Phylogenetic Prediction

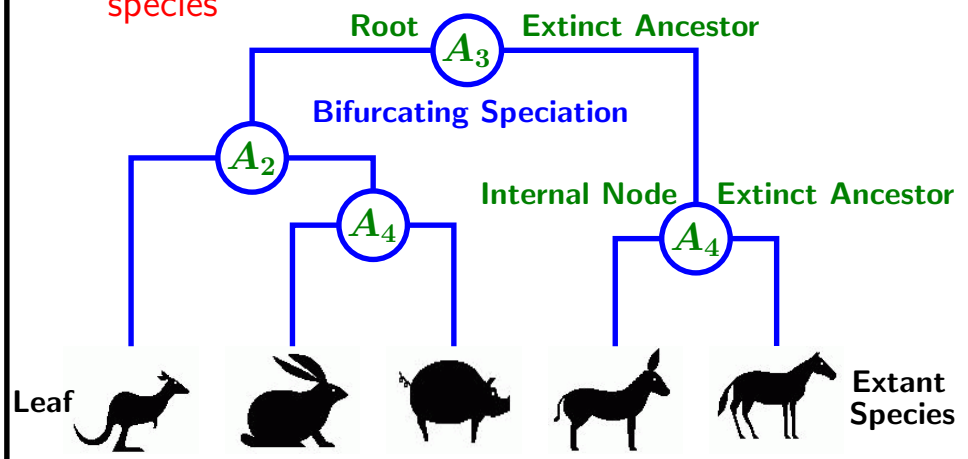
Chin Lung Lu

Computational Biology

Analyses and Applications of Sequences

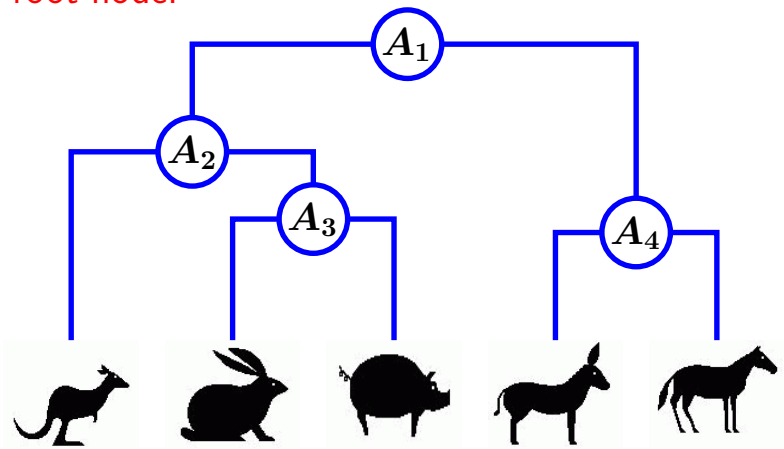
Evolutionary tree

- To describe the evolutionary relationship among species



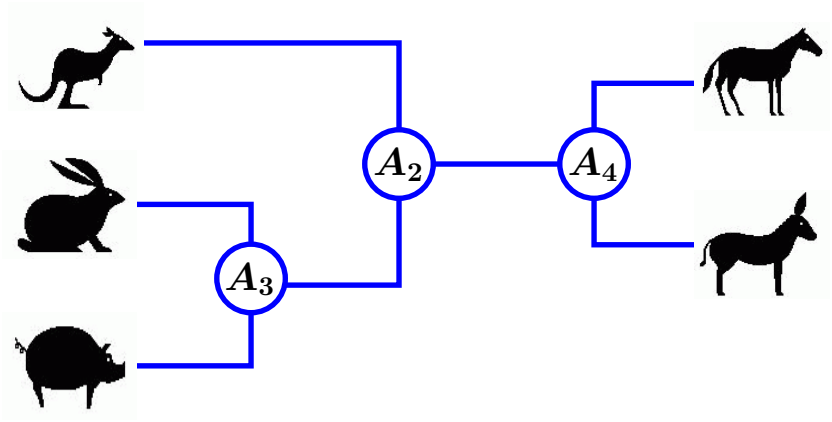
Rooted Evolutionary Tree

- The degree of each internal node is 3, except the root node.

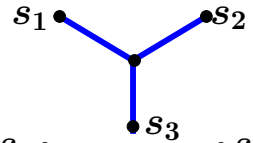
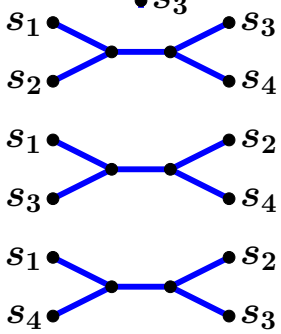


Unrooted Evolutionary Tree

- The degree of each internal node is 3.

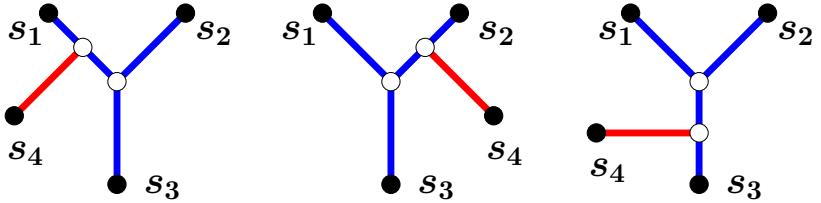


Number of unrooted trees ❶

	No of Trees	Structure of Trees	No of Edges
$n = 2$	1		1
$n = 3$	1		3
$n = 4$	3		5

Number of unrooted trees ❷

- How to add a new species into the tree?



- If a new species is added to an unrooted tree, the number of edges is increased by 2.
- $NE(n)$: number of edges of an unrooted tree with n species
  - By induction, we have $NE(n) = 2n - 3$.

Number of unrooted trees ❸

- Basic step: $NE(3) = 2 * 3 - 3 = 3$
- Hypothesis step: $NE(k-1) = 2(k-1) - 3$
  $NE(k) = NE(k-1) + 2 = 2 * k - 3$
- $TU(n)$: number of unrooted trees for n species
- Since $NE(n-1) = 2(n-1) - 3$, we have

$$TU(n) = (2n-5)TU(n-1)$$

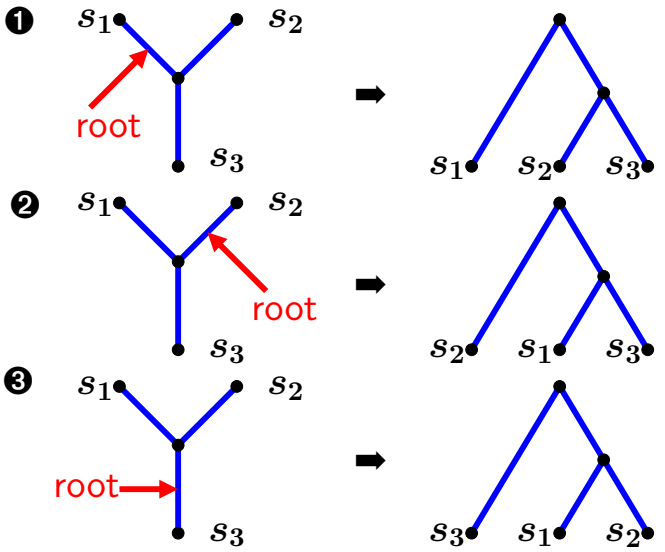
That is,

$$TU(n) = (2n-5)(2n-7)\cdots 1$$

Number of unrooted trees ❹

No of Species	No of Unrooted Trees
2	1
3	1
4	3
⋮	⋮
17	6,190,283,353,629,375
18	191,898,783,962,510,625
19	6,332,659,870,762,850,625
20	221,643,095,476,699,771,875

Change unrooted into rooted



Number of rooted trees ❶

- $TR(n)$: number of rooted trees for n species
- Since there are $2n-3$ edges in every unrooted tree for n species, we have

$$\begin{aligned} TR(n) &= (2n-3)TU(n) \\ &= (2n-3)(2n-5)(2n-7)\cdots 1 \\ &= TU(n+1) \end{aligned}$$

Number of rooted trees ❷

No of Species	No of Rooted Trees
2	1
3	3
4	15
⋮	⋮
17	191,898,783,962,510,625
18	6,332,659,870,762,850,626
19	221,643,095,476,699,771,875
20	8,200,794,532,637,891,559,375

Methods of constructing trees

1. Character-based data:
   - Perfect phylogeny method
   - Maximum parsimony (likelihood) method
2. Distance-based data:
   - Additive (Ultrametric) tree constructing method
   - UPGMA method
   - Neighbor-joining method

Character-based data

1. Non-molecular data (morphological features):
   - Character: a morphological feature
   - State: presence or absence (value, number) of a morphological feature
   - **Example**: Vertebrate? (yes or no) , Height? (162-182 cm), Number of fingers? (4, 5, or 6)
2. Molecular data (DNA, protein sequences):
   - Character: a position in the aligned sequences
   - State: the particular nucleotide or amino acid at this position

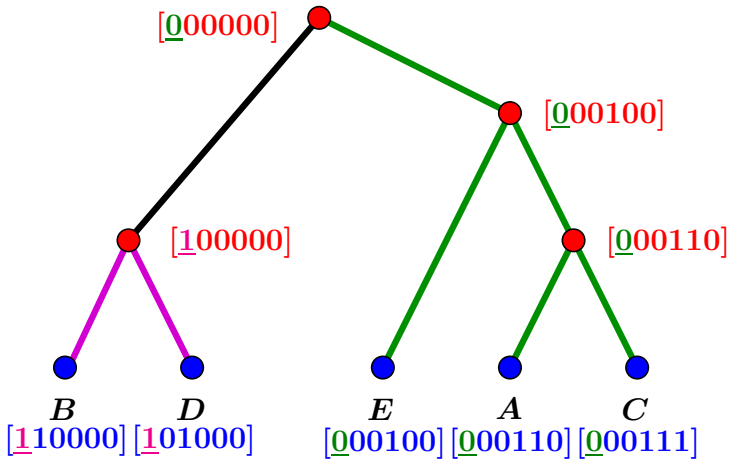
Character-based data

- Character state matrix: a matrix $\mathcal{M}$ with n rows (species, objects) and m columns (characters) in which each $\mathcal{M}_{i,j}$ denotes the state of species i for character j

Object\Character	C_1	C_2	C_3	C_4	C_5	C_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

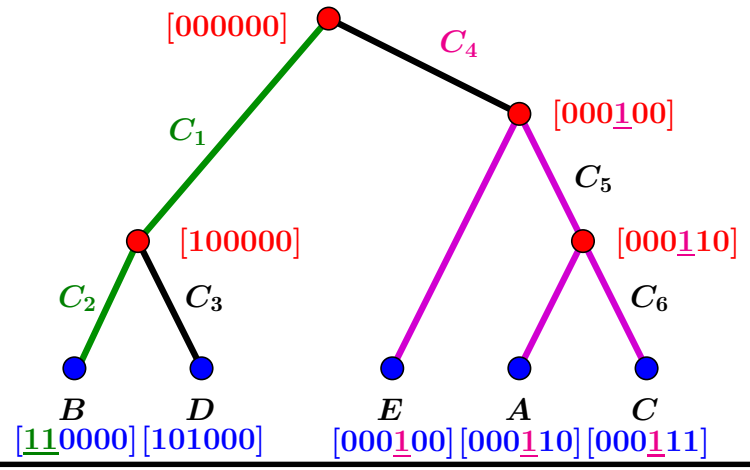
Perfect phylogeny

- Perfect phylogeny: for each C_i, the set of nodes with the same state of C_i form a subtree



Perfect phylogeny

- Each C_i labels exactly one edge (i.e., any node below this edge has state 1 for C_i)



Perfect phylogeny problem

- **Instance**: a character state matrix $\mathcal{M}_{n\times m}$, each character having at most r states
- **Question**: Is there a perfect phylogeny for $\mathcal{M}$?
- NP-compelte, by Bodlaender et al. [BFW92] and Steel [Ste92] independently
- Solvable in polynomial time for $r \leq 4$
  - $r = 2$ by Gusfield [Gus91],
  - $r = 3$ by Dress and Steel [DS92]
  - $r = 4$ by Kannan and Warnow [KW94]
- Solvable in $\mathcal{O}(2^{2r}nm^2)$ time for $r > 4$ [KW95]

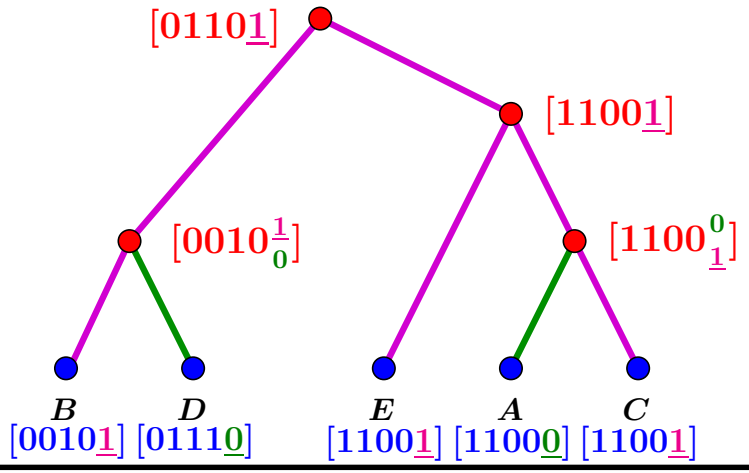
Maximum parsimony tree

- If $\mathcal{M}$ has no pefect phylogeny, there are some states which will induce a forest, instead of a subtree

Object\Character	C_1	C_2	C_3	C_4	C_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

Maximum parsimony tree

- Given a tree topology, the state 0 of C_5 induces a forest (2 subtrees), instead of a tree.



Parsimony tree length

- For two sequences $u = u_1 \cdots u_m$, $v = v_1 \cdots v_m$ of the same length, their Hamming distance is

$$H(u,v) = \sum_{i=1}^{m} |\{i : u_i \neq v_i\}|$$

- For every node v of a phylogeny $\mathcal{T} = (V, E)$, let $S(v)$ denote the sequence labeled to v.
- Given $\mathcal{T} = (V, E)$, its parsimony tree length is

$$PTlegth(\mathcal{T}) = \sum_{(u,v)\in E} H(S(u), S(v))$$

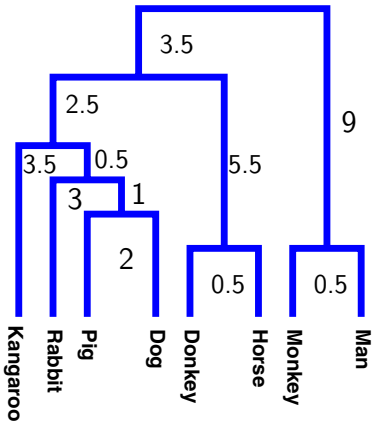
Maximum parsimony problem

- **Instance**: a character state matrix $\mathcal{M}_{n \times m}$, each character having at most r states
- **Question**: find a tree in which each internal node is labeled by a sequence of length m such that its parsimony tree length is minimized
- NP-complete, by Day et al. [DJS86]
- Branch and bound, by Hendy and Penny [HP82]
- Linearly solvable if the topology of a leaf-labeling tree is given, by Fitch (gave the algorithm) and Hartigan (proved the correctness) [Fit71, Har73]

Distance-based data

	Man	Monkey	Dog	Horse	Donkey	Pig	Rabbit	Kangaroo
Man	0	1	13	17	16	13	12	12
Monkey		0	12	16	15	12	11	13
Dog			0	10	8	4	6	7
Horse				0	1	5	11	11
Donkey					0	4	10	12
Pig						0	6	7
Rabbit							0	7
Kangaroo								0

Distance Matrix M



Evolutionary Tree T

Additive and ultrametric matrices

- A distance matrix M is additive if it is possible to find an edge-weighted unrooted tree T such that for any 2 species i and j, $d^M_{i,j} = d^T_{i,j}$.
  - $d^M_{i,j}$: the distance of species i and j in M
  - $d^T_{i,j}$: the distance of path between i and j in T
- Ultrametric matrix: an additive matrix whose additive tree can be rooted in such a way that the lengths of all the root-leaf paths are equal
- Given an additive (ultrametric) matrix of n species, the unique additive (ultrametric) tree can be constructed in $\mathcal{O}(n^2)$ time [WSSB77] ([KLW90]) .

Distance-based problem

- **Input**: A distance matrix M of n species
- **Output**: Find an evolution tree T such that when two species are close to each other in the distance matrix, they should be close in the evolution tree
- Almost all of the distance-based problems have been shown to be NP-complete [Day87, FKW93].
- Heuristic methods:
  - UPGMA (Unweighted Pair-Group Method using Arithmetic mean) mehtod [SS63]
  - Neighbor joining method [SN87]

PP problem: binary characters ❶

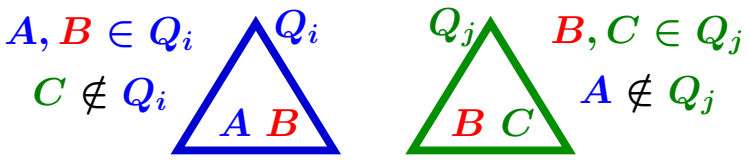
- **Instance**: a character state matrix $\mathcal{M}_{n\times m}$, each character having 2 states
- **Question**: Is there a perfect phylogeny for $\mathcal{M}$?
- Solvable in $O(nm)$ time, by Gusfield [Gus91]
- O_i= the set of species whose state of C_i is 1
- $\overline{O_i}$= the set of species whose state of C_i is 0
- **Assume**: 0 is ancestral state and 1 is derived state.
  - The state of each character of root is zero.
  - $1 \longrightarrow 0$ is not allowed.

PP problem: binary characters ❷

Lemma: $\mathcal{M}$ has a perfect phylogeny iff for each pair of i and j, O_i and O_j are disjoint or one of them contains the other.

Proof ($\Rightarrow$):

1. Each C_i labels exactly one edge $u \longrightarrow v$ of tree.
2. The subtree rooted at v contains all nodes having 1 for C_i and contains no node having 0 for C_i.



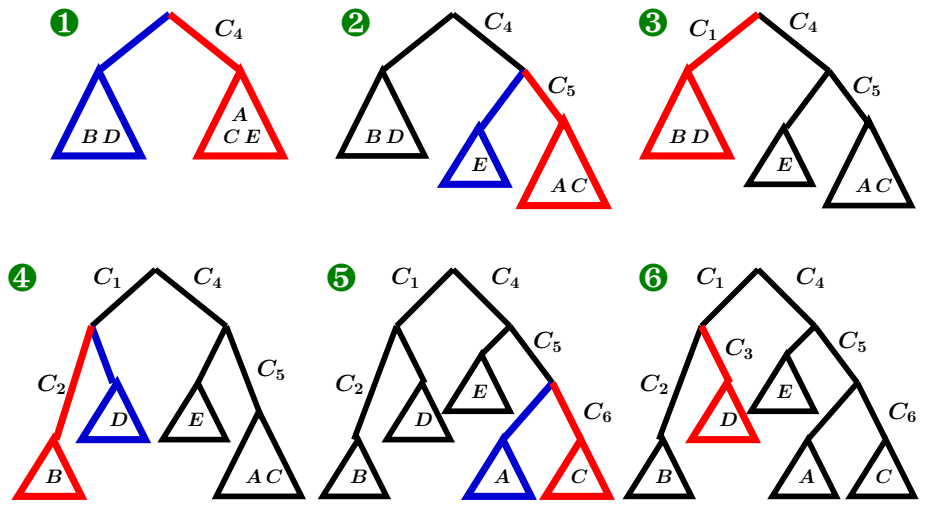
PP problem: binary characters ❸

Proof ($\Leftarrow$): Create a perfect phylogeny as follows

1. Consider each column of $\mathcal{M}$ as a binary number and sort all columns in decreasing order of these numbers and place the largest one in column 1.

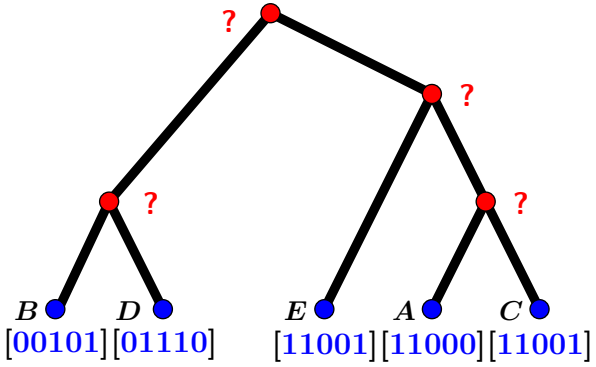
Object\Character	C_4	C_5	C_1	C_2	C_6	C_3
A	1	1	0	0	0	0
B	0	0	1	1	0	0
C	1	1	0	0	1	0
D	0	0	1	0	0	1
E	1	0	0	0	0	0

PP problem: binary characters ❹



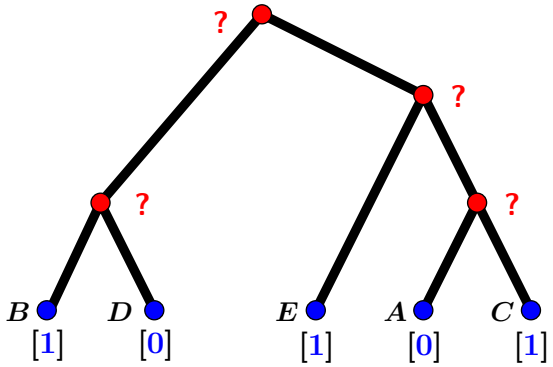
Parsimony problem: a fixed tree ❶

- **Instance**: The topology of a rooted tree $\mathcal{T}$ with leaves having sequence labels of the same length
- **Question**: What is the labeling of the internal nodes with the minimum parsimony tree length?



Fitch's algorithm: notation ❷

- Since characters are mutually independent, we consider each character separately.



- $l(v)$: the value of the character for node v in $\mathcal{T}$

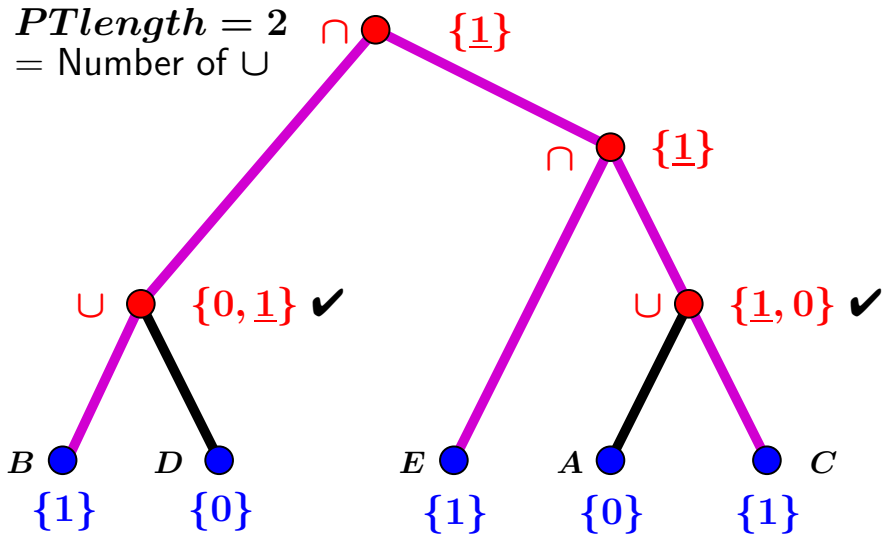
Fitch's algorithm ❸

1: **for** each node v of $\mathcal{T}$ in postorder **do**
  if v is a leaf **then**
    $S(v) = \{l(v)\}$;
  else /*let v_l and v_r be two children of v*/

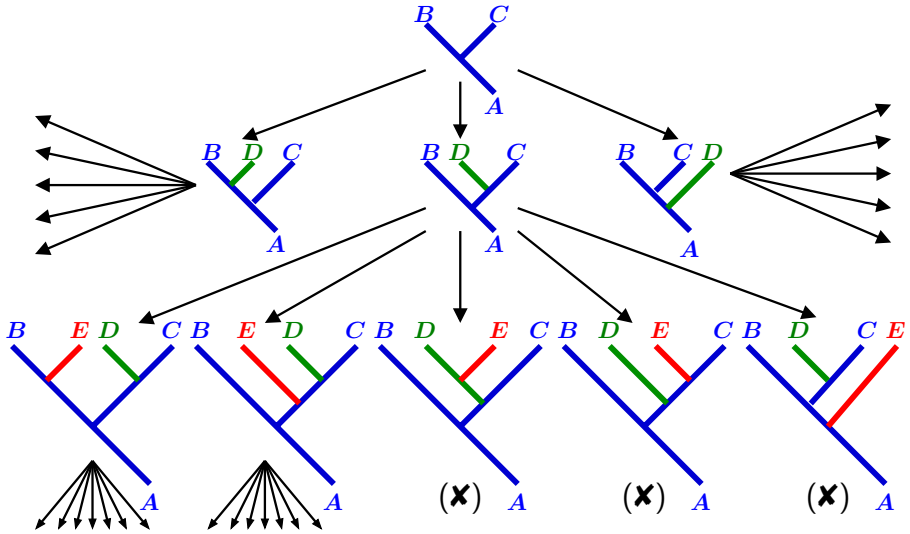
$$S(v)=\begin{cases} S(v_l) \cap S(v_r) & \text{if } S(v_l) \cap S(v_r) \neq \varnothing \\ S(v_l) \cup S(v_r) & \text{otherwise} \end{cases}$$

2: Arbitrarily assign any one in $S(root)$ to $l(root)$;
for each internode v of $\mathcal{T}$ in preorder **do**
  if the parent u of v satisfies $l(u) \in S(v)$ **then**
    $l(v) = l(u)$;
  else arbitrarily assign any one in $S(v)$ to $l(v)$;

Fitch's algorithm: example ❹



Branch and bound method ❺

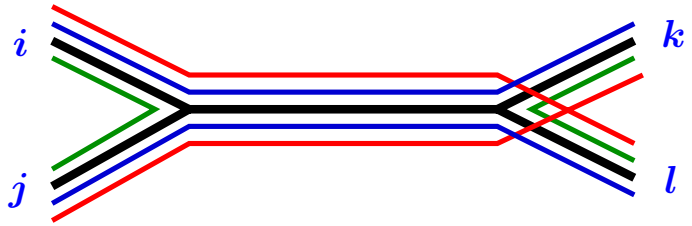


Distance metric ❶

- A distance function d: $\{\text{species}\}^2 \longrightarrow R^+$ is a metric if it satisfies the following conditions:
  1. $d_{i,j} > 0$ for $i \neq j$
  2. $d_{i,j} = 0$ for $i = j$
  3. $d_{i,j} = d_{j,i}$ for all i and j (symmetric)
  4. $d_{i,j} \leq d_{i,k} + d_{k,j}$ for all i, j and k (the triangle inequality)
- Given a distance matrix M and a phylogeny T,
  - $d^M_{i,j}$: the distance of species i and j in M
  - $d^T_{i,j}$: the distance of path between i and j in T

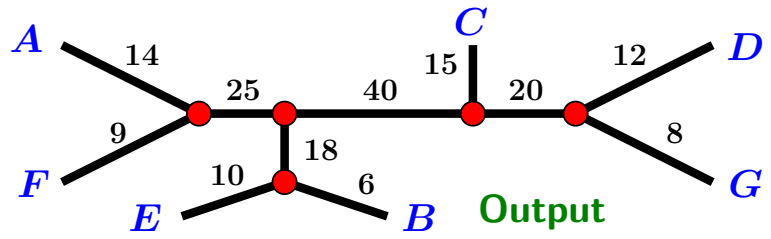
Additive matrix and tree ❷

- A distance matrix M is additive if it is possible to find an edge-weighted unrooted tree T such that for any two species i and j, $d^M_{i,j} = d^T_{i,j}$.
- **Four point condition**: A matrix M is additive iff for all i, j, k and l, the maximum of $d^M_{i,j} + d^M_{k,l}$, $d^M_{i,k} + d^M_{j,l}$ and $d^M_{i,l} + d^M_{j,k}$ is not unique.



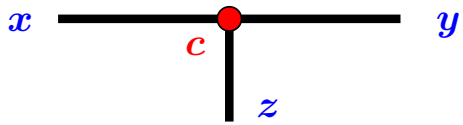
Additive tree problem ❸

Input: Additive	B	C	D	E	F	G
A	63	94	111	67	23	107
B		79	96	16	58	92
C			47	83	89	43
D				100	106	20
E					62	96
F						102



Additive tree of 3 species ❹

- Create a tree T_3 for three species x, y and z:



$\because d^M_{x,z} = d^T_{x,c} + d^T_{c,z}$ ❶ and $d^M_{y,z} = d^T_{y,c} + d^T_{c,z}$ ❷

$\therefore$ ❶ $-$ ❷ $\Rightarrow d^M_{x,z} - d^M_{y,z} = d^T_{x,c} - d^T_{y,c}$ ❸

$\because d^M_{x,y} = d^T_{x,c} + d^T_{y,c}$ ❹

$\therefore$ ❸ $+$ ❹ $\Rightarrow d^M_{x,z} + d^M_{x,y} - d^M_{y,z} = 2d^T_{x,c}$

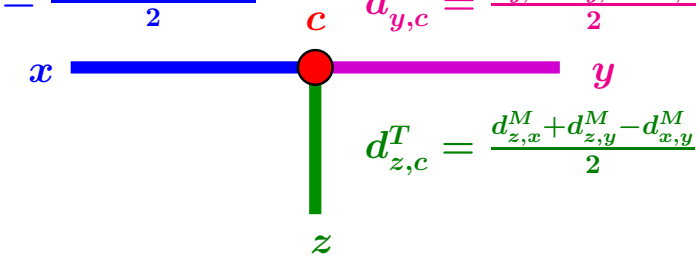
$$\therefore d^T_{x,c} = \frac{d^M_{x,z} + d^M_{x,y} - d^M_{y,z}}{2}$$

Additive tree of 3 species ❺

- The position of the center c of x, y, z is unique (i.e., T_3 is uniuqe).

$$d^T_{x,c} = \frac{d^M_{x,z}+d^M_{x,y}-d^M_{y,z}}{2} \qquad d^T_{y,c} = \frac{d^M_{y,x}+d^M_{y,z}-d^M_{x,z}}{2}$$

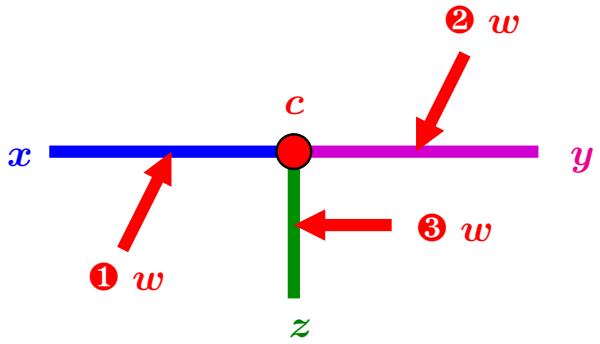
$$d^T_{z,c} = \frac{d^M_{z,x}+d^M_{z,y}-d^M_{x,y}}{2}$$



- How to add the fourth species w into T_3?

Add w into T_3 ❻

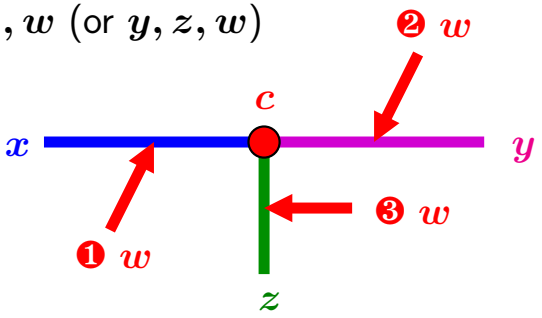
- There are 3 possibilities for adding w into T_3?



- It is impossible that there are at least two choices for adding w into T_3. (Why?)

Add w into T_3 ❼

1. Select any two species (say x and y) in T_3 and compute the position of the center c' of x, y, w
2. If $c' \neq c$, then done (i.e., $c' \in (x, c)$ or (y, c))
3. If $c' = c$, then $c' \in (z, c)$ and compute the center of x, z, w (or y, z, w)



Evolution Tree Problem

- **Input**: A distance matrix M of n species
- **Output**: Find an evolution tree T under some criterion
- **Basic criterion**: $d_T(s_i, s_j) \geq d_M(s_i, s_j)$
  - $d_M(s_i, s_j)$: the distance between species s_i and s_j in M
  - $d_T(s_i, s_j)$: the distance between s_i and s_j in T
- **Further criterion**: when two species are close to each other in the distance matrix, they should be close in the evolution tree

Criteria of Evolution Tree

- **MINIMAX** evolution tree: an evolutionary tree with $d_T(s_i, s_j) \geq d_M(s_i, s_j)$ such that $\max_{1 \leq i < j \leq n} [d_T(s_i, s_j) - d_M(s_i, s_j)]$ is minimized
- **MINISUM** evolution tree: an evolutionary tree with $d_T(s_i, s_j) \geq d_M(s_i, s_j)$ such that $\sum_{1 \leq i < j \leq n} d_T(s_i, s_j)$ is minimized
- **MINISIZE** evolution tree: an evolutionary tree with $d_T(s_i, s_j) \geq d_M(s_i, s_j)$ such that the total length of the tree is minimized

Complexities of Evolutionary Tree

	MINIMAX	MINISUM	MINISIZE
Rooted	$\mathcal{O}(n^2)$	NP-C	NP-C
Unrooted	NP-C	NP-C	?

- It is still unknown whether the unrooted MINISIZE evolutionary tree problem is NP-complete.

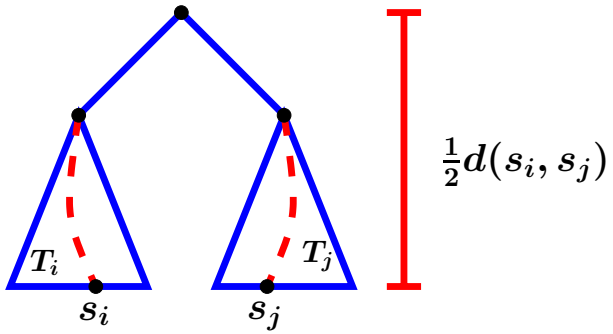
Minimax rooted tree algorithm

Input: a distance matrix of a set S of n species

1. If S contains only one species, return it as the tree.
2. Find a minimal spanning tree T of S.
3. Find the longest $d(s_i, s_j)$ in the distance matrix. Find the longest edge e in the path linking s_i and s_j in T. Let S_i and S_j be the two sets of species obtained by breaking edge e.
4. Use this algorithm recursively to find subtrees T_i and T_j for S_i and S_j respectively.
5. Construct a rooted tree with T_i and T_j as subtrees such that $dt(s_i, s_j) = d(s_i, s_j)$.

Basic priciple of Minimax

- Let s_i and s_j be the two species which have the longest distance in the distance matrix.



- The longest distance is exactly preserved.

Example of Minimax algorithm ❶

1. Consider the distance matrix as follows:

	s_1	s_2	s_3	s_4
s_1	0	**2**	**3**	3.1
s_2		0	3.6	5
s_3			0	**1**
s_4				0

2. Construct a minimal spanning tree T as follows:

Example of Minimax algorithm ❷

3. The distance between s_2 and s_4 is the longest.

	s_1	s_2	s_3	s_4
s_1	0	2	3	3.1
s_2		0	3.6	**5**
s_3			0	1
s_4				0

4. The path linking s_2 and s_4 in T in which (s_1, s_3) is the longest edge.

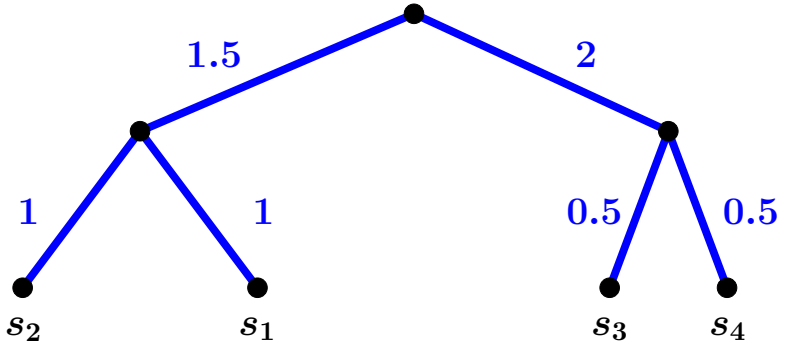
Example of Minimax algorithm ❸

5. Breaking (s_1, s_3) obtains two subsets of species

6. Construct two subtrees T_1 and T_2 for S_1 and S_2 respectively

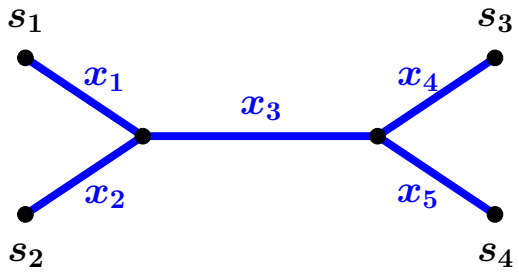
Example of Minimax algorithm ❹

7. Combine T_1 and T_2 by making sure that $dt(s_2, s_4) = d(s_2, s_4) = 5$

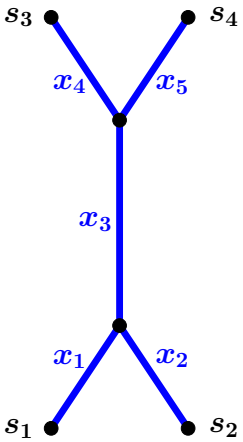


Determination of edge weights

- How to determine the edge weights if the evolutionary tree structure is given?
- **Example**: Given a distance matrix and an unrooted evolutionary tree, determne the edge weights such that the tree size is minimized



Determination of edge weights



Unrooted Tree

Determine x_i by linear programming

Minimize $x_1 + x_2 + x_3 + x_4 + x_5$

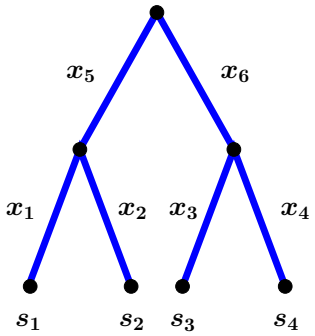
Subject to

$$\begin{cases} x_1 + x_2 \geq d_{12} \\ x_1 + x_3 + x_4 \geq d_{13} \\ x_1 + x_3 + x_5 \geq d_{14} \\ x_2 + x_3 + x_4 \geq d_{23} \\ x_2 + x_3 + x_5 \geq d_{24} \\ x_4 + x_5 \geq d_{34} \end{cases}$$

Determination of edge weights

Minimize $x_1 + x_2 + x_3 + x_4 + x_5 + x_6$

Subject to



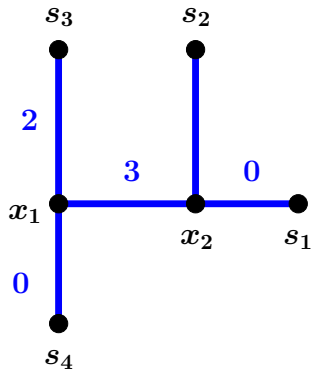
Rooted Tree

$$\begin{cases} x_1 + x_2 \geq d_{12} \\ x_1 + x_5 + x_6 + x_3 \geq d_{13} \\ x_1 + x_5 + x_6 + x_4 \geq d_{14} \\ x_2 + x_5 + x_6 + x_3 \geq d_{23} \\ x_2 + x_5 + x_6 + x_4 \geq d_{24} \\ x_3 + x_4 \geq d_{34} \\ x_5 + x_1 = x_5 + x_2 = \\ x_6 + x_3 = x_6 + x_4 \end{cases}$$

Unrooted minisize: approximation

- How to design a 2-approximation algorithm for the unrooted minisize evolution tree problem?

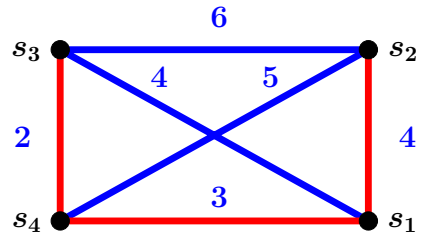
$d(\cdot,\cdot)$	s_1	s_2	s_3	s_4
s_1	0	4	4	3
s_2		0	6	5
s_3			0	2
s_4				0



Unrooted minisize: approximation

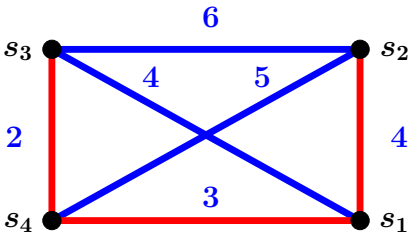
- Construct a minimal spanning tree out of the distance matrix

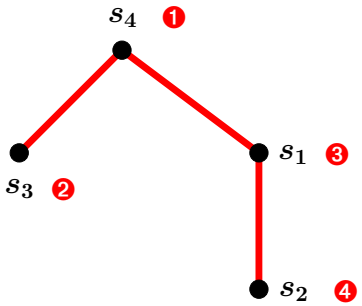
$d(\cdot,\cdot)$	s_1	s_2	s_3	s_4
s_1	0	4	4	3
s_2		0	6	5
s_3			0	2
s_4				0



Unrooted minisize: approximation

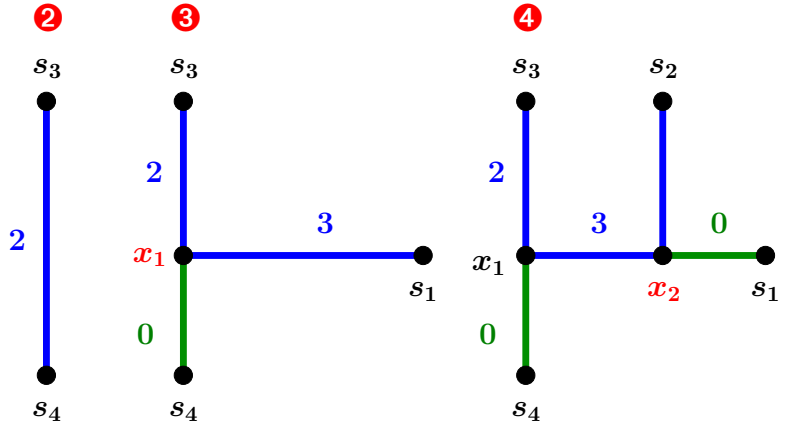
- Conduct a bradth first search on the minimal spaning tree by choosing an arbitrary node as root
  - Start from the root and visit all of the first level descendants first, then the second level descendants, and so on.





Unrooted minisize: approximation

- Transform the minimal spanning tree into an evolutionary tree by adding nodes one by one



Unrooted minisize: approximation

- The created tree above is indeed an evolution tree:
  - Each species is at leaf.
  - The degree of each internal node is three.
  - For any two species s_i and s_j, we have

$$d_T(s_i, s_j) \geq d_M(s_i, s_j)$$

because the distance $d_T(s_i, s_j)$ between any two species s_i and s_j on the evolution tree is exactly the same as that on the minimal spanning tree, which is then $\geq d_M(s_i, s_j)$ by triangular inequality.

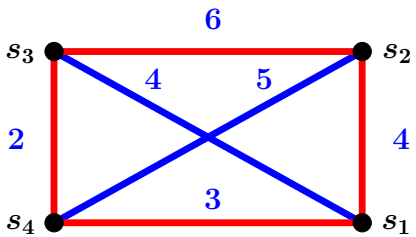
Unrooted minisize: approximation

- How to prove $APP < 2 \cdot |OPT|$?
  - APP: the length of our approximate solution
  - OPT: an optimal solution

1. $APP = |MST| < |TSP|$
   - MST: the mininal spanning tree of distance matrix
   - TSP: the optimal solution of the travelling salesperson problem for distance matrix
2. $|TSP| \leq 2 \cdot |OPT|$

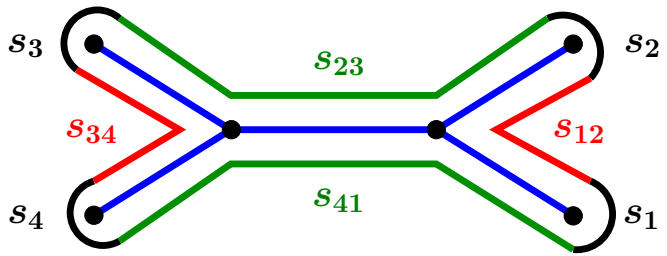
Unrooted minisize: approximation

- $APP = |MST| < |TSP|$
  - By construction, $APP = |MST|$
  - TSP: find a Hamiltonian cycle (visiting all of the nodes exactly once) with minimum length
  - Deleting any edge from TSP obtains a spanning tree



Unrooted minisize: approximation

- $|TSP| \leq 2 \cdot |OPT|$
  - ET: an Euler tour (visiting all of edges exactly once) of OPT ($|ET| = 2 \cdot |OPT|$)
  - CET: the cycle of species corresponding to ET ($|TSP \leq |CET|$ $\because CET$ = H. C.)
  - $|CET| \leq |ET|$ since $d_T(s_i, s_j) \geq d(s_i, s_j)$



UPGMA: create a rooted tree ❶

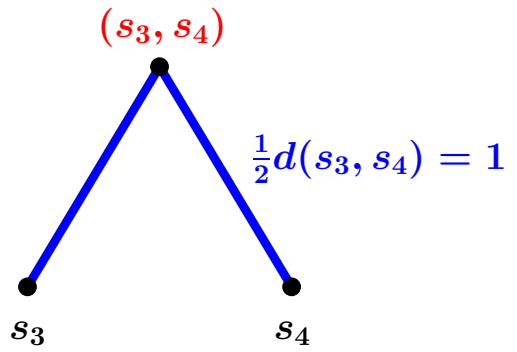
- UPGMA: Unweighted Pair Group Method with Arithmetic Mean [SS63]
- **Example**: consider the distance matrix as follows

	s_1	s_2	s_3	s_4
s_1	0	4	4	3
s_2		0	6	5
s_3			0	**2**
s_4				0

Output: a rooted evolutionary tree

UPGMA: create a rooted tree ❷

1. Select the pair of species (s_3, s_4) with the smallest distance and construct a rooted tree with s_3 and s_4 as leaf nodes



UPGMA: create a rooted tree ❸

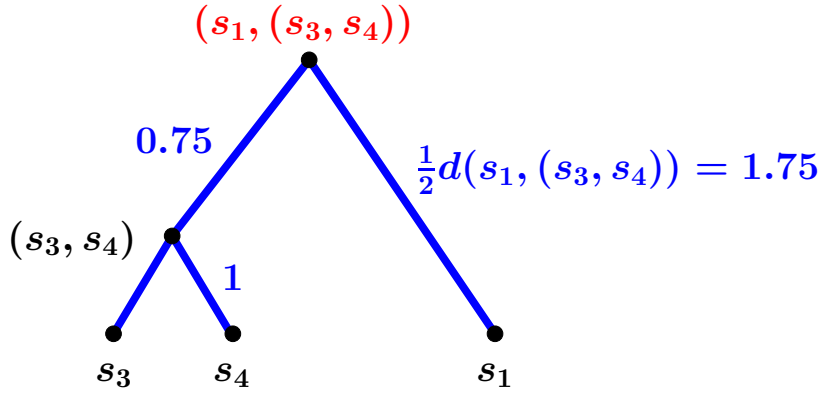
2. Consider (s_3, s_4) as a new species and the distances are updated as follows:

$$\begin{aligned} d(s_1, (s_3, s_4)) &= \tfrac{1}{2}(d(s_1, s_3) + d(s_1, s_4)) \\ &= \tfrac{1}{2}(4 + 3) = 3.5 \\ d(s_2, (s_3, s_4)) &= \tfrac{1}{2}(d(s_2, s_3) + d(s_2, s_4)) \\ &= \tfrac{1}{2}(6 + 5) = 5.5 \end{aligned}$$

	s_1	s_2	(s_3, s_4)
s_1	0	4	**3.5**
s_2		0	5.5
(s_3, s_4)			0

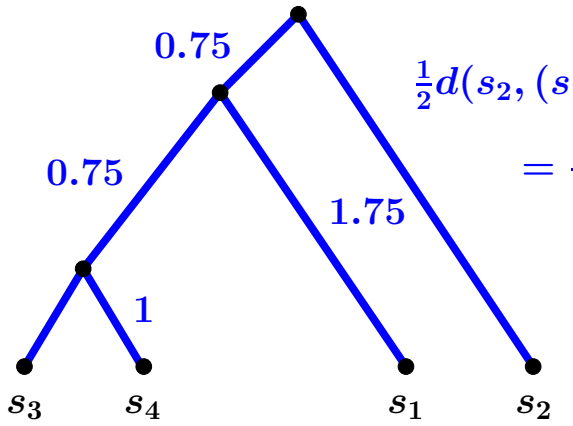
UPGMA: create a rooted tree ❹

3. Select the pair of species $(s_1, (s_3, s_4))$ with the smallest distance and construct a rooted as follows:



UPGMA: create a rooted tree ❺

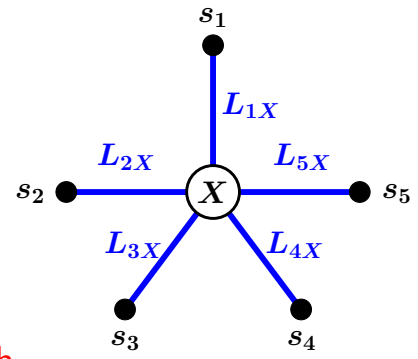
4. Since s_2 is the only specie left, the final tree will look like as follows:



$$\frac{1}{2}d(s_2,(s_1,(s_3,s_4))) = \frac{\frac{4+5+6}{3}}{2} = 2.5$$

Neighbor joining: unrooted tree ❶

- Construct a starlike tree as follows:

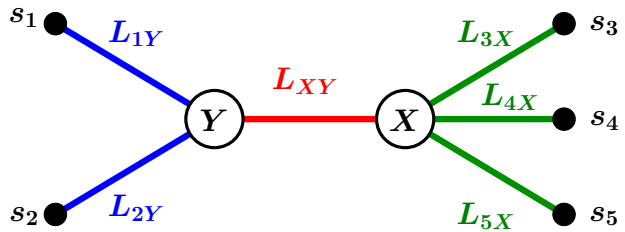


Tree length

$$S_0 = \sum_{i=1}^{n} L_{iX} = \frac{1}{n-1} \sum_{1 \le i < j \le n} d(s_i, s_j)$$

Neighbor joining: unrooted tree ❷

- Consider the tree $\mathcal{T}_{12}$ as follows:

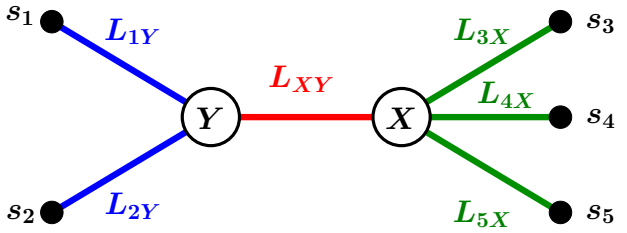


$$L_{XY} = \frac{1}{2(n-2)} \left[\sum_{k=3}^{n} (d(s_1,s_k) + d(s_2,s_k)) - (n-2)(L_{1Y}+L_{2Y}) - 2\sum_{i=3}^{n} L_{iX} \right]$$

Neighbor joining: unrooted tree ❸

- The tree length S_{12} of $\mathcal{T}_{12}$ is:

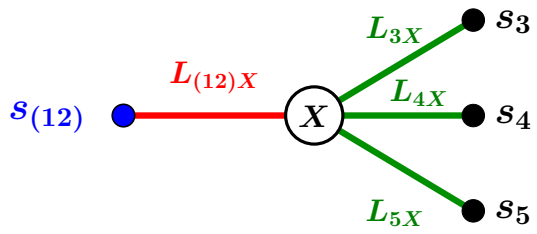
$$\begin{aligned} S_{12} &= L_{XY} + (L_{1Y} + L_{2Y}) + \sum_{i=3}^{n} L_{iX} \\ &= \frac{1}{2(n-2)} \sum_{k=3}^{n} (d(s_1,s_k) + d(s_2,s_k)) \\ &\quad + \frac{1}{2} d(s_1,s_2) + \frac{1}{n-2} \sum_{3 \le i < j \le n} d(s_i,s_j) \end{aligned}$$



Neighbor joining: unrooted tree ❹

- Consider (s_1, s_2) as a new speices $s_{(12)}$ and the distances are updated as follows: for $3 \le i \le n$

$$d(s_{(12)}, s_i) = \frac{d(s_1, s_i) + d(s_2, s_i)}{2}$$



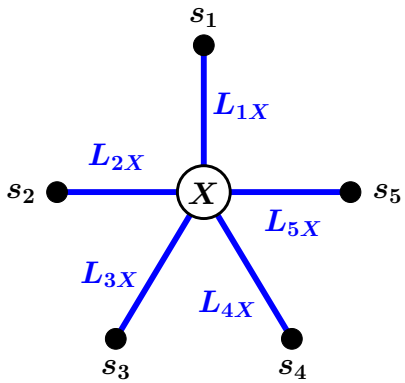
Neighbor joining: algorithm ❺

1. For all pairs of s_i and s_j, we compute the tree size S_{ij} of tree $\mathcal{T}_{ij}$. ($\frac{n(n-1)}{2}$ such trees)
2. Choose the pair with the smallest tree size. Suppose S_{12} is such a pair. Then consider (s_1, s_2) as a new speices $s_{(12)}$ and the distances are updated as follows: $d(s_{(12)}, s_i) = \frac{d(s_1,s_i)+d(s_2,s_i)}{2}$ for $3 \le i \le n$
3. The number of species is reduced by one and for the new distance matrix, the above procedure is again applied to find the next pair of neighbors until the number of species becomes 3.

Neighbor joining: example ❻

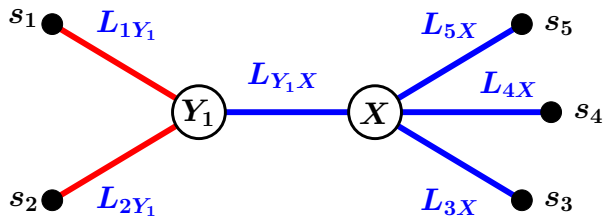
- **Example**: consider the distance matrix as follows

$d(s_i, s_j)$	s_2	s_3	s_4	s_5
s_1	7	8	11	10
s_2		5	8	7
s_3			5	4
s_4				5



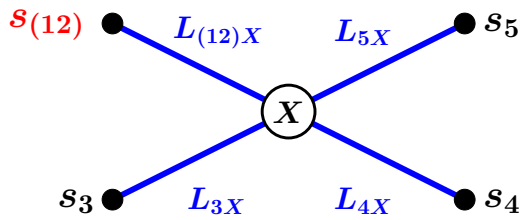
Neighbor joining: example ❼

S_{ij}	s_2	s_3	s_4	s_5
s_1	13	17.7	18	18
s_2		17.7	18	18
s_3				17.3
s_4				16.7



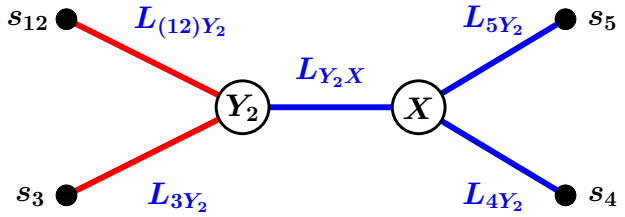
Neighbor joining: example ❽

$d(s_i, s_j)$	s_3	s_4	s_5
$s_{(12)}$	$\frac{8+5}{2} = 6.5$	$\frac{11+8}{2} = 9.5$	$\frac{10+7}{2} = 8.5$
s_3		5	4
s_4			5



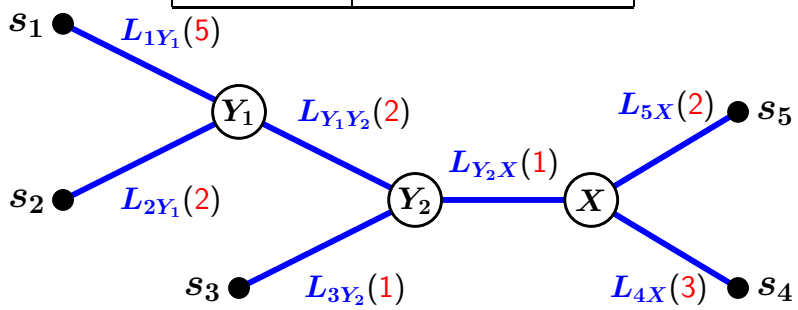
Neighbor joining: example ❾

S_{ij}	s_3	s_4	s_5
s_{12}	12.5	> 12.5	> 12.5
s_3			> 12.5
s_4			16.7



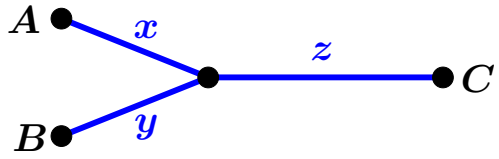
Neighbor joining: example ❿

$d(s_i, s_j)$	s_2	s_3	s_4	s_5
s_1	7	8	11	10
s_2		5	8	7
s_3			5	4
s_4				5



Estimate edge lengths

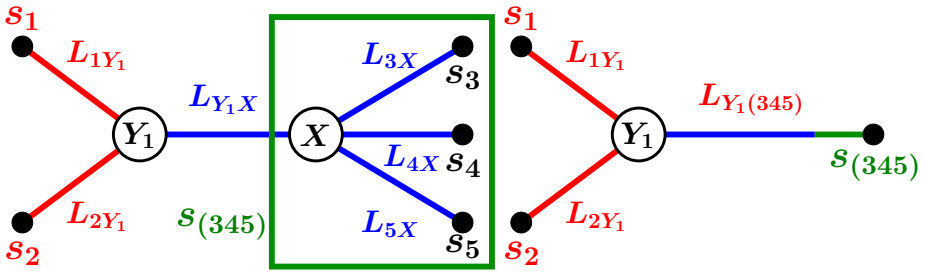
- Fitch and Margoliash's (1967) mehtod:



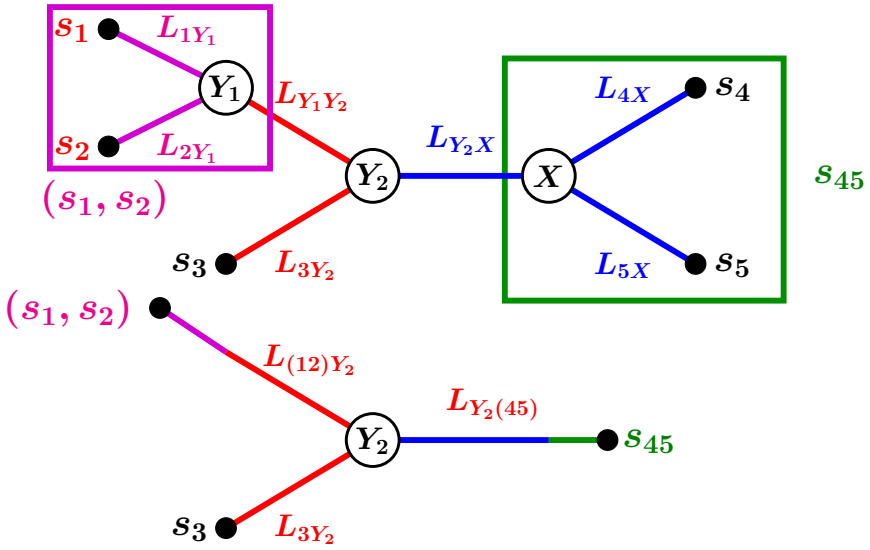
  - $d(A,B) = x + y$, $d(A,C) = x + z$, $d(B,C) = y + z$
  - $x = \frac{d(A,B)+d(A,C)-d(B,C)}{2}$
    $y = \frac{d(B,A)+d(B,C)-d(A,C)}{2}$
    $z = \frac{d(C,A)+d(C,B)-d(A,B)}{2}$

Estimate edge lengths

- $L_{1Y_1} = \frac{d(s_1,s_2)+d(s_1,s_{(345)})-d(s_2,s_{(345)})}{2}$
  - $d(s_1, s_{(345)}) = \frac{d(s_1,s_3)+d(s_1,s_4)+d(s_1,s_5)}{3}$
  - $d(s_2, s_{(345)}) = \frac{d(s_2,s_3)+d(s_2,s_4)+d(s_2,s_5)}{3}$
- $L_{2Y_1} = \frac{d(s_2,s_1)+d(s_2,s_{(345)})-d(s_1,s_{(345)})}{2}$

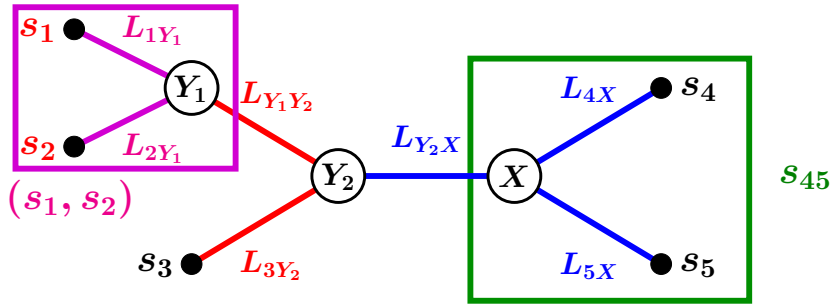


Estimate edge lengths



Estimate edge lengths

- Compute $L_{(12)Y_2}$ and L_{3Y_2} using the same way.



- $\because L_{(12)Y_2} = \frac{L_{1Y_1}+L_{Y_1Y_2}+L_{2Y_1}+L_{Y_1Y_2}}{2}$
  $= L_{Y_1Y_2} + \frac{L_{1Y_1}+L_{2Y_1}}{2} = L_{Y_1Y_2} + \frac{d(s_1,s_2)}{2}$
  $\therefore L_{Y_1Y_2} = L_{(12)Y_2} - \frac{d(s_1,s_2)}{2}$

References

[BFW92] H. L. Bodlaender, M. R. Fellows, and T. Warnow. Two strikes against perfect phylogeny. In Werner Kuich, editor, *Proceedings of the 19th International Colloquium Automata, Languages and Programming*, volume 623 of *Lecture Notes in Computer Science*, pages 273–283. Springer-Verlag, 1992.

[Day87] W. H. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49:461–467, 1987.

[DJS86] W. E. Day, D. S. Johnson, and D. Sankoff. The computational complexity of inferring rooted phylogenies by parsimony. *Mathematical Biosciences*, 83:33–42, 1986.

[DS92] S. Dress and M. Steel. Convex tree realizations of partitions. *Appl. Math. Lett.*, 5(3):3–6, 1992.

[Fit71] W. M. Fitch. Toward defining the course of evolution: minimum change for a specified tree topology. *Systematic Zoology*, 20:406–416, 1971.

[FKW93] M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees (ex-

In *Proceedings of the 6th Annual Symposium on Discrete Algorithms*, pages 595–603. ACM Press, 1995.

[SN87] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[SS63] R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy*. Freeman, San Francisco (CA), USA, 1963.

[Ste92] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992.

[WSSB77] M. S. Waterman, T. F. Smith, M. Singh, and W. A. Beyer. Additive evolutionary trees. *J. Theoret. Biol.*, 64:199–213, 1977.

tended abstract). In *Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 137–145, 1993.

[Gus91] D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21:19–28, 1991.

[Har73] J. A. Hartigan. Minimum mutation fits to a given tree. *Biometrics*, 29:53–65, 1973.

[HP82] M. D. Hendy and D. Penny. Branch and bound algorithms to determine minimal evolutionary trees. *Mathematical Biosciences*, 60:133–142, 1982.

[KLW90] S. Kannan, E. Lawler, and T. Warnow. Determining the evolutionary tree. In David Johnson, editor, *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '90)*, pages 475–484. SIAM, 1990.

[KW94] S. K. Kannan and T. J. Warnow. Inferring evolutionary history from DNA sequences. *SIAM Journal on Computing*, 23(4):713–737, 1994.

[KW95] S. Kannan and T. Warnow. A fast algorithm for the computation and enumeration of perfect phylogenies when the number of character states is fixed.