CLUSTERING METHODS AND ALGORITHMS IN GENOMICS AND EVOLUTION

Session 9.1

Distance based methods for tree inference

In Summary…

- **Additive Phylogeny:**
  - good: produces the tree fitting an *additive* matrix
  - bad: fails completely on a *non-additive* matrix

- **UPGMA:**
  -good: produces a tree for any matrix
  - bad: tree doesn't necessarily fit an additive matrix

- **?????:**
  - good: produces the tree fitting an additive matrix
  - good: provides heuristic for a non-additive matrix

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

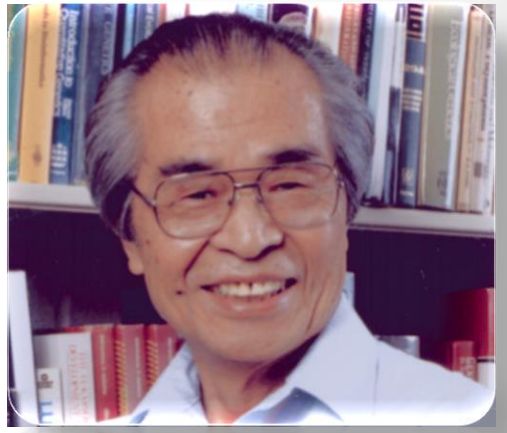
Introduction to Neighbor-Joining

- **Neighbor joining** is a bottom-up (agglomerative) clustering method for the creation of phylogenetic trees.
- Fundamental for bioinformatics: **30,000** citations
- One of the **top 20 most cited papers** over all scientific fields.

1987

Naruya Saitou

&



Masatoshi Nei

Saitou N, Nei M. "The neighbor-joining method: a new method for reconstructing phylogenetic trees." Molecular Biology and Evolution, volume 4, issue 4, pp. 406-425, July 1987

Neighbor-Joining Theorem

Given an n x n distance matrix D, its **neighbor-joining** matrix is the matrix *D** defined as

$$D^*_{i,j} = (n - 2) \cdot D_{i,j} - TotalDistance_D(i) - TotalDistance_D(j),$$

Where $TotalDistance_D(i)$ is the sum of distance from *i* to all other leaves .

D

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

$TotalDistance_D$

56
38
46
48

D*

	i	*j*	*k*	*l*
i	0	-68	-60	-60
j	-68	0	-60	-60
k	-60	-60	0	-68
l	-60	-60	-68	0

Neighbor-Joining in Action

Neighbor-Joining Theorem: If *D* is additive, then the smallest element of *D** corresponds to neighboring leaves in *Tree(D)*.

D

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

$TotalDistance_D$
56
38
46
48

D^*

	i	j	k	l
i	0	**-68**	-60	-60
j	**-68**	0	-60	-60
k	-60	-60	0	**-68**
l	-60	-60	**-68**	0

Neighbor-Joining in Action

D^*

	i	j	k	l
i	0	-68	-60	-60
j	-68	0	-60	-60
k	-60	-60	0	-68
l	-60	-60	-68	0

$TotalDistance_D$
56
38
46
48

1. Construct neighbor-joining matrix ***D**** from ***D***.

Neighbor-Joining in Action

D^*

	i	j	k	l
i	0	**-68**	-60	-60
j	**-68**	0	-60	-60
k	-60	-60	0	**-68**
l	-60	-60	**-68**	0

$TotalDistance_D$
56
38
46
48

2. Find a minimum element D^*_{ij} of D^*.

Neighbor-Joining in Action

D^*

	i	j	k	l
i	0	**-68**	-60	-60
j	**-68**	0	-60	-60
k	-60	-60	0	-68
l	-60	-60	-68	0

$TotalDistance_D$
56
38
46
48

2. Find a minimum element D^*_{ij} of D^*.

Neighbor-Joining in Action

D^*

	i	j	k	l
i	0	**-68**	-60	-60
j	**-68**	0	-60	-60
k	-60	-60	0	-68
l	-60	-60	-68	0

$TotalDistance_D$
56
38
46
48

$\Delta_{i,j} = (\mathbf{56}\text{-}\mathbf{38})/(4\text{-}2) = 9$

3. Compute $\Delta_{i,j} = (TotalDistance_D(i) - TotalDistance_D(j))/(n\text{-}2)$.

Neighbor-Joining in Action

D

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

$TotalDistance_D$
56
38
46
48

$\Delta_{i,j} = (56\text{-}38)/(4\text{-}2) = 9$

LimbLength(i) = ½(13 + 9) = 11
LimbLength(j) = ½(13 - 9) = 2

4. Set *LimbLength(i)* equal to ½($D_{i,j}$ + $\Delta_{i,j}$) and *LimbLength(j)* equal to ½($D_{i,j}$ - $\Delta_{i,j}$).

Neighbor-Joining in Action

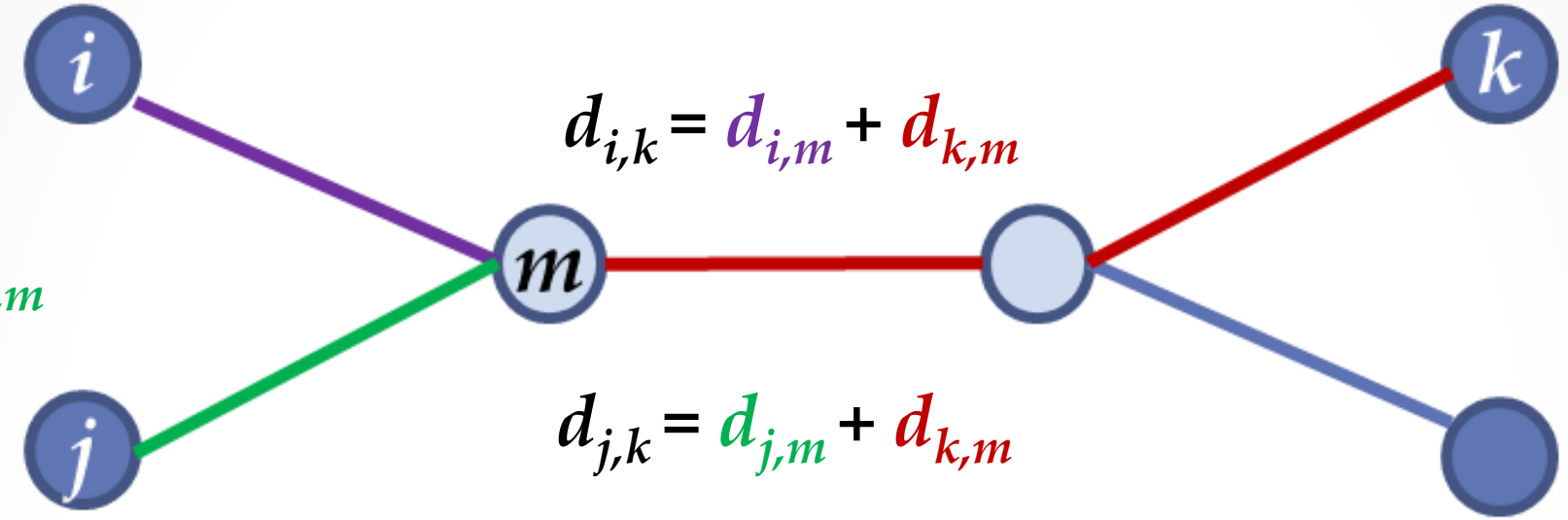
D'

	m	k	l
m	0	10	11
k	10	0	13
l	11	13	0

$TotalDistance_D$
21
23
24

5. Form a matrix D' by removing i-th and j-th row/column from D and adding an m-th row/column such that for any *k*, $D_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j})/2$.

Flashlback: Computation of $d_{k,m}$



$d_{i,j} = d_{i,m} + d_{j,m}$

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

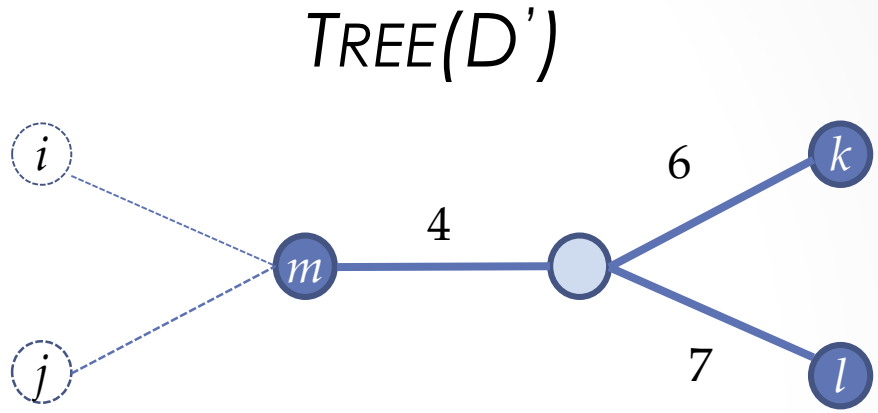
$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j})/ 2$$

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

Neighbor-Joining in Action

D'

	m	k	l
m	0	10	11
k	10	0	13
l	11	13	0

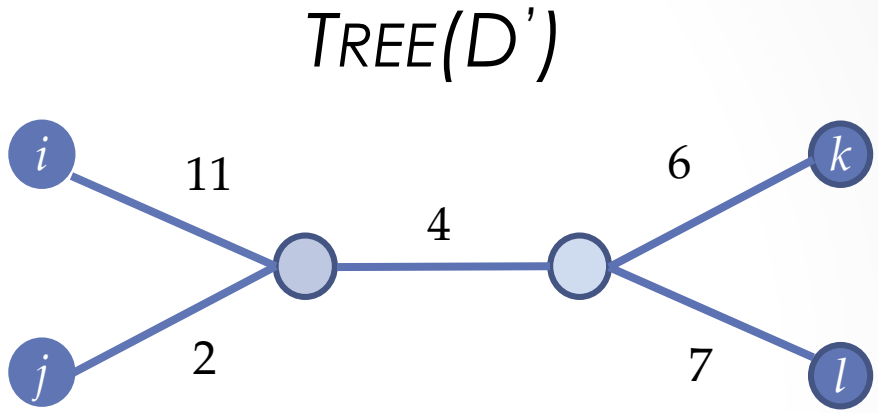


6. Apply Neighbor-Joining to *D'* to obtain *Tree(D')*.

Neighbor-Joining in Action

D'

	m	k	l
m	0	10	11
k	10	0	13
l	11	13	0



LimbLength(i) = ½(13 + 9) = 11
LimbLength(j) = ½(13 - 9) = 2

7. Reattach limb of *i* and *j* to obtain *Tree(D)*.

Neighbor-Joining

1. Construct neighbor-joining matrix D* from D.

2. Find a minimum element D^*_{ij} of D^*.

3. Compute $\Delta_{i,j} = (TotalDistance_D(i) - TotalDistance_D(j))/(n-2)$.

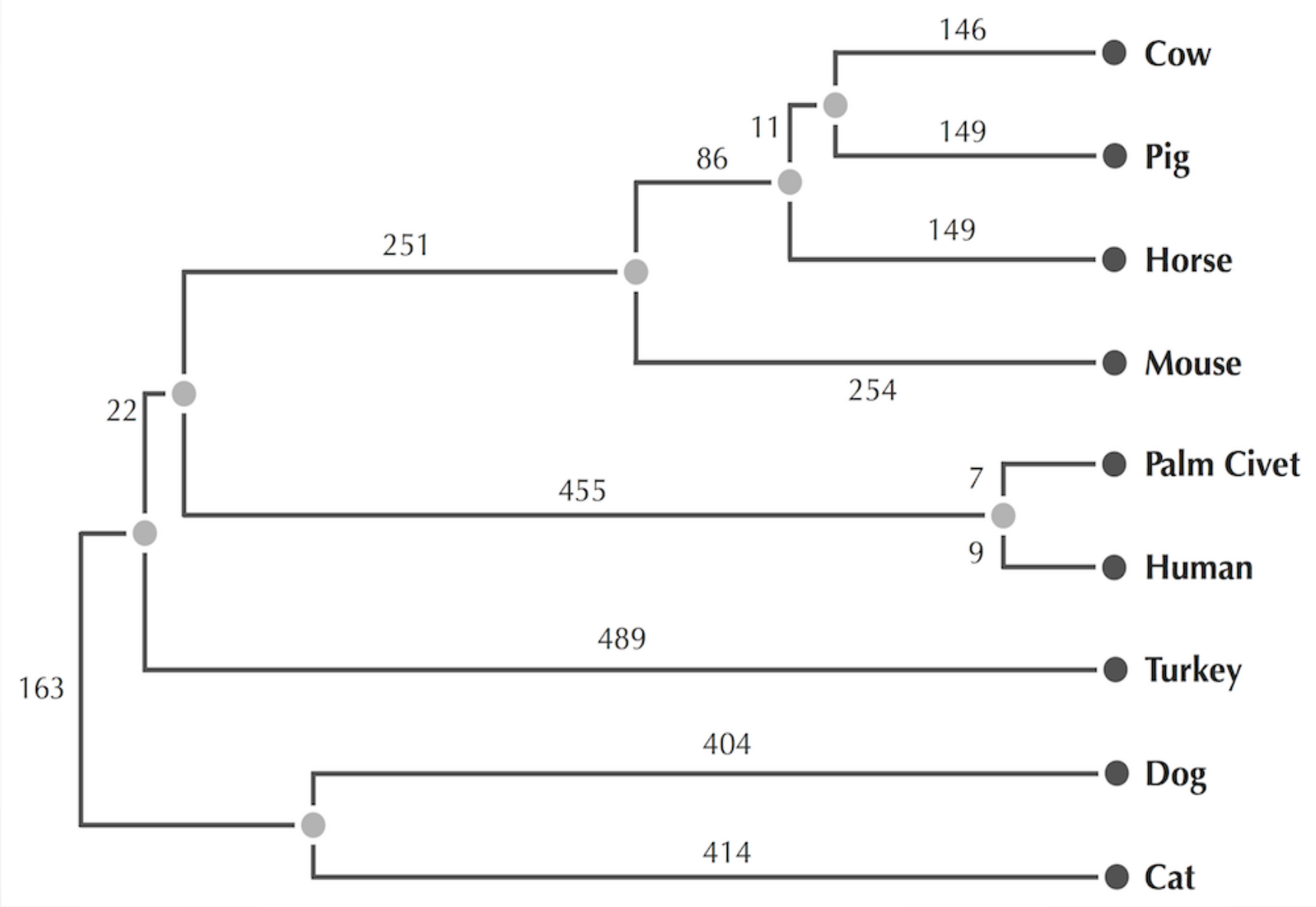
4. Set $LimbLength(i)$ equal to $\frac{1}{2}(D_{i,j} + \Delta_{i,j})$ and $LimbLength(j)$ equal to $\frac{1}{2}(D_{i,j} - \Delta_{i,j})$.

5. Form a matrix D' by removing i-th and j-th row/column from D and adding an m-th row/column such that for any k, $D_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j})/2$.

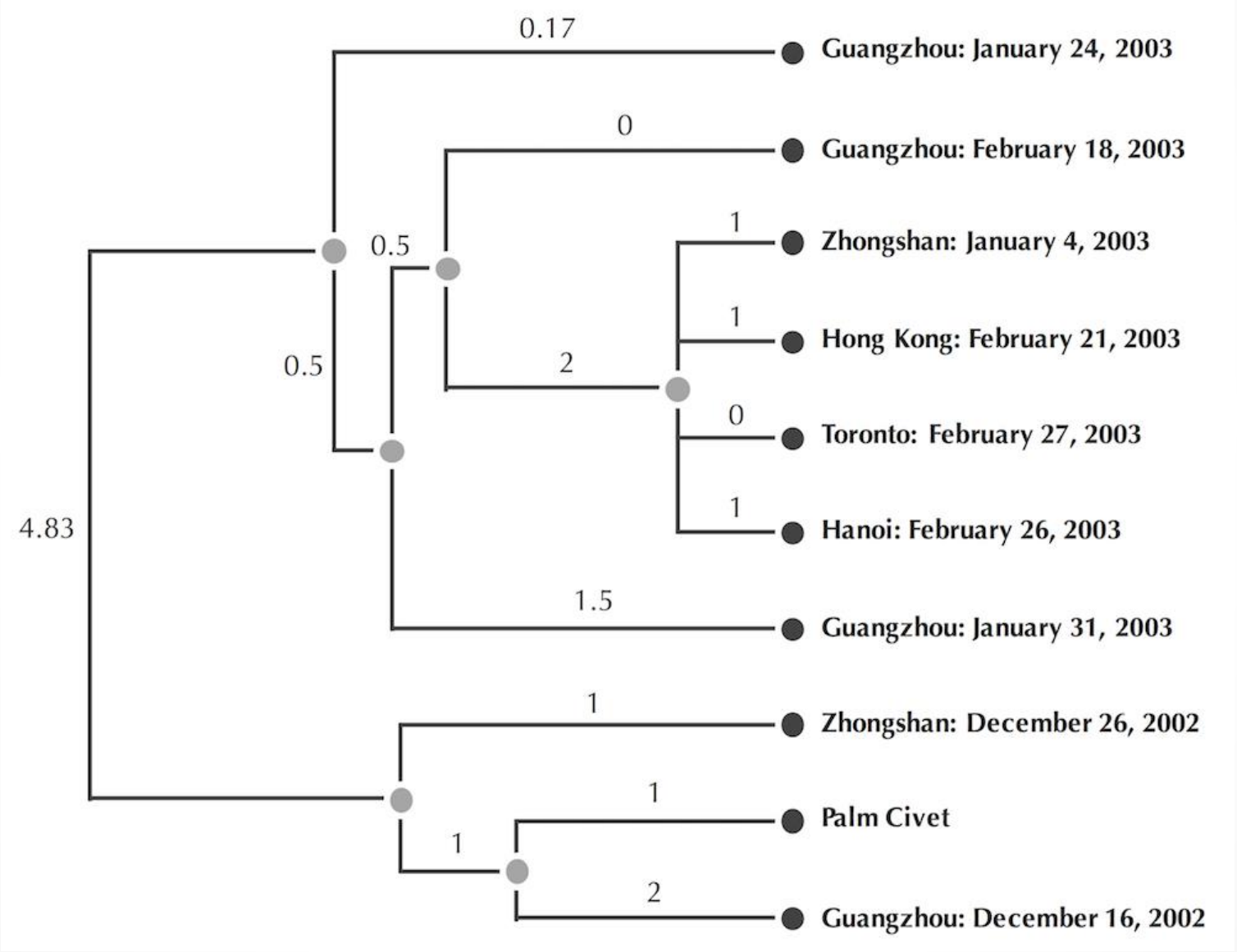
6. Apply Neighbor-Joining to D' to obtain $Tree(D')$.

7. Reattach limb of i and j to obtain $Tree(D)$.

Neighbor-Joining on Coronoviruses



Neighbor-Joining on Coronoviruses



Exercise Break

- Below is a distance matrix *D*. Compute $D^*_{k,l}$ where D^* is the neighbor-joining matrix of *D*.

	i	***j***	***k***	***l***
i	0	13	16	10
j	13	0	21	15
k	16	21	0	18
l	10	15	18	0

- Below is a distance matrix *D*. After the neighbor- joining algorithm decides that *j* and *k* are neighbors, compute *LimbLength(k)*.

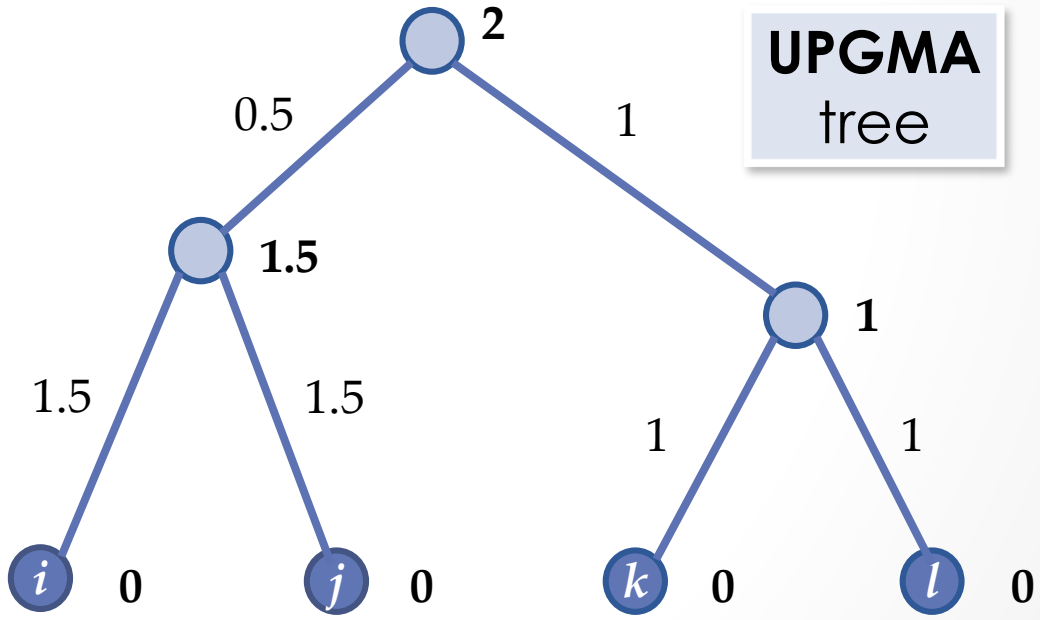
	i	***j***	***k***	***l***
i	0	14	17	17
j	14	0	7	13
k	17	7	0	16
l	17	13	16	0

Neighbor-Joining

Exercise Break: Find the tree returned by **Neighbor-Joining** on the following non-additive matrix. How does the result compare with the tree produced by **UPGMA**?

D

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

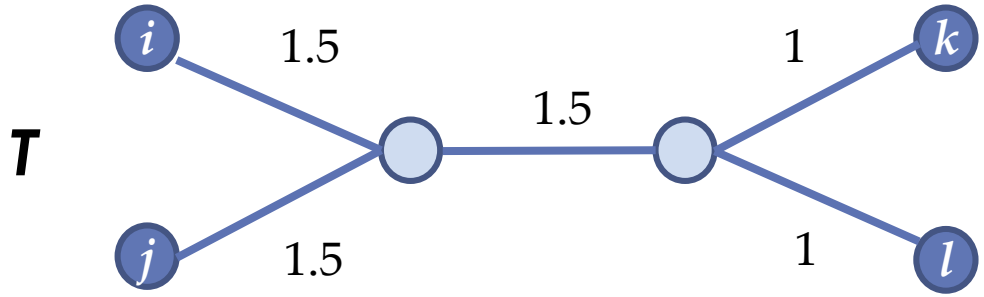


UPGMA tree

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**

Sum of Squared Errors

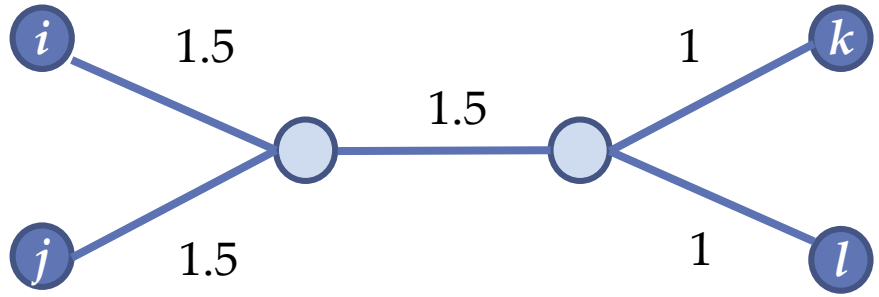


D

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

Sum of Squared Errors

T



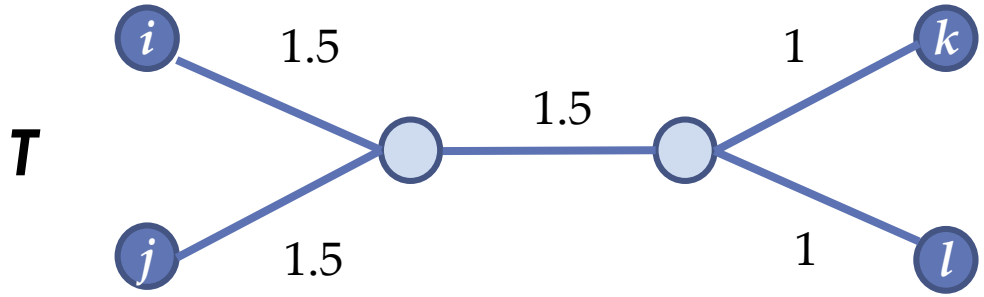
D

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

d

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

Sum of Squared Errors



D

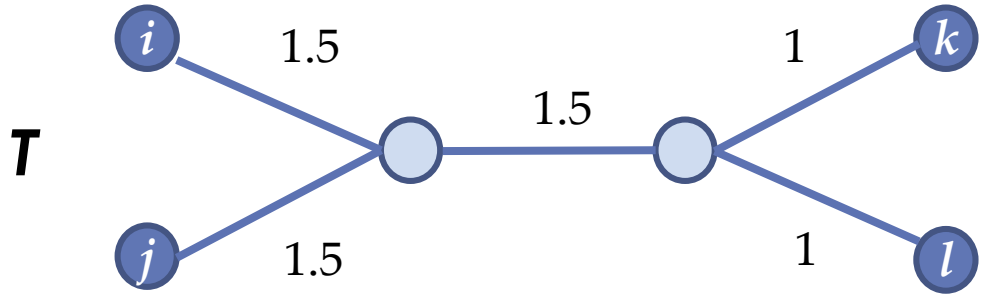
	i	j	k	l
i	0	3	4	**3**
j	3	0	4	**5**
k	4	4	0	2
l	3	5	2	0

d

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

Sum of Squared Errors

$$\textbf{Discrepancy}(T, D) = \sum_{1 \le i < j \le n} (d_{i,j}(T) - D_{i,j})^2$$



D

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

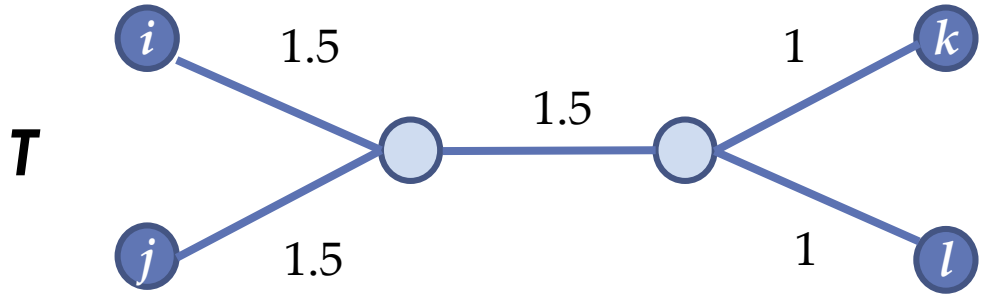
d

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

Sum of Squared Errors

$$\textbf{Discrepancy}(T, D) = \sum_{1 \le i < j \le n} (d_{i,j}(T) - D_{i,j})^2 = 1^2 + 1^2 = 2$$

(4-3)-(4-5) --> 1^2-(-1^2)=1^2+1^2



D

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

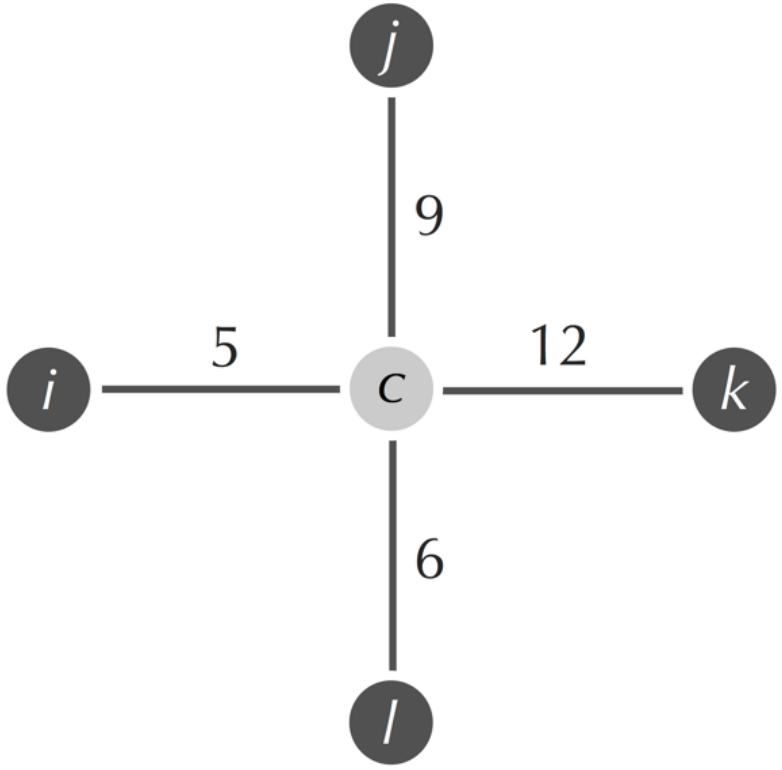
d

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

Exercise Break

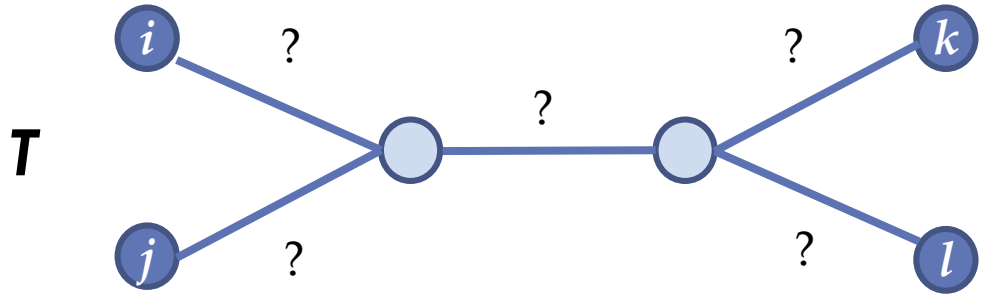
Compute the sum of squared errors *Discrepancy(T,D)* for the tree T and distance martrix *D* given below.

	i	***j***	***k***	***l***
i	0	13	16	10
j	13	0	21	15
k	16	21	0	18
l	10	15	18	0



Sum of Squared Errors

Exercise Break: Assign lengths to edges in T in order to minimize *Discrepancy(T, D)*



D

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

d

	i	j	k	l
i	0	?	?	?
j	?	0	?	?
k	?	?	0	?
l	?	?	?	0

Least-Squares Phylogeny

Least-Squares Distance-Based Phylogeny Problem:

Given a distance matrix, find the tree that minimizes the sum of squared errors.

- **Input:** An *n* x *n* distance matrix *D*.
- **Output:** A weighted tree *T* with *n* leaves minimizing *Discrepancy(T, D)* over all weighted trees with *n* leaves.

Least-Squares Phylogeny

NP– complete problem

Least-Squares Distance-Based Phylogeny Problem:

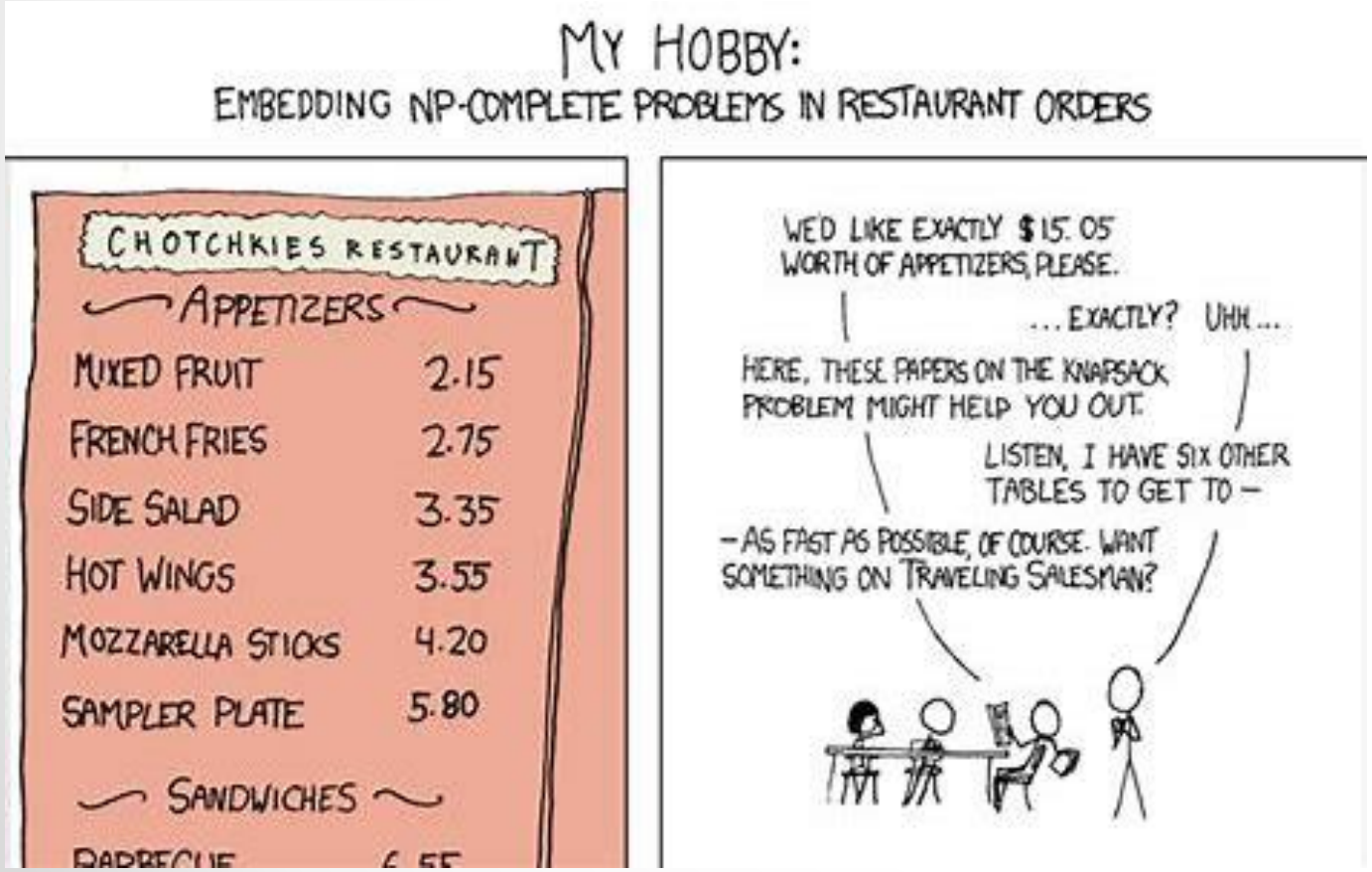
Given a distance matrix, find the tree that minimizes the sum of squared errors.

- **Input:** An *n* x *n* distance matrix *D*.
- **Output:** A weighted tree *T* with *n* leaves minimizing *Discrepancy(T, D)* over all weighted trees with *n* leaves.

In general there is a polynomial algorithm that will minimize the sum of squared errors if we are given the structure of the tree (T) in advance. But in practice it is not possible. So, we'll need to minimize the sum of squared errors over all possible tree structures, which are exponential numbers and it will not help to fit a tree to the non-additive matrix...

NP -complete problem

NP(Non-deterministic Polynomial time) is a set of all decision problems (question with yes-or-no answer) for which the 'yes'- answers can be verified in polynomial time ($O(n^k)$ where n is the problem size, and k is a constant) by a **deterministic Turing machine.** Polynomial time is sometimes used as the definition of fast or quickly.



The ways to solve the NP-complete problems:

- Approximation,
- Randomization,
- Parametrization,
- Heuristic search,
- etc…

Alan Turing (1912-1954)

Weakness of Distance-Based Methods

Distance-based algorithms for evolutionary tree reconstruction say nothing about ancestral states at internal nodes.

We lost information when we converted a multiple alignment to a distance matrix...

Species	Alignment	Distance Matrix			
		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

CLUSTERING METHODS AND ALGORITHMS IN GENOMICS AND EVOLUTION

Session 9.2

Character based methods for tree inference

Outline

- **Character-Based Tree Reconstruction**
- The Small Parsimony Problem
- The Large Parsimony Problem
- Tree thinking
- Evolutionary Tree Reconstruction in the Modern Era

Character Tables

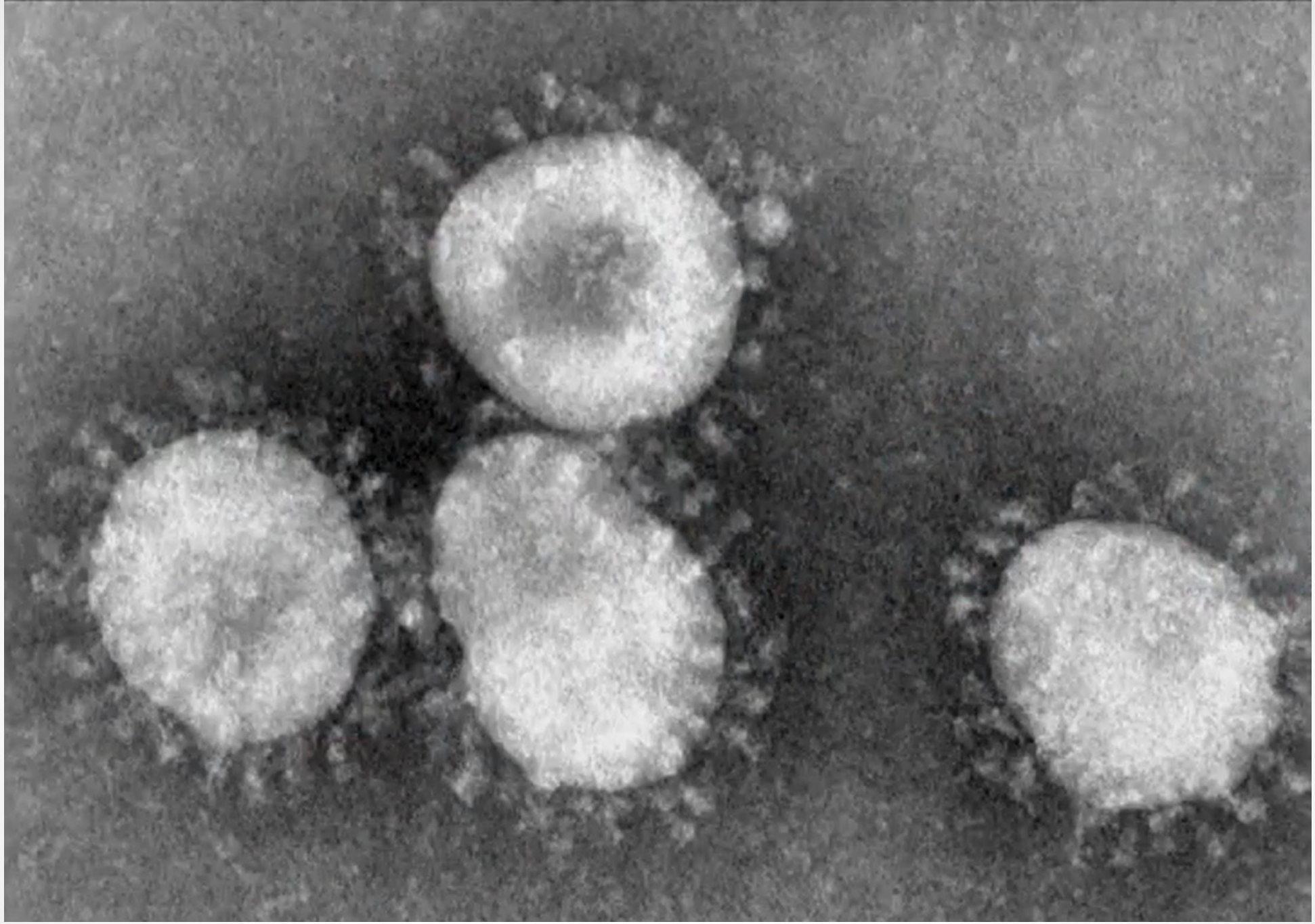
Fifty years ago, researcher constructed trees from anatomical/physiological properties called **characters.**

	Wings	Legs
Winged stick insect	Yes	6
Wingless stick insect	No	6
Giant cetipede	No	42

Character-Based Phylogeny

Character-Based Phylogeny Problem: *Reconstruct a phylogeny from characters.*

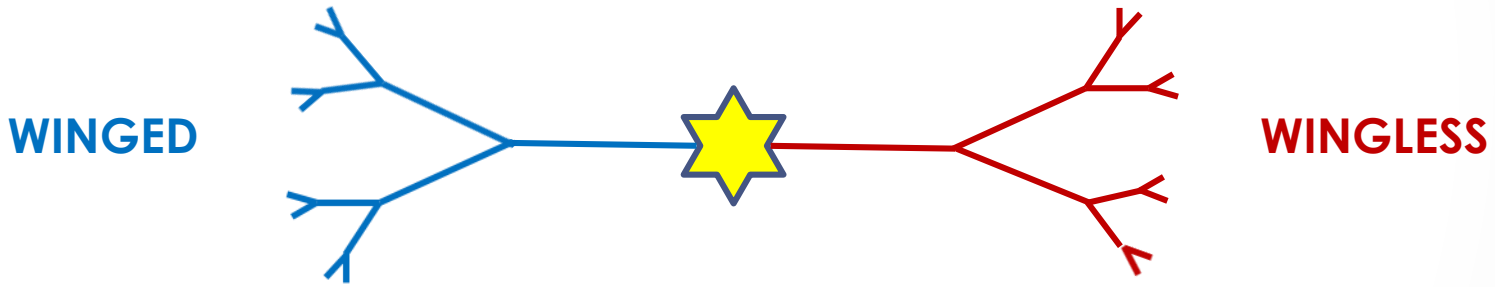
- **Input:** An *n* x *m* character table for n species and *m* characters.
- **Output:** A tree in which species with similar character values occur near each other.



Try to construct a coronavirus phylogeny from anatomical characters....

From Characters to a Phylogeny

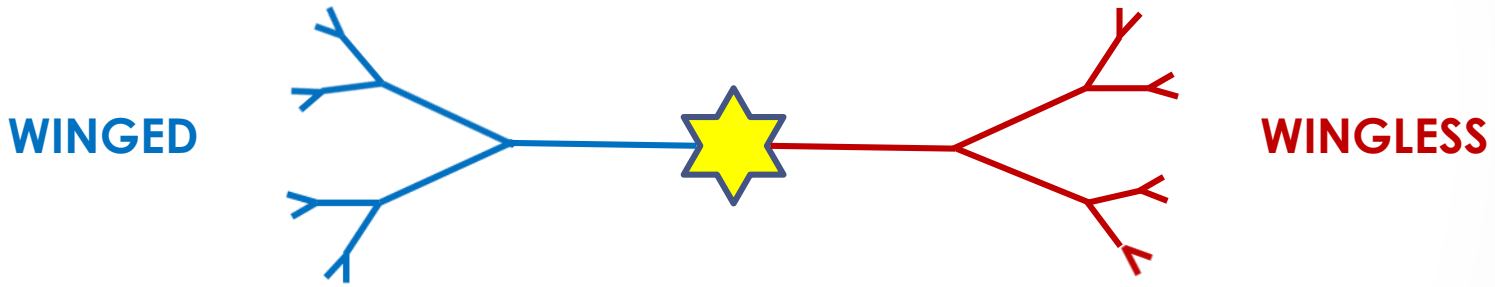
STOP and Think: How would you construct an evolutionary tree from character?



This strategy is completely reasonable.

From Characters to a Phylogeny

STOP and Think: How would you construct an evolutionary tree from character?

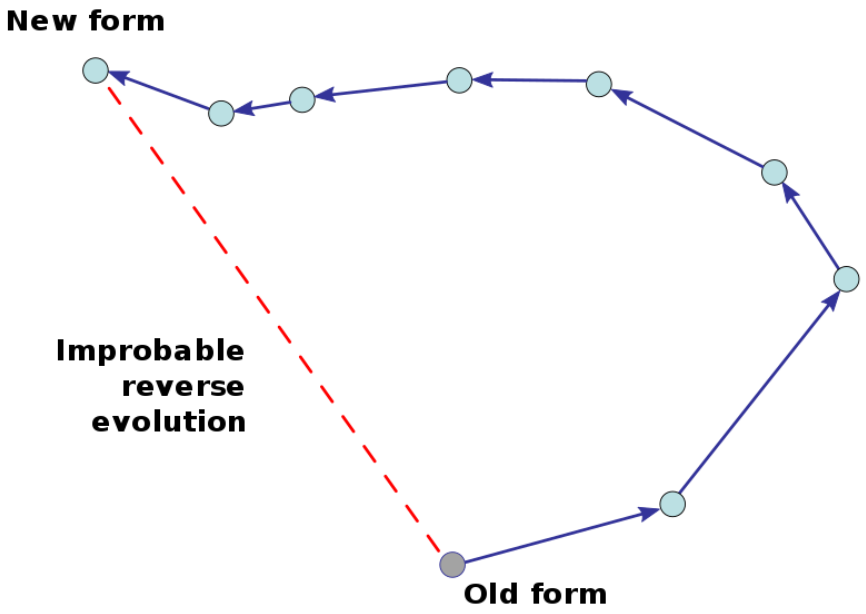


Dollo's principle of irreversibility (1893): evolution doesn't reinvent the same organ (e.g. insect wings).

Dollo's law of irreversibility

...an organism never returns exactly to a former state, even if it finds itself placed in conditions of existence identical to those in which it has previously lived ... it always keeps some trace of the intermediate stages through which it has passed.

Louis Dollo, 1893

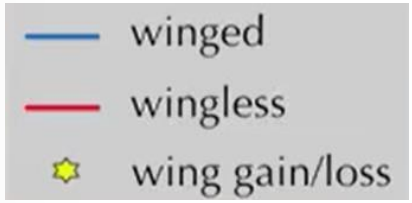


Once an organism has evolved in a certain way, it will not return exactly to a previous form. This is illustrated here in two dimensions; in reality, both biomolecules and organisms evolve in many different dimensions.

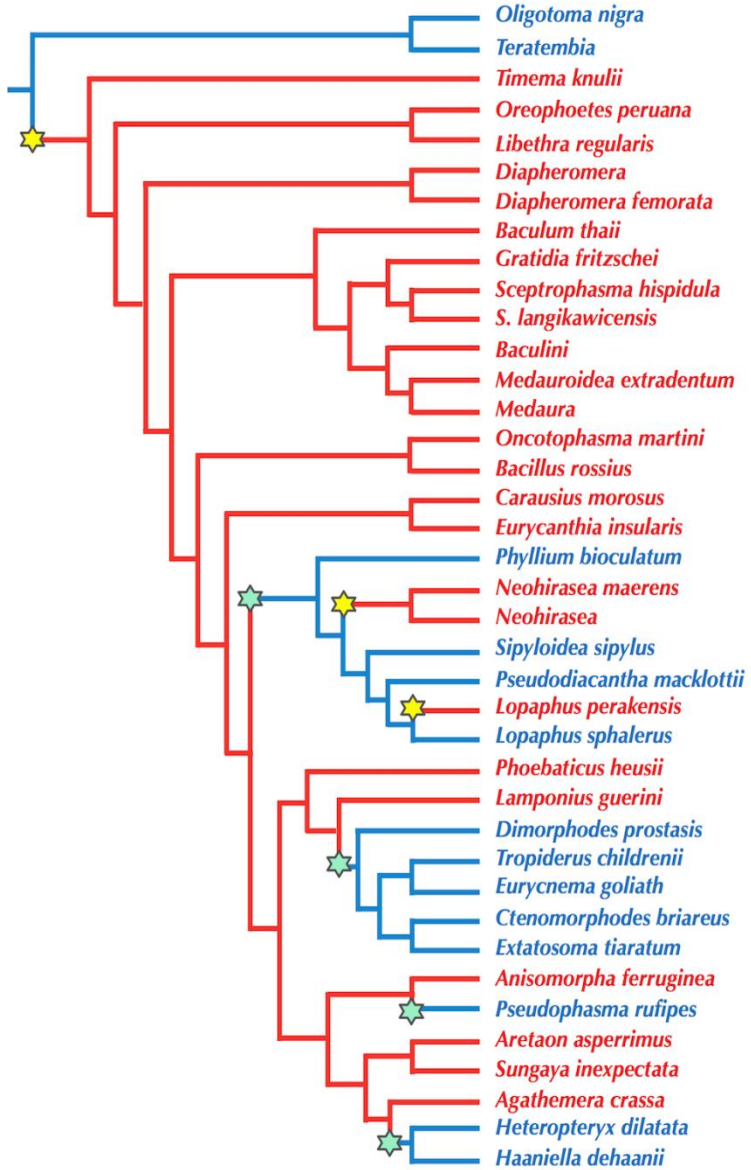
1857-1931

Dollo's Principle Violated

In 2003, Michael Whiting studied various winged and wingless stick insects from around the world and refuted this argument.



Wings were gained or lost at seven different times in stick insects alone, shown here by stars.

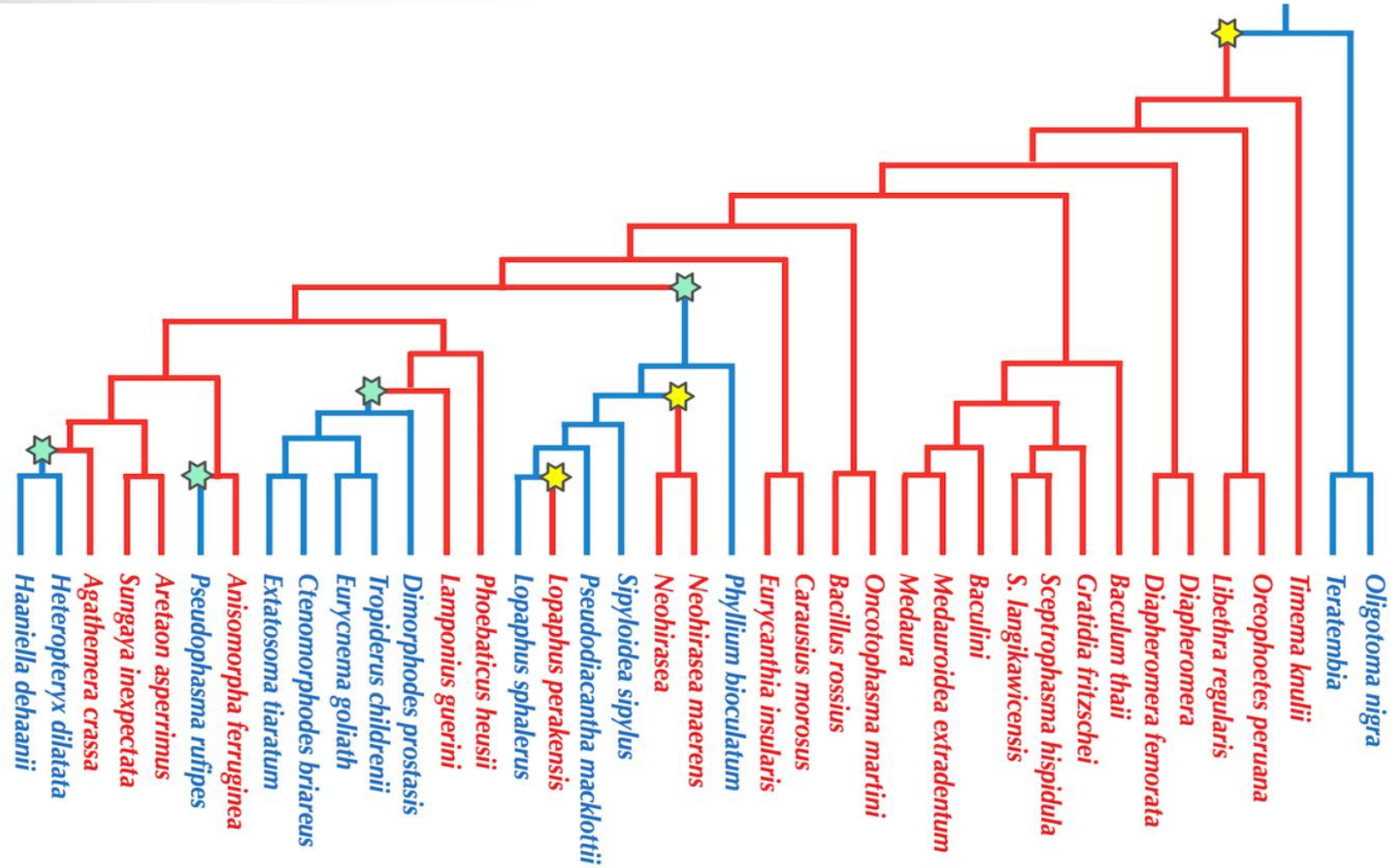


Evolutionary tree of **winged** and **wingless** stick insects constructed from 18S ribosomal RNA genes.

Transitions from winged to wingless species are shown by **yellow** stars; transitions from wingless to winged species are shown by **blue** stars.

18S ribosomal RNAs are slow-evolving, making them ideal for reconstructing ancient divergences.

Dollo's Principle Violated



Evolution isn't reinventing the wings from scratch.

The pathways needed for flight aren't being eliminated completely by eroding into pseudogenes.

They're just being used for something else in wingless species; then, they're getting turned back on when they're needed in a winged species to accommodate flight.

STOP and Think: What do you think has happened?

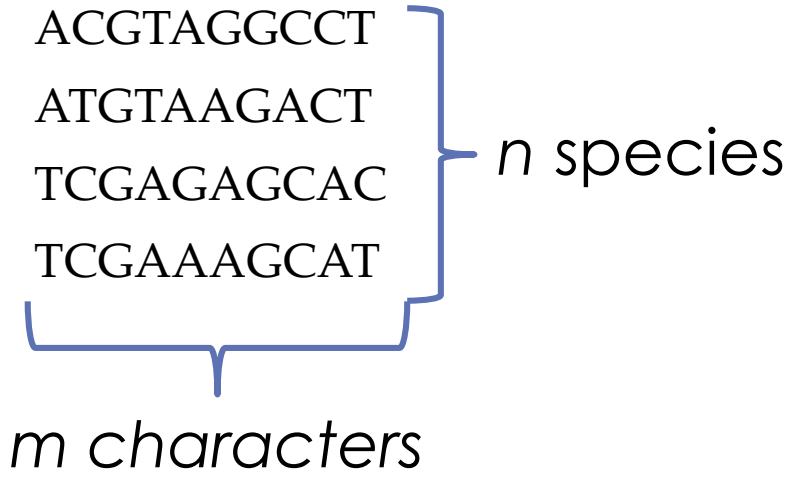
An Alignment As a Character Table

Species	Alignment
Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT

We can bypass such issues using genetic data as characters.

An Alignment As a Character Table

SPECIES	ALIGNMENT
Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT



The question is how to use this character table constructed from the multiple alignment, in order to reconstruct ancestral states.

Outline

- Character-Based Tree Reconstruction
- **The Small Parsimony Problem**
- The Large Parsimony Problem
- Evolutionary Tree Reconstruction in the Modern Era

Definition of Parsimony

- **Maximum parsimony** is an optimality criterion under which the phylogenetic tree that minimizes the total number of character-state changes is to be preferred:

the shortest possible tree that explains the data is considered best.

(J.S. Farris 1970; W.M. Fitch 1971)

- **Occam's razor** - a principle of theoretical parsimony: all else being equal—the simplest hypothesis that explains the data should be selected (William of Ockham, 1320s).

English Franciscan friar,
scholastic philosopher and theologian

Toward a Computational Problem

Species	Alignment
Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT

n species

m characters (4 values)

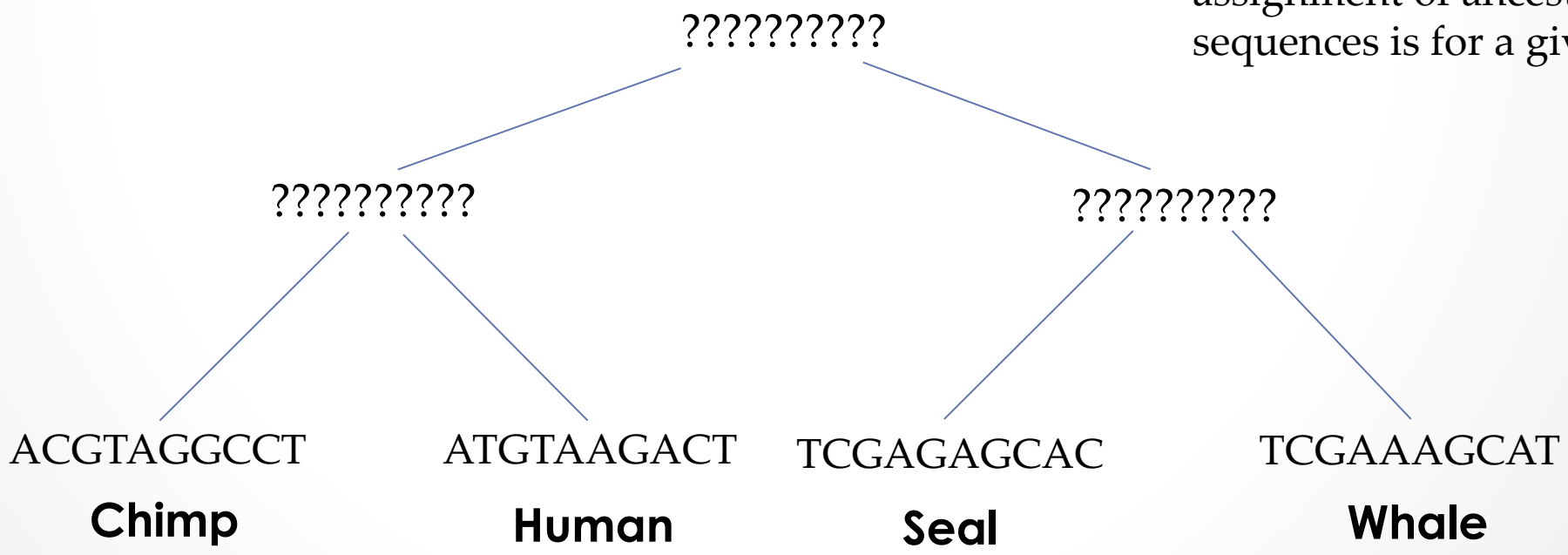
We can think about a multiple alignment as a character table of its own, for which each of the *m* columns of this multiple alignment can be viewed as a character.

Toward a Computational Problem

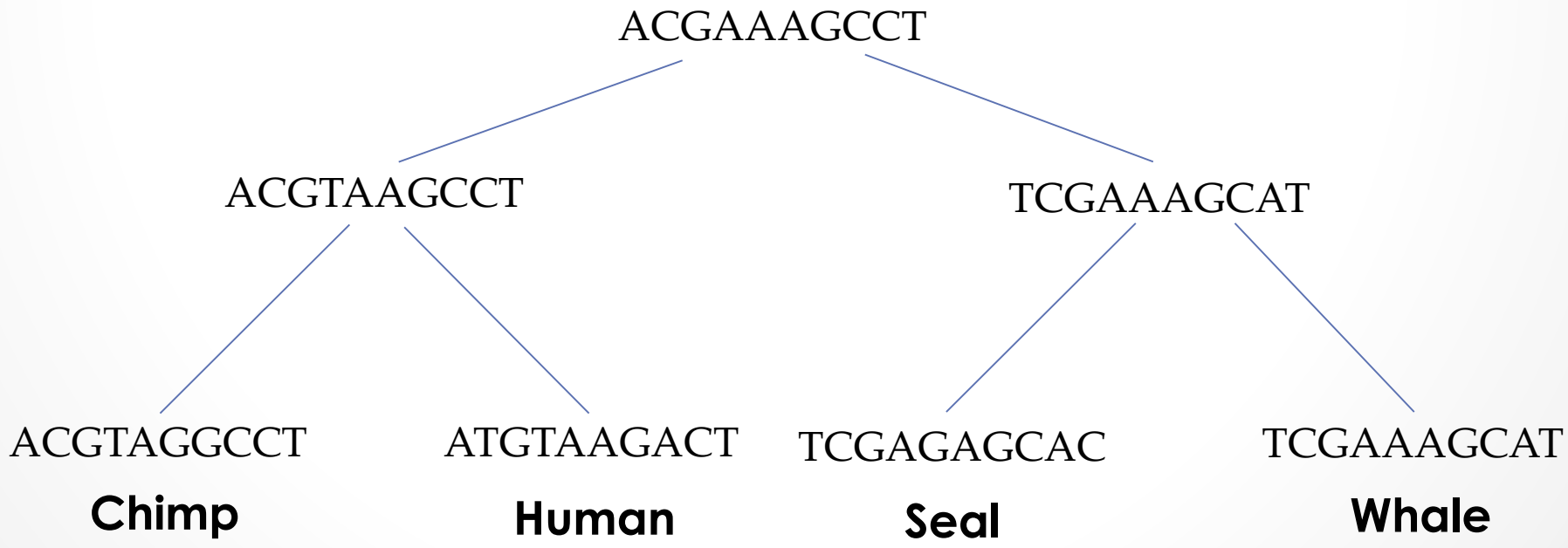
Species	Alignment
Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT

Our goal: using this multiple alignment, reconstruct the most likely ancestral sequences.

So, we need some way of determining how good an assignment of ancestral sequences is for a given tree.



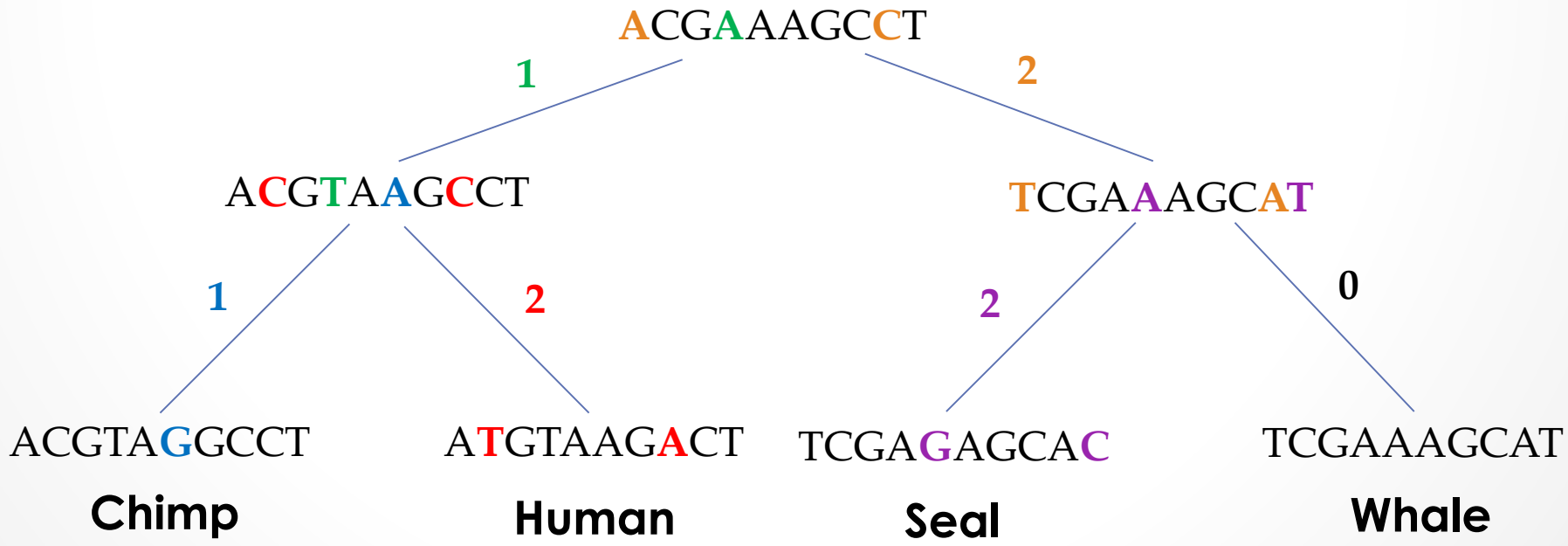
Toward a Computational Problem



Toward a Computational Problem

Parsimony score: sum of *Hamming distances* along each edge.

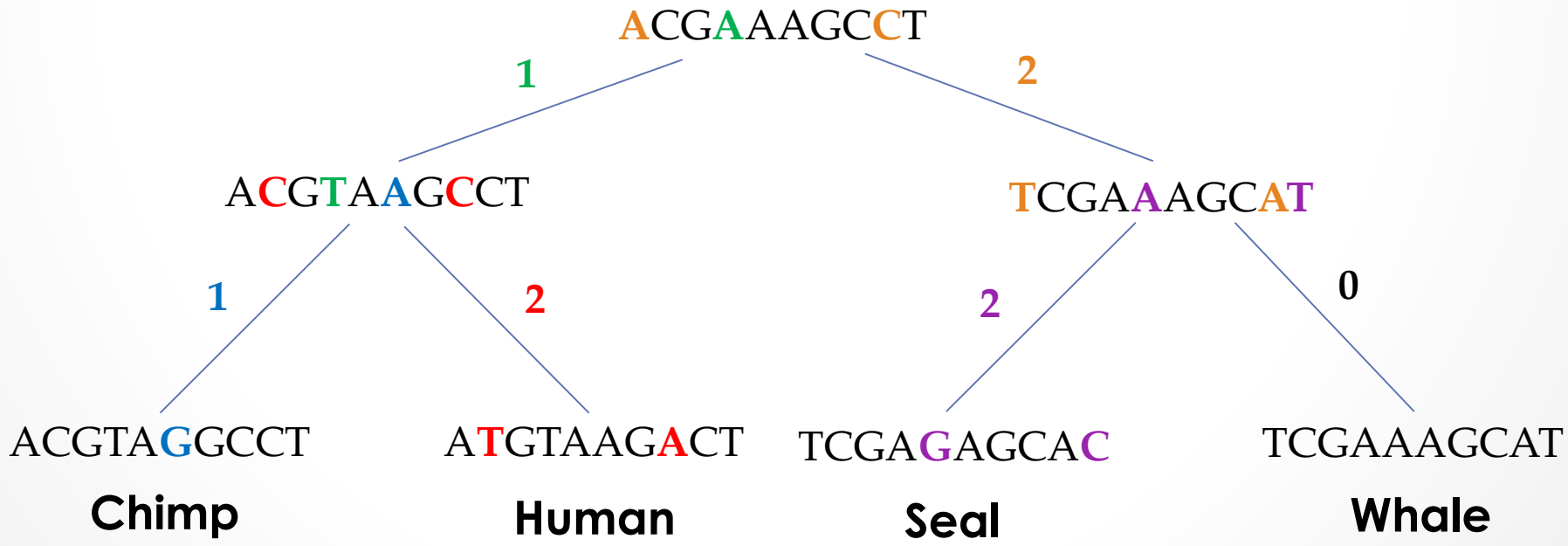
Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. It measures the minimum number of *substitutions* required to change one string into the other.



Toward a Computational Problem

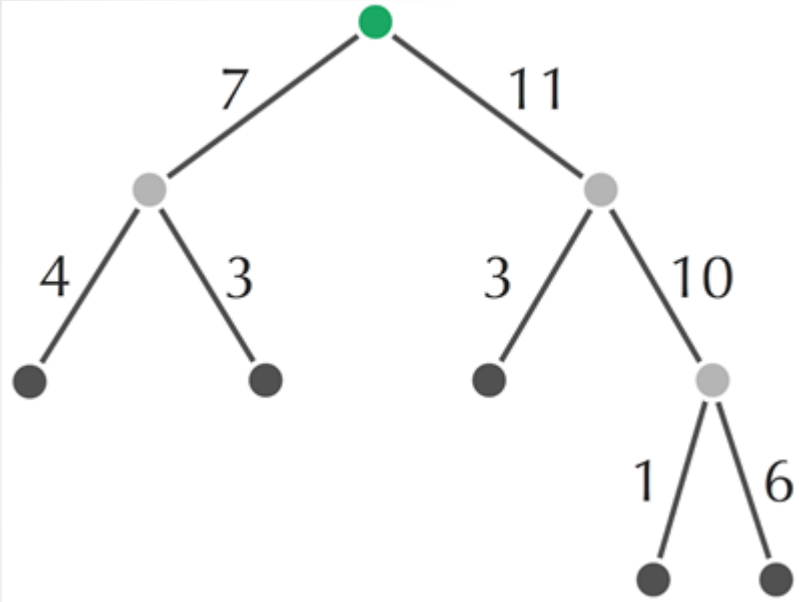
Parsimony score: sum of *Hamming distances* along each edge.

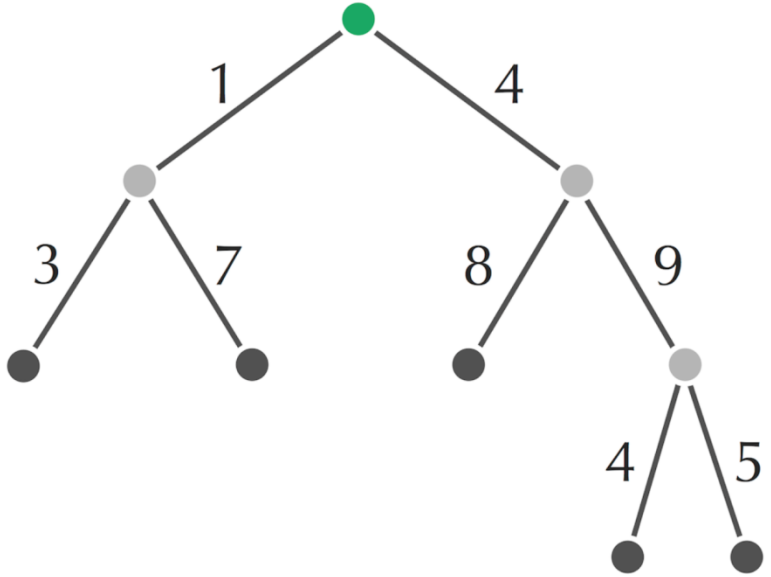
Parsimony score: 8

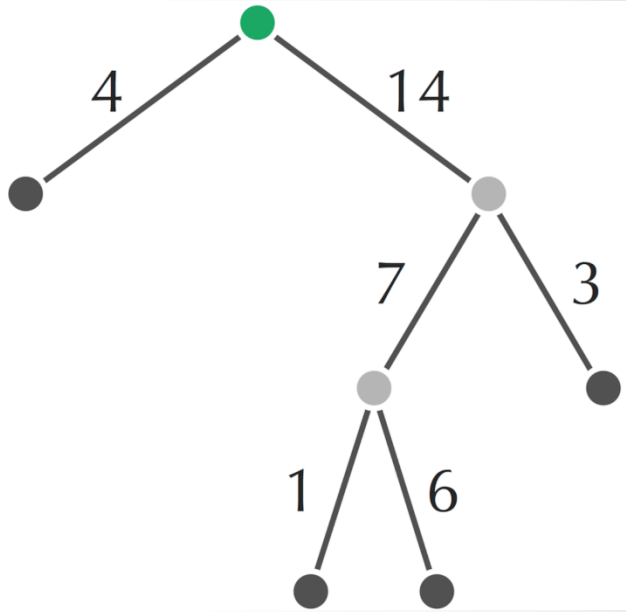


Exercise Break

Compute the parsimony score of the following trees.







Toward a Computational Problem

Small Parsimony Problem: *Find the most parsimonious labeling of the internal nodes of a rooted tree.*

- **Input:** A rooted binary tree with each leaf labeled by a string of length *m*.
- **Output:** A labeling of all other nodes of the tree by string of length *m* that minimizes the tree's parsimony score.

Our goal is to assign **ancestral sequences** to this tree in order to minimize the parsimony score.

Toward a Computational Problem

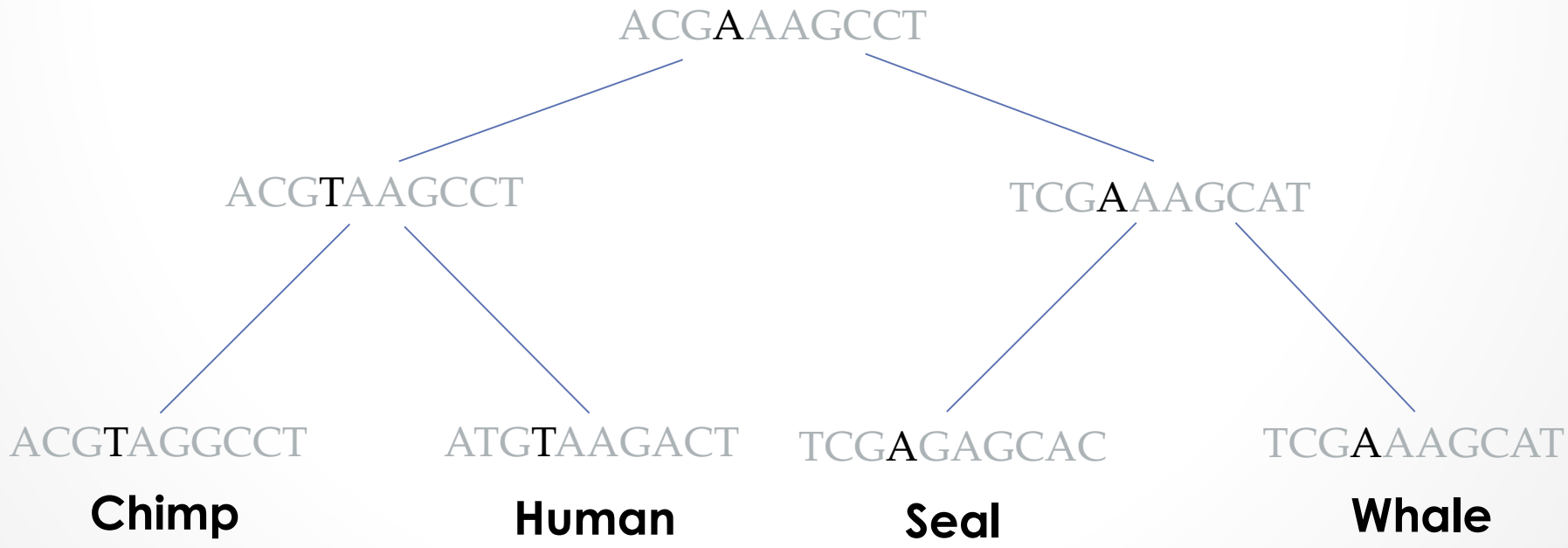
Small Parsimony Problem: *Find the most parsimonious labeling of the internal nodes of a rooted tree.*

- **Input:** A rooted binary tree with each leaf labeled by a **single symbol**.
- **Output:** A labeling of all other nodes of the tree by **single symbols** that minimizes the tree's parsimony score.

If we treat the columns of the multiple alignment that we're given as **independent** of each other, then it means that we'll be able to solve the small parsimony problem on **each column** of the multiple alignment **separately**.

Toward a Computational Problem

We can work with one symbol of each string at a time.

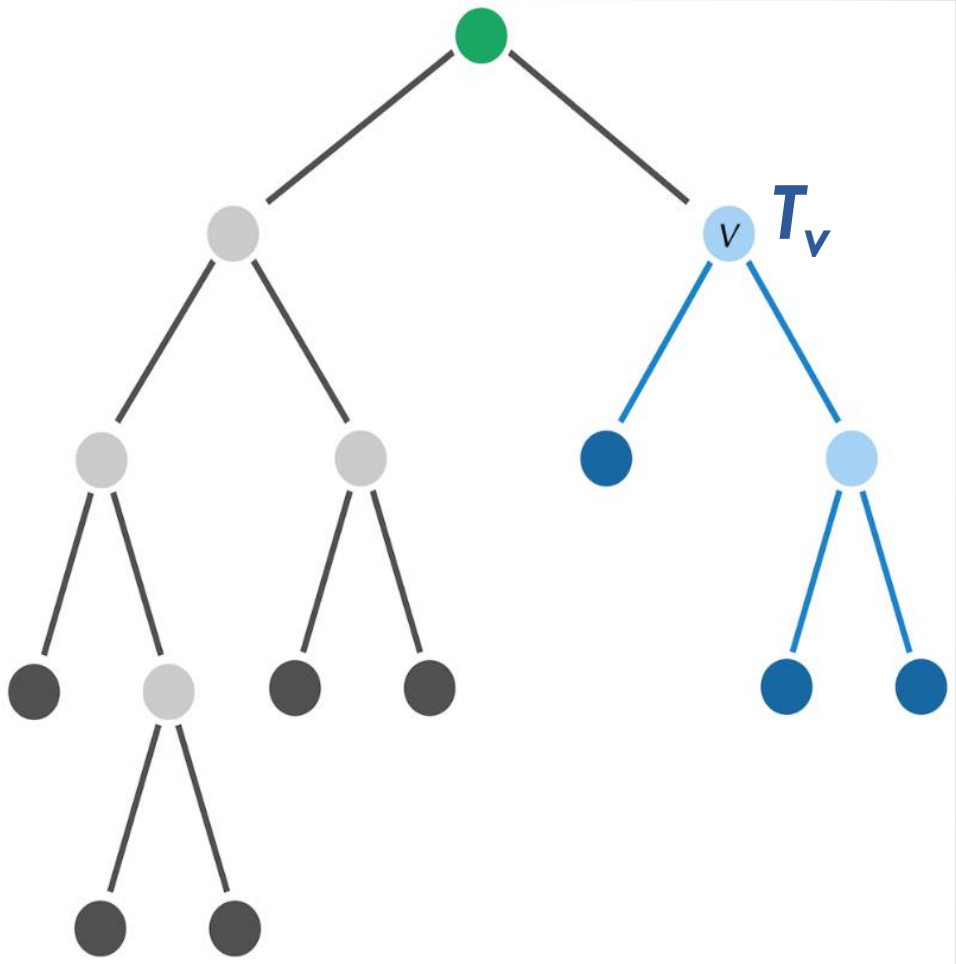


A Dynamic Programming Algorithm

Let T_v denote the subtree of T whose root is v.

Define $s_k(v)$ as the minimum parsimony score of T_v over all labelings of T_v, assuming that v is labeled by k.

The minimum parsimony score for the tree T is equal the minimum value of $s_k(root)$ over all symbols k.



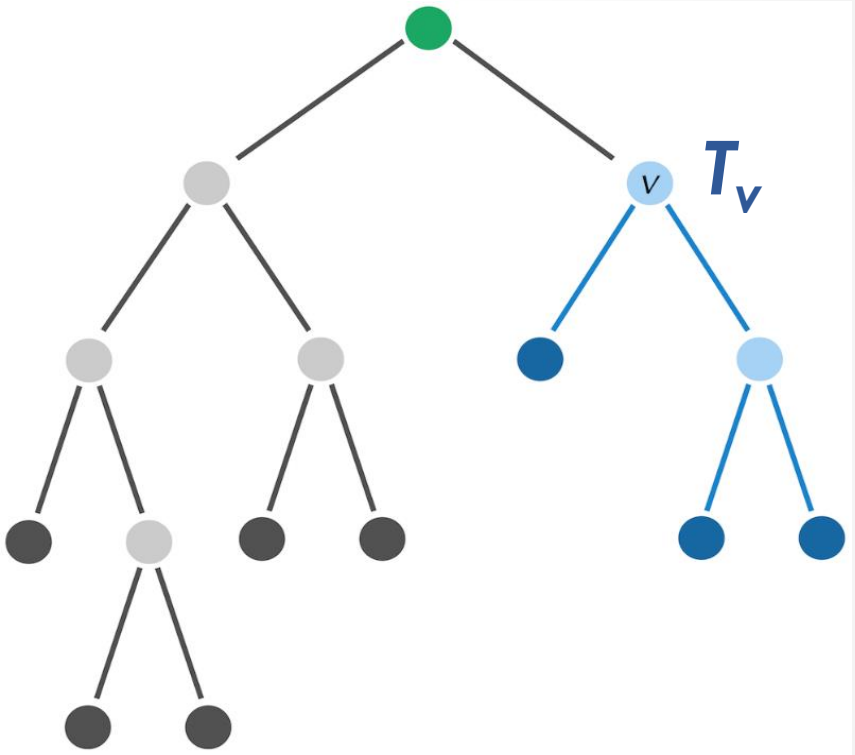
- The idea of dynamic programming algorithm: starting at the leaves of the tree, compute the values s_k upward from the leaves to the root, and then use this fact to find the minimum parsimony score.

A Dynamic Programming Algorithm

We need a recurrence relation for the score of a node ***v*** in terms of its two "children".

For symbols *i* and *j*, define

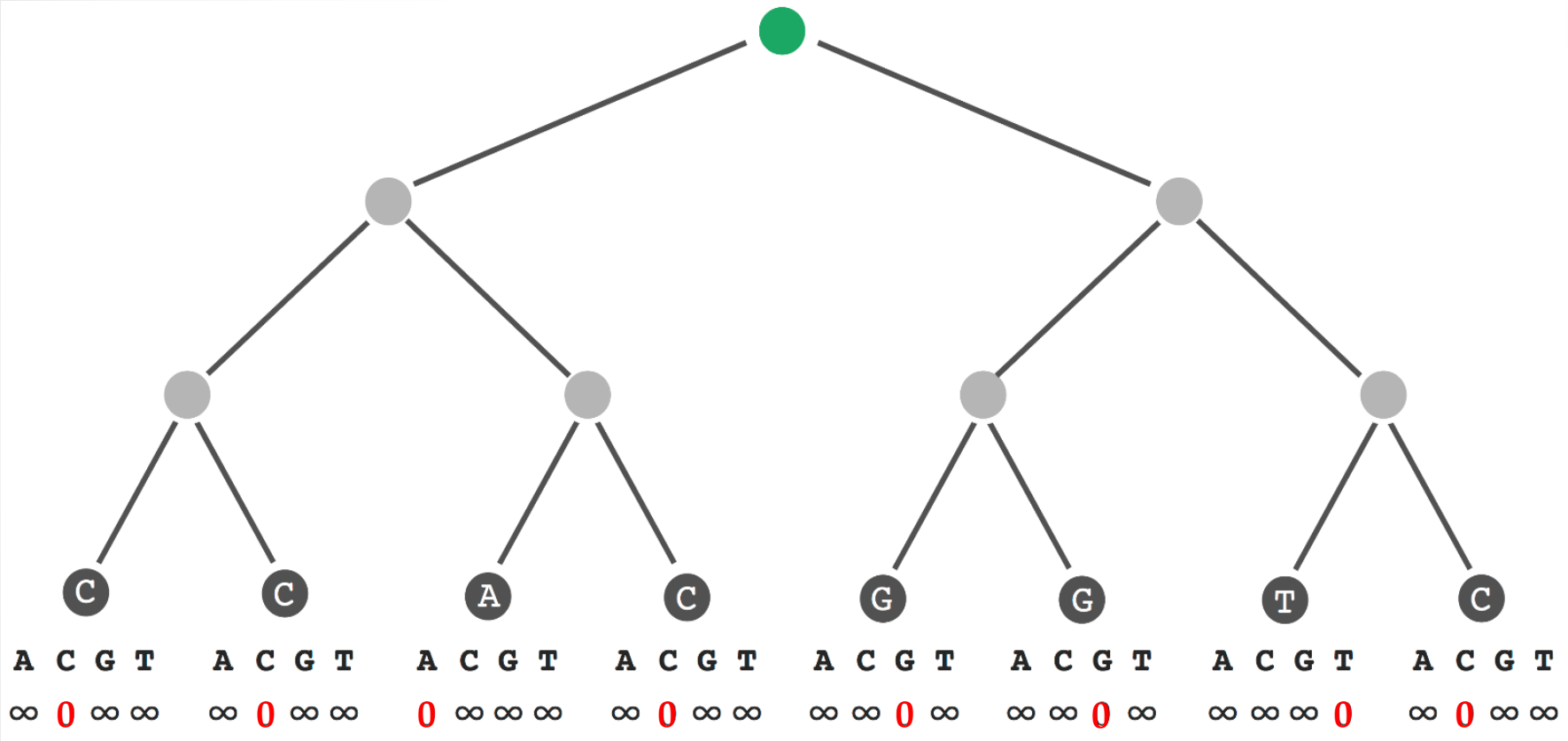
- $\delta_{i,j} = 0$ if $i = j$
- $\delta_{i,j} = 1$ otherwise.



Recurrence relation:

$$s_k(v)=\min_{\text{all symbols } i}\{s_i(Daughter(v))+\delta_{i,k}\}+\min_{\text{all symbols } j}\{s_j(Son(v))+\delta_{j,k}\}$$

A Dynamic Programming Algorithm



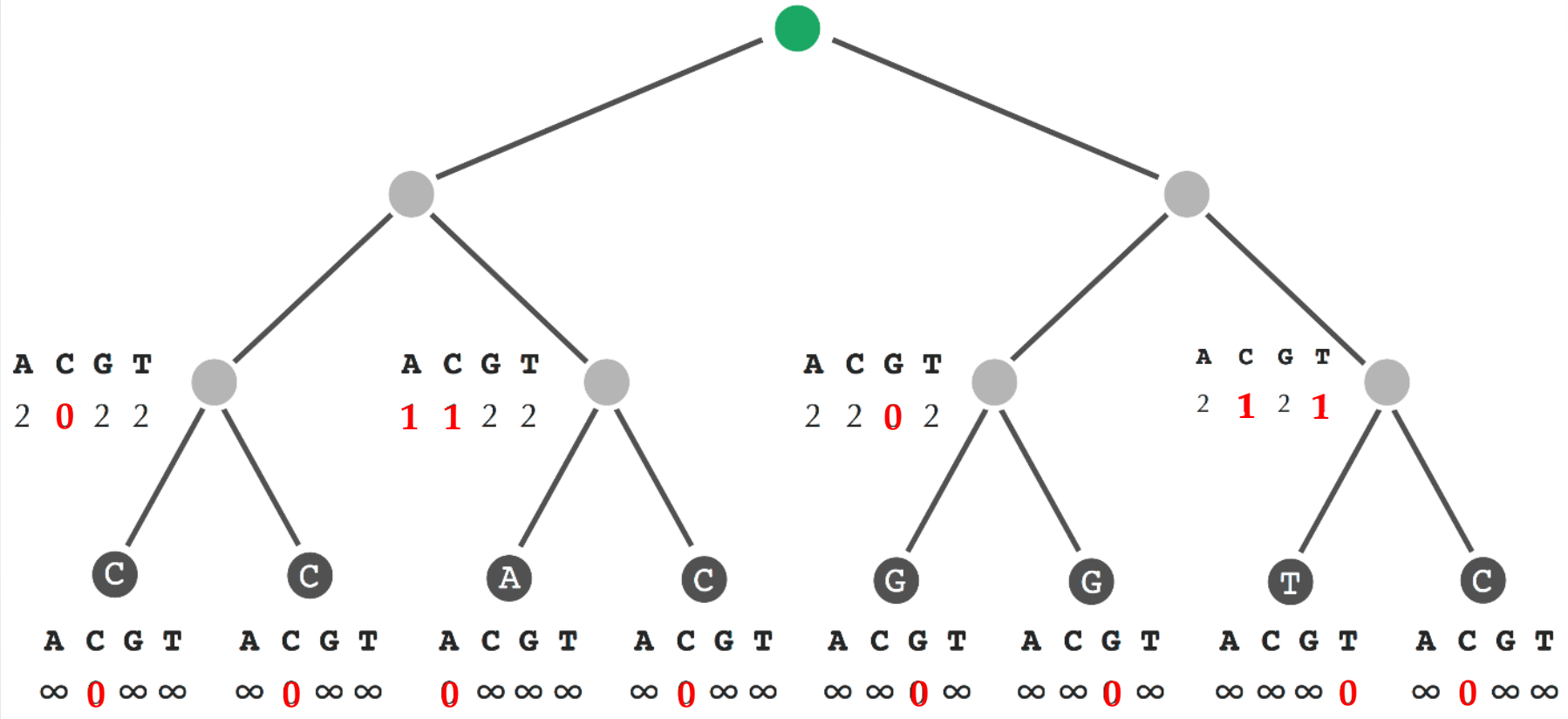
Initializing values $s_k(v)$ for all leaves v, represented below as an array for each leaf.

We set $s_k(v)$ equal to zero if the leaf is labeled by symbol k; otherwise, we set $s_k(v)$ equal to infinity.

At each node v, an array is going to hold the set of scores $s_k(v)$ for each possible symbol k. Here we're working with DNA, so there will be four symbols k.

$$s_k(v)=\min_{\text{all symbols } i}\{s_i(Daughter(v))+\delta_{i,k}\}+\min_{\text{all symbols } j}\{s_j(Son(v))+\delta_{j,k}\}$$

A Dynamic Programming Algorithm



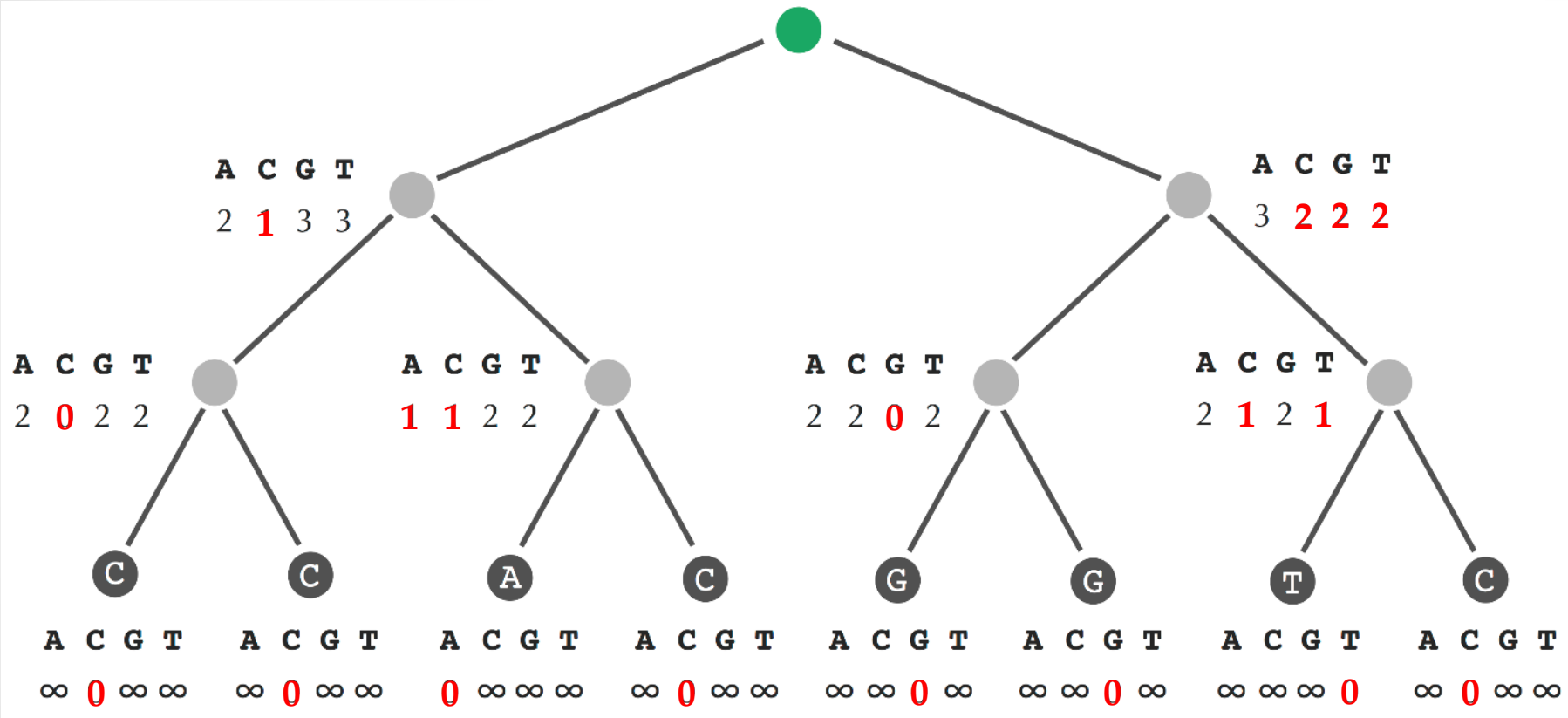
Moving up a level, we apply the recurrence relation.

In the left most node, the two children are fixed as equal to C. So the score of this node is going to be 0 if we assign it a C, since there will be no conflicts between these two edges.

Red indicates the minimum score at each step.

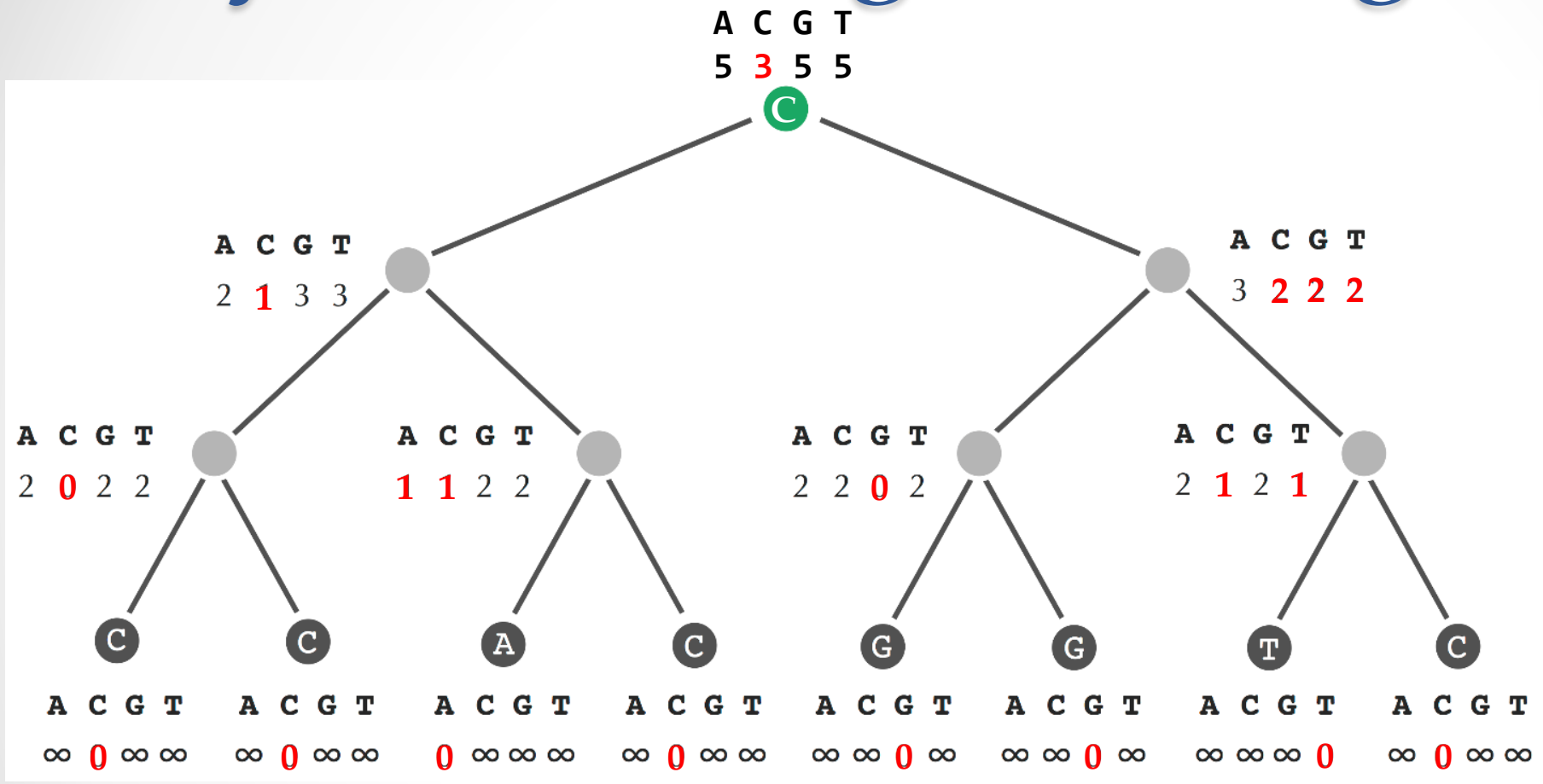
$$s_k(v)=\min_{\text{all symbols } i}\{s_i(Daughter(v))+\delta_{i,k}\}+\min_{\text{all symbols } j}\{s_j(Son(v))+\delta_{j,k}\}$$

A Dynamic Programming Algorithm



$$s_k(v)=\min_{\text{all symbols } i}\{s_i(Daughter(v))+\delta_{i,k}\}+\min_{\text{all symbols } j}\{s_j(Son(v))+\delta_{j,k}\}$$

A Dynamic Programming Algorithm



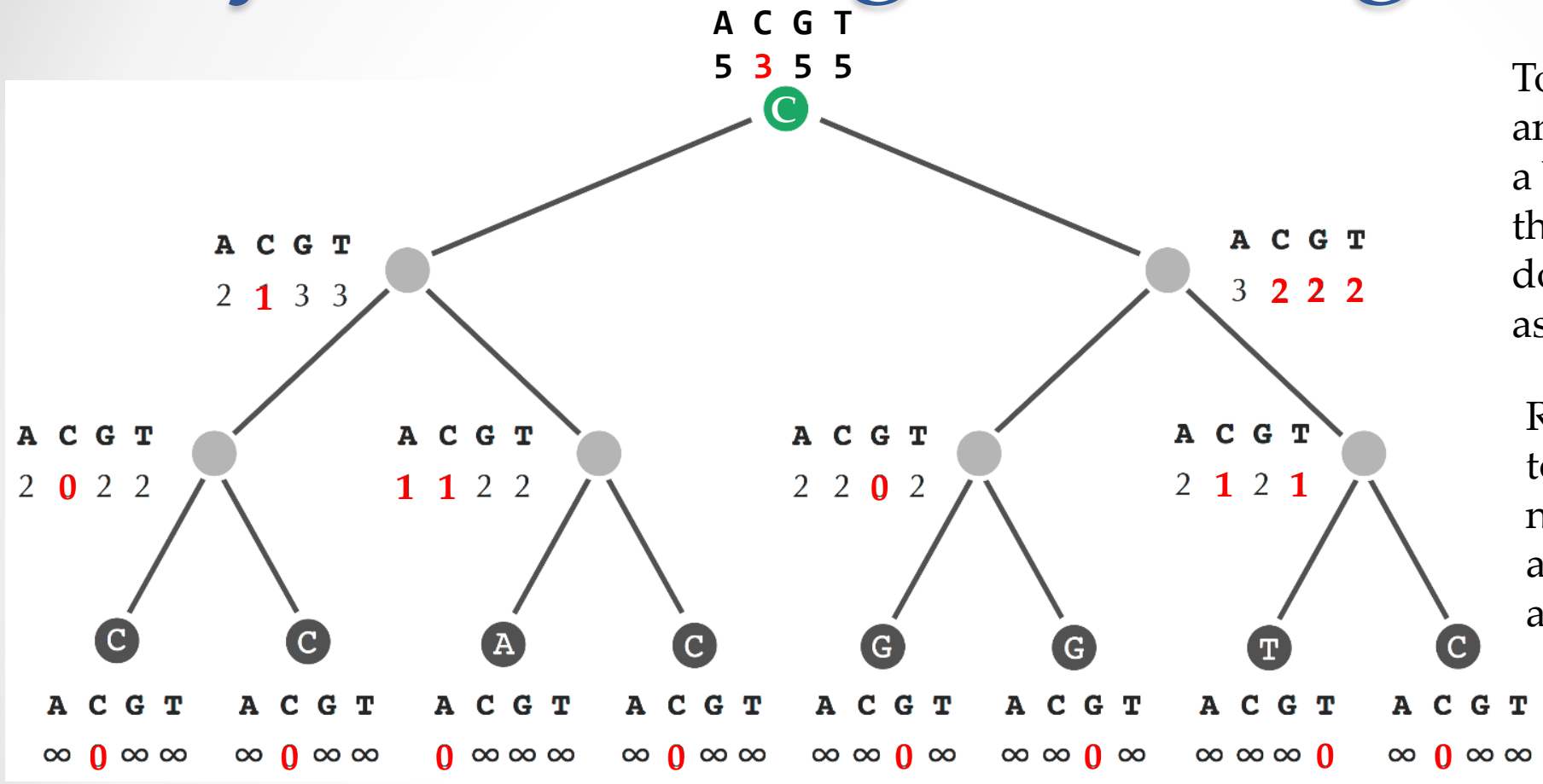
An illustration of **Small Parsimony** after initialization.

The min parsimony score is equal to the min score at the root, which for this tree is equal to 3.

This value corresponds to symbol C, and so when we begin backtracking to assign symbols to internal nodes, we assign nucleotide C to the root.

$$s_k(v)=\min_{\text{all symbols } i}\{s_i(Daughter(v))+\delta_{i,k}\}+\min_{\text{all symbols } j}\{s_j(Son(v))+\delta_{j,k}\}$$

A Dynamic Programming Algorithm



To reconstruct the other ancestral states implement a backtracking approach that goes from the root down to the leaves and assign symbols.

Red elements correspond to minima, but they don't necessarily correspond to an optimal assignment of ancestral symbols.

Exercise break: "Backtrack" to fill in remaining nodes of the tree.

Small Parsimony for Unrooted Trees

Small Parsimony in an Unrooted Tree Problem: *Find the most parsimonious labeling of the internal nodes of an unrooted tree.*

- **Input:** An unrooted binary tree with each leaf labeled by a string of length *m*.
- **Output:** A position of the root and a labeling of all other nodes of the tree by string of length *m* that minimizes the tree's parsimony score.

You can find the most parsimonious labeling of the internal nodes in an unrooted tree by placing a root on an arbitrary edge, solving the small parsimony problem for the resulting rooted tree, and then simply removing the root.

Outline

- Character-Based Tree Reconstruction
- The Small Parsimony Problem
- **The Large Parsimony Problem**
- Tree Thinking
- Evolutionary Tree Reconstruction in the Modern Era

Finding the Most Parsimonious Tree

Parsimony score: 8

Human

ATGTAAGACT

2

Seal

TCGAGAGCAC

2

3

ACGTAAGCCT

TCGAAAGCAT

1

ACGTAGGCCT

Chimp

Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT

0

TCGAAAGCAT

Whale

Dynamic programming algorithm is able to reconstruct ancestral states in binary trees. But this algorithm assumes that the structure of the tree was given.

Finding the Most Parsimonious Tree

Parsimony score: 11

Seal

TCGAGAGCAC

4

ACGTAGGCCT

Chimp

2

ACGTAAGCAT

0

ACGTAAGCAT

3

Human

ATGTAAGACT

2

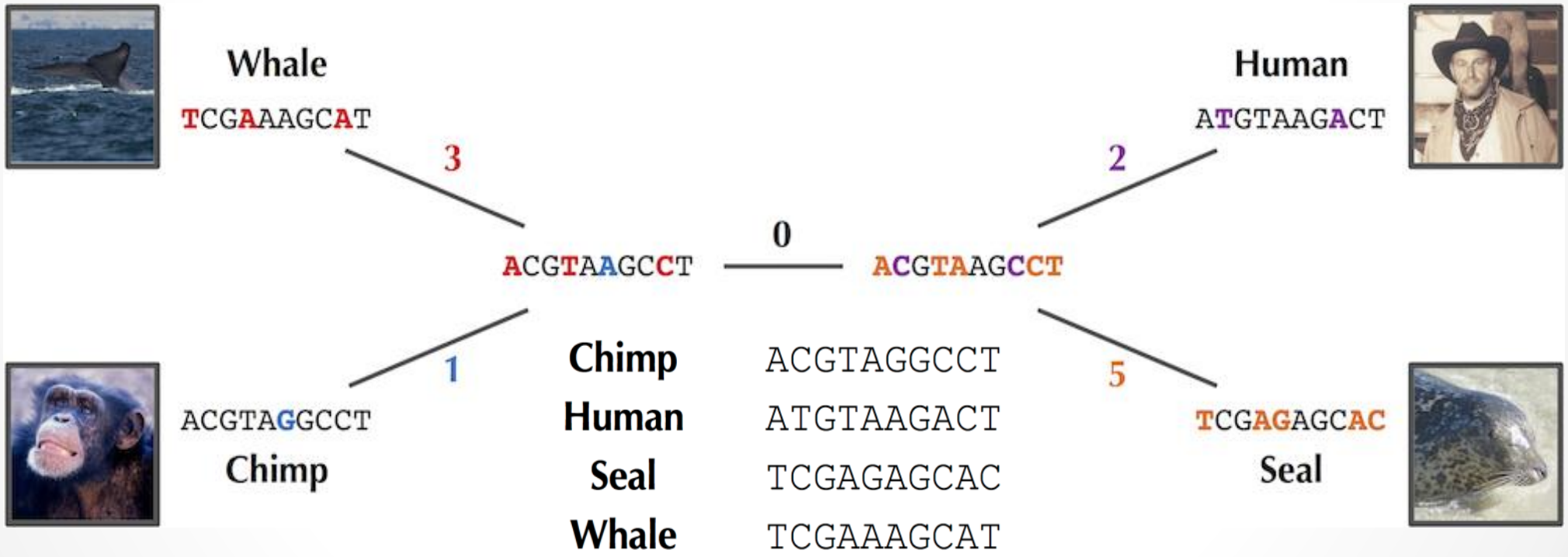
TCGAAAGCAT

Whale

Chimp	ACGTAGGCCT
Human	ATGTAAGACT
Seal	TCGAGAGCAC
Whale	TCGAAAGCAT

Finding the Most Parsimonious Tree

Parsimony score: 11



- When applying the algorithm for small parsimony, it is allowed an internal edge to have weight 0. Because if we were to compress this edge, then we would no longer have a binary tree.

Finding the Most Parsimonious Tree

Large Parsimony Problem: *Given a set of strings, find a tree (with leaves labeled by all these strings) having minimum parsimony score.*

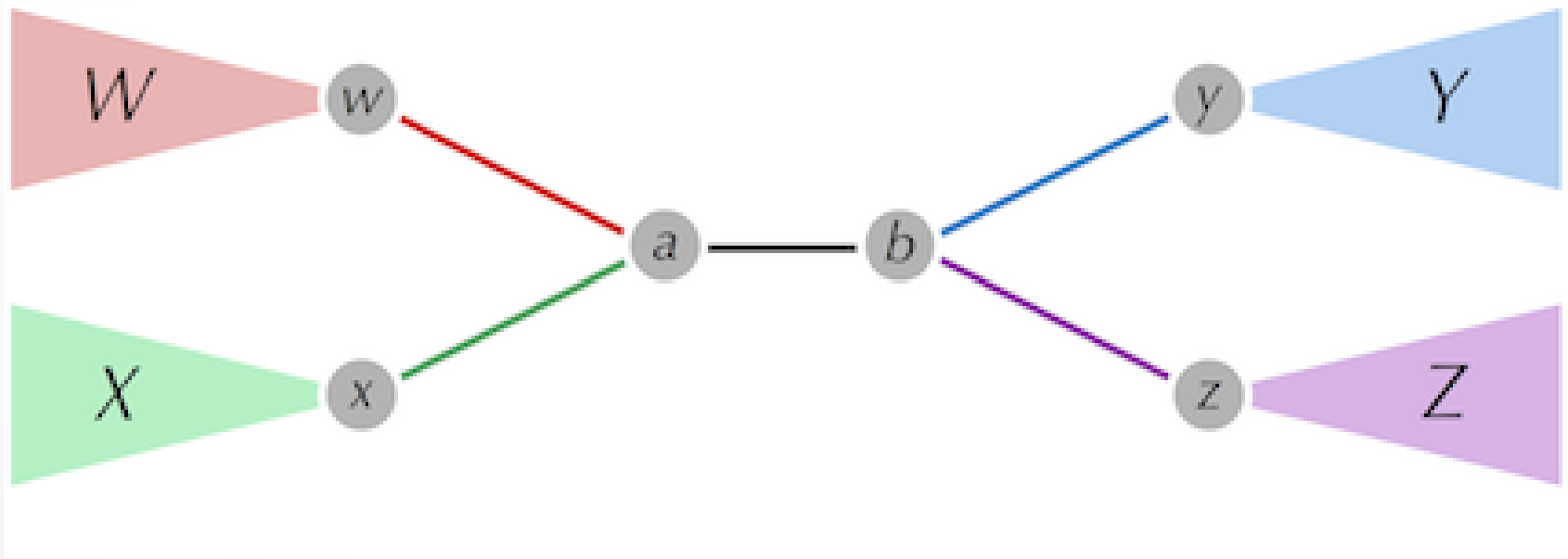
- **Input:** A collection of strings of equal length.
- **Output:** An unrooted binary tree T minimizing the parsimony score among all possible unrooted binary trees with leaves labeled by these strings.

Unfortunately, this problem is NP-Complete…

So, let's design a greedy heuristic for this problem.

A Greedy Heuristic for Large Parsimony

Removing an **internal edge**, an edge connecting two internal nodes (along with the nodes), produces four disconnected subtrees (W, X, Y, Z).

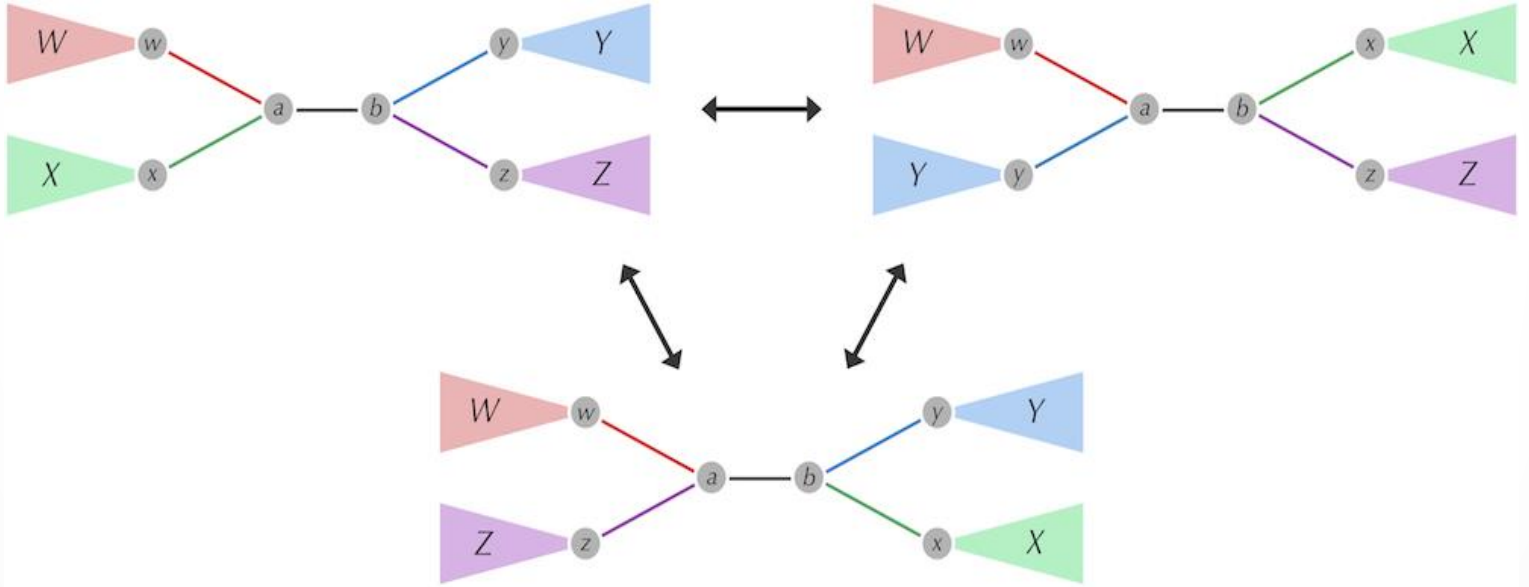


A Greedy Heuristic for Large Parsimony

Removing an **internal edge**, an edge connecting two internal nodes (along with the nodes), produces four disconnected subtrees (W, X, Y, Z).

A Greedy Heuristic for Large Parsimony

Rearranging these subtrees is called a **nearest neighbor interchange**.



A nearest neighbor interchange on the internal edge (*a*,*b*), shown in black, results from rearranging the four colored subtrees *W*, *X*, *Y*, and *Z*, which are rooted at *w*, x, *y*, and z, respectively. The nearest neighbor interchange operation removes one edge connected to *a* and another edge connected to *b*, then replaces these edges with two new edges. The three possible tree structures resulting from nearest neighbor interchanges on (*a*, *b*) can be represented as *WX*|*YZ* (top left), *WY*|*XZ* (top right), and *WZ*|*XY* (bottom).

A Greedy Heuristic for Large Parsimony

Nearest Neighbors of a Tree Problem: *Given an internal edge in an unrooted binary tree, generate the tree's nearest neighbors .*

- **Input:** An internal edge in an unrooted binary tree.
- **Output:** The two nearest neighbors of this tree (with respect to the given internal edge).

A Greedy Heuristic for Large Parsimony

Nearest Neighbors Interchange Heuristic:

1. Set current tree equal to arbitrary unrooted binary tree structure.

2. Go through all internal edges and perform all possible nearest neighbor interchange.

3. Solve Small Parsimony Problem on each tree.

4. If any tree has parsimony score improving over optimal tree, set it equal to the current tree.

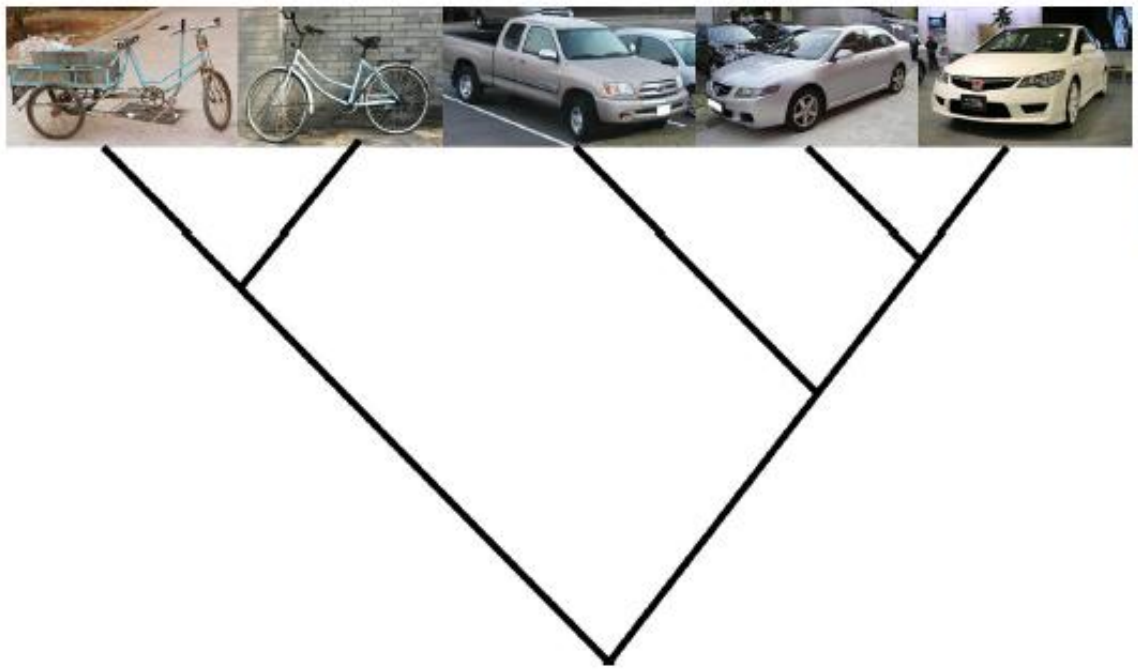
   Otherwise, return the current tree.

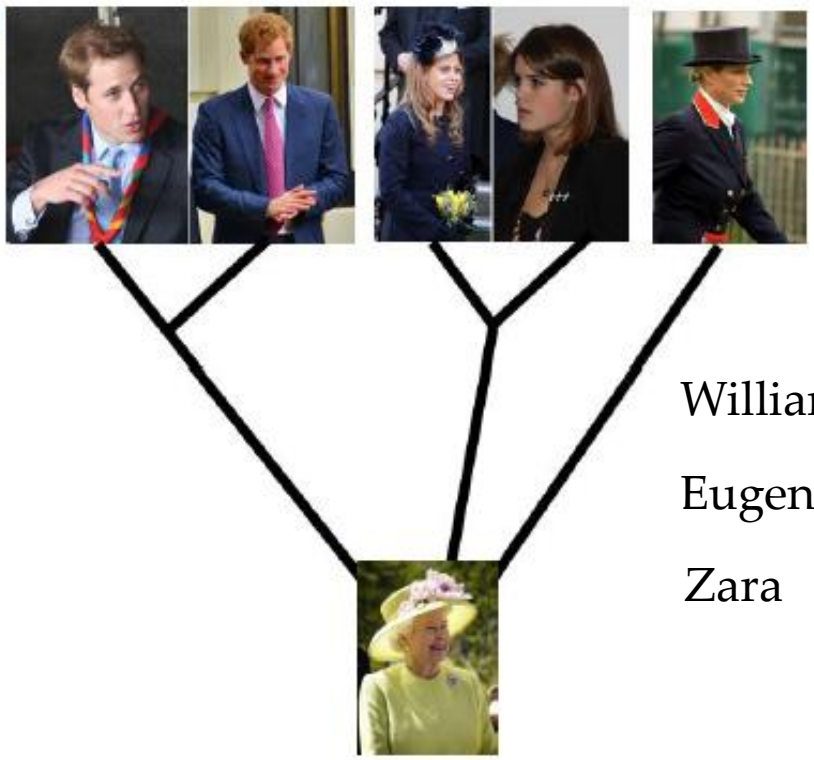
Outline

- Character-Based Tree Reconstruction
- The Small Parsimony Problem
- The Large Parsimony Problem
- **Tree Thinking**
- Evolutionary Tree Reconstruction in the Modern Era

Two (related) meanings of trees

- Groupings of similarity/ classification
- Ancestry relationships





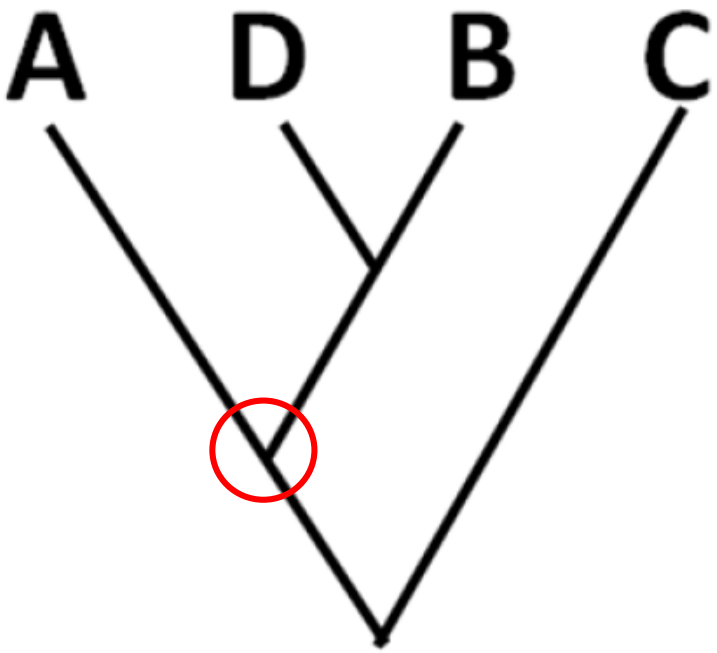
William and Harry

Eugenie and Beatrice

Zara

Queen Elizabeth II

Understanding trees

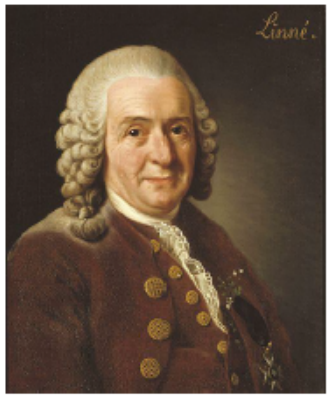


Which species is/ are most closely related to "A"?

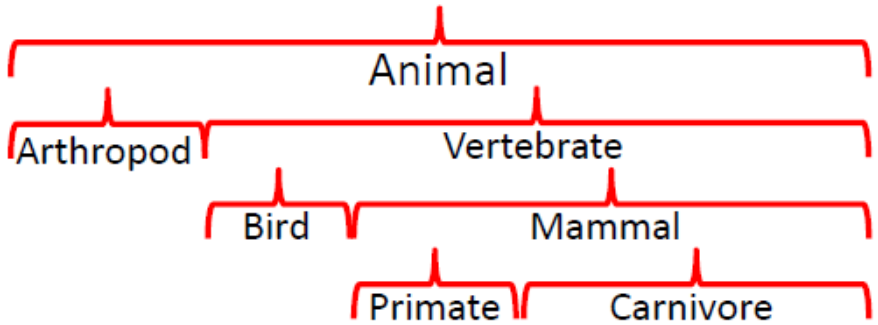
Classification groups imply closer evolutionary relationships

Most general → Most specific

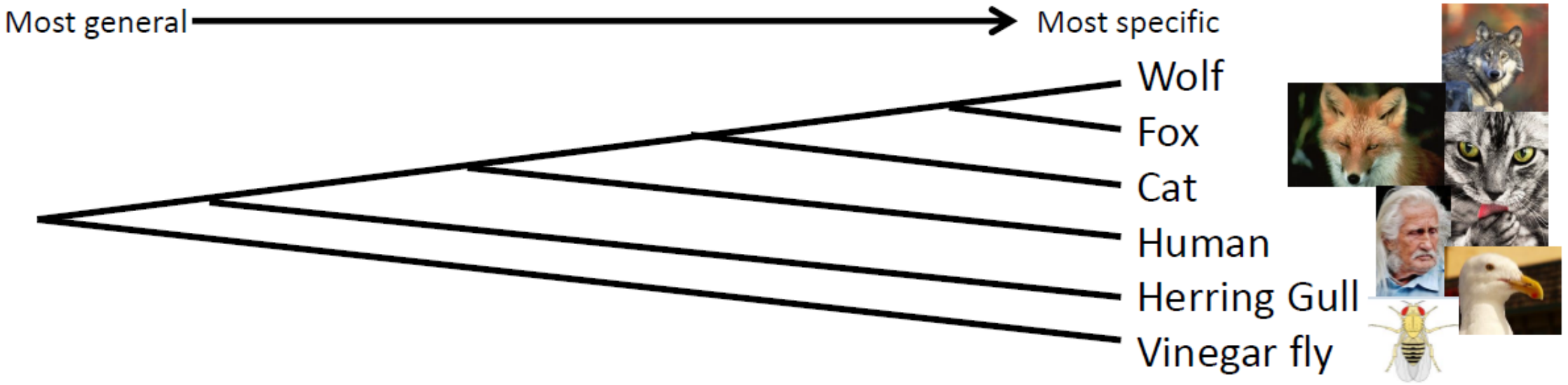
Animal –	Vertebrate –	Mammal –	Carnivore –	Canine –	Wolf
Animal –	Vertebrate –	Mammal –	Carnivore –	Canine –	Fox
Animal –	Vertebrate –	Mammal –	Carnivore –	Feline –	Cat
Animal –	Vertebrate –	Mammal –	Primate –	Hominid –	Human
Animal –	Vertebrate –	Bird –	Seabird –	Gull –	Herring Gull
Animal –	Arthropod –	Insect –	Fly –	Drosophilid –	Vinegar fly



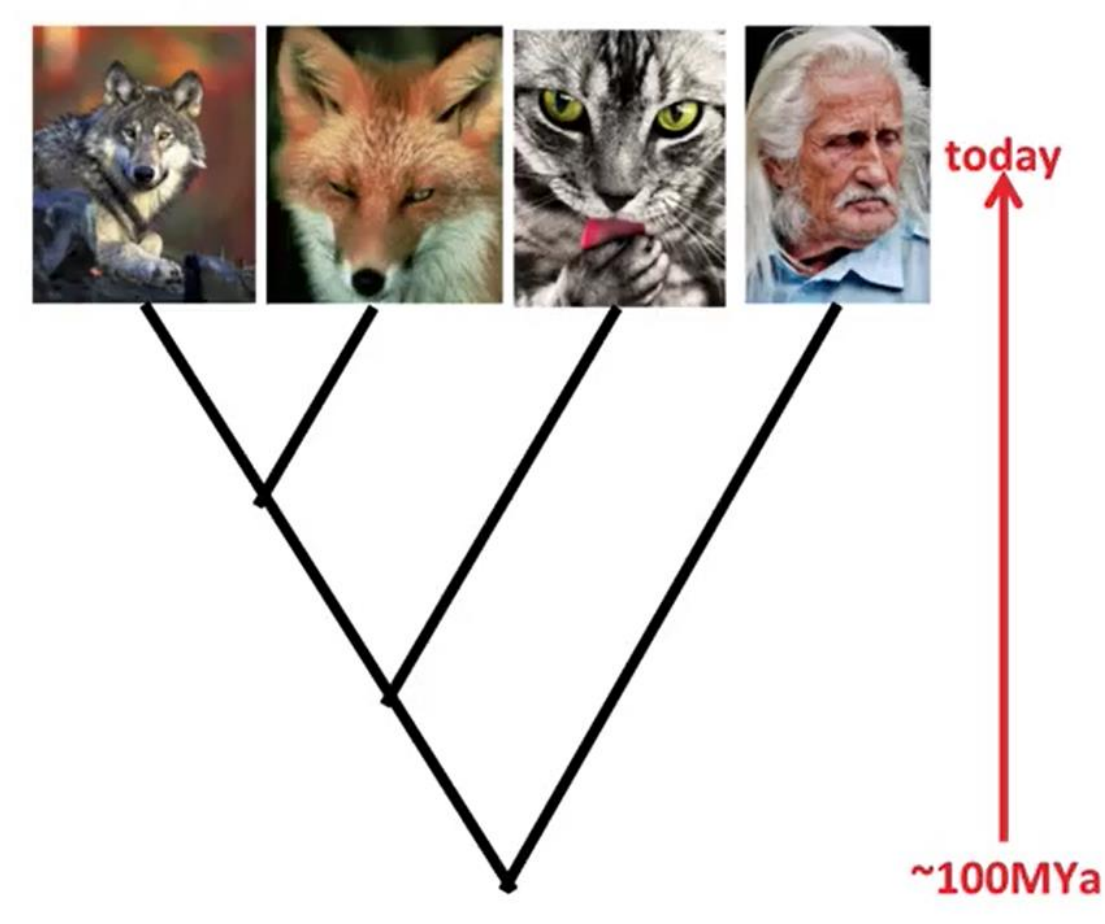
Carl Linneaus
1707-1778



Classification groups imply closer evolutionary relationships



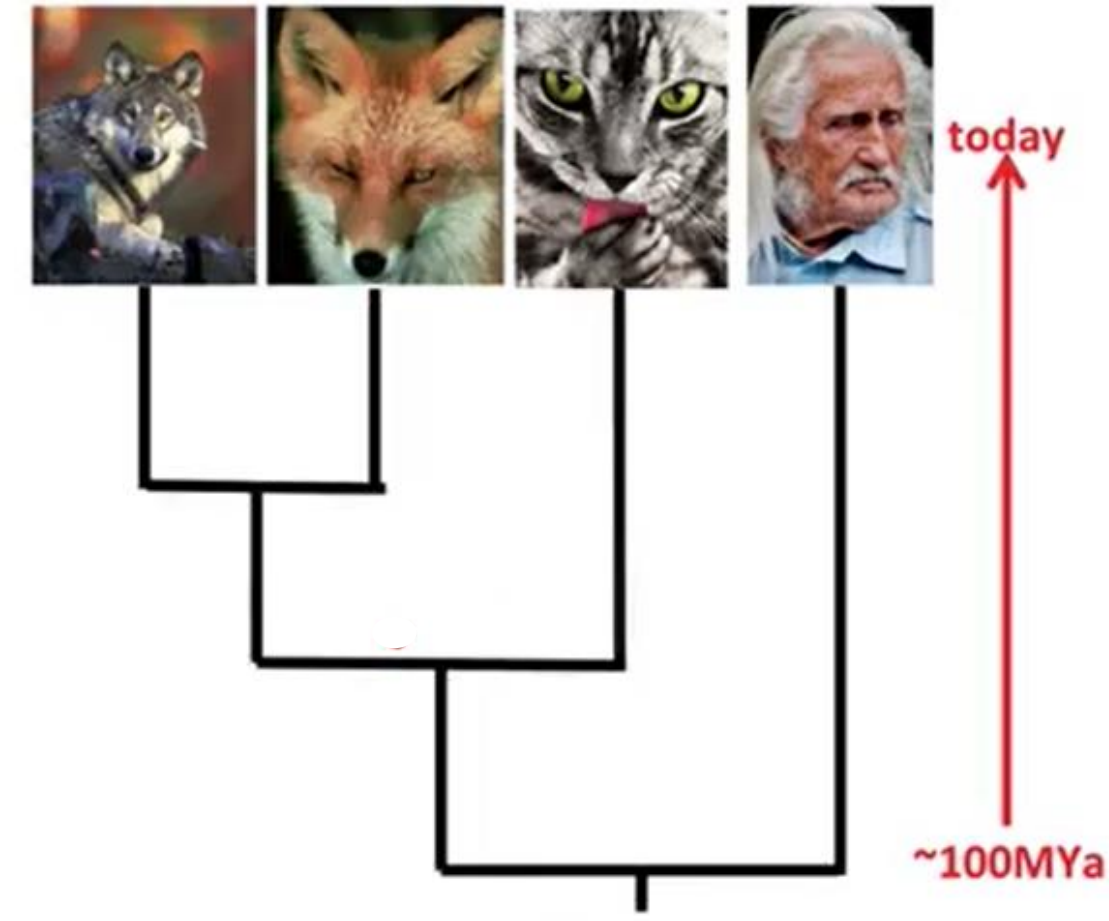
Sample Tree



- Leaf, node, branch (edge), root.

Phylogenetic tree depicts the relationship between Operational Taxonomic Units (OTUs).

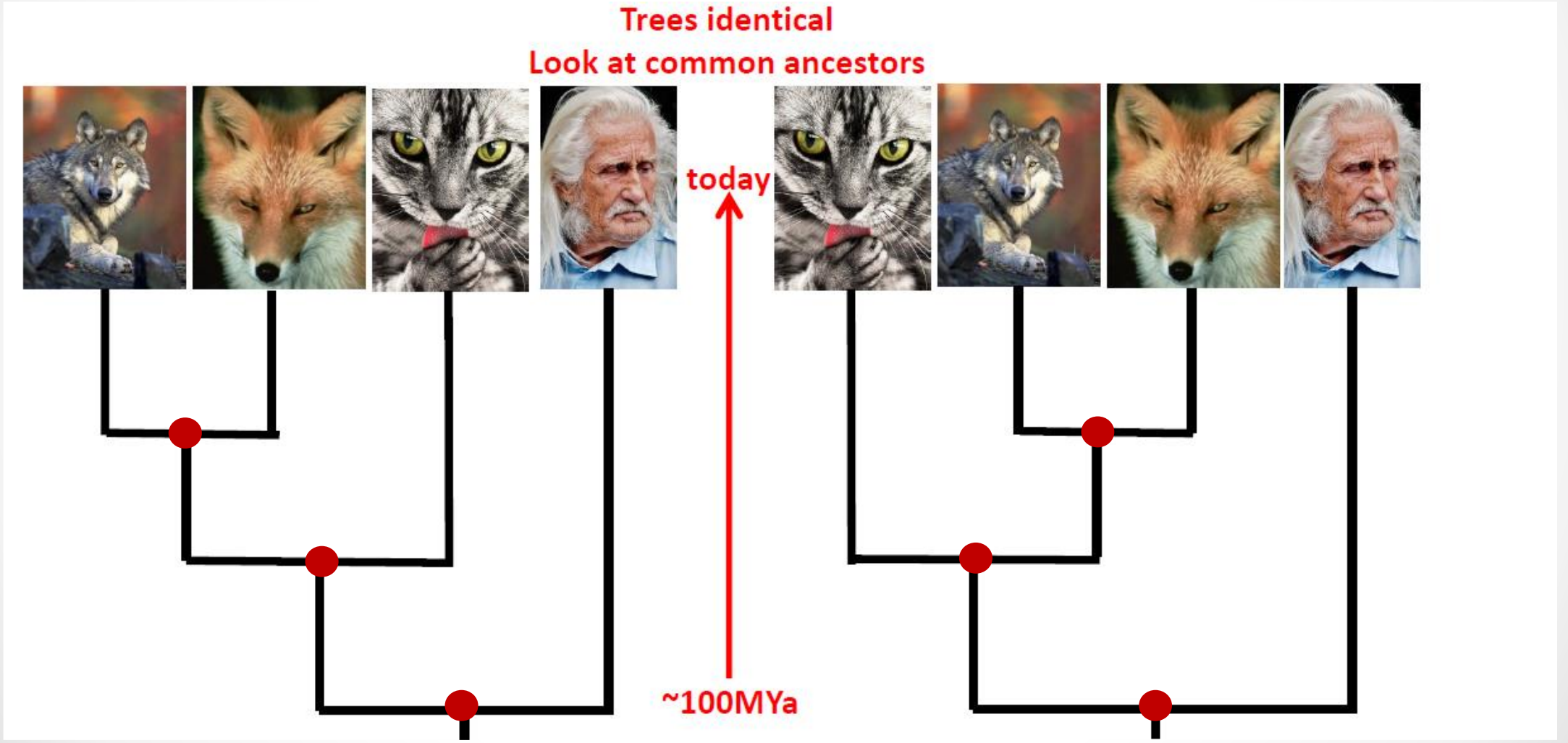
Sample Tree



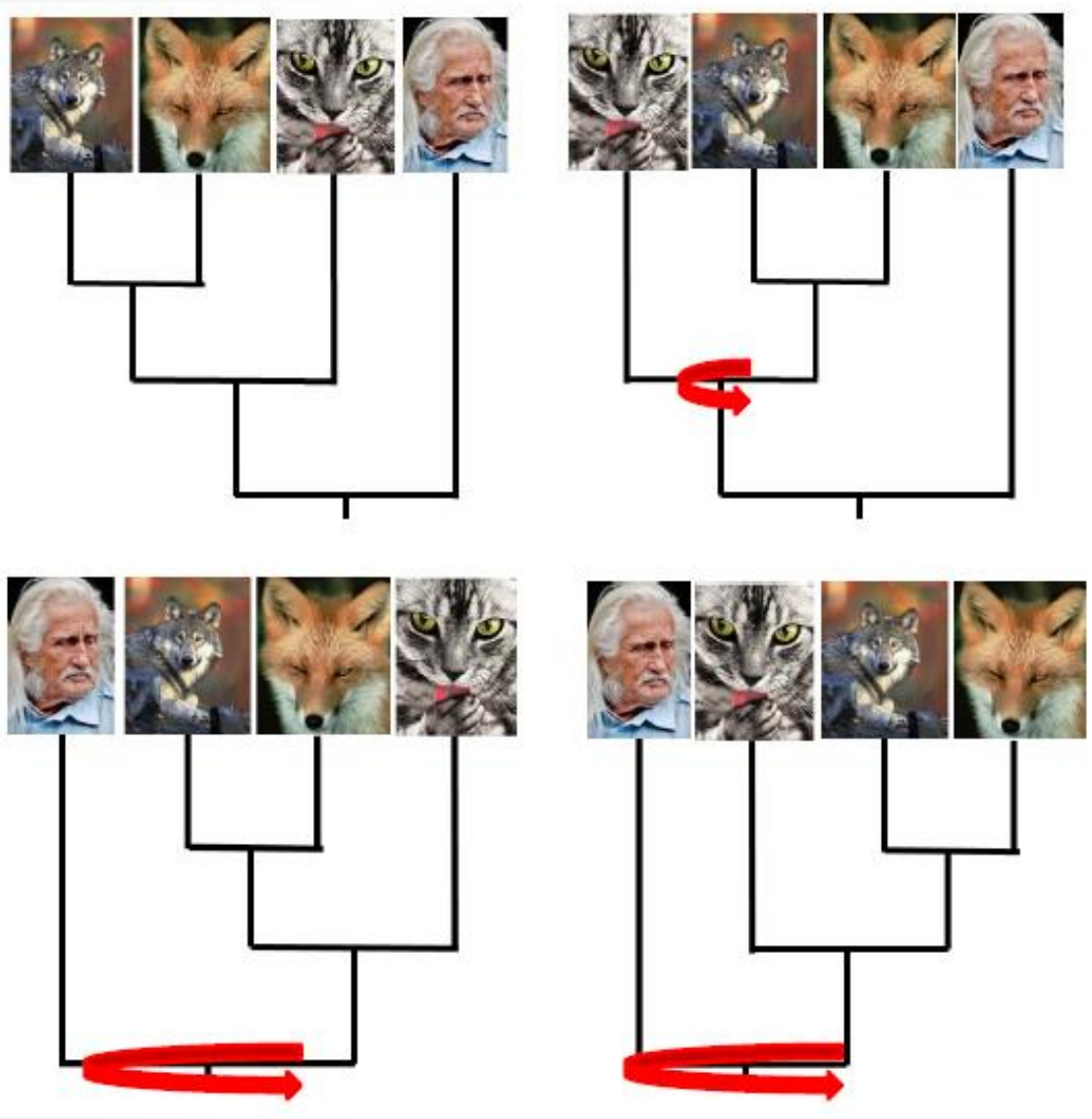
- Leaf, node, branch (edge), root.
- Different ways of representing.
- Assess relationship by looking at who shares "common ancestor".

Phylogenetic tree depicts the relationship between Operational Taxonomic Units (OTUs).

Are these trees different?

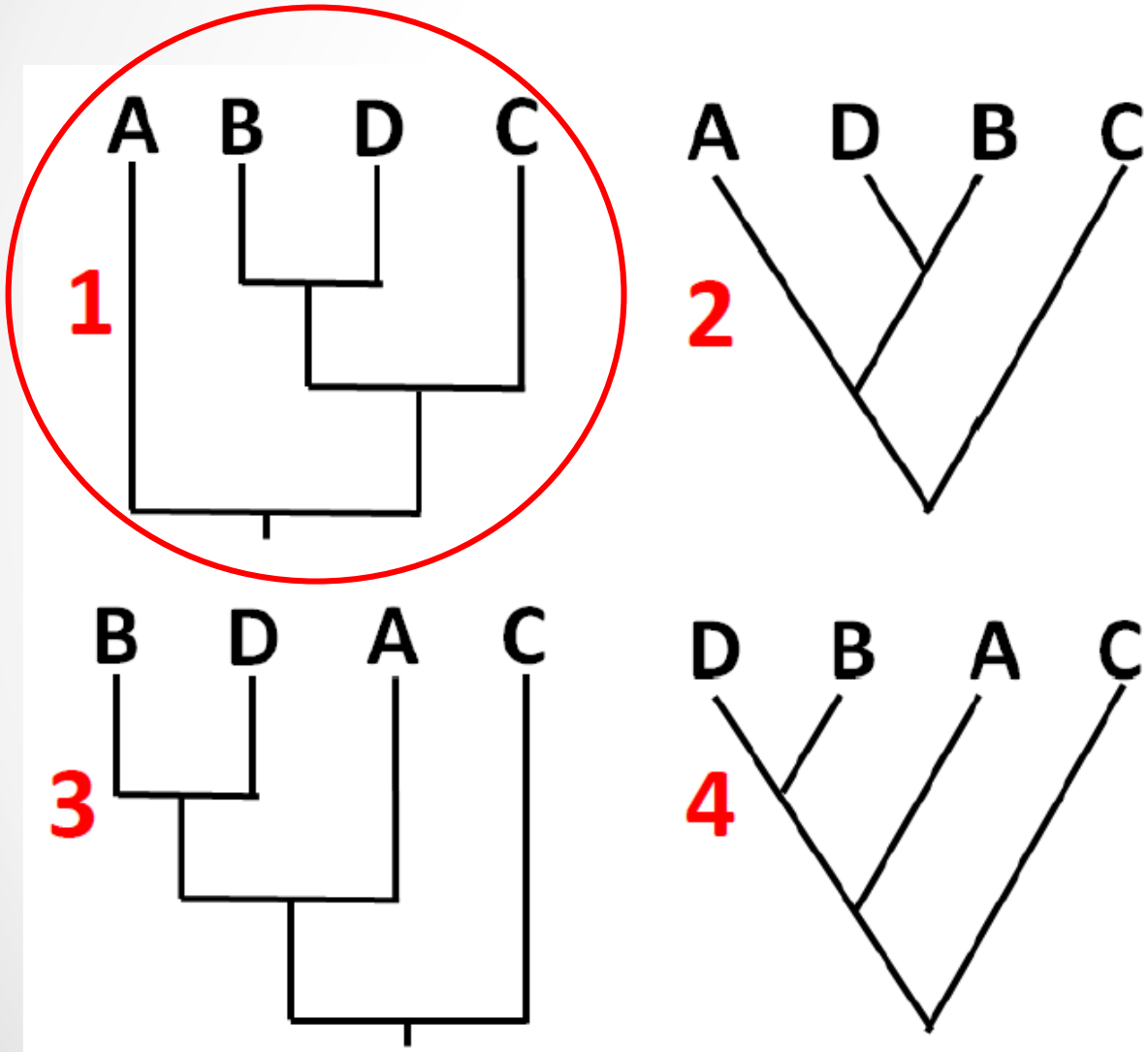


Comparing trees



- Rotating around a node does not change **who is more closely related**.

Comparing trees

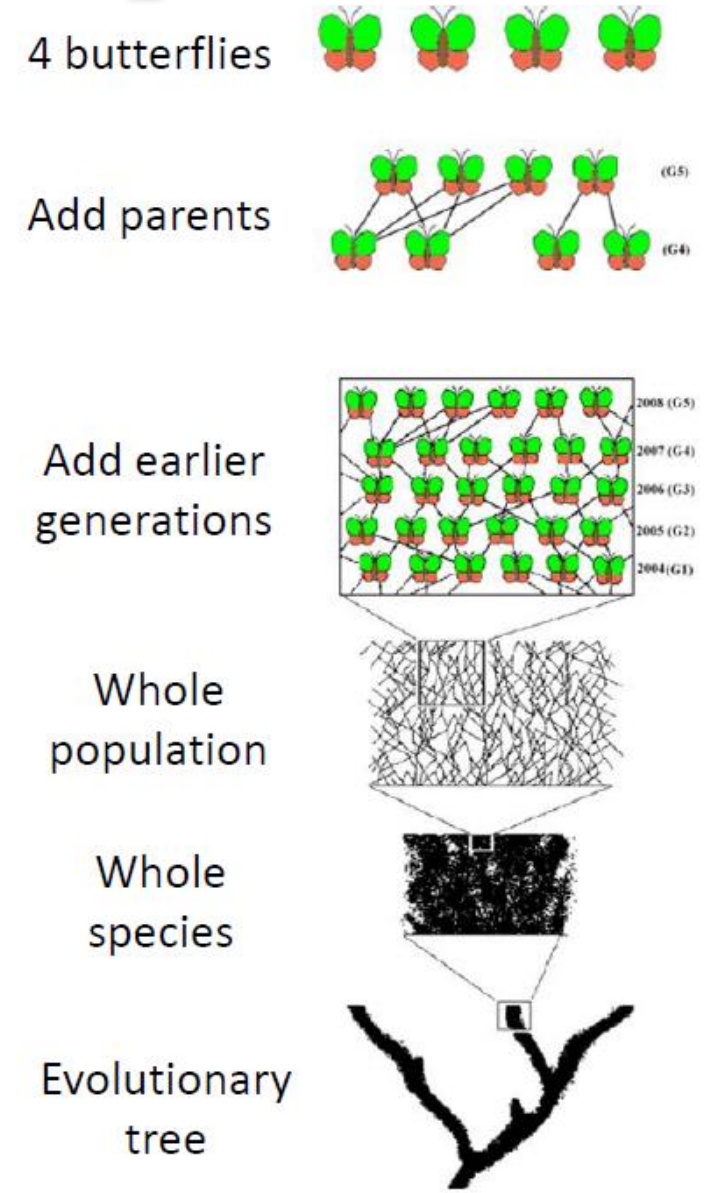


Which of these trees is different?

A has more recent common ancestor to B&D in 2, 3, 4.

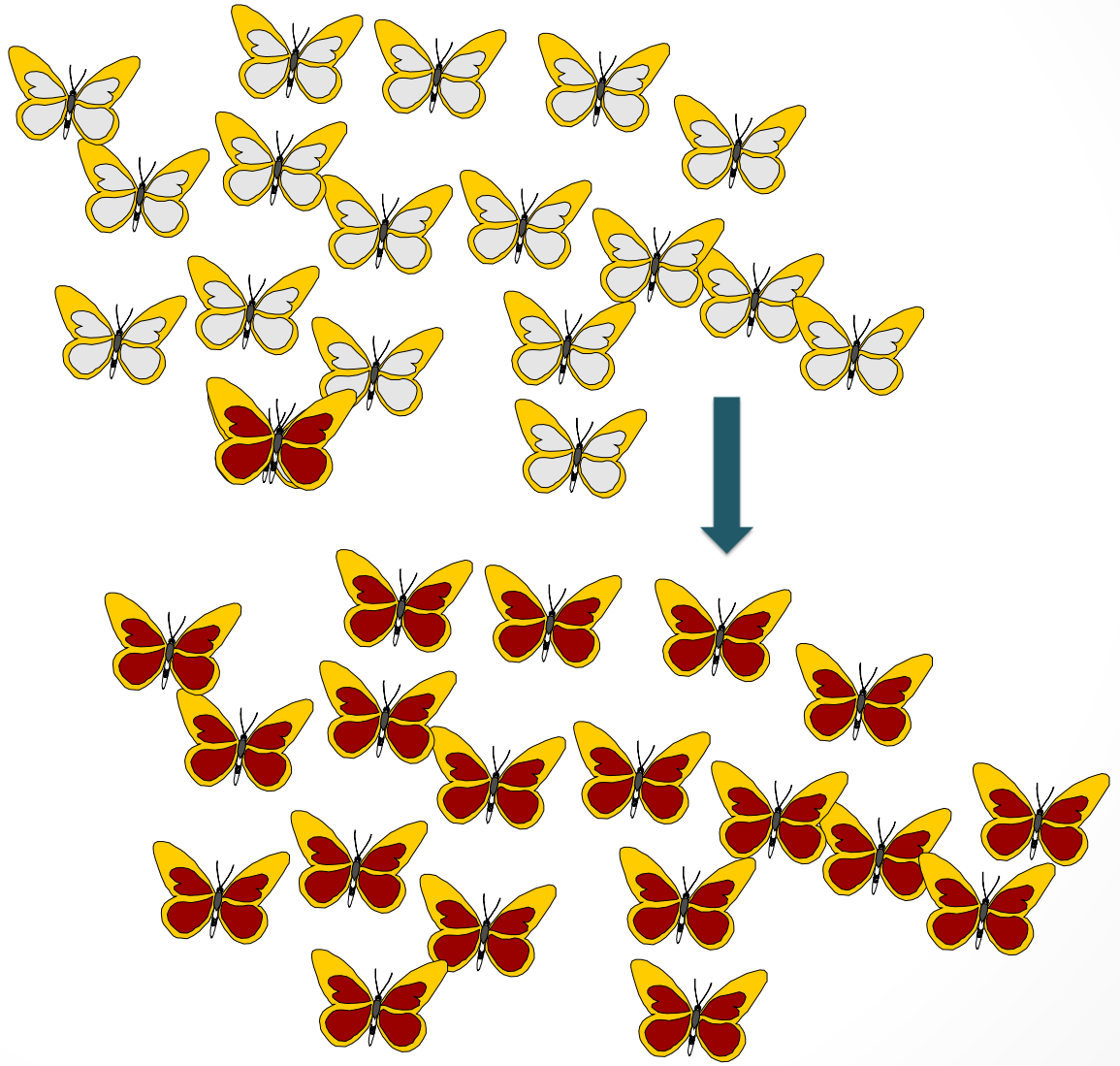
Evolutionary assumptions

- Summation of relationships
- Tree reflects evolutionary history
  - Common ancestry of all species
- Clean splits into two or more taxa (bifurcate)
  - Once species diverge don't come back together

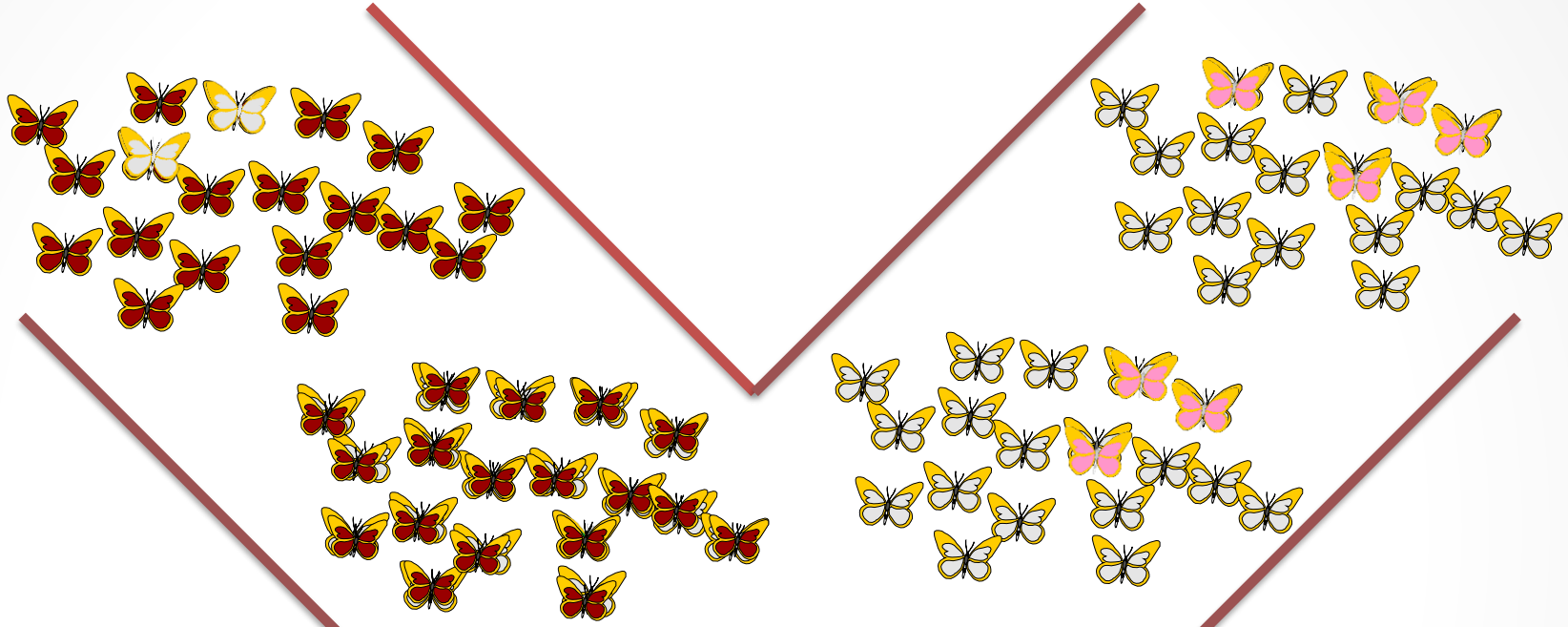


Tree thinking: *descend with modification*

Mutation rate
Time
Fixation rate



Tree thinking: *descend with modification*



SPECIATION

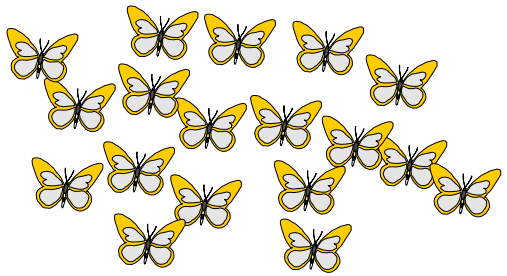
Mutation rate
Time
Fixation rate

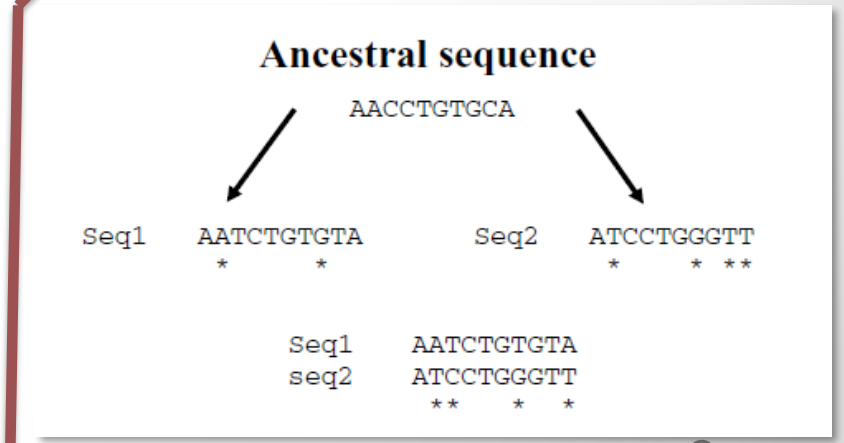
+

Reproductive isolation

⇩

Divergence





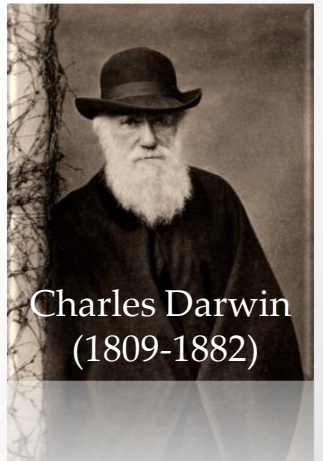
Molecular revolution in the living beings classification

Morphological features *vs* molecular data

Molecular data (DNA and proteins; genomes)

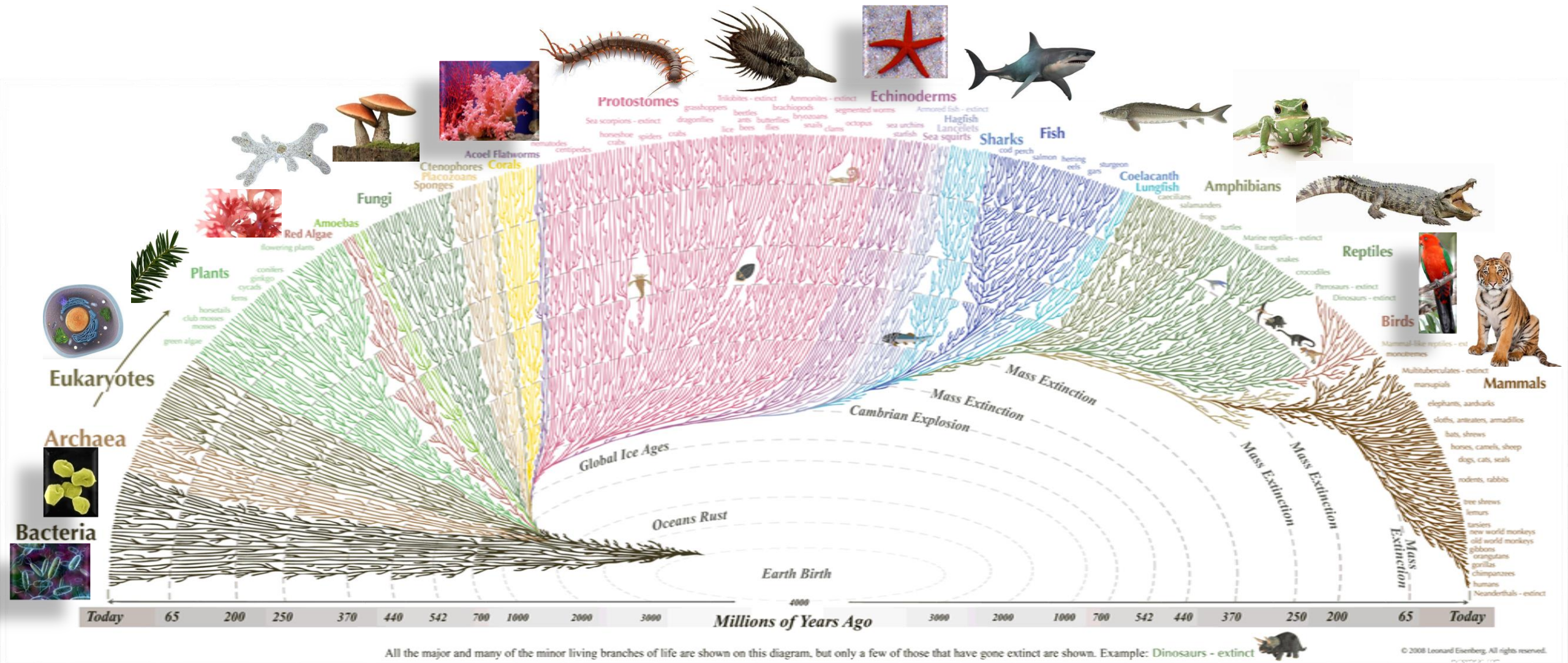
- Universal
- Evolve more uniformly
- More adequate for quantitative analysis
- Much more abundant

Now it is possible to meet the Darwin's dream and reconstruct "a true genealogy tree of each great kingdom of Nature".



Charles Darwin
(1809-1882)

Tree of life



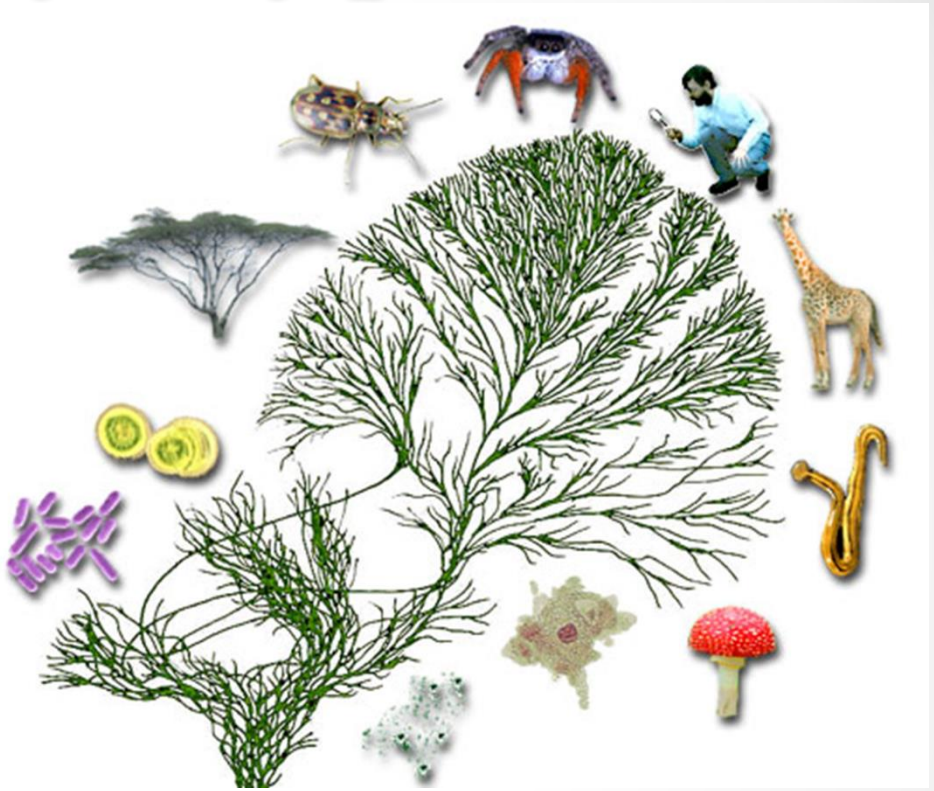
All the major and many of the minor living branches of life are shown on this diagram, but only a few of those that have gone extinct are shown. Example: Dinosaurs - extinct

Outline

- Character-Based Tree Reconstruction
- The Small Parsimony Problem
- The Large Parsimony Problem
- Tree Thinking
- **Evolutionary Tree Reconstruction in the Modern Era**

Testing Evolutionary Hypothesis

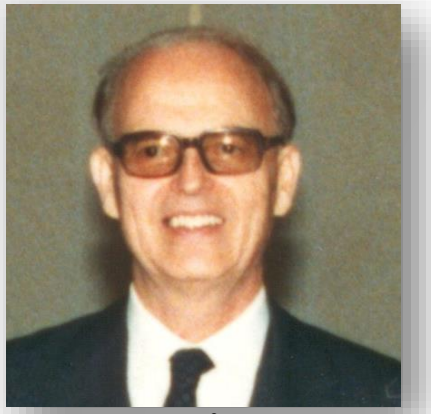
- Broad evolutionary inference requires comparisons among groups
- Those comparisons are most meaningful when studied in the context of evolutionary history
  - Phylogenetics is an estimate of evolutionary history



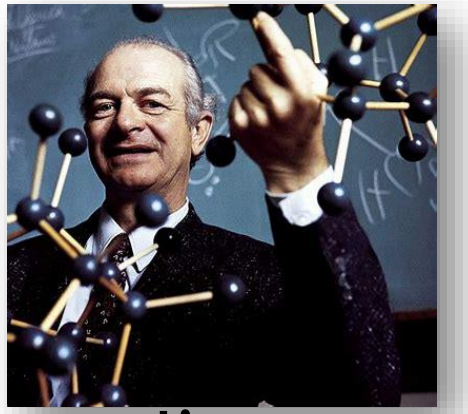
Phylogenies are fundamental to comparative biology; there is no doing it without taking them into account.

J. Felsenstein

1963: Paradigm Shift in Genetic Analysis



Emile Zuckerkandl



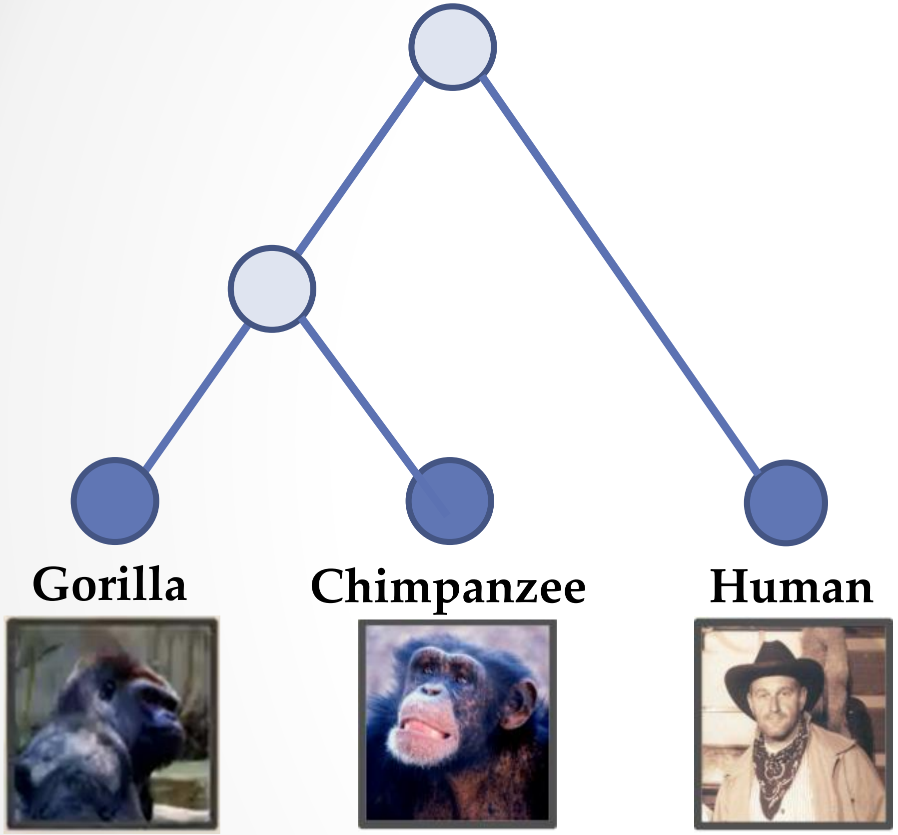
Linus Pauling

Gaylord Simpson

From the point of view of hemoglobin structure, it appears that gorilla is just an abnormal human.

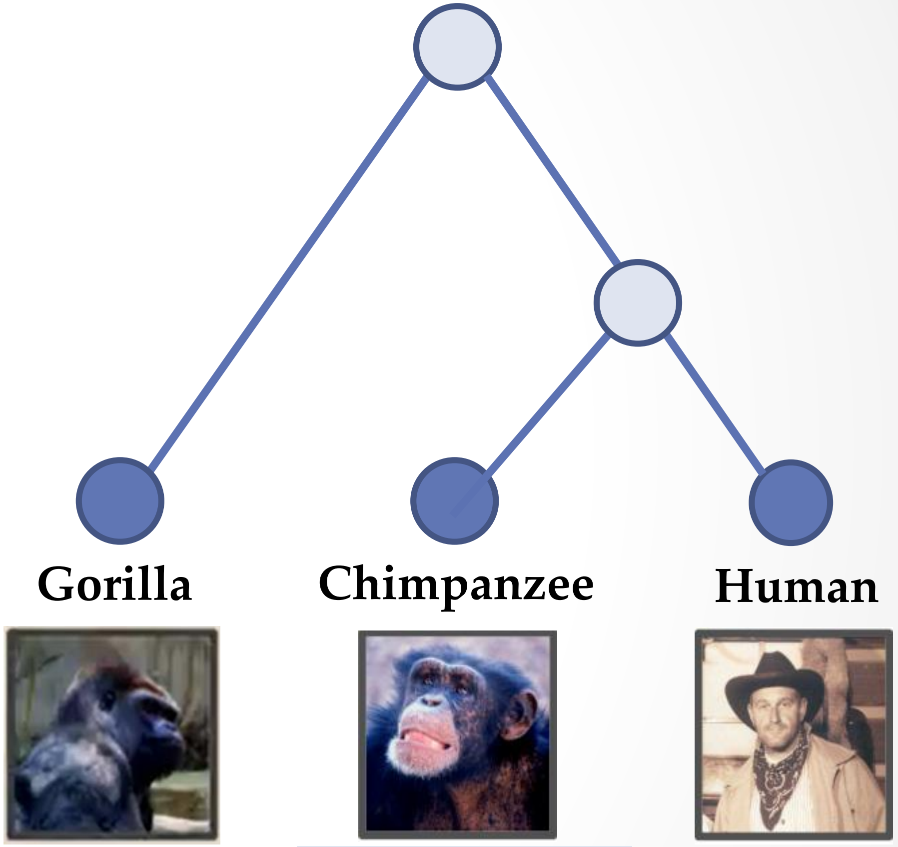
...that is of course nonsense. What the comparison really indicates is that hemoglobin is a bad choice and has nothing to tell us about attributes, or indeed tells a lie.

1996: Human-Chimp-Gorilla Ancestry



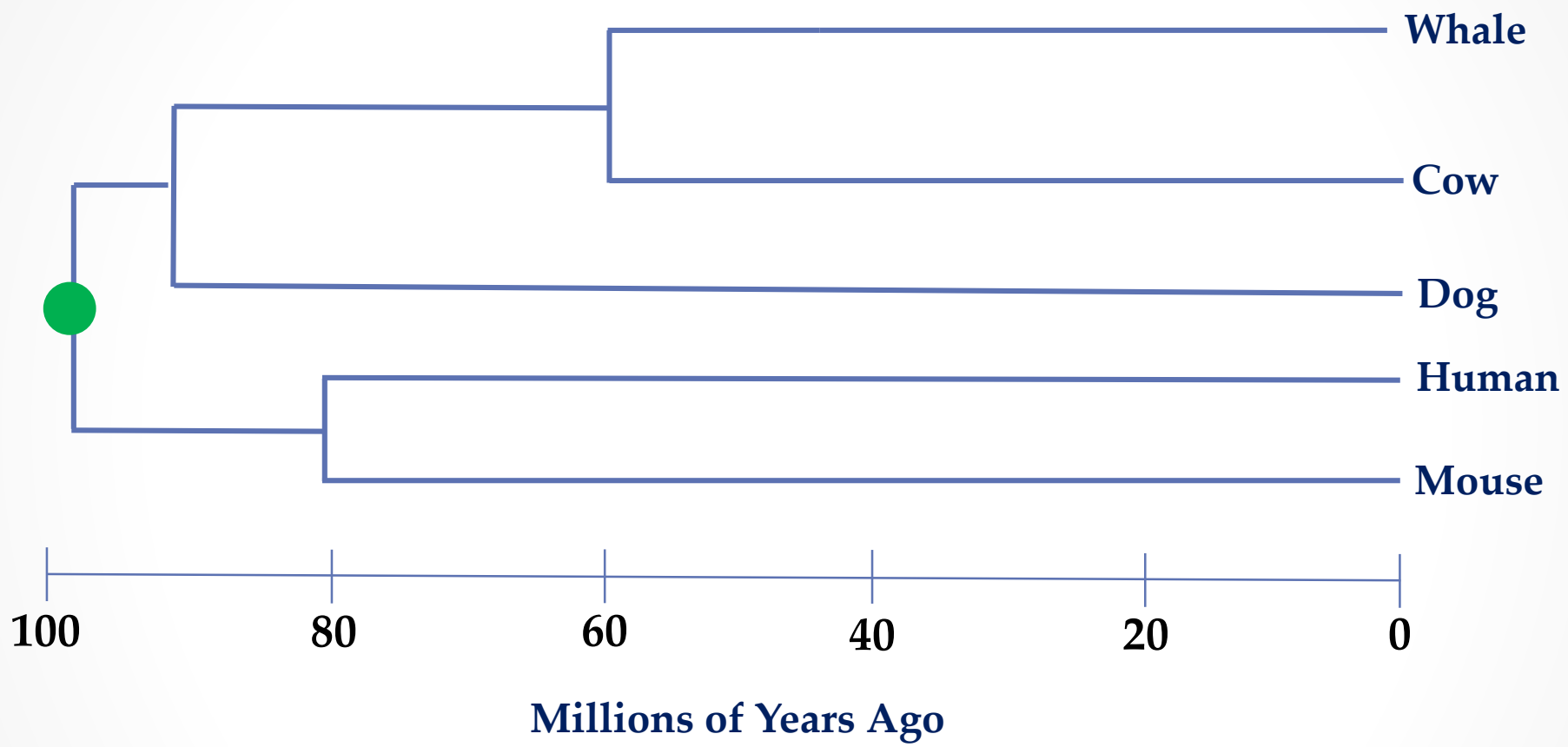
Dopamine D4 receptor

vs.



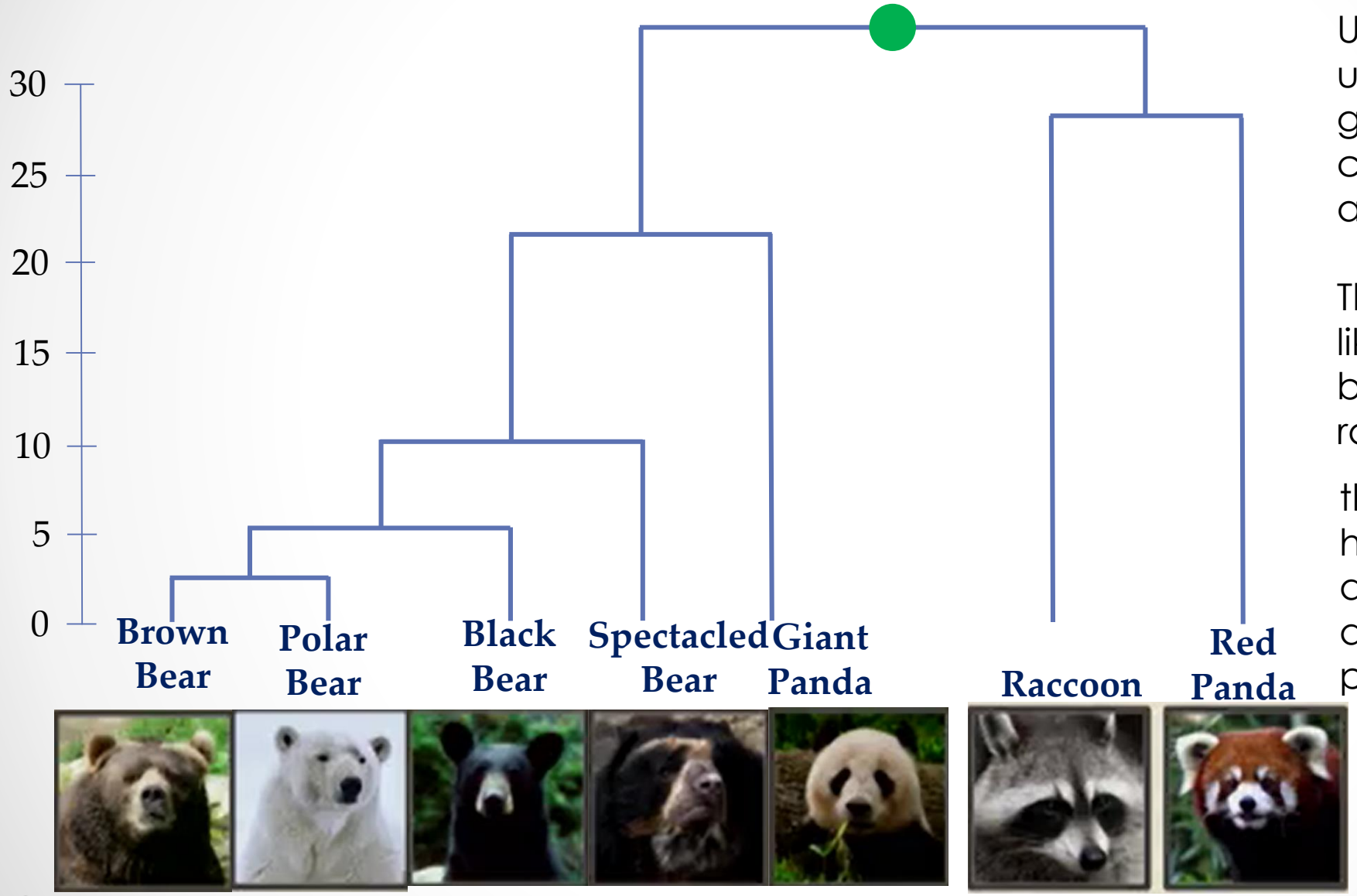
Beta globin

2000s: Human and Mice are Relatives



Many biologists believed that humans are more closely related to dogs than to mice. But recent research has shown otherwise, as indicated by this phylogeny.

1985: Panda's Pedigree Decoded

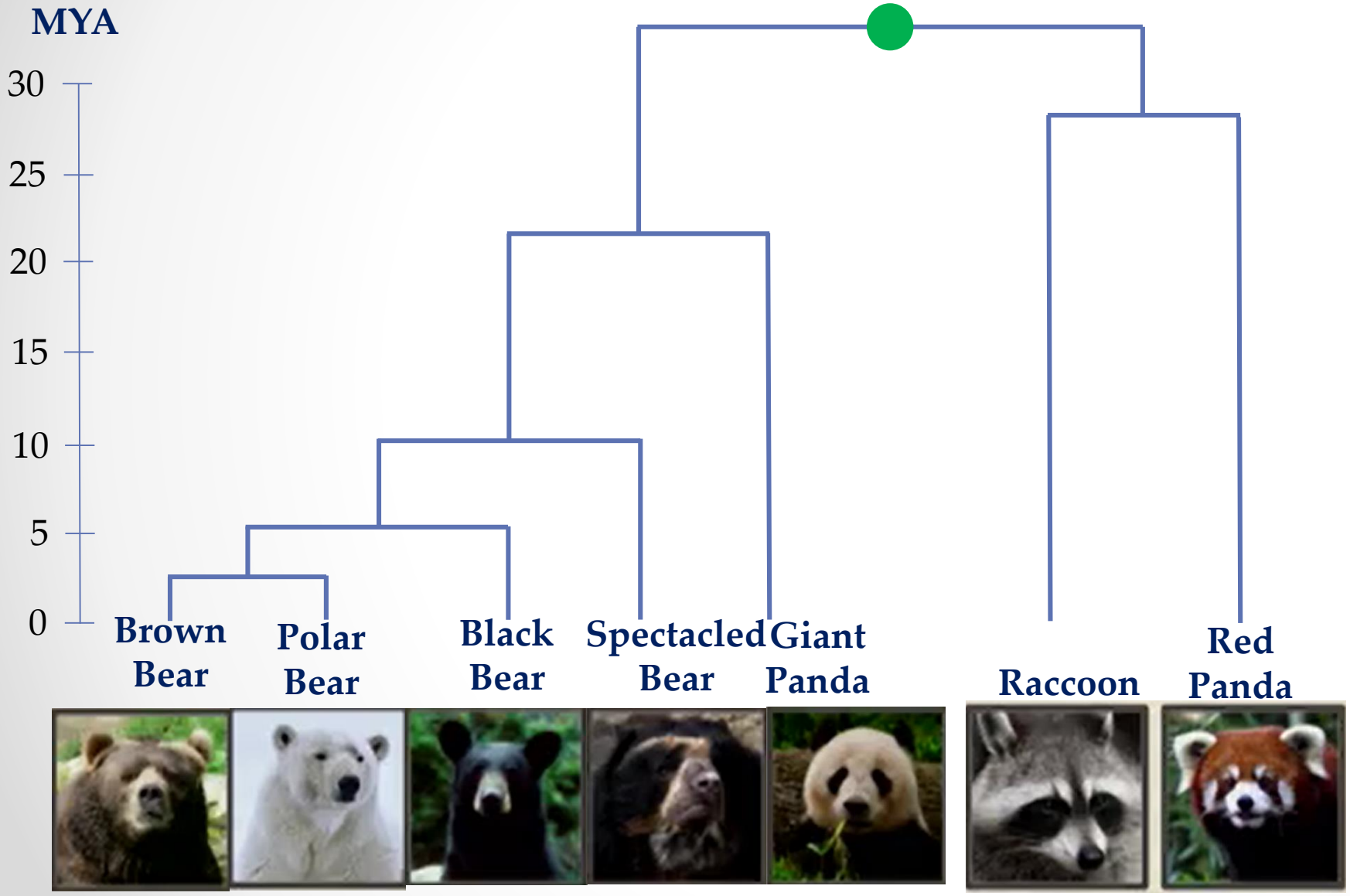


Until the 1980s, it was unclear whether the giant panda should be classified as a bear or as a raccoon.

The giant panda looks like a bear in some ways, but it's also similar to raccoons in other ways:

the panda doesn't hibernate in the winter, and its male genitalia are tiny and backward-pointing.

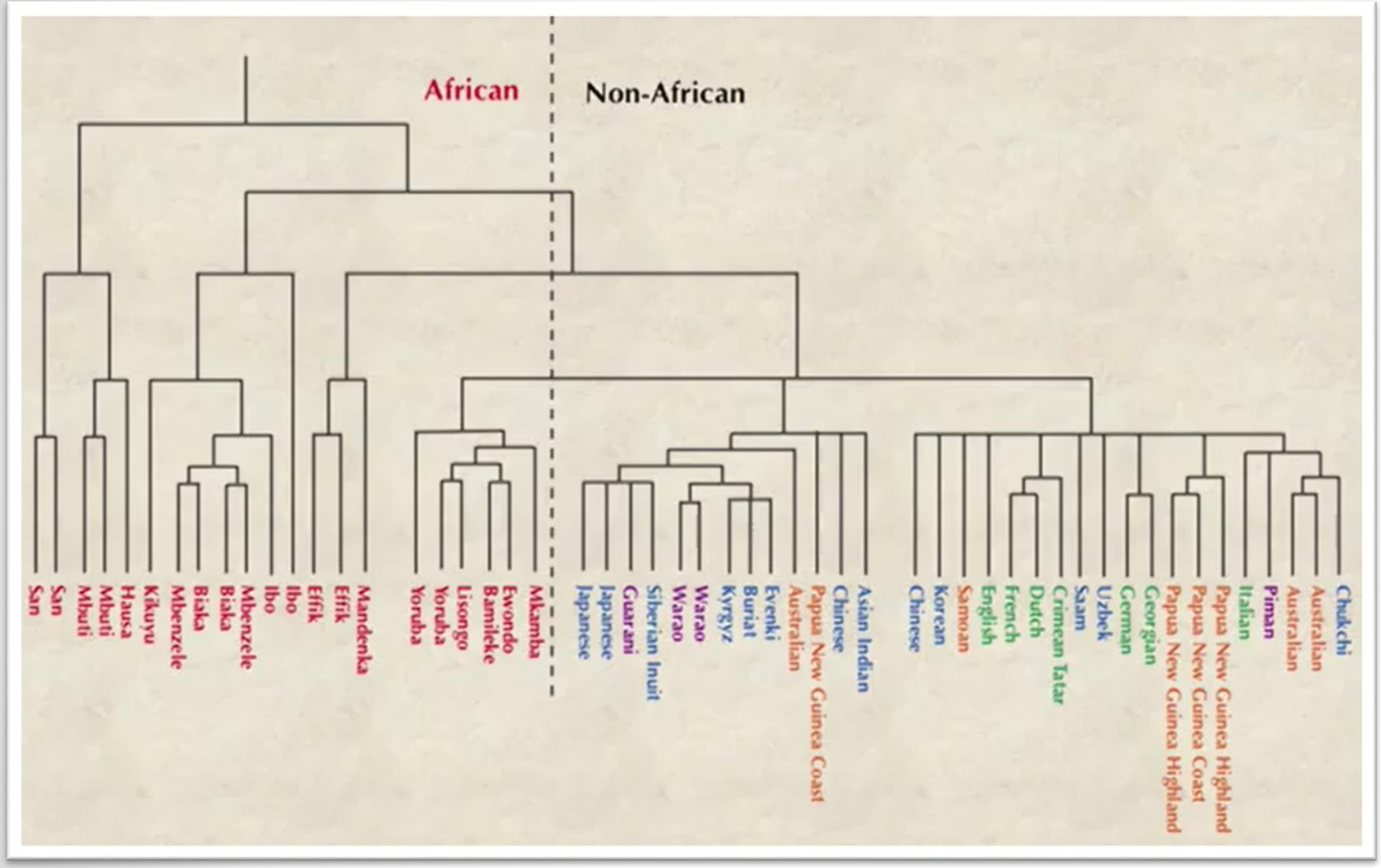
1985: Panda's Pedigree Decoded



Stephen J. O'Brien

Whereas using anatomical or behavioral characters led to endless debate in this subject, Steve O'Brien used genetic data in 1985 to demonstrate that the giant panda should, in fact, be considered a bear - a conclusion that has lasted to this day.

1987: Tracing Human Origins



Evolutionary tree for human populations.

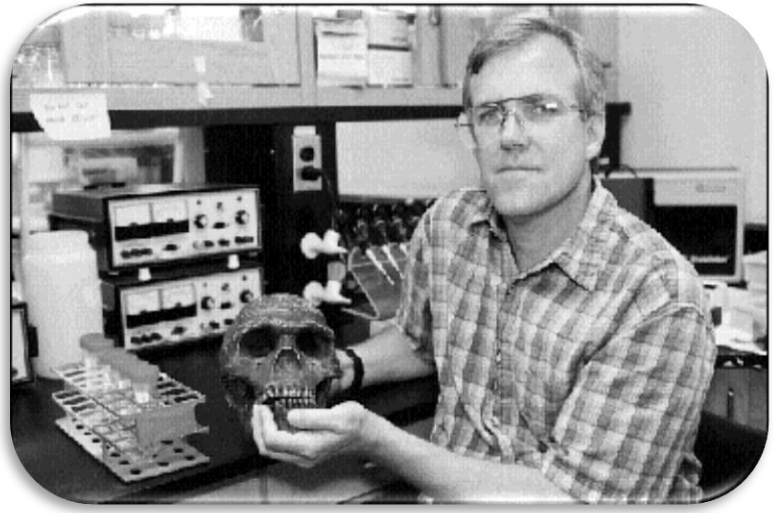
Clear division into Africans, shown in red, and non-Africans, shown in other colors, where each color represents a different continent is observed.

The most recent common ancestor of all non-Africans is at a much more recent point than the most recent common ancestor of all Africans, which is at the root of the tree.

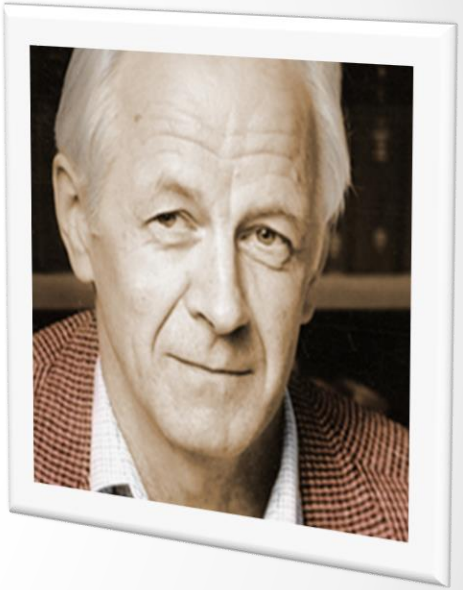
1987: Tracing Human Origins

Because the origin of modern humans predates the origin of all non Africans, all non-Africans must have originated somewhere in Africa.

Rebecca Cann

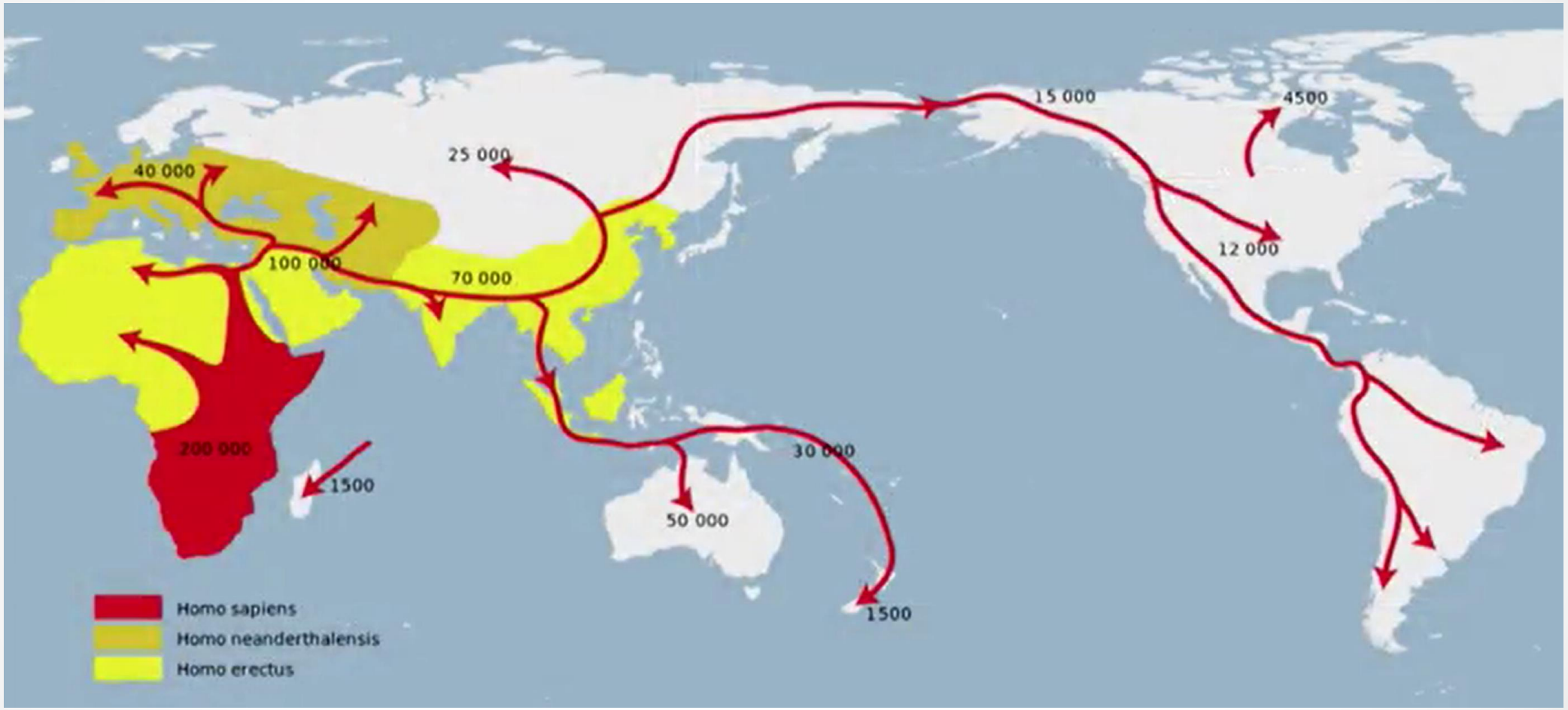


Mark Stoneking



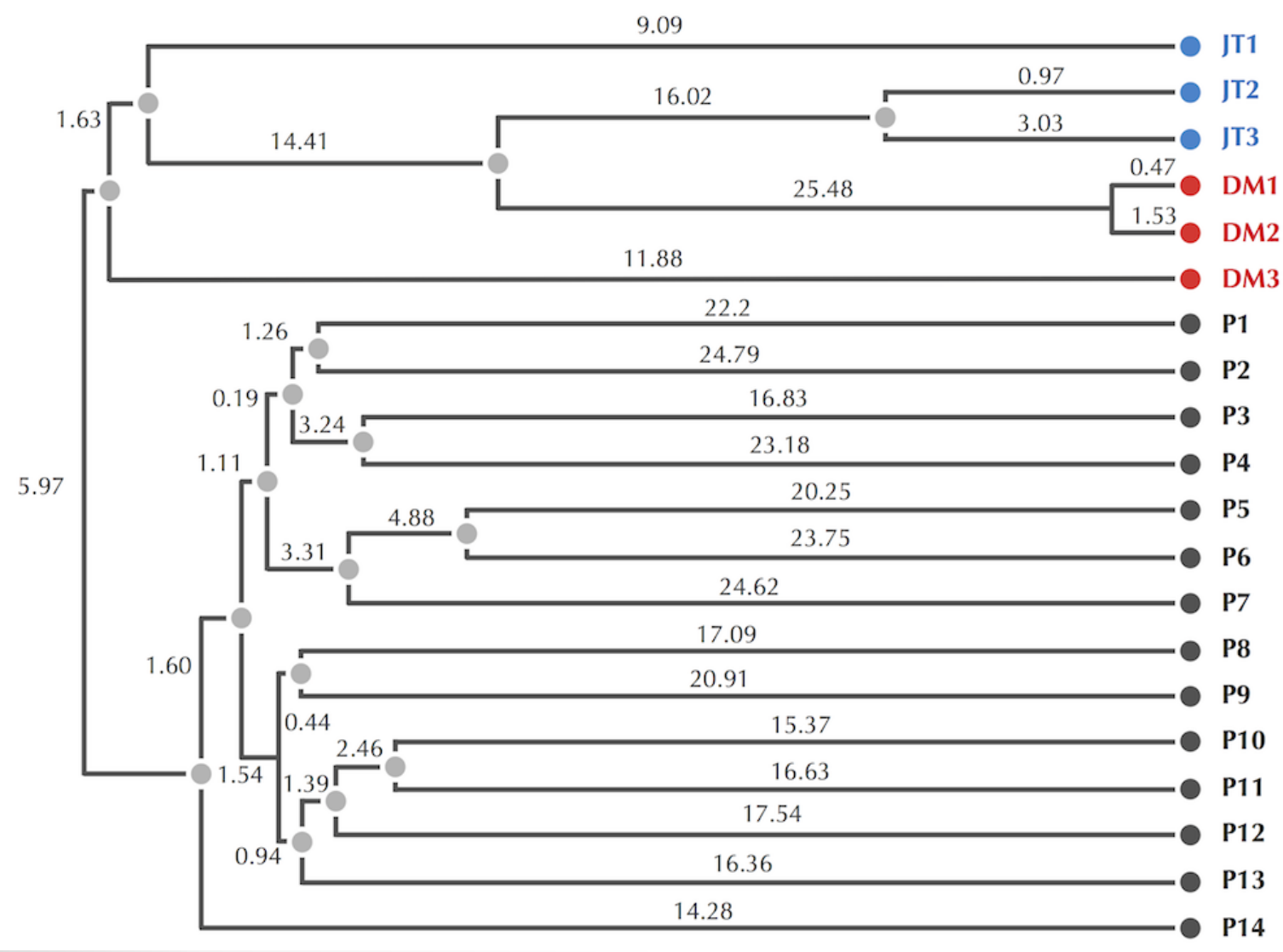
Allan Wilson

1987: Tracing Human Origins



As more genotyping data has been obtained, researchers have been able to infer human migrations around the world from evolutionary trees.

1998: Evolutionary Trees Fight Crime



A doctor in Louisiana went on trial for the attempted murder of his estranged mistress. She claimed that he had injected her with an HIV-tainted syringe. The evidence supported conviction.

An evolutionary tree of HIV viruses taken from various patients in Lafayette. Samples from Janice Trahan (blue leaves JT1, JT2, and JT3) and Donald McClelland (red leaves DM1, DM2, and DM3) are clustered together and are rather different than sequences from other patients from Lafayette (labeled P1 to P14).

Applications of Evolutionary Trees

- Classify nature and understand evolution
  - Human origins
- Medicine / Law
  - Personalized medicine
  - Cancer colonies
  - Identify flu strains for vaccine
  - Trace HIV infection to source

    State of Washington vs. A. Whitfi

    State of Texas vs. P.Padieu