

# Lecture 09: Phylogenetic Analyses (II)

CSCI-478/CSCI-578/BIOL-510 Bioinformatics, Fall 2021



Hua Wang, Ph.D.

Department of Computer Science  
Colorado School of Mines

September 28, 2021

# The UPGMA method — an example

---



# The UPGMA method

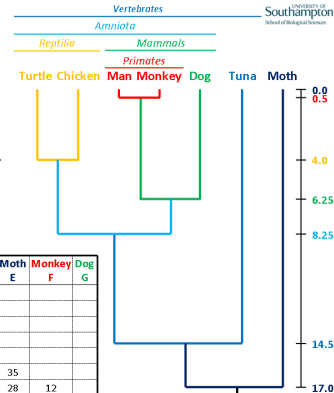
- A critical assumption of the UPGMA approach is that the rate of nucleotide or amino acid substitution is **constant** for **all the branches in the tree**, that is, the molecular clock applies to **all evolutionary lineages**.
- If this assumption is true, **branch lengths** can be used to estimate the dates of divergence, and the sequence-based tree mimics a species tree.
  - Edge lengths correspond to times measured by a clock.
  - Divergences of (sequences) assumed to occur at constant rate at all points in the tree.
  - Sum of times down a path to the leaves from any node is the same, whatever choice of path.

# The UPGMA method

- A critical assumption of the UPGMA approach is that the rate of nucleotide or amino acid substitution is **constant** for **all the branches in the tree**, that is, **the molecular clock** applies to **all evolutionary lineages**.
- If this assumption is true, **branch lengths** can be used to estimate the dates of divergence, and the sequence-based tree mimics a species tree.
  - Edge lengths correspond to times measured by a clock.
  - Divergences of (sequences) assumed to occur at constant rate at all points in the tree.
  - Sum of times down a path to the leaves from any node is the same, whatever choice of path.

The UPGMA tree based on this Cytochrome C data supports the known evolutionary relationships of these organisms.

	Turtle A	Man B	Tuna C	Chicken D	Moth E	Monkey F	Dog G
Turtle							
Man	19						
Tuna	27	31					
Chicken	8	18	26				
Moth	33	36	41	31			
Monkey	18	1	32	17	35		
Dog	13	13	29	14	28	12	





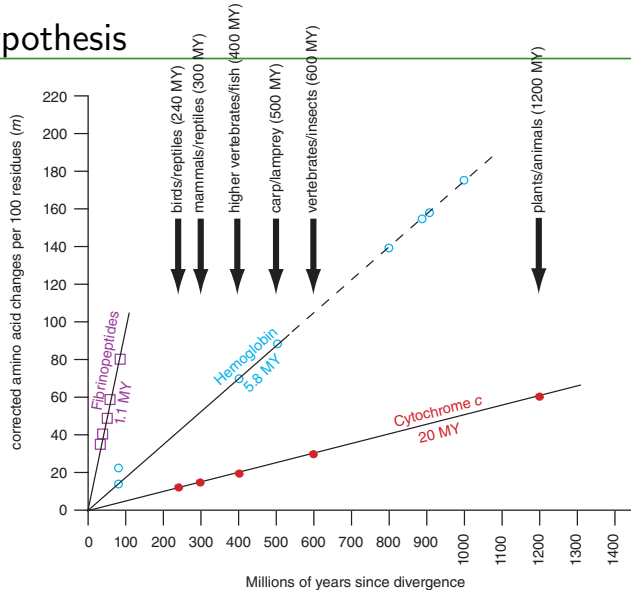
# The molecular clock hypothesis

- In the 1960s, sequence data were accumulated for small, abundant proteins such as globins, cytochromes c, and fibrinopeptides.
  - Some proteins appeared to evolve slowly, while others evolved rapidly.
- Linus Pauling, *et al.* proposed the hypothesis of a molecular clock:

## The molecular clock hypothesis

For every given protein, the rate of molecular evolution is approximately **constant** in all evolutionary lineages.

# The molecular clock hypothesis



## The molecular clock hypothesis and ultrametric data

If the molecular clock assumption holds, then the data is said to be ultrametric:

### The ultrametric tree

The strong triangle inequality:

$$d(x, z) \leq \max(d(x, y), d(y, z)) \quad .$$

which implies that for any triplet of sequences  $i, j, k$ , the distances are either

- **all equal**, or
- **two are equal and one is smaller**.

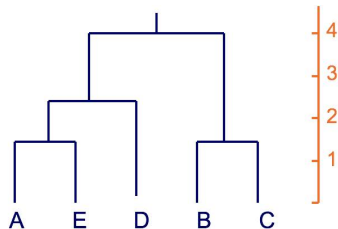
That is, (1) branch length represent time, and (2) the length from the root to the leaves are the same.

# The molecular clock hypothesis and ultrametric data

If the molecular clock assumption holds, then the data is said to be ultrametric:

## The ultrametric tree

	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



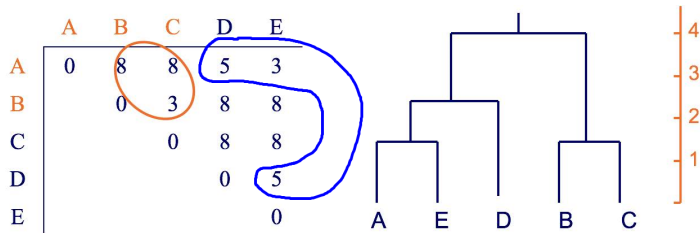
That is, (1) branch length represent time, and (2) the length from the root to the leaves are the same.



# The molecular clock hypothesis and ultrametric data

If the molecular clock assumption holds, then the data is said to be ultrametric:

## The ultrametric tree



That is, (1) branch length represent time, and (2) the length from the root to the leaves are the same.



# The molecular clock hypothesis

---

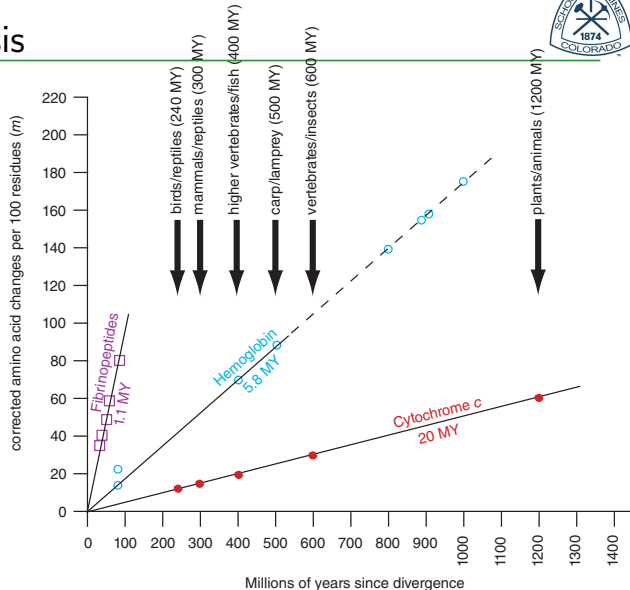
Yet, the molecular clock assumption  
does **not** always hold ...

# The molecular clock hypothesis

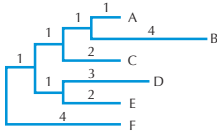
Yet, the molecular clock assumption does **not** always hold ...

Selection pressures, thereby evolution speed, vary across

- time periods,
- organisms,
- genes within an organism,
- regions within a gene
- ...



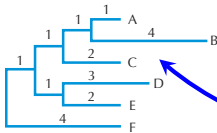
# The molecular clock hypothesis — when it does **NOT** hold



Distance matrix

	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

# The molecular clock hypothesis — when it does **NOT** hold



Distance matrix

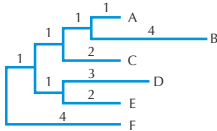
	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8




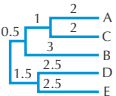
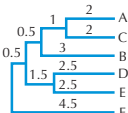
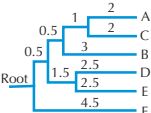
The data is **NOT** ultrametric, because

$$7 > \max(5, 4) = 5 ,$$

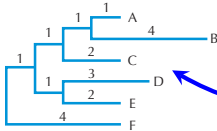
$$9 > \max(8, 5) = 8 .$$

# The molecular clock hypothesis — when it does **NOT** hold




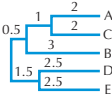
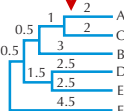
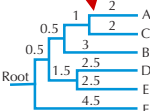


	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6
Distance matrix	<div> <div>A B C D E</div> <div> <div>B 5</div> <div>C 4 7</div> <div>D 7 10 7</div> <div>E 6 9 6 5</div> <div>F 8 11 8 9 8</div> </div> </div>	<div> <div>AC B D E</div> <div> <div>B 4</div> <div>D 7 10</div> <div>E 6 9 5</div> <div>F 8 11 8 9</div> </div> </div>	<div> <div>AC B DE</div> <div> <div>B 4</div> <div>DE 6.5 9.5</div> <div>F 8 11 8.5</div> </div> </div>	<div> <div>ACB DE</div> <div> <div>DE 8</div> <div>F 9.5 9.5</div> </div> </div>	<div> <div>ABCDE</div> <div> <div>F 9</div> </div> </div>	No new matrix
Identify smallest D	$A \leftrightarrow C = 4$	$D \leftrightarrow E = 5$	$AC \leftrightarrow B = 4$	$ACB \leftrightarrow DF = 8$	$ABCDE \leftrightarrow F = 9$	
Taxa joined	A and C	D and E	A and C with B	ABC with DE	ABCDE with F	
Subtree						

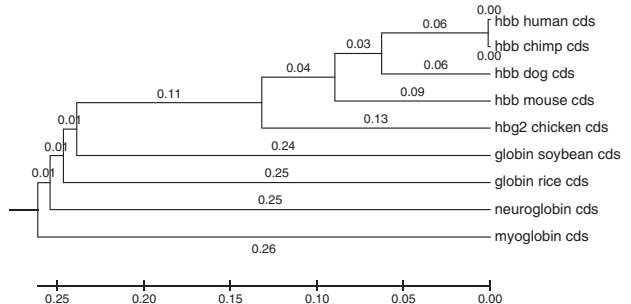
# The molecular clock hypothesis — when it does **NOT** hold



Note how this is **not** the same as the starting tree.

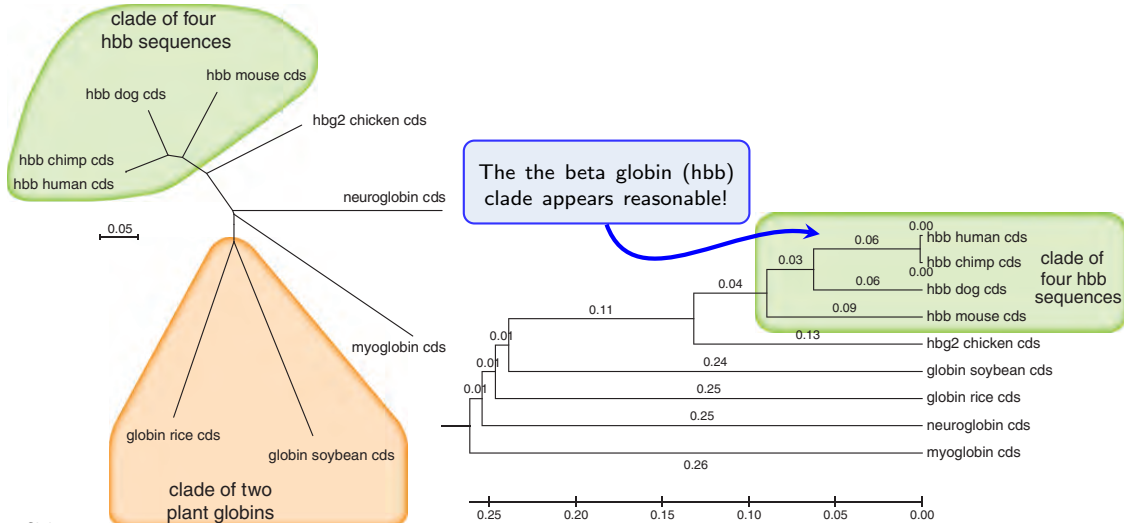
	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6																																																																																										
Distance matrix	<table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th></tr><tr><th>B</th><td>5</td><td></td><td></td><td></td><td></td></tr><tr><th>C</th><td>4</td><td>7</td><td></td><td></td><td></td></tr><tr><th>D</th><td>7</td><td>10</td><td>7</td><td></td><td></td></tr><tr><th>E</th><td>6</td><td>9</td><td>6</td><td>5</td><td></td></tr><tr><th>F</th><td>8</td><td>11</td><td>8</td><td>9</td><td>8</td></tr></table>		A	B	C	D	E	B	5					C	4	7				D	7	10	7			E	6	9	6	5		F	8	11	8	9	8	<table><tr><th></th><th>AC</th><th>B</th><th>D</th><th>E</th></tr><tr><th>B</th><td>4</td><td></td><td></td><td></td></tr><tr><th>D</th><td>7</td><td>10</td><td></td><td></td></tr><tr><th>E</th><td>6</td><td>9</td><td>5</td><td></td></tr><tr><th>F</th><td>8</td><td>11</td><td>8</td><td>9</td></tr></table>		AC	B	D	E	B	4				D	7	10			E	6	9	5		F	8	11	8	9	<table><tr><th></th><th>AC</th><th>B</th><th>DE</th></tr><tr><th>B</th><td>4</td><td></td><td></td></tr><tr><th>DE</th><td>6.5</td><td>9.5</td><td></td></tr><tr><th>F</th><td>8</td><td>11</td><td>8.5</td></tr></table>		AC	B	DE	B	4			DE	6.5	9.5		F	8	11	8.5	<table><tr><th></th><th>ACB</th><th>DE</th></tr><tr><th>DE</th><td>8</td><td></td></tr><tr><th>F</th><td>9.5</td><td>9.5</td></tr></table>		ACB	DE	DE	8		F	9.5	9.5	<table><tr><th></th><th>ABCDE</th></tr><tr><th>F</th><td>9</td></tr></table>		ABCDE	F	9	No new matrix
	A	B	C	D	E																																																																																											
B	5																																																																																															
C	4	7																																																																																														
D	7	10	7																																																																																													
E	6	9	6	5																																																																																												
F	8	11	8	9	8																																																																																											
	AC	B	D	E																																																																																												
B	4																																																																																															
D	7	10																																																																																														
E	6	9	5																																																																																													
F	8	11	8	9																																																																																												
	AC	B	DE																																																																																													
B	4																																																																																															
DE	6.5	9.5																																																																																														
F	8	11	8.5																																																																																													
	ACB	DE																																																																																														
DE	8																																																																																															
F	9.5	9.5																																																																																														
	ABCDE																																																																																															
F	9																																																																																															
Identify smallest D	$A \leftrightarrow C = 4$	$D \leftrightarrow E = 5$	$AC \leftrightarrow B = 4$	$ACB \leftrightarrow DE = 8$	$ABCDE \leftrightarrow F = 9$																																																																																											
Taxa joined	A and C	D and E	A and C with B	ABC with DE	ABCDE with F																																																																																											
Subtree																																																																																																

# The molecular clock hypothesis — when it does **NOT** hold

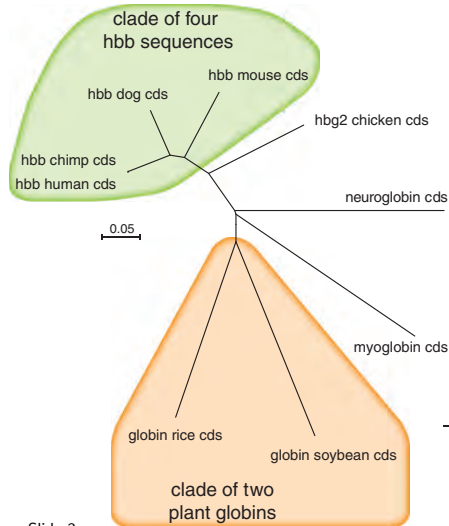




# The molecular clock hypothesis — when it does **NOT** hold

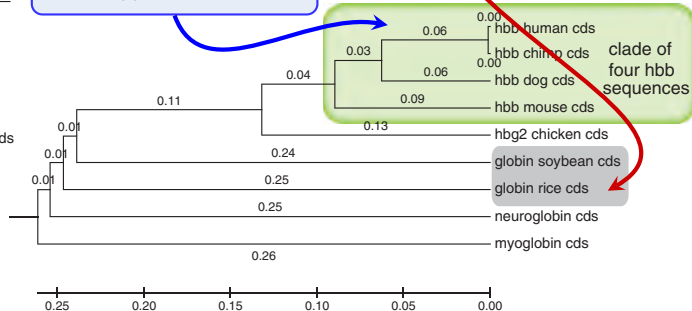


# The molecular clock hypothesis — when it does **NOT** hold



The two plant globins **unrealistically** share a common ancestor with vertebrate globins, but **not** than neuroglobin and myoglobin!!

The the beta globin (hbb) clade appears reasonable!





# Additivity

---

## The UPGMA algorithm

- assumes the constant molecular clock and ultrametric distances, and
- **implicitly** assumes additivity.



# Additivity

---

## The UPGMA algorithm

- assumes the constant molecular clock and ultrametric distances, and
- **implicitly** assumes additivity: the distance between any pair of leaves is the sum of the lengths of the edges on the path connecting them.



# Additivity

---

## The UPGMA algorithm

- assumes the constant molecular clock and ultrametric distances, and
- **implicitly** assumes additivity: the distance between any pair of leaves is the sum of the lengths of the edges on the path connecting them.

Can we design an algorithm that **only assumes the additivity?**

# Additivity and the neighbour-joining algorithm

## The UPGMA algorithm

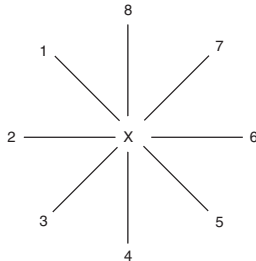
- assumes the constant molecular clock and ultrametric distances, and
- **implicitly** assumes additivity: the distance between any pair of leaves is the sum of the lengths of the edges on the path connecting them.

Can we design an algorithm that **only assumes the additivity?**

- **Yes!** We have the neighbour-joining algorithm.
- Like UPGMA, it constructs a tree by iteratively joining subtrees, but it is different in that:
  - How pair of subtrees to be merged is selected on each iteration
  - How distances are updated after each merge

# The neighbour-joining algorithm

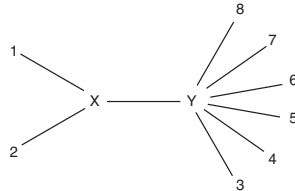
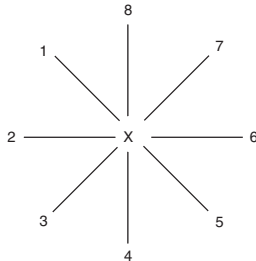
- **Initialization:** The sequences are first clustered in a starlike tree.
  - **Neighbors** are defined as OTUs that are connected by a single, interior node in an unrooted, multifurcating tree.
- **Repeat:** The two closest sequences are identified, such as sequences 1 and 2. These neighbors are connected to the other sequences via the internal branch XY.



But, how to identify the two closest sequences?

# The neighbour-joining algorithm

- **Initialization:** The sequences are first clustered in a starlike tree.
  - Neighbors are defined as OTUs that are connected by a **single, interior node** in an unrooted, multifurcating tree.
- **Repeat:** The two **closest** sequences are identified, such as sequences 1 and 2. These neighbors are connected to the other sequences via the internal branch XY.

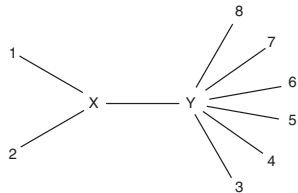
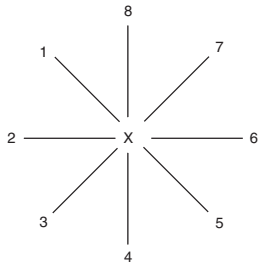


But, how to identify the two closest sequences?



# The neighbour-joining algorithm

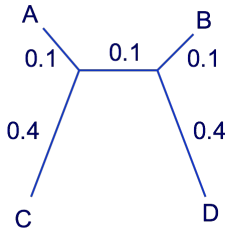
- **Initialization:** The sequences are first clustered in a starlike tree.
  - Neighbors are defined as OTUs that are connected by a **single, interior node** in an unrooted, multifurcating tree.
- **Repeat:** The two **closest** sequences are identified, such as sequences 1 and 2. These neighbors are connected to the other sequences via the internal branch XY.



**But, how to identify the two closest sequences?**

## Picking pairs of nodes to join in the neighbour-joining algorithm

- At each step, pick a pair of nodes to join; should we pick a pair with minimal  $d_{ij}$ ?
- Suppose that the real tree looks like this and we're picking the first pair of nodes to join. Shall we pick up  $A$  and  $B$ ? or shall we pick up  $A$  and  $C$ ?

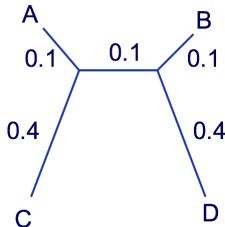


$$d_{AB} = 0.3$$

$$d_{AC} = 0.5$$

## Picking pairs of nodes to join in the neighbour-joining algorithm

- At each step, pick a pair of nodes to join; should we pick a pair with minimal  $d_{ij}$ ?
- Suppose that the real tree looks like this and we're picking the first pair of nodes to join. Shall we pick up  $A$  and  $B$ ? or shall we pick up  $A$  and  $C$ ?



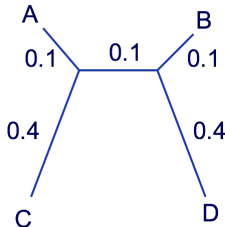
$$d_{AB} = 0.3$$

$$d_{AC} = 0.5$$

- **Wrong decision to join  $A$  and  $B$ !!**
  - $A$  and  $C$  are neighbors, but not  $A$  and  $B$ , because they are connected via **two** internal nodes.

## Picking pairs of nodes to join in the neighbour-joining algorithm

- At each step, pick a pair of nodes to join; should we pick a pair with minimal  $d_{ij}$ ?
- Suppose that the real tree looks like this and we're picking the first pair of nodes to join. Shall we pick up  $A$  and  $B$ ? or shall we pick up  $A$  and  $C$ ?



$$d_{AB} = 0.3$$

$$d_{AC} = 0.5$$

- **Wrong decision to join  $A$  and  $B$ !!**
  - $A$  and  $C$  are neighbors, but not  $A$  and  $B$ , because they are connected via **two** internal nodes.
- **We need to consider distance of pair to other leaves!**



## Picking pairs of nodes to join in the neighbour-joining algorithm

To avoid this, given  $N$  sequences, we pick a pair to join based on  $D_{ij}$  (Saitou & Nei '87, Studier & Kepper '88):

$$D_{ij} = d_{ij} - (S_i + S_j) \quad , \quad \text{where } S_i = \frac{1}{N-2} \sum_{k \neq i} d_{ik} \quad .$$

Selecting nodes  $i$  and  $j$  by smallest  $D_{ij}$ , we join them via a **new** internal node  $k$ .

## Updating distances in the neighbour-joining algorithm

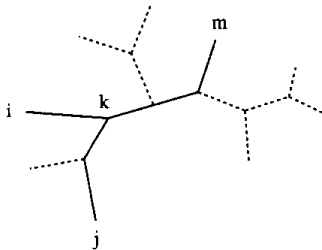
Given a new internal node  $k$ , the distance to another node  $m$  is given by:

$$d_{km} = \frac{1}{2} (d_{im} + d_{jm} - d_{ij}) \ .$$

## Updating distances in the neighbour-joining algorithm

Given a new internal node  $k$ , the distance to another node  $m$  is given by:

$$d_{km} = \frac{1}{2} (d_{im} + d_{jm} - d_{ij}) .$$



Using the additivity, we have:

$$\begin{cases} d_{im} = d_{ik} + d_{km} , \\ d_{jm} = d_{jk} + d_{km} , \\ d_{ij} = d_{ik} + d_{jk} , \end{cases}$$

by which we can derive that:

$$d_{km} = \frac{1}{2} (d_{im} + d_{jm} - d_{ij}) .$$

# The neighbour-joining algorithm

## The neighbour-joining algorithm

- Define the tree  $T$  = set of leaf nodes.
- Set  $L = T$
- While more than two subtrees in  $T$ :
  - Pick the pair  $i$  and  $j$  in  $L$  with minimal  $D_{ij}$ .
  - Add to  $T$  a new node  $k$  that joins  $i$  and  $j$
  - Determine new distances:

$$\begin{cases} d_{ik} &= \frac{1}{2} (d_{ij} + S_i - S_j) \quad , \quad \text{where } S_i = \frac{1}{N-2} \sum_{k \neq i} d_{ik} \quad , \\ d_{jk} &= d_{ij} - d_{ik} \quad , \\ d_{km} &= \frac{1}{2} (d_{im} + d_{jm} - d_{ij}) \quad . \end{cases}$$

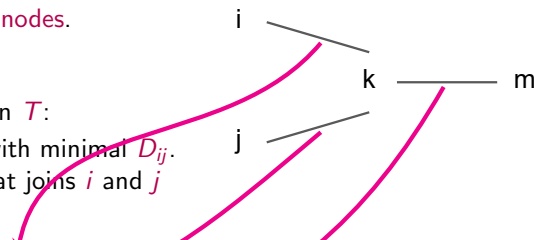
- Remove  $i$  and  $j$  from  $L$  and insert  $k$  (treat it like a leaf)



# The neighbour-joining algorithm

## The neighbour-joining algorithm

- Define the tree  $T = \text{set of leaf nodes}$ .
- Set  $L = T$
- While more than two subtrees in  $T$ :
  - Pick the pair  $i$  and  $j$  in  $L$  with minimal  $D_{ij}$ .
  - Add to  $T$  a new node  $k$  that joins  $i$  and  $j$
  - Determine new distances:

$$\begin{cases} d_{ik} = \frac{1}{2} (d_{ij} + S_i - S_j) \\ d_{jk} = d_{ij} - d_{ik} \\ d_{km} = \frac{1}{2} (d_{im} + d_{jm} - d_{ij}) \end{cases} \quad \text{where } S_i = \frac{1}{N-2} \sum_{k \neq i} d_{ik} ,$$


- Remove  $i$  and  $j$  from  $L$  and insert  $k$  (treat it like a leaf)

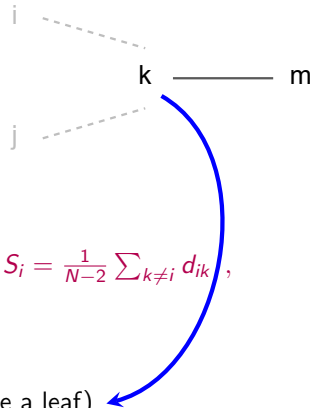
# The neighbour-joining algorithm

## The neighbour-joining algorithm

- Define the tree  $T = \text{set of leaf nodes}$ .
- Set  $L = T$
- While more than two subtrees in  $T$ :
  - Pick the pair  $i$  and  $j$  in  $L$  with minimal  $D_{ij}$ .
  - Add to  $T$  a new node  $k$  that joins  $i$  and  $j$
  - Determine new distances:

$$\begin{cases} d_{ik} &= \frac{1}{2} (d_{ij} + S_i - S_j) \text{ , where } S_i = \frac{1}{N-2} \sum_{k \neq i} d_{ik} \text{ ,} \\ d_{jk} &= d_{ij} - d_{ik} \text{ ,} \\ d_{km} &= \frac{1}{2} (d_{im} + d_{jm} - d_{ij}) \text{ .} \end{cases}$$

- Remove  $i$  and  $j$  from  $L$  and insert  $k$  (treat it like a leaf)



## Cycle 1

Distance matrix

	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

### Step 1

S calculations

$$S_A = (5+4+7+6+8)/4 = 7.5$$

$$S_B = (5+7+10+9+11)/4 = 10.5$$

$$S_C = (4+7+7+6+8)/4 = 8$$

$$S_D = (7+10+7+5+9)/4 = 9.5$$

$$S_E = (6+9+6+5+8)/4 = 8.5$$

$$S_F = (8+11+8+9+8)/4 = 11$$

### Step 2

Calculate pair with smallest ( $M$ ), where  $M_{ij} = D_{ij} - S_i - S_j$ .

Smallest are

$$M_{AB} = 5 - 7.5 - 10.5 = -13$$

$$M_{DE} = 5 - 9.5 - 8.5 = -13$$

Choose one of these (AB here).

### Step 3

Create a node (U) that joins pair with lowest  $M_{ij}$  such that  $S_U = D_{ij}/2 + (S_i - S_j)/2$ .

$U_1$  joins A and B:

$$S_{AU_1} = D_{AB}/2 + (S_A - S_B)/2 = 1$$

$$S_{BU_1} = D_{AB}/2 + (S_B - S_A)/2 = 4$$

## Cycle 2

	U <sub>1</sub>	C	D	E
C	3			
D	6	7		
E	5	6	5	
F	7	8	9	8

$$S_{U_1} = (3+6+5+7)/3 = 7$$

$$S_C = (3+7+6+8)/3 = 8$$

$$S_D = (6+7+5+9)/3 = 9$$

$$S_E = (5+6+5+8)/3 = 8$$

$$S_F = (7+8+9+8)/3 = 10.6$$

Smallest is

$$M_{CU_1} = 3 - 7 - 8 = -12$$

$$M_{DE} = 5 - 9 - 8 = -12$$

Choose one of these (DE here).

## Cycle 3

	U <sub>1</sub>	C	U <sub>2</sub>
C	3		
U <sub>2</sub>	3	4	
F	7	8	6

$$S_{U_1} = (3+3+7)/2 = 6.5$$

$$S_C = (3+7+6+8)/2 = 7.5$$

$$S_{U_2} = (3+4+6)/2 = 6.5$$

$$S_F = (7+8+6)/2 = 10.5$$

Smallest is

$$M_{CU_1} = 3 - 6.5 - 7.5 = -11$$

## Cycle 4

	U <sub>2</sub>	U <sub>3</sub>
U <sub>3</sub>	2	
F	6	6

$$S_{U_2} = (2+6)/1 = 8$$

$$S_{U_3} = (2+6)/1 = 8$$

$$S_F = (6+6)/1 = 12$$

Smallest is

$$M_{U_2F} = 6 - 8 - 12 = -14$$

$$M_{U_3F} = 6 - 8 - 12 = -14$$

$$M_{U_2U_3} = 2 - 8 - 8 = -14$$

Choose one of these ( $M_{U_2U_3}$  here).

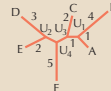
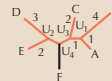
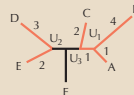
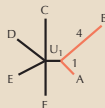
## Cycle 5

	U <sub>4</sub>
F	5

Because  $N - 2 = 0$ , we cannot do this calculation.

### Step 4

Join  $i$  and  $j$  according to S above and make all other taxa in form of a star. Branches in black are of unknown length. Branches in red are of known length.



### Step 5

Calculate new distance matrix of all other taxa to U with  $D_{AU} = D_{ix} + D_{jx} - D_{ij}$ , where  $i$  and  $j$  are those selected from above.

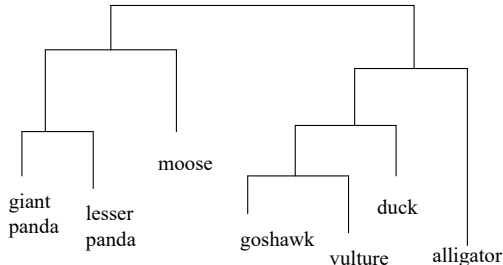
### Comments

Note this is the same tree we started with (drawn in unrooted form here).

# Rooting trees

Finding a root in an unrooted tree is sometimes accomplished by using an outgroup: a species known to be **more distantly** related to remaining species than they are to each other.

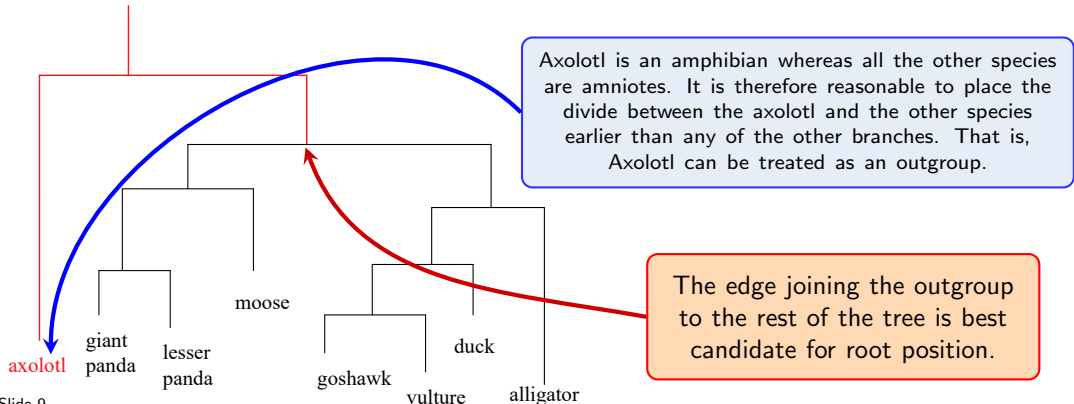
- In other words, finding a species that **diverged earlier** than the rest species in the tree.



# Rooting trees

Finding a root in an unrooted tree is sometimes accomplished by using an **outgroup**: a species known to be **more distantly** related to remaining species than they are to each other.

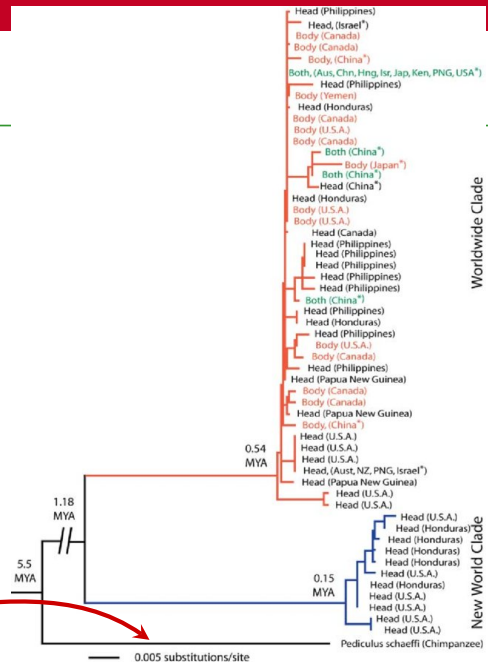
- In other words, finding a species that **diverged earlier** than the rest species in the tree.



# Rooting trees



Chimpanzee lice used as outgroup in human lice study.





## Distance-based methods — a summary

- 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 only** (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.
- Neighbor joining is commonly used!!



# Approaches to phylogenetic trees

---

Three general types of methods:

- Distance: find tree that accounts for estimated evolutionary distances
- Parsimony: find tree that requires minimum number of changes to explain the data
- Maximum likelihood: find tree that maximizes the likelihood of the data





# Approaches to phylogenetic trees

## Three general types of methods:

- Distance: find tree that accounts for estimated evolutionary distances
- Parsimony: find tree that requires minimum number of changes to explain the data
- Maximum likelihood: find tree that maximizes the likelihood of the data



# Building the phylogentic trees using parsimony — the motivations

## The motivations

- The main idea behind maximum parsimony is that the best tree is that with **the shortest branch lengths possible** (Czelusniak et al., 1990).
- According to maximum parsimony theory, **having fewer changes** to account for the way a group of sequences evolved is preferable to more complicated explanations of molecular evolution.



# Building the phylogentic trees using parsimony — the task

## Task definition

### Given:

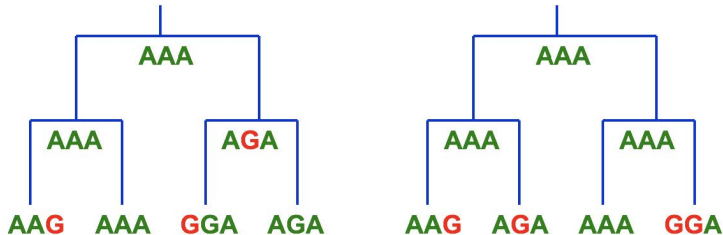
- Character-based data

### Do:

- Find the tree that explains the data with the minimal number of changes
- Focus on finding the right topology, not on estimating branch lengths

## Building the phylogentic trees using parsimony — the explanations

There are various trees that could explain the phylogeny of sequences **AAG**, **AAA**, **GGA**, **AGA** including these two:



Parsimony prefers the left tree because it requires fewer substitution events: 3 substitutions vs. 4 substitutions.

# Parsimony-Based Approaches

Usually these approaches involve two separate components:

- A procedure to find the minimum number of changes needed to explain the data (for a given tree topology)
- A search through the space of trees

## Basic assumptions

- Any state (e.g. nucleotide, amino acid, *etc.*) can convert to any other state
- The “costs” of these changes are uniform
- Positions are independent; we can compute the min number of changes for each position separately



## Copyright statement

This slide set is designed only for teaching CSCI-478/CSCI-578/BIOL-510 Bioinformatics at the Department of Computer Science of Colorado School of Mines in Fall 2021.

Some contents in this slide set are obtained from Internet and maybe copyright sensitive. Copyright and all rights therein are retained by the respective authors or by other copyright holders. Distributing or reposting the whole or part of this slide set not for academic purpose is **HIGHLY** prohibited, unless the explicit permissions from all copyright holders are granted.