# Phylogenetic Methods

9 December 2024

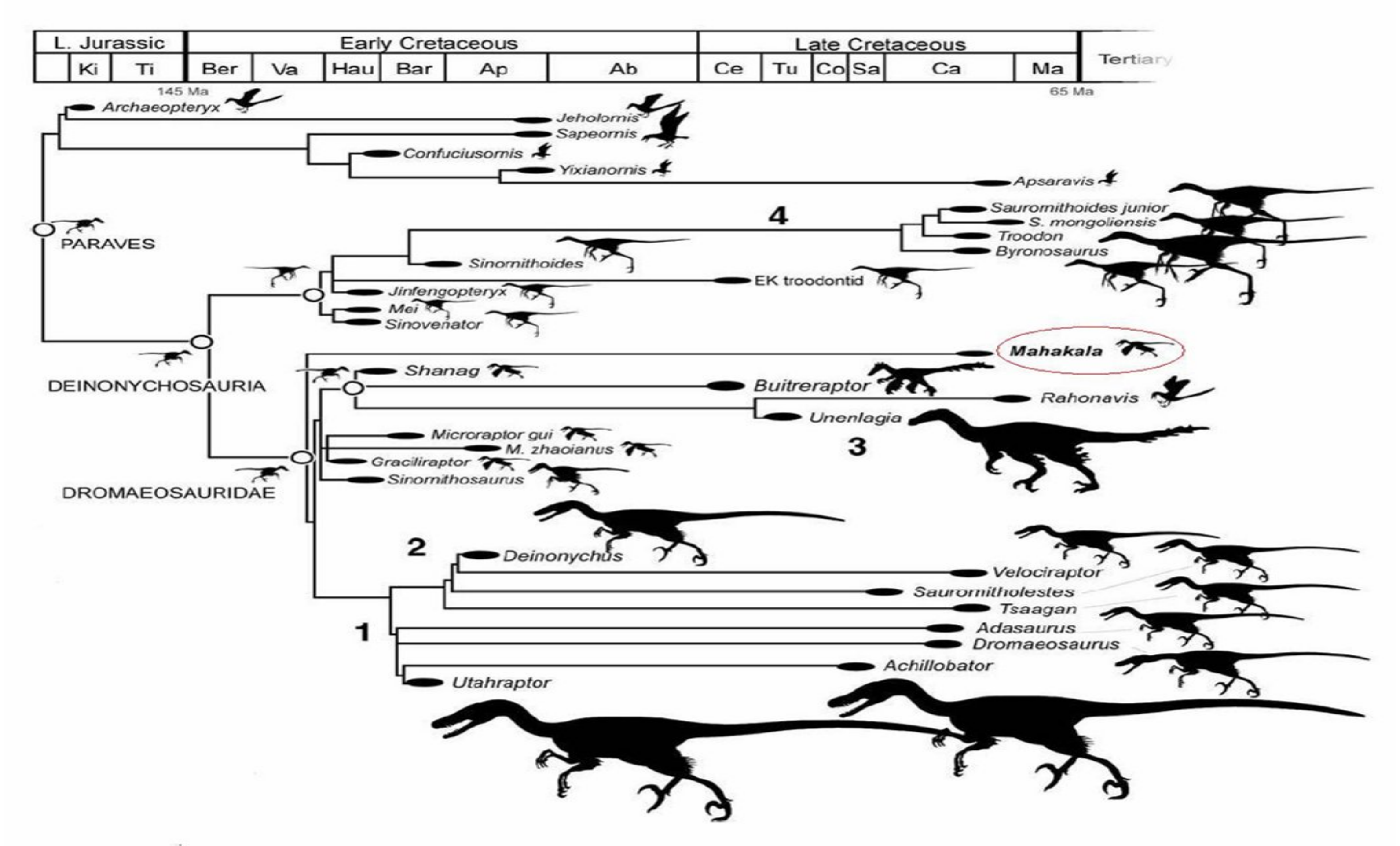
Slides courtesy of
Mark Craven
University of Wisconsin-Madison

## Background

- **Phylogenetic tree**: diagram showing evolutionary lineages of species/sequences/genes

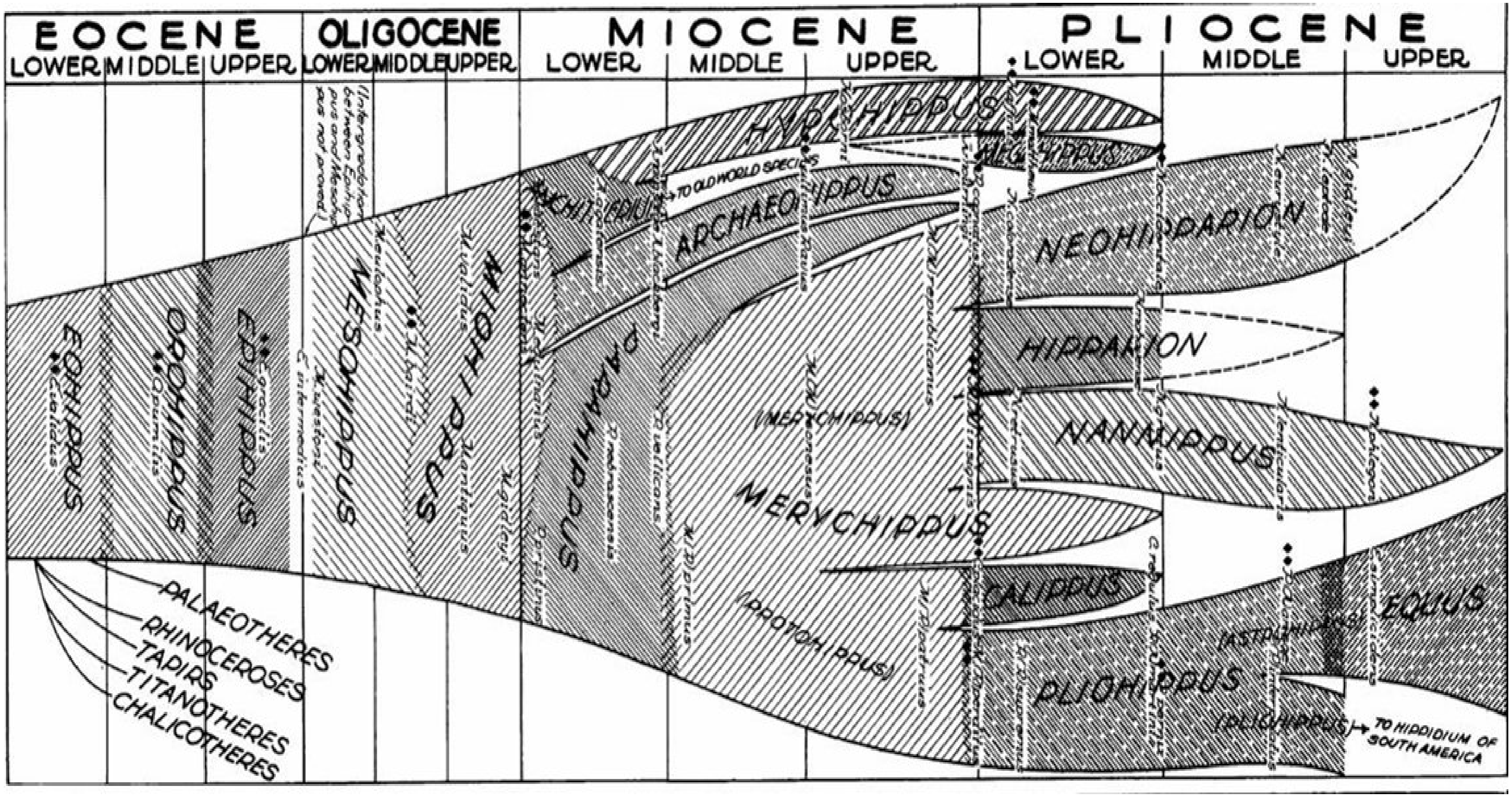
- **Why construct trees**?
  - to try to explain the evolutionary history of species
  - to understand the lineage of various species
  - to understand how various functions evolved
  - to inform multiple alignments

# Dinosaurs



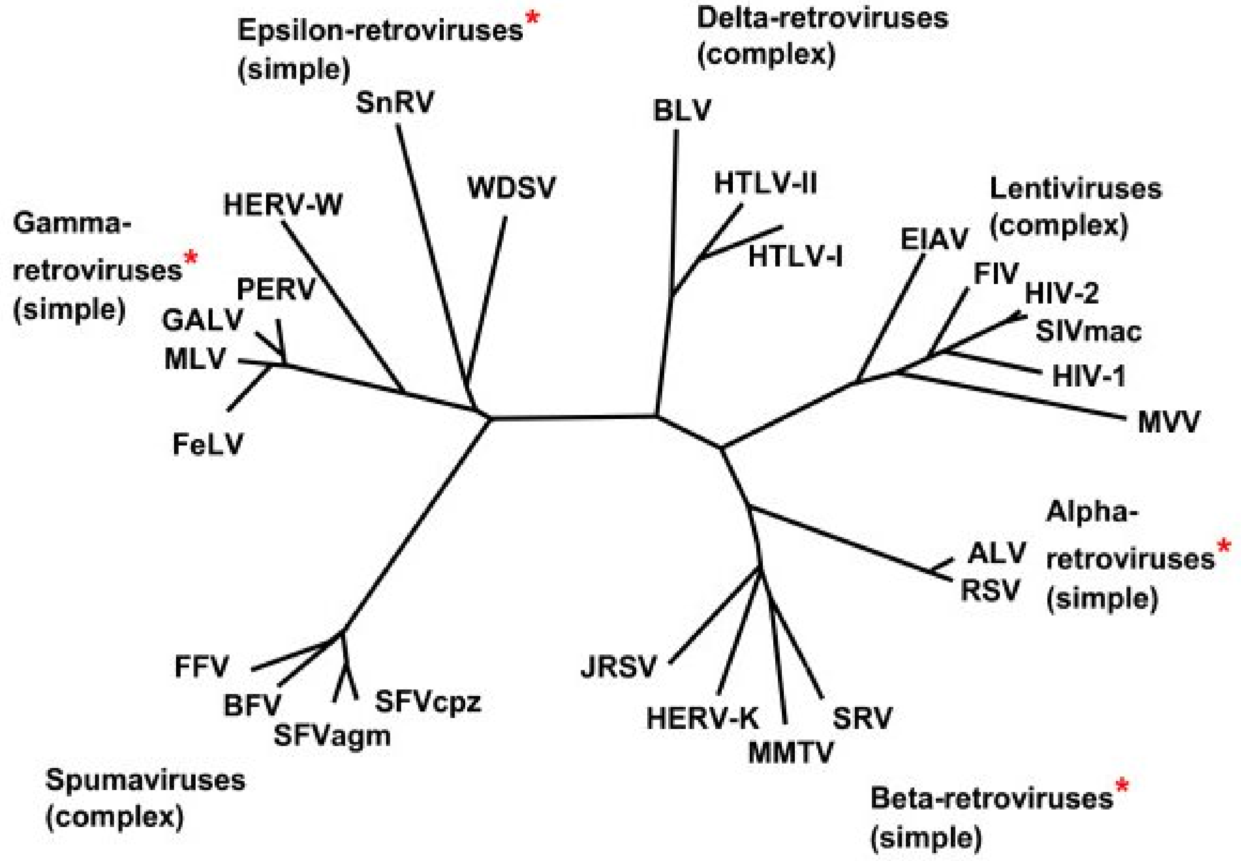
Turner, A.H. et al. “A Basal Dromaeosaurid and Size Evolution Preceding Avian Flight” *Science* 317, 1378 (2007)

# Horses



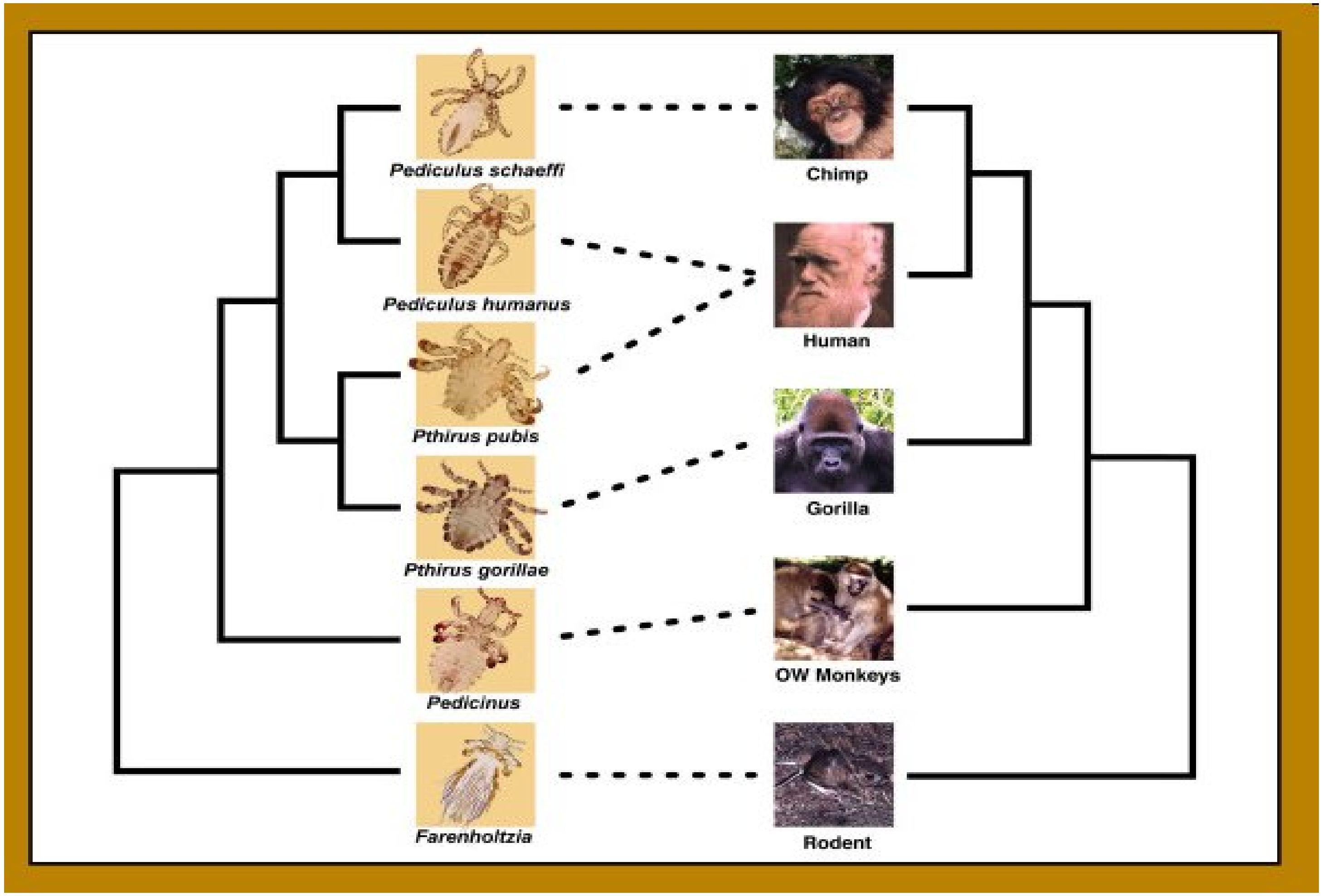
Stirton, R. A. 1940. "Phylogeny of North American Equidae". *Bull. Dept. Geol. Sci.*, Univ. California 25(4): 165-198.

## Retroviruses



Weiss *Retrovirology* 2006 **3**:67 doi:10.1186/1742-4690-3-67

## Two phylogenies



Reed *et al. BMC Biology* 2007 **5**:7 doi:10.1186/1741-7007-5-7

## Evolution

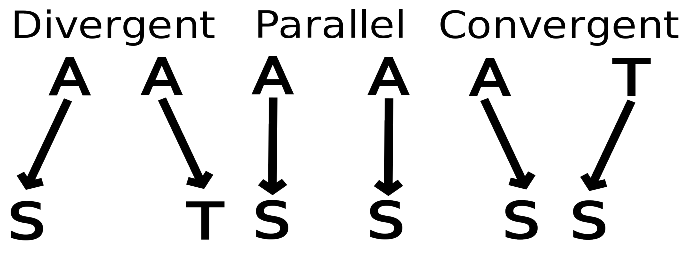
- A phylogeny explains partially the evolutionary history of species
- Associated with a model of evolution
- **Phylogenetic tree**: leaves are associated with today’s species and internal nodes associated with their hypothetical ancestors
- Trees are built from **taxa** associated with species: characters, morphology, DNA sequences, protein sequences, order of genes, repetitions and motifs, etc.

## Phylogenetic Tree

- **Binary tree**
- **Leaves** represent *elements*, called **taxa**, corresponding to species being compared and clustered
- **Internal nodes** are hypothetical ancestral units
- In a **rooted** tree, path from root to a node represents an evolutionary path
- An **unrooted** tree specifies relationships among things, but not evolutionary paths; it represents clusters of species

## Character states

- Character examples: #legs; nucleotide at some location; etc. corresponding states: 2, 4, 8; A, C, G, T; etc.
- Intention: species that share common character states are genetically close
- **Common hypotheses:**
  - **independence**: characters are evolving independently
  - **neither convergence** (parallel evolution leading the same state)



  - **nor reversal** (character turns back to previous state)

# Phylogenetic Tree

- **character-based:** the topology of the tree describes branching events associated with the character (beak shape, #fingers, wings, presence of a given gene/protein/motif, etc.)

| | **2 legs** | **6 legs** | **wings** | **venom** |
|---|---|---|---|---|
| human | 1 | 0 | 0 | 0 |
| bird | 1 | 0 | 1 | 0 |
| bee | 0 | 1 | 1 | 1 |
| spider | 0 | 0 | 0 | 1 |

- **distance-based:** distance between nodes estimates the evolution time between hypothetical or real species (e.g. based on sequence distance)

| | **A** | **B** | **C** | **D** | **E** |
|---|---|---|---|---|---|
| A | 0 | 8 | 8 | 5 | 3 |
| B | | 0 | 3 | 8 | 8 |
| C | | | 0 | 8 | 8 |
| D | | | | 0 | 5 |
| E | | | | | 0 |

## Data for building trees

Trees can be constructed from various types of data

- **character-based**: morphological features (e.g. #legs), DNA/protein sequences
- **distance-based**: measures of distance between species/genes/sequences
- **gene-order**: linear order of orthologous genes in given genomes
- etc.

# Phylogenetic Tree Approaches

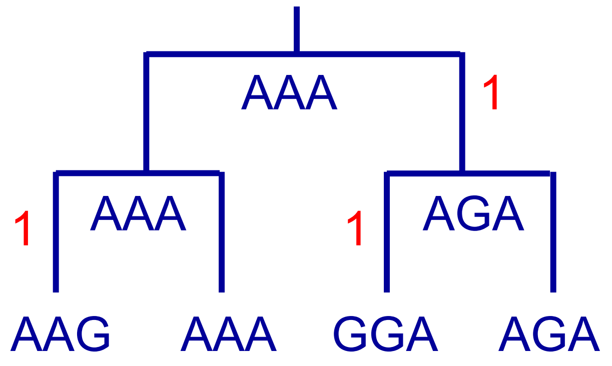
Three general types of methods

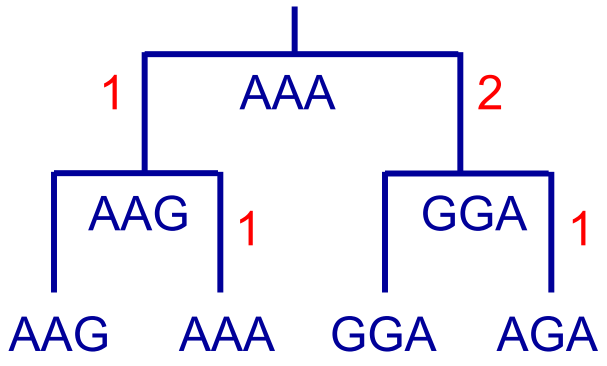
- **parsimony**: find the tree that requires the minimum number of changes (mutations) to explain the data (Fitch's algorithm)
- **distance**: find tree that accounts for estimated evolutionary distances (UPGMA and NJ algorithms)
- **maximum likelihood**: find the tree that maximizes the likelihood of the data (not discussed today)

## Parsimony-based approaches

# Parsimony Based Approaches

- **given**: character-based data, e.g. sequences
- **do**: find tree that explains the data with a minimal number of changes





- Parsimony prefers the first tree because it requires the fewest substitution events

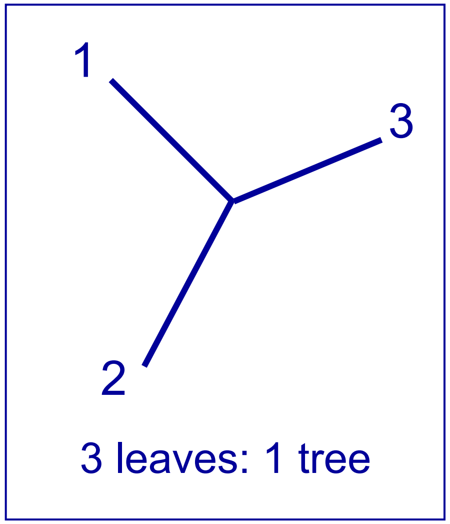
## Parsimony Based Approaches

- Usually involve two separate components
  - a **search** through the space of trees
  - a procedure to find the **minimum number of changes/mutations** needed to explain the data (for a given tree topology)
- Search with a **branch and bound** algorithm based on criteria adapted to the problem
- Explore **branches** of tree representing subsets of solution set
- Branch is checked against upper and lower **bounds**
- Requires an efficient algorithm to label a given tree

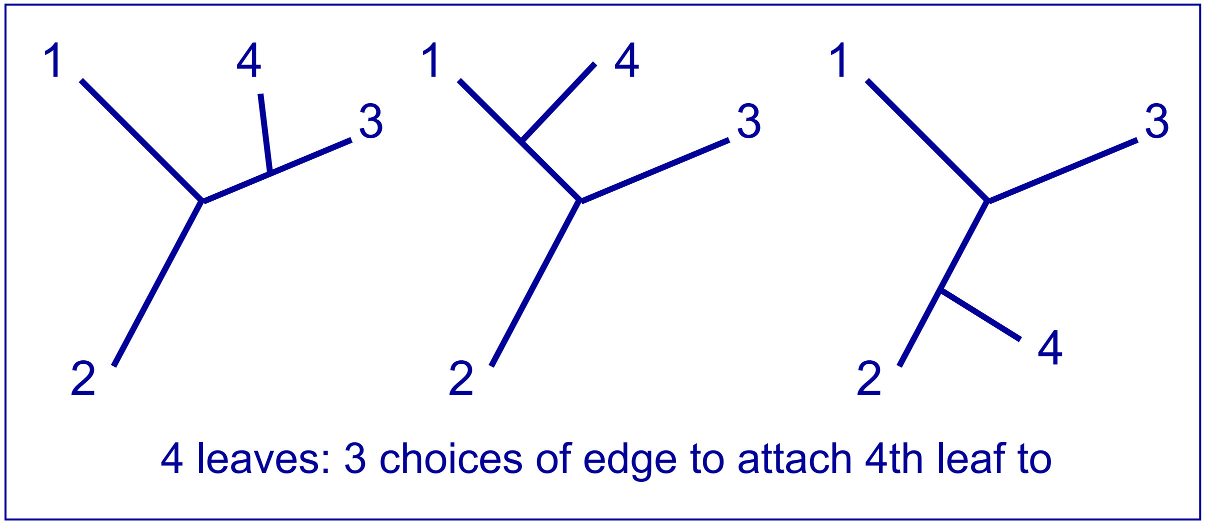
## Number of possible trees

For $\mathrm{n} > 2$ leaves, there are:

- $u_n = 1 \cdot 3 \cdot \ldots \cdot (2n-7) \cdot (2n-5)$ possible unrooted trees



3 leaves: 1 tree



4 leaves: 3 choices of edge to attach 4th leaf to

- $r_n = (2n-3) \cdot u_n$ possible rooted trees:

  place the root on one of the $2n-3$ edges in the unrooted tree

## Number of possible trees

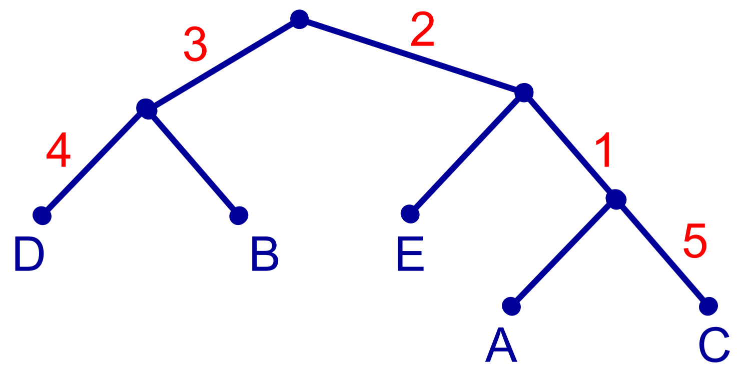
| $n$ | $u_n$ | $r_n$ |
|---|---|---|
| 3 | 1 | 3 |
| 4 | 3 | 15 |
| 5 | 15 | 105 |
| 6 | 105 | 945 |
| 8 | 10 395 | 135 135 |
| 10 | 2 027 025 | 34 459 425 |

Impossible to generate them all and filter the appropriate trees for a given set of species!

# Binary-character tree

- Binary-characters: states are 0 are 1
- ***n*** objects, ***m*** binary-char. matrix ***M***, phylogenetic tree:
  – objects label leaves (1-to-1)
  – each character labels exactly one edge
  – state-1 characters of an object are along the path from the root to the object

$$\begin{array}{c|ccccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} \\ \hline \mathbf{A} & 1 & 1 & 0 & 0 & 0 \\ \mathbf{B} & 0 & 0 & 1 & 0 & 0 \\ \mathbf{C} & 1 & 1 & 0 & 0 & 1 \\ \mathbf{D} & 0 & 0 & 1 & 1 & 0 \\ \mathbf{E} & 0 & 1 & 0 & 0 & 0 \end{array}$$



# Perfect phylogeny

- A **phylogenetic tree is perfect** if, for any taxon f, the characters labeling the edges along the unique path from the root to leaf f specify all characters that f possesses (neither convergence nor reversal hypotheses)
- **problem**: $n$ objects, $m$ binary-characters, is there a perfect phylogeny of the objects?

C and E have the same traits

|   | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| A | 1 | 1 | 0 | 0 | 0 |
| B | 0 | 0 | 1 | 0 | 1 |
| C | 1 | 1 | 0 | 0 | 1 |
| D | 0 | 1 | 1 | 1 | 0 |
| E | 1 | 1 | 0 | 0 | 1 |

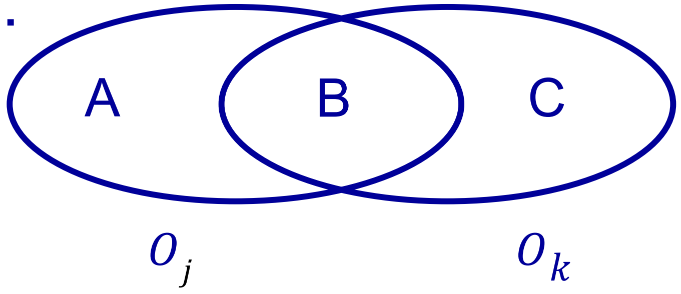
**NO**

|   | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| A | 0 | 0 | 0 | 1 | 1 | 0 |
| B | 1 | 1 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 1 | 1 | 1 |
| D | 1 | 0 | 1 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 1 | 0 | 0 |

**YES**

# Perfect phylogeny

- Let $O_j$ be the set of objects having state 1 for character $j$
- **lemma**: the binary-character matrix $M$ admits a perfect phylogeny if and only if for all $O_j, O_k$: either $O_j \cap O_k = \emptyset$ or $O_j \subseteq O_k$ or $O_k \subseteq O_j$ (either disjoint or comparable)
- Sketch of proof:



| | j | k |
|---|---|---|
| A | 1 | 0 |
| B | 1 | 1 |
| C | 0 | 1 |

impossible because A, B should be in the same subtree without C, and B, C should be in the same subtree without A

## Testing perfect phylogeny

- **Algorithm:**
  – sort columns of $M$ in decreasing order of their number of 1s, which gives $M'$
  – compute new matrix $L$ from $M'$

Find the max k less than j where M_ik is 1

$$\boldsymbol{L_{ij}} = \begin{cases} \max k < j : M'_{ik} = 1 & \text{if } \mathrm{M}'_{\mathrm{ij}} = 1 \\ -1 & \text{if no k} \\ 0 & \text{if } \mathrm{M}'_{\mathrm{ij}} \neq 1 \end{cases}$$

  – if in each column of $L$, the non-zero elements are all the same, then we have a perfect phylogeny

# Running test

$$
\begin{array}{c|cccccc}
\mathbf{M} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \\
\hline
\mathbf{A} & 0 & 0 & 0 & 1 & 1 & 0 \\
\mathbf{B} & 1 & 1 & 0 & 0 & 0 & 0 \\
\mathbf{C} & 0 & 0 & 0 & 1 & 1 & 1 \\
\mathbf{D} & 1 & 0 & 1 & 0 & 0 & 0 \\
\mathbf{E} & 0 & 0 & 0 & 1 & 0 & 0
\end{array}
\xrightarrow[\text{by number of 1s}]{\text{sorting olumns}}
\begin{array}{c|cccccc}
 & 4 & 5 & 1 & 2 & 6 & 3 \\
\mathbf{M'} & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
\mathbf{A} & 1 & 1 & 0 & 0 & 0 & 0 \\
\mathbf{B} & 0 & 0 & 1 & 1 & 0 & 0 \\
\mathbf{C} & 1 & 1 & 0 & 0 & 1 & 0 \\
\mathbf{D} & 0 & 0 & 1 & 0 & 0 & 1 \\
\mathbf{E} & 1 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

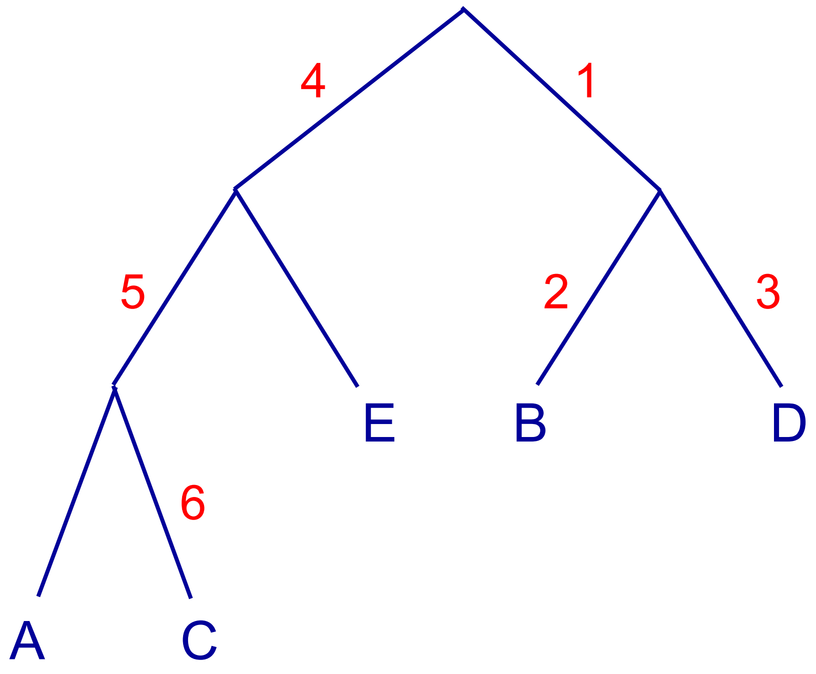
$$
\begin{array}{c|cccccc}
\mathbf{L} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \\
\hline
\mathbf{A} & \mathbf{-1} & \mathbf{1} & 0 & 0 & 0 & 0 \\
\mathbf{B} & 0 & 0 & \mathbf{-1} & \mathbf{3} & 0 & 0 \\
\mathbf{C} & \mathbf{-1} & \mathbf{1} & 0 & 0 & \mathbf{2} & 0 \\
\mathbf{D} & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{3} \\
\mathbf{E} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0
\end{array}
\xrightarrow[\text{numbers in columns}]{\text{same non-zero}}
\text{perfect phylogeny exists}
$$

All -1s, all 1s, all 3s, etc.

## Output perfect phylogeny

$$\begin{array}{c|cccccc}
M & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
A & 0 & 0 & 0 & 1 & 1 & 0 \\
B & 1 & 1 & 0 & 0 & 0 & 0 \\
C & 0 & 0 & 0 & 1 & 1 & 1 \\
D & 1 & 0 & 1 & 0 & 0 & 0 \\
E & 0 & 0 & 0 & 1 & 0 & 0
\end{array}$$

$$\begin{array}{c|cccccc}
\mathbf{L} & \mathbf{4} & \mathbf{5} & \mathbf{1} & \mathbf{2} & \mathbf{6} & \mathbf{3} \\
\hline
\mathbf{A} & \mathbf{-1} & \mathbf{1} & 0 & 0 & 0 & 0 \\
\mathbf{B} & 0 & 0 & \mathbf{-1} & \mathbf{3} & 0 & 0 \\
\mathbf{C} & \mathbf{-1} & \mathbf{1} & 0 & 0 & \mathbf{2} & 0 \\
\mathbf{D} & 0 & 0 & \mathbf{-1} & 0 & 0 & \mathbf{3} \\
\mathbf{E} & \mathbf{-1} & 0 & 0 & 0 & 0 & 0
\end{array}$$



**running time:**
$\boldsymbol{O(mn)}$

## Complexity

- **unordered characters**, all state changes possible

  NP-complete
- **ordered characters** (cladistic model), changes are restricted, linear order, partial order (e.g. A→T→C→G)

  polynomial solutions
- **binary characters** (above algorithm)

  $O(mn)$

## Minimum Changes for a Given Tree

- We are given the tree's topology and a mapping of taxa to the tree's leaves
- Our goal is to label internal nodes, minimizing the total number of changes
- Fitch's algorithm [1971]

  – assumes any state (e.g. nucleotide, amino acid) can convert to any other state

  – assumes positions are independent

  – processes sequences position per position

## Fitch’s Algorithm

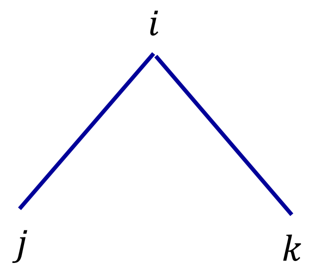
- Two steps (per position on sequences)
  - **bottom-up**: traverse tree from leaves to root determining a set of possible states (e.g. nucleotides) for each internal node
  - **top-down:** traverse tree from root to leaves picking ancestral states for internal nodes

# Fitch's Algorithm: bottom-up

- Set of possible states for internal node $i$: $R_i$
- Compute $R_i$s during a post-order (from leaves to root) traversal of the tree with formula:

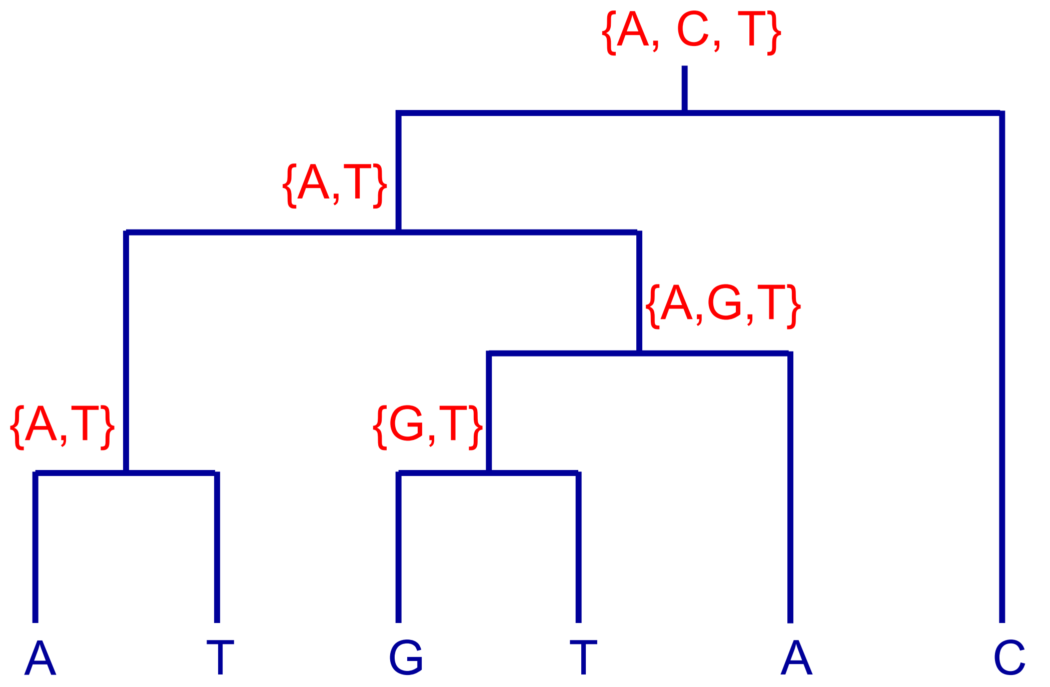
$$R_i = \begin{cases} R_j \cap R_k & R_j \cap R_k \neq \emptyset \\ R_j \cup R_k & otherwise \end{cases}$$

for internal node $i$ with children $j$ and $k$



## Fitch's Algorithm: bottom-up

- Computing sets $R_i$

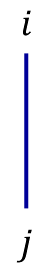


## Fitch's Algorithm: top-down

- Selected states for internal node $i$: $r_i \in R_i$
- Compute $r_i$s during a pre-order (from root to leaves) traversal of the tree with formula:

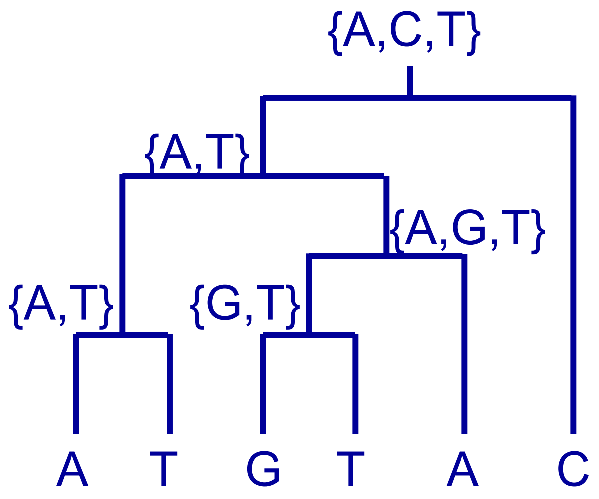
$$r_j = \begin{cases} r_i & if \;\; r_i \in R_j \\ arbitrary\; state\; in\;\; R_j & otherwise \end{cases}$$

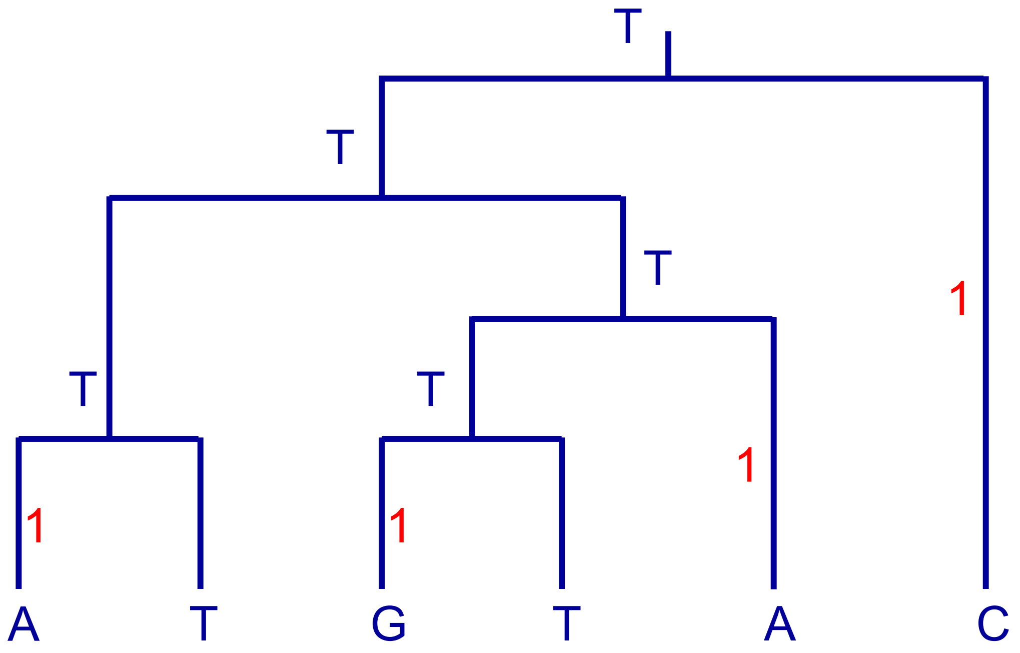
for node $j$ with parent $i$



## Fitch's Algorithm: top-down

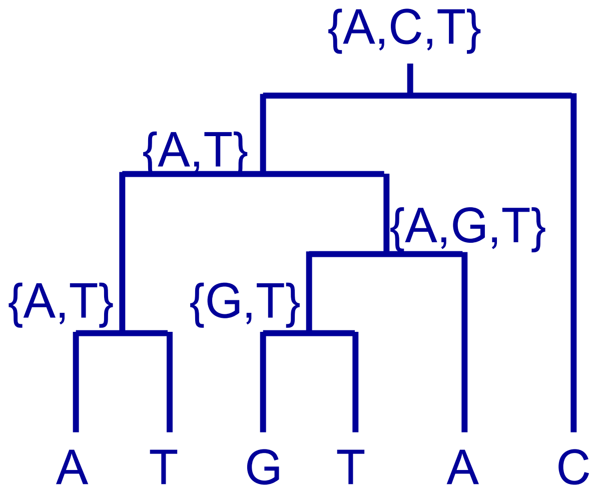
- Selecting states $r_i$ (first choice T): 4 mutations

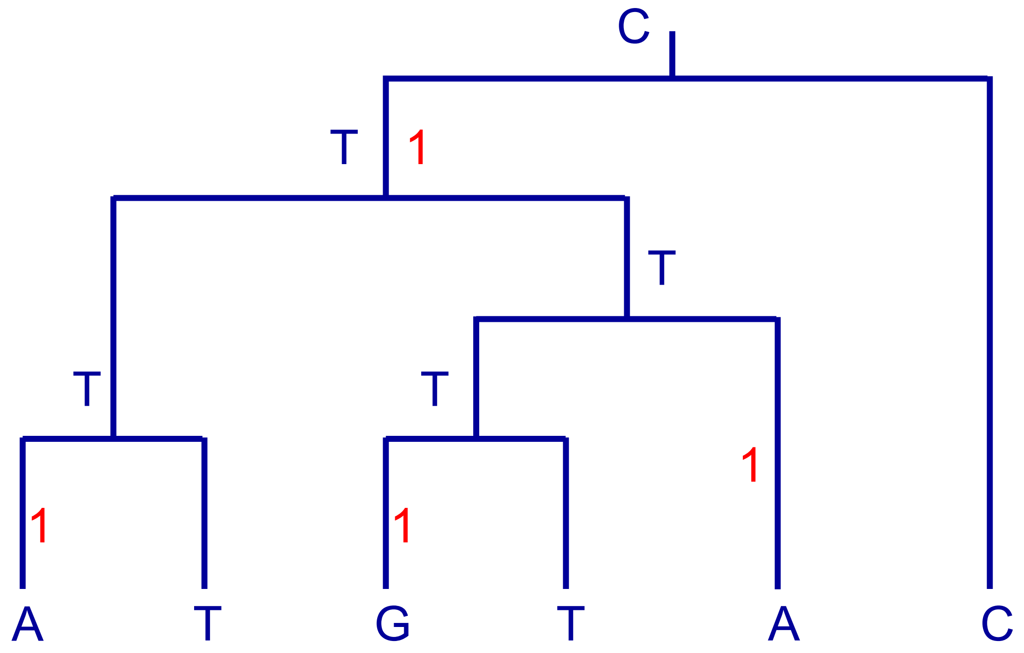




## Fitch's Algorithm: top-down

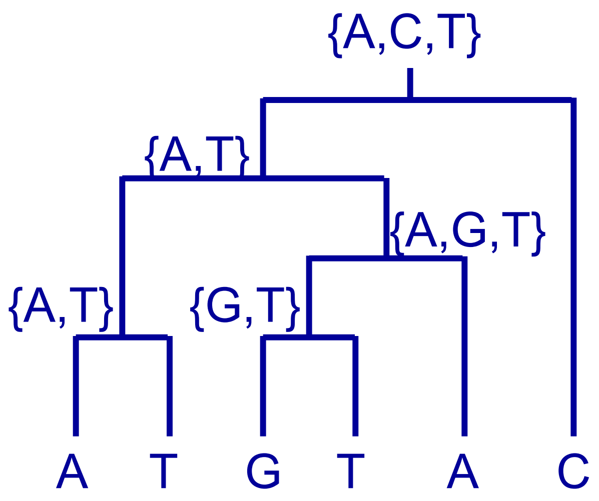
- Selecting states $r_i$ (first choice C, then T): 4 mutations

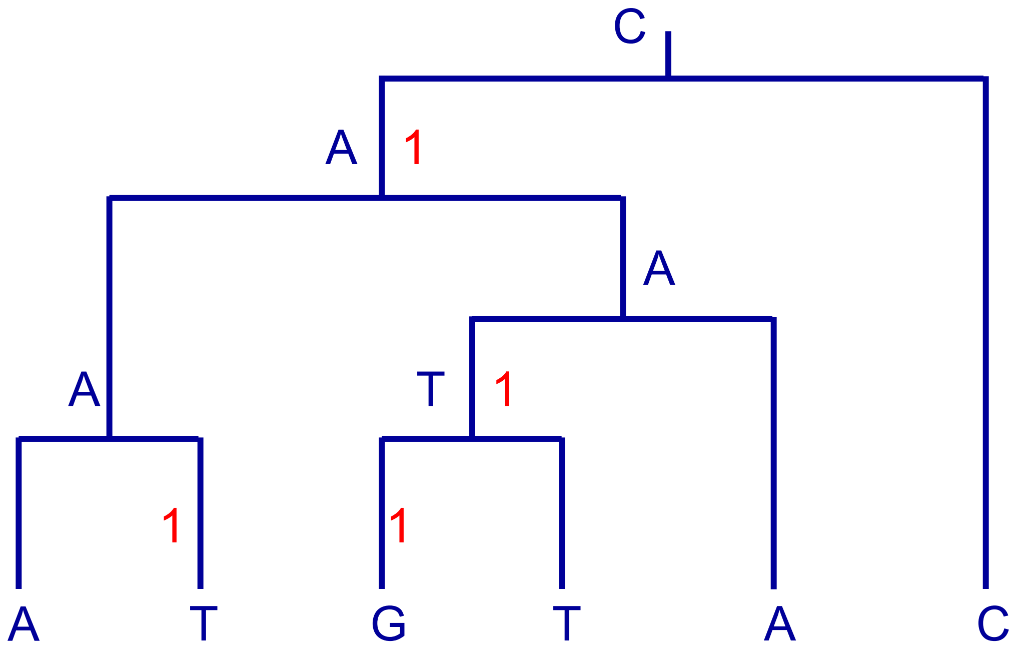




## Fitch's Algorithm: top-down

- Selecting states $r_i$ (first choice C, then A): 4 mutations





## Distance-based approaches

## Distance-based Approaches

- **Input**: an $n \times n$ symmetric matrix $M$ of pairwise distances between the objects
  - $M_{ij}$ gives the distance between $i$ and $j$
- **Output**: an edge-weighted tree such that the distance between leaves $i$ and $j$ is $M_{ij}$
- Two basic methods:
  - **UPGMA** produces a rooted tree under the molecular clock asumption if data is *ultrametric*
  - **NJ** produces an unrooted tree if data is *additive*

## The UPGMA Method

**U**nweighted **P**air **G**roup **M**ethod using Arithmetic **A**verages

- basic idea:
  - iteratively pick two clusters and merge them
  - create a new node in tree for the merged cluster
  - update the distances between clusters and objects
- clusters are groups of objects sharing a common ancestor
- distance between two clusters $C_i$ and $C_j$ defined as:

$$d_{ij} = \frac{\sum_{q \in C_j}^{p \in C_i} d_{pq}}{|C_i| \cdot |C_j|}$$

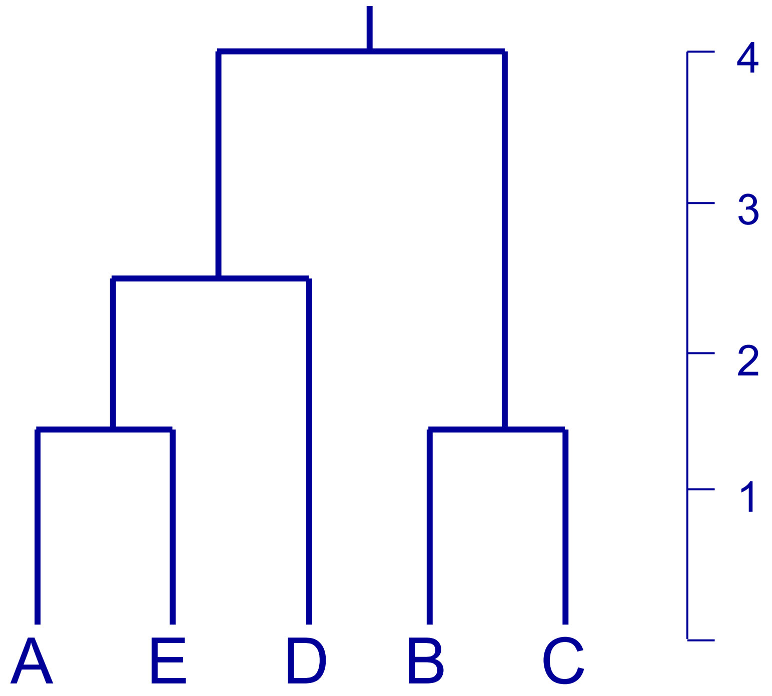
(avg. distance between pairs of elements from clusters)

# The UPGMA Method

**Input**: matrix of distances

|   | **A** | **B** | **C** | **D** | **E** |
|---|---|---|---|---|---|
| A | 0 | 8 | 8 | 5 | 3 |
| B |  | 0 | 3 | 8 | 8 |
| C |  |  | 0 | 8 | 8 |
| D |  |  |  | 0 | 5 |
| E |  |  |  |  | 0 |

**Output**: rooted tree



## UPGMA Algorithm

**Build a phylogenetic tree with UPGMA method**

- define a leaf for each taxon/species
  and place it at height $0$
- make each leaf a cluster
- while there exists more than one cluster
  – find two clusters $C_i$,$C_j$ with smallest $d_{ij}$
- define a new cluster $C_k = C_i \cup C_j$
  – define new node $k$ with children $i$ and $j$ at height $\frac{1}{2} d_{ij}$
  – replace clusters $C_i$ and $C_j$ by $C_k$

## New distance

How do we define the distance of $C_f$ to the **new cluster $\boldsymbol{C_k}$**?

That is, how do we find $d_{fk}$ from $d_{fi}$ and $d_{fj}$ for a given $C_f$?

$$d_{fk} = \frac{\sum_{q \in C_k}^{p \in C_f} d_{pq}}{|C_f| \cdot |C_k|} = \frac{\sum_{q \in C_i}^{p \in C_f} d_{pq} + \sum_{q \in C_j}^{p \in C_f} d_{pq}}{|C_f| \cdot |C_k|}$$

$$= \frac{\dfrac{\sum_{q \in C_i}^{p \in C_f} d_{pq}}{|C_f|} + \dfrac{\sum_{q \in C_j}^{p \in C_f} d_{pq}}{|C_f|}}{|C_k|} = \frac{d_{fi} \cdot |C_i| + d_{fj} \cdot |C_j|}{|C_i| + |C_j|}$$

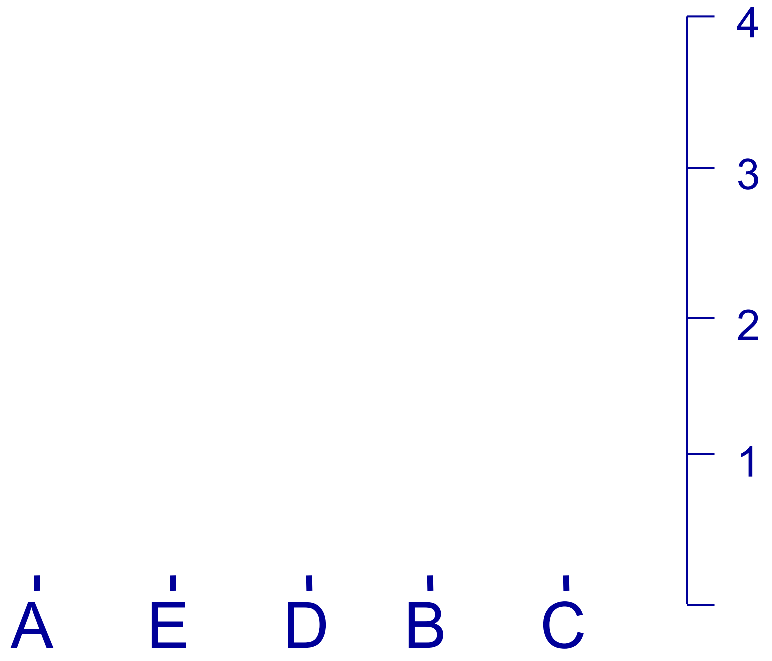
Now it only takes constant time to compute $d_{fk}$.

# Running UPGMA algorithm (1)

Matrix of distances

|   | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 8 | 8 | 5 | 3 |
| B |   | 0 | 3 | 8 | 8 |
| C |   |   | 0 | 8 | 8 |
| D |   |   |   | 0 | 5 |
| E |   |   |   |   | 0 |

Tree



# Running UPGMA algorithm (2)

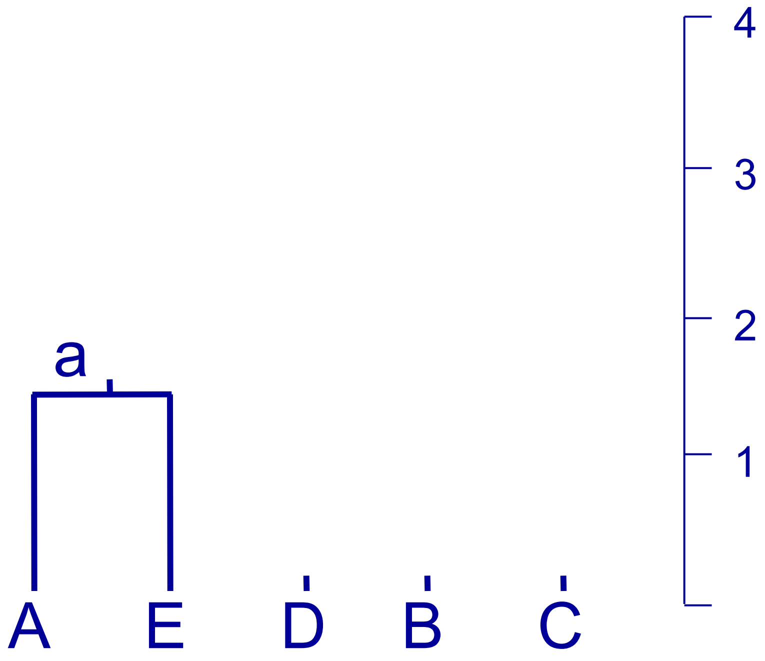
Matrix of distances

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 8 & 8 & 5 & 3 \\ B & & 0 & 3 & 8 & 8 \\ C & & & 0 & 8 & 8 \\ D & & & & 0 & 5 \\ E & & & & & 0 \end{array}$$

$$d_{Da} = \frac{5 \cdot 1 + 5 \cdot 1}{2} = 5$$

Tree

height = 3/2

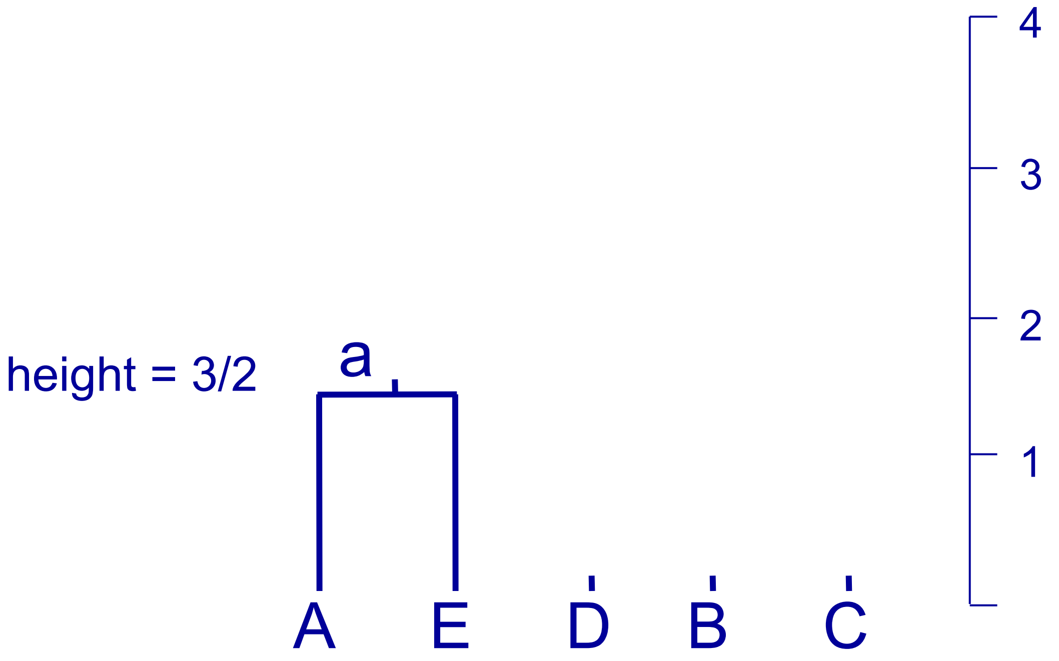


## Running UPGMA algorithm (2)

Matrix of distances

$$\begin{array}{c|cccc} & a & B & C & D \\ \hline a & 0 & 8 & 8 & 5 \\ B & & 0 & 3 & 8 \\ C & & & 0 & 8 \\ D & & & & 0 \end{array}$$

Tree



$$d_{Da} = \frac{5 \cdot 1 + 5 \cdot 1}{2} = 5$$

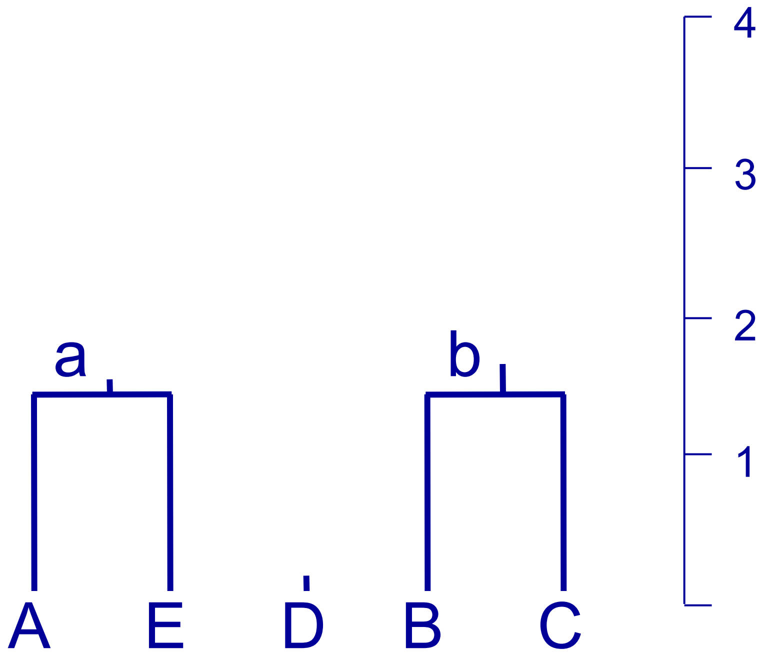
# Running UPGMA algorithm (3)

Matrix of distances

$$\begin{array}{c|cccc} & a & B & C & D \\ \hline a & 0 & 8 & 8 & 5 \\ B & & 0 & 3 & 8 \\ C & & & 0 & 8 \\ D & & & & 0 \end{array}$$

Tree

height = 3/2

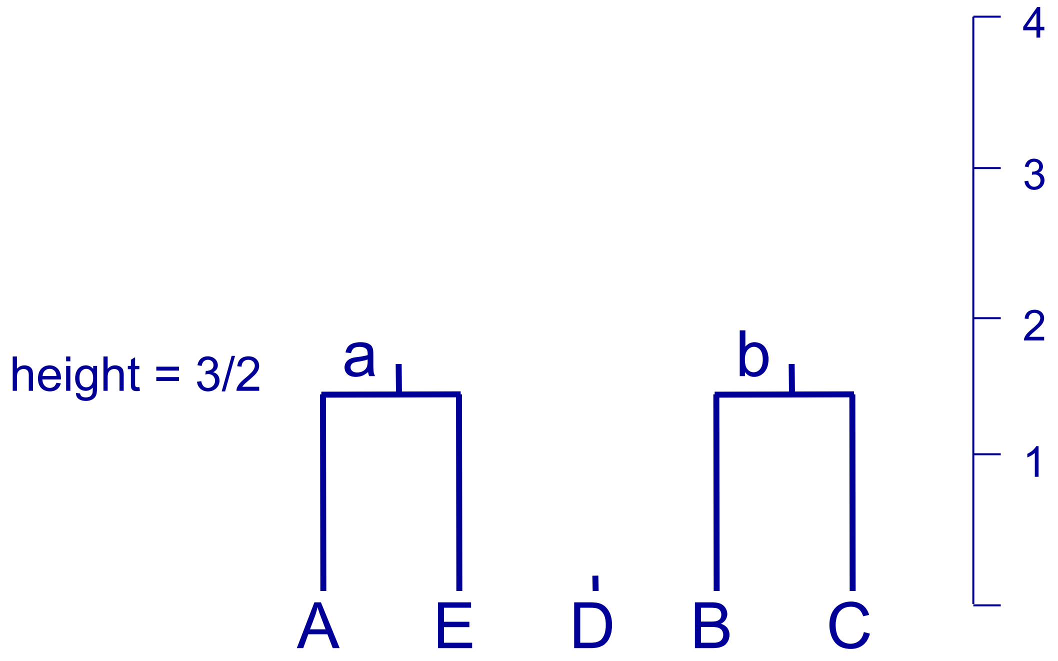


# Running UPGMA algorithm (3)

Matrix of distances

$$\begin{array}{c|ccc} & a & b & D \\ \hline a & 0 & 8 & 5 \\ b & & 0 & 8 \\ D & & & 0 \end{array}$$

Tree



# Running UPGMA algorithm (4)

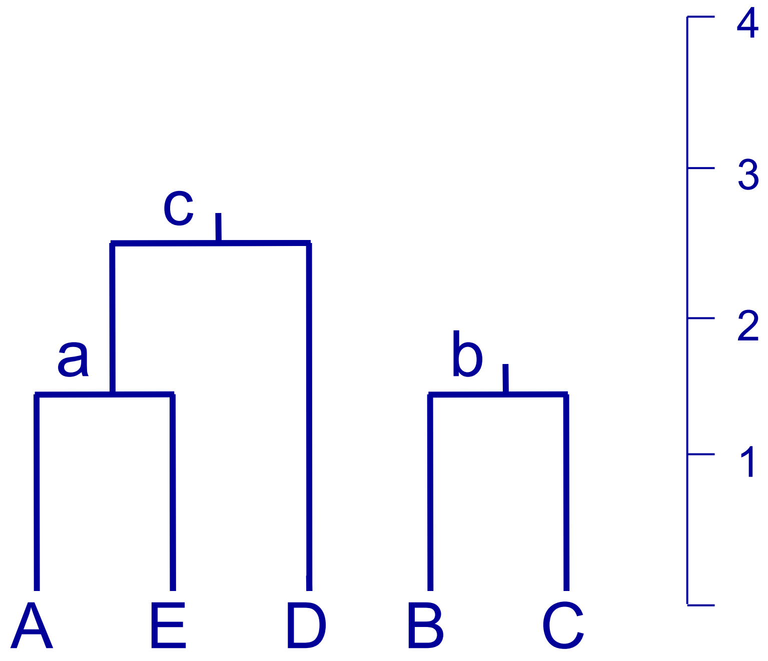
Matrix of distances

$$\begin{array}{c|ccc} & a & b & D \\ \hline a & 0 & 8 & 5 \\ b & & 0 & 8 \\ D & & & 0 \end{array}$$

$$d_{bc} = \frac{8 \cdot 2 + 8 \cdot 1}{3} = 8$$

height = 5/2

Tree



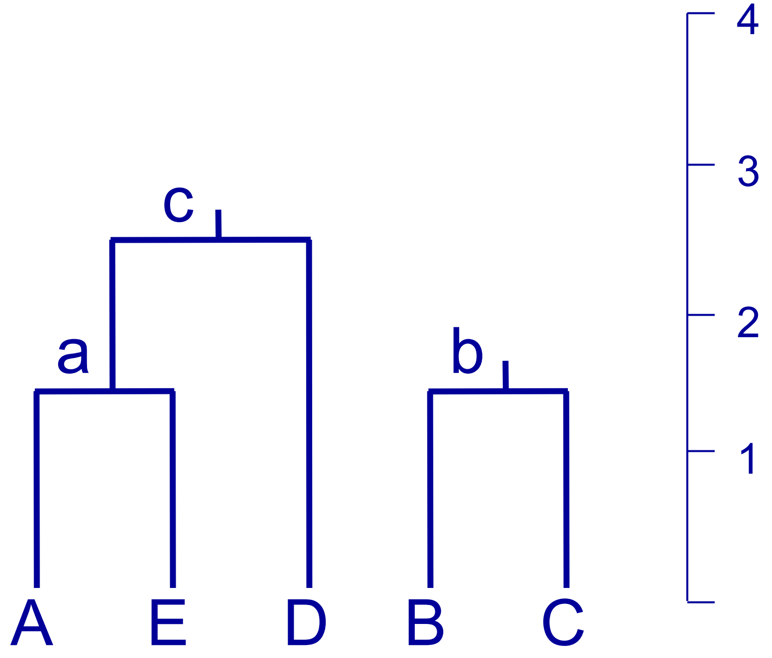
# Running UPGMA algorithm (4)

Matrix of distances

$$\begin{array}{c|cc} & b & c \\ \hline b & 0 & 8 \\ c & & 0 \end{array}$$

Tree

height = 5/2

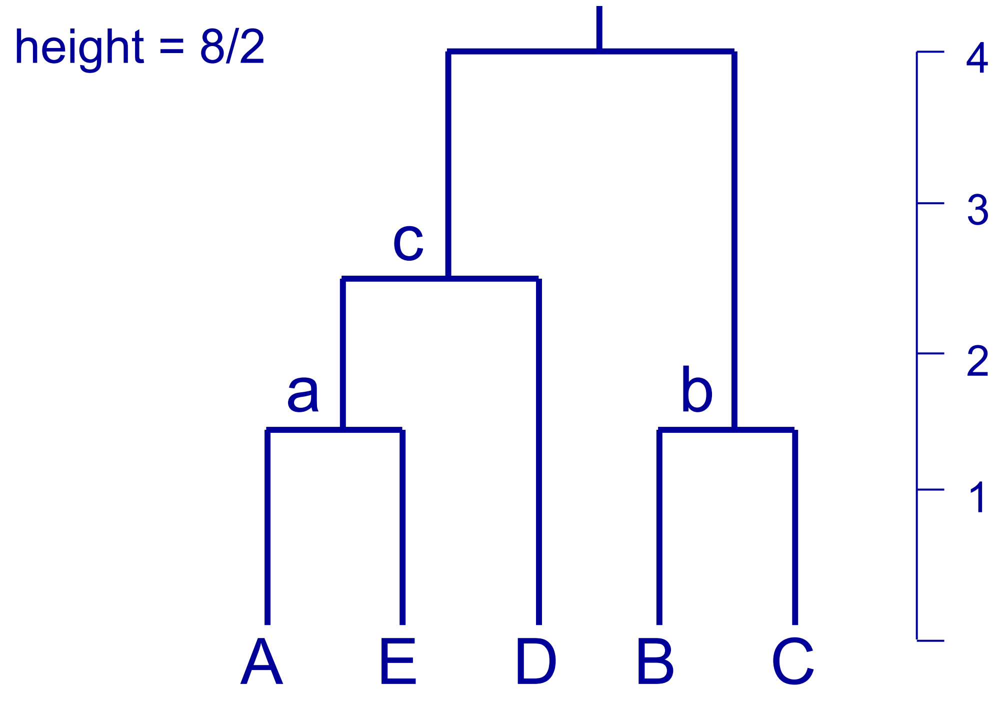


$$d_{bc} = \frac{8 \cdot 2 + 8 \cdot 1}{3} = 8$$

# Running UPGMA algorithm (5)

Last step, joining b and c

Tree

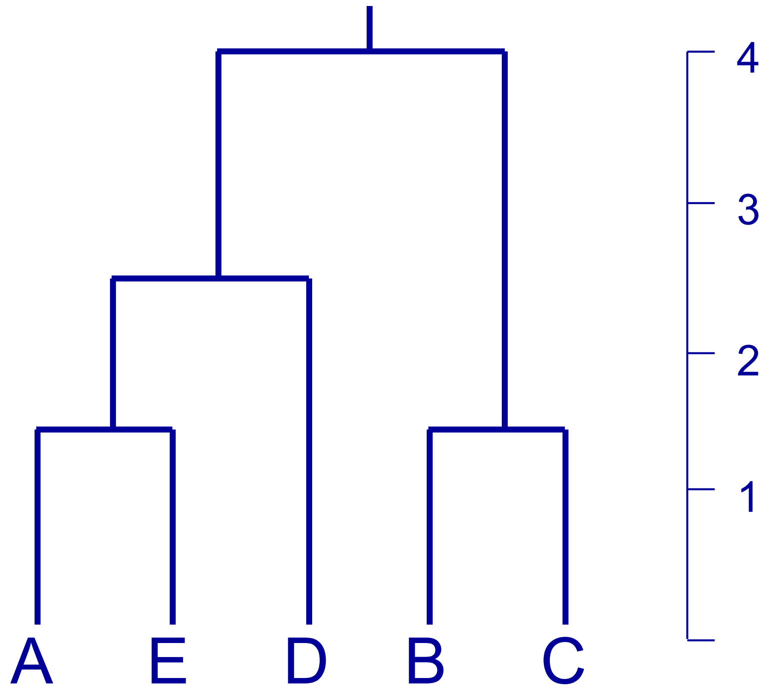


# Running UPGMA algorithm (6)

Matrix of distances

$$\begin{array}{c|ccccc}
 & A & B & C & D & E \\
\hline
A & 0 & 8 & 8 & 5 & 3 \\
B & & 0 & 3 & 8 & 8 \\
C & & & 0 & 8 & 8 \\
D & & & & 0 & 5 \\
E & & & & & 0
\end{array}$$

Tree



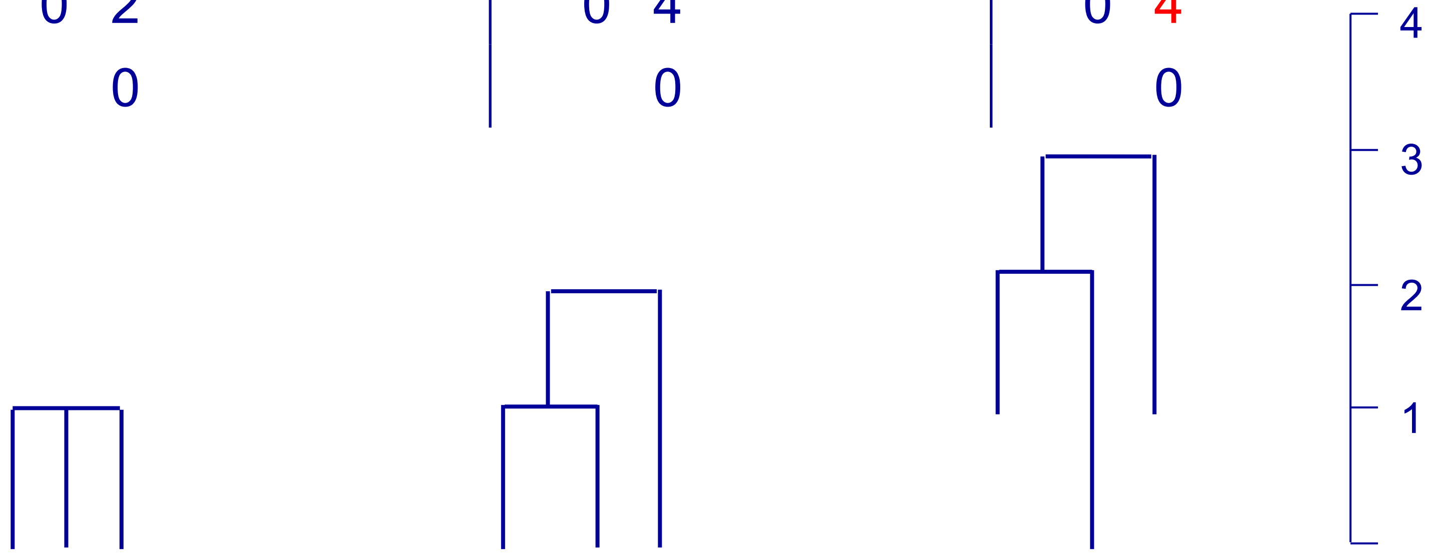
## The Molecular Clock Assumption

- the molecular clock assumption: divergence of sequences/species assumed to occur at the **same rate** at all points in the tree
- generally not true due to: changes in the intensity of natural selection, change in function of the protein studied, species-specific differences, population size, changing generation times, etc.
- if it does hold, then the data is said to be **ultrametric**
- **UPGMA algorithm only works on ultrametric data**

# Ultrametric Data

- Ultrametric data: for every triplet of taxa, $i, j, k$, their distances are either all equal or two are equal and the remaining one is smaller

$$\begin{array}{|ccc} 0 & 2 & 2 \\ & 0 & 2 \\ & & 0 \end{array} \qquad \begin{array}{|ccc} 0 & 2 & 4 \\ & 0 & 4 \\ & & 0 \end{array} \qquad \begin{array}{|ccc} 0 & \color{red}{3} & \color{red}{5} \\ & 0 & \color{red}{4} \\ & & 0 \end{array}$$



## Neighbor Joining

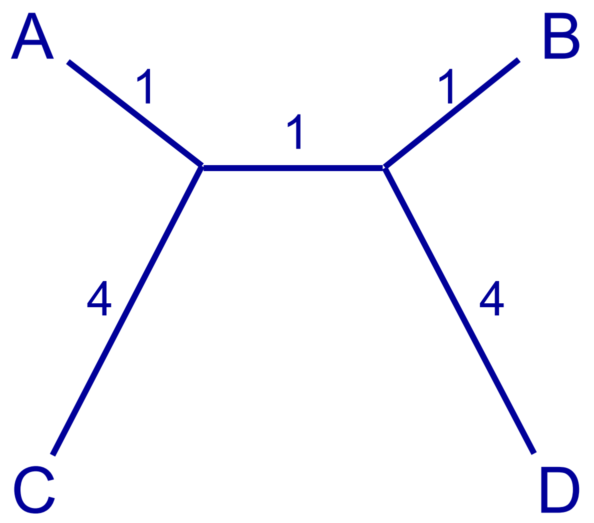
- Like UPGMA, constructs a tree by iteratively joining subtrees
- Unlike UPGMA
  - does not assume the molecular clock assumption
  - produces an unrooted tree
- More general assumption than ultrametricity: **additivity**

## Neighbor Joining

**Input**: matrix of distances

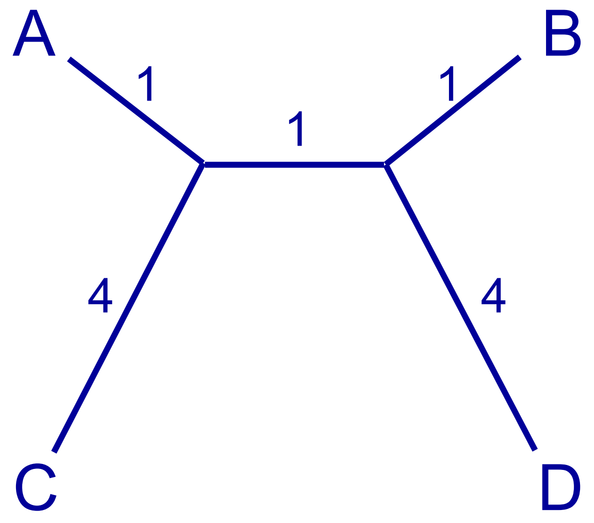
$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 3 & 5 & 6 \\ B & & 0 & 6 & 5 \\ C & & & 0 & 9 \\ D & & & & 0 \end{array}$$

**Output**: unrooted tree



## Picking Pairs of Nodes to Join

- At each step, we pick a pair of nodes to join; should we pick pair with smallest $d_{ij}$? **NO**
- Suppose the real tree looks like this and we're picking the first pair of nodes to join

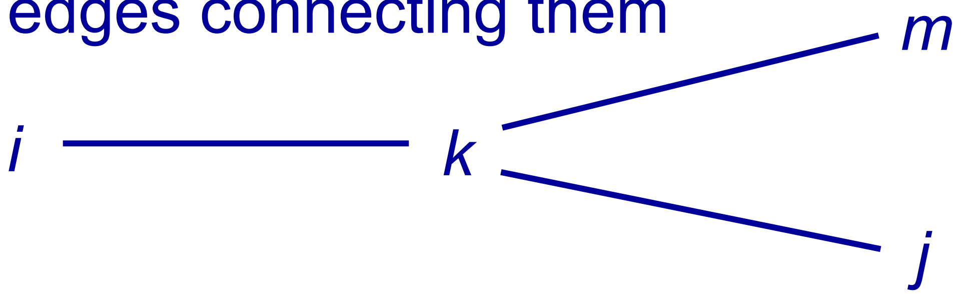


$d_{AB} = 3$ is the smallest distance between two taxa

- Wrong decision: we will never get the tree

# Distances in Neighbor Joining

- Consider that we want to join $i$ and $j$ to a new node $k$
- The distance between any pair of leaves is the sum of lengths of edges connecting them



- We can thus compute the new distance from i to k as follows:

$$\left.\begin{aligned} d_{im} &= d_{ik} + d_{km} \\ d_{jm} &= d_{jk} + d_{km} \\ d_{ij} &= d_{ik} + d_{kj} \end{aligned}\right\} d_{ik} = \frac{1}{2}(d_{ij} + d_{im} - d_{jm})$$

## Distances in Neighbor Joining

- We now generalize the previous identity to account for the distance to all other leaves:

$$w_{ij} = \frac{1}{2}(d_{ij} + r_{ij} - r_{ji})$$

(*$w$* for weight) where

$$r_{ij} = \frac{\sum_{m \in L \setminus \{i,j\}} d_{im}}{|L| - 2}$$

and $L$ is the set of leaves (avg. distance of $i$ to all leaves except $j$)

- Beware: $w_{ij}$ not always equals $w_{ji}$

# NJ Algorithm

**Build a phylogenetic tree with NJ method**

- define one leaf for each taxon/species
- while more than two remaining nodes
  - find ordered pair $i$ and $j$ with smallest weight $w_{ij}$
  - join $i$ and $j$ to a new node $k$ with $d_{ik} = w_{ij}$
  - replace nodes $i$ and $j$ by $k$ and update distances

    with $d_{km} = d_{im} - d_{ik}$ for every node $m \neq i$

## Running NJ algorithm (1)

Find the smallest $w_{ij}$

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 3 & 5 & 6 \\ B & & 0 & 6 & 5 \\ C & & & 0 & 9 \\ D & & & & 0 \end{array}$$

$$w_{AB} = \frac{1}{2}\left(3 + \frac{5+6}{2} - \frac{6+5}{2}\right) = 1.5$$

$$w_{AC} = \frac{1}{2}\left(5 + \frac{3+6}{2} - \frac{6+9}{2}\right) = 1$$

$$w_{AD} = \frac{1}{2}\left(6 + \frac{3+5}{2} - \frac{5+9}{2}\right) = 1.5$$

$$w_{BA} = \frac{1}{2}\left(3 + \frac{6+5}{2} - \frac{5+6}{2}\right) = 1.5$$

$$w_{BC} = \frac{1}{2}\left(6 + \frac{3+5}{2} - \frac{5+9}{2}\right) = 1.5$$

$$\vdots$$

$$w_{BD} = \frac{1}{2}\left(5 + \frac{3+6}{2} - \frac{6+9}{2}\right) = 1$$

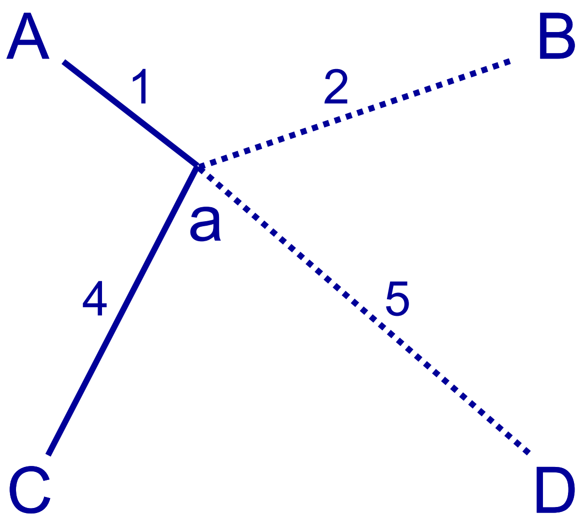
$$\vdots$$

# Running NJ algorithm (2)

Matrix of distances

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 3 & 5 & 6 \\ B & & 0 & 6 & 5 \\ C & & & 0 & 9 \\ D & & & & 0 \end{array}$$

Tree



nodes ***A*** and ***C*** with smallest weight $w_{AC} = 1$

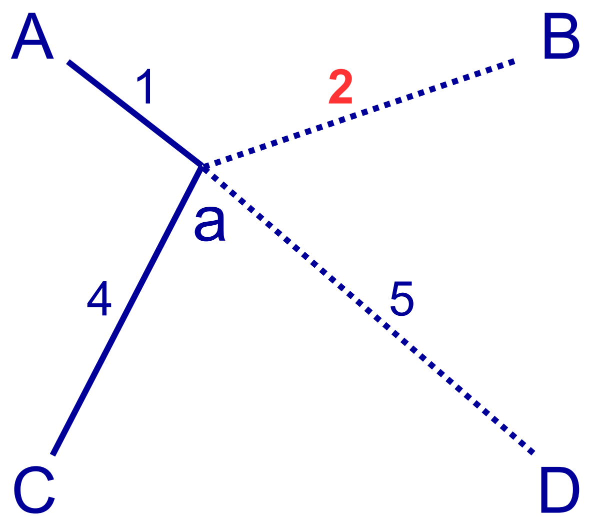
join ***A*** and ***C*** to a new node ***a*** with $d_{Aa} = w_{AC} = 1$

# Running NJ algorithm (2)

Matrix of distances

$$\begin{array}{c|ccc} & a & B & D \\ \hline a & 0 & \mathbf{2} & 5 \\ B & & 0 & 5 \\ D & & & 0 \end{array}$$

Tree



update distance $d_{aB} = d_{AB} - d_{aA} = 3 - 1 = \mathbf{2}$

## Running NJ algorithm (2)

Find the smallest $w_{ij}$

$$\begin{array}{c|ccc} & a & B & D \\ \hline a & 0 & 2 & 5 \\ B & & 0 & 5 \\ D & & & 0 \end{array}$$

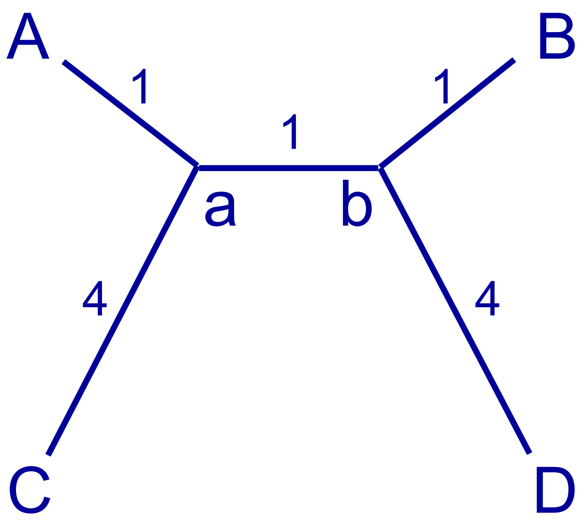
$$w_{aB} = \frac{1}{2}\left(2 + \frac{5}{1} - \frac{5}{1}\right) = 1$$
$$w_{aD} = \frac{1}{2}\left(5 + \frac{2}{1} - \frac{5}{1}\right) = 1$$
$$w_{Ba} = \frac{1}{2}\left(2 + \frac{5}{1} - \frac{5}{1}\right) = 1$$
$$w_{BD} = \frac{1}{2}\left(5 + \frac{2}{1} - \frac{5}{1}\right) = 1$$
$$w_{Da} = \frac{1}{2}\left(5 + \frac{5}{1} - \frac{2}{1}\right) = 4$$
$$w_{DB} = \frac{1}{2}\left(5 + \frac{5}{1} - \frac{2}{1}\right) = 4$$

# Running NJ algorithm (3)

Matrix of distances

$$\begin{array}{c|ccc} & a & B & D \\ \hline a & 0 & 2 & 5 \\ B & & 0 & 5 \\ D & & & 0 \end{array}$$

Tree

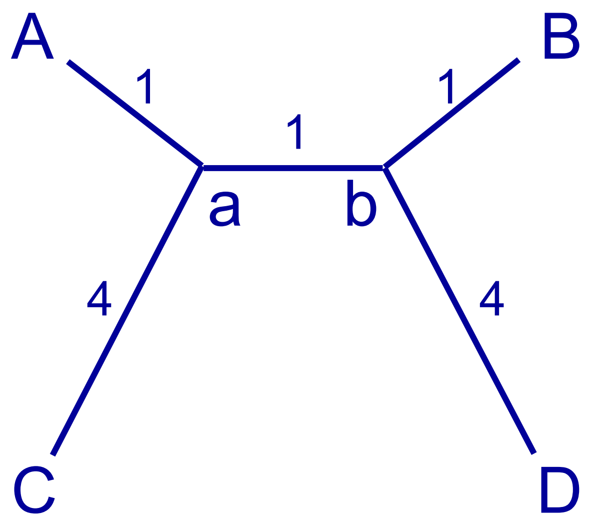


## Running NJ algorithm (3)

Matrix of distances

$$\begin{array}{c|cc} & b & D \\ \hline b & 0 & 4 \\ D & & 0 \end{array}$$

Tree

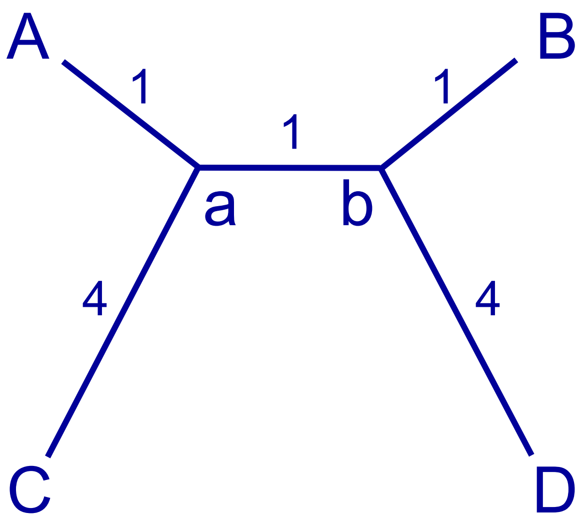


## Running NJ algorithm (4)

Matrix of distances

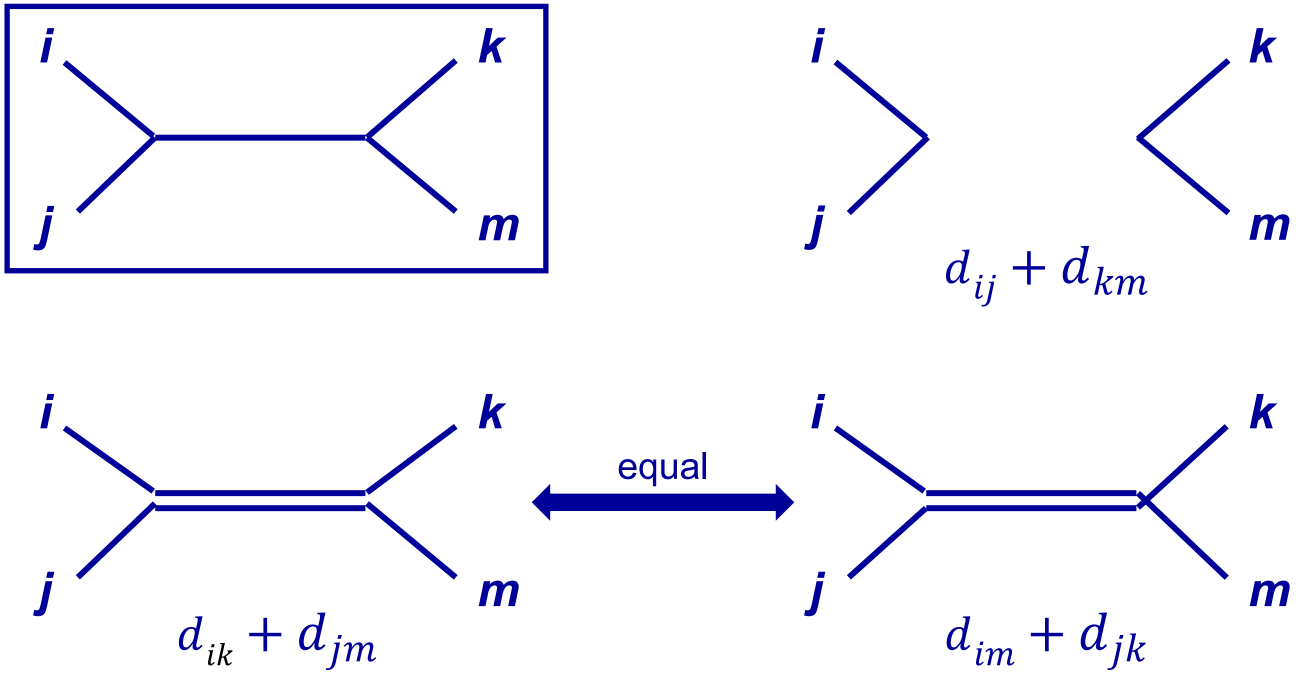
$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 3 & 5 & 6 \\ B & & 0 & 6 & 5 \\ C & & & 0 & 9 \\ D & & & & 0 \end{array}$$

Final tree



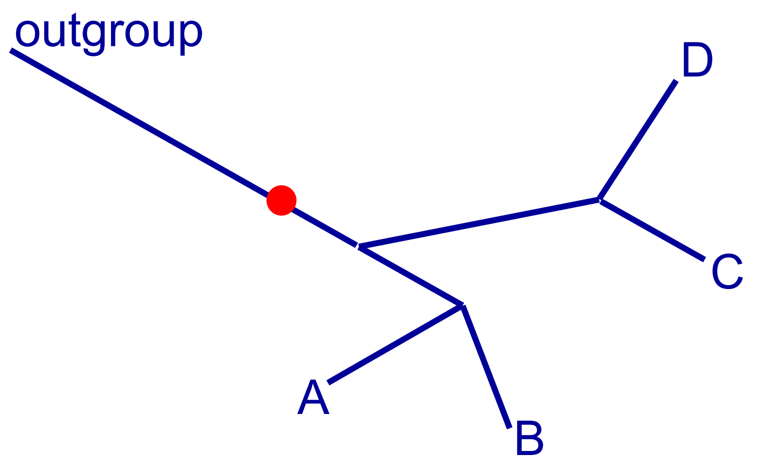
## Testing for Additivity

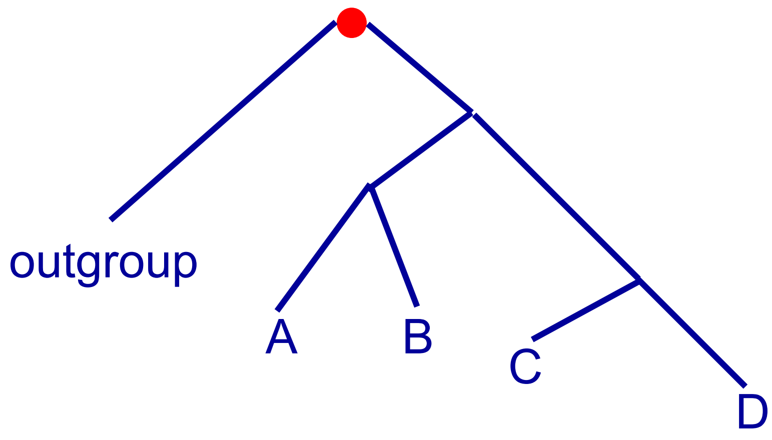
- For every set of four leaves, $i$, $j$, $k$ and $m$, two of the sums $d_{ij} + d_{km}$, $d_{ik} + d_{jm}$, $d_{im} + d_{jk}$ must be equal and larger than the third (triplet condition)



## Rooting Trees

- Finding a root in an unrooted tree is sometimes accomplished by using an **outgroup**
- Outgroup: a species known to be more distantly related to remaining species than they are to each other
- Best candidate for root position on the edge joining the outgroup to the rest of the tree





## Comments on Distance Based Methods (1)

- If the given distance data is **ultrametric** (and these distances represent real distances), then UPGMA will identify the correct tree
- If the data is **additive** (and these distances represent real distances), then neighbor joining will identify the correct tree
- Otherwise, the methods may not recover the correct tree, but they may still be reasonable heuristics
- If the data is ultrametric, it is also additive

## Comments on Distance Based Methods (2)

- Both UPGMA and NJ work by iteratively joining two "nearby" nodes and replacing them with the new node
- UPGMA can be implemented in $O(n^2 \log n)$ time using an efficient priority queue
- NJ can be implemented in $O(n^3)$ time
- NJ is slower because we need to recompute the weights after every iteration, whereas UPGMA allows quick dynamic updates

## Conclusion

  - finding the right tree structure is hard
  - perfect phylogeny: each feature labels exactly one edge
  - Fitch's algorithm minimizes the number of mutations for each character (maximum parsimony) given a tree
- distance-based methods
  - UPGMA requires ultrametric data, assuming the molecular clock assumption
  - NJ requires additive data, but produces an unrooted tree