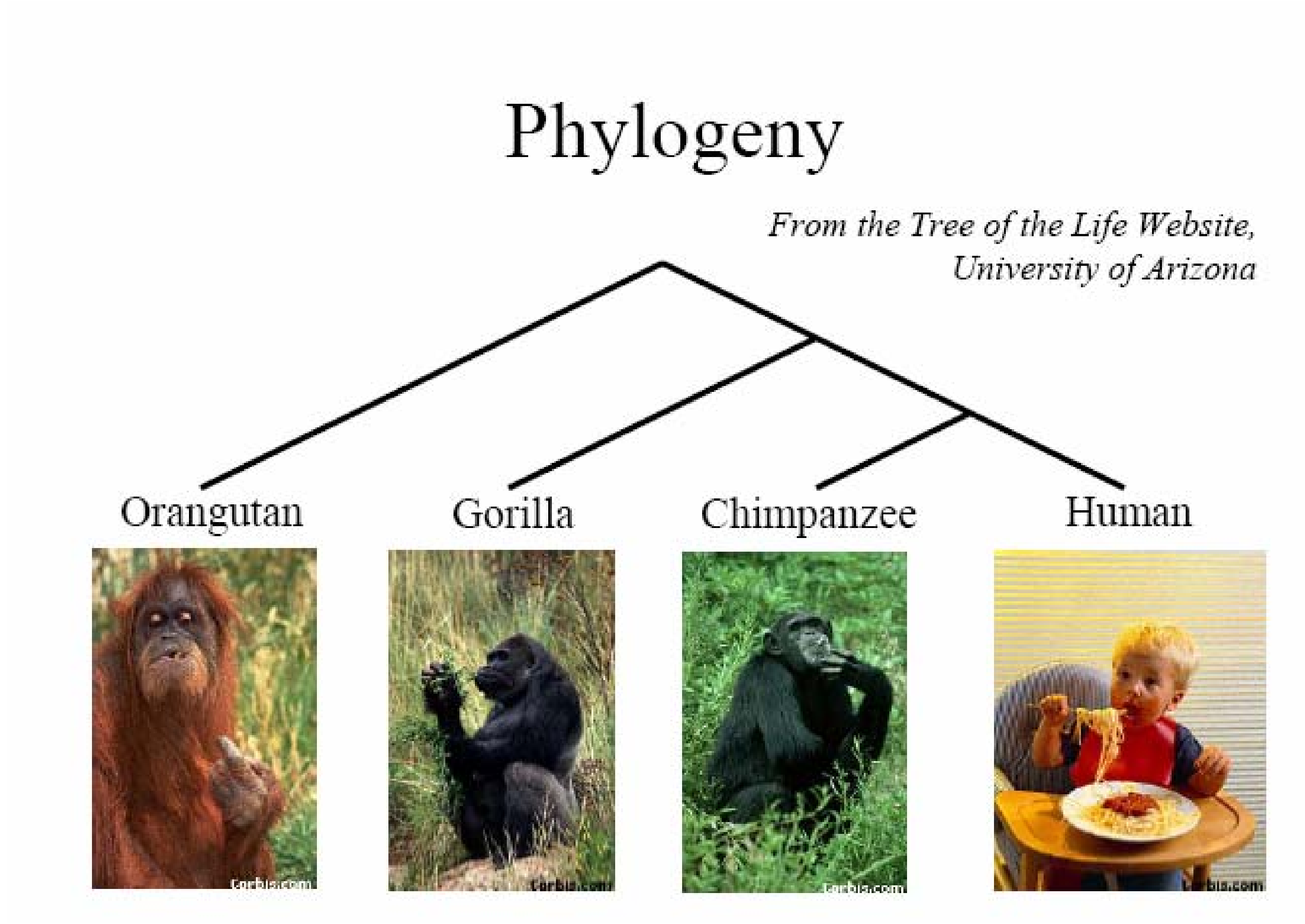


**Source: Alberts et al**

# Primate Evolution



S. Moran and I. Wexler



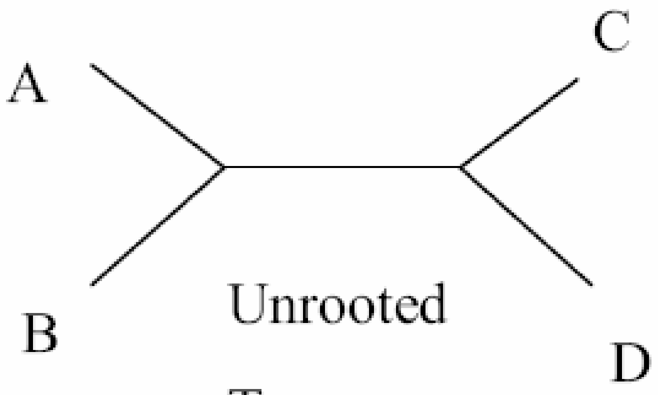
# Phylogeny

*From the Tree of the Life Website,*
*University of Arizona*

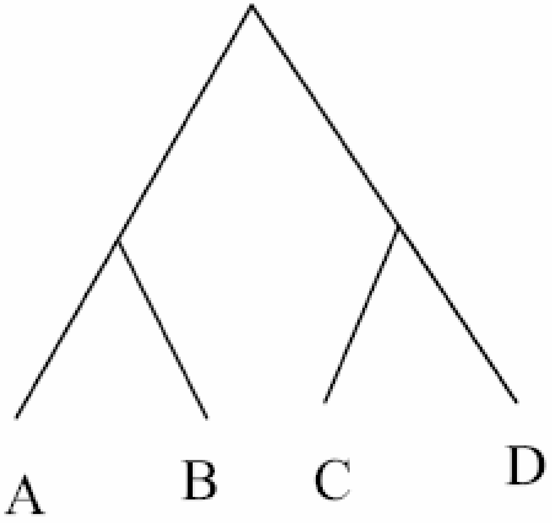
T. Warnow, 2004

# Example

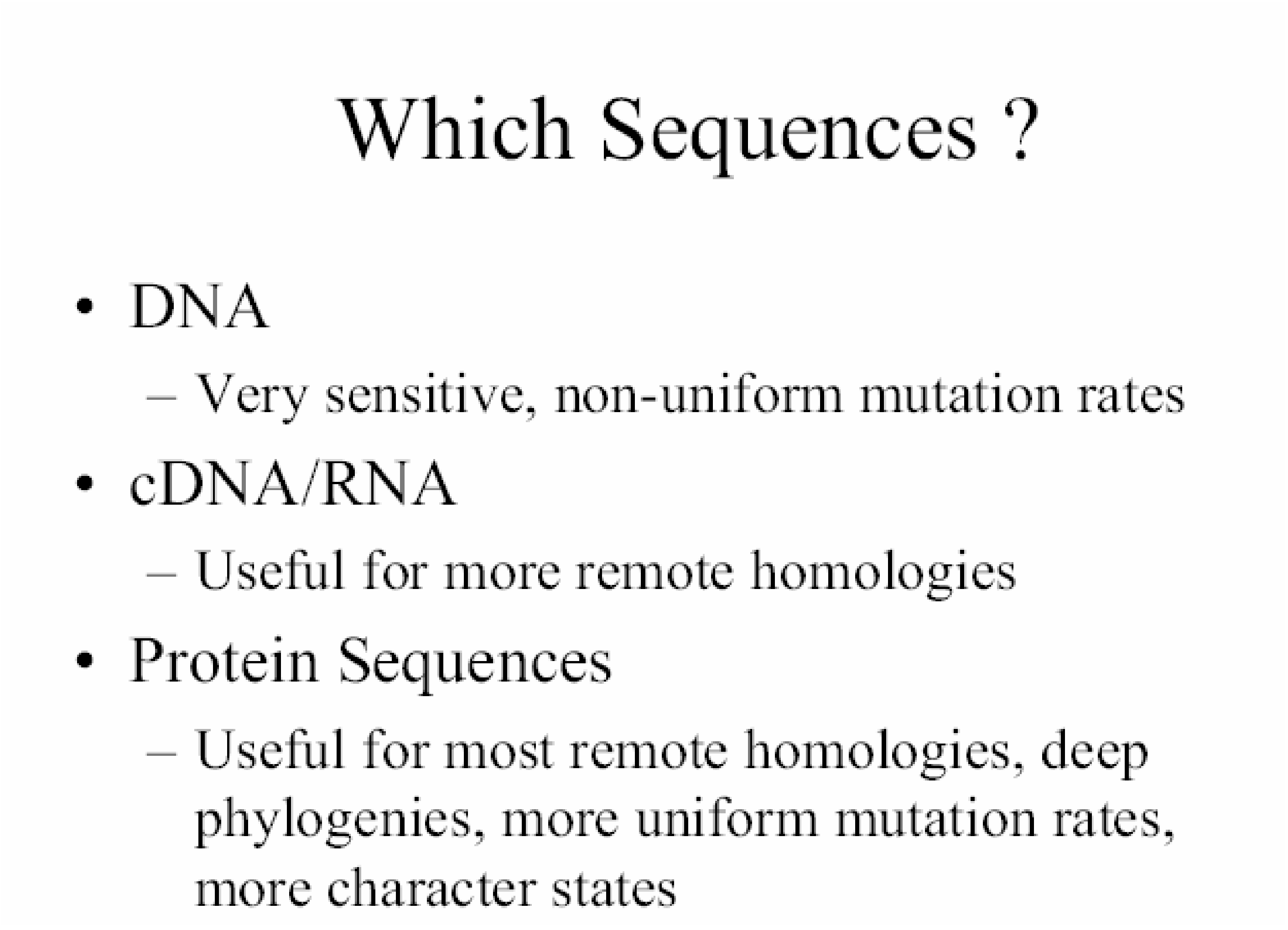
- Seq. A = A A C C G G T T
- Seq. B = A A C C G G T G
- Seq. C = A C C C G G T C
- Seq. D = A C C C G G T A



Unrooted Tree



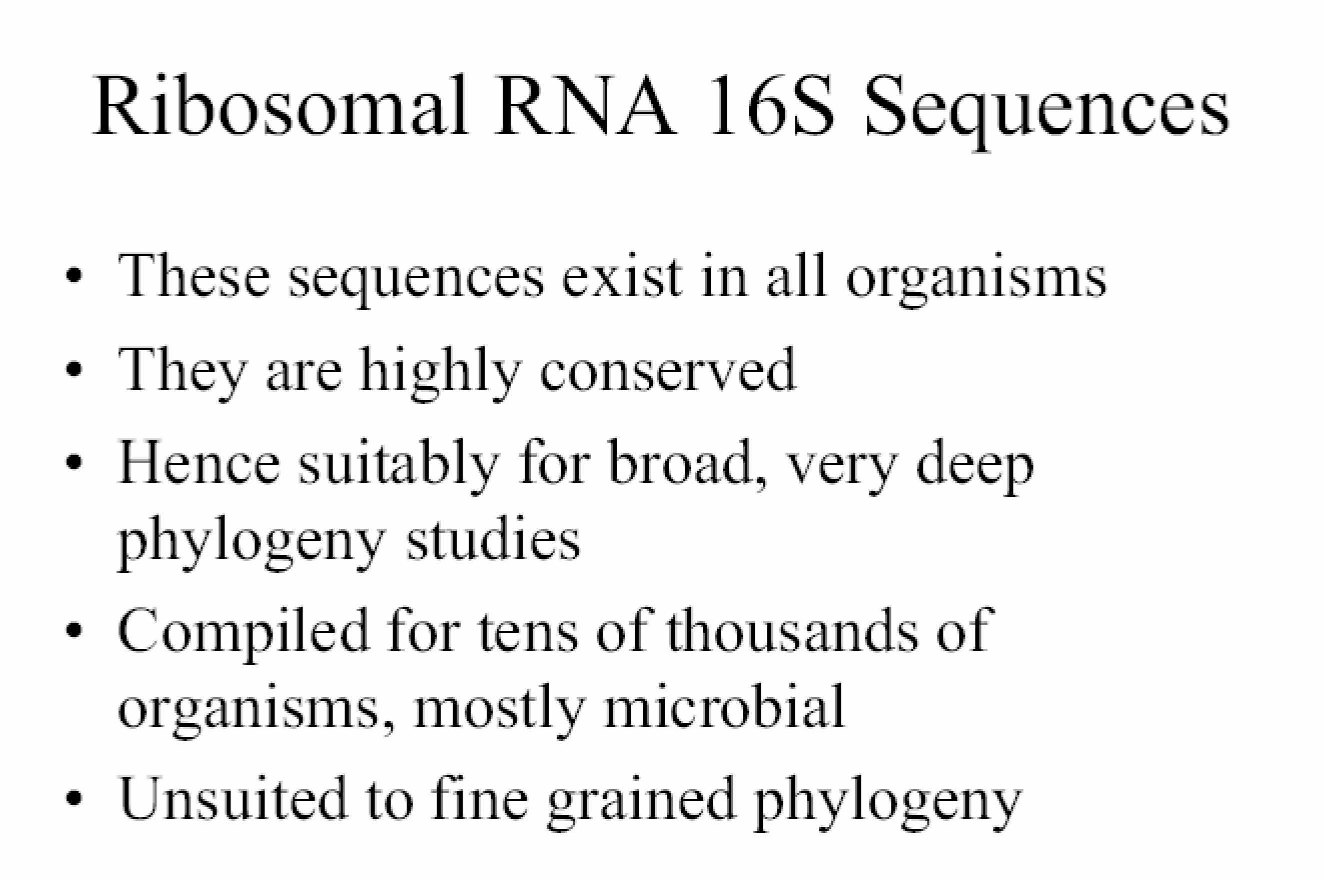
Rooted Tree



# Which Sequences ?

- DNA
  - Very sensitive, non-uniform mutation rates
- cDNA/RNA
  - Useful for more remote homologies
- Protein Sequences
  - Useful for most remote homologies, deep phylogenies, more uniform mutation rates, more character states

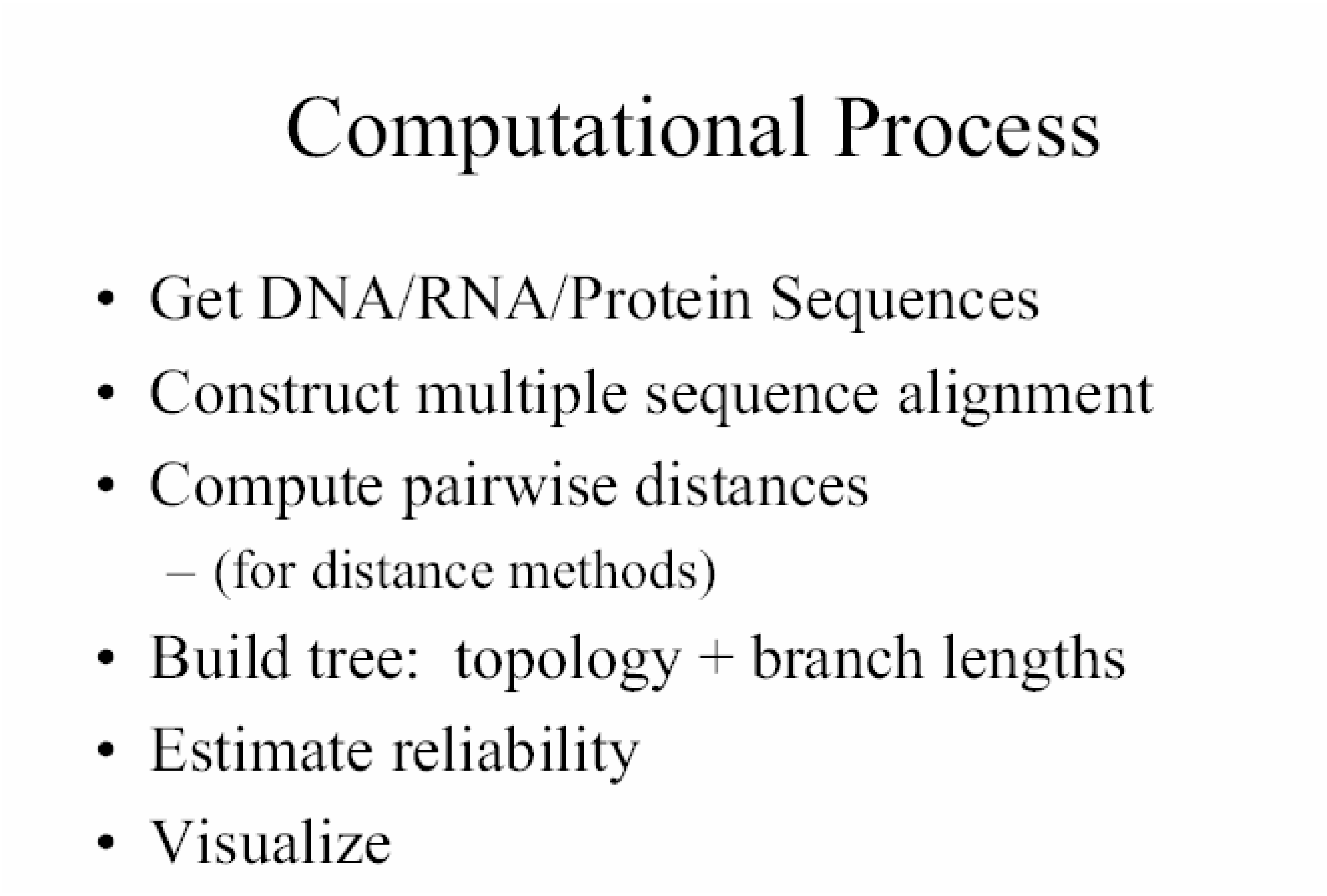
Frank Olken, 2002



# Ribosomal RNA 16S Sequences

- These sequences exist in all organisms
- They are highly conserved
- Hence suitably for broad, very deep phylogeny studies
- Compiled for tens of thousands of organisms, mostly microbial
- Unsuited to fine grained phylogeny

Frank Olken, 2002



# Computational Process

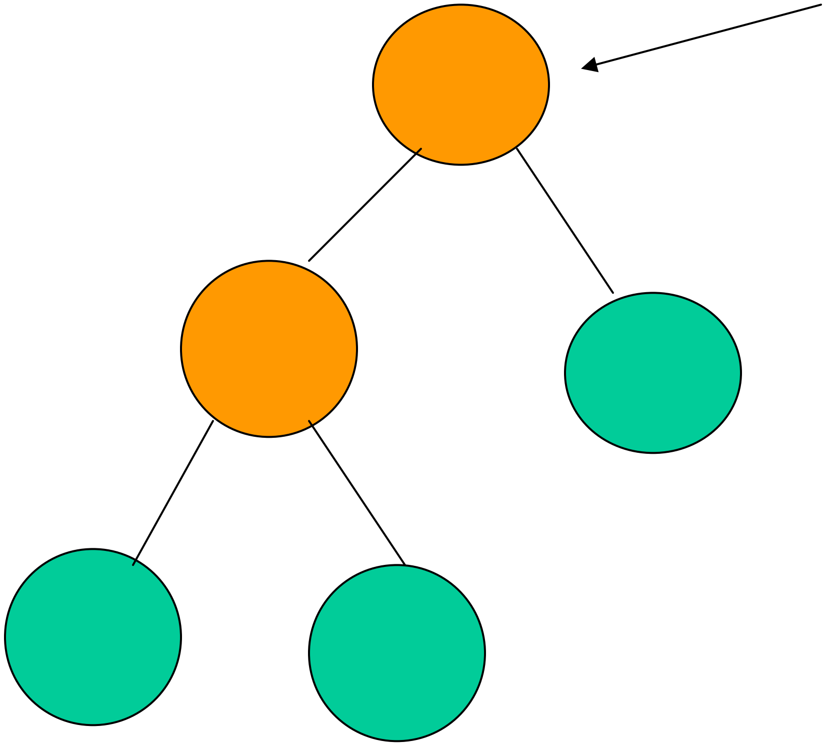
- Get DNA/RNA/Protein Sequences
- Construct multiple sequence alignment
- Compute pairwise distances
  - (for distance methods)
- Build tree:  topology + branch lengths
- Estimate reliability
- Visualize

Frank Olken, 2002

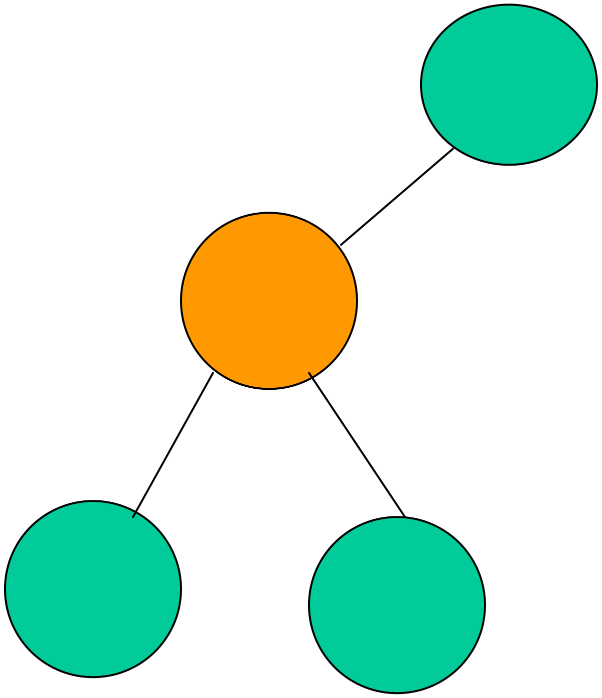
# Phylogeny Tree Space

- The space of phylogeny tree is exponential.
- For n sequences, the number of unrooted tree is (2n-5)!!
- For n sequences, the number of rooted tree is (2n-3)!!

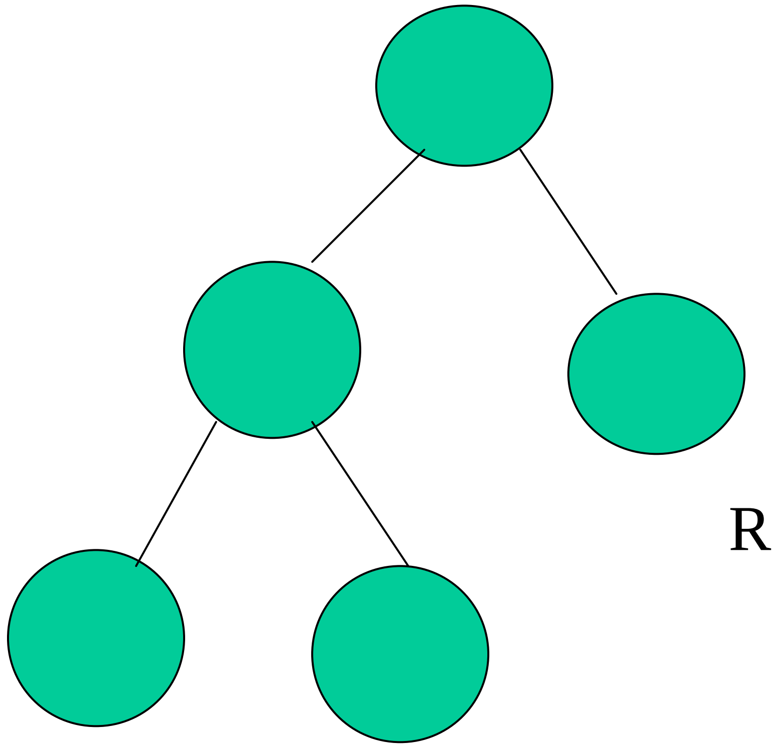
Root



Rooted Tree



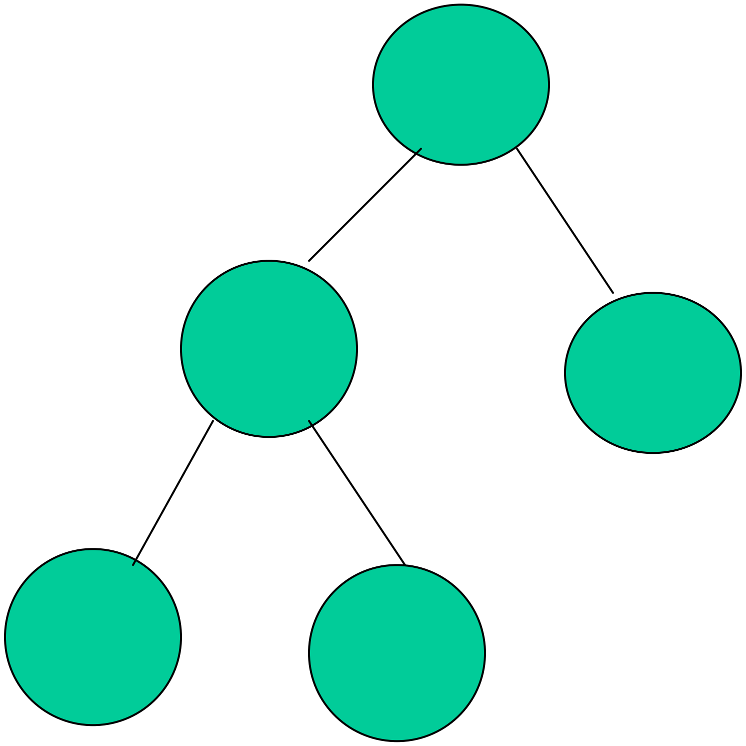
Unrooted Tree



Definition of a tree:
Edge num: E
Internal node num: I
Leaf node num: L

Rule 1: $E = I + L - 1$ (why?)

# Rooted Phylogeny Tree



$E = 2 * I$ (degree)

$E = I + L - 1$

↓

$2I = I + L - 1$

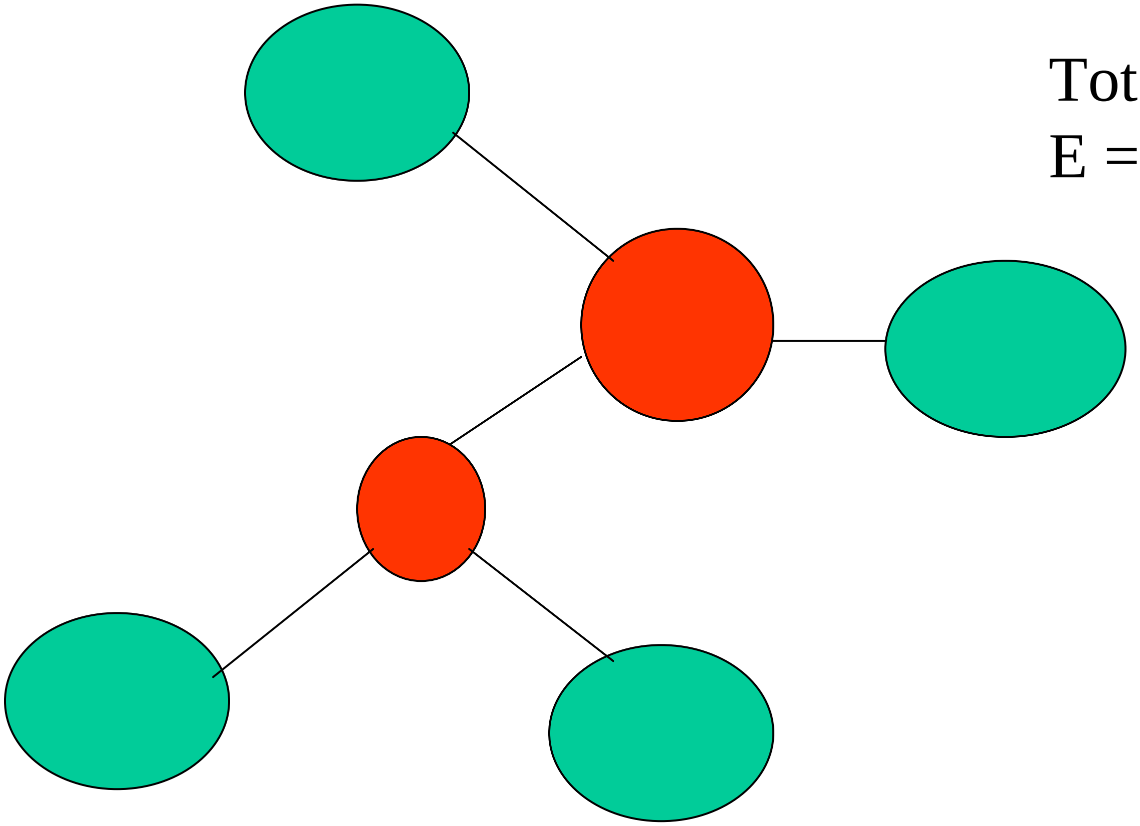
↓

$I = L - 1$ (# internal node = #leaf – 1)

↓

$E = 2L - 2$

# Un-Rooted Tree



Total degree = $(L+3*I) = 2E$

$E = L + I - 1$

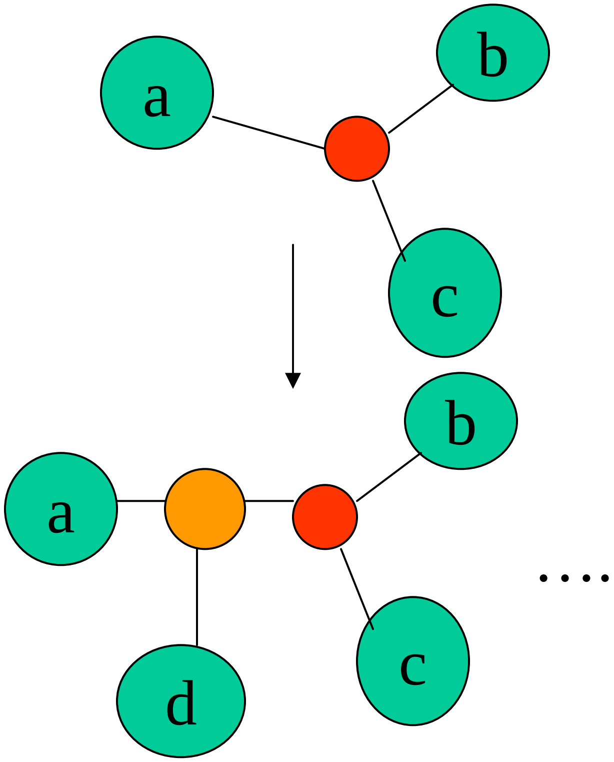
$I = L - 2$

$E = 2L - 3$

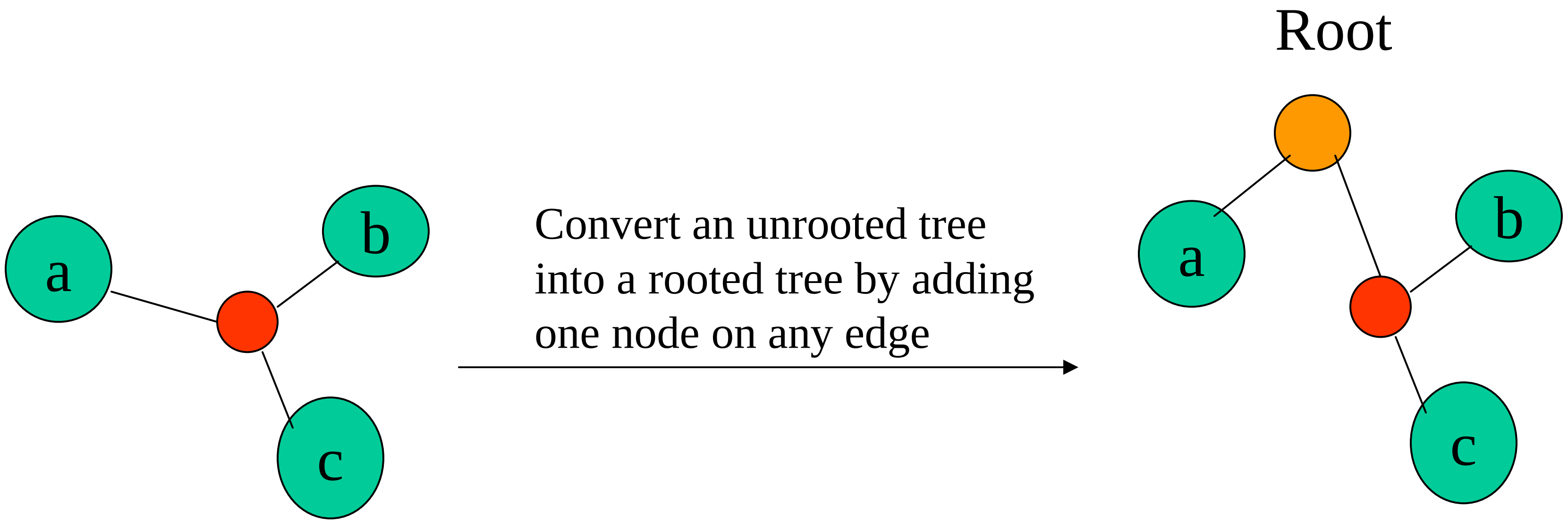
# Total number of unrooted tree

- Given n species (n >= 3), there are (2i-5)!! Unrooted bifurcating trees.

| #leaf node | #edge | #tree |
|---|---|---|
| 3 | 3 | 1 |
| 4 | 5 | 1*3 |
| 5 | 7 | 1*3*5 |
| … | | |
| n | 2n-3 | 1*3*5*…*2n-5 |



# Total number of rooted tree



Convert an unrooted tree into a rooted tree by adding one node on any edge

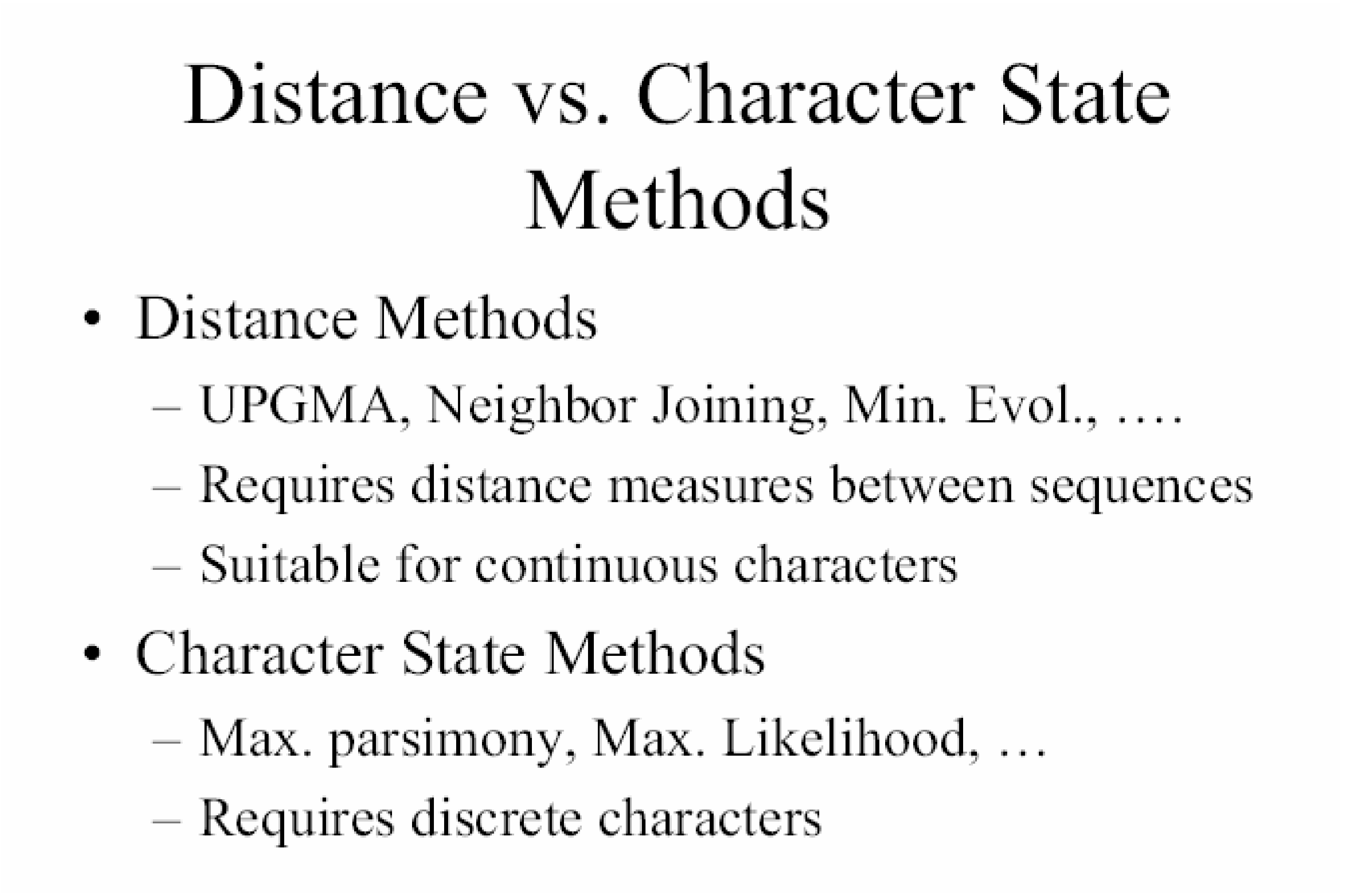
Total number of rooted tree for n leaf nodes:
Total number of unrooted tree * total number of edge
= (2n-5)!! * (2n-3) = (2n-3)!!

# Number of Rooted and Unrooted Trees

| | Unrooted | rooted |
|---|---|---|
| 3 | 1 | 3 |
| 4 | 3 | 15 |
| 5 | 15 | 105 |
| 6 | 105 | 945 |
| 7 | 945 | 10395 |
| 8 | 10395 | 135135 |
| 9 | 135135 | 2027025 |
| 10 | 2027025 | 34459425 |

# Phylogeny Tree Algorithms

- Distance-Based (UPGMA, Neighbor Join)
- Maximum Parsimony (character-based)
- Maximum Likelihood (character-based)



# Distance vs. Character State Methods

- Distance Methods
  - UPGMA, Neighbor Joining, Min. Evol., ….
  - Requires distance measures between sequences
  - Suitable for continuous characters
- Character State Methods
  - Max. parsimony, Max. Likelihood, …
  - Requires discrete characters

Frank Olken, 2002

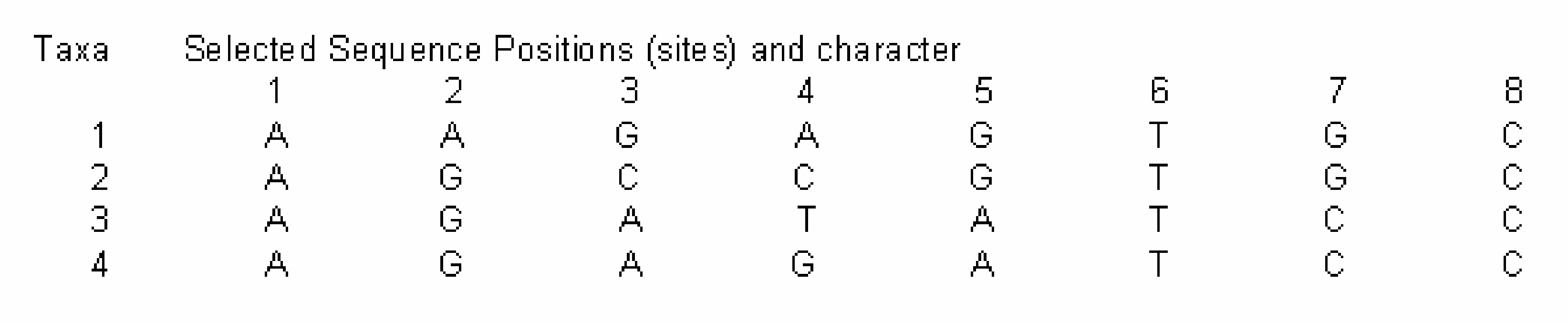
# How to choose methods

- Very similar sequences: Maximum Parsimony (time intensive)
- Medium similar sequences: distance based method (fast, $O(n^2)$)
- Very dissimilar sequences: Maximum Likelihood method (very time intensive)

# Maximum Parsimony Method

- Predict the evolutionary tree that minimizes the number of steps required to generate the observed variation in the sequences.
- Find a tree that explains data with a minimal number of changes.
- Appropriate for very similar sequences and a small number of sequences
- Time Consuming (try to examine all possible trees)
- PHYLIP and PAUP offer maximum parsimony method

# Select Informative Sites

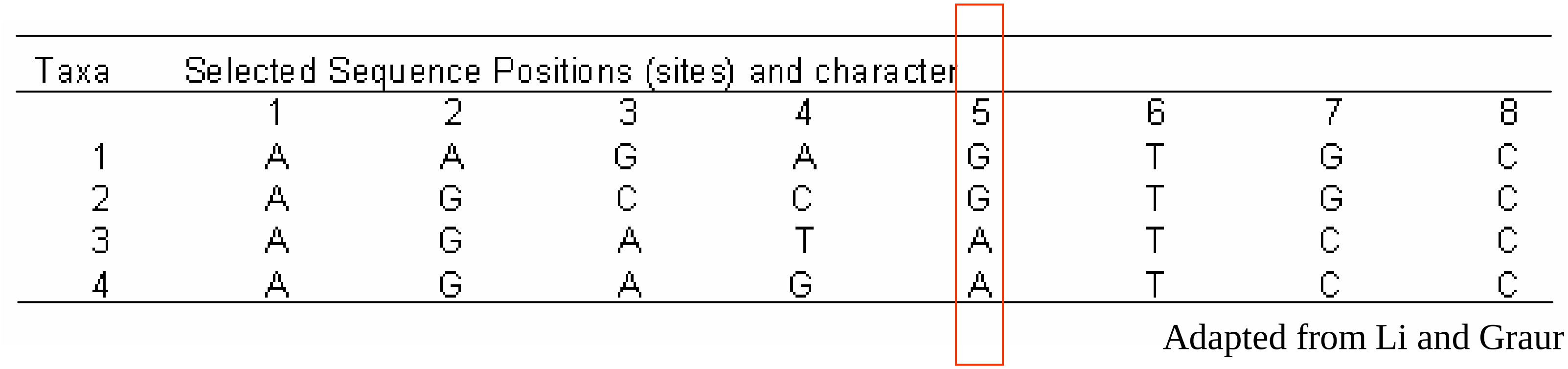


| Taxa | Selected Sequence Positions (sites) and character | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | A | A | G | A | G | T | G | C |
| 2 | A | G | C | C | G | T | G | C |
| 3 | A | G | A | T | A | T | C | C |
| 4 | A | G | A | G | A | T | C | C |

Sites 1,6,8: not informative<br>
Site 2: not informative (doesn’t favor any tree)<br>
Site 3: not informative (doesn’t favor any tree)<br>
Site 4: not informative (doesn’t favor any tree)<br>
**Sites 5, 7: informative**

**Rule of thumb: to be informative, a character must appear in at least two taxa and there are at least two characters.**

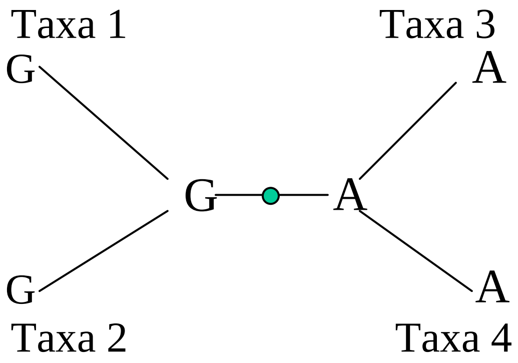
# Example



| Taxa | Selected Sequence Positions (sites) and character | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | A | A | G | A | G | T | G | C |
| 2 | A | G | C | C | G | T | G | C |
| 3 | A | G | A | T | A | T | C | C |
| 4 | A | G | A | G | A | T | C | C |

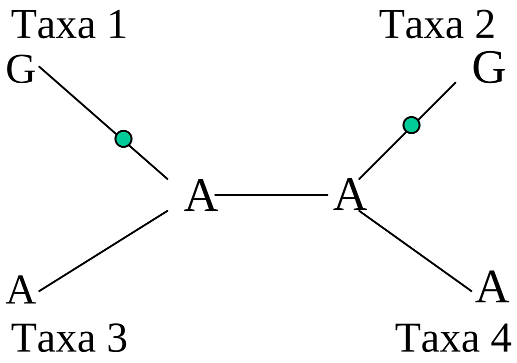
Adapted from Li and Graur 1991

**Tree 1**



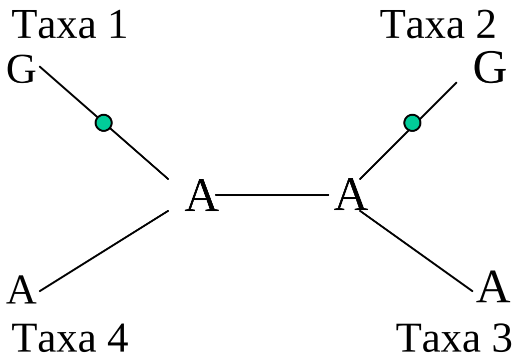
Length = 1

Tree 2



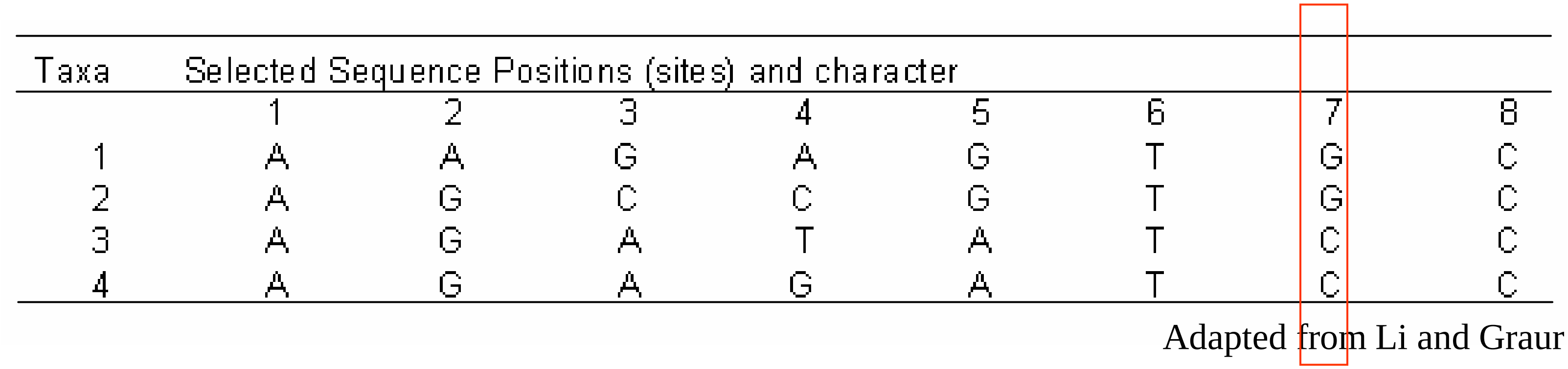
Length = 2

Tree 3



Length = 2

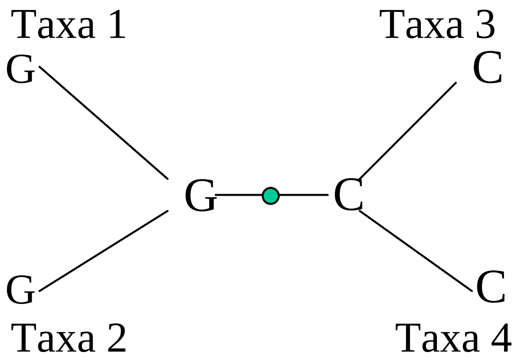
# Example



| Taxa | Selected Sequence Positions (sites) and character | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | A | A | G | A | G | T | G | C |
| 2 | A | G | C | C | G | T | G | C |
| 3 | A | G | A | T | A | T | C | C |
| 4 | A | G | A | G | A | T | C | C |

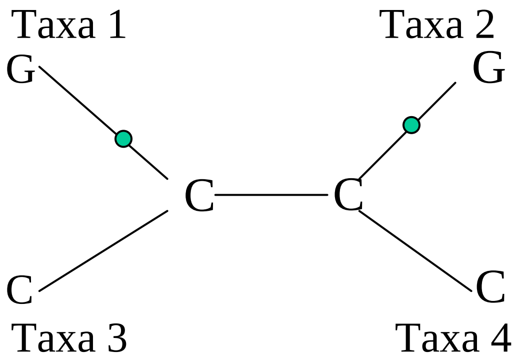
Adapted from Li and Graur 1991

**Tree 1**



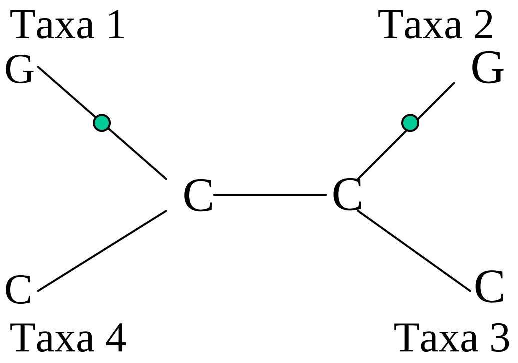
Length = 1

Tree 2



Length = 2

Tree 3



Length = 2

# Comments

- For a small number of sequences, exhaustive search is ok. (n=10, about 2 million trees)
- For many sequences, exhaustive search is very time consuming (NP-Complete). Branch-Bound method or even heuristic methods must be used. (PAUP provides both options)
- Maximum parsimony tree provides an explicit evolutionary model
- The rates of changes along all branches of the tree are assumed to be equal
- PHYLIP: DNAPENNY (branch-bound to analyze up to 11-12 sequences), DNACOMP performs phylogenetic analysis using the compatibility criterion (find a tree that supports the largest number of sites). PROTPARS (for protein parsimony tree)

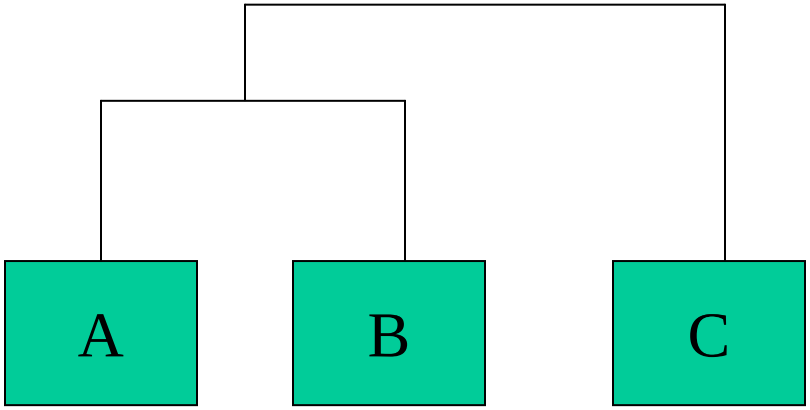
# Distance-Based Method

- Goal is to generate a tree in which similar sequences with short distance are closer and the sum of branch lengths of two nodes is equal to their distance.
- ClustalW (neighbor-join method)
- PAUP also has distance method
- PHYLIP: DNADIST, PROTDIST (PAM) to generate distance matrix

# UPGMA

- UPGMA: unweighted pair group method with arithmetic mean.
- Assume a molecular clock (constant evolution rate)
- Produce a rooted tree
- Ultrametric condition: for any three taxa (a,b,c), $d_{ac}$ <= max($d_{ab}$, $d_{bc}$).

# UPGMA condition



dAB <= max(dAC,dBC)
dAC <= max(dAB,dBC)
dBC <= max(dAB, dAC)

In another words: two greatest distance must be equal.
Or: constant evolutionary rate for all branches.

# UPGMA Algorithm

**Initialization**: Define T to be the set of leaf nodes, one for each sequence. Height of each node is 0. Let L = T

**Repeat**

- Select closest two  nodes (A,B) and create parent node K for them. Join A, B and K respectively. Set the height of node K to $d_{AB}$ / 2. Set branch length between K and A = height K -  height A, set branch length between K and B = height K – height B.
- Remove A, B from L and add K into L. Re-compute the distance between K and other nodes in L. Distance between K and other nodes is average distance of leaf sequences below K and the other node.

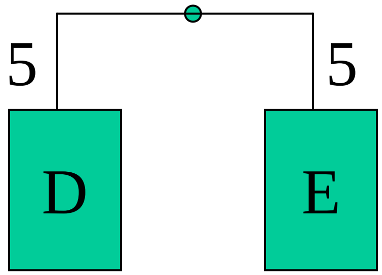
**Until** there is only one node

# Example of UPGMA (perfect)

Step 1: Select D and E

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | - | 20 | 26 | 26 | 26 |
| B | | - | 26 | 26 | 26 |
| C | | | - | 16 | 16 |
| D | | | | - | 10 |
| E | | | | | - |

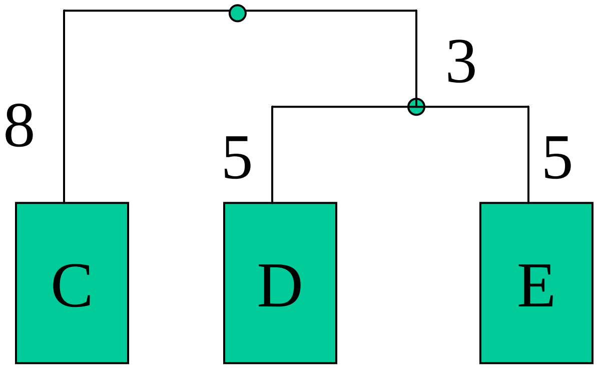
| | A | B | C | DE |
|---|---|---|---|---|
| A | - | 20 | 26 | 26 |
| B | | - | 26 | 26 |
| C | | | - | 16 |
| DE | | | | - |



## Step 2: Select (DE) and C

| | A | B | C | DE |
|---|---|---|---|---|
| A | - | 20 | 26 | 26 |
| B | | - | 26 | 26 |
| C | | | - | 16 |
| DE | | | | - |

| | A | B | DEC |
|---|---|---|---|
| A | - | 20 | 26 |
| B | | - | 26 |
| DEC | | | - |



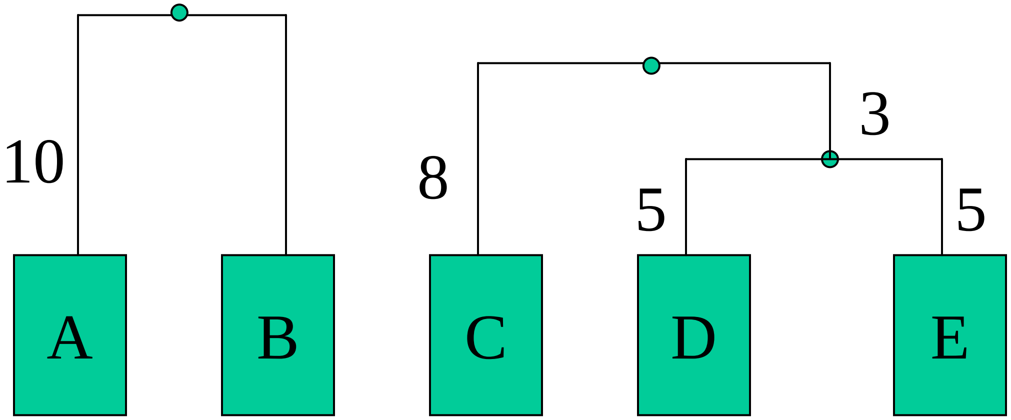
$\text{dist(DEC,A)} = (d_{DA}+d_{EA}+d_{CA})/3 = 26$

$\text{dist(DEC,B)} = (d_{DB}+d_{EB}+d_{CB})/3 = 26$

Step 3: select A, B

|  | A | B | DEC |
|---|---|---|---|
| A | - | 20 | 26 |
| B |  | - | 26 |
| DEC |  |  | - |

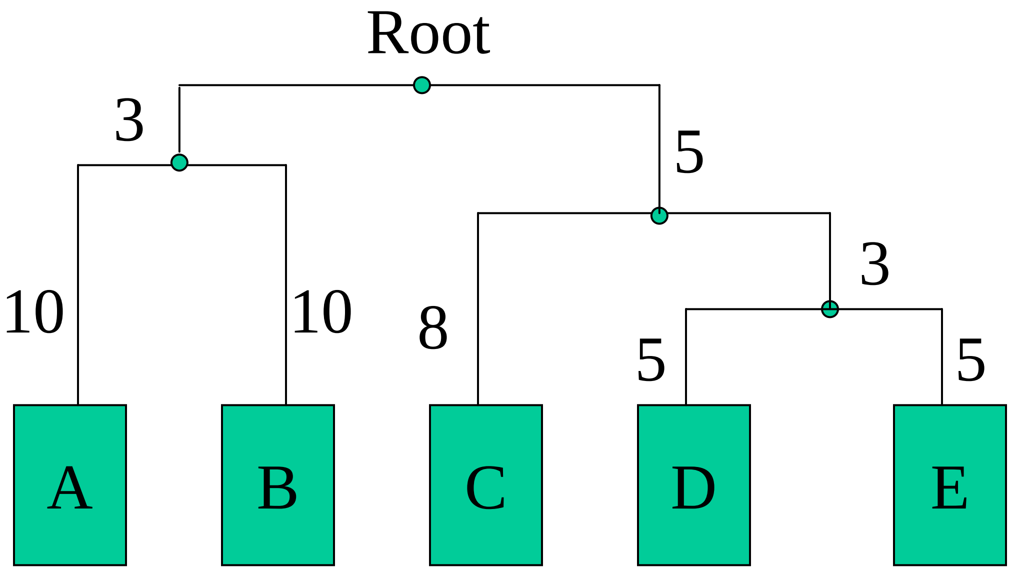
|  | AB | DEC |
|---|---|---|
| AB | - | 26 |
| DEC |  | - |



$$dist(DEC,AB)= (d_{DA}+d_{DB}+d_{EA}+d_{EB}+d_{CA}+d_{CB})/6$$
$$= 26$$

|  | AB | DEC |
|---|---|---|
| AB | - | 26 |
| DEC |  | - |

Step 4: select (A,B), (D,E,C)

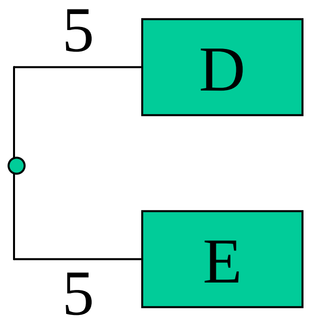


# Example of UPGMA (imperfect)

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | - | 22 | 39 | 39 | 41 |
| B |   | - | 41 | 41 | 43 |
| C |   |   | - | 18 | 20 |
| D |   |   |   | - | 10 |
| E |   |   |   |   | - |

|   | A | B | C | DE |
|---|---|---|---|---|
| A | - | 22 | 39 | 40 |
| B |   | - | 41 | 42 |
| C |   |   | - | 19 |
| DE |   |   |   | - |

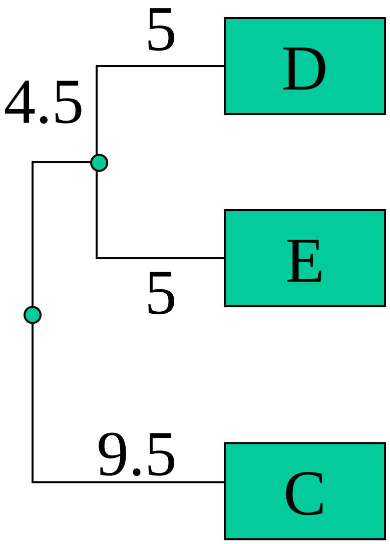
Step 1: Select D and E



## Step 2: Select (DE) and C

|  | A | B | C | DE |
|---|---|---|---|---|
| A | - | 22 | 39 | 40 |
| B |  | - | 41 | 42 |
| C |  |  | - | 19 |
| DE |  |  |  | - |

|  | A | B | DEC |
|---|---|---|---|
| A | - | 22 | 39.7 |
| B |  | - | 41.7 |
| DEC |  |  | - |



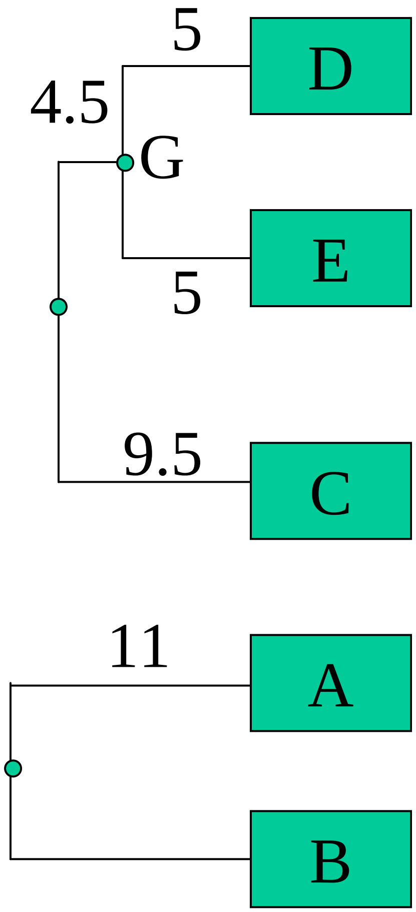
$\text{dist(DEC,A)} = (d_{DA}+d_{EA}+d_{CA})/3 = 39.7$

$\text{dist(DEC,B)} = (d_{DB}+d_{EB}+d_{CB})/3 = 41.7$

## Step 3: select A, B

|  | A | B | DEC |
|---|---|---|---|
| A | - | 22 | 39.7 |
| B |  | - | 41.7 |
| DEC |  |  | - |

|  | AB | DEC |
|---|---|---|
| AB | - | 40.7 |
| DEC |  | - |

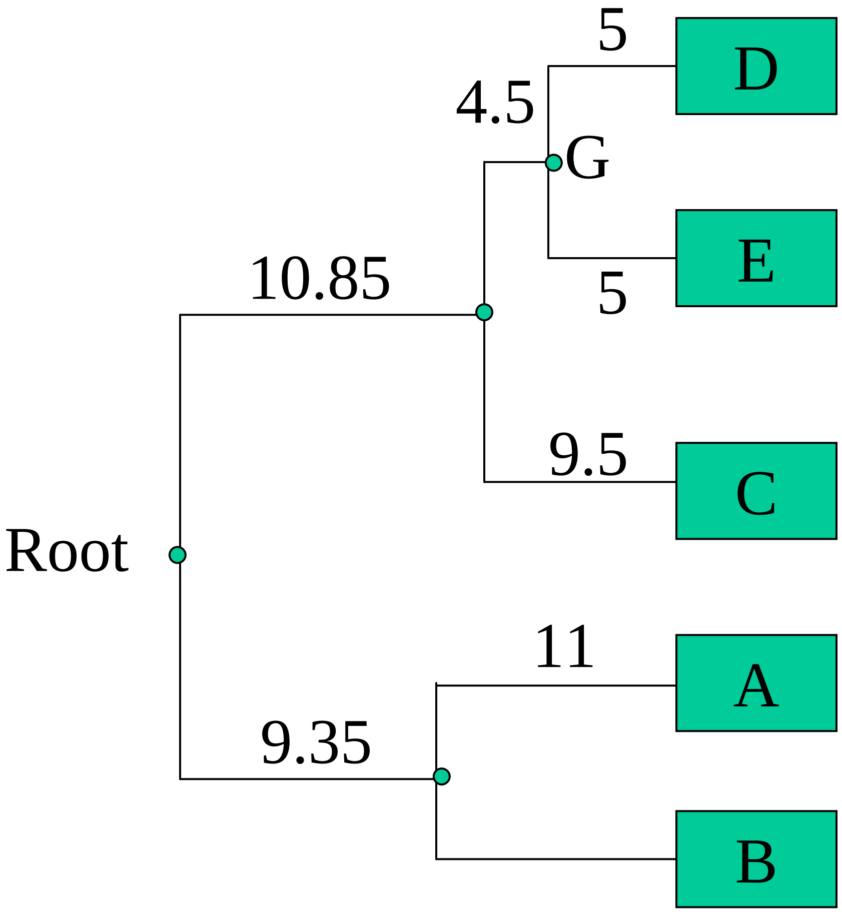


$$\mathrm{dist(DEC,AB)} = (d_{DA}+d_{DB}+d_{EA}+d_{EB}+d_{CA}+d_{CB})/6$$

$$=$$

Step 4: select (A,B), (D,E,C)

|  | AB | DEC |
|---|---|---|
| AB | - | 40.7 |
| DEC |  | - |



20.35 – 9.5 = 10.85

20.35 – 11 = 9.35

# Neighbor-Join Method

- Do not assume molecular clock
- Assume additivity of distance matrix (ideal)
- Work for non-additivity of distance matrix (non-ideal situation)
- Most reliable when the branch lengths of trees are allowed to vary
- Goal is to find a tree that minimize the square errors of pairwise distances.

# Additive Condition

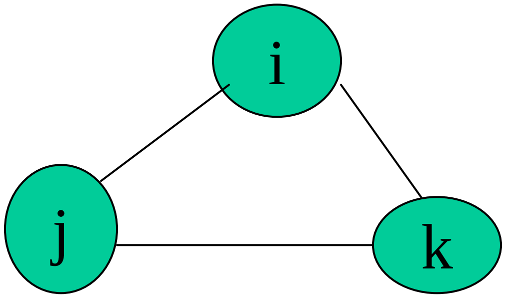
Given a L*L distance matrix M,

$d(i,i) = 0$,

$d(i,j) > 0$ for $i \neq j$

$d(i,j) = d(j,i)$

For all i,j,k it holds that $d(i,k) <= d(i,j) + d(j,k)$

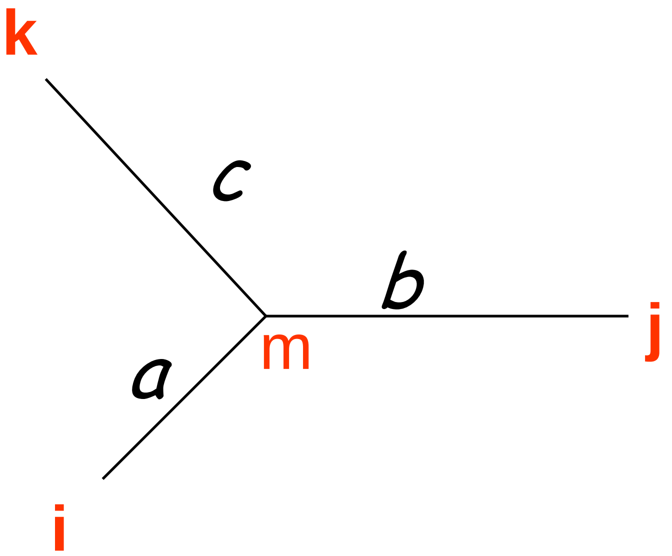


# Additive Condition (cont)

We say that the distance matrix M with L objects is **additive** if there is a tree T, *L* of its nodes correspond to the *L* objects, with positive weights on the edges, such that for all $i,j$, $d(i,j) = d_T(i,j)$, the length of the path from $i$ to $j$ in T.

# Three objects sets always additive:

For L=3: There is always a tree with one internal node.

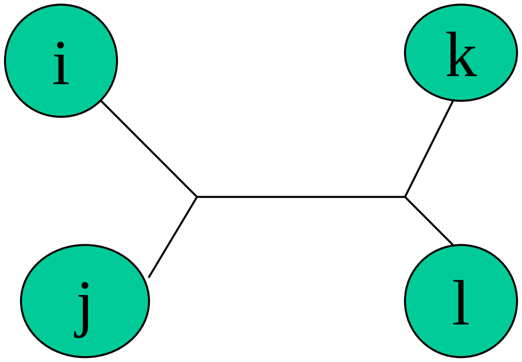


$$d(i,j) = a + b$$
$$d(i,k) = a + c$$
$$d(j,k) = b + c$$

Thus

$$c = d(k,m) = \frac{1}{2}[d(i,k) + d(j,k) - d(i,j)] \geq 0$$

# Additive Tree



↕

For four point condition:

$d(i,k)+d(j,l) = d(i,l) + d(k,j) >= d(i,j)+d(k,l)$

- Theorem: A distance matrix M of $L$ objects is additive iff *any* subset of four objects can be labeled $i,j,k,l$ so that:
- $d(i,k) + d(j,l) = d(i,l) + d(k,j) \geq d(i,j) + d(k,l)$
- We call $\{\{i,j\},\{k,l\}\}$ the “split” of $\{i,j,k,l\}$.

# Neighbor Join Algorithm

- **Initialization:**

  Define T to be the set of leaf nodes, one for each sequence. And let L = T
- **Iteration**:

  Pick a pair i,j for which $D_{ij}$ is minimal.

  $D_{ij} = d_{ij} – (r_i+r_j)$, $r_i = \sum d_{ik}$ / (|L|-2) , $r_i$: average distance from i to all other sequences (k) except j.

  Remove i, j from L.

  Define a new node k, for other node m in L, $d_{km}=1/2*(d_{im}+d_{jm}-d_{ij}) = (d_{im}+d_{jm})/2 – d_{ij}/2$

  Add k to T with edge of length $d_{ik} = ½(d_{ij}+r_i-r_j)$, $d_{jk}=d_{ij}-d_{jk}$

  Join k to i and j respectively, remove i,j from L and add k into L
- **Termination**

  When L consists of only two nodes (i,j). Add one edge between i and j with length $d_{ij}$.

# Example of Neighbor Join

|   | A | B | C | D |
|---|---|---|---|---|
| A | - | 3 | 7 | 8 |
| B |   | - | 6 | 7 |
| C |   |   | - | 3 |
| D |   |   |   | - |

Satisfy additive condition

$D_{AB} = 3 – ( (7+8)/2 + (6+7)/2) =-11$

$D_{AD} = 8 – ( (3+7)/2 + (3+7)/2) = -2$

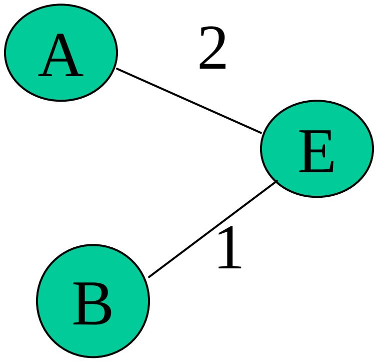
$D_{AC} = 7 – ( (3+8)/2 + (6+3)/2) = -3$

$D_{BC} = 6 – ( (3+7)/2 + (7+3)/2) = -4$

$D_{BD} = 7 – ( (3+6)/2 + (8+3)/2) = -3$

$D_{CD} = 3 – ( (7+6)/2 + (8+7)/2) = -11$

Step 1: Select A and B



$d_{EC} = (d_{AC}+d_{BC}-d_{AB})/2 = (7+6-3)/2=5$

$d_{ED\ =}\ (d_{AD}+d_{BD} –d_{AB})/2 = (8+7-3)/2=6$

$d_{AE} = (d_{AB}+r_A-r_B)/2=(3+1)/2 = 2$

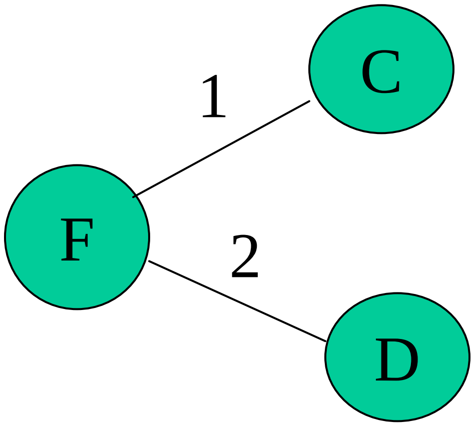
$d_{BE} = d_{AB} – d_{AE} = 3 – 2 = 1$

|   | E | C | D |
|---|---|---|---|
| E |   | 5 | 6 |
| C |   |   | 3 |
| D |   |   |   |

$D_{CD} = d_{CD} – (r_C+r_D)=3-(5+6)=-8$

$D_{CE} = d_{CE}-(r_C+r_E)=5-(3+6)=-4$

$D_{DE}= d_{DE}-(r_D+r_E) = 6-(3+5) = -2$

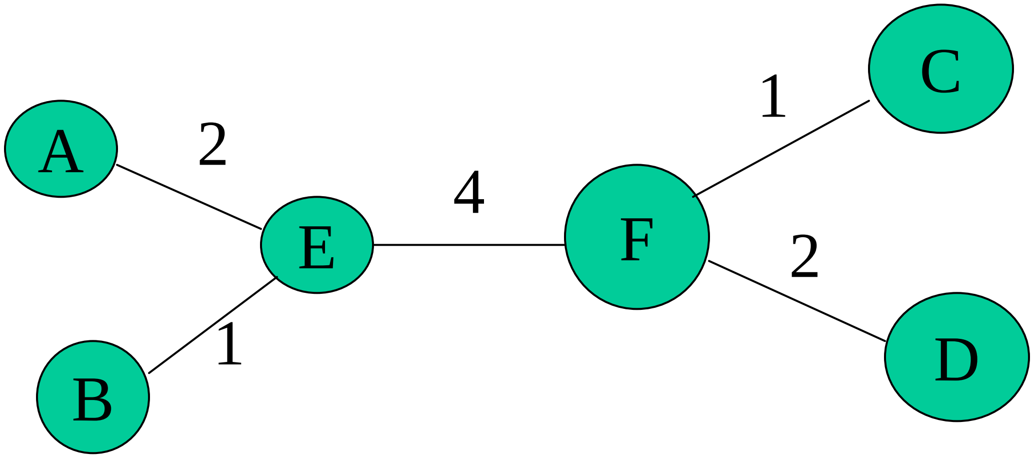


$d_{FE} = (d_{CE}+d_{DE}-d_{CD})/2=(5+6-3)/2=4$

$d_{FC} = (d_{CD}+r_C-r_D)/2=(3+5-6)/2=1$

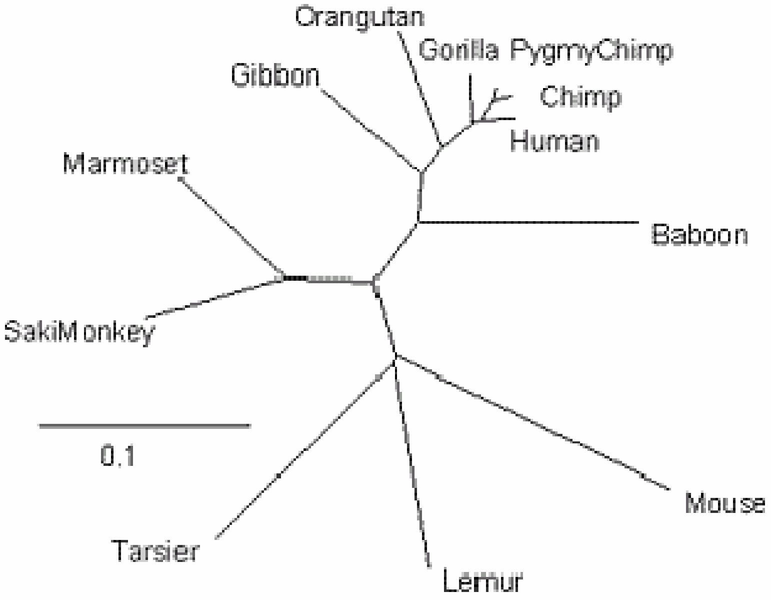
$d_{FD} = d_{CD} – d_{FC}= 2$

Join E and F with length 4

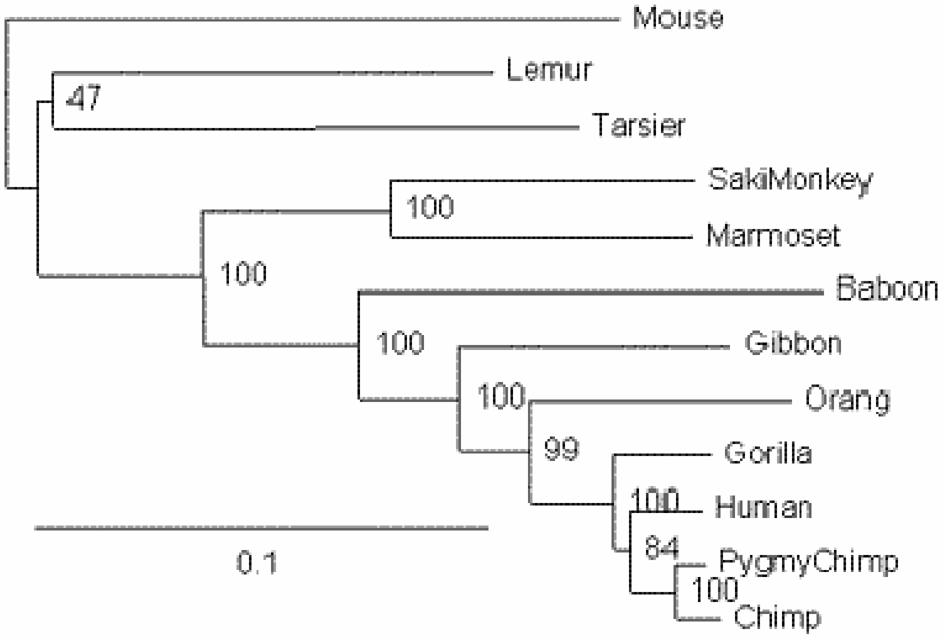


Now, verify if the sum of branch lengths matches with sequence distances.

Additive means distance between species = distance summed along internal branches



The tree has been rooted using the Mouse as outgroup



S. Moran and I. Wexler

# Comments

- Given a distance matrix constituting an additive metric, the topology of the corresponding additive tree is unique.
- Can run NJ algorithm on non-additive matrix. In that case, tree may not be unique.

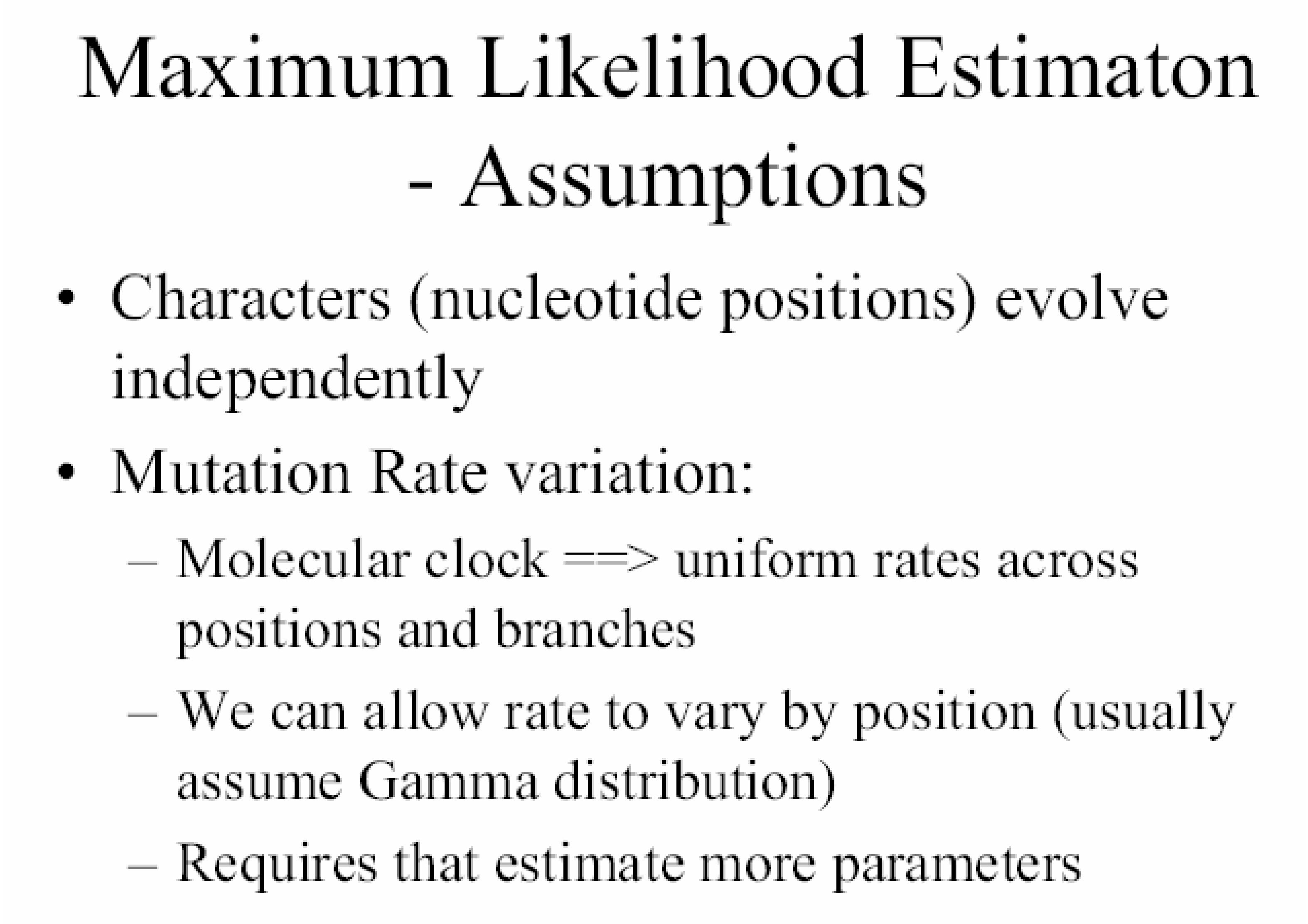
# How to compute distance from alignment

- Count mismatch
- Count both mismatch and gaps
- Use substitution matrix to generate similarity scores, then convert it to distance

Normalize score into the range [0,1]: $S = (S_{real} – S_{rand}) / (S_{ident} – S_{rand})$. Then 1 – normalized score is distance. $S_{real}$ is the alignment score of sequence A and B. $S_{rand}$ is the alignment score of two random sequences generated from A and B. $S_{ident}$ is average alignment score of aligning A with A, B with B.

# Maximum Likelihood Method

- Use probability calculations to find a tree that best accounts for variation in a set of sequences.
- Analysis is performed on each column of a multiple sequence alignment.
- All possible trees are considered. Only can be used for a small number of sequences (at most 10 ?)
- PAUP version 4 and up has maximum likelihood function
- PHYLIP (DNAML and DNAMLK (molecular clock))



# Maximum Likelihood Estimaton - Assumptions

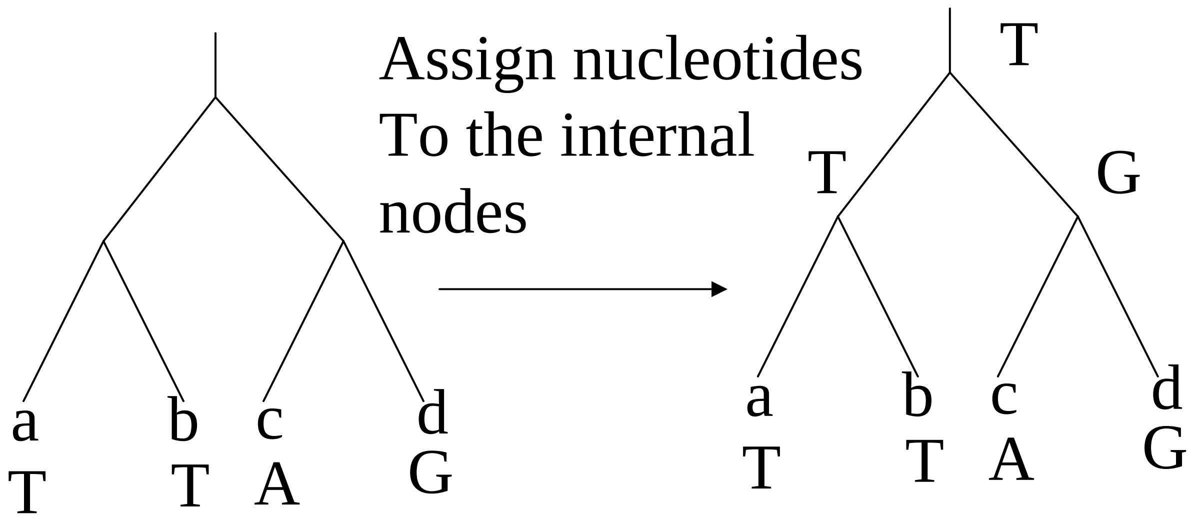
- Characters (nucleotide positions) evolve independently
- Mutation Rate variation:
  - Molecular clock ==> uniform rates across positions and branches
  - We can allow rate to vary by position (usually assume Gamma distribution)
  - Requires that estimate more parameters

Frank Olken, 2002

# Example of ML

```
Sequence A: ACGCGTTGGG
Sequence B: ACGCGTTGGG
Sequence C: ACGCAATGAA
Sequence D: ACACAGGGAA
```

Look at one column (5 possible columns), one tree (15 Possible trees), one assignment of nucleotides (64 possible combinations)

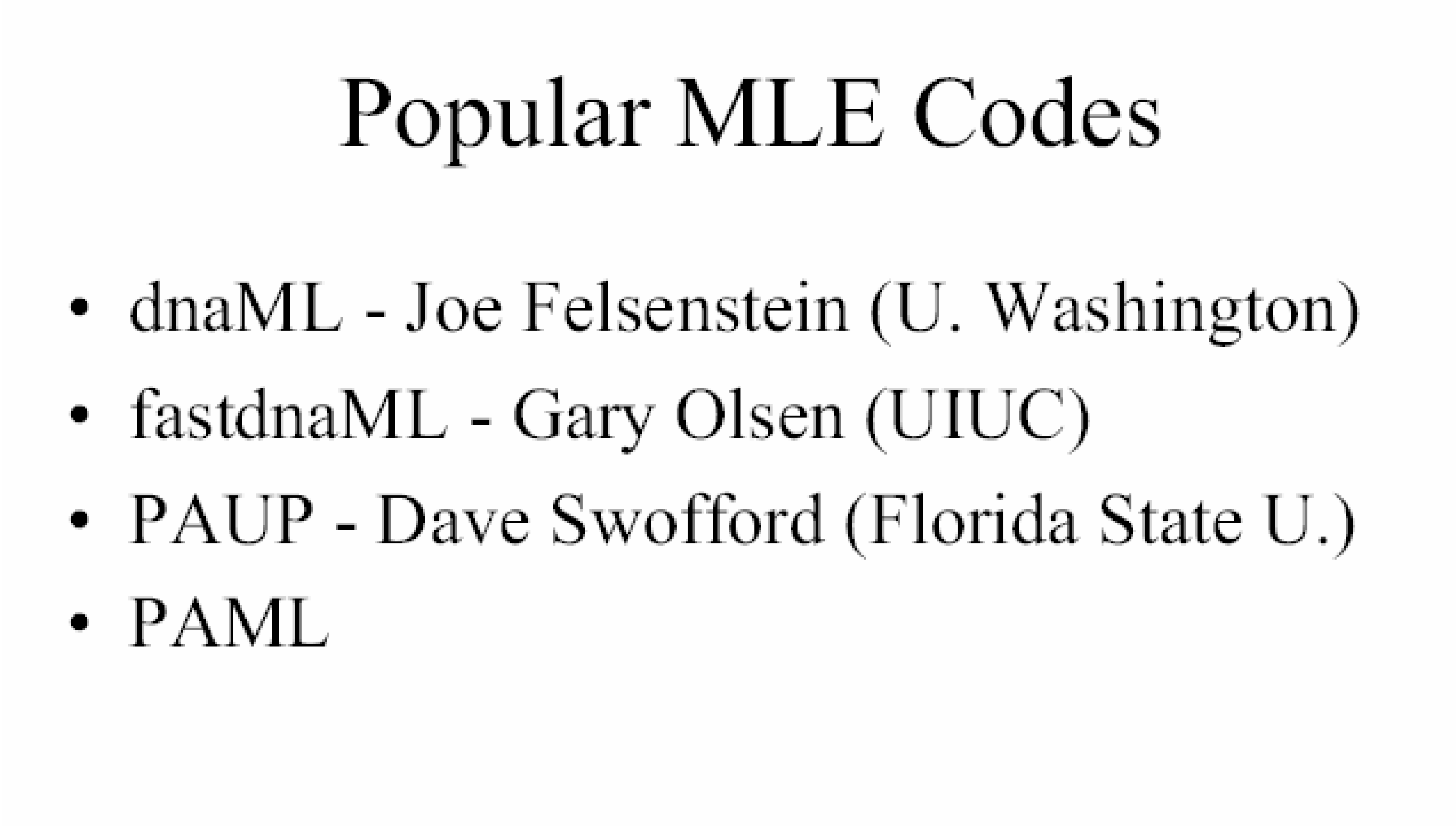

Assign nucleotides To the internal nodes

Prob = prob(T) *Prob(T->G)* Prob(G->A) $=0.25*10^{-6}*10^{-6}$

Find a best tree that with maximum probability.

# Advantages and Disadvantages of ML Method

- Explicit Statistical Model
- Likelihood
- Efficient use of data
- Very expensive to compute (use heuristics)

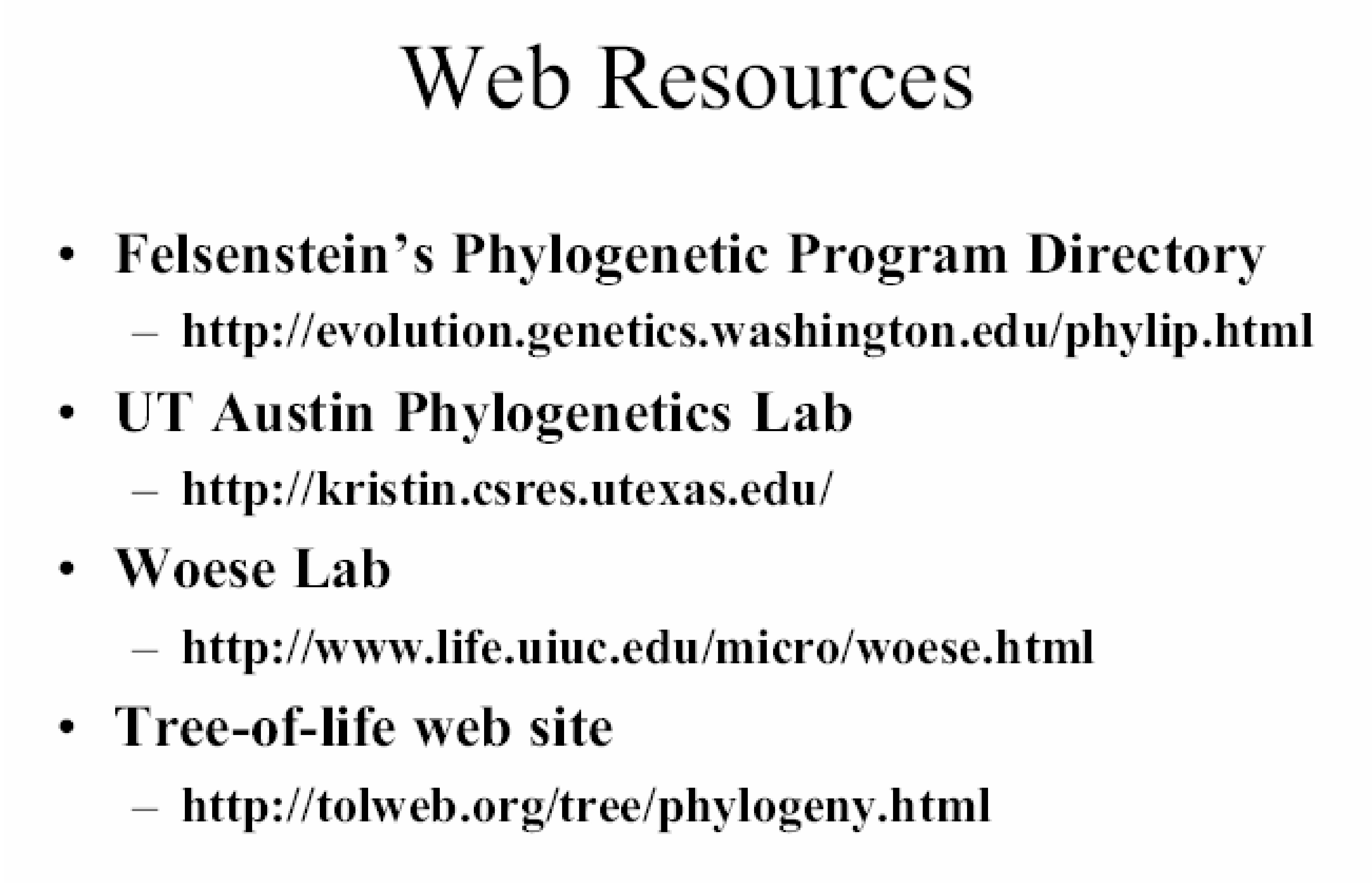


# Popular MLE Codes

- dnaML - Joe Felsenstein (U. Washington)
- fastdnaML - Gary Olsen (UIUC)
- PAUP - Dave Swofford (Florida State U.)
- PAML

# Reliability of Tree Construction

- **Boostraping**
- Given an multiple sequence alignment, randomly sample n-columns with replacement and construct a tree
- Construct a lot of trees as above
- Check the relation between two sequences. If their relationship (split, branch) is stable (appearing in the most trees), then the tree more likely close to the true tree.



# Web Resources

- **Felsenstein's Phylogenetic Program Directory**
  - **http://evolution.genetics.washington.edu/phylip.html**
- **UT Austin Phylogenetics Lab**
  - **http://kristin.csres.utexas.edu/**
- **Woese Lab**
  - **http://www.life.uiuc.edu/micro/woese.html**
- **Tree-of-life web site**
  - **http://tolweb.org/tree/phylogeny.html**

# Ten Topics

- 1. Introduction to Molecular Biology and Bioinformatics
- 2. Pairwise Sequence Alignment Using Dynamic Programming
- 3. Practical Sequence/Profile Alignment Using Fast Heuristic Methods (BLAST and PSI-BLAST)
- 4. Multiple Sequence Alignment
- 5. Gene Identification
- 6. **Phylogenetic Analysis**
- 7. Protein Structure Analysis and Prediction
- 8. RNA Secondary Structure Prediction
- 9. Clustering and Classification of Gene Expression Data
- 10. Search and Mining of Biological Databases, Databanks, and Literature