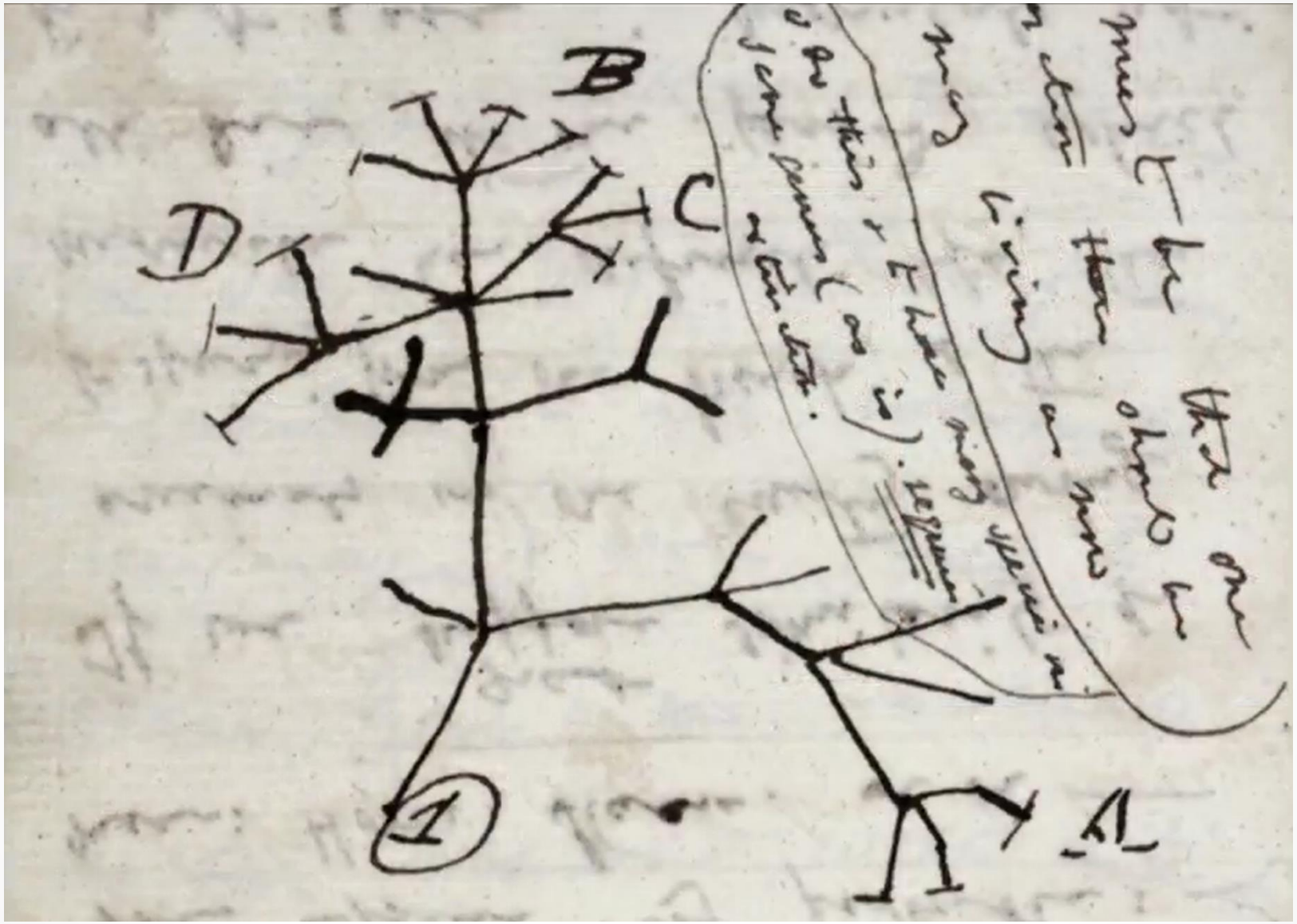
CLUSTERING METHODS AND ALGORITHMS IN GENOMICS AND EVOLUTION

Session 7

Distance based methods for tree inference

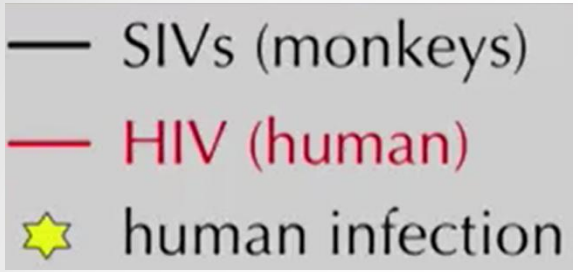
Outline

- Transforming Distance Matrices into Evolutionary Trees
- Toward an Algorithm for Distance-Based Phylogeny Construction
- Additive Phylogeny
- Ultrametric Evolutionary Trees (UPGMA reconstruction)
- The Neibour-Joining Algorithm
- Using Least-Squares to construct Distance-Based Phylogenies

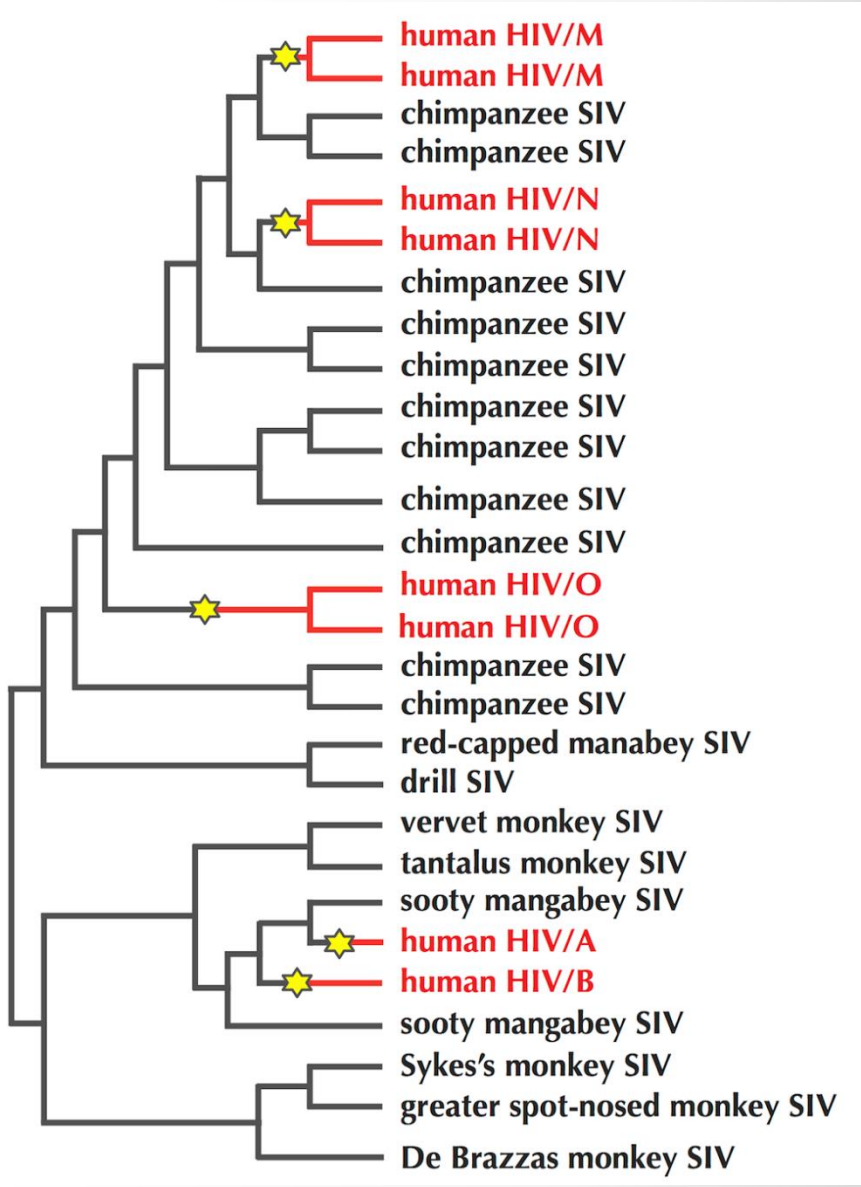


First evolutionary tree drawn by Charles Darwin in 1837.

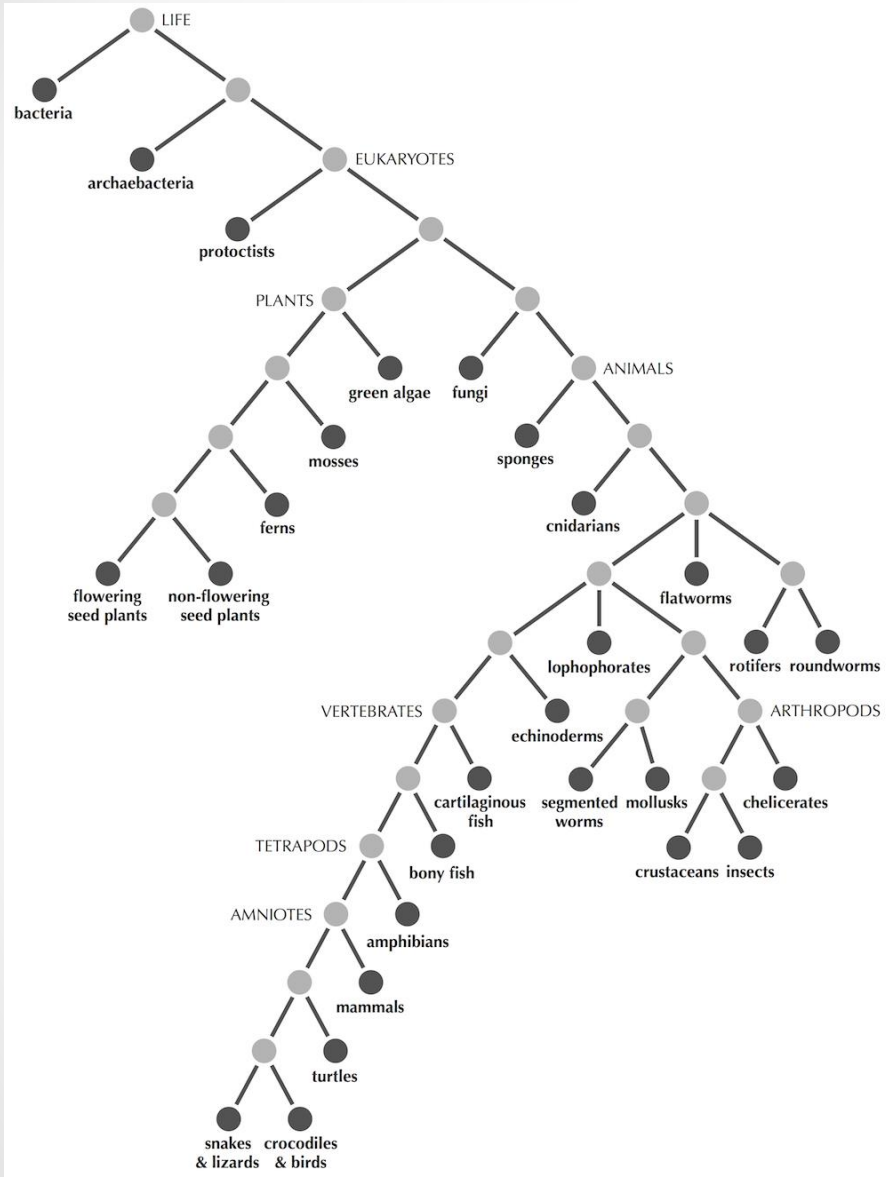
Example: HIV Evolutionary Tree



- HIV comprises five different viral families, denoted as A, B, M, N, and O, with the M family responsible for 95% of all HIV infections.
- The five families are different offshoots of the evolutionary tree for Simian Immunodeficiency Virus (SIV), which infects primates.
- Stars indicate viruses transitioning from primates to humans. The A and B families originated in sooty mangabey monkeys, whereas the M, N, and O families originated in chimpanzees.
- If we want to know which animal gave us SARS we should construct an evolutionary tree for the SARS coronovirus as well.



Trees

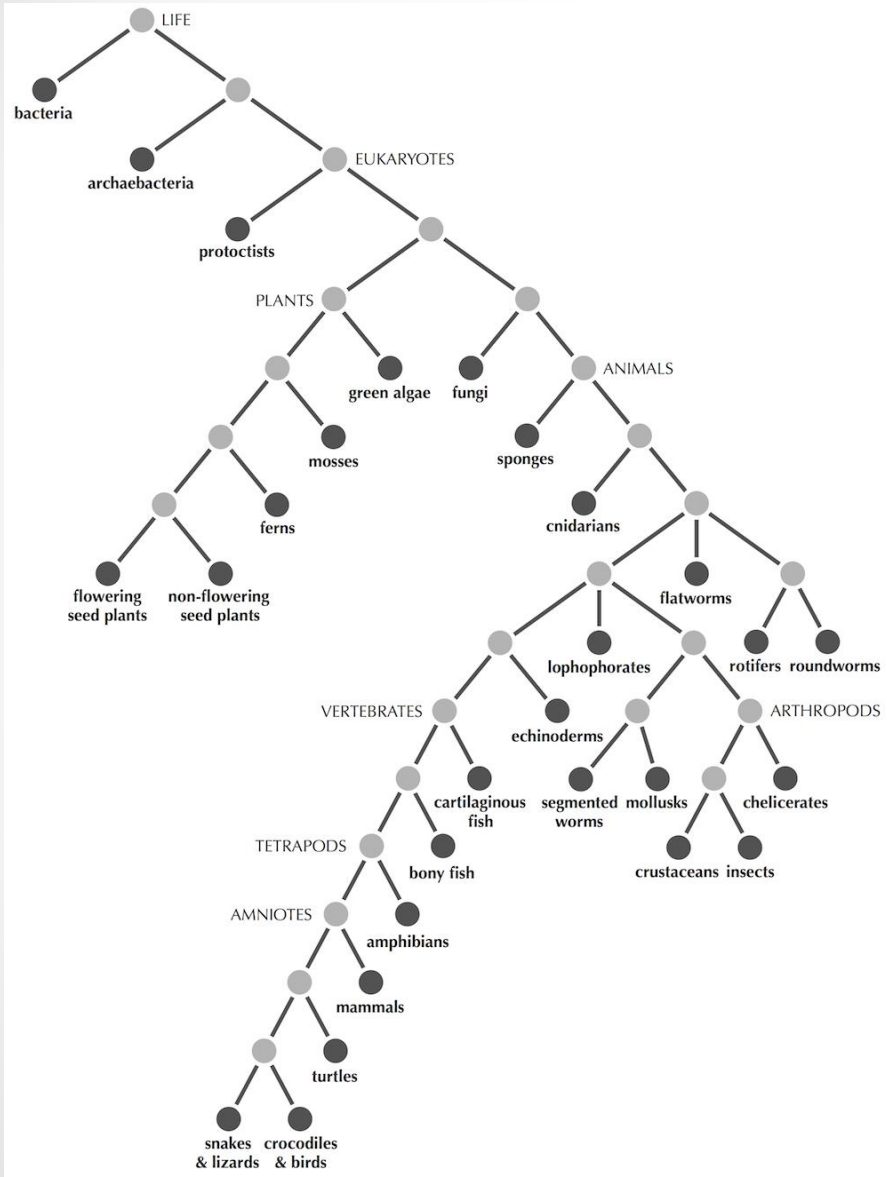


***Tree**: Connected graph containing no cycles.*

- **Connected:** the tree holds in one piece.
- **Acyclic:** the tree can branch out without growing back in on itself and forming a cycle.

A connected graph without cycles that models an evolutionary tree of life on Earth. Present-day species are shown as darker nodes.

Trees



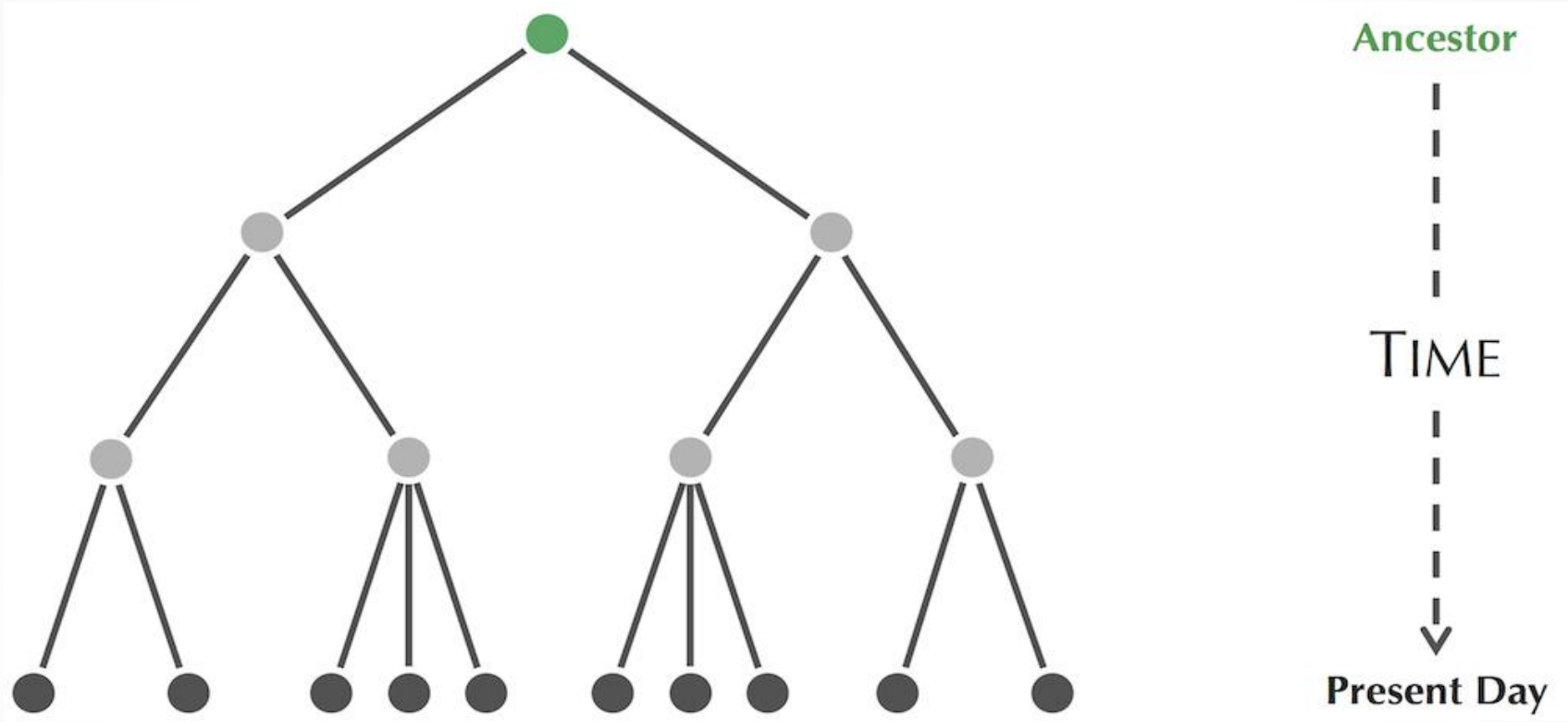
Tree: *Connected graph containing no cycles.*

Leaves (degree = 1): *present-day species.*

- Should be at the ending nodes of the tree.
- **Degree**- a number of edges connecting a node to other nodes.

A connected graph without cycles that models an evolutionary tree of life on Earth. Present-day species are shown as darker nodes.

Trees



A rooted tree, with the root (representing an ancestor of all species in the tree) indicated in green at the top of the tree. The presence of the root implies an orientation of edges in the tree away from the root such that time flows downward from the root to the leaves in the sense that each edge of the tree connects an older species to a more recent species.

Rooted tree: one node is designated as the **root** (most recent common ancestor)

Outline

- **Transforming Distance Matrices into Evolutionary Trees**
- Toward an Algorithm for Distance-Based Phylogeny Construction
- Additive Phylogeny
- Ultrametric Evolutionary Trees (UPGMA reconstruction)
- The Neibour-Joining Algorithm
- Using Least-Squares to construct Distance-Based Phylogenies

Constructing a Distance Matrix

Species	Alignment
Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT

A toy multiple alignment of hypothetical DNA sequences from four species…

Constructing a Distance Matrix

$D_{i,j}$ = number of differing symbols between *i*-th and *j*-th rows of a multiple alignment.

Species	Alignment	Distance Matrix (*D*)			
		Chimp	Human	Seal	Whale
Chimp	ACGTAGGCCT	0	3	6	4
Human	ATGTAAGACT	3	0	7	5
Seal	TCGAGAGCAC	6	7	0	2
Whale	TCGAAAGCAT	4	5	2	0

A multiple alignment of hypothetical DNA sequences from four species, along with the distance matrix produced by counting the number of differing symbols between each pair of rows in this multiple alignment.

Constructing a Distance Matrix

$D_{i,j}$ = number of differing symbols between *i*-th and *j*-th rows of a multiple alignment.

Species	Alignment	Distance Matrix (*D*)				
		Chimp	Human	Seal	Whale	
Chimp	ACGTAGGCCT	0	**3** ($D_{2,1}$)	6	4	$D_{1,2} = D_{2,1} = 3$
Human	ATGTAAGACT	**3** ($D_{1,2}$)	0	7	5	
Seal	TCGAGAGCAC	6	7	0	2	
Whale	TCGAAAGCAT	4	5	2	0	

A multiple alignment of hypothetical DNA sequences from four species, along with the distance matrix produced by counting the number of differing symbols between each pair of rows in this multiple alignment.

Constructing a Distance Matrix

$D_{i,j}$ = number of differing symbols between *i*-th and *j*-th rows of a multiple alignment.

Species	Alignment	Distance Matrix (*D*)			
		Chimp	Human	Seal	Whale
Chimp	ACGTAGGCCT	0	3	6	4
Human	ATGTAAGACT	3	0	7	5
Seal	TCGAGAGCAC	6	7	0	2
Whale	TCGAAAGCAT	4	5	2	0

Regardless of which distance function we use, in order to be a distance matrix, *D* must satisfy three properties. It must be **symmetric** (for all *i* and *j*, $D_{i,j} = D_{j,i}$), **non-negative** (for all *i* and *j*, $D_{i,j} \geq 0$) and satisfy the **triangle inequality** (for all *i*, *j*, and *k*, $D_{i,j} + D_{j,k} \geq D_{i,k}$), where k is any third species.

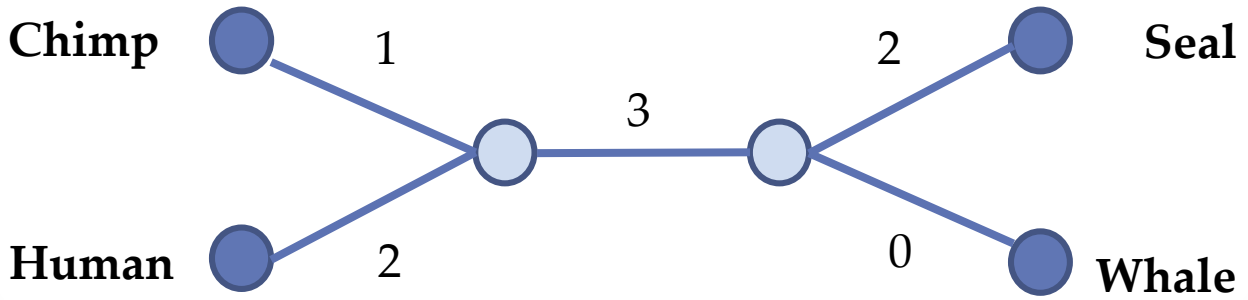
Fitting a Tree to a Matrix

	Chimp	**Human**	**Seal**	**Whale**
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

The toy distance matrix constructed from a multiple alignment.

Fitting a Tree to a Matrix

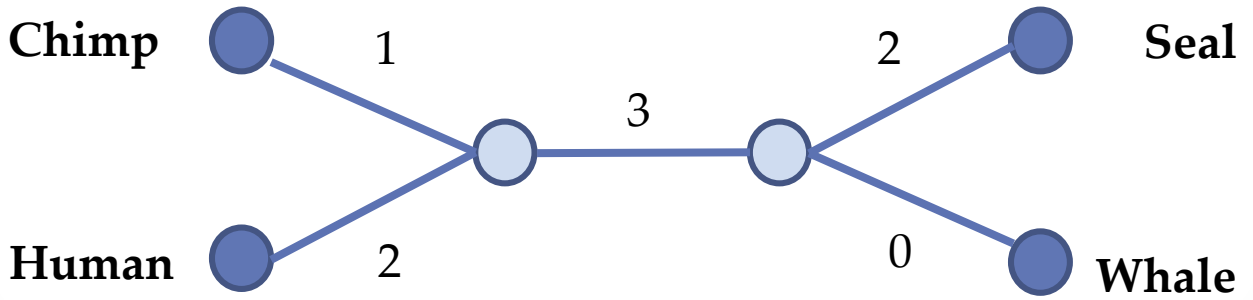
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



Unrooted tree fitting the distance matrix.

Fitting a Tree to a Matrix

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



Unrooted tree fitting the distance matrix.

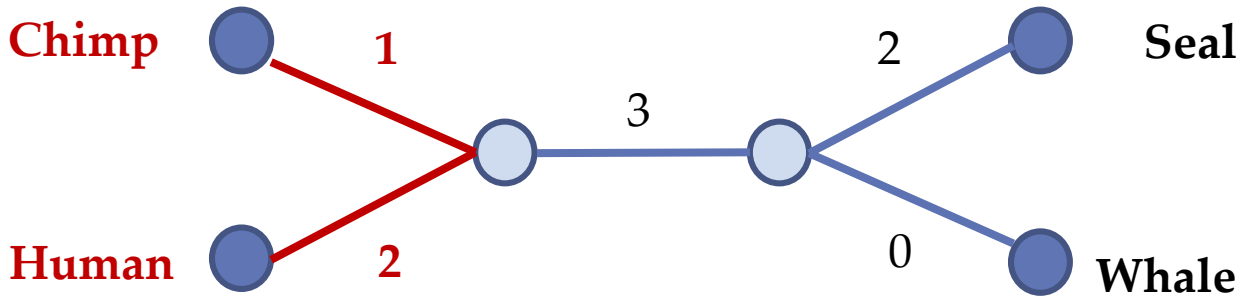
Deriving an unrooted tree from a distance matrix:

1) The leaves of this tree should correspond to the species represented by the matrix.

2) Assign each edge a non-negative length representing the evolutionary distance between the organisms that the edge connects.

Fitting a Tree to a Matrix

	Chimp	Human	Seal	Whale
Chimp	0	**3**	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



Unrooted tree fitting the distance matrix.

We will need to assign weights to the edges of this tree so that the sum of weights along a path that connects two leaves corresponds to the distance matrix value for those two leaves.

Distance-Based Phylogeny Problem

Distance-Based Phylogeny Problem: *Construct an evolutionary tree from a distance matrix.*

- **Input:** A distance matrix.
- **Output:** The unrooted tree "fitting" this distance matrix.

STOP and Think: Does the Distance-Based Phylogeny Problem always have a solution?

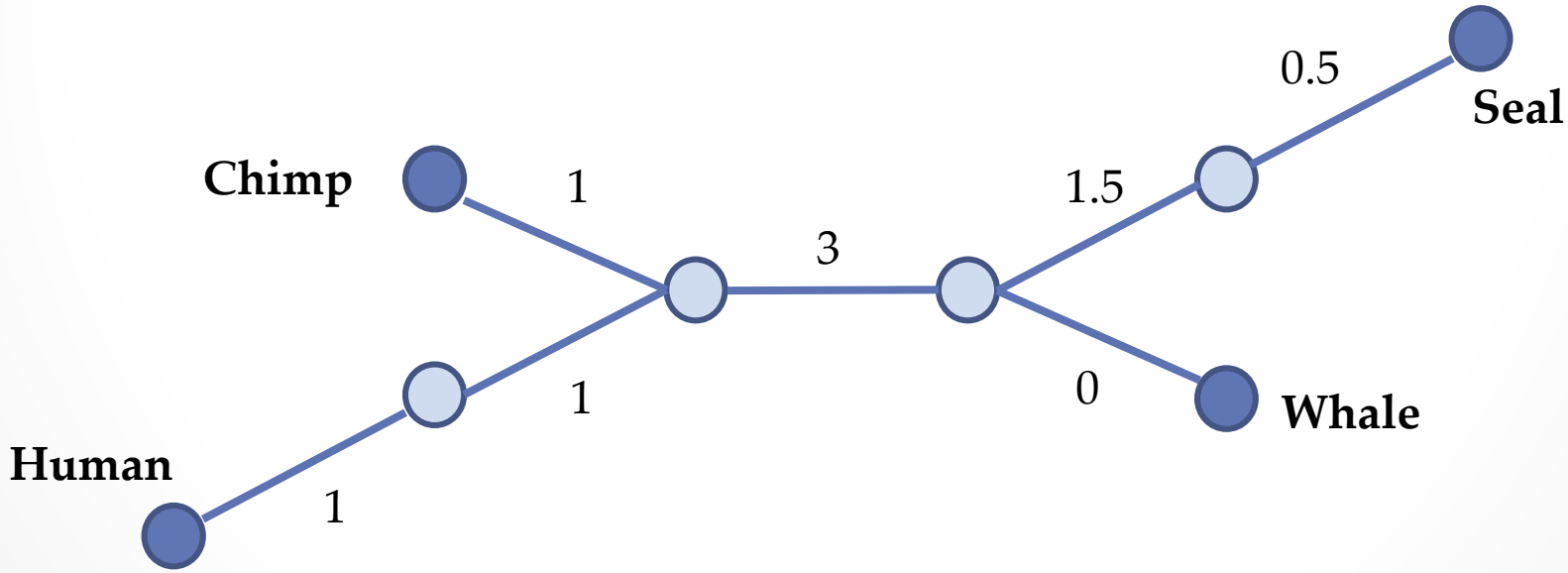
Return to Distance-Based Phylogeny

Exercise Break: Try fitting a tree to the following matrix.

	i	j	k	l
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0

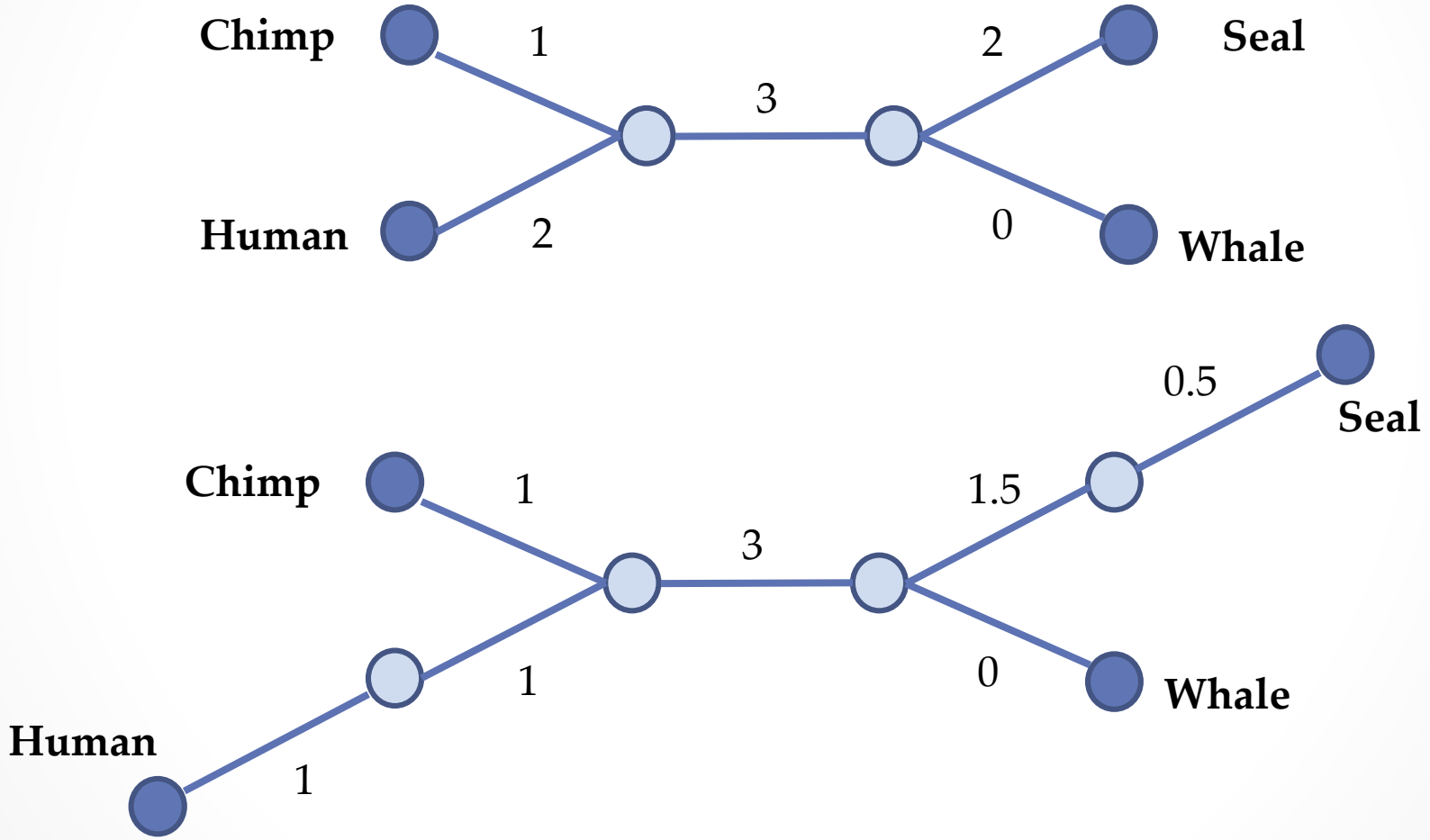
More than one Tree fits a matrix

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



We simply stretch out the edges of the tree we had before into longer paths and still have a tree that fits the distance matrix.

Which Tree is "Better"?

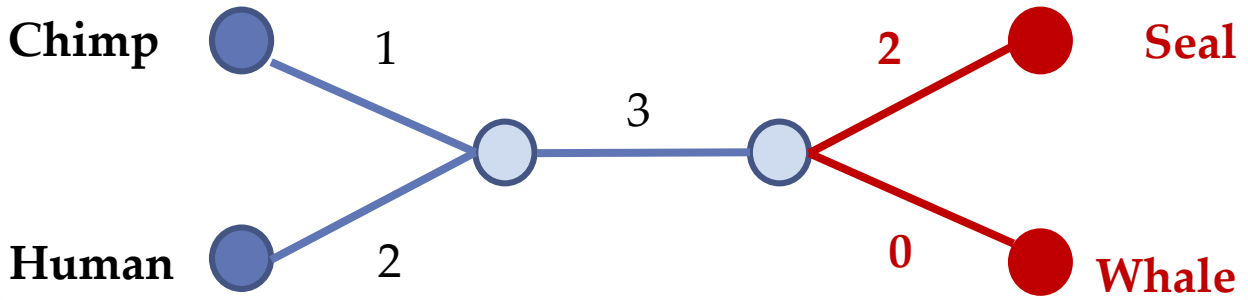


Outline

- Transforming Distance Matrices into Evolutionary Trees
- **Toward an Algorithm for Distance-Based Phylogeny Construction**
- Additive Phylogeny
- Ultrametric Evolutionary Trees (UPGMA reconstruction)
- The Neibour-Joining Algorithm
- Using Least-Squares to construct Distance-Based Phylogenies

An Idea of Distance-Based Phylogeny

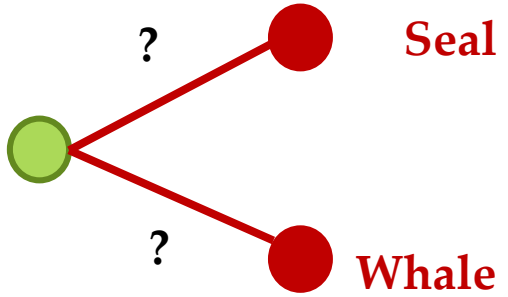
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	**2**
Whale	4	5	2	0



The minimum element of this matrix corresponds to two leaves that are next to each other on the tree.

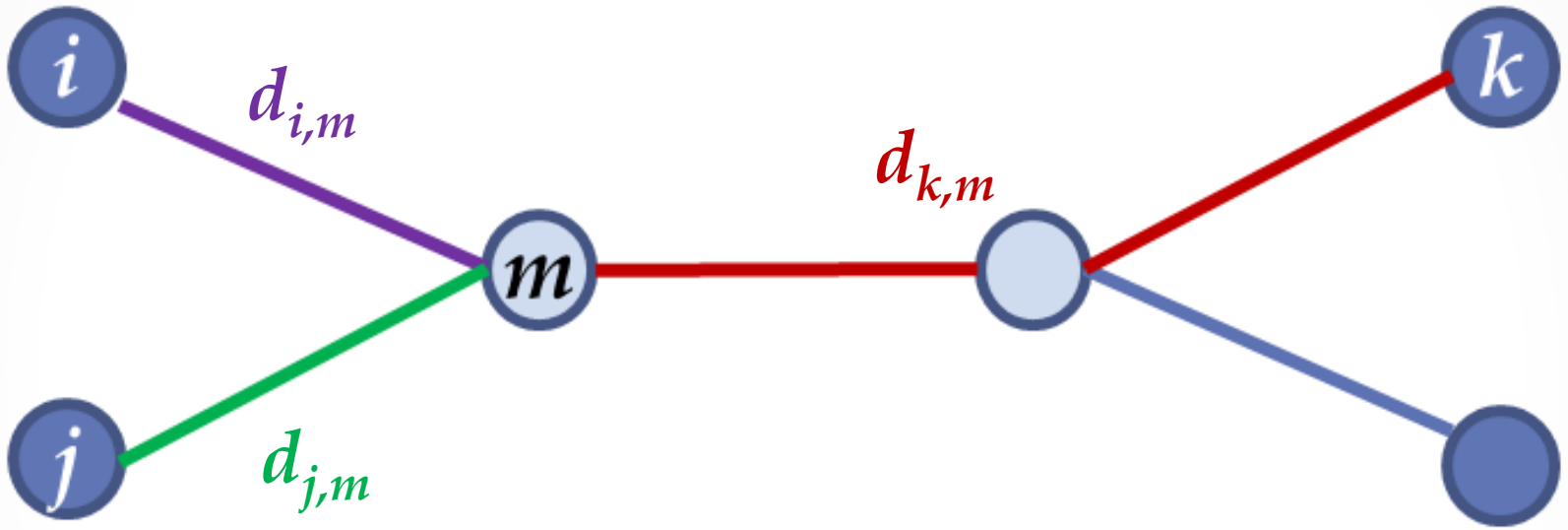
An Idea of Distance-Based Phylogeny

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	**2**
Whale	4	5	2	0



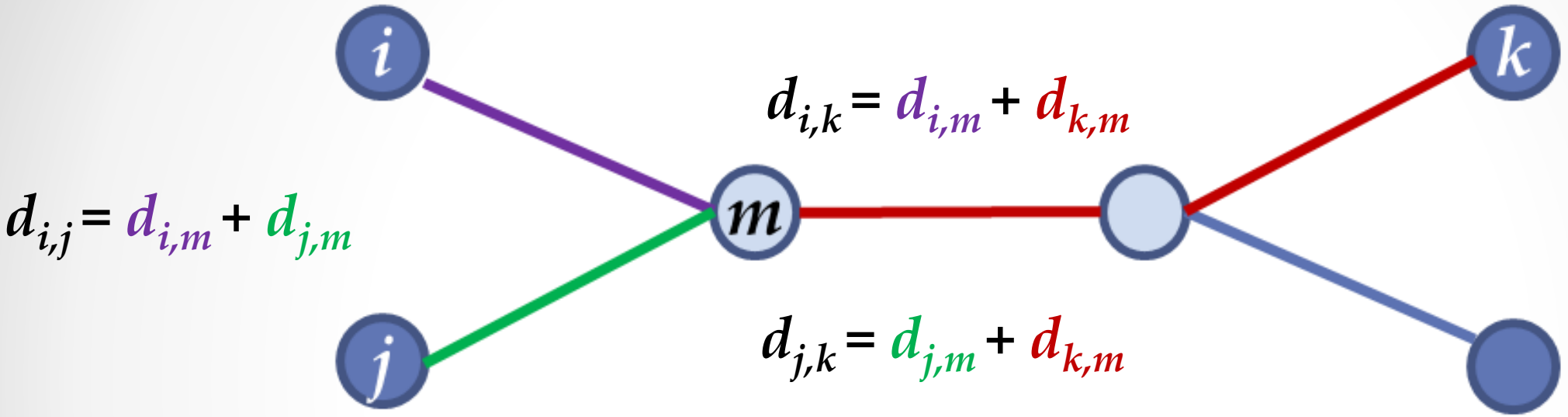
Let's pretend that we don't know the tree that fits the distance matrix, and see if we can use the fact, that seal and whale are neighbors in order to reconstruct the tree.

Toward a Recursive Algorithm

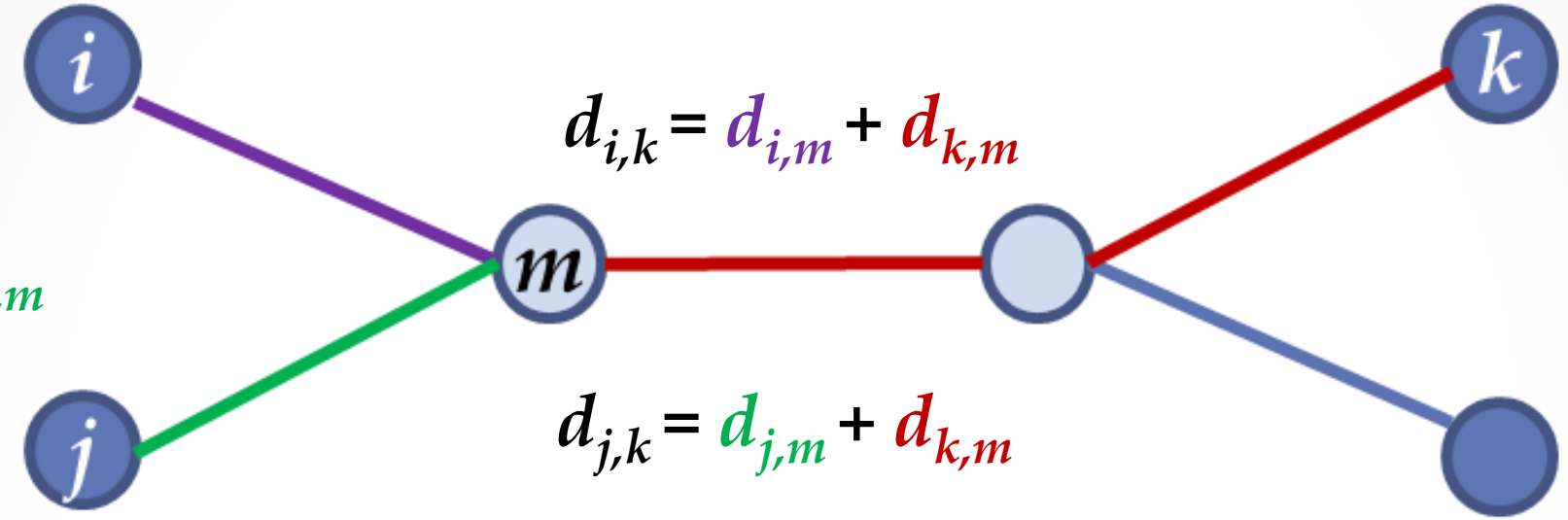


A tree with neighboring leaves ***i*** and ***j*** that share a parent ***m***. We try to reconstruct the ***green*** and ***purple*** distances. But if ***k*** is some other leaf in the tree, the ***red*** distance will help us out.

Toward a Recursive Algorithm



Toward a Recursive Algorithm



$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j})/ 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j})/ 2$$

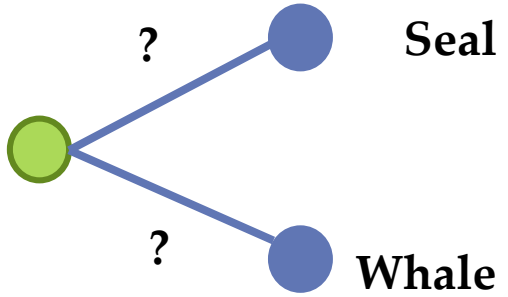
$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k})/ 2$$

$$d_{j,m} = (D_{j,k} + D_{i,j} - D_{i,k})/ 2$$

If it is known that *i* and *j* are neighbors, we can compute the distance from them to their parent, just from the distance matrix alone.

An Idea of Distance-Based Phylogeny

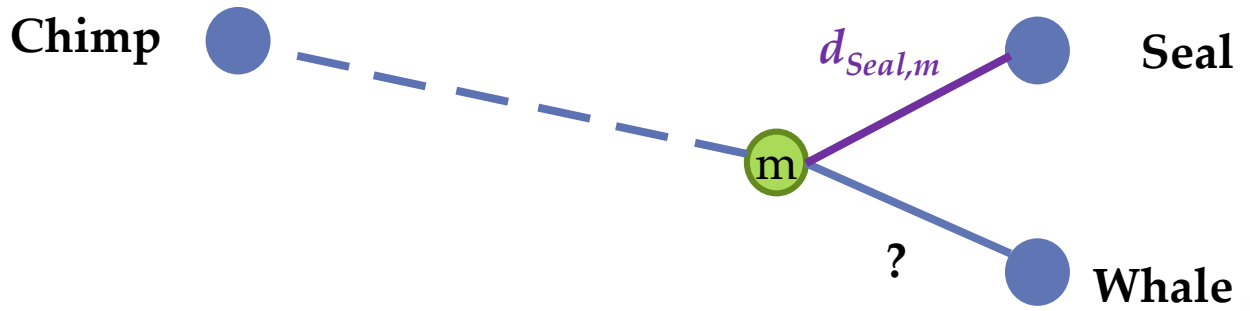
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	**2**
Whale	4	5	2	0



$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k})/\,2$$

An Idea of Distance-Based Phylogeny

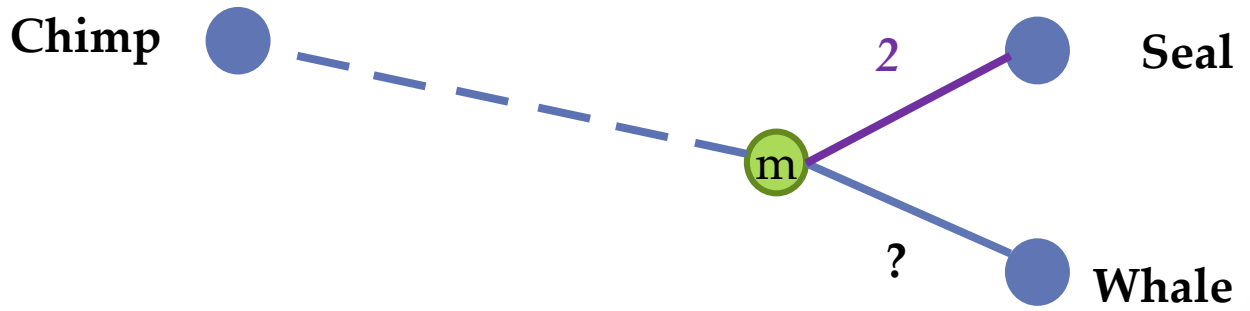
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



$$d_{Seal,m} = (D_{Seal,Chimp} + D_{Seal,Whale} - D_{Whale,Chimp})/ 2$$

An Idea of Distance-Based Phylogeny

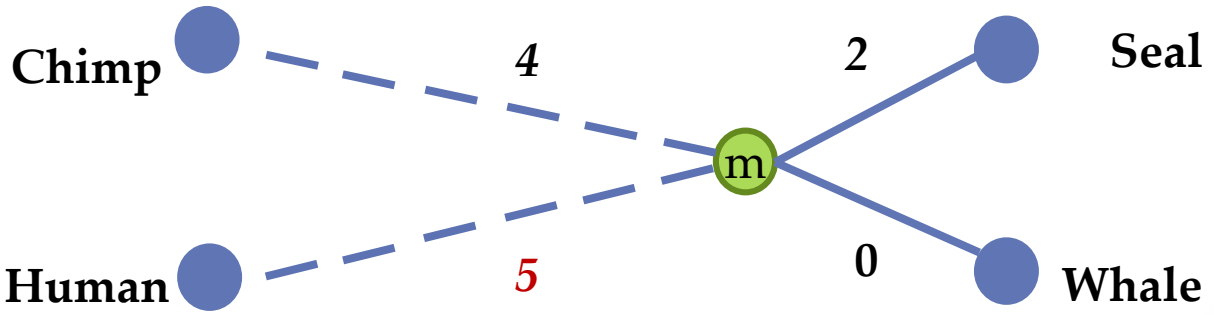
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



$d_{Seal,m} = 2$

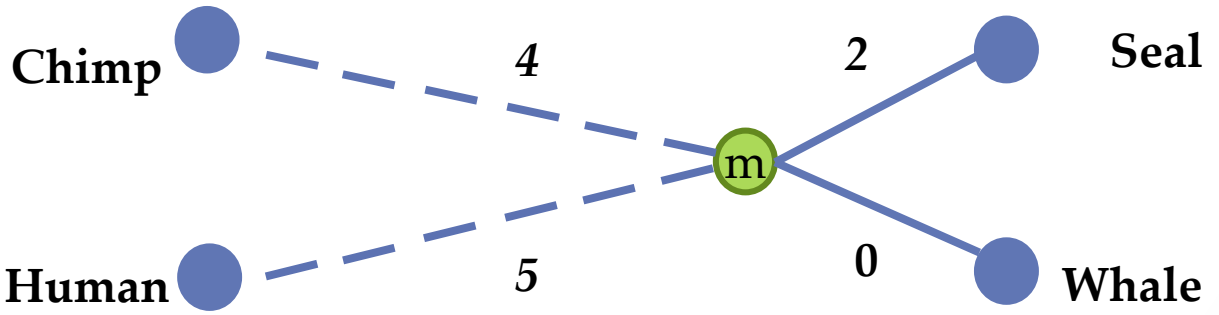
An Idea of Distance-Based Phylogeny

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	**7**	5
Seal	6	7	0	2
Whale	4	5	2	0



An Idea of Distance-Based Phylogeny

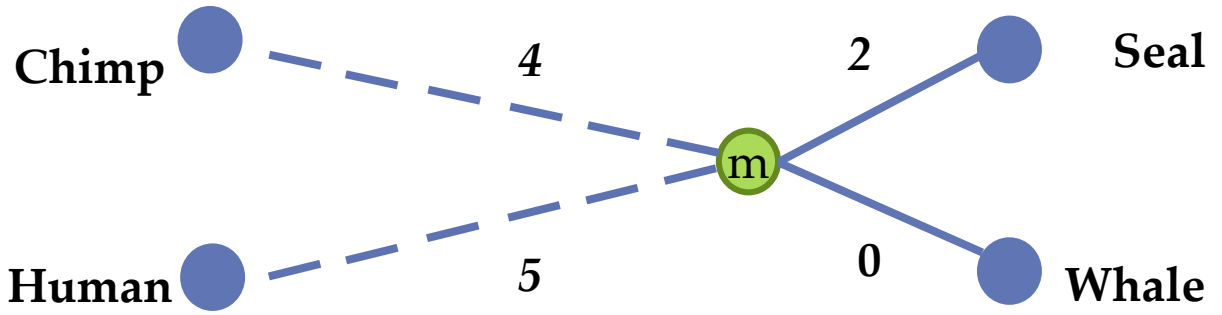
	Chimp	Human	Seal	Whale	m
Chimp	0	3	6	4	4
Human	3	0	7	5	5
Seal	6	7	0	2	2
Whale	4	5	2	0	0
m	4	5	2	0	0



An Idea of Distance-Based Phylogeny

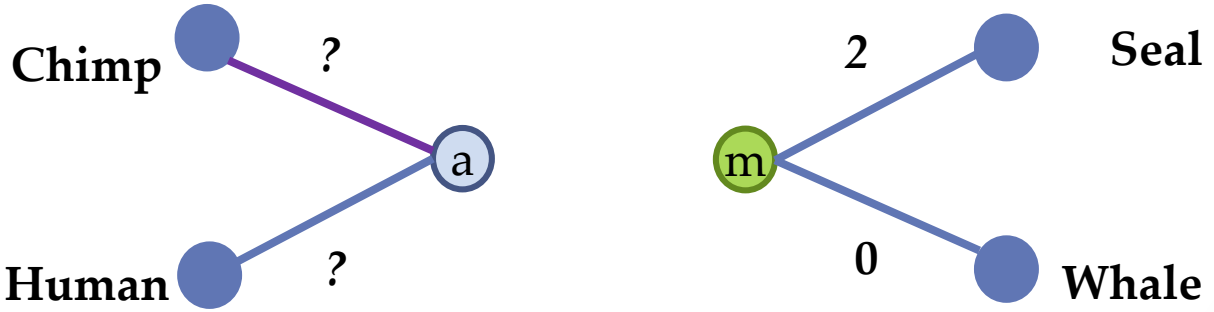
	Chimp	Human	m
Chimp	0	3	4
Human	3	0	5
m	4	5	0

Getting rid of Seal and Whale entirely yields a smaller 3x3 matrix.



An Idea of Distance-Based Phylogeny

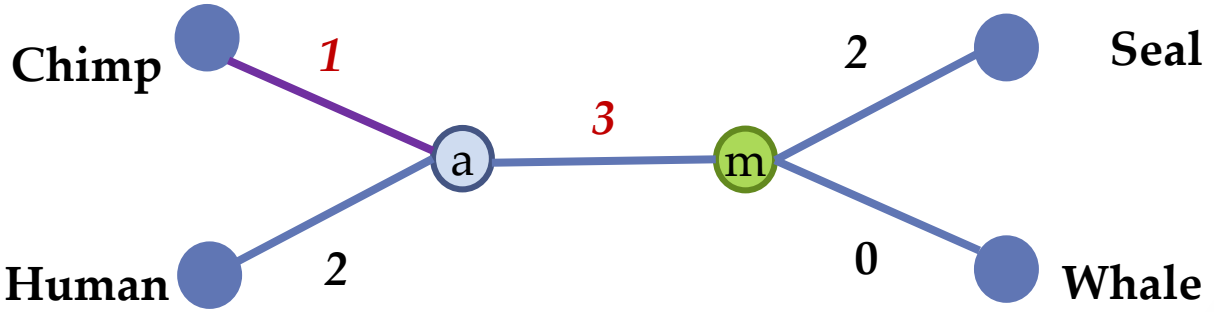
	Chimp	Human	m
Chimp	0	**3**	4
Human	**3**	0	5
m	4	5	0



$$d_{Chimp,a} = (D_{Chimp,m} + D_{Chimp,Human} - D_{Human,m})/ 2$$

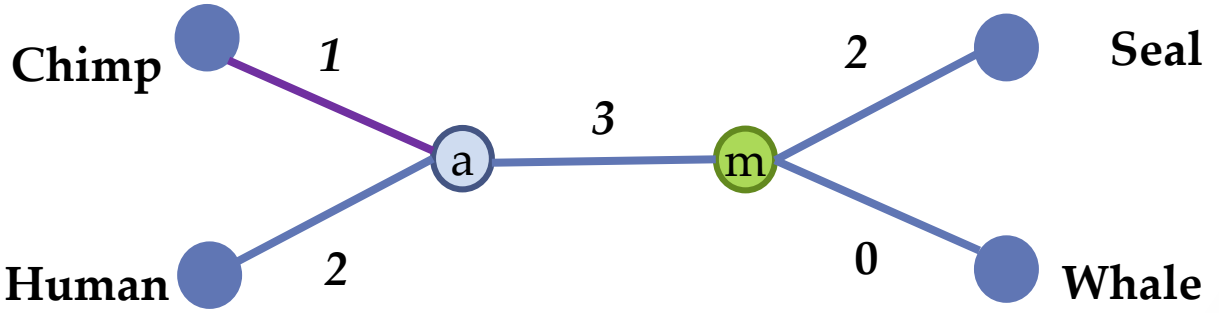
An Idea of Distance-Based Phylogeny

	Chimp	Human	m
Chimp	0	3	**4**
Human	3	0	5
m	4	5	0



An Idea of Distance-Based Phylogeny

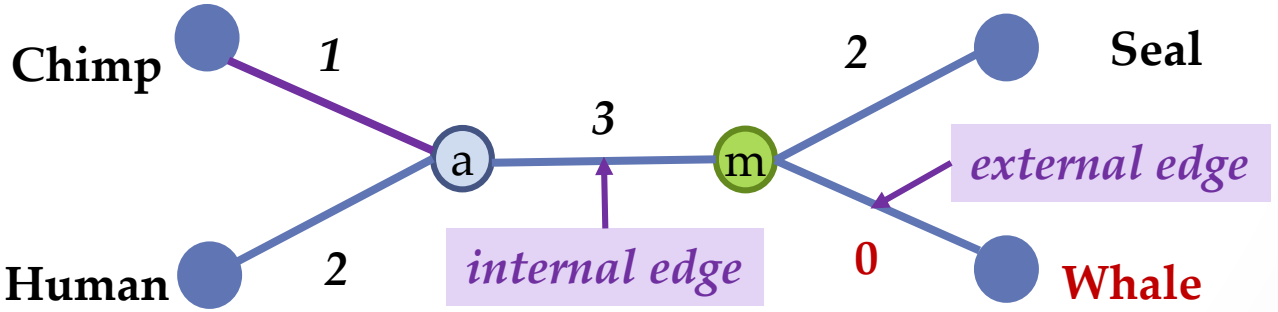
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



The ***simple tree*** that fits to the original matrix.

An Idea of Distance-Based Phylogeny

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



An Idea of Distance-Based Phylogeny

Exercise: Apply this recursive approach to distance matrix below.

	i	j	k	l
i	0	13	21	22
j	13	0	12	13
k	21	12	0	13
l	22	13	13	0

Outline

- Transforming Distance Matrices into Evolutionary Trees
- Toward an Algorithm for Distance-Based Phylogeny Construction
- **Additive Phylogeny**
- Ultrametric Evolutionary Trees (UPGMA reconstruction)
- The Neibour-Joining Algorithm
- Using Least-Squares to construct Distance-Based Phylogenies

Computing Limb Length

Limb Length Theorem: *LimbLength(i)* is equal to the minimum value of $(D_{i,k} + D_{i,j} - D_{j,k})/2$ over all leaves *j* and *k*.

Limb Length Problem: Compute the length of a limb in the simple tree fitting an additive distance matrix.

- **Input:** An additive distance matrix *D* and an integer *j*.
- **Output:** The length of the limb connecting leaf *j* to its parent, *LimbLength(j)*.

Computing Limb Length

Limb Length Theorem: *LimbLength(i)* is equal to the minimum value of $(D_{i,k} + D_{i,j} - D_{j,k})/2$ over all leaves *j* and *k*.

	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0

$$(D_{chimp,human} + D_{chimp,\ seal} - D_{human,seal}) / 2$$

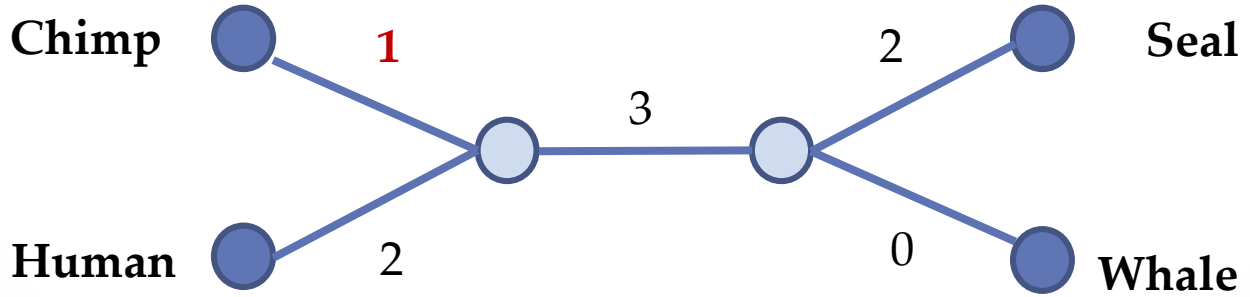
$$(D_{chimp,human} + D_{chimp,\ whale} - D_{human,whale}) / 2$$

$$(D_{chimp,whale} + D_{chimp,seal} - D_{whale,seal}) / 2$$

Computing Limb Length

Limb Length Theorem: *LimbLength(i)* is equal to the minimum value of $(D_{i,k} + D_{i,j} - D_{j,k})/2$ over all leaves *j* and *k*.

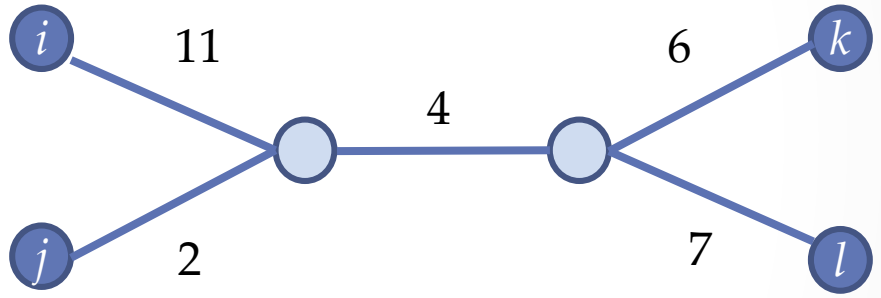
	Chimp	Human	Seal	Whale
Chimp	0	3	6	4
Human	3	0	7	5
Seal	6	7	0	2
Whale	4	5	2	0



Additive Phylogeny In Action

		i	*j*	*k*	*l*
	i	0	13	21	22
D	*j*	13	0	12	13
	k	21	12	0	13
	l	22	13	13	0

TREE(***D***)



Additive Phylogeny In Action

		i	j	k	l
	i	0	13	21	22
D	j	13	0	12	13
	k	21	12	0	13
	l	22	13	13	0

1. Pick an arbitrary leaf j.

Additive Phylogeny In Action

		i	j	k	l
	i	0	13	21	22
D	j	13	0	12	13
	k	21	12	0	13
	l	22	13	13	0

$LimbLength(j) = 2$

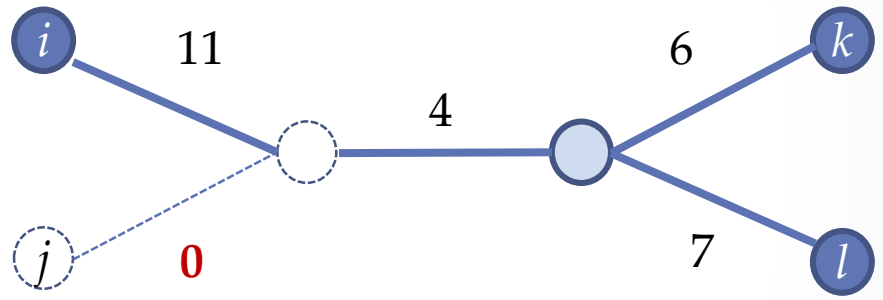
2. Compute its limb length, *LimbLength(j)*.

Additive Phylogeny In Action

D^{bald}

	i	j	k	l
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0

TREE(D^{bald})



3. Substruct *LimbLength(j)* from each j row and j column to produce D^{bald} in which *j* is a **bald limb** (length 0).

Additive Phylogeny In Action

D^{trim}

	i	j	k	l
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0

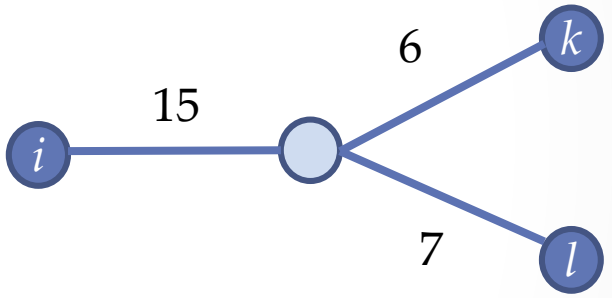
4. Remove the *j*-th row and column of the matrix to form the (*n*-1) x (*n*-1) matrix D^{trim}.

Additive Phylogeny In Action

D^{trim}

	i	j	k	l
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0

TREE(D^{trim})

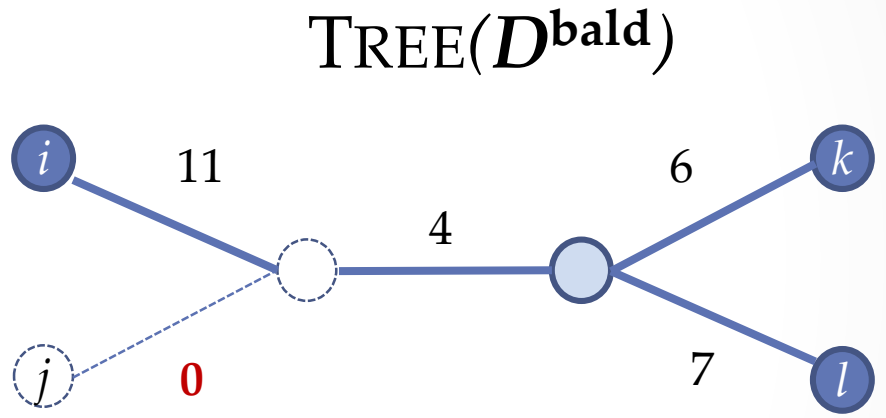


5. Construct Tree(D^{trim}).

Additive Phylogeny In Action

D^{bald}

	i	j	k	l
i	0	11	21	22
j	11	0	10	11
k	21	10	0	13
l	22	11	13	0



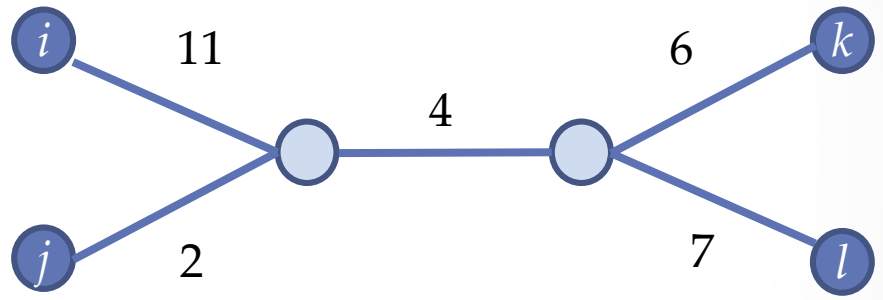
6. Identify the point in Tree(D^{trim}) where leaf j should be attached.

Additive Phylogeny In Action

		i	*j*	*k*	*l*
	i	0	13	21	22
D	*j*	13	0	12	13
	k	21	12	0	13
	l	22	13	13	0

***LimbLength(j)* = 2**

TREE(***D***)



7. Attach j by an edge of length *LimbLength(j)* in order to form Tree(*D*).

Additive Phylogeny

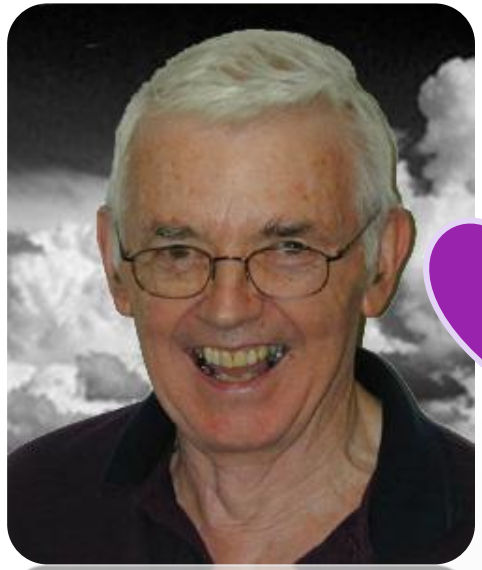
1. Pick an arbitrary leaf *j*.
2. Compute its limb length, *LimbLength(j)*.
3. Substruct *LimbLength(j)* from each row and column to produce D^{bald} in which *j* is a **bald limb** (length 0).
4. Remove the *j*-th row and column of the matrix to form the (*n*-1) x (*n*-1) matrix D^{trim}.
5. Construct Tree(D^{trim}).
6. Identify the point in Tree(D^{trim}) where leaf *j* should be attached.
7. Attach j by an edge of length *LimbLength(j)* in order to form Tree(*D*).

Distance Matrix Methods

Luidgi L. Cavalli-Sforza

Anthony W.F. Edwards

1967



Walter M. Fitch

Emanuel Margoliash

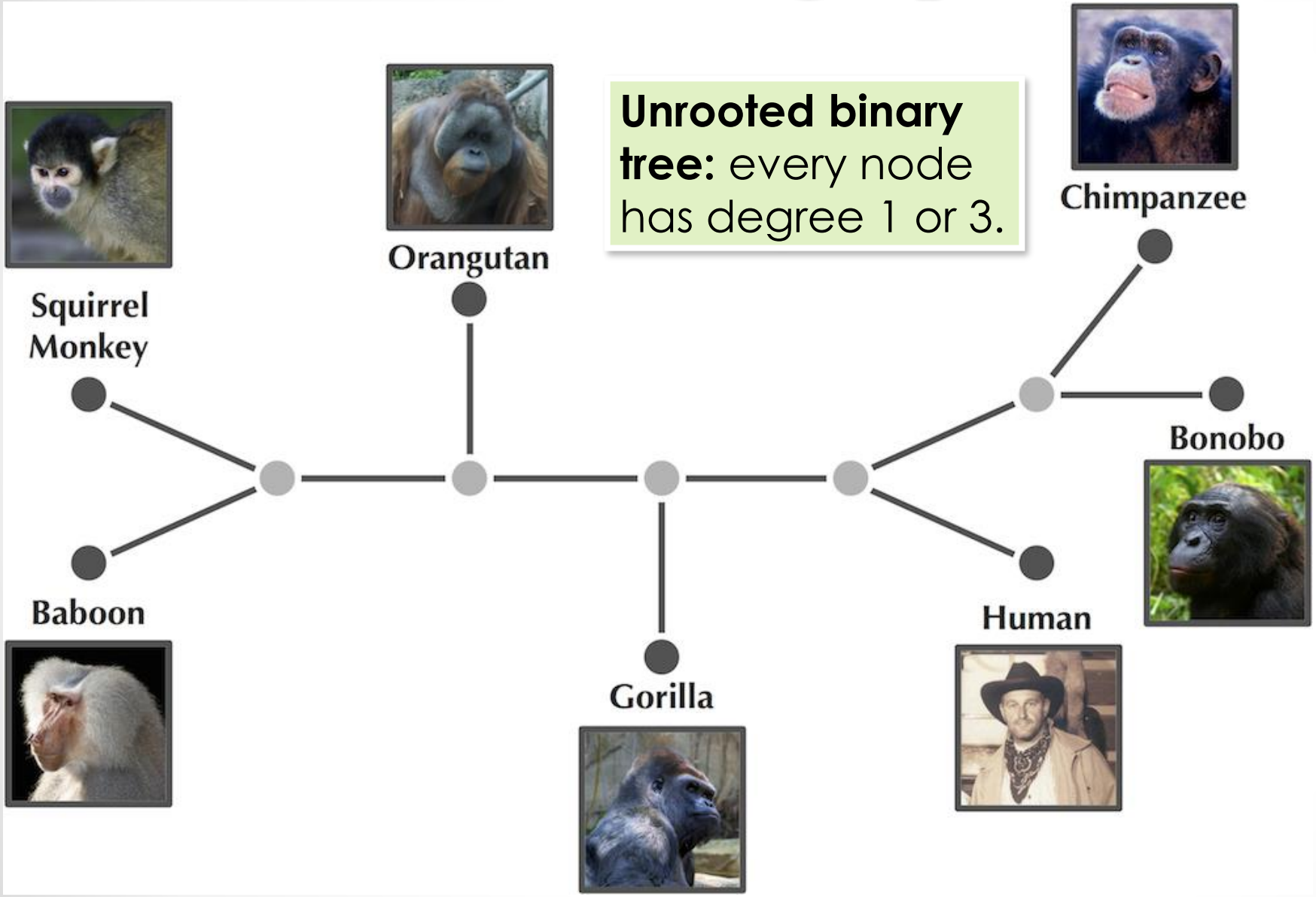
Outline

- Transforming Distance Matrices into Evolutionary Trees
- Toward an Algorithm for Distance-Based Phylogeny Construction
- Additive Phylogeny
- **Ultrametric Evolutionary Trees (UPGMA reconstruction)**
- The Neibour-Joining Algorithm
- Using Least-Squares to construct Distance-Based Phylogenies

Modeling Speciations

Researches often assume that all internal nodes correspond to **speciations**, where one species splits into two.

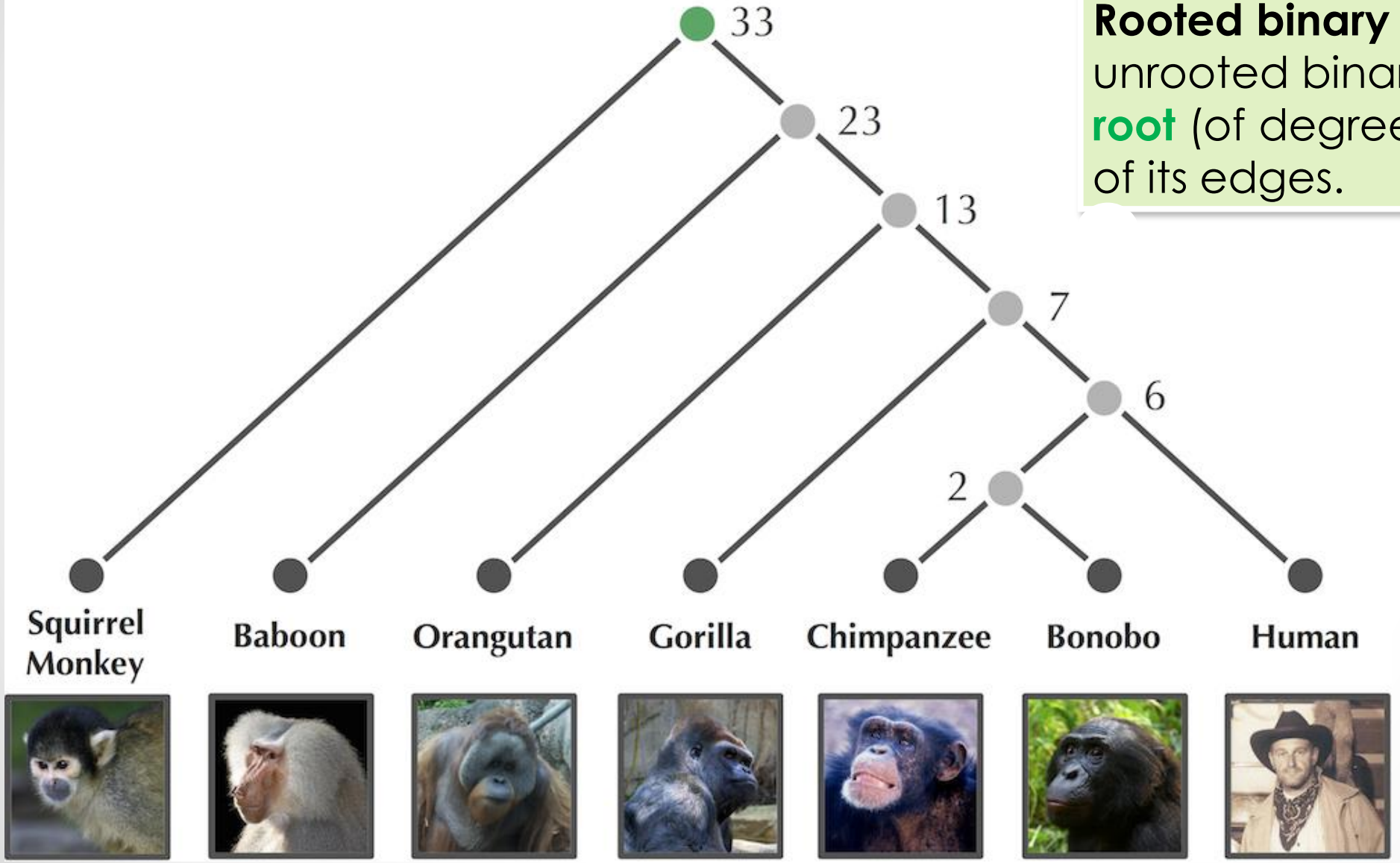
Modeling Speciations



In computer science, a **binary tree** is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.

We need to place limits on the internal nodes of the tree: every internal node needs to have degree 3.
Progressing from the root to a leaf, every time we encounter an internal node, the tree splits into two pieces.

Modeling Speciations

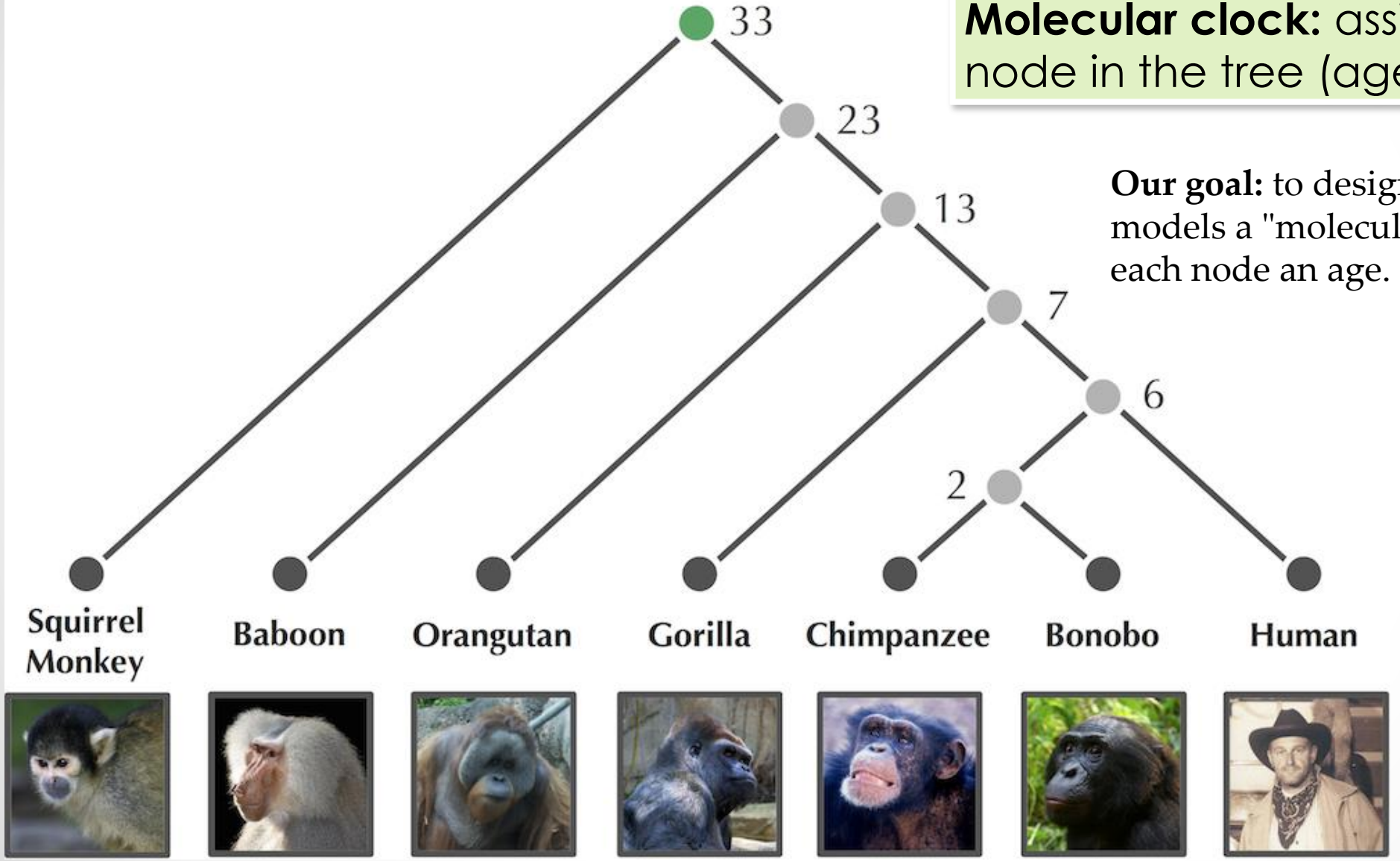


Rooted binary tree: an unrooted binary with a **root** (of degree 2) on 1 of its edges.

Placing a root on the squirrel monkey's limb results in a rooted binary tree.

The number at each node corresponds to the number of million years ago that the divergence at this node occurred.

Ultrametric Trees



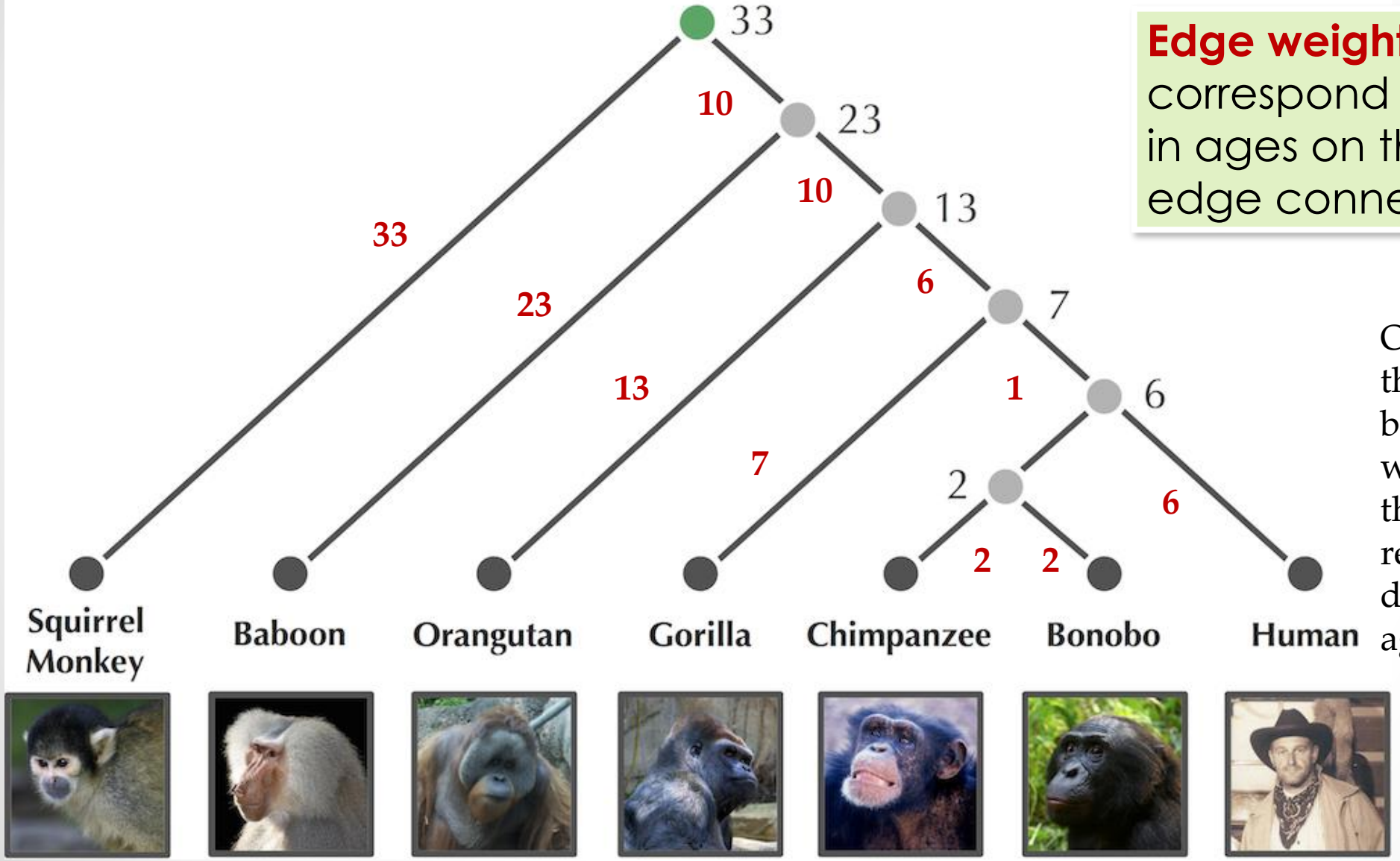
Molecular clock: assigns **ages** to each node in the tree (age of leaves=0)

Our goal: to design a heuristic that models a "molecular clock" that assigns each node an age.

The age of an internal node corresponds to how long ago the speciation event represented at that internal node occurred.

Then we need to assign ages to the internal edges.

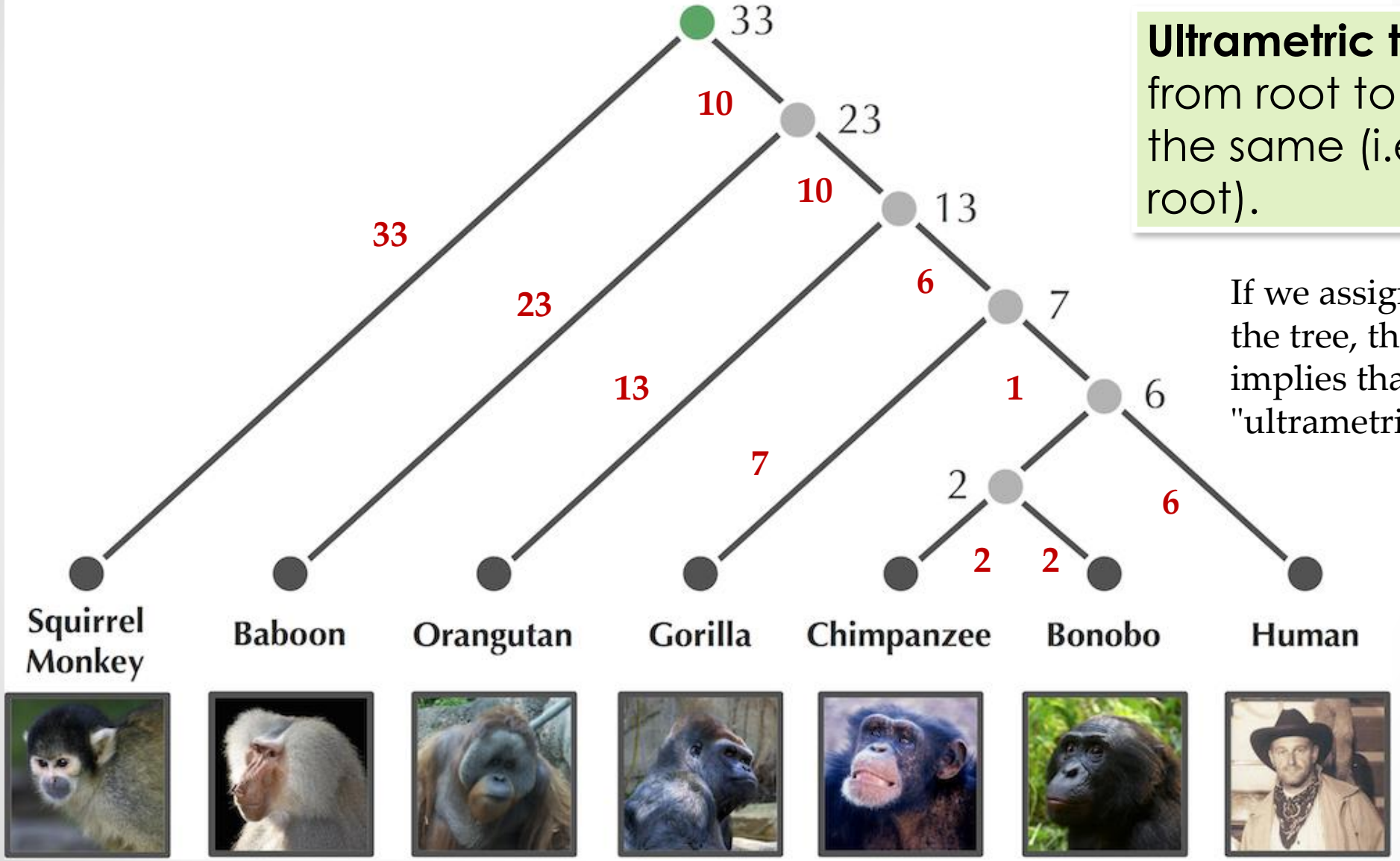
Ultrametric Trees



Edge weights: correspond to difference in ages on the nodes the edge connects.

Once we assign ages to the nodes of a rooted binary tree, we get the weights of the edges of this tree (here shown in red) simply by taking the difference between the ages.

Ultrametric Trees

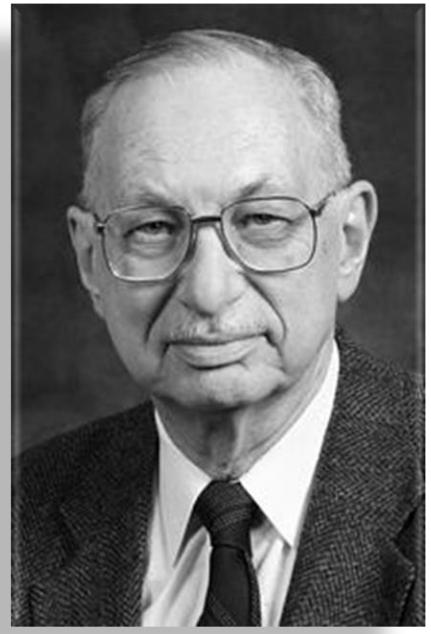


Ultrametric tree: distance from root to any leaf is the same (i.e., age of root).

If we assign ages to the nodes of the tree, then it automatically implies that the tree is "ultrametric".

UPGMA: A Clustering Heuristic

- **UPGMA** (**U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic Mean) is a simple agglomerative (bottom-up) hierarchical clustering method.



Robert A. Sokal, biostatistician

&

Charles D. Michener, entomologist

*Sokal R and Michener C (1958). "A statistical method for evaluating systematic relationships". University of Kansas Science Bulletin. **38**: 1409–1438.*

UPGMA: A Clustering Heuristic

1. Form a cluster for each present-day species, each containing a single leaf.

	i	j	k	l
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0

UPGMA constructs an evolutionary tree by clustering the species from the distance matrix into larger and larger clusters, beginning with single element clusters.

UPGMA: A Clustering Heuristic

2. Find the two closest clusters C1 and C2 according to the average distance

$$D_{avg}(C_1, C_2) = \sum_{i \text{ in } C1, j \text{ in } C2} D_{i,j} / |C_1| \cdot |C_2|$$

Where $|C|$ denotes the number of elements in C.

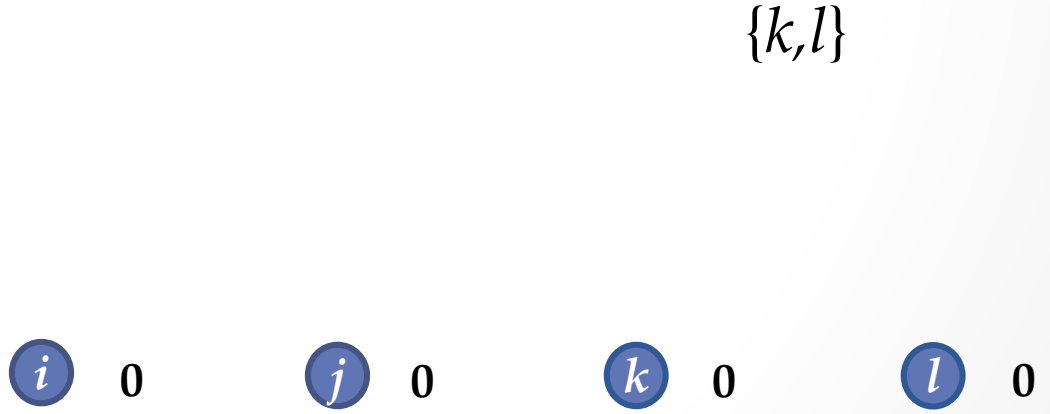
	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	**2**
l	3	5	**2**	0

At each step it looks for the two closest clusters, according to the average distance among all pairs of elements taken from the two clusters. At this stage of the algorithm, we're dealing with single element clusters, so if we're looking for the closest clusters, that's just the smallest element of the distance matrix, which corresponds to k and l.

UPGMA: A Clustering Heuristic

3. Merge C_1 and C_2 into a single cluster C.

	i	j	k	l
i	0	3	4	3
j	3	0	4	5
k	4	4	0	**2**
l	3	5	**2**	0

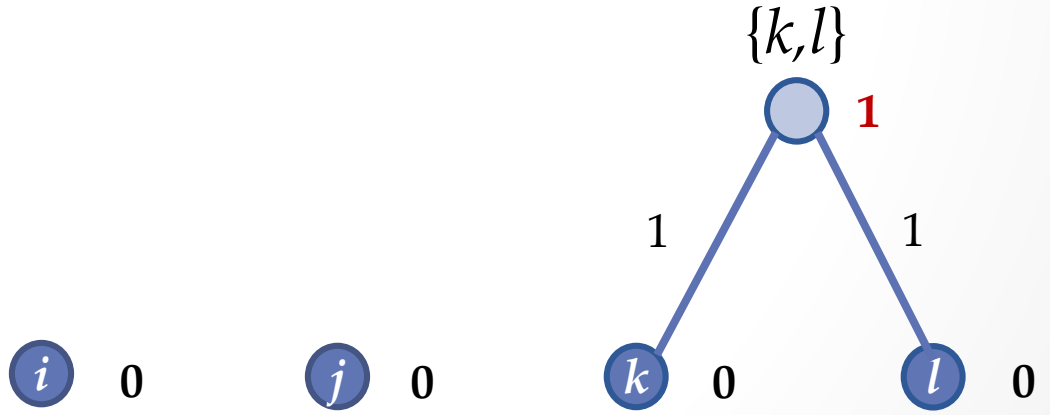


Once the two closest clusters C1 and C2 are found, we can merge them into a single cluster, C. Here we put k and l into a cluster together because they're the closest.

UPGMA: A Clustering Heuristic

4. Form a new node for C and connect to C_1 and C_2 by an edge. Set age of C as $D_{avg}(C_1, C_2)/2$.

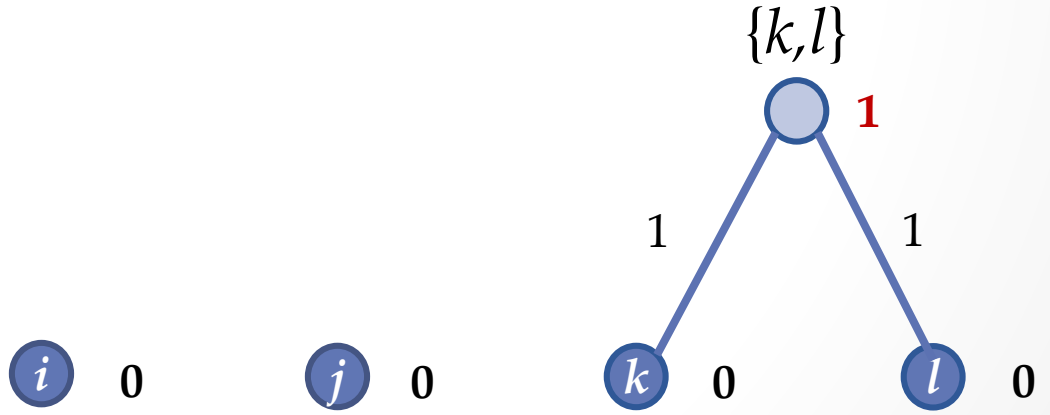
	i	j	k	l
i	0	3	4	3
j	3	0	4	5
k	4	4	0	**2**
l	3	5	**2**	0



UPGMA: A Clustering Heuristic

5. Update the distance matrix by computing the average distance between each pair of clusters.

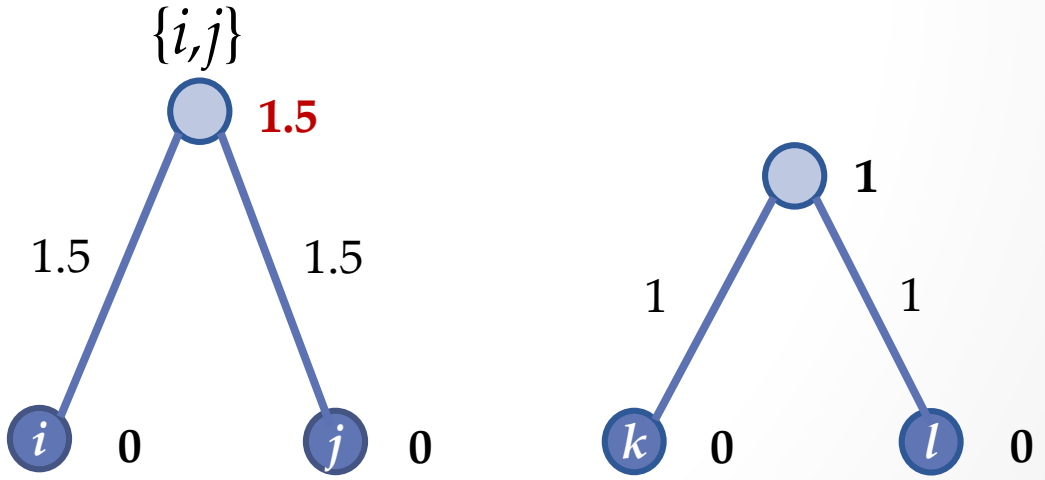
	i	j	$\{k,l\}$
i	0	3	3.5
j	3	0	4.5
$\{k,l\}$	3.5	4.5	0



UPGMA: A Clustering Heuristic

6. Iterate until a single cluster contains all species.

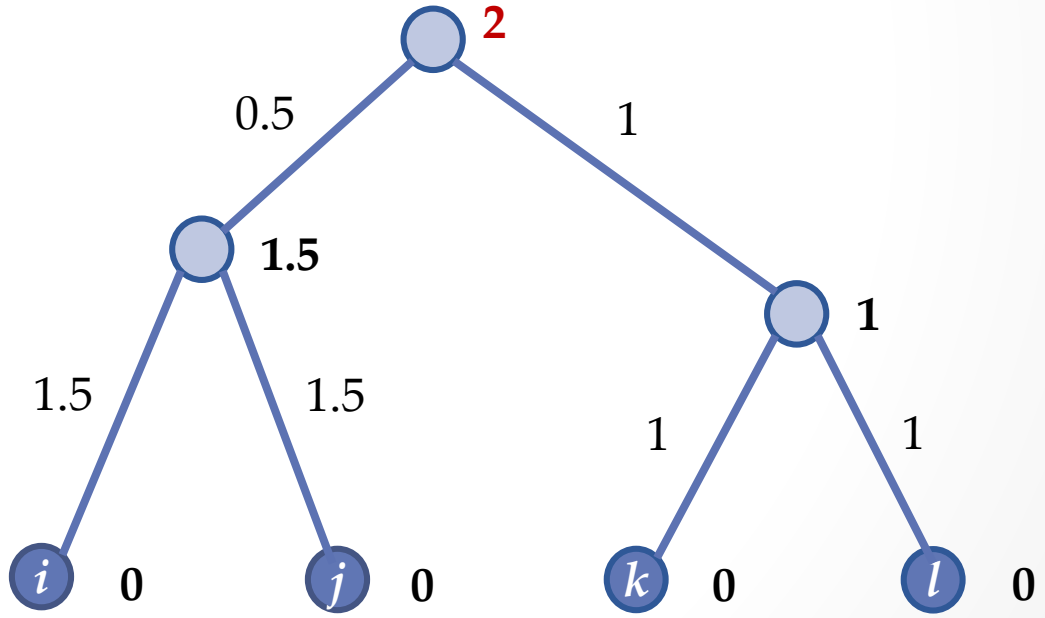
	i	j	$\{k,l\}$
i	0	**3**	3.5
j	**3**	0	4.5
$\{k,l\}$	3.5	4.5	0



UPGMA: A Clustering Heuristic

6. Iterate until a single cluster contains all species.

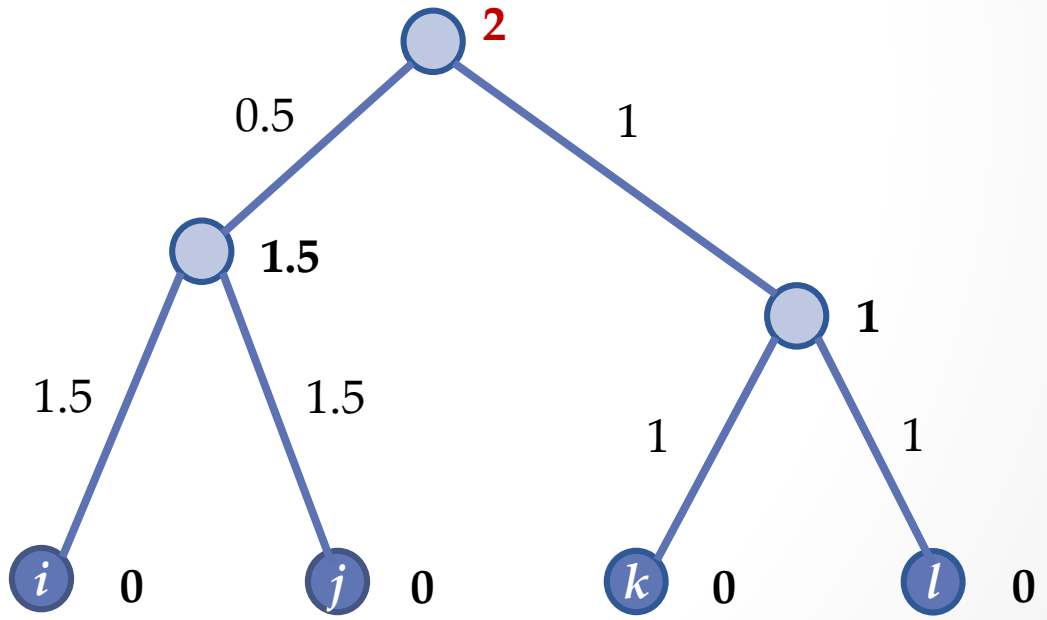
	$\{i,j\}$	$\{k,l\}$
$\{i,j\}$	0	4
$\{k,l\}$	4	0



UPGMA: A Clustering Heuristic

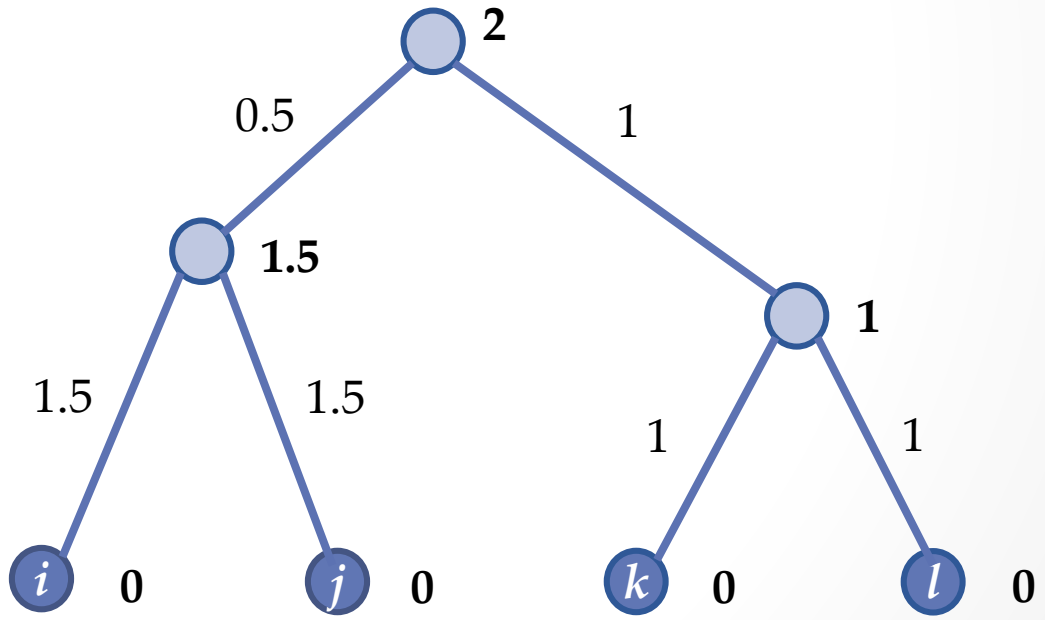
6. Iterate until a single cluster contains all species.

	$\{i,j\}$	$\{k,l\}$
$\{i,j\}$	0	**4**
$\{k,l\}$	**4**	0



UPGMA: A Clustering Heuristic

6. Iterate until a single cluster contains all species.



UPGMA: A Clustering Heuristic

1. Form a cluster for each present-day species, each containing a single leaf.

2. Find the two closest clusters C1 and C2 according to the average distance
   $D_{avg}(C_1, C_2) = \sum_{i \text{ in } C1, j \text{ in } C2} D_{i,j} / |C_1| \cdot |C_2|$
   Where $|C|$ denotes the number of elements in C.

3. Merge C_1 and C_2 into a single cluster C.

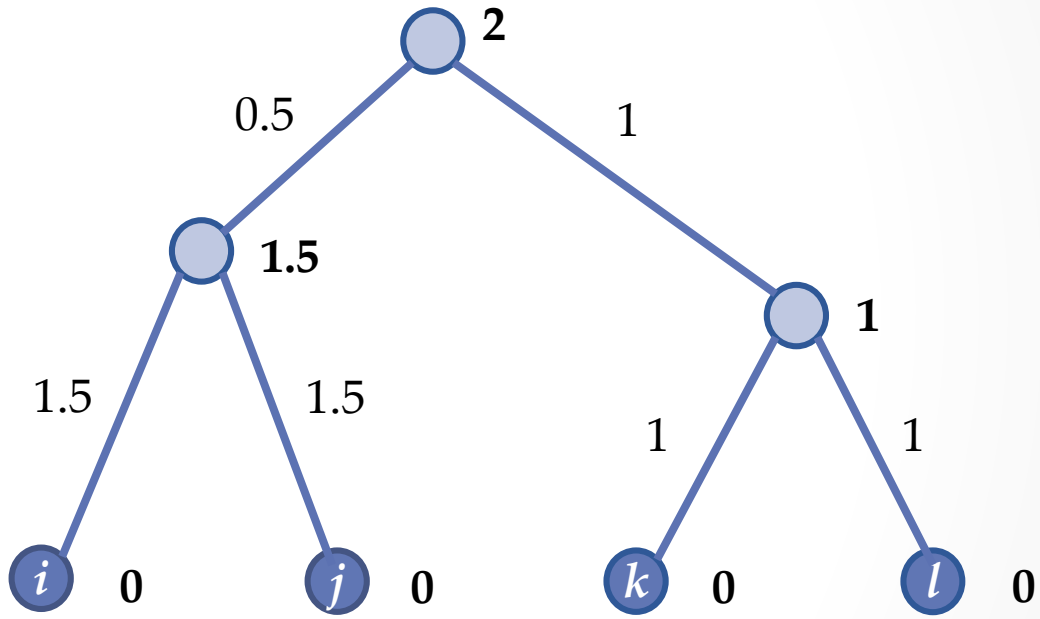
4. Form a new node for C and connect to C_1 and C_2 by an edge. Set age of C as $D_{avg}(C_1, C_2)/2$.

5. Update the distance matrix by computing the average distance between each pair of clusters.

6. Iterate until a single cluster contains all species.

UPGMA Doesn't "Fit" a Tree to a Matrix

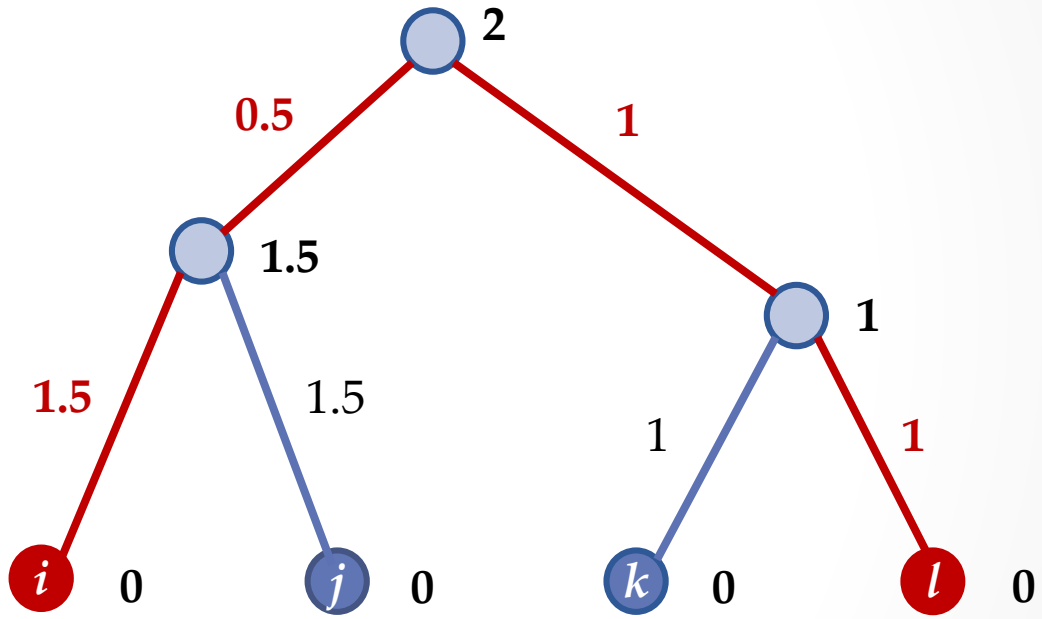
	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0



The goal of UPGMA is not to fit a distance matrix, but to provide a reliable method that always can construct an ultrametric tree, regardless of what the input data is.

UPGMA Doesn't "Fit" a Tree to a Matrix

	i	*j*	*k*	*l*
i	0	3	4	3
j	3	0	4	5
k	4	4	0	2
l	3	5	2	0



The tree produced by UPGMA can't possibly fit this matrix, because the matrix is non-additive.
For example, the distance from i to l is 3 in the distance matrix, but 4 according to the UPGMA tree.

Exercise Break

- Below is a distance matrix D. If C1 is the cluster containing i and k, and C2 is the cluster containing j and l, compute D(C1, C2).

	i	***j***	***k***	***l***
i	0	20	9	11
j	20	0	17	11
k	9	17	0	8
l	11	11	8	0

Exercise Break

- Reconstruct phylogenetic tree from the following distance matrix using UPGMA approach:

OTUS	A	B	(CD)	E
B	6			
(CD)	29	31		
E	24	26	32	
F	30	28	15	30

- What would be the topology of this tree? (Parentheses indicate the order of grouping):

a) ((EAB(CDF)); b) (EA(B))(CD)F); c) ((AB)(CD)F)E; d) (E(AB))((CD)F)

- If in the previous exercise dF(CD) – d(CD)=9, what is the distace of the both taxons C and D to their most recent common ancestor?

a) 3,0 ; b) 1,5 ; c) 1,0 ; d) 2,0