BIOINFORMATICS AND STATISTICAL GENETICS

GABRIEL VALIENTE

ALGORITHMS, BIOINFORMATICS, COMPLEXITY AND FORMAL METHODS RESEARCH GROUP,
TECHNICAL UNIVERSITY OF CATALONIA

2023–2024

Phylogenetic reconstruction I

- Character-based phylogenetic reconstruction
- Compatibility
- Perfect phylogenies
- Distance-based phylogenetic reconstruction
- Ultrametric trees
- Additive trees

Example

The alcohol dehydrogenase enzyme is one of the most abundant proteins in *Drosophila melanogaster*, and it is encoded by a single gene.

The alcohol dehydrogenase gene was studied on a sample of eleven species from five natural populations of Drosophila melanogaster, and the sampled sequences contain 44 polymorphic (segregating) sites.

```
CCCCAATATGGGCGCTACTTTCACAATAACCCACTAGACAGCCT
CCGCAATATGGGCGCTACCCCCCGGAATCTCCACTAAACAGTCA
CCGCAATATGGGCGCTGTCCCCCGGAATCTCCACTAAACTACCT
CCGAGATAAGTCCGAGGTCCCCCGGAATCTCCACTAGCCAGCCT
CCCCAATATGGGCGCGACCCCCCGGAATCTCTATTCACCAGCTT
CCCCAATATGGGCGCGACCCCCCGGAATCTGTCTCCGCCAGCCT
TGCAGATAAGTCGGCGACCCCCCGGAATCTGTCTCCGCGAGCCT
TGCAGATAAGTCGGCGACCCCCCGGAATCTGTCTCCGCGAGCCT
TGCAGATAAGTCGGCGACCCCCCGGAATCTGTCTCCGCGAGCCT
TGCAGGGGAGGGCTCGACCCCACGGGATCTGTCTCCGCCAGCCT
```

Example

Under the infinite sites assumption, by which mutations are rare enough to discard the possibility of more than one mutation to occur at the same site in a sample of sequences, no site of a sample can contain more than two different nucleotides.

The most frequent nucleotide along a site is often taken as the base, with the least frequent nucleotide being taken as the mutant.

The base and mutant nucleotides for each site of the previous sequences are as follows.

```
CCCCAATAAGGGCGCGACCCCCCCGGAATCTCTATTCGCCAGCCT
TGGAGGGGTTTCGTATGTTTTAACAGTAACGCCCCAAAGTATTA
```

Example

This allows for a binary representation of a sample of sequences, where the base nucleotide in each site is encoded as 0 and the mutant nucleotide is encoded as 1.

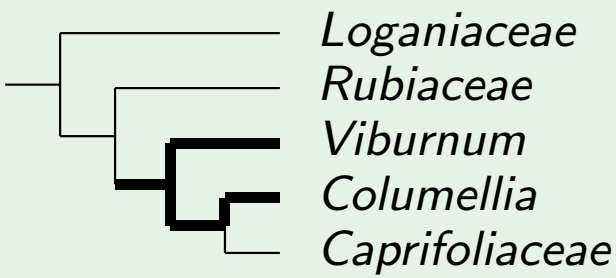
```
000000001000000100111011101111010101010000000
001000001000000100000000000000010101110001010
001000001000000111000000000000010101110110000
001110000011001011000000000000010101001000000
000000001000000000000000000000000000010000010
000000001000000000000000000000010101000000000
110110000011100000000000000000010101000100000
110110000011100000000000000000010101000100000
110110000011100000000000000000010101000100000
110111110000010000000100010000101010000000000
```

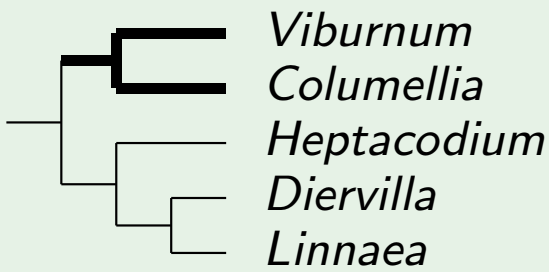
- Compatible trees with overlapping taxa can be combined into a single supertree containing the evolutionary information of the given trees.
- Incompatible trees do not admit their simultaneous inclusion into a common supertree.
- Two or more phylogenetic trees with nested taxa are ancestrally compatible if they can be refined into a common supertree.
- Two or more phylogenetic trees with nested taxa are perfectly compatible if there exists a common supertree whose topological restriction to the taxa in each tree is isomorphic to that tree.

- P. Daniel and C. Semple. Supertree algorithms for nested taxa. In O. R. P. Bininda-Emonds, editor, *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, volume 4 of *Computational Biology*, chapter 7, pages 151–171. Kluwer, 2004
- C. Semple, P. Daniel, W. Hordijk, R. D. M. Page, and M. A. Steel. Supertree algorithms for ancestral divergence dates and nested taxa. *Bioinformatics*, 20(15):2355–2360, 2004

Example

Two compatible phylogenetic trees obtained from TreeBASE.





Incompatible phylogenetic trees can still be partially combined into a maximum agreement subtree.

- M. A. Steel and T. Warnow. Kaikoura tree theorems: Computing the maximum agreement subtree. *Information Processing Letters*, 48(2):77–82, 1993

Compatible phylogenetic trees can be combined into a common supertree.

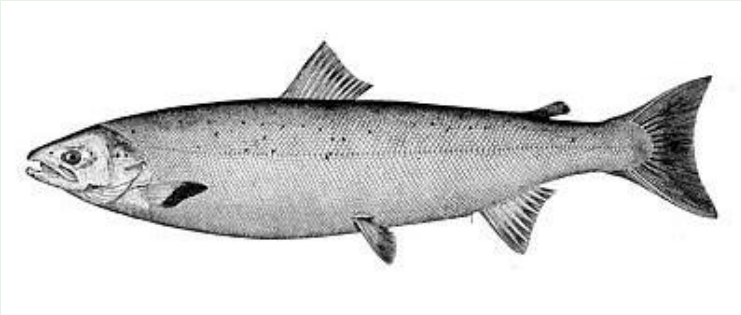
- B. R. Baum. Combining trees as a way of combining datasets for phylogenetic inference, and the desirability of combining gene trees. *Taxon*, 41(1):3–10, 1992
- M. A. Ragan. Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics and Evolution*, 1(1):53–58, 1992
- C. Semple and M. A. Steel. A supertree method for rooted trees. *Discrete Applied Mathematics*, 105(1–3):147–158, 2000
- R. D. M. Page. Modified mincut supertrees. In *Proc. 2nd Int. Workshop Algorithms in Bioinformatics*, volume 2452 of *Lecture Notes in Computer Science*, pages 537–552, Berlin, Heidelberg, 2002. Springer

- Given an $n \times m$ genomic matrix M, find a tree T with n leaves that fits the data

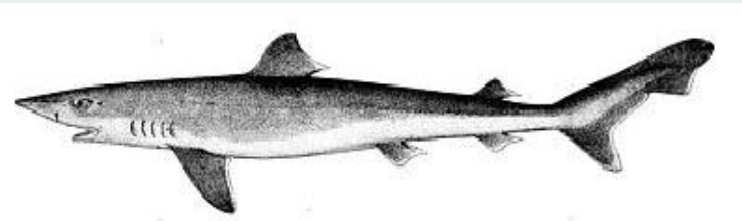
Example

- Lamprey

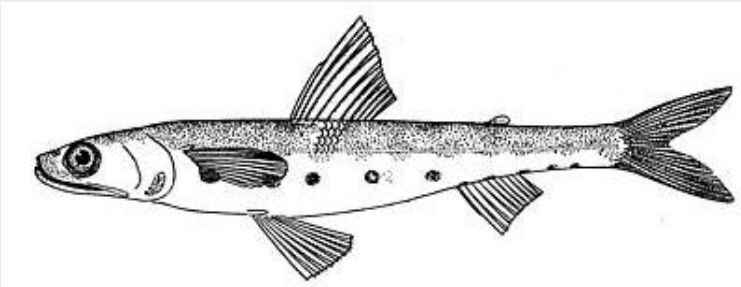
- Salmon



- Shark



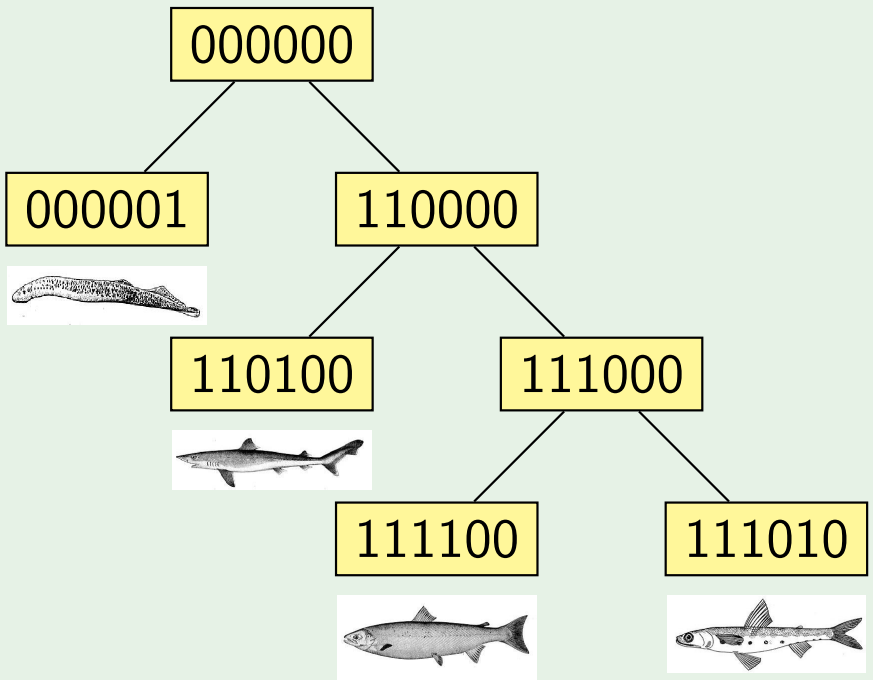
- Lizard



- Given an $n \times m$ genomic matrix M, find a tree T with n leaves that fits the data

Example

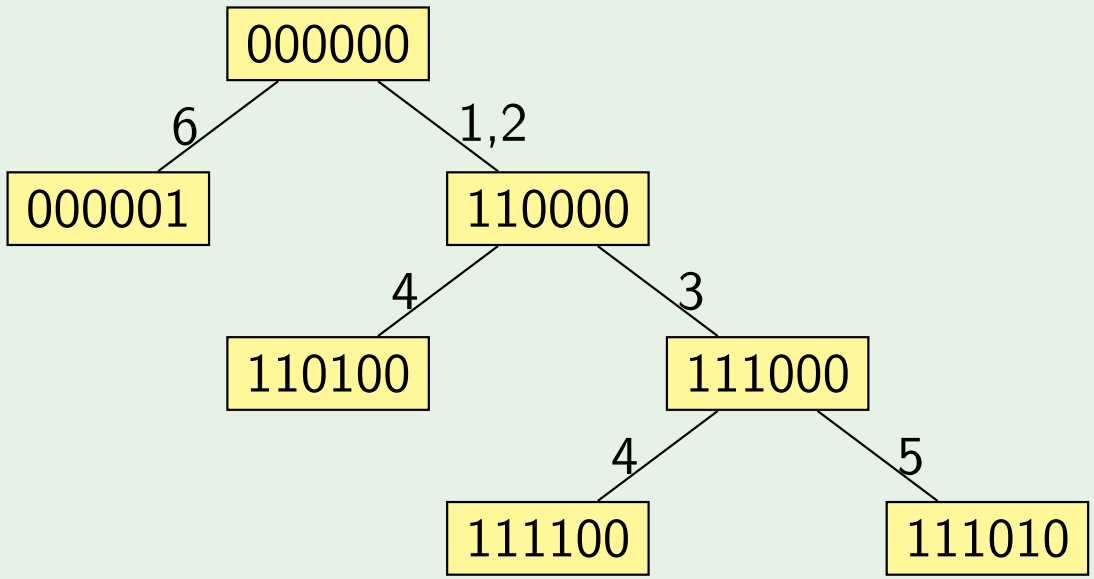
	paired fins	jaws	large dermal bones	fin rays	lungs	rasping tongue
lamprey	0	0	0	0	0	1
shark	1	1	0	1	0	0
salmon	1	1	1	1	0	0
lizard	1	1	1	0	1	0



- Let M be an $n \times m$ genomic matrix
- Biological interpretation (Cladistics)
  - n taxa
  - m cladistic characters
  - two states, 0 (absent) and 1 (present)
  - unordered

Example

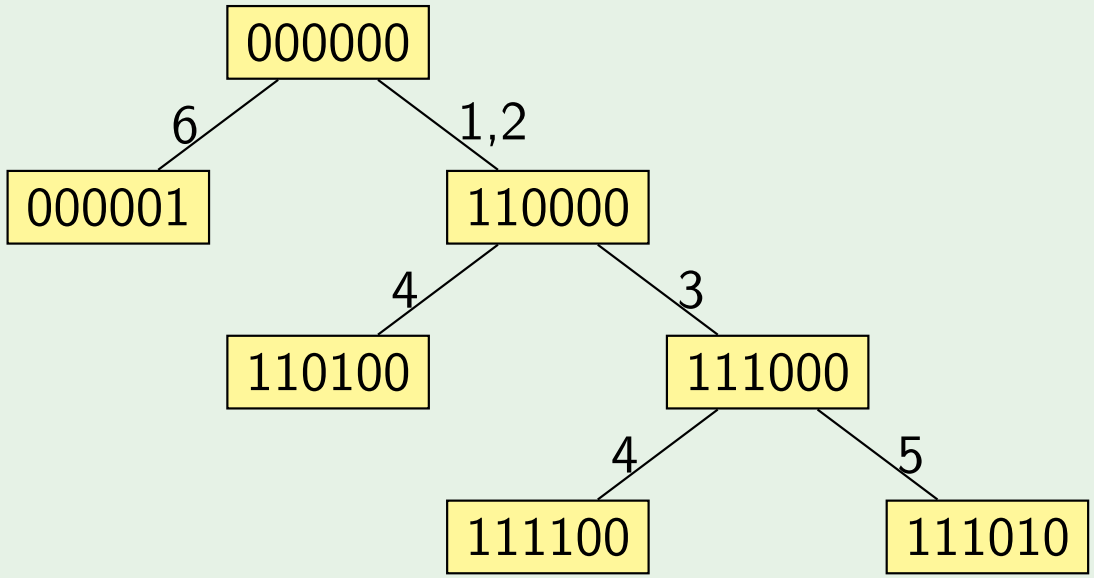
$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$



- Let M be an $n \times m$ genomic matrix
- Biological interpretation (Genomics)
  - n sequences
  - m sites, possibly SNP sites
  - two states, 0 and 1
  - ordered (on the chromosome)

Example

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$



- Given an $n \times m$ genomic matrix M, find a tree T with n leaves that fits the data

  Radix sort M by columns in decreasing order
  for $k = 1$ **to** m **do**
    $O_k = \{i \mid M_{i,k} = 1\}$

- D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, England, 1997

Theorem

M has a phylogenetic tree if and only if for $1 \leqslant i < j \leqslant m$, either $O_i \cap O_j = \emptyset$ or $O_i \subseteq O_j$ or $O_j \subseteq O_i$.

Corollary

It can be determined in $O(nm^2)$ time whether there is a perfect phylogeny for M.

- Given an $n \times m$ genomic matrix M, find a tree T with n leaves that fits the data

  Radix sort M by columns in decreasing order
  Transform M into M' by removing repeated columns
  Let O be the set of entries in M' with value 1
  for each $(i, j) \in O$ **do**
    set $L(i, j)$ to the largest index $k < j$ such that $M'(i, k) \in O$
    set $L(i, j)$ to 0 if there is no such index k
  for each $1 \leqslant j \leqslant m$ **do**
    set $L(j)$ to the largest $L(i, j)$ such that $(i, j) \in O$

- D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21(1):19–28, 1991

Theorem

M has a phylogenetic tree if and only if $L(i, j) = L(j)$ for every $(i, j) \in O$

Corollary

It can be determined in $O(nm)$ time there is a perfect phylogeny for M.

- Given an $n \times m$ genomic matrix M, find a tree T with n leaves that fits the data

  Create a node n_j for each column j of M'
  for each node n_j with $L(j) > 0$ **do**
    Make node $n_{L(j)}$ the parent of node n_j
    Label the edge with j and the indexes of all columns identical to j
  Create a root node r
  for each node n_j with $L(j) = 0$ **do**
    Make node r the parent of node n_j
    Label the edge with j and the indexes of all columns identical to j

- D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21(1):19–28, 1991

- Given an $n \times m$ genomic matrix M, find a tree T with n leaves that fits the data

for each $1 \leqslant i \leqslant n$ **do**
  Let c_i be the largest index such that $M'[i, c_i] = 1$
  Let (n_j, n_k) be the edge labeled with c_i
  if node n_k is a leaf **then**
    Label node n_k with i
  else
    Create a leaf node n_ℓ
    Make node n_k the parent of node n_ℓ
    Label node n_ℓ with i

- D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21(1):19–28, 1991

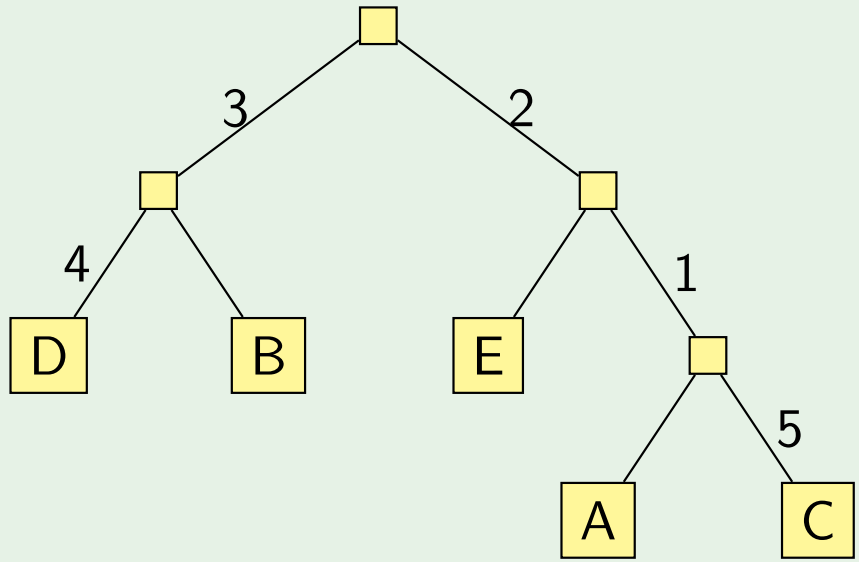
Theorem

The resulting tree T is a phylogenetic tree for the genomic matrix M

- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

Example

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$



- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

Example (Test M for a phylogenetic tree)

$$M' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad L = \begin{pmatrix} 0 & & & 1 & \\ & 0 & & & \\ 0 & & & 1 & 4 \\ & 0 & 2 & & \\ 0 & & & & \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 3 & 4 & 1 & 5 \end{pmatrix} \qquad L = \begin{pmatrix} 0 & 0 & 2 & 1 & 4 \end{pmatrix}$$

- M has a phylogenetic tree, because $L(i,j) = L(j)$ for every $(i,j) \in O$

- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

Example (Test M for a phylogenetic tree)

$$M' = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad L = \begin{pmatrix} & 0 & & & 2 \\ 0 & & 1 & & \\ 0 & 1 & & & 2 \\ & & 0 & 3 & \\ 0 & 1 & & & \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 5 & 2 & 3 & 4 & 1 \end{pmatrix} \qquad L = \begin{pmatrix} 0 & 1 & 1 & 3 & 2 \end{pmatrix}$$

- M has no phylogenetic tree, because $L(1,2) \neq L(2)$, and also because $L(4,3) \neq L(3)$

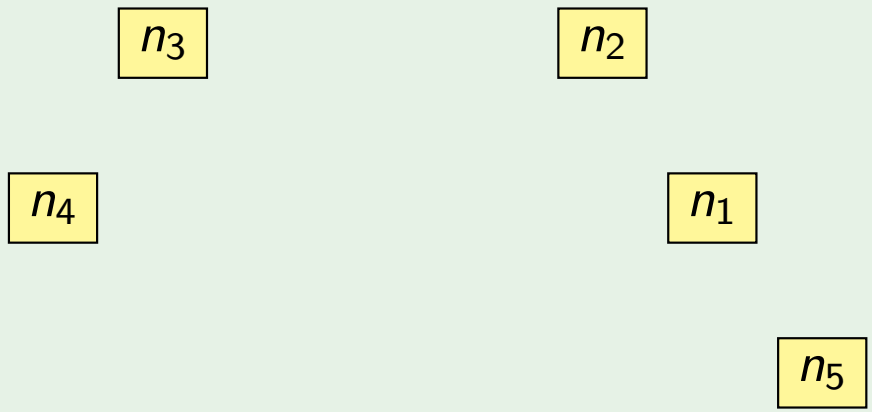
- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

Example (Build a phylogenetic tree T for M)

$$M' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & 2 & 1 & 4 \end{pmatrix}$$



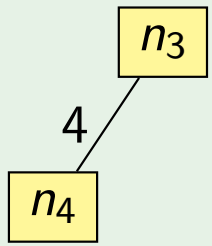
- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

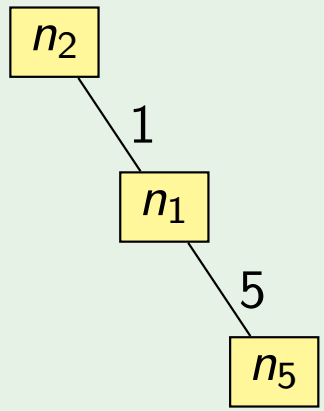
Example (Build a phylogenetic tree T for M)

$$M' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & 2 & 1 & 4 \end{pmatrix}$$





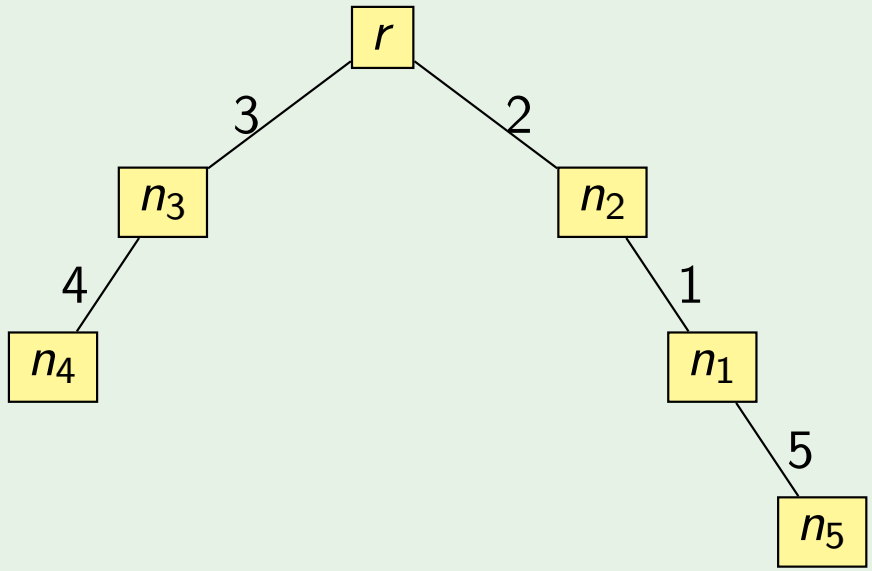
- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

Example (Build a phylogenetic tree T for M)

$$M' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & 2 & 1 & 4 \end{pmatrix}$$



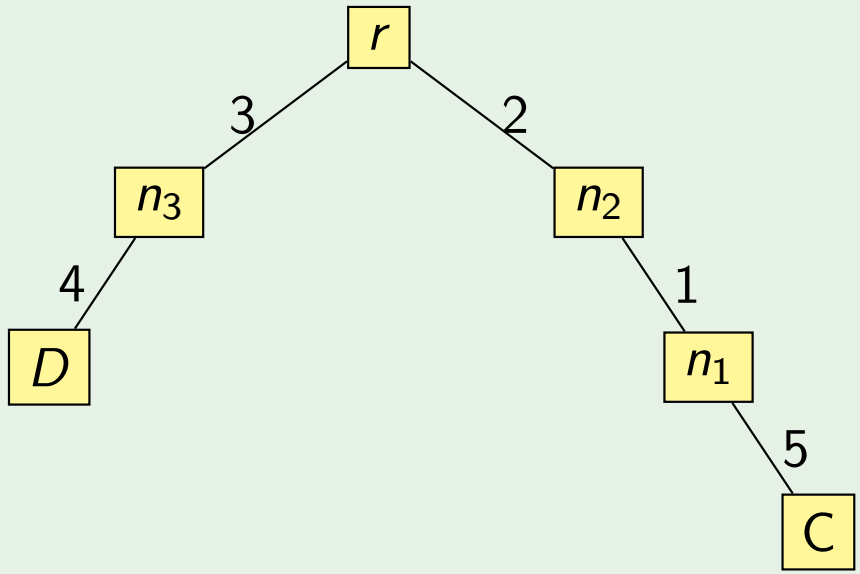
- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

Example (Build a phylogenetic tree T for M)

$$M' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & 2 & 1 & 4 \end{pmatrix}$$



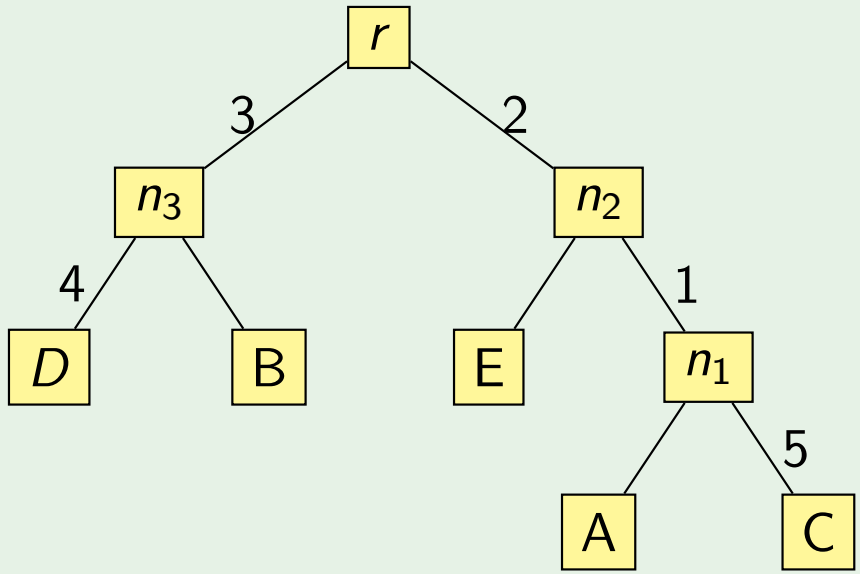
- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

Example (Build a phylogenetic tree T for M)

$$M' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & 2 & 1 & 4 \end{pmatrix}$$



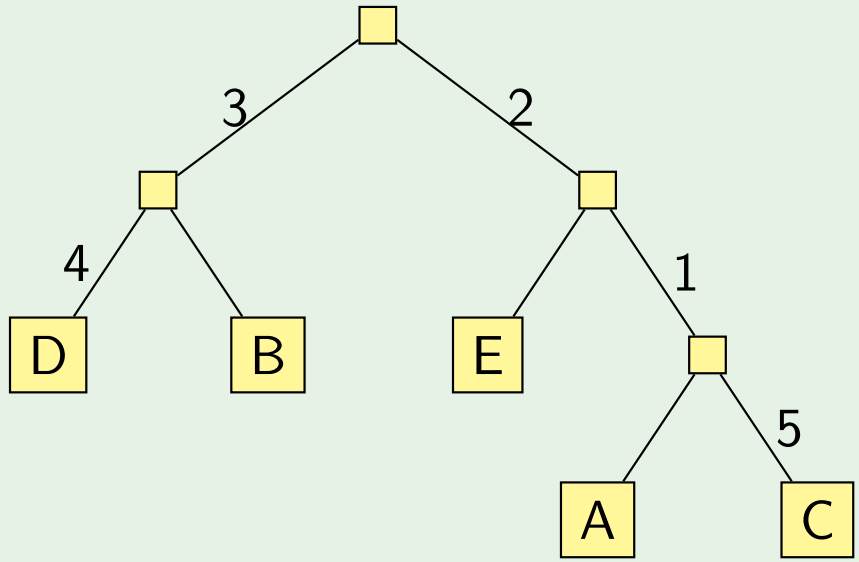
- Given an $n \times m$ genomic matrix M, the unique tree T with n leaves that fits the data, if it exists, can be reconstructed in $O(nm)$ time using the previous algorithm

Example (Build a phylogenetic tree T for M)

$$M' = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\Pi = \begin{pmatrix} 2 & 3 & 4 & 1 & 5 \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & 2 & 1 & 4 \end{pmatrix}$$



- Given an $n \times n$ distance matrix D, find a tree T with n leaves that fits the data, that is, such that $d_{i,j}^T = D_{i,j}$ for every two leaves i and j

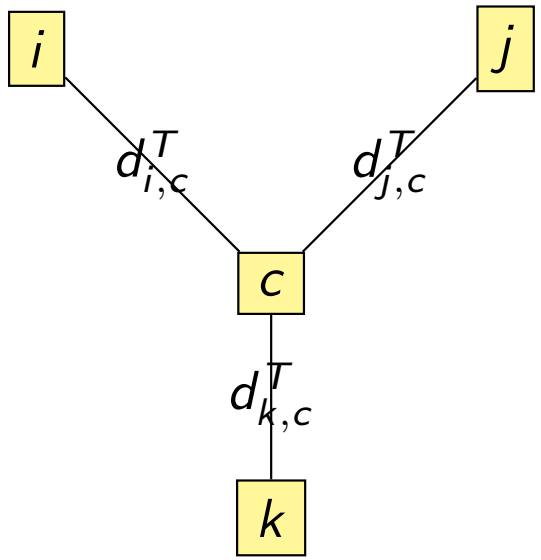
$$\begin{pmatrix} 0 & 37 & 55 & 69 & 68 & 25 \\ 37 & 0 & 42 & 56 & 55 & 38 \\ 55 & 42 & 0 & 46 & 45 & 56 \\ 69 & 56 & 46 & 0 & 25 & 70 \\ 68 & 55 & 45 & 25 & 0 & 69 \\ 25 & 38 & 56 & 70 & 69 & 0 \end{pmatrix}$$

  - A matrix D if symmetric non-negative if $D_{i,j} = D_{j,i}$ and $D_{i,j} \geqslant 0$ for all i and j
  - A matrix D satisfies the triangle inequality if $D_{i,j} + D_{j,k} \geqslant D_{i,k}$ for all i, j, and k
  - A matrix D is a distance matrix if it is symmetric non-negative, it satisfies the triangle inequality, and $D_{i,j} \neq 0$ for all $i \neq j$

- There are many ways in which distance matrices can be generated
  - Sequence a particular gene in n species and define $D_{i,j}$ as the edit distance between this gene in species i and species j
  - Sequence a particular gene in n species and define $D_{i,j}$ as the alignment distance between this gene in species i and species j

- Given an $n \times n$ distance matrix D, find a tree T with n leaves that fits the data, that is, such that $d^T_{i,j} = D_{i,j}$ for every two leaves i and j

– There is only one unrooted binary tree topology T with $n = 3$ leaves



– The lengths of each edge in T are defined by three equations with three variables

$$\begin{cases} d^T_{i,c} + d^T_{j,c} = D_{i,j} \\ d^T_{i,c} + d^T_{k,c} = D_{i,k} \\ d^T_{j,c} + d^T_{k,c} = D_{j,k} \end{cases} \qquad \begin{aligned} d^T_{i,c} &= (D_{i,j} + D_{i,k} - D_{j,k})/2 \\ d^T_{j,c} &= (D_{i,j} + D_{j,k} - D_{i,k})/2 \\ d^T_{k,c} &= (D_{i,k} + D_{j,k} - D_{i,j})/2 \end{aligned}$$

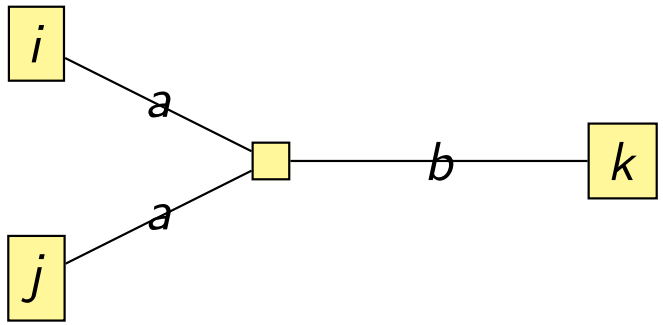
- Given an $n \times n$ distance matrix D, find a tree T with n leaves that fits the data, that is, such that $d_{i,j}^T = D_{i,j}$ for every two leaves i and j
  - An unrooted binary tree with n leaves has $2n - 3$ edges
  - Fitting any given tree T with n leaves to an $n \times n$ distance matrix D involves solving a system of $\binom{n}{2}$ equations with $2n - 3$ variables
  - For $n = 4$, this amounts to solving a system of six equations with only five variables, and it is not always possible to solve this system, making it hard or impossible to construct such a tree T from D

- A distance matrix D is ultrametric if for every three leaves i, j, and k, of the three distances

$$D_{i,j} \qquad D_{i,k} \qquad D_{j,k}$$

the two largest are equal (three point condition)

- $D_{i,j} \leqslant D_{i,k} = D_{j,k}$
- $D_{i,k} \leqslant D_{i,j} = D_{j,k}$
- $D_{j,k} \leqslant D_{i,j} = D_{i,k}$

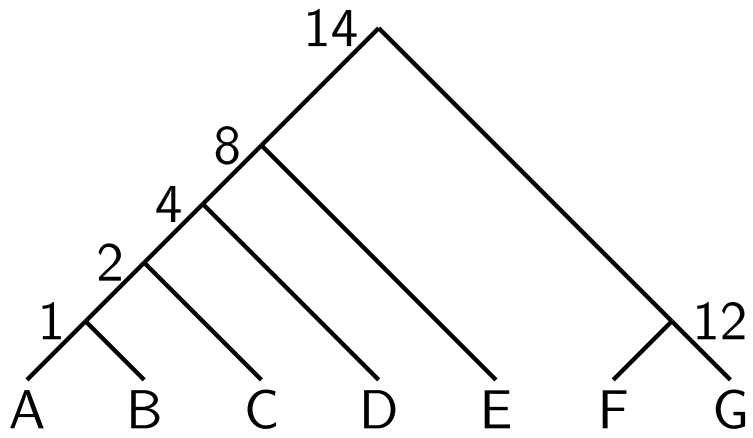


- $D_{i,j} = 2a \leqslant a + b = D_{i,k} = D_{j,k}$ implies $a \leqslant b$

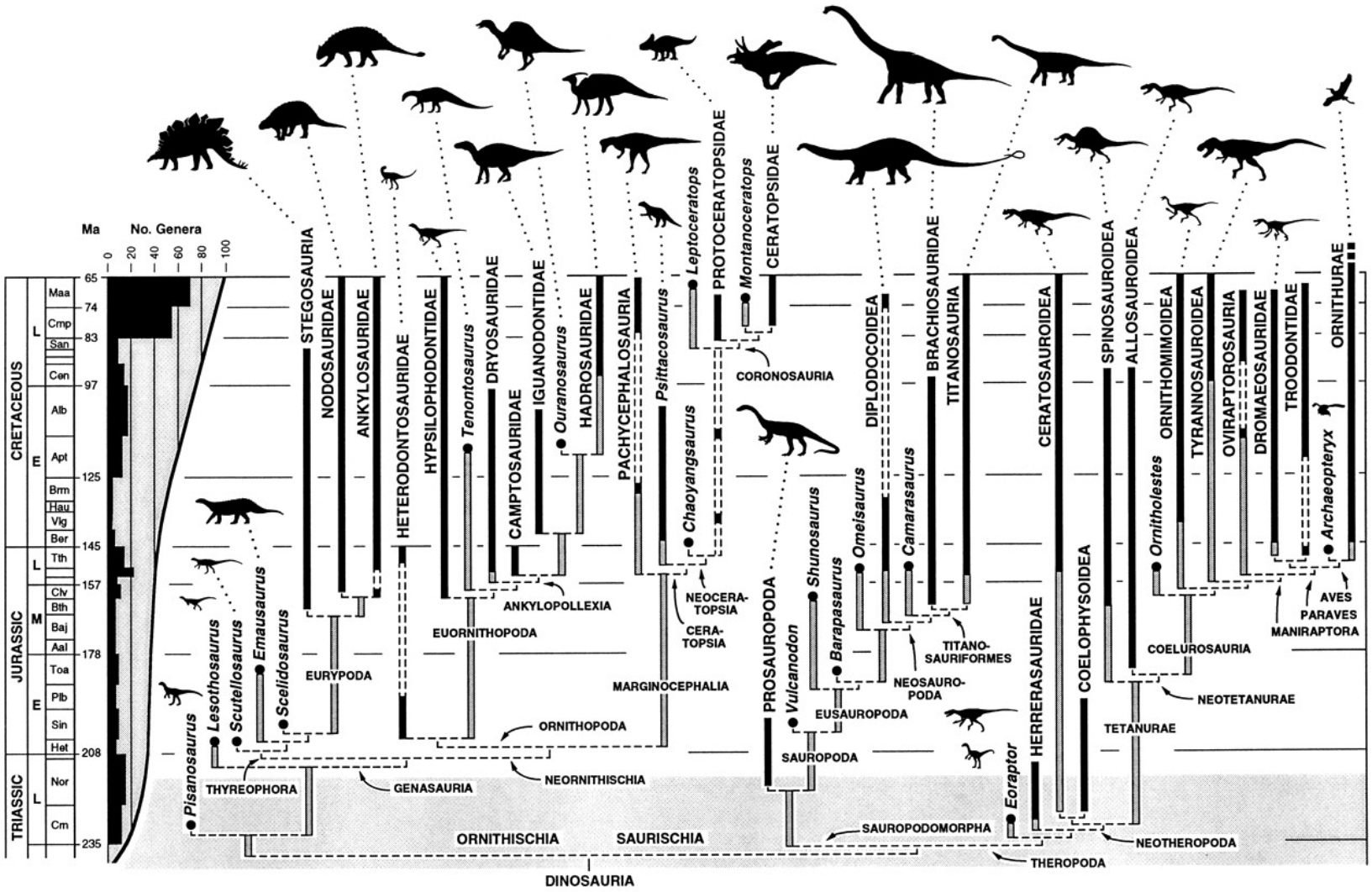
- It can be determined in $O(n^3)$ time whether or not an $n \times n$ distance matrix D is ultrametric

- Ultrametric distance matrices model evolutionary trees
  - An evolutionary tree is a rooted binary tree with internal nodes labeled by a number and with strictly decreasing labels along any root-to-leaf path
  - For every two leaves i and j, their distance $D_{i,j}$ is the label of the least common ancestor of species i and j

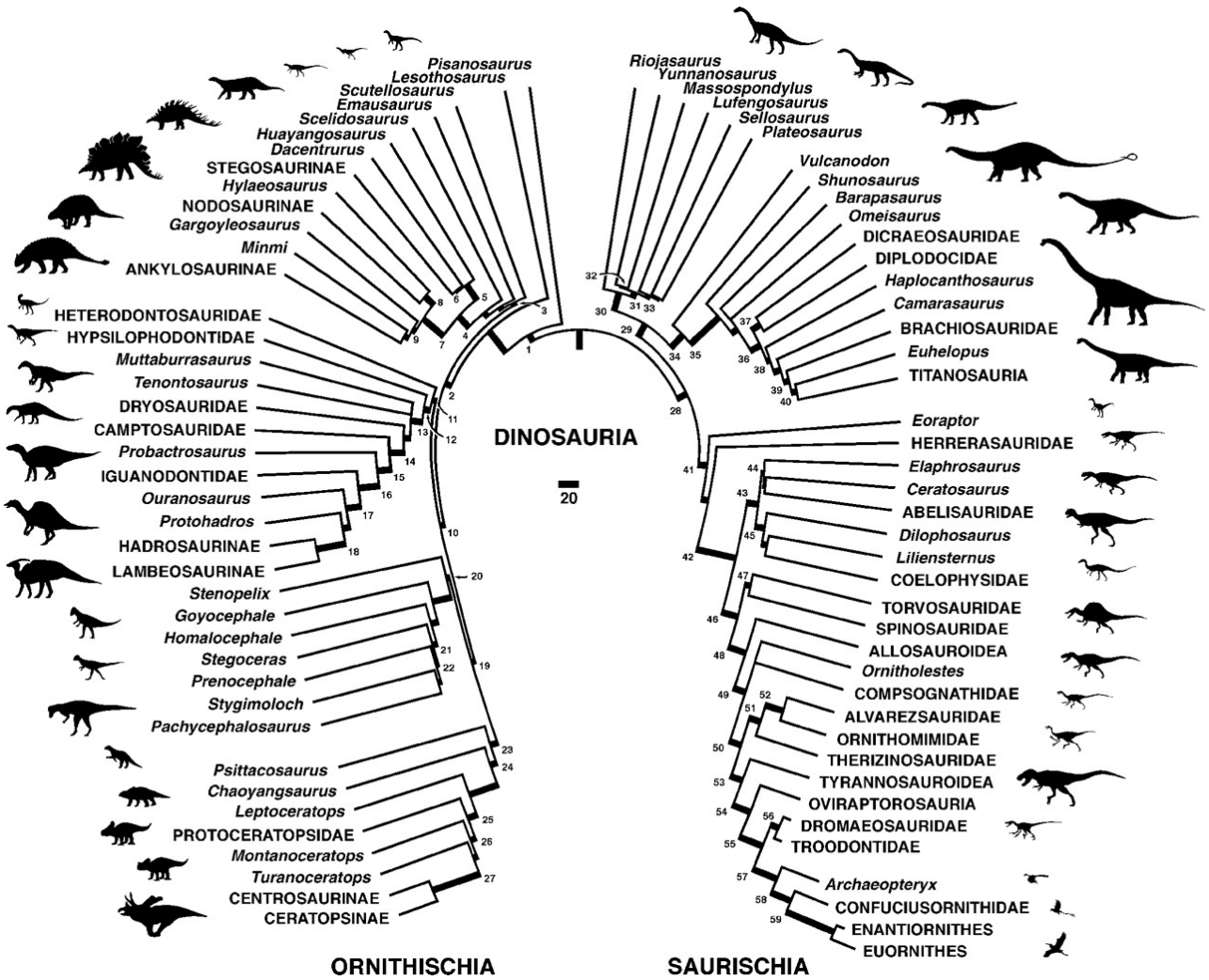
$$\begin{pmatrix} 0 & 1 & 2 & 4 & 8 & 14 & 14 \\ 1 & 0 & 2 & 4 & 8 & 14 & 14 \\ 2 & 2 & 0 & 4 & 8 & 14 & 14 \\ 4 & 4 & 4 & 0 & 8 & 14 & 14 \\ 8 & 8 & 8 & 8 & 0 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 & 0 & 12 \\ 14 & 14 & 14 & 14 & 14 & 12 & 0 \end{pmatrix}$$



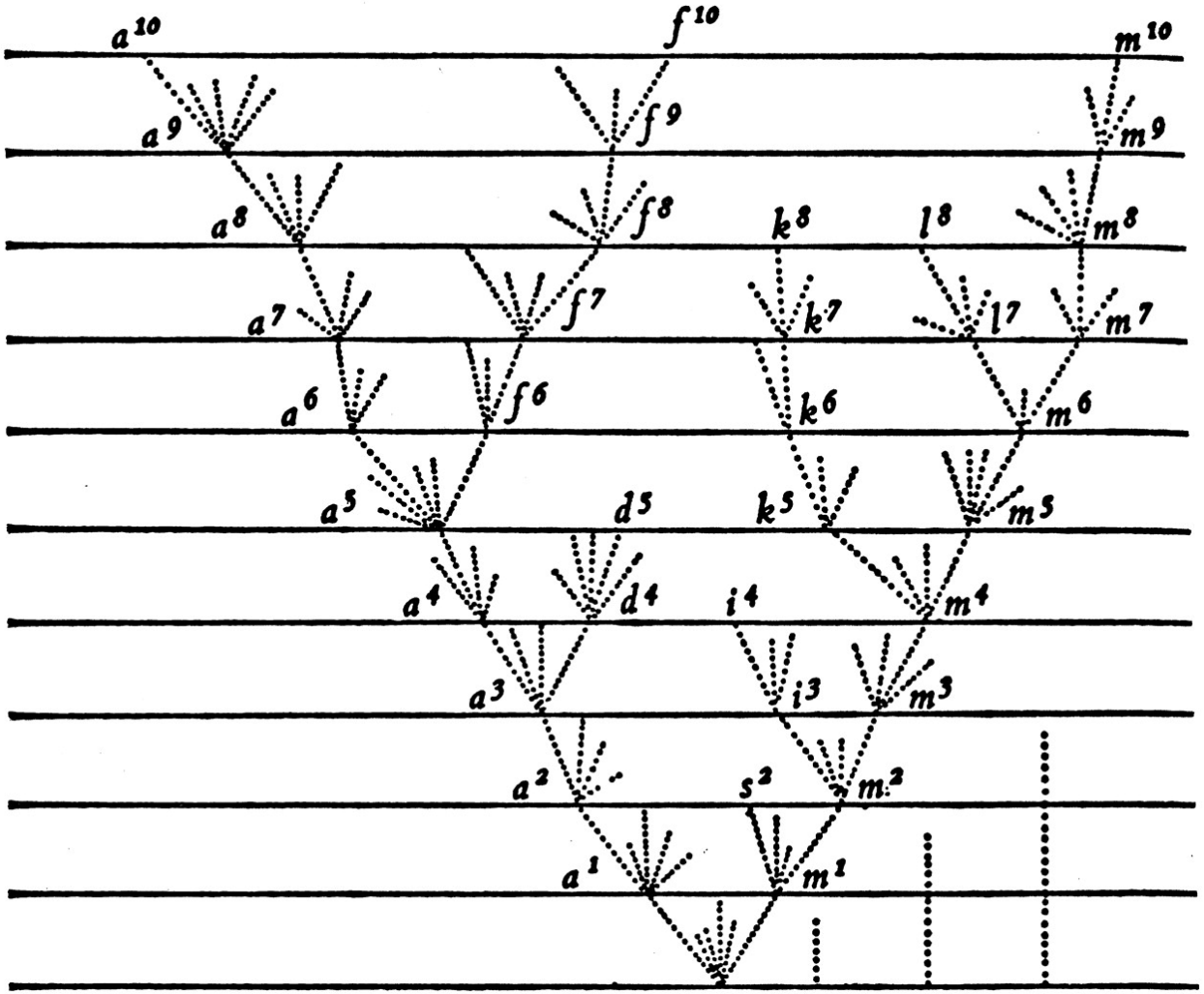
- The (evolutionary) distance $D_{i,j}$ between two species i and j measures the length of time since the species diverged



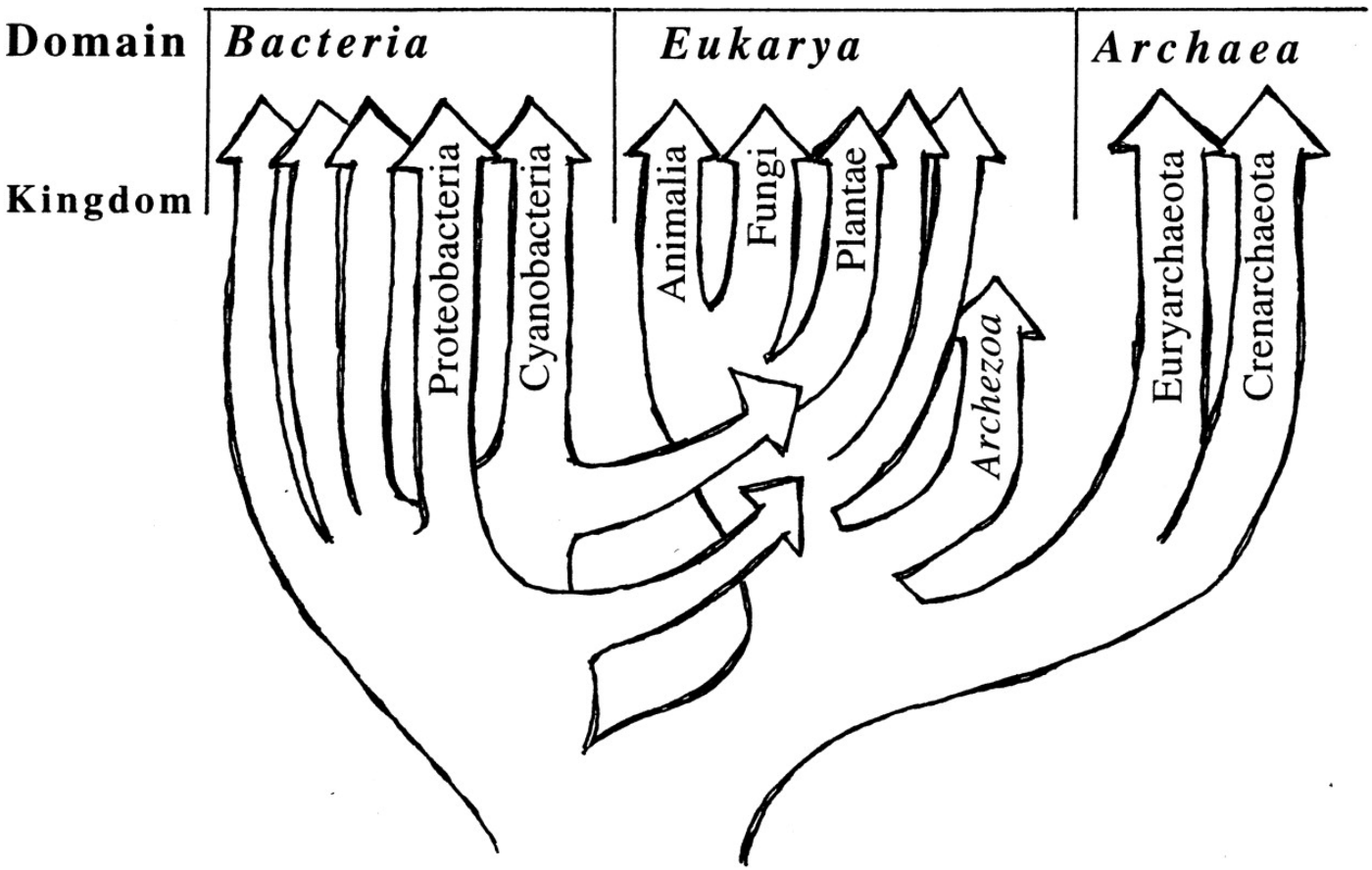
- The (evolutionary) distance $D_{i,j}$ between two species i and j measures the length of time since the species diverged



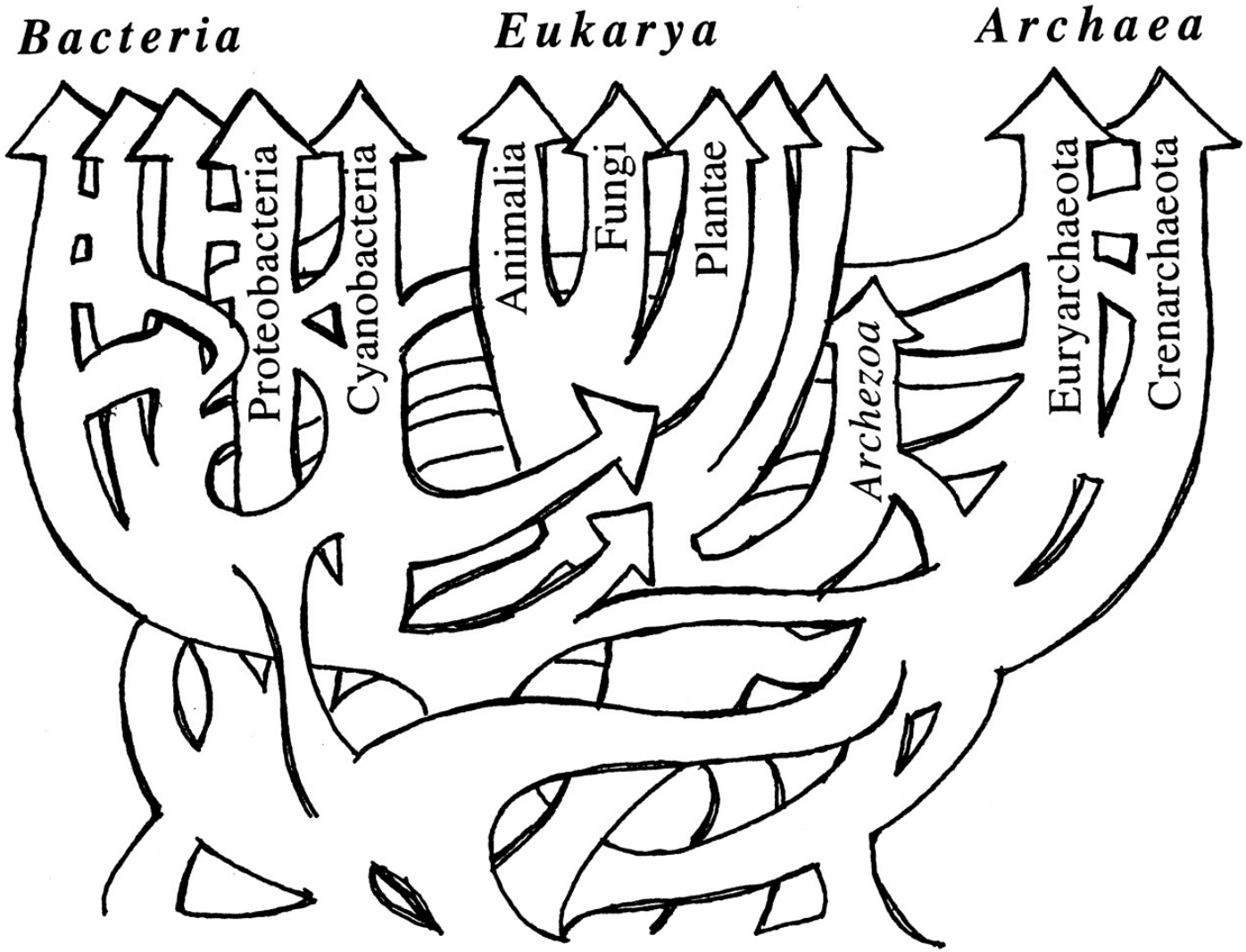
- The (evolutionary) distance $D_{i,j}$ between two species i and j measures the length of time since the species diverged



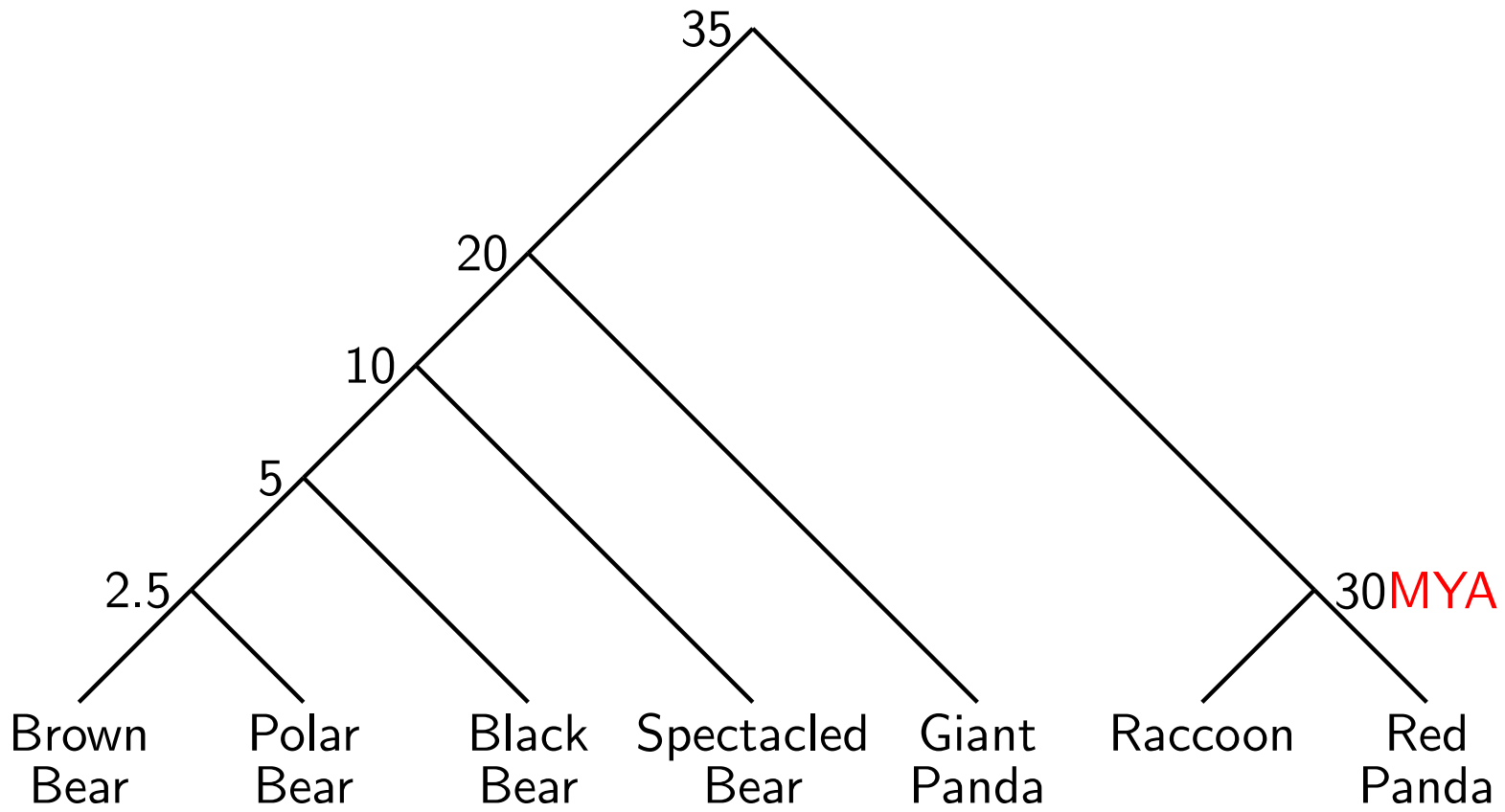
- The (evolutionary) distance $D_{i,j}$ between two species i and j measures the length of time since the species diverged



- The (evolutionary) distance $D_{i,j}$ between two species i and j measures the length of time since the species diverged

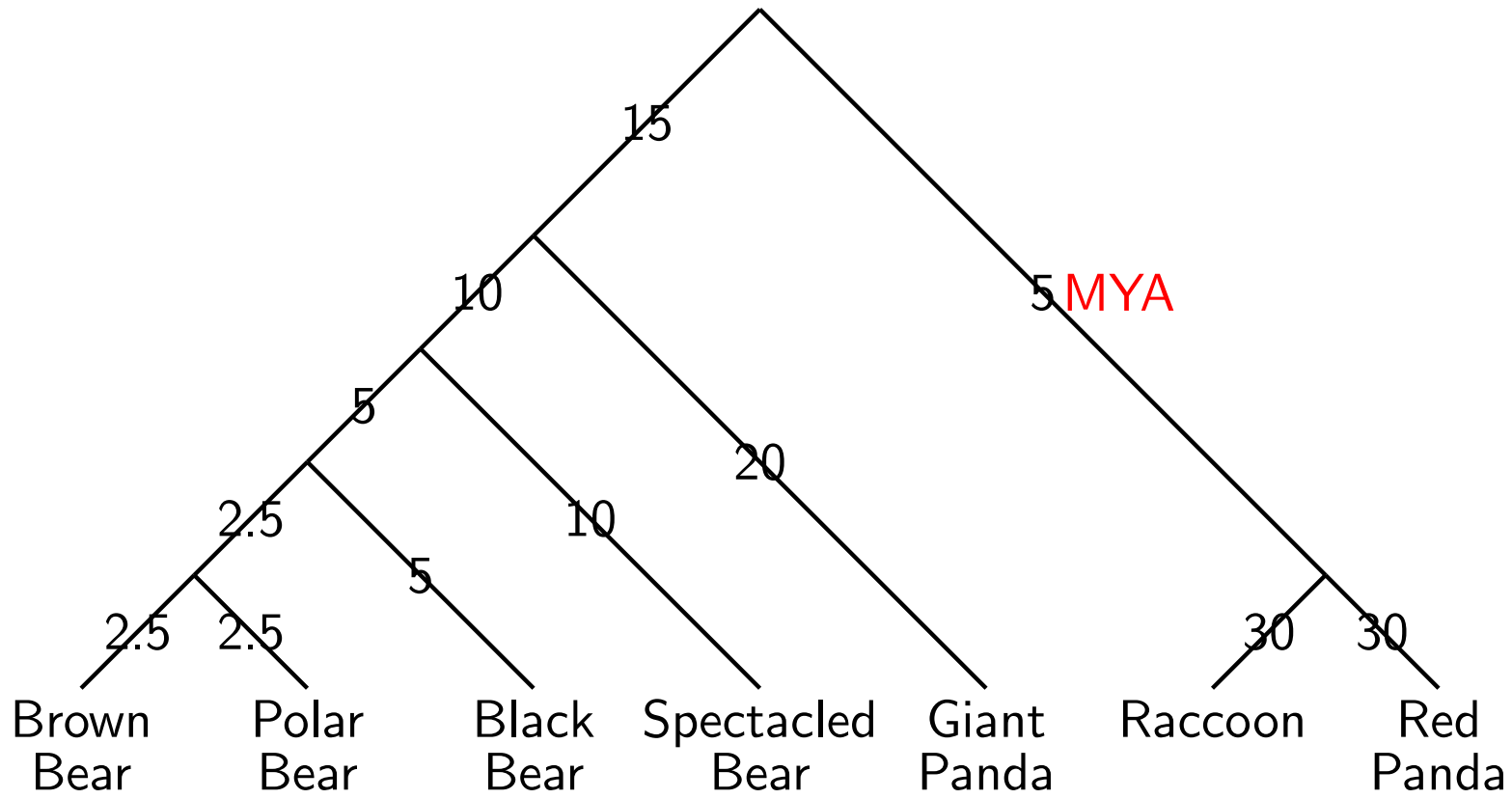


- The (evolutionary) distance $D_{i,j}$ between two species i and j measures the length of time since the species diverged



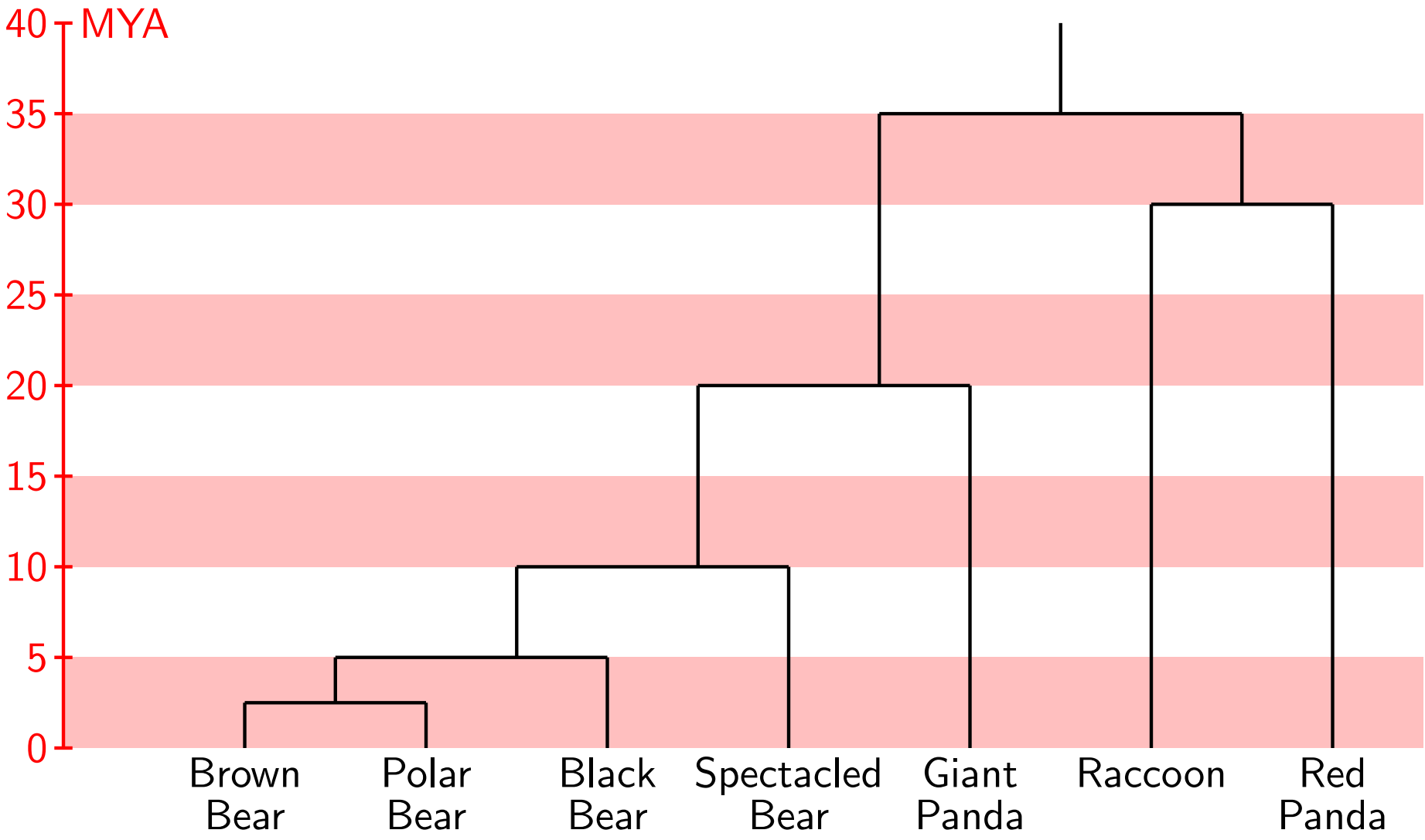
– No correlation between evolutionary distances and edge lengths

- The (evolutionary) distance $D_{i,j}$ between two species i and j measures the length of time since the species diverged

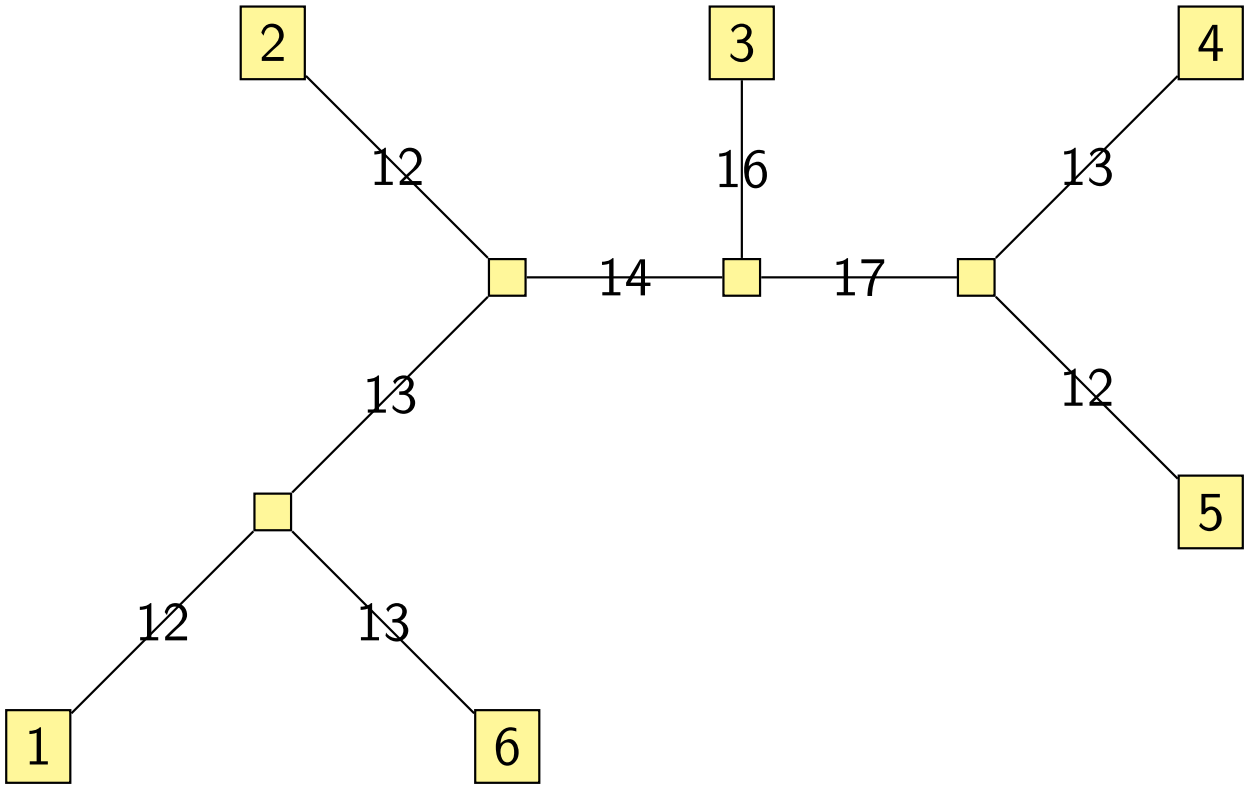


– No correlation between evolutionary distances and edge lengths

- The (evolutionary) distance $D_{i,j}$ between two species i and j measures the length of time since the species diverged



- Given a weighted tree T with n leaves, compute the length $d_{i,j}^T$ of the path between any two leaves i and j



- The length of the path between any two nodes can be calculated as the sum of the weights of the edges in the path between them
- For example, $d_{1,5}^T = 12 + 13 + 14 + 17 + 12 = 68$

- Unweighted Pair Group Method with Arithmetic Mean is an algorithm for reconstructing a tree T from an ultrametric distance matrix D
- P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W. H. Freeman, San Francisco, CA, 1973

– Starting with n clusters of one element each, merge the two closest clusters until only a single cluster remains

– The distance between two disjoint clusters C_i and C_j is defined as the average inter-cluster pairwise distance,

$$D(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{i \in C_i} \sum_{j \in C_j} D_{i,j}$$

– The length of an edge (u, v) is defined as the difference in heights of the vertices u and v

– The height plays the role of the molecular clock, and allows one to date the divergence point for every vertex in the evolutionary tree

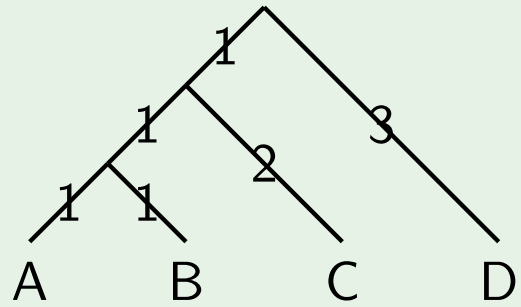
- Unweighted Pair Group Method with Arithmetic Mean is an algorithm for reconstructing a tree T from an ultrametric distance matrix D

Form n clusters, each with a single element
Construct a graph T with a vertex v of height $h(v) = 0$ for each cluster
while there is more than one cluster **do**
    Find the two closest clusters C_i and C_j
    Merge C_i and C_j into a new cluster C
    for every cluster $C' \neq C$ **do**
        Set $D(C, C')$ to the average distance between elements of C and C'
    Add a new vertex C to T and connect it to vertices C_i and C_j
    Assigh $h(C) = D(C_i, C_j)/2$
    Assign length $h(C) - h(C_i)$ to edge (C_i, C)
    Assign length $h(C) - h(C_j)$ to edge (C_j, C)
    Remove rows and columns of D corresponding to C_i and C_j
    Add a row and column to D for the new cluster C

- Given an $n \times n$ ultrametric distance matrix D, the unique tree T with n leaves that fits the data can be reconstructed in $O(n^2)$ time using the UPGMA algorithm

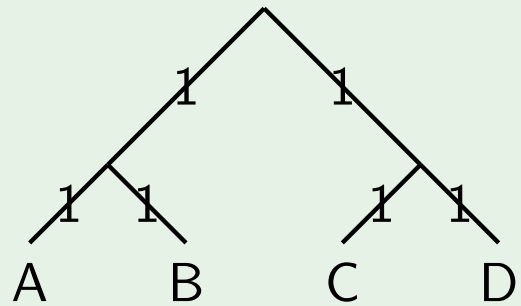
Example

$$\begin{pmatrix} 0 & 2 & 4 & 6 \\ 2 & 0 & 4 & 6 \\ 4 & 4 & 0 & 6 \\ 6 & 6 & 6 & 0 \end{pmatrix}$$



Example

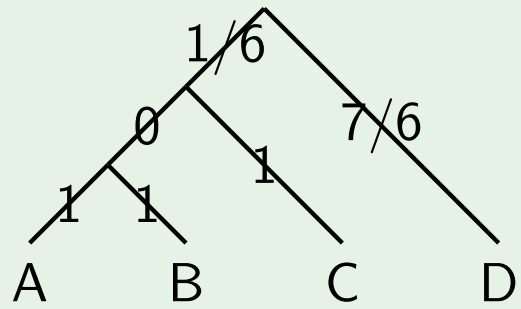
$$\begin{pmatrix} 0 & 2 & 4 & 4 \\ 2 & 0 & 4 & 4 \\ 4 & 4 & 0 & 2 \\ 4 & 4 & 2 & 0 \end{pmatrix}$$



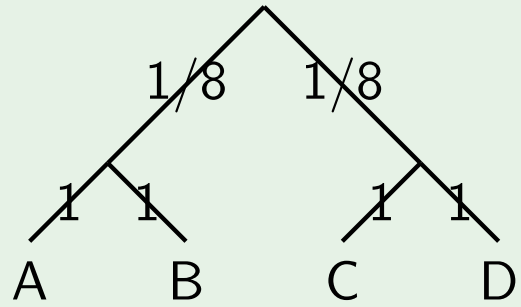
- Given an $n \times n$ distance matrix D, the tree T with n leaves that can be reconstructed using the UPGMA algorithm is not necessarily unique nor does it fit the data unless D is ultrametric

Example

$$\begin{pmatrix} 0 & 2 & 2 & 2 \\ 2 & 0 & 3 & 2 \\ 2 & 3 & 0 & 2 \\ 2 & 2 & 2 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 2 & 2 & 2 \\ 2 & 0 & 3 & 2 \\ 2 & 3 & 0 & 2 \\ 2 & 2 & 2 & 0 \end{pmatrix}$$

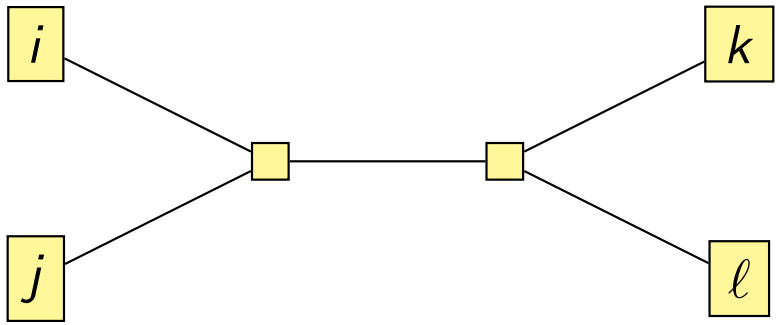


- A distance matrix D is additive if for every four leaves i, j, k, and ℓ, of the three sums of distances

$$D_{i,j} + D_{k,\ell} \qquad D_{i,k} + D_{j,\ell} \qquad D_{i,\ell} + D_{j,k}$$

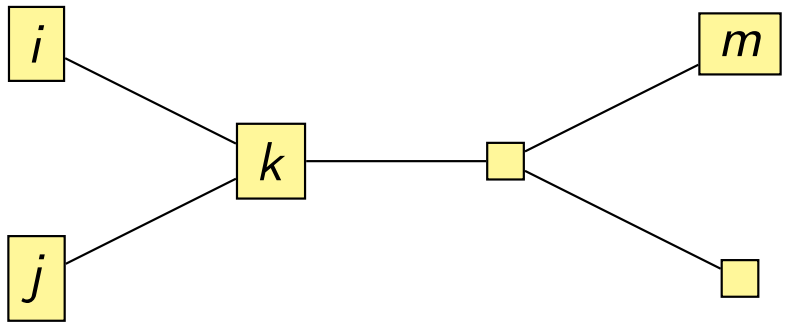
the two largest are equal (four point condition)

- $D_{i,j} + D_{k,\ell} \leqslant D_{i,k} + D_{j,\ell} = D_{i,\ell} + D_{j,k}$
- $D_{i,k} + D_{j,\ell} \leqslant D_{i,j} + D_{k,\ell} = D_{i,\ell} + D_{j,k}$
- $D_{i,\ell} + D_{j,k} \leqslant D_{i,j} + D_{k,\ell} = D_{i,k} + D_{j,\ell}$



- It can be determined in $O(n^4)$ time whether or not an $n \times n$ distance matrix D is additive

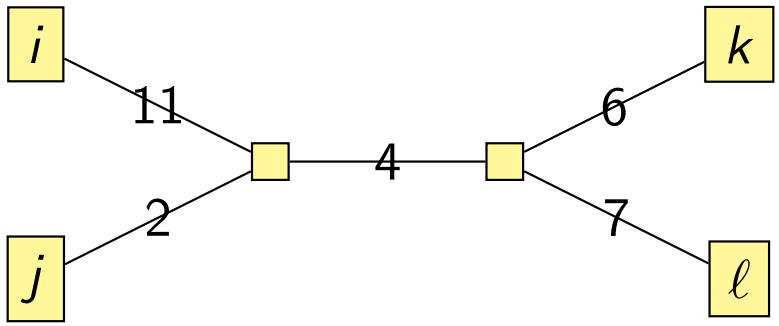
- Neighbor Joining is an algorithm for reconstructing a tree T from an additive distance matrix D
- N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987



- Find neighboring leaves i and j, and assign parent k to them
- Remove rows and columns of i and j
- Add a row and column for k, with distance to every other leaf m

$$D_{k,m} = \frac{D_{i,m} + D_{j,m} - D_{i,j}}{2}$$

- Neighbor Joining is an algorithm for reconstructing a tree T from an additive distance matrix D
  - Closest leaves (leaves i and j with minimum $D_{i,j}$) are not necessarily neighbors



  - Leaves i and j are neighbors, but $D_{i,j} = 13 > 12 = D_{j,k}$

- Neighbor Joining is an algorithm for reconstructing a tree T from an additive distance matrix D
  - Starting with n clusters of one element each, merge the two closest, and far apart from the rest, clusters until only a single cluster remains
  - Define the separation of cluster C from other clusters as

$$u(C) = \frac{1}{\# - 2} \sum_{C' \neq C} D(C, C')$$

  - Simultaneously minimize $D(C_i, C_j)$ and maximize $u(C_i) + u(C_j)$
  - Minimize $D(C_i, C_j) - u(C_i) - u(C_j)$

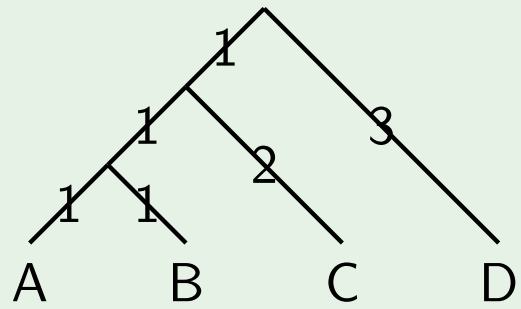
- Neighbor Joining is an algorithm for reconstructing a tree T from an additive distance matrix D

Form n clusters, each with a single element
Construct a graph T with an isolated vertex for each cluster
while there is more than one cluster **do**
    Find clusters C_i and C_j minimizing $D(C_i, C_j) - u(C_i) - u(C_j)$
    Merge C_i and C_j into a new cluster C
    for every cluster $C' \neq C$ **do**
        Set $D(C, C')$ to the average of $D(C_i, C')$ and $D(C_j, C')$
    Add a new vertex C to T and connect it to vertices C_i and C_j
    Assign length $(D(C_i, C_j) + u(C_i) - u(C_j))/2$ to edge (C_i, C)
    Assign length $(D(C_i, C_j) + u(C_j) - u(C_i))/2$ to edge (C_j, C)
    Remove rows and columns of D corresponding to C_i and C_j
    Add a row and column to D for the new cluster C

- Given an $n \times n$ additive distance matrix D, the unique tree T with n leaves that fits the data can be reconstructed in $O(n^5)$ time using the NJ algorithm

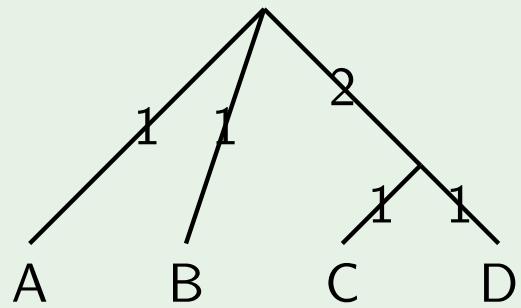
Example

$$\begin{pmatrix} 0 & 2 & 4 & 6 \\ 2 & 0 & 4 & 6 \\ 4 & 4 & 0 & 6 \\ 6 & 6 & 6 & 0 \end{pmatrix}$$



Example

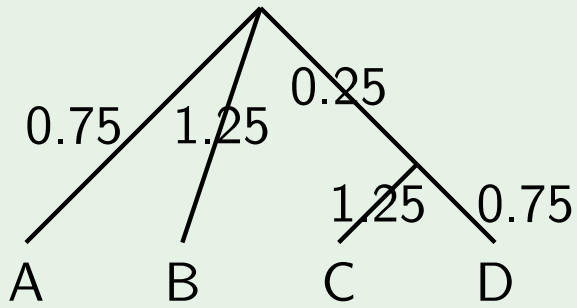
$$\begin{pmatrix} 0 & 2 & 4 & 4 \\ 2 & 0 & 4 & 4 \\ 4 & 4 & 0 & 2 \\ 4 & 4 & 2 & 0 \end{pmatrix}$$



- Given an $n \times n$ distance matrix D, the tree T with n leaves that can be reconstructed using the NJ algorithm does not necessarily fit the data unless D is additive

Example

$$\begin{pmatrix} 0 & 2 & 2 & 2 \\ 2 & 0 & 3 & 2 \\ 2 & 3 & 0 & 2 \\ 2 & 2 & 2 & 0 \end{pmatrix}$$



- Neighbor Joining is an algorithm for reconstructing a tree T from an additive distance matrix D
- N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987 $O(n^5)$
- J. A. Studier and K. J. Keppler. A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution*, 5(6):729–731, 1988 $O(n^3)$
- R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, England, 1998 Appendix 7.8: Proof of neighbour-joining theorem
- T. Mailund, G. S. Brodal, R. Fagerberg, C. N. S. Pedersen, and D. Phillips. Recrafting the neighbor-joining method. *BMC Bioinformatics*, 7:29, 2006 $O(n^3)$