

Molecular Evolution

Outline

- Evolutionary Tree Reconstruction
 - “Out of Africa” hypothesis
 - Did we evolve from Neanderthals?
 - Distance Based Phylogeny
 - Neighbor Joining Algorithm
 - Additive Phylogeny
 - Least Squares Distance Phylogeny
 - UPGMA
 - Character Based Phylogeny
 - Small Parsimony Problem
 - Fitch and Sankoff Algorithms
 - Large Parsimony Problem
 - Evolution of Wings
 - HIV Evolution
 - Evolution of Human Repeats
-

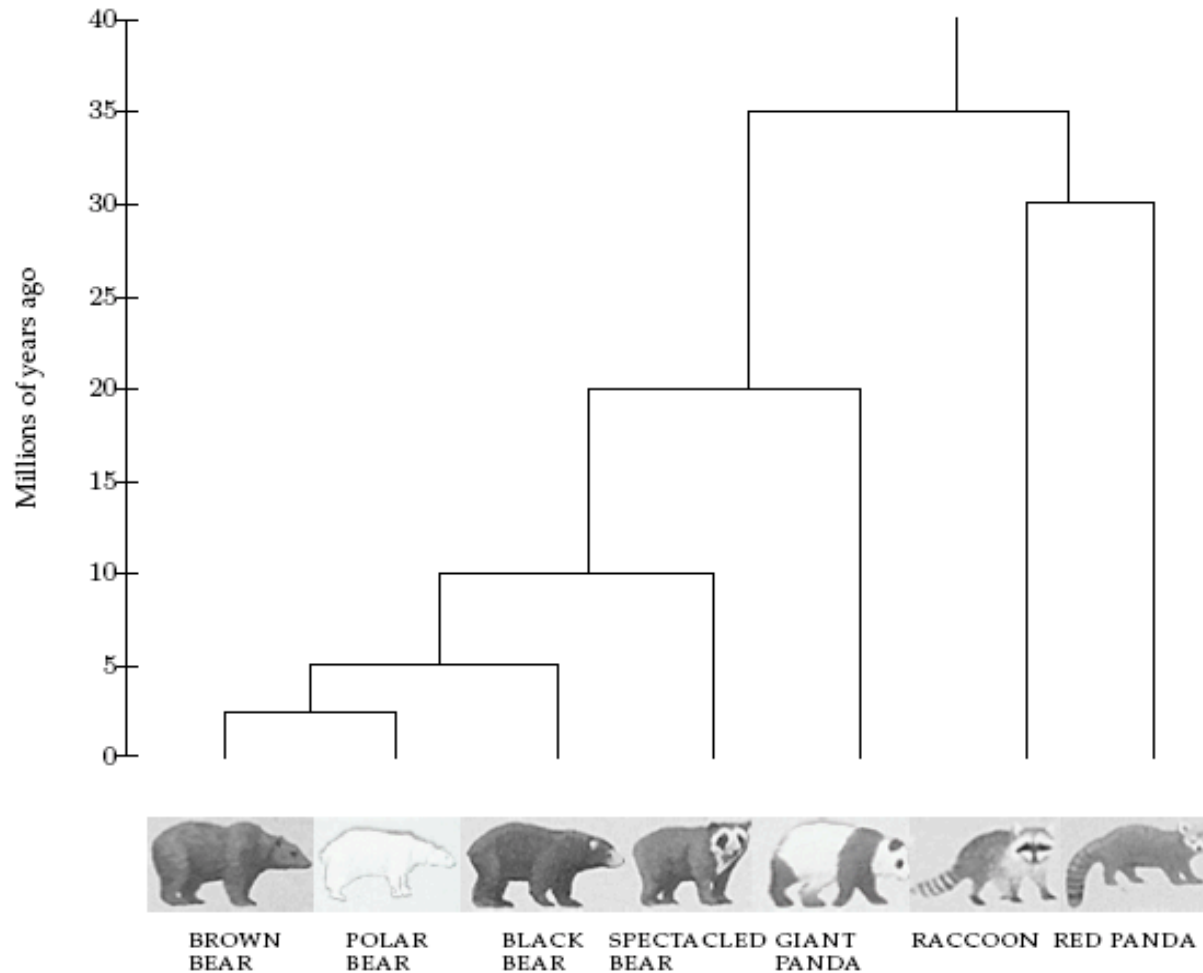
Early Evolutionary Studies

- Anatomical features were the dominant criteria used to derive evolutionary relationships between species since Darwin till early 1960s
- The evolutionary relationships derived from these relatively subjective observations were often inconclusive. Some of them were later proved incorrect

Evolution and DNA Analysis: the Giant Panda Riddle

- For roughly 100 years scientists were unable to figure out which family the giant panda belongs to
- Giant pandas look like bears but have features that are unusual for bears and typical for raccoons, e.g., they do not hibernate
- In 1985, Steven O'Brien and colleagues solved the giant panda classification problem using DNA sequences and algorithms

Evolutionary Tree of Bears and Raccoons



Evolutionary Trees: DNA-based Approach

- 40 years ago: Emile Zuckerkandl and Linus Pauling brought reconstructing evolutionary relationships with DNA into the spotlight
 - In the first few years after Zuckerkandl and Pauling proposed using DNA for evolutionary studies, the possibility of reconstructing evolutionary trees by DNA analysis was hotly debated
 - Now it is a dominant approach to study evolution.
-

Emile Zuckerkandl on human-gorilla evolutionary relationships:



From the point of hemoglobin structure, it appears that gorilla is just an abnormal human, or man an abnormal gorilla, and the two species form actually one continuous population.

***Emile Zuckerkandl,*
Classification and Human Evolution, 1963**

Gaylord Simpson vs. Emile Zuckerkandl:



From the point of hemoglobin structure, it appears that gorilla is just an abnormal human, or man an abnormal gorilla, and the two species form actually one continuous population.

Emile Zuckerkandl,
Classification and Human Evolution, 1963

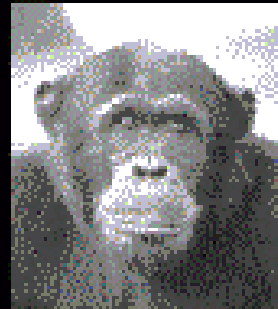


From any point of view other than that properly specified, that is of course nonsense. What the comparison really indicate is that hemoglobin is a bad choice and has nothing to tell us about attributes, or indeed tells us a lie.

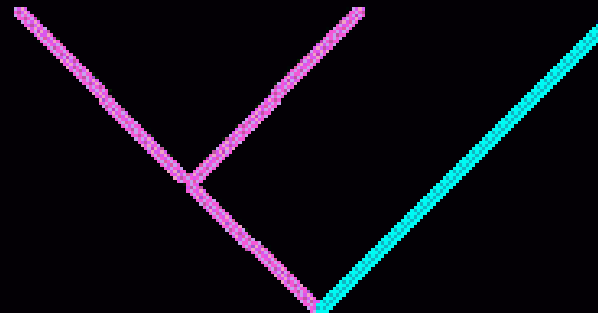
Gaylord Simpson,
Science, 1964

Who are closer?

Does genetics show that humans and chimps
are each other's closest relative?



Gorillas



Human-Chimpanzee Split?



Beta-globins (Mol. Phyl. Evol., 1:97, 1992)

Zinc Finger Y (Science, 268:1183, 1995)

α -1,3-galactosyltransferase

(PNAS, 88:7401, 1991)

mtDNA, 4.9 kb (J. Mol. Evol., 35:32, 1992)

c-myc (J. Mol. Evol., 41:262, 1995)

Chimpanzee-Gorilla Split?



Tyrosine hydroxylase intron 1

(J. Mol. Evol., 41:10, 1995)

Complement component C4 intron 9

(Immunogenet., 42:41, 1995)

Dopamine D4 receptor (PNAS, 92:427, 1995)

Protamine P1 (J. Mol. Evol., 37:426, 1993)

Involucrin (PNAS, 86:8447-8451, 1989)

Terminal heterochromatin (AJPA, 90:237, 1993)

Three-way Split?



mtDNA restriction sites (PNAS 78:2434, 1981)

mtDNA sequence (J. Mol. Evol., 18:225, 1982;

Mol. Biol. Evol., 3:1, 1986)

opsins (PNAS, 91: 7262, 1994)

cytochrome P450c21

(Am. J. Hum. Genet., 50:766, 1992)

X-Y pseudoautosomal (Cell, 63:977, 1990)

HOX2B (Mol. Biol. Evol., 9:575, 1992)

Immunoglobulins (J. Mol. Evol., 27:77, 1988

Mol. Biol. Evol., 8:743, 1991;

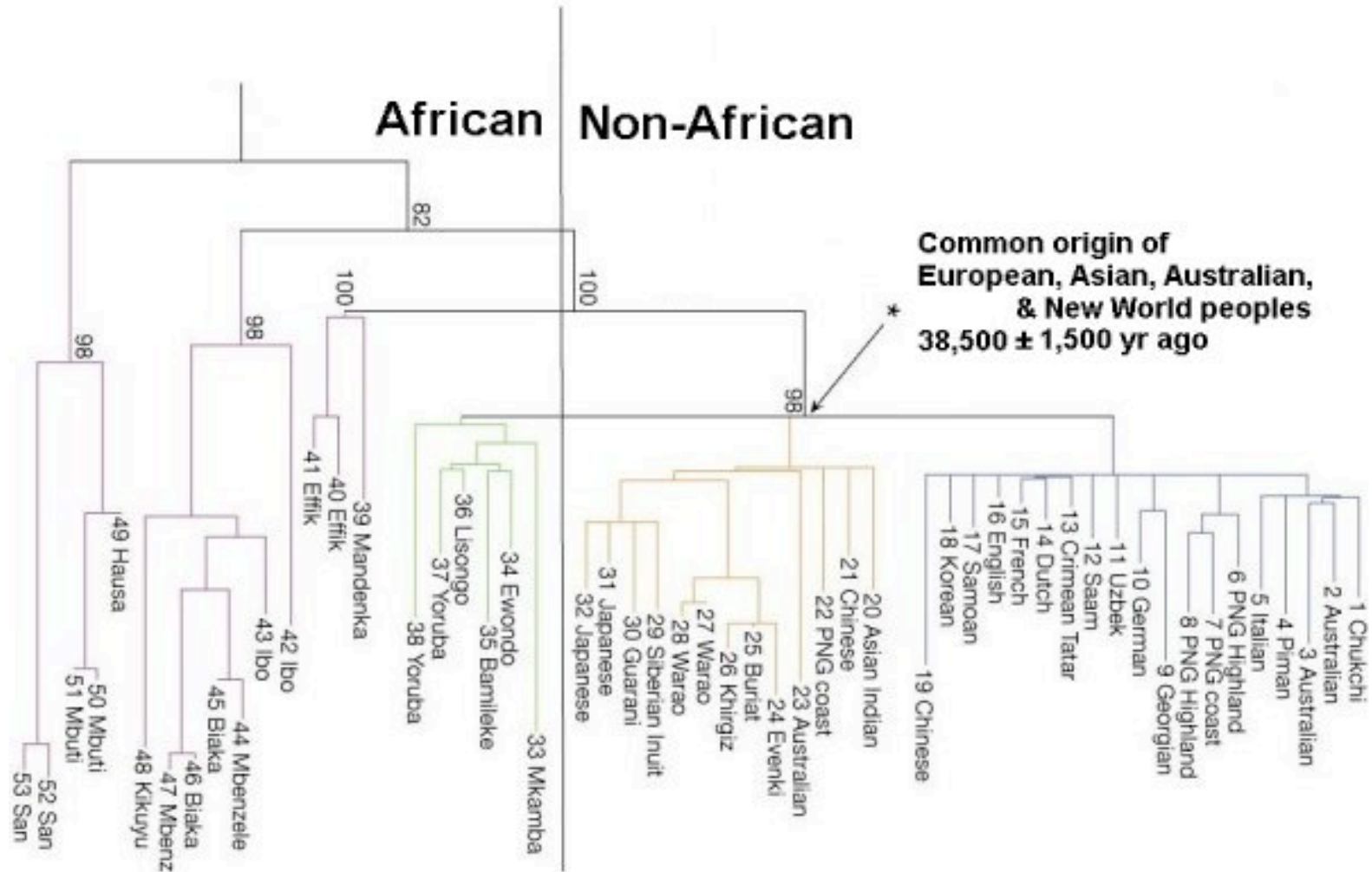
J. Mol. Biol., 205:85, 1989)

rDNA (Mol. Biol. Evol., 7:203, 1990)

Out of Africa Hypothesis

- Around the time the giant panda riddle was solved, a DNA-based reconstruction of the human evolutionary tree led to the **Out of Africa Hypothesis** that claims our most ancient ancestor lived in Africa roughly 200,000 years ago
-

Human Evolutionary Tree (cont'd)



The Origin of Humans:

"Out of Africa" vs Multiregional Hypothesis

Out of Africa:

- Humans evolved in Africa ~150,000 years ago
- Humans migrated out of Africa, replacing other shumanoids around the globe
- There is no direct descendance from Neanderthals

Multiregional:

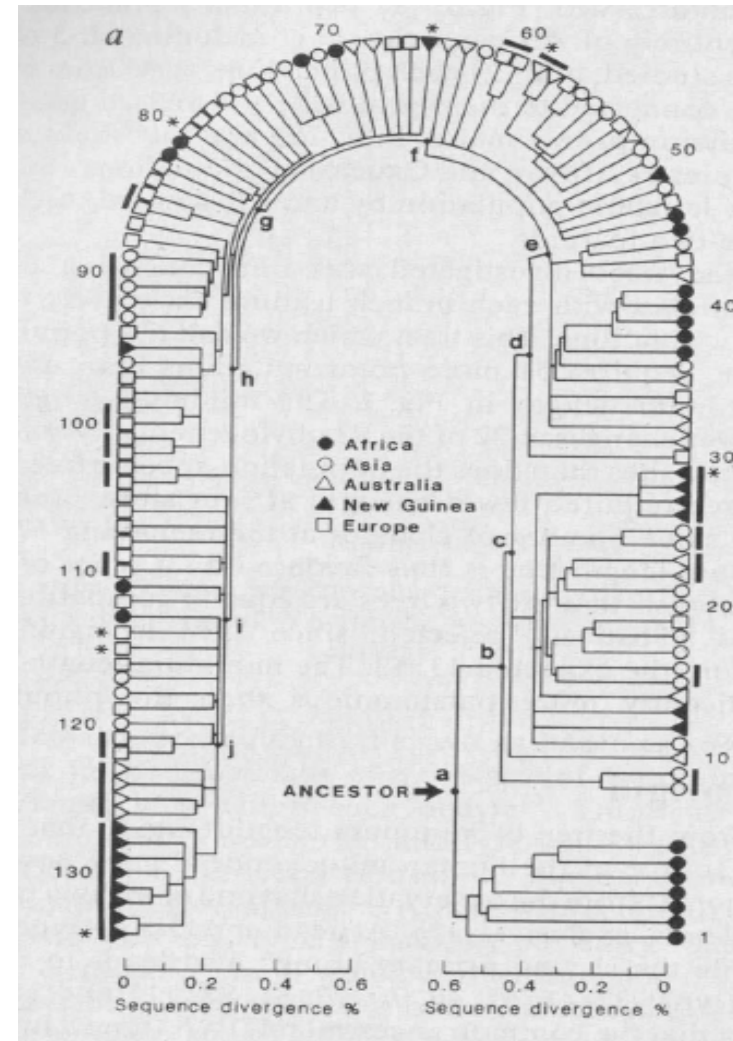
- Humans evolved in the last two million years as a single species. Independent appearance of modern traits in different areas
- Humans migrated out of Africa mixing with other humanoids on the way
- There is a genetic continuity from Neanderthals to humans

mtDNA analysis supports “Out of Africa” Hypothesis

- African origin of humans inferred from:
 - African population was the most diverse
(sub-populations had more time to diverge)
 - The evolutionary tree separated one group of Africans from a group containing all five populations.
 - Tree was rooted on branch between groups of greatest difference.

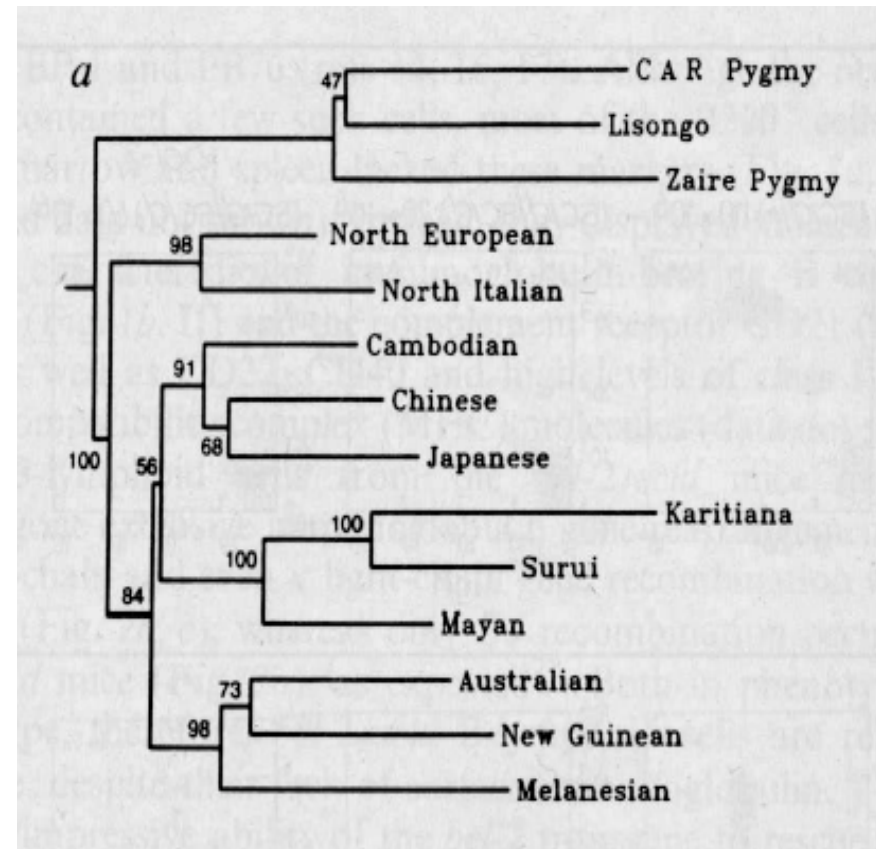
Evolutionary Tree of Humans (mtDNA)

The evolutionary tree separates one group of Africans from a group containing all five populations.



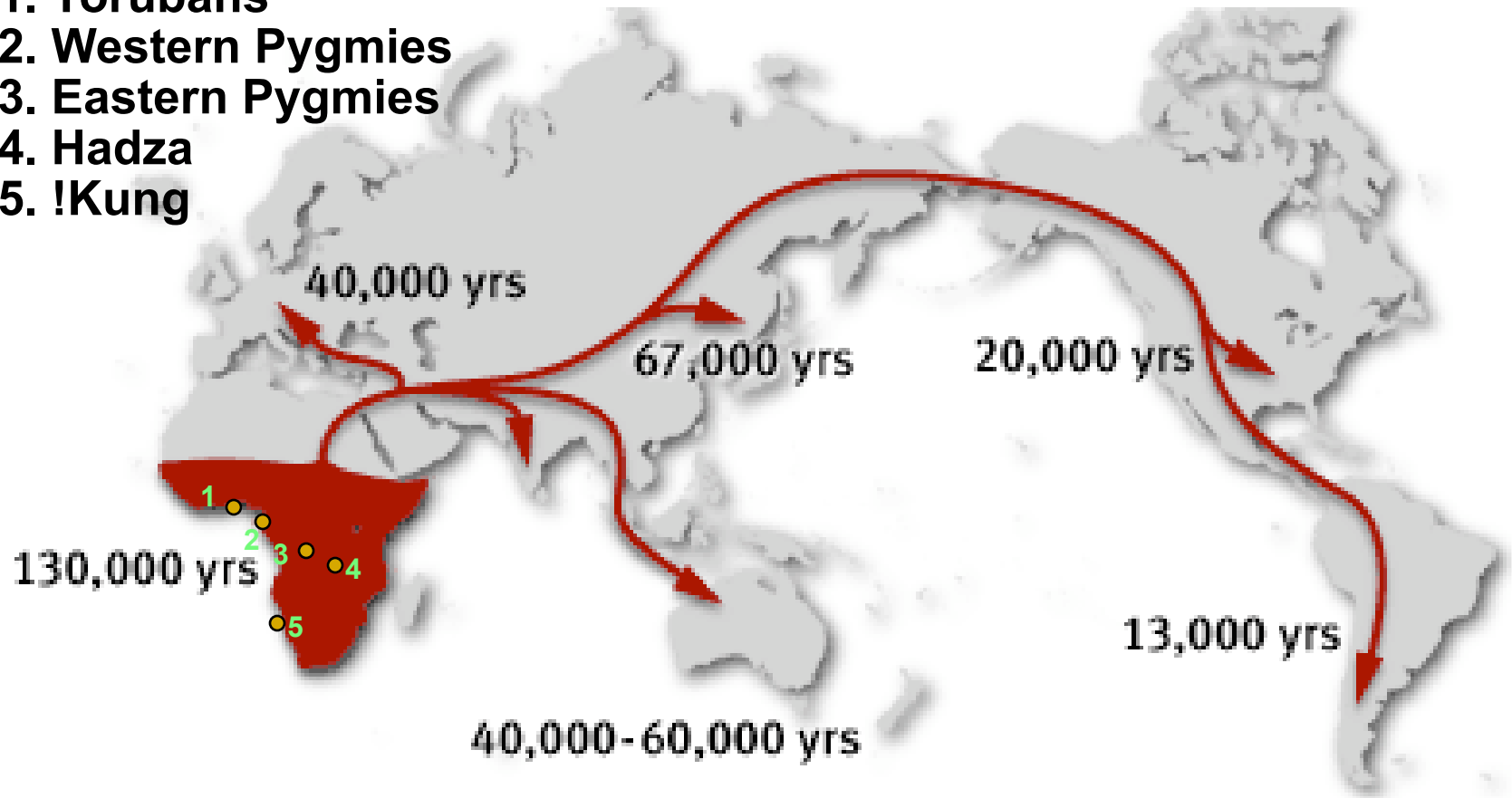
Evolutionary Tree of Humans: (microsatellites)

- Neighbor joining tree for 14 human populations genotyped with 30 microsatellite loci.



Human Migration Out of Africa

1. Yorubans
2. Western Pygmies
3. Eastern Pygmies
4. Hadza
5. !Kung



Two Neanderthal Discoveries

Feldhofer,
Germany

Mezmaiskaya,
Caucasus

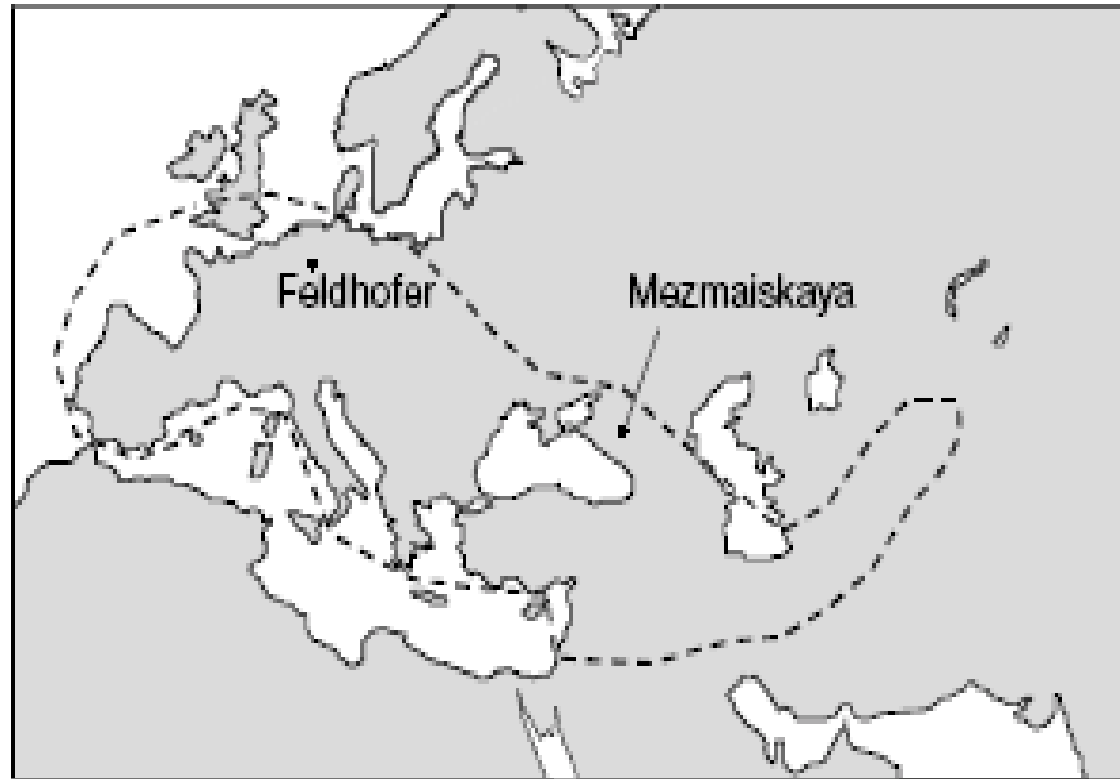
Distance:
25,000km



Two Neanderthal Discoveries



Figure 1. Illustration of Neanderthal Man
(Reprinted by permission from John Gurche/National Geographic.)

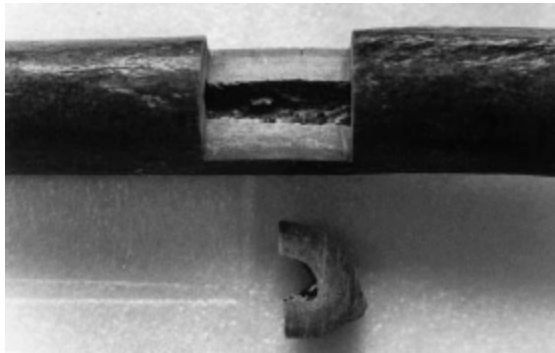


- *Is there a connection between Neanderthals and today's Europeans?*
- *If humans did not evolve from Neanderthals, whom did we evolve from?*

Multiregional Hypothesis?

- May predict some genetic continuity from the Neanderthals through to the Cro-Magnons up to today's Europeans
- Can explain the occurrence of varying regional characteristics

Sequencing Neanderthal's mtDNA



silico plug or cotton plug

- mtDNA from the bone of Neanderthal is used because it is up to 1,000x more abundant than nuclear DNA
- DNA decay overtime and only a small amount of ancient DNA can be recovered (upper limit: 100,000 years)
- PCR of mtDNA (fragments are too short, human DNA may mixed in)

Neanderthals vs Humans: surprisingly large divergence

- AMH vs Neanderthal:
 - 22 substitutions and 6 indels in 357 bp region
- AMH vs AMH
 - only 8 substitutions



Figure 1. Illustration of Neanderthal Man
(Reprinted by permission from John Gurche/National Geographic.)

Evolutionary Trees

How are these trees built from DNA sequences?

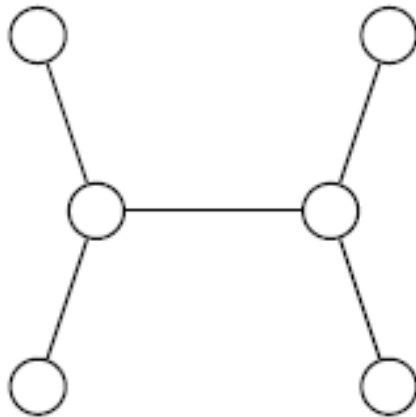
Evolutionary Trees

How are these trees built from DNA sequences?

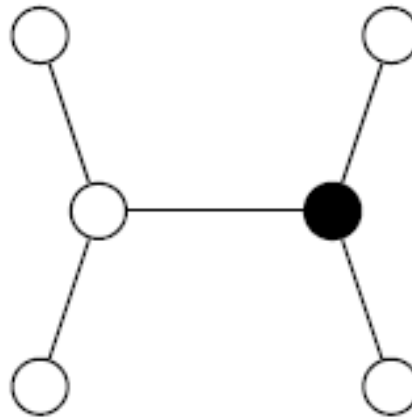
- leaves represent existing species
 - internal vertices represent ancestors
 - root represents the oldest evolutionary ancestor
-

Rooted and Unrooted Trees

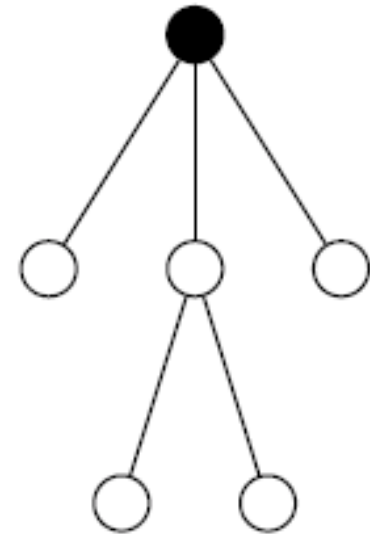
In the unrooted tree the position of the root (“oldest ancestor”) is unknown. Otherwise, they are like rooted trees



(a) Unrooted tree



(b) Rooted tree



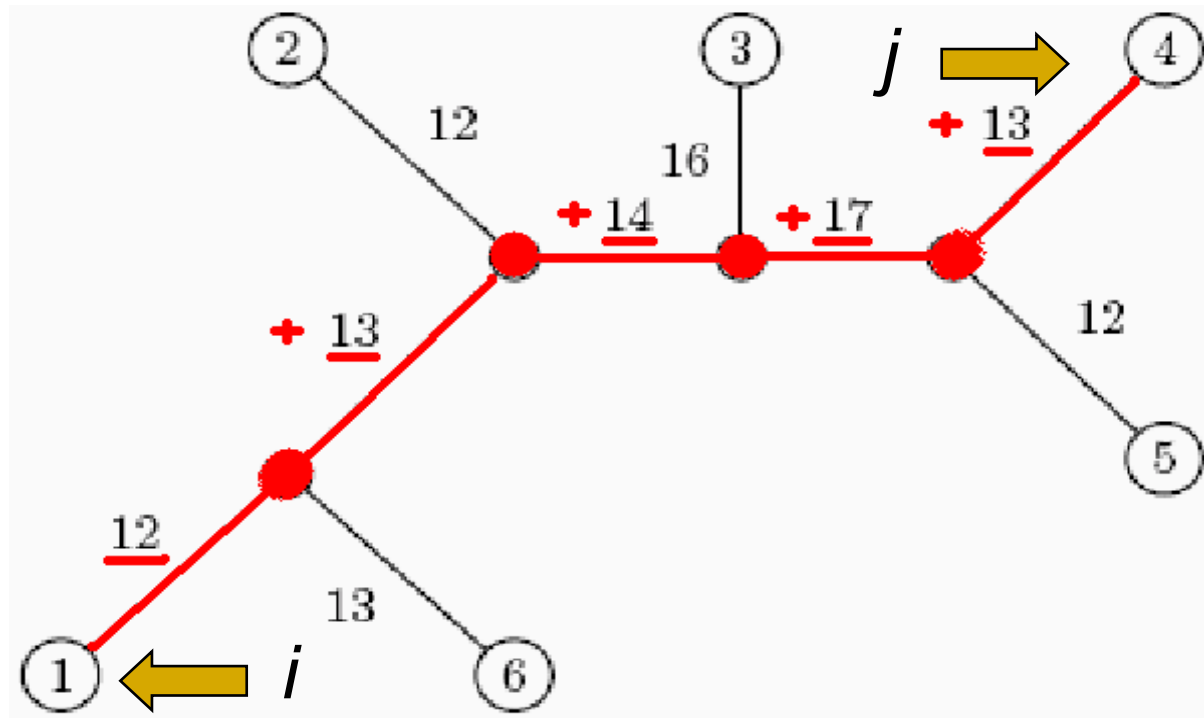
(c) The same rooted tree

Distances in Trees

- Edges may have weights reflecting:
 - Number of mutations on evolutionary path from one species to another
 - Time estimate for evolution of one species into another
- In a tree T , we often compute $d_{ij}(T)$ - the length of a path between leaves i and j

$d_{ij}(T)$ – ***tree distance between i and j***

Distance in Trees: an Example



$$d_{1,4} = 12 + 13 + 14 + 17 + 12 = 68$$

Distance Matrix

- Given n species, we can compute the $n \times n$ ***distance matrix*** D_{ij}
- D_{ij} may be defined as the edit distance between a gene in species i and species j , where the gene of interest is sequenced for all n species.

D_{ij} – edit distance between i and j

Edit Distance vs. Tree Distance

- Given n species, we can compute the $n \times n$ ***distance matrix*** D_{ij}
- D_{ij} may be defined as the edit distance between a gene in species i and species j , where the gene of interest is sequenced for all n species.

D_{ij} – ***edit distance between i and j***

- Note the difference with

$d_{ij}(T)$ – ***tree distance between i and j***

Fitting Distance Matrix

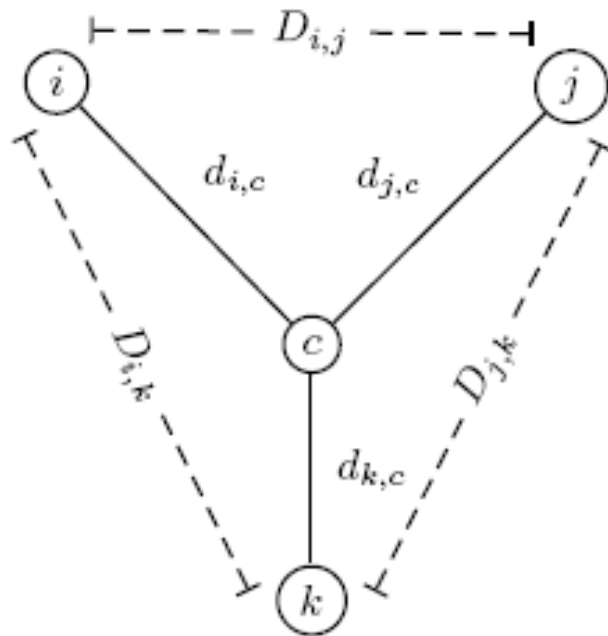
- Given n species, we can compute the $n \times n$ **distance matrix** D_{ij}
- Evolution of these genes is described by a tree that **we don't know**.
- We need an algorithm to construct a tree that best **fits** the distance matrix D_{ij}

Fitting Distance Matrix

- Fitting means $\underbrace{D_{ij}}_{\text{Edit distance between species (*known*)}} = \overbrace{d_{ij}(T)}^{\text{Lengths of path in an (*unknown*) tree } T}$

Reconstructing a 3 Leaved Tree

- Tree reconstruction for any 3x3 matrix is straightforward
- We have 3 leaves i, j, k and a center vertex c



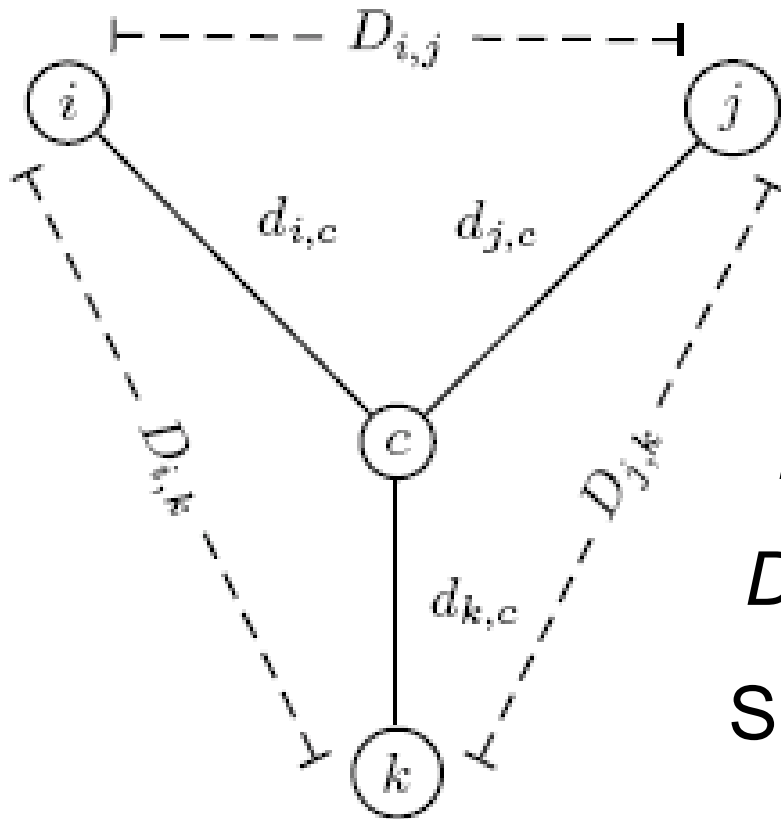
Observe:

$$d_{ic} + d_{jc} = D_{ij}$$

$$d_{ic} + d_{kc} = D_{ik}$$

$$d_{jc} + d_{kc} = D_{jk}$$

Reconstructing a 3 Leaved Tree (cont'd)



$$d_{ic} + d_{jc} = D_{ij}$$

$$+ \underline{d_{ic} + d_{kc} = D_{ik}}$$

$$\begin{aligned} 2d_{ic} + \underbrace{d_{jc} + d_{kc}}_{D_{jk}} &= D_{ij} + D_{ik} \\ 2d_{ic} + D_{jk} &= D_{ij} + D_{ik} \end{aligned}$$


Similarly,

$$\begin{aligned} d_{jc} &= (D_{ij} + D_{ik} - D_{jk})/2 \\ d_{kc} &= (D_{ki} + D_{kj} - D_{ij})/2 \end{aligned}$$

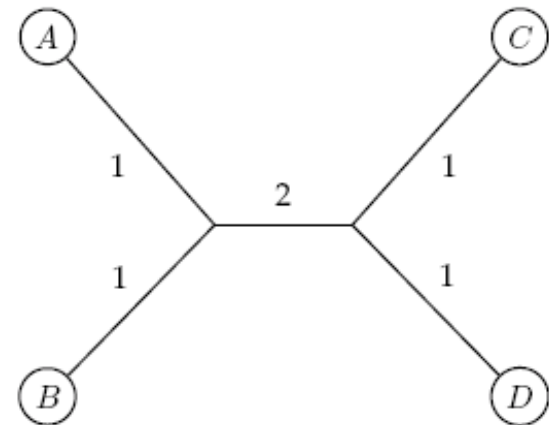
Trees with > 3 Leaves

- An tree with n leaves has $2n-3$ edges
- This means fitting a given tree to a distance matrix D requires solving a system of “ n choose 2” equations with $2n-3$ variables
- This is not always possible to solve for $n > 3$

Additive Distance Matrices

Matrix D is  ADDITIVE if there exists a tree T with $d_{ij}(T) = D_{ij}$

	A	B	C	D
A	0	2	4	4
B	2	0	4	4
C	4	4	0	2
D	4	4	2	0



NON-ADDITIVE
otherwise 

	A	B	C	D
A	0	2	2	2
B	2	0	3	2
C	2	3	0	2
D	2	2	2	0

?

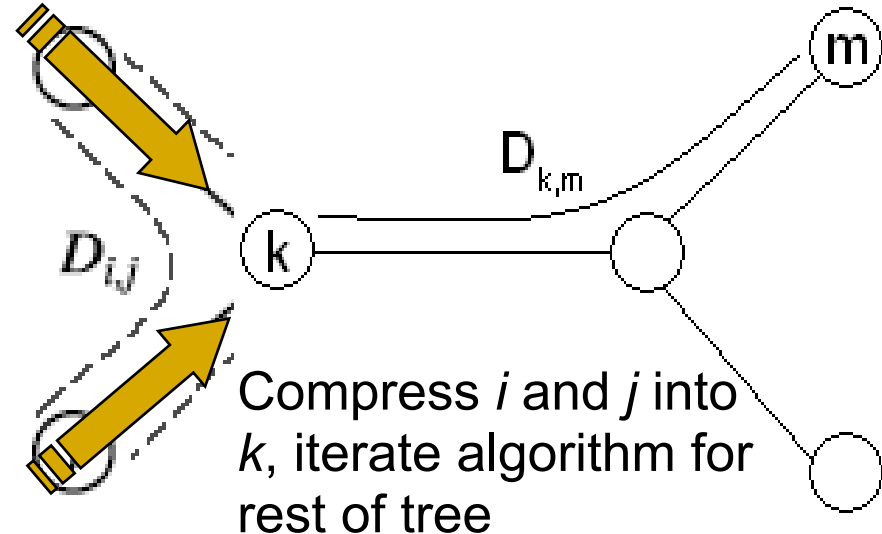
Distance Based Phylogeny Problem

- Goal: Reconstruct an evolutionary tree from a distance matrix
- Input: $n \times n$ distance matrix D_{ij}
- Output: weighted tree T with n leaves fitting D
- If D is additive, this problem has a solution and there is a simple algorithm to solve it

Using Neighboring Leaves to Construct the Tree

- Find **neighboring leaves** i and j with parent k
- Remove the rows and columns of i and j
- Add a new row and column corresponding to k , where the distance from k to any other leaf m can be computed as:

$$D_{km} = (D_{im} + D_{jm} - D_{ij})/2$$



Finding Neighboring Leaves

- To find neighboring leaves we simply select a pair of closest leaves.

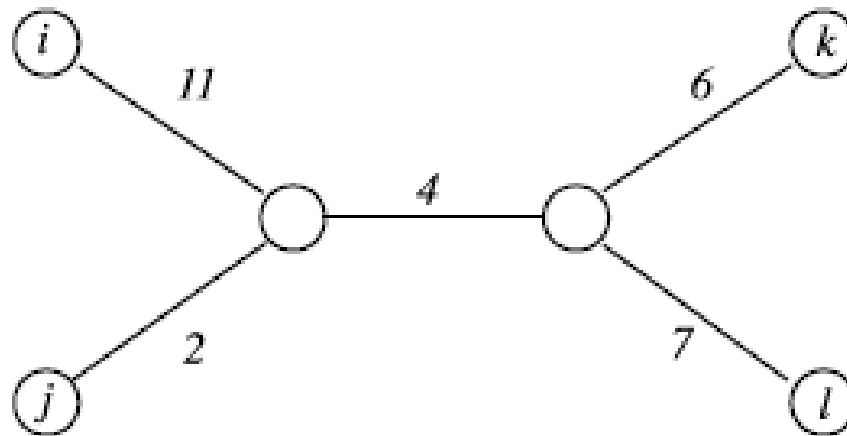
Finding Neighboring Leaves

- To find neighboring leaves we simply select a pair of closest leaves.

WRONG

Finding Neighboring Leaves

- Closest leaves aren't necessarily neighbors
- i and j are neighbors, but $(d_{ij} = 13) > (d_{jk} = 12)$



- Finding a pair of neighboring leaves is a nontrivial problem!

Neighbor Joining Algorithm

- In 1987 Naruya Saitou and Masatoshi Nei developed a neighbor joining algorithm for phylogenetic tree reconstruction
- **Finds a pair of leaves that are close to each other but far from other leaves:** implicitly finds a pair of neighboring leaves
- Advantages: works well for additive and other non-additive matrices, it does not have the flawed molecular clock assumption

Degenerate Triples

- A degenerate triple is a set of three distinct elements $1 \leq i, j, k \leq n$ where $D_{ij} + D_{jk} = D_{ik}$
- Element j in a degenerate triple i, j, k lies on the evolutionary path from i to k (or is attached to this path by an edge of length 0).

Looking for Degenerate Triples

- If distance matrix D **has** a degenerate triple i,j,k then j can be “removed” from D thus reducing the size of the problem.
- If distance matrix D **does not have** a degenerate triple i,j,k , *one can “create” a degenerative triple in D by shortening all hanging edges (in the tree).*

Shortening Hanging Edges to Produce Degenerate Triples

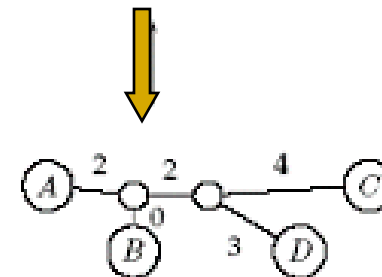
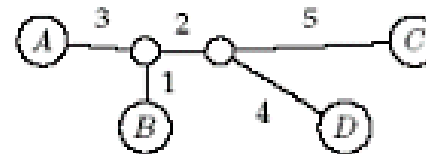
- Shorten all “hanging” edges (edges that connect leaves) until a degenerate triple is found

	A	B	C	D
A	0	4	10	9
B	4	0	8	7
C	10	8	0	9
D	9	7	9	0

$\delta = 1$

	A	B	C	D
A	0	2	8	7
B	2	0	6	5
C	8	6	0	7
D	7	5	7	0

$i \leftarrow A$
 $j \leftarrow B$
 $k \leftarrow C$



Finding Degenerate Triples

- If there is no degenerate triple, all hanging edges are reduced by the same amount δ , so that all pairwise distances in the matrix are reduced by 2δ .
- Eventually this process collapses one of the leaves (when $\delta = \text{length of shortest hanging edge}$), forming a degenerate triple i, j, k and reducing the size of the distance matrix D .
- The attachment point for j can be recovered in the reverse transformations by saving D_{ij} for each collapsed leaf.

Reconstructing Trees for Additive Distance Matrices

	A	B	C	D
A	0	4	10	9
B	4	0	8	7
C	10	8	0	9
D	9	7	9	0

$\delta = 1$

	A	B	C	D
A	0	2	8	7
B	2	0	6	5
C	8	6	0	7
D	7	5	7	0

$i \leftarrow A$
 $j \leftarrow B$
 $k \leftarrow C$

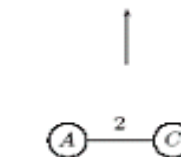
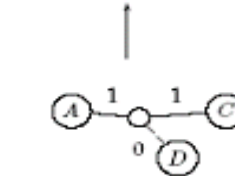
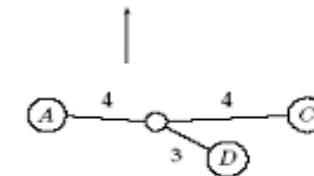
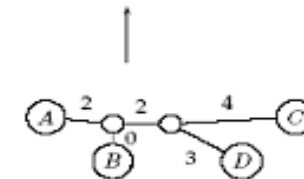
	A	C	D
A	0	8	7
C	8	0	7
D	7	7	0

$\delta = 3$

	A	C	D
A	0	2	1
C	2	0	1
D	1	1	0

$i \leftarrow A$
 $j \leftarrow D$
 $k \leftarrow C$

	A	C
A	0	2
C	2	0



AdditivePhylogeny Algorithm

1. AdditivePhylogeny(D)
2. if D is a 2 x 2 matrix
3. T = tree of a single edge of length $D_{1,2}$
4. return T
5. if D is non-degenerate
6. = trimming parameter of matrix D
7. for all $1 \leq i, j \leq n$
8. $D_{ij} = D_{ij} - 2$
9. else
10. = 0

AdditivePhylogeny (cont'd)

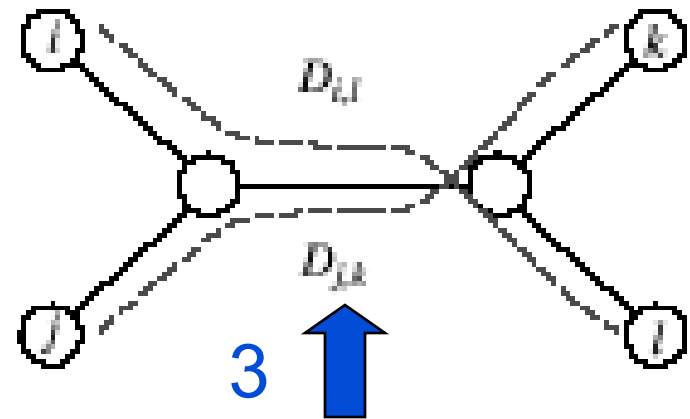
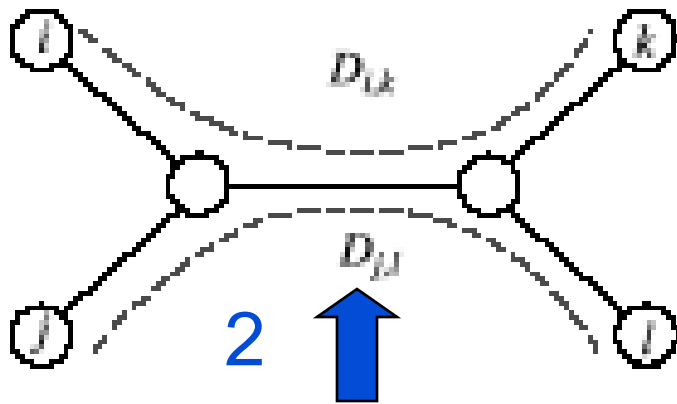
1. Find a triple i, j, k in D such that $D_{ij} + D_{jk} = D_{ik}$
2. $x = D_{ij}$
3. Remove j^{th} row and j^{th} column from D
4. $T = \text{AdditivePhylogeny}(D)$
5. Add a new vertex v to T at distance x from i to k
6. Add j back to T by creating an edge (v, j) of length 0
7. for every leaf l in T
8. if distance from l to v in the tree $\neq D_{l,j}$
9. output “matrix is not additive”
10. return
11. Extend all “hanging” edges by length
12. return T

The Four Point Condition

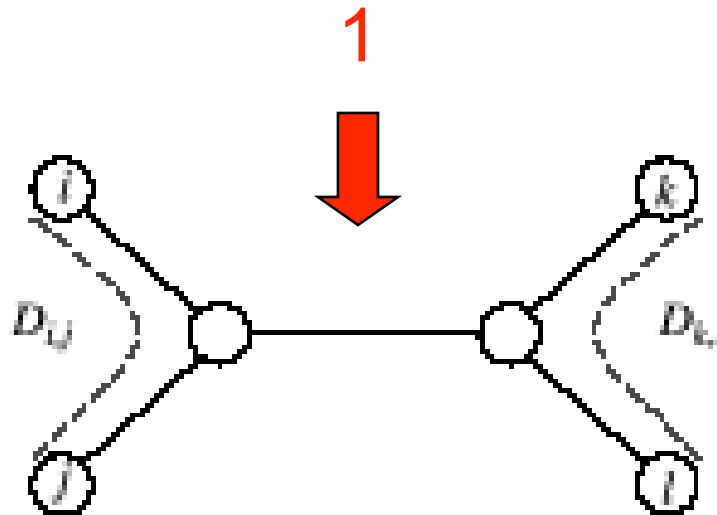
- AdditivePhylogeny provides a way to check if distance matrix D is additive
- **An even more efficient additivity check is the “four-point condition”**
- Let $1 \leq i, j, k, l \leq n$ be four distinct leaves in a tree

The Four Point Condition (cont'd)

Compute: 1. $D_{ij} + D_{kl}$, 2. $D_{ik} + D_{jl}$, 3. $D_{il} + D_{jk}$



2 and **3** represent the **same** number: **the length of all edges + the middle edge (it is counted twice)**



1 represents a **smaller** number: **the length of all edges – the middle edge**

The Four Point Condition: Theorem

- The four point condition for the quartet i,j,k,l is satisfied if two of these sums are the same, with the third sum smaller than these first two
- **Theorem** : An $n \times n$ matrix D is additive if and only if the four point condition holds for **every** quartet $1 \leq i,j,k,l \leq n$

Least Squares Distance Phylogeny Problem

- If the distance matrix D is NOT additive, then we look for a tree T that approximates D the best:

$$\textbf{Squared Error} : \sum_{i,j} (d_{ij}(T) - D_{ij})^2$$

- Squared Error is a measure of the quality of the fit between distance matrix and the tree: we want to minimize it.
- **Least Squares Distance Phylogeny Problem:** finding the best approximation tree T for a non-additive matrix D (NP-hard).

UPGMA: Unweighted Pair Group Method with Arithmetic Mean

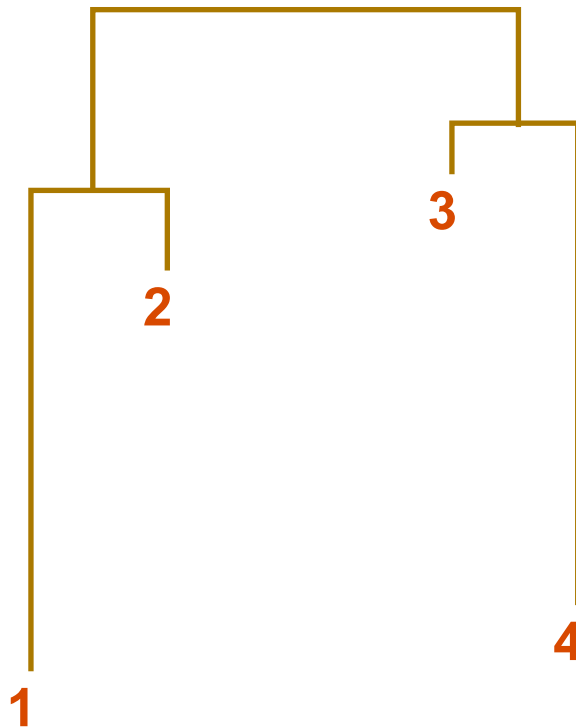
- UPGMA is a clustering algorithm that:
 - computes the distance between clusters using average pairwise distance
 - assigns a *height* to every vertex in the tree, effectively assuming the presence of a molecular clock and dating every vertex

UPGMA's Weakness

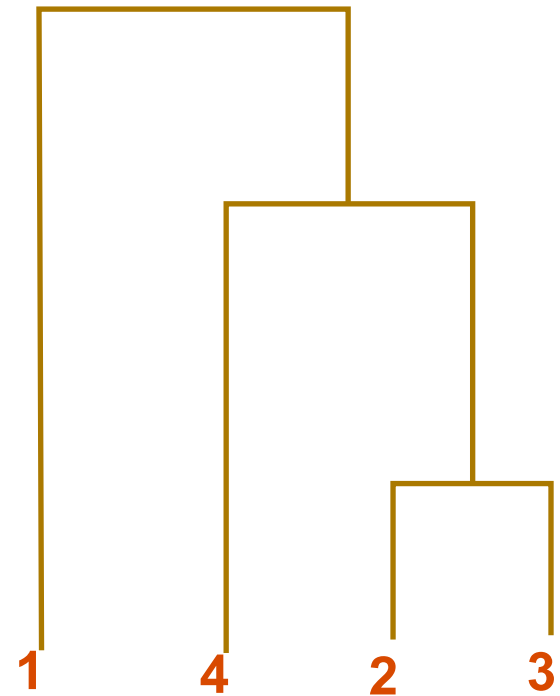
- The algorithm produces an **ultrametric** tree : the distance from the root to any leaf is the same
- UPGMA assumes a constant molecular clock: all species represented by the leaves in the tree are assumed to accumulate mutations (and thus evolve) at the same rate. This is a major pitfalls of UPGMA.

UPGMA's Weakness: Example

Correct tree



UPGMA



Clustering in UPGMA

Given two disjoint clusters C_i , C_j of sequences,

$$d_{ij} = \frac{1}{|C_i| + |C_j|} \sum_{\{p \in C_i, q \in C_j\}} d_{pq}$$

Note that if $C_k = C_i \cup C_j$, then distance to another cluster C_l is:

$$d_{kl} = \frac{d_{il}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}$$

UPGMA Algorithm

Initialization:

Assign each x_i to its own cluster C_i

Define one leaf per sequence, each at height 0

Iteration:

Find two clusters C_i and C_j such that d_{ij} is min

Let $C_k = C_i \dot{\cup} C_j$

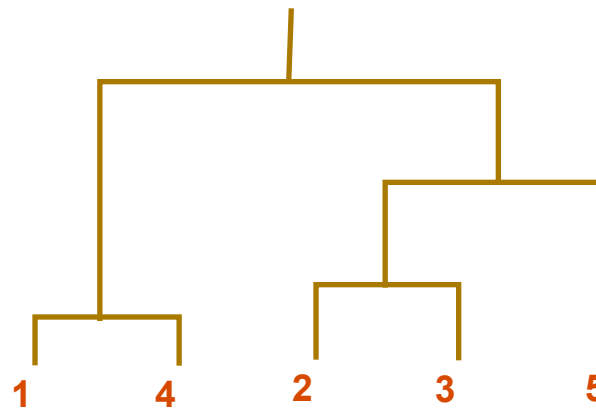
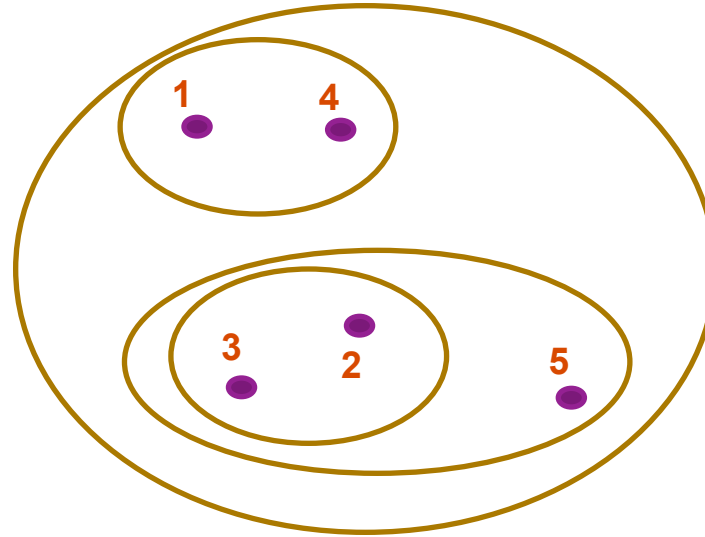
Add a vertex connecting C_i , C_j and place it at height $d_{ij} / 2$

Delete C_i and C_j

Termination:

When a single cluster remains

UPGMA Algorithm (cont'd)

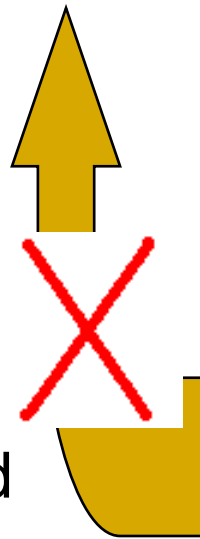


Alignment Matrix vs. Distance Matrix

Sequence a gene of length m
nucleotides in n species to generate an...

$n \times m$ alignment matrix

CANNOT be
transformed back
into alignment
matrix because
information was
lost on the forward
transformation



Transform
into...

$n \times n$ distance
matrix

Character-Based Tree Reconstruction

- **Better technique:**
 - Character-based reconstruction algorithms use the $n \times m$ alignment matrix
($n = \# \text{ species}$, $m = \# \text{ characters}$)
directly instead of using distance matrix.
 - **GOAL:** determine what character strings at internal nodes would best explain the character strings for the n observed species

Character-Based Tree Reconstruction

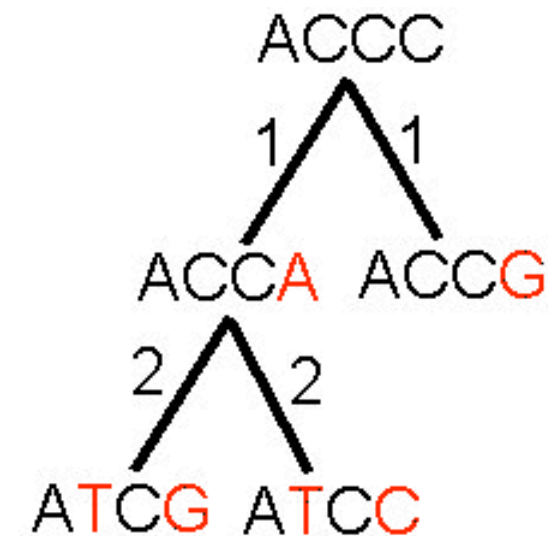
(cont'd)

- Characters may be nucleotides, where A, G, C, T are **states** of this character. Other characters may be the # of eyes or legs or the shape of a beak or a fin.
- By setting the length of an edge in the tree to the Hamming distance, we may define the **parsimony score** of the tree as the sum of the lengths (weights) of the edges

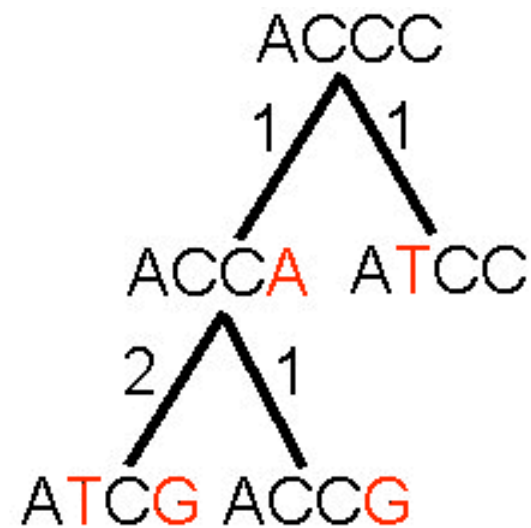
Parsimony Approach to Evolutionary Tree Reconstruction

- Applies Occam's razor principle to identify the simplest explanation for the data
- Assumes observed character differences resulted from the fewest possible mutations
- Seeks the tree that yields lowest possible **parsimony score** - sum of cost of all mutations found in the tree

Parsimony and Tree Reconstruction

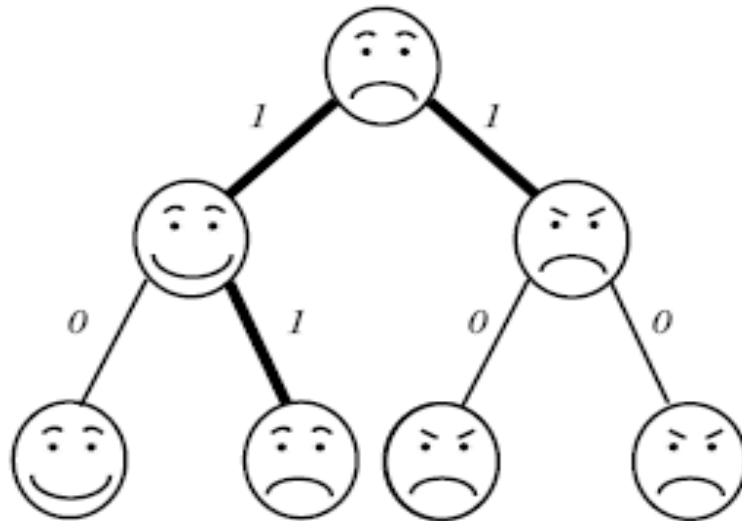


**Less
Parsimonious
Score: 6**

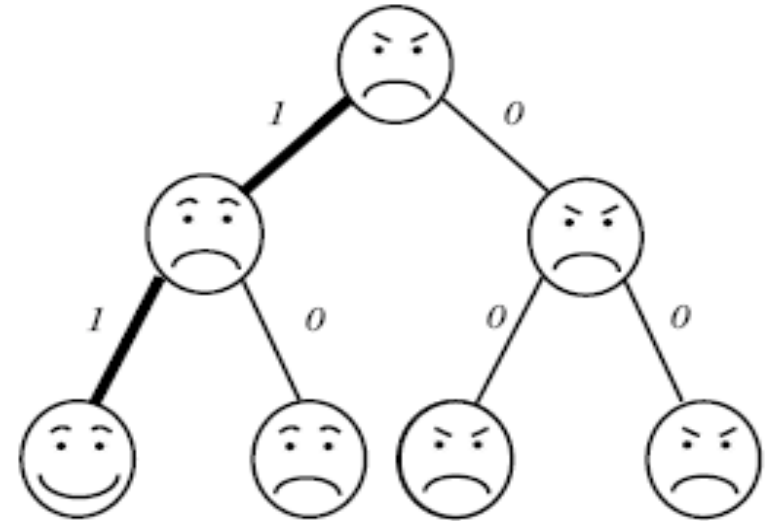


**More
Parsimonious
Score: 5**

Character-Based Tree Reconstruction (cont'd)



(a) *Parsimony Score=3*



(b) *Parsimony Score=2*

Figure 10.16 If we label a tree's leaves with characters (in this case, eyebrows and mouth, each with two states), and choose labels for each internal vertex, we implicitly create a *parsimony* score for the tree. By changing the labels in (a) we are able to create a tree with a better parsimony score in (b).

Small Parsimony Problem

- Input: Tree T with each leaf labeled by an m -character string.
- Output: Labeling of internal vertices of the tree T minimizing the parsimony score.
- We can assume that every leaf is labeled by a single character, because the characters in the string are independent.

Weighted Small Parsimony Problem

- A more general version of Small Parsimony Problem
- Input includes a $k * k$ scoring matrix describing the cost of transformation of each of k states into another one
- For Small Parsimony problem, the scoring matrix is based on Hamming distance

$$d_H(v, w) = 0 \text{ if } v=w$$

$$d_H(v, w) = 1 \text{ otherwise}$$

Scoring Matrices

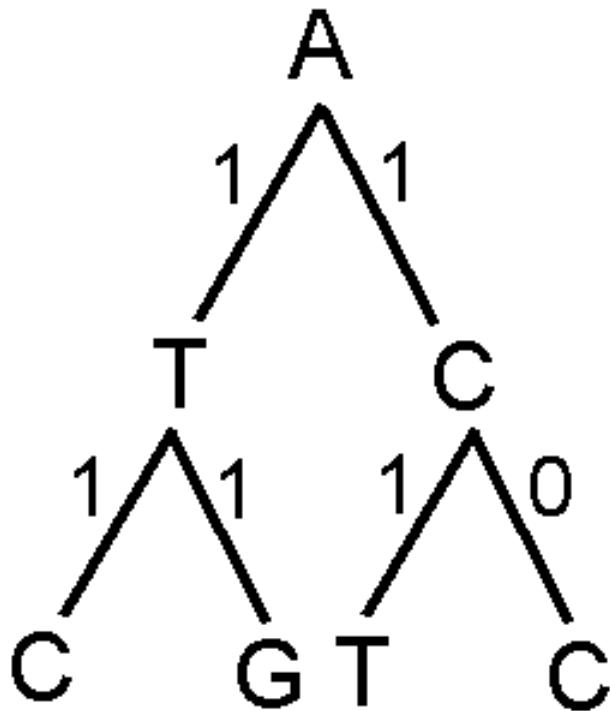
Small Parsimony Problem

	A	T	G	C
A	0	1	1	1
T	1	0	1	1
G	1	1	0	1
C	1	1	1	0

Weighted Parsimony Problem

	A	T	G	C
A	0	3	4	9
T	3	0	2	4
G	4	2	0	4
C	9	4	4	0

Unweighted vs. Weighted

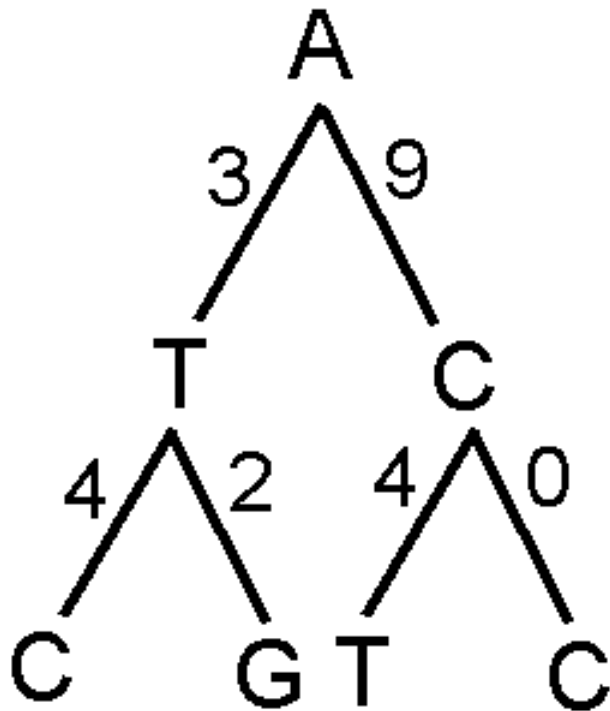


Small Parsimony Scoring Matrix:

	A	T	G	C
A	0	1	1	1
T	1	0	1	1
G	1	1	0	1
C	1	1	1	0

Small Parsimony Score: 5

Unweighted vs. Weighted



Weighted Parsimony Scoring Matrix:

	A	T	G	C
A	0	3	4	9
T	3	0	2	4
G	4	2	0	4
C	9	4	4	0

Weighted Parsimony Score: 22

Weighted Small Parsimony

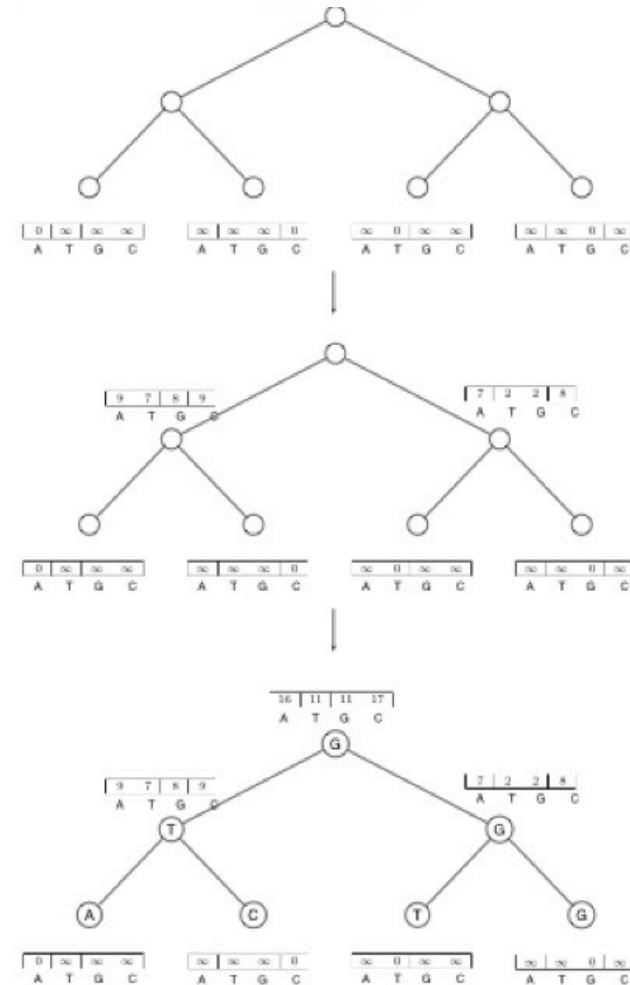
Problem: Formulation

- Input: Tree T with each leaf labeled by elements of a k -letter alphabet and a $k \times k$ scoring matrix (d_{ij})
- Output: Labeling of internal vertices of the tree T minimizing the weighted parsimony score

Sankoff's Algorithm

- Check children's every vertex and determine the minimum between them
- An example

δ	A	T	G	C
A	0	3	4	9
T	3	0	2	4
G	4	2	0	4
C	9	4	4	0

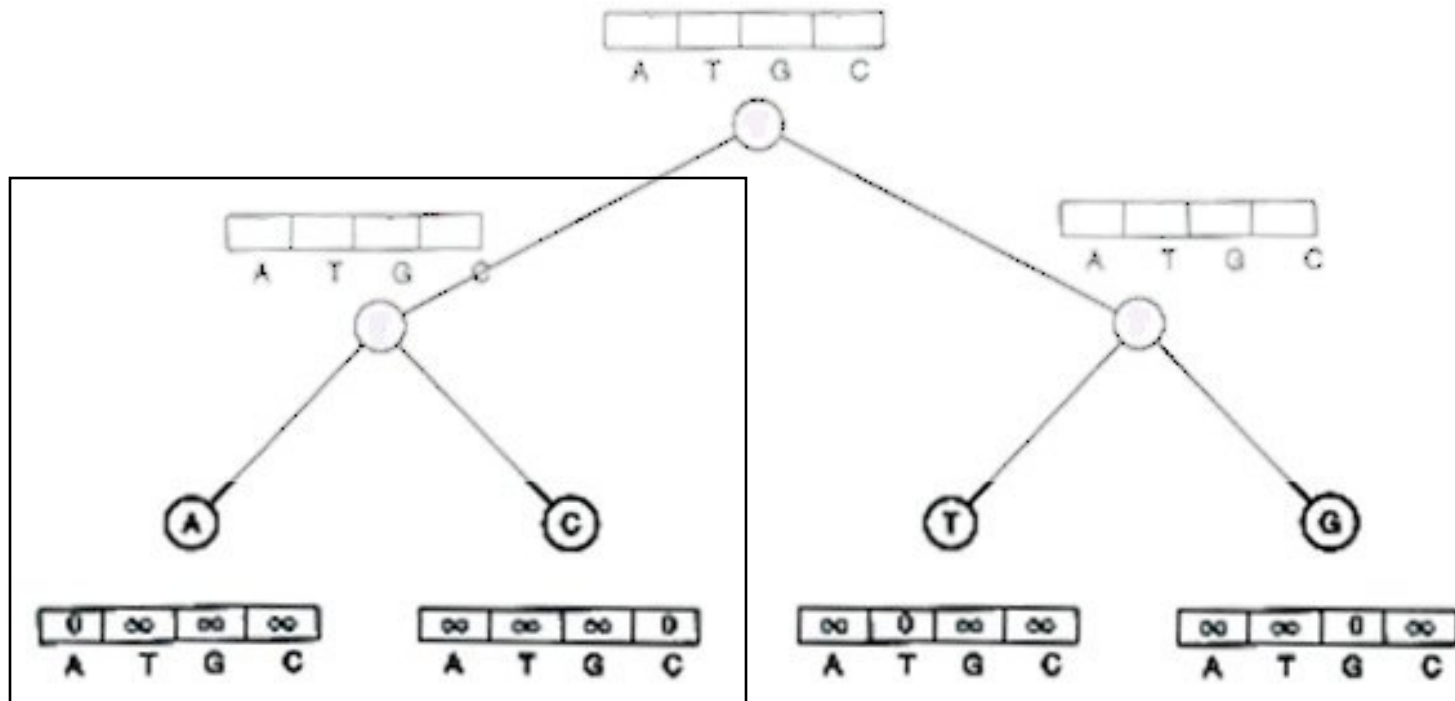


Sankoff Algorithm: Dynamic Programming

- Calculate and keep track of a score for every possible label at each vertex
 - $s_t(v)$ = minimum parsimony score of the **subtree** rooted at vertex v if v has character t
- The score at each vertex is based on scores of its children:
 - $s_t(\textit{parent}) = \min_i \{s_i(\textit{left child}) + d_{i,t}\} + \min_j \{s_j(\textit{right child}) + d_{j,t}\}$

Sankoff Algorithm (cont.)

- Begin at leaves:
 - If leaf has the character in question, score is 0
 - Else, score is ∞



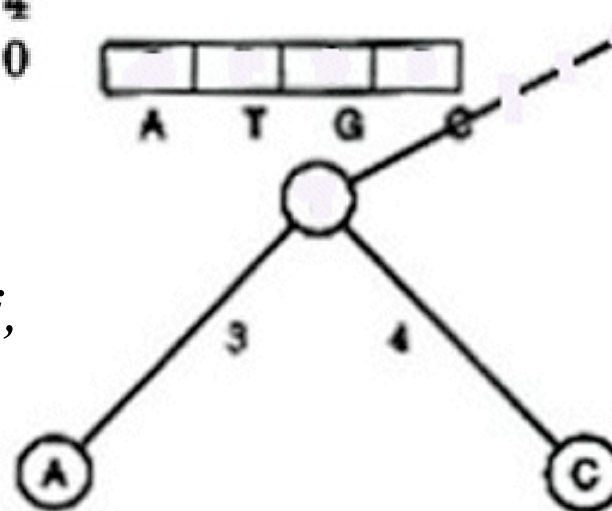
Sankoff Algorithm (cont.)

δ	A	T	G	C
A	0	3	4	9
T	3	0	2	4
G	4	2	0	4
C	9	4	4	0

$$s_t(v) = \min_i \{s_i(u) + d_{i,t}\} + \min_j \{s_j(w) + d_{j,t}\}$$

$$s_A(v) = 0$$

$$A\} + \min_j \{s_j(w) + d_{j,A}\}$$



	$s_i(u)$	$d_{i,t}$	sum
A	0	0	0
T	¥	3	¥
G	¥	4	¥
C	¥	9	¥

0	∞	∞	∞
A	T	G	C

∞	∞	∞	0
A	T	G	C

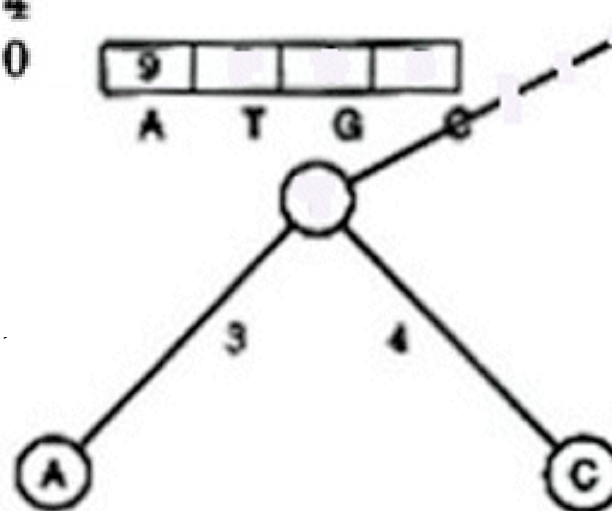
Sankoff Algorithm (cont.)

δ	A	T	G	C
A	0	3	4	9
T	3	0	2	4
G	4	2	0	4
C	9	4	4	0

$$s_t(v) = \min_i \{s_i(u) + d_{i,t}\} + \min_j \{s_j(w) + d_{j,t}\}$$

$$s_A(v) = 0 + 9 = 9$$

A



	$s_j(u)$	$d_{j,t}$	sum
A	¥	0	¥
T	¥	3	¥
G	¥	4	¥
C	0	9	9

0	∞	∞	∞
A	T	G	C

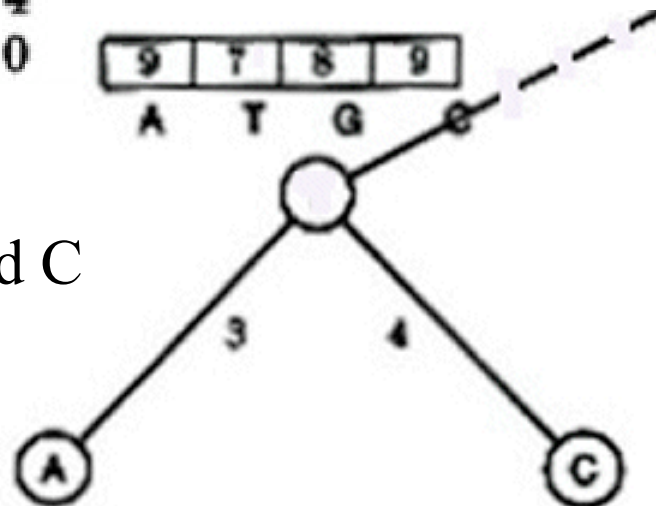
∞	∞	∞	0
A	T	G	C

Sankoff Algorithm (cont.)

δ	A	T	G	C
A	0	3	4	9
T	3	0	2	4
G	4	2	0	4
C	9	4	4	0

$$s_t(v) = \min_i \{s_i(u) + d_{i,t}\} + \min_j \{s_j(w) + d_{j,t}\}$$

Repeat for T, G, and C

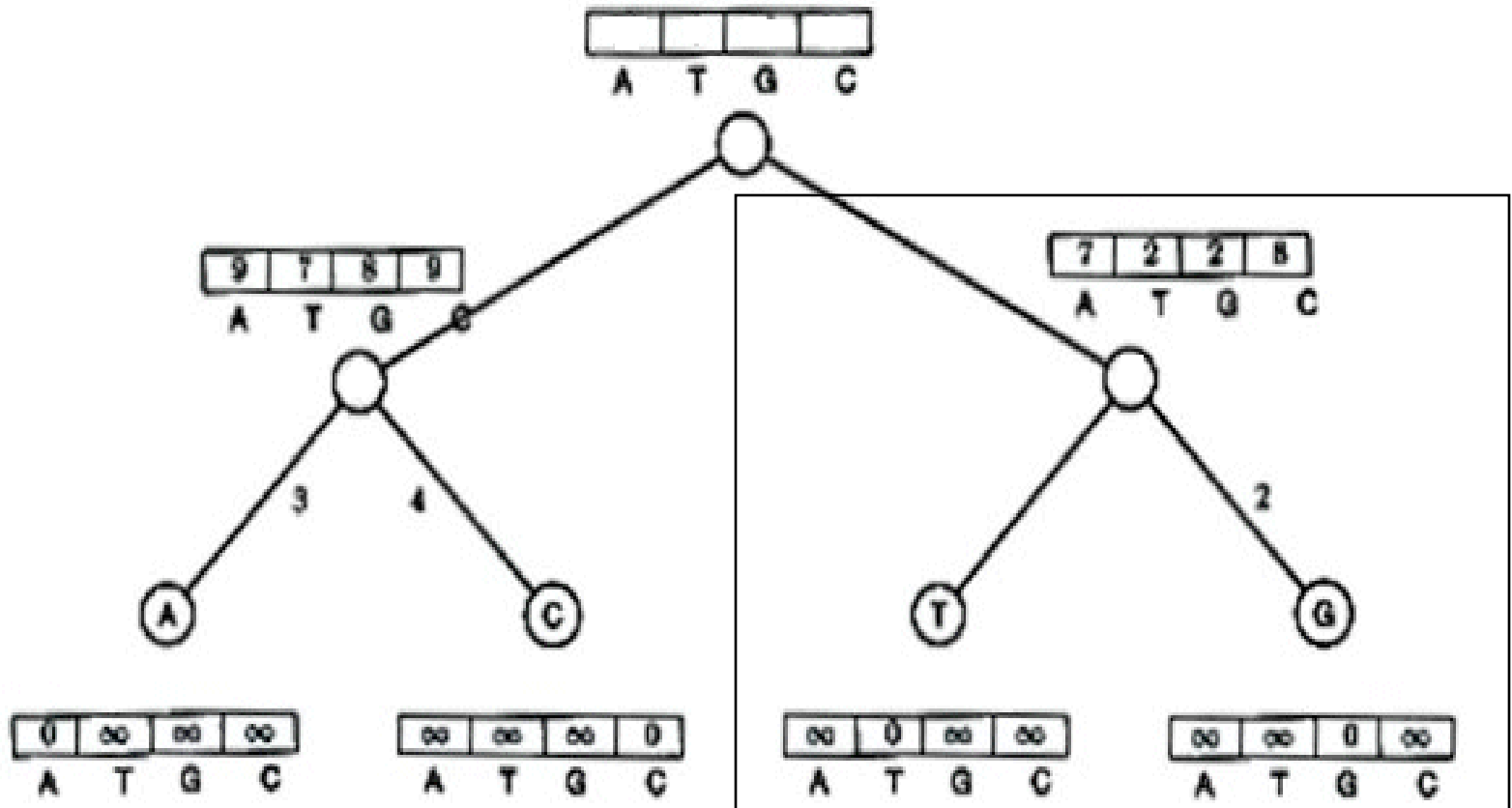


0	∞	∞	∞
A	T	G	C

∞	∞	∞	0
A	T	G	C

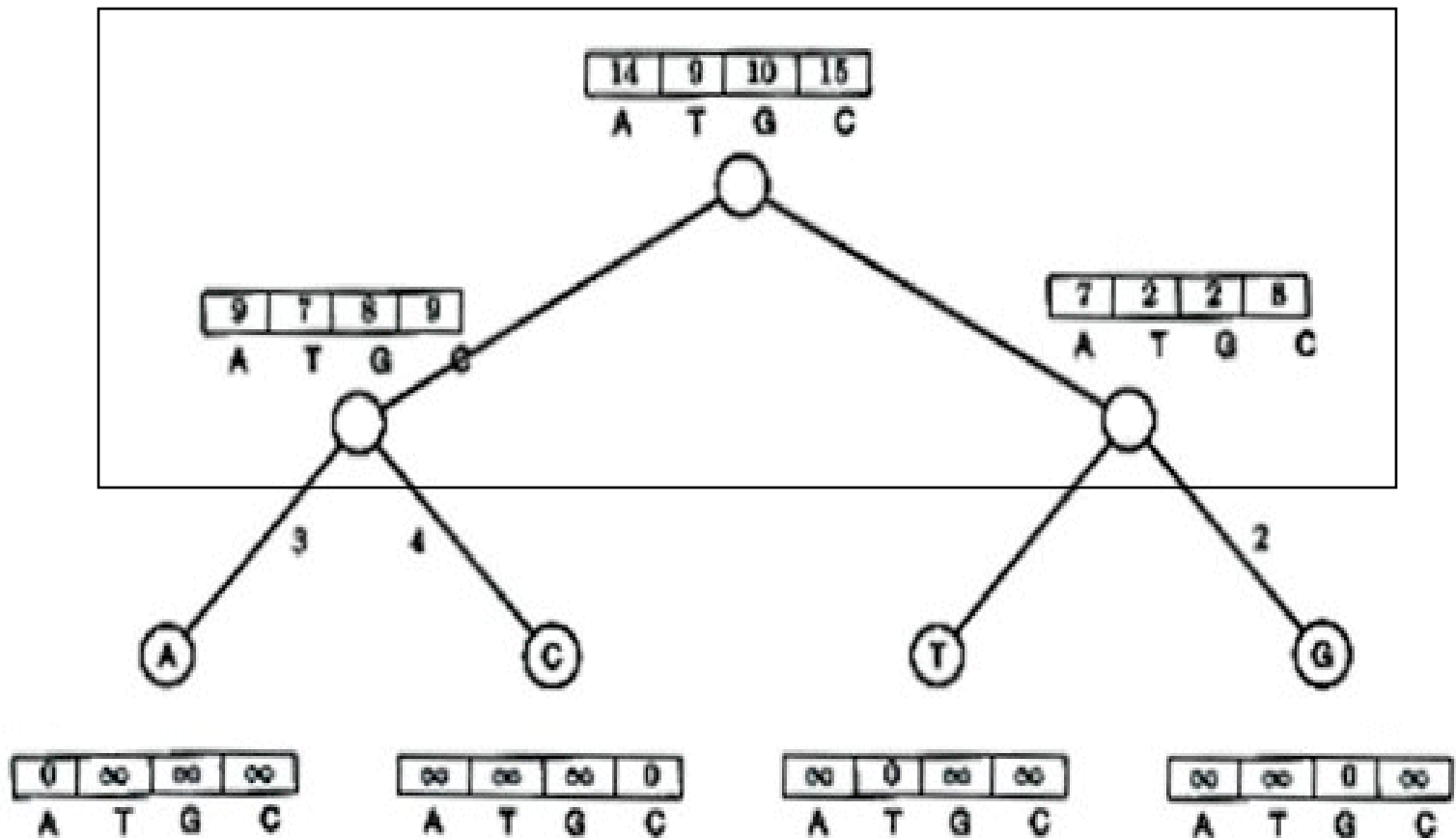
Sankoff Algorithm (cont.)

Repeat for right subtree



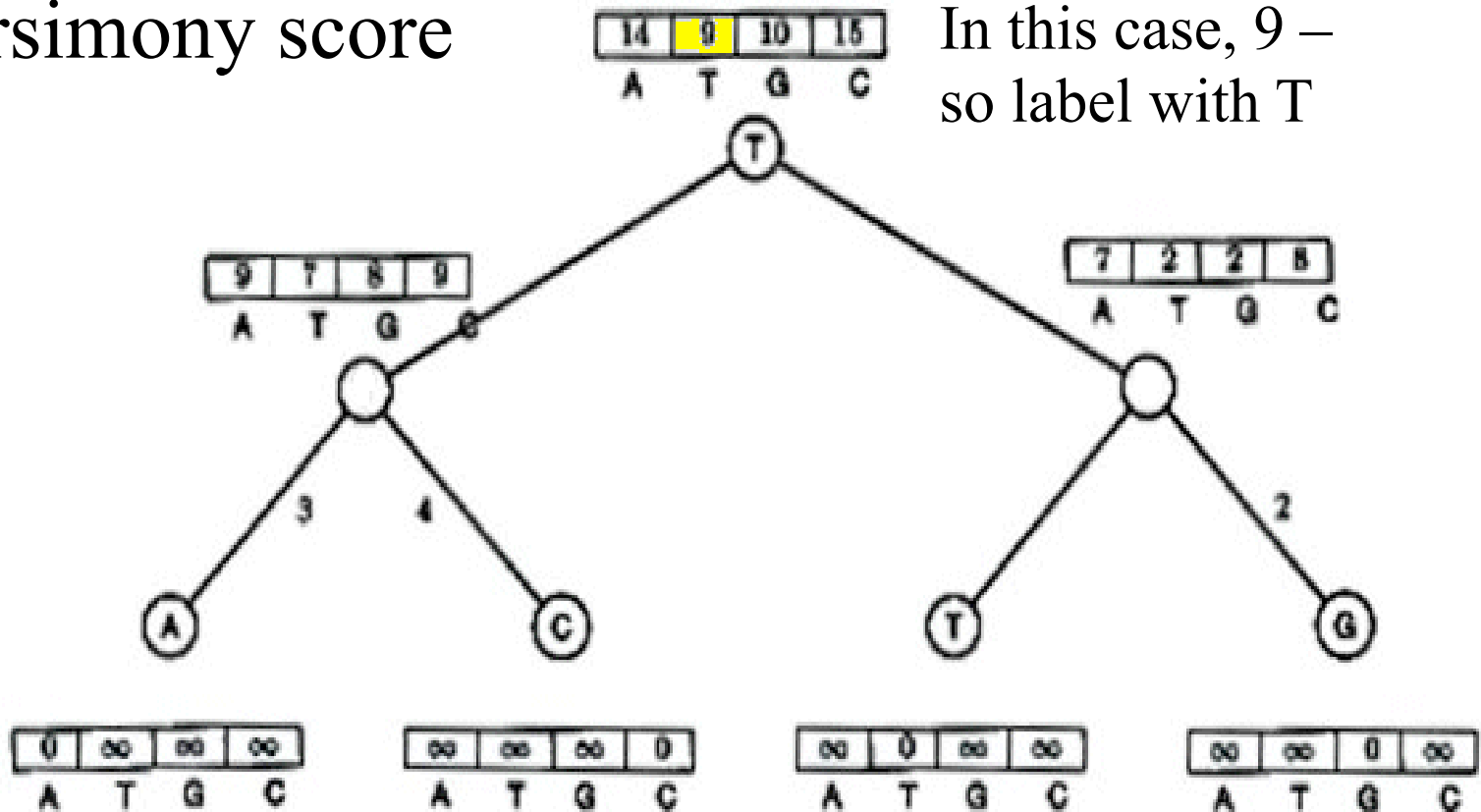
Sankoff Algorithm (cont.)

Repeat for root



Sankoff Algorithm (cont.)

Smallest score at root is minimum weighted parsimony score



Sankoff Algorithm: Traveling down the Tree

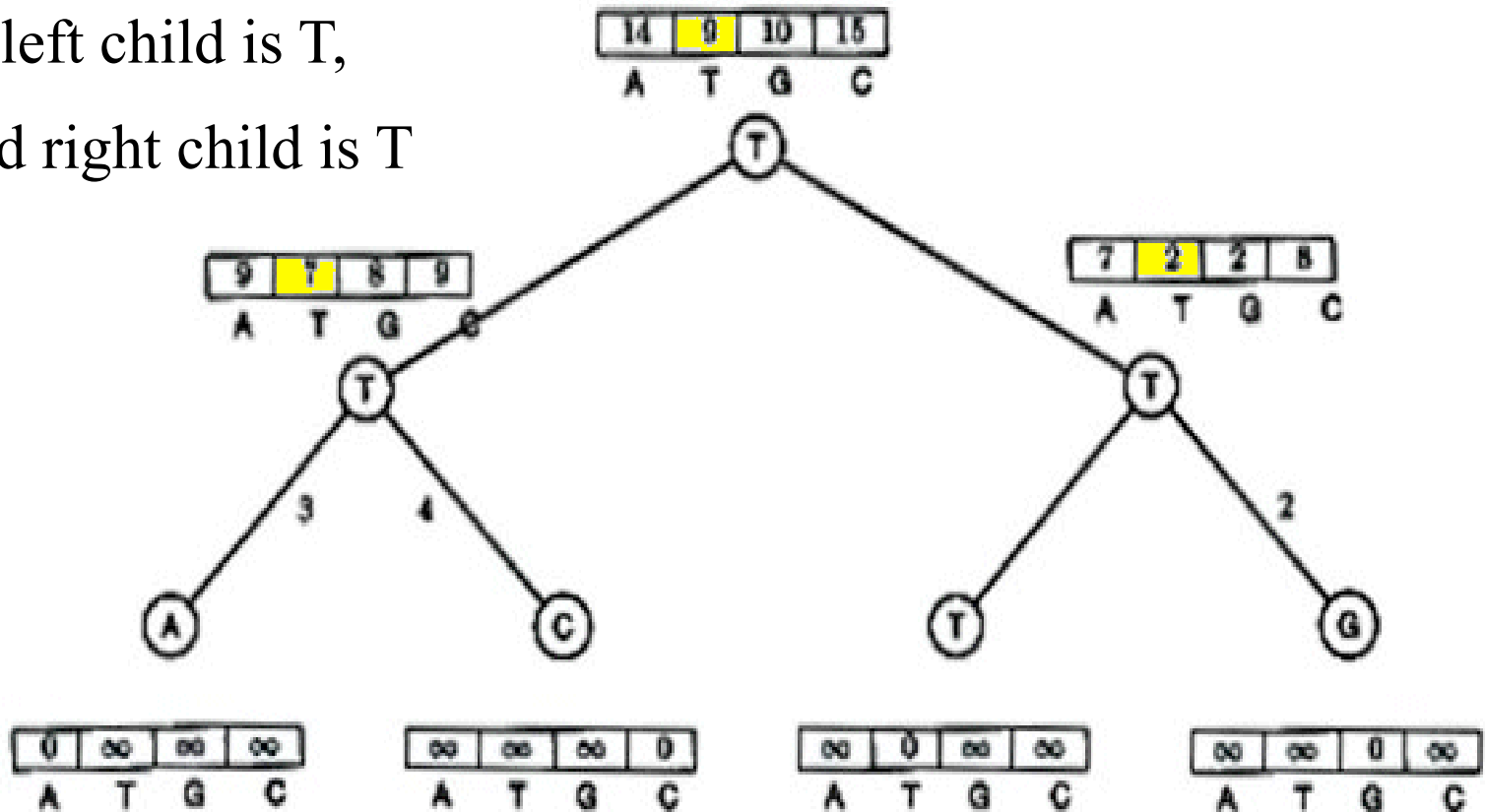
- The scores at the root vertex have been computed by going up the tree
- After the scores at root vertex are computed the Sankoff algorithm moves down the tree and assign each vertex with optimal character.

Sankoff Algorithm (cont.)

9 is derived from $7 + 2$

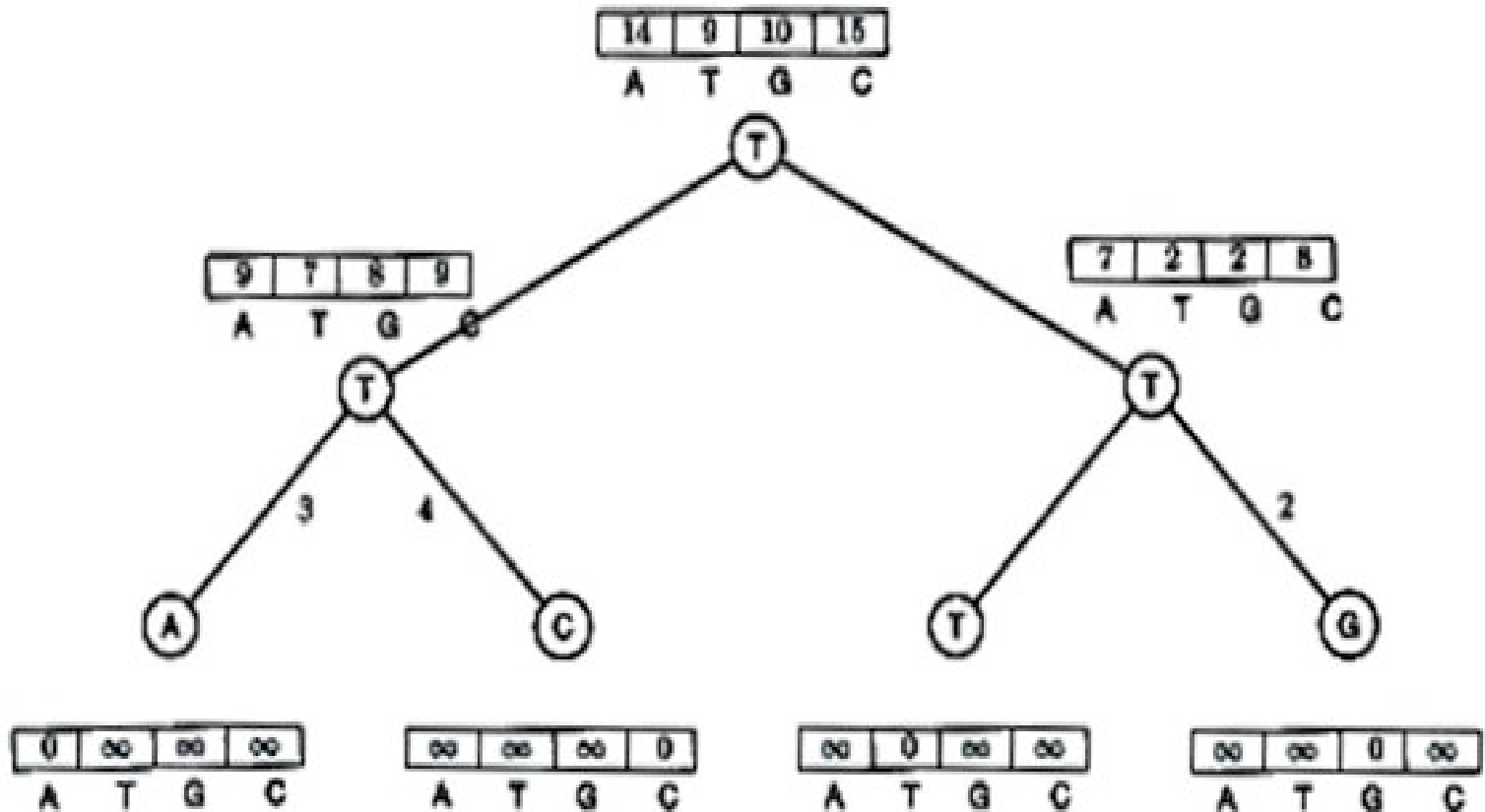
So left child is T,

And right child is T



Sankoff Algorithm (cont.)

And the tree is thus labeled...

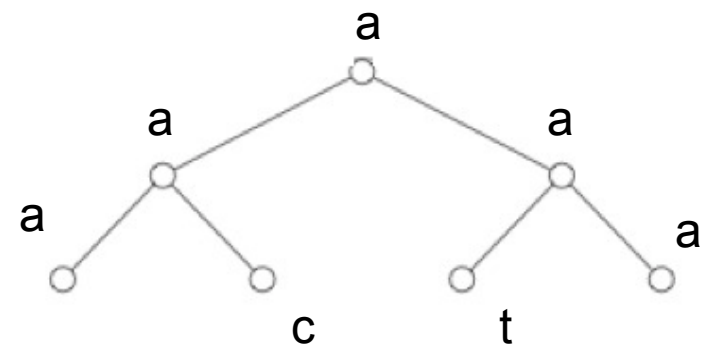
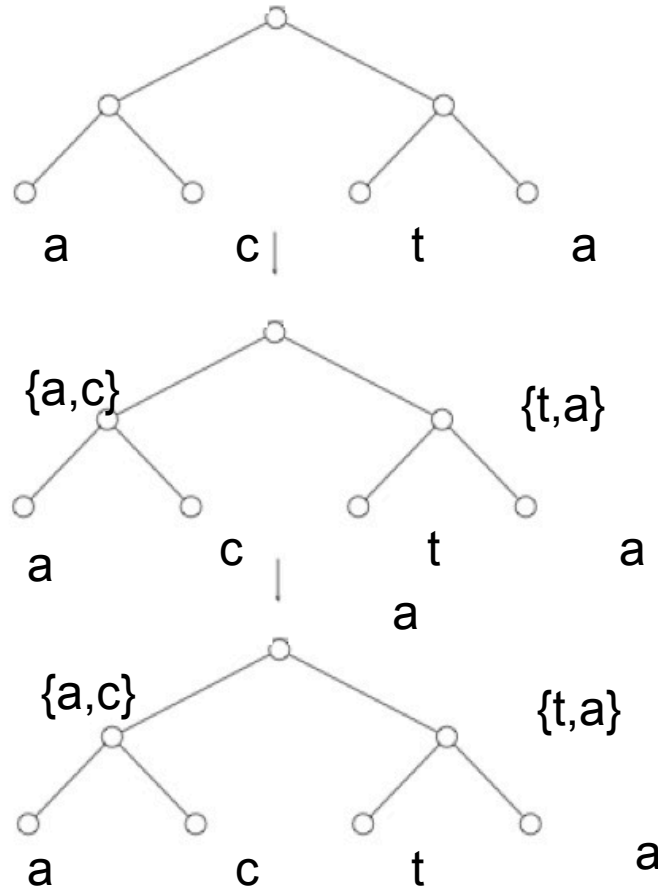


Fitch's Algorithm

- Solves Small Parsimony problem
 - Dynamic programming in essence
 - Assigns a set of letter to every vertex in the tree.
 - If the two children's sets of character overlap, it's the common set of them
 - If not, it's the combined set of them.
-

Fitch's Algorithm (cont'd)

An example:



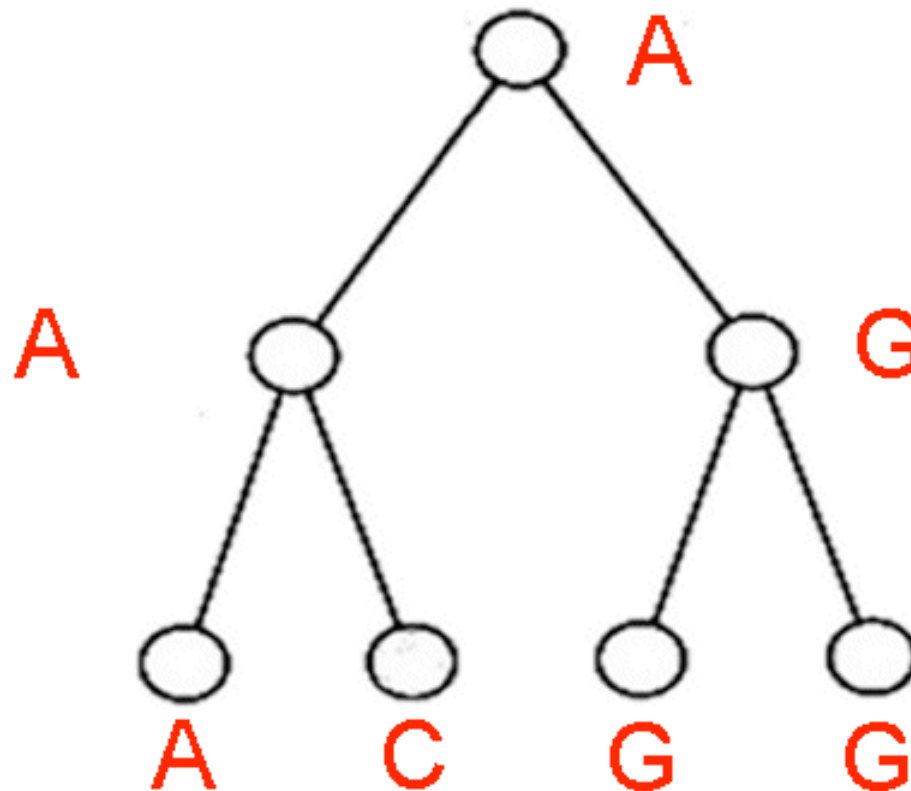
Fitch Algorithm

- 1) Assign a **set of possible letters** to every vertex, traversing the tree from leaves to root
- Each node's set is the combination of its children's sets (leaves contain their label)
 - E.g. if the node we are looking at has a left child labeled {A, C} and a right child labeled {A, T}, the node will be given the set {A, C, T}

Fitch Algorithm (cont.)

- 2) Assign **labels** to each vertex, traversing the tree from root to leaves
- Assign root arbitrarily from its set of letters
 - For all other vertices, if its parent's label is in its set of letters, assign it its parent's label
 - Else, choose an arbitrary letter from its set as its label

Fitch Algorithm (cont.)

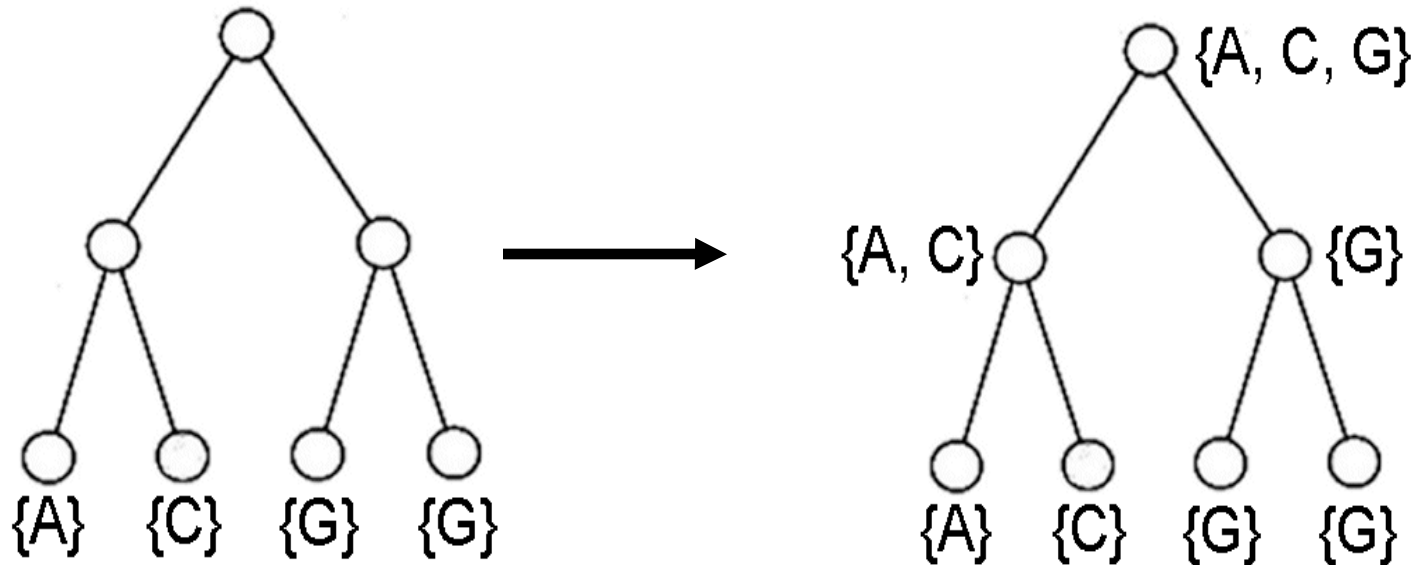


Fitch vs. Sankoff

- Both have an $O(nk)$ runtime
- Are they actually different?
- Let's compare ...

Fitch

As seen previously:



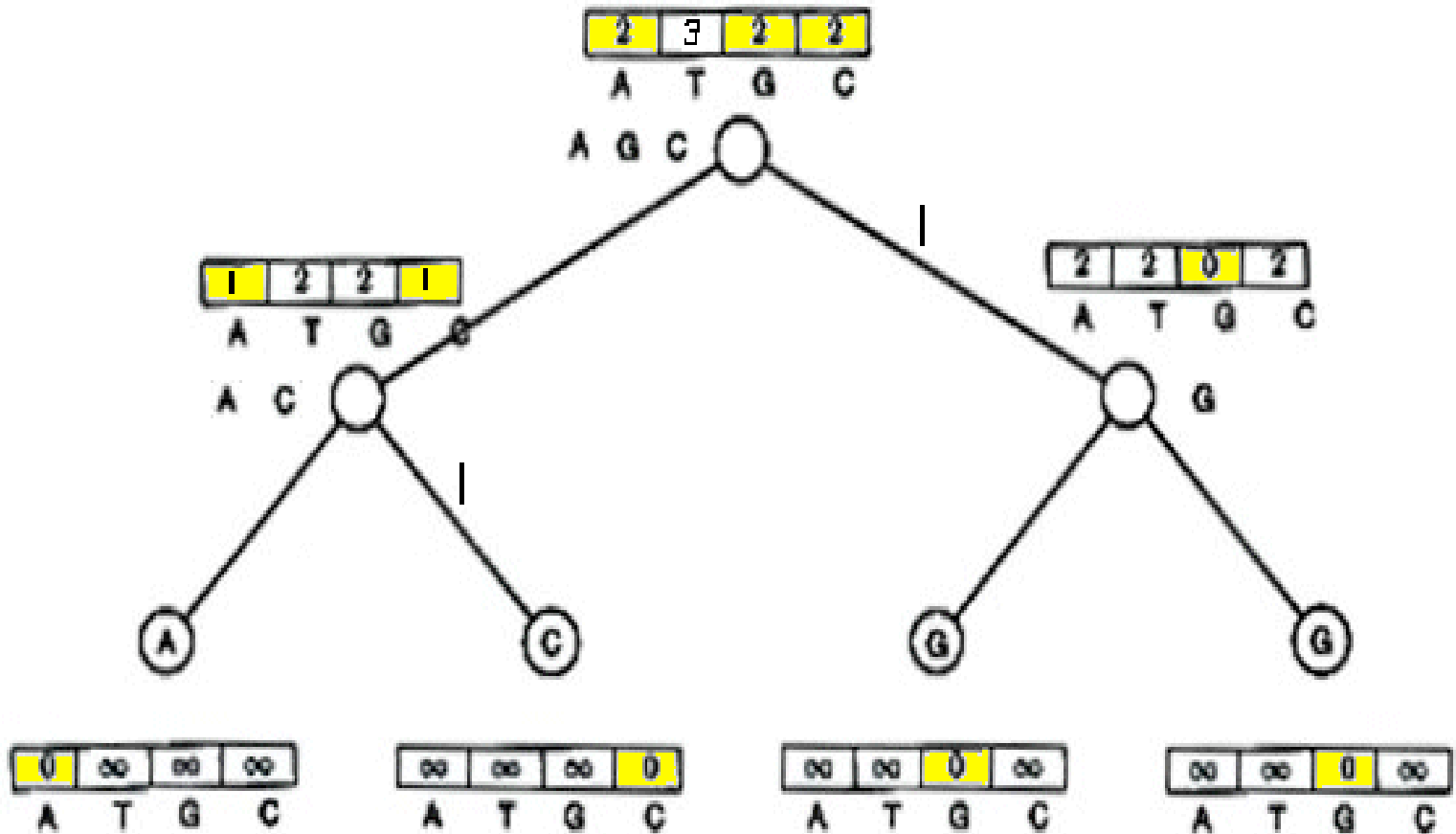
Comparison of Fitch and Sankoff

- As seen earlier, the scoring matrix for the Fitch algorithm is merely:

	A	T	G	C
A	0	1	1	1
T	1	0	1	1
G	1	1	0	1
C	1	1	1	0

- So let's do the same problem using Sankoff algorithm and this scoring matrix

Sankoff



Sankoff vs. Fitch

- The Sankoff algorithm gives the **same** set of **optimal** labels as the Fitch algorithm
- For Sankoff algorithm, character t is *optimal* for vertex v if $s_t(v) = \min_{1 \leq j \leq k} s_j(v)$
 - Denote the set of optimal letters at vertex v as $S(v)$
 - If $S(\text{left child})$ and $S(\text{right child})$ overlap, $S(\text{parent})$ is the intersection
 - Else it's the union of $S(\text{left child})$ and $S(\text{right child})$
 - This is also the Fitch recurrence
- The two algorithms are **identical**

Large Parsimony Problem

- Input: An $n \times m$ matrix M describing n species, each represented by an m -character string
- Output: A tree T with n leaves labeled by the n rows of matrix M , and a labeling of the internal vertices such that the parsimony score is minimized over all possible trees and all possible labelings of internal vertices

Large Parsimony Problem (cont.)

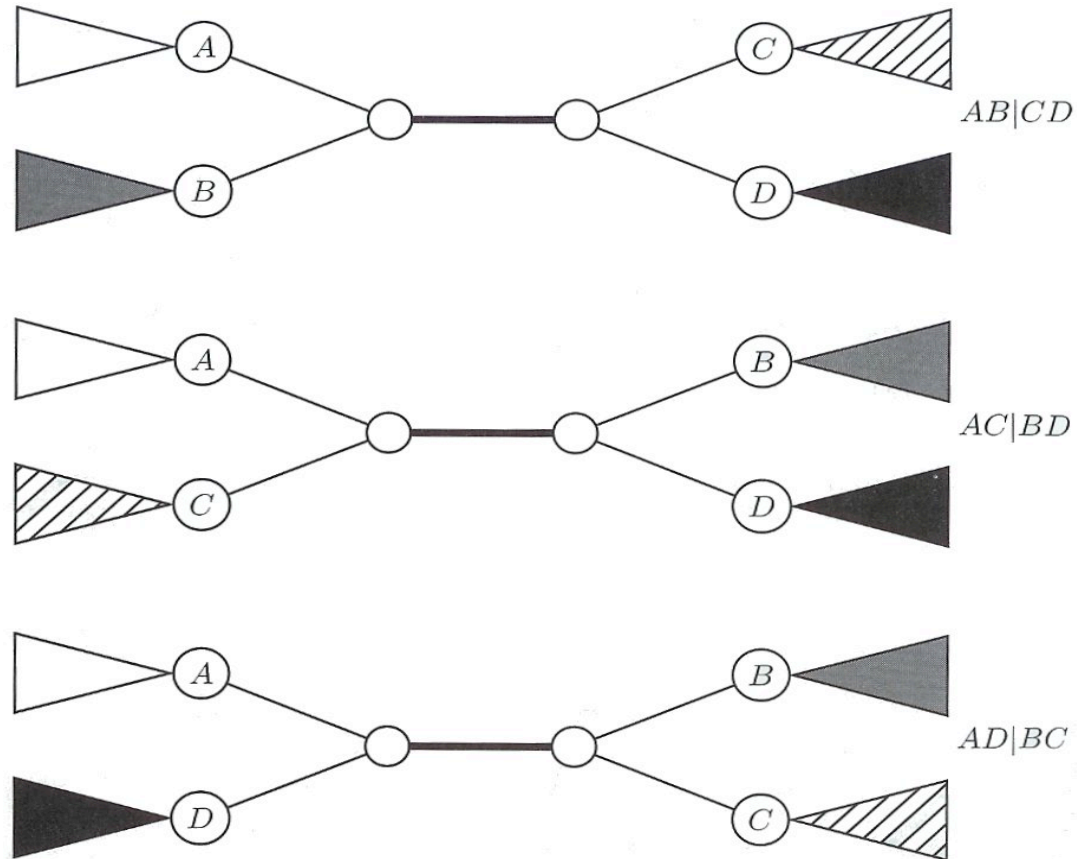
- Possible search space is huge, especially as n increases
 - $(2n - 3)!!$ possible rooted trees
 - $(2n - 5)!!$ possible unrooted trees
- Problem is NP-complete
 - Exhaustive search only possible w/ small $n(< 10)$
- Hence, branch and bound or heuristics used

Nearest Neighbor Interchange

A Greedy Algorithm

- A Branch Swapping algorithm
- Only evaluates a subset of all possible trees
- Defines a *neighbor* of a tree as one reachable by a *nearest neighbor interchange*
 - A rearrangement of the four subtrees defined by one internal edge
 - Only three different rearrangements per edge

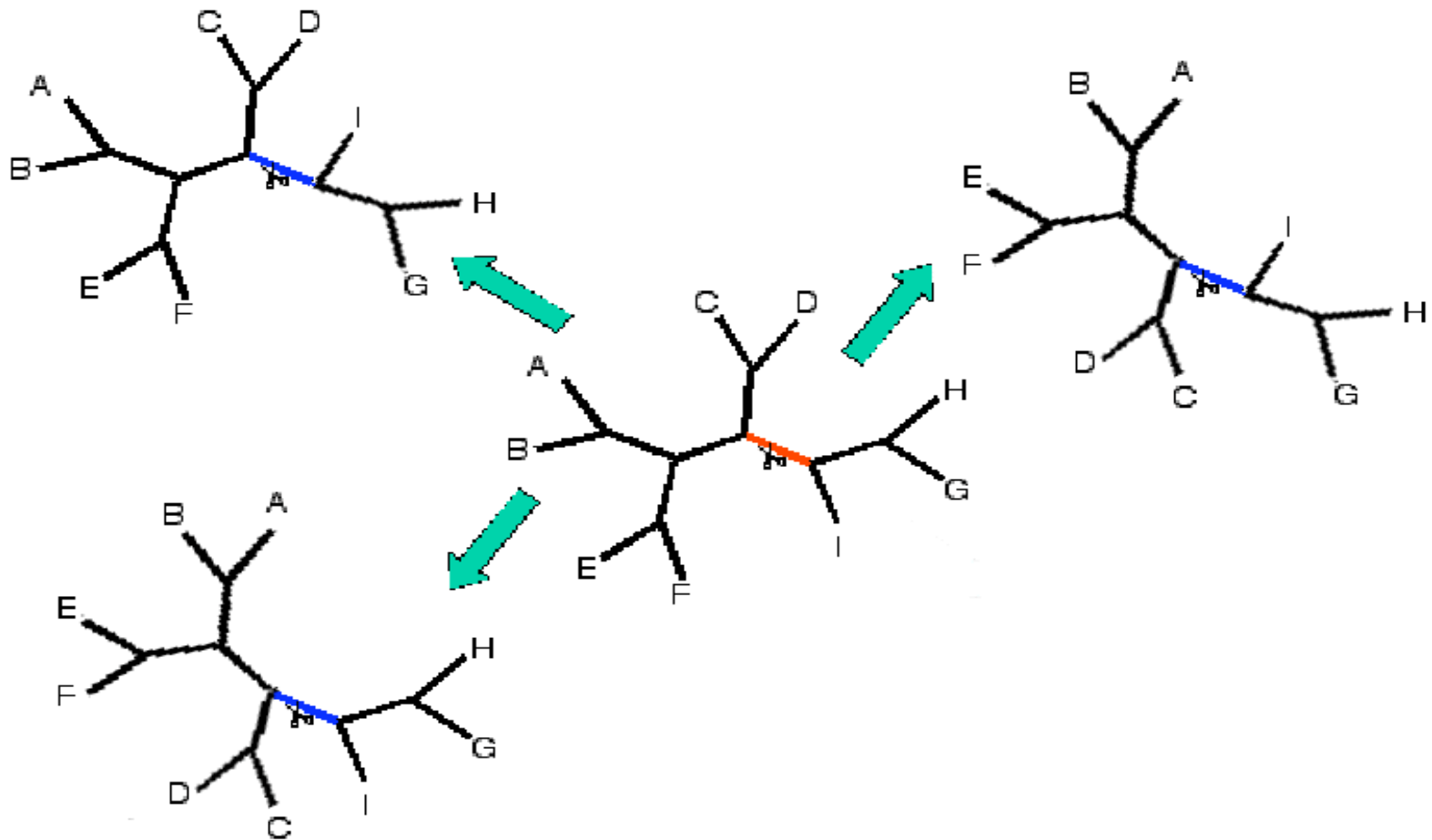
Nearest Neighbor Interchange (cont.)



Nearest Neighbor Interchange (cont.)

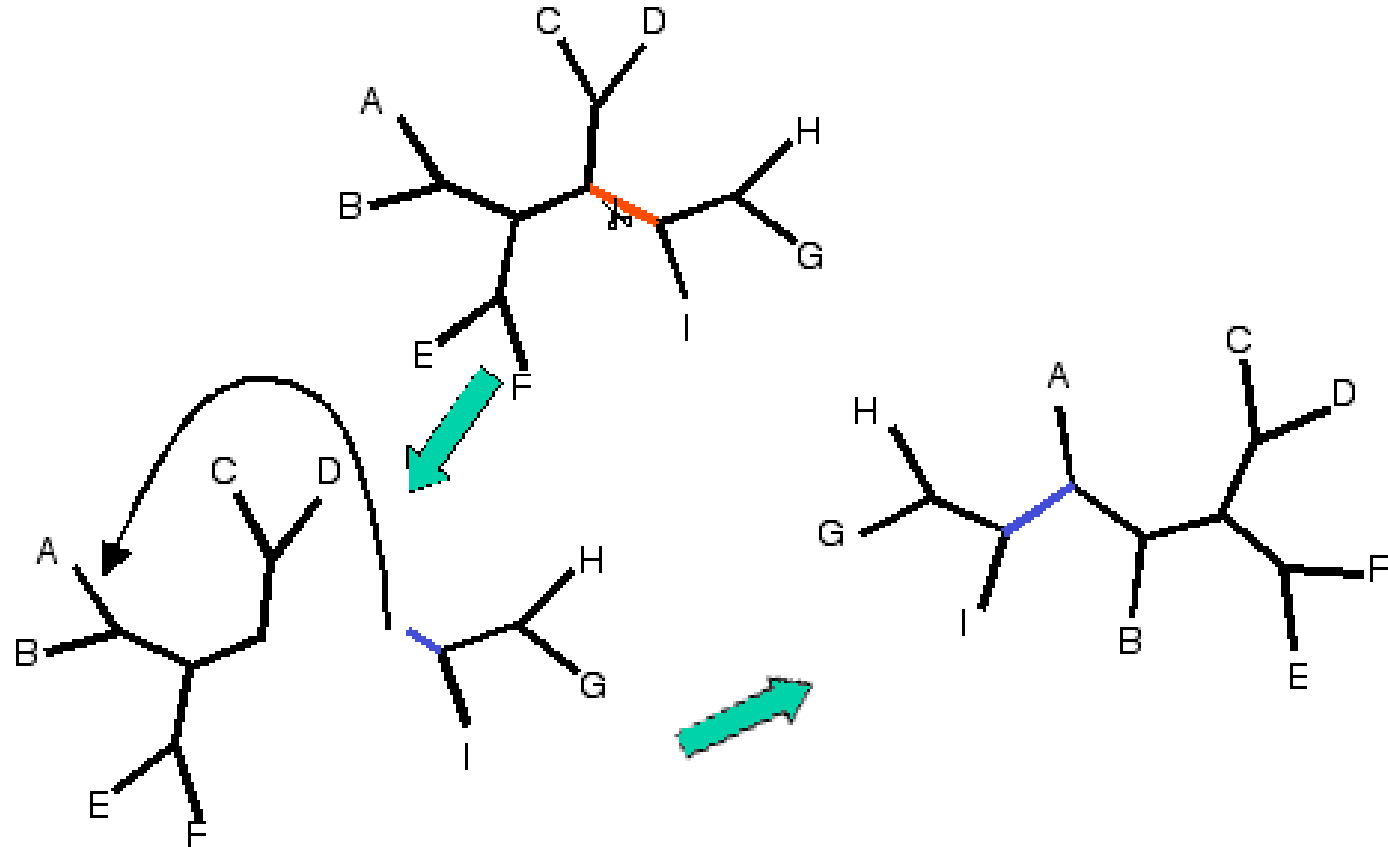
- Start with an arbitrary tree and check its neighbors
- Move to a neighbor if it provides the best improvement in parsimony score
- No way of knowing if the result is the **most** parsimonious tree
- Could be stuck in local optimum

Nearest Neighbor Interchange



Subtree Pruning and Regrafting

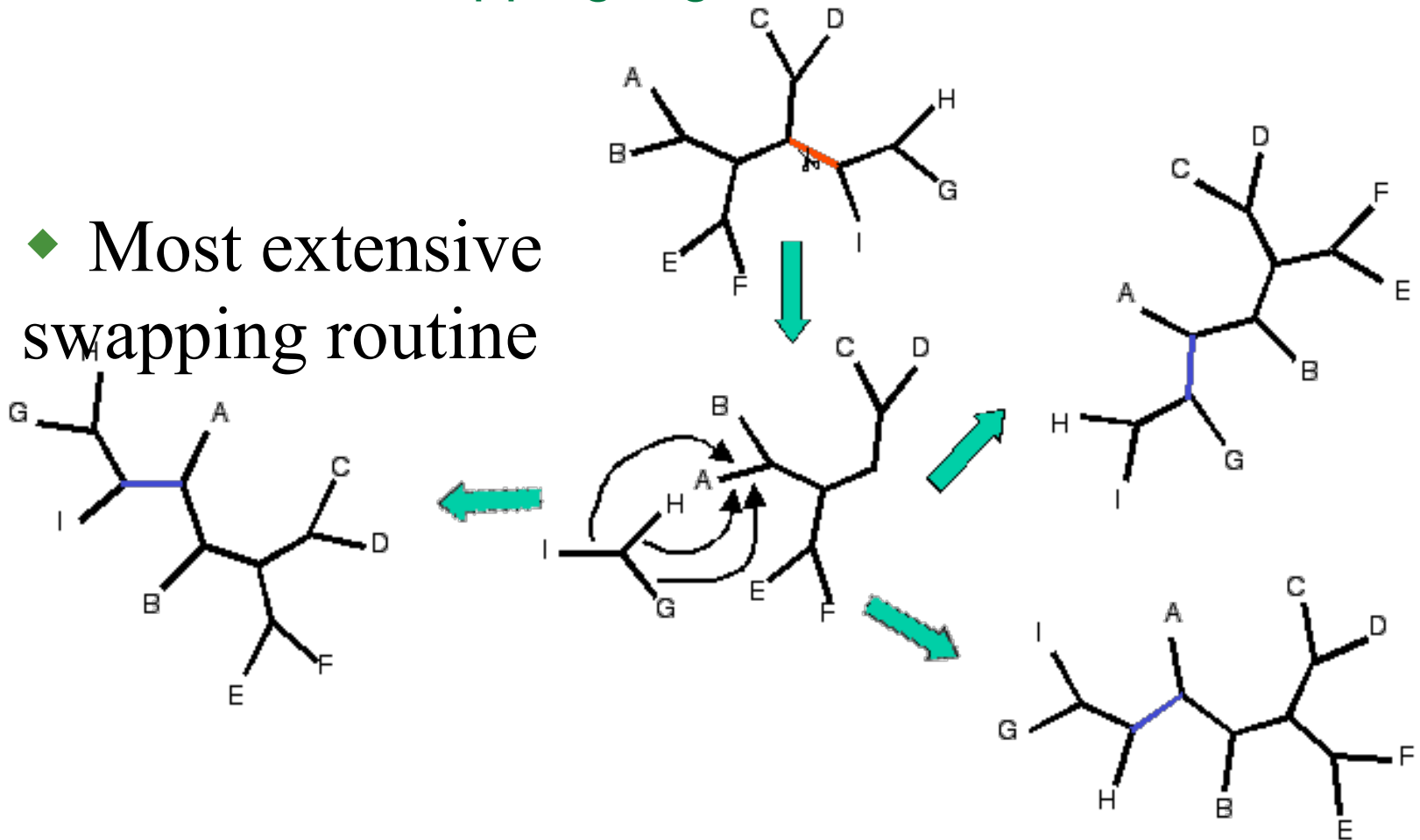
Another Branch Swapping Algorithm



Tree Bisection and Reconnection

Another Branch Swapping Algorithm

- ◆ Most extensive swapping routine

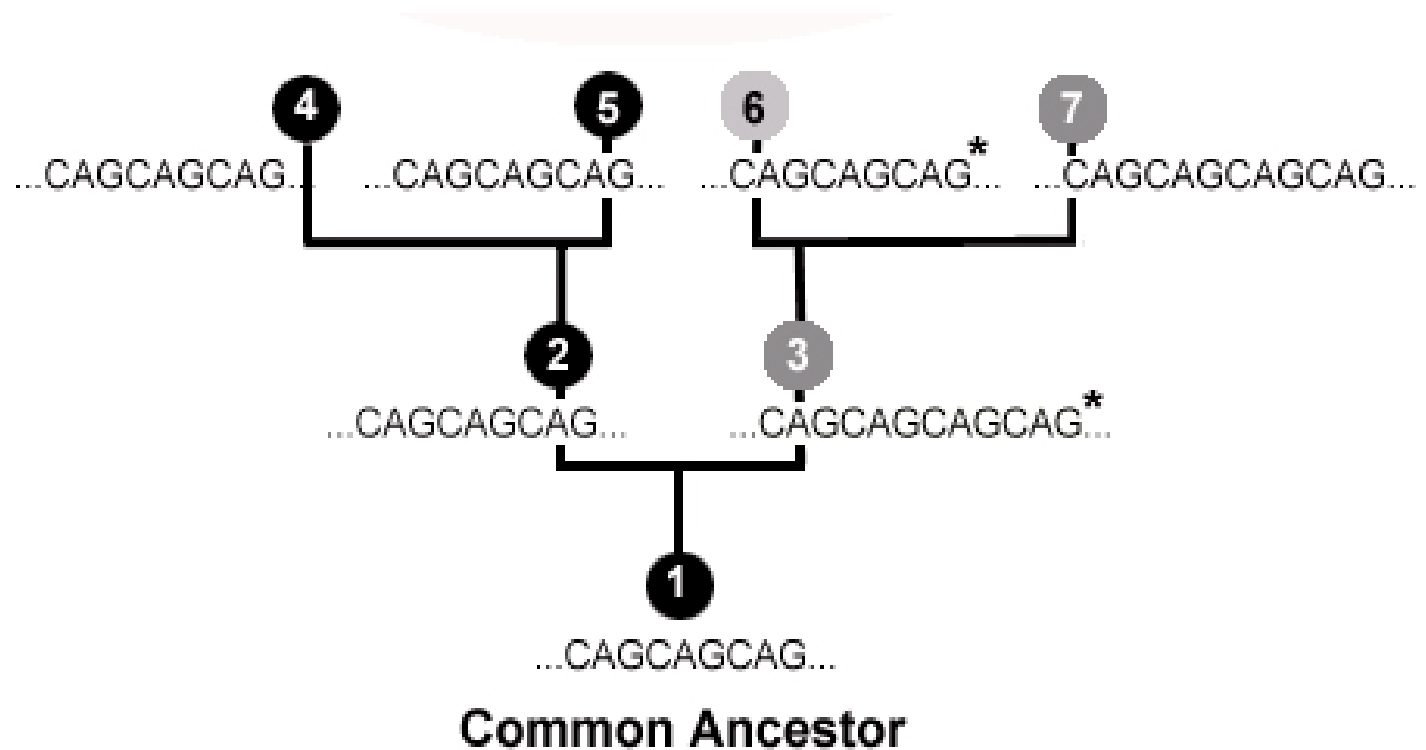


Homoplasy

- Given:
 - 1: CAGCAGCAG
 - 2: CAGCAGCAG
 - 3: CAGCAGCAGCAG
 - 4: CAGCAGCAG
 - 5: CAGCAGCAG
 - 6: CAGCAGCAG
 - 7: CAGCAGCAGCAG
- Most would group 1, 2, 4, 5, and 6 as having evolved from a common ancestor, with a single mutation leading to the presence of 3 and 7

Homoplasy

- But what if this was the real tree?

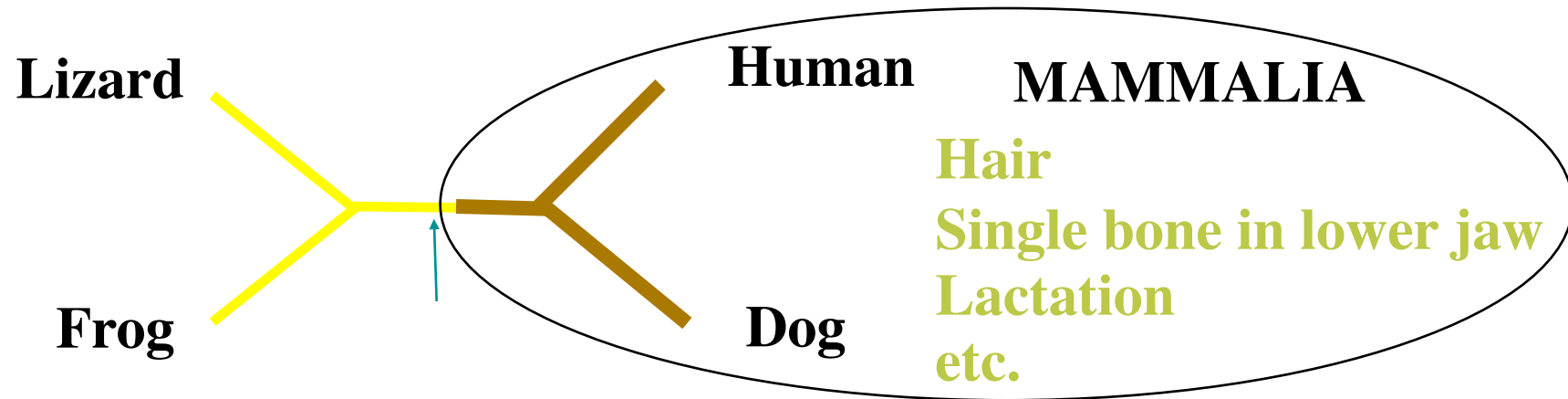


Homoplasy

- 6 evolved separately from 4 and 5, but parsimony would group 4, 5, and 6 together as having evolved from a common ancestor
- Homoplasy: Independent (or parallel) evolution of same/similar characters
- Parsimony results **minimize** homoplasy, so if homoplasy is common, parsimony may give wrong results

Contradicting Characters

- An evolutionary tree is more likely to be correct when it is supported by multiple characters, as seen below



◆ *Note: In this case, tails are homoplastic*

Problems with Parsimony

- Important to keep in mind that reliance on purely one method for phylogenetic analysis provides incomplete picture
- When different methods (parsimony, distance-based, etc.) all give same result, more likely that the result is correct

How Many Times Evolution Invented Wings?

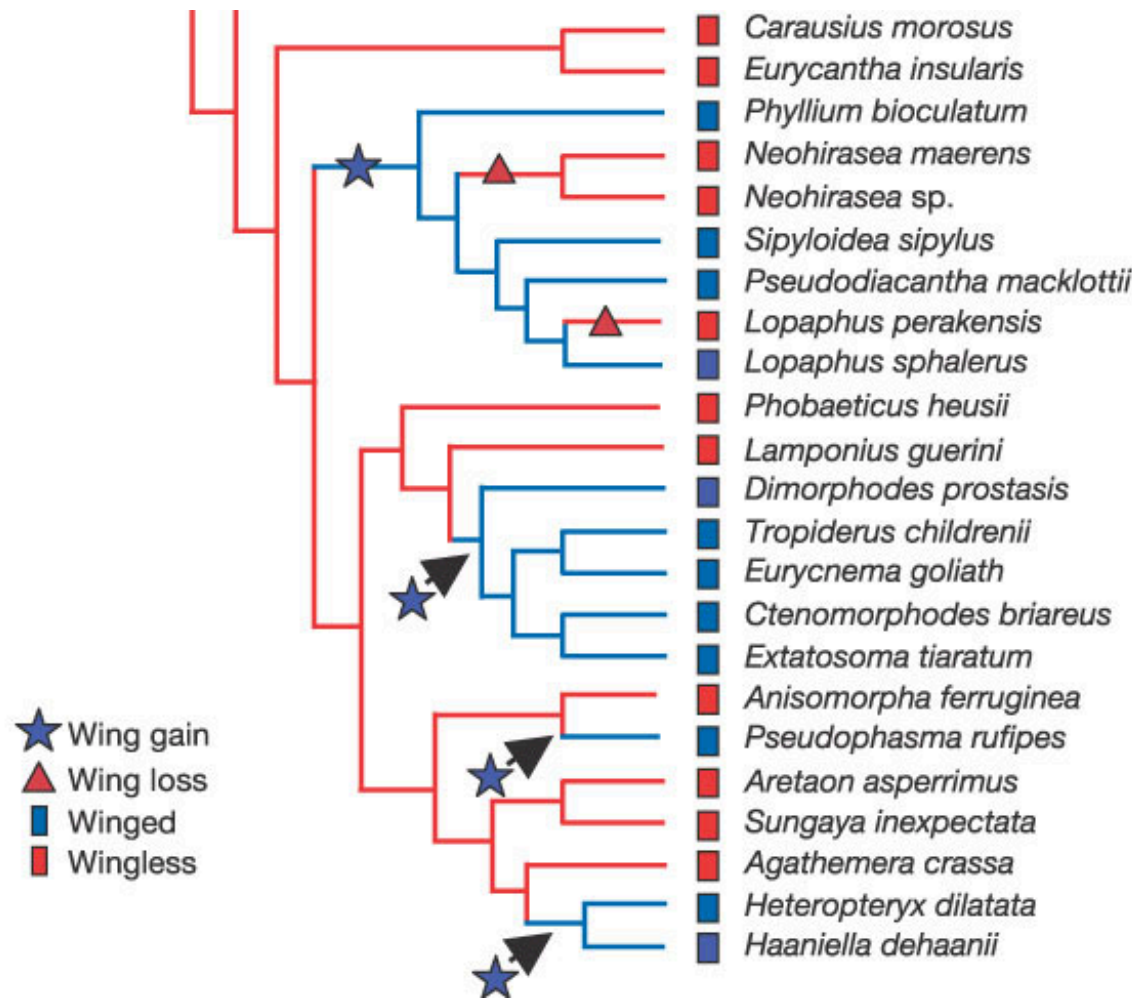
- Whiting, et. al. (2003) looked at winged and wingless stick insects



Reinventing Wings

- Previous studies had shown winged à wingless transitions
- Wingless → winged transition much more complicated (need to develop many new biochemical pathways)
- Used multiple tree reconstruction techniques, all of which required re-evolution of wings

Most Parsimonious Evolutionary Tree of Winged and Wingless Insects



- The evolutionary tree is based on **both** DNA sequences and presence/absence of wings
- Most parsimonious reconstruction gave a **wingless ancestor**

Will Wingless Insects Fly Again?

- Since the most parsimonious reconstructions all required the re-invention of wings, it is most likely that wing developmental pathways are conserved in wingless stick insects

Phylogenetic Analysis of HIV Virus

- Lafayette, Louisiana, 1994 – A woman claimed her ex-lover (who was a physician) injected her with HIV+ blood
- Records show the physician had drawn blood from an HIV+ patient that day
- But how to prove the blood from that HIV+ patient ended up in the woman?

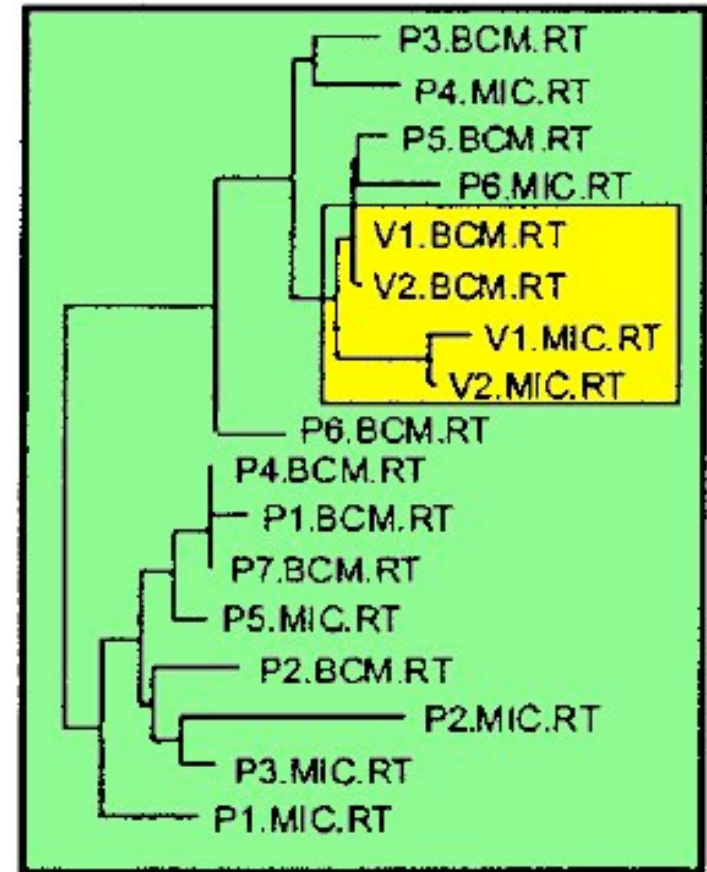
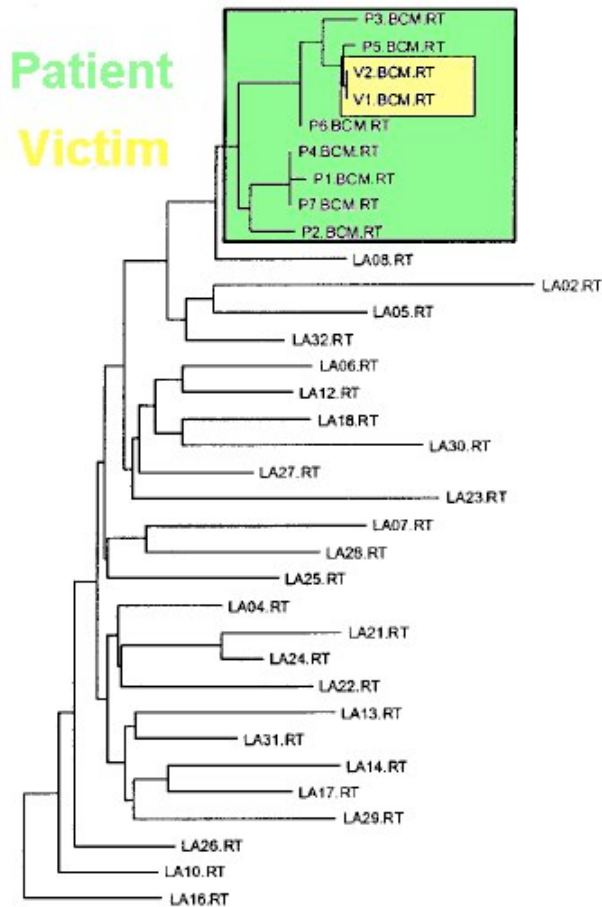
HIV Transmission

- HIV has a high mutation rate, which can be used to trace paths of transmission
- Two people who got the virus from two different people will have very different HIV sequences
- Three different tree reconstruction methods (including parsimony) were used to track changes in two genes in HIV (gp120 and RT)

HIV Transmission

- Took multiple samples from the patient, the woman, and controls (non-related HIV+ people)
- In every reconstruction, the woman's sequences were found to be evolved from the patient's sequences, indicating a close relationship between the two
- Nesting of the victim's sequences within the patient sequence indicated the direction of transmission was from patient to victim
- This was the first time phylogenetic analysis was used in a court case as evidence (Metzker, et. al., 2002)

Evolutionary Tree Leads to Conviction



Alu Repeats

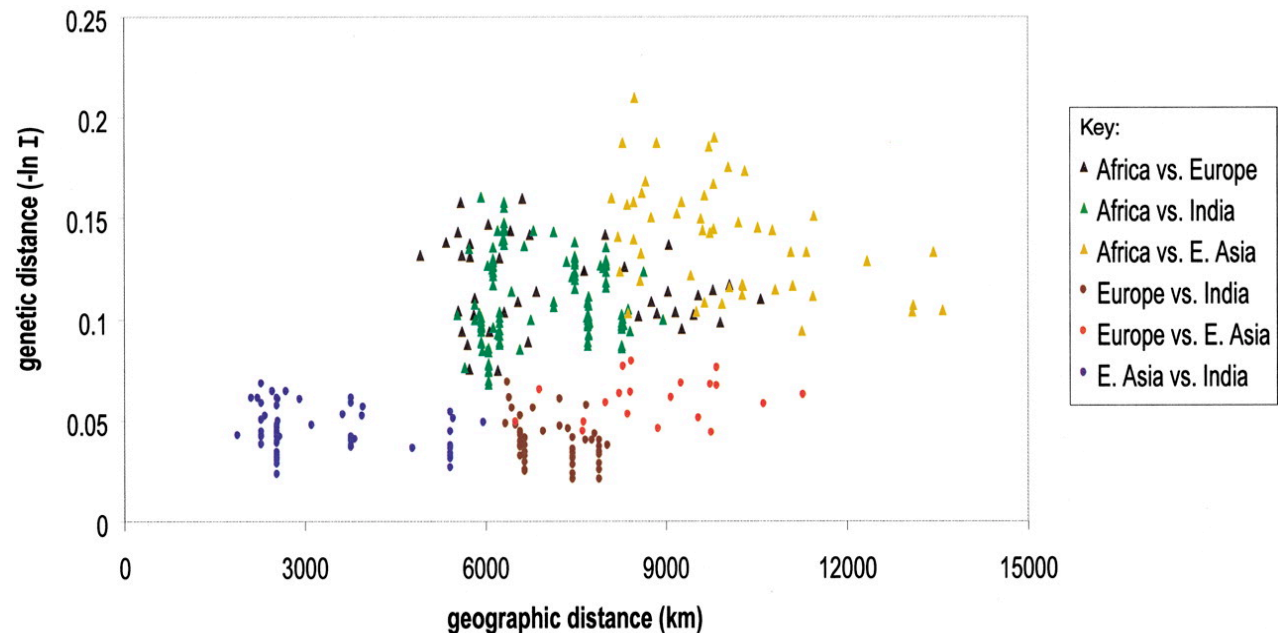
- Alu repeats are most common repeats in human genome (about 300 bp long)
- About 1 million Alu elements make up 10% of the human genome
- They are retrotransposons
 - they don't code for protein but copy themselves into RNA and then back to DNA via reverse transcriptase
 - Alu elements have been called “selfish” because their only function seems to be to make more copies of themselves

What Makes Alu Elements Important?

- Alu elements began to replicate 60 million years ago. Their evolution can be used as a fossil record of primate and human history
- Alu insertions are sometimes disruptive and can result in genetic disorders
- Alu mediated recombination can cause cancer
- Alu insertions can be used to determine genetic distances between human populations and human migratory history

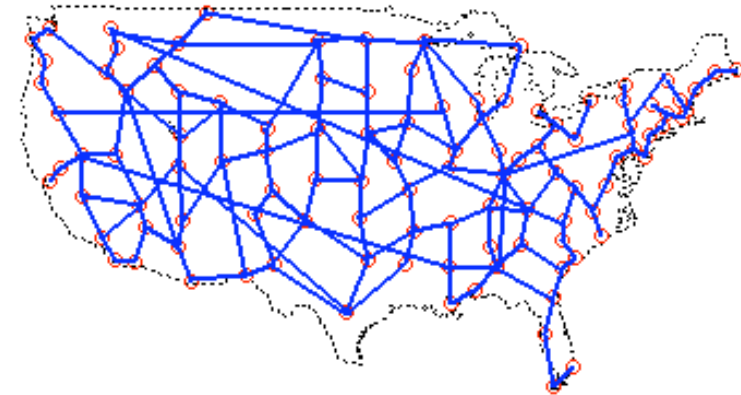
Diversity of Alu Elements

- Alu Diversity on a scale from 0 to 1
 - Africans 0.3487 origin of modern humans
 - E. Asians 0.3104
 - Europeans 0.2973
 - Indians 0.3159



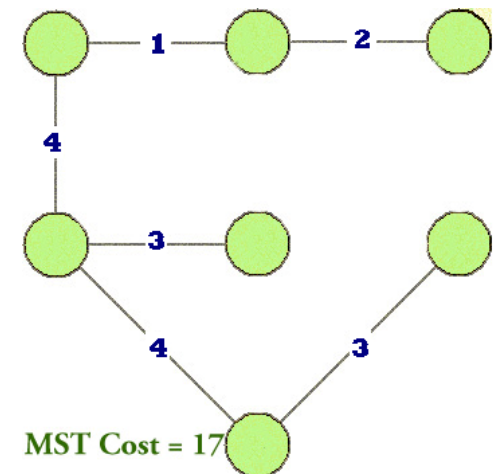
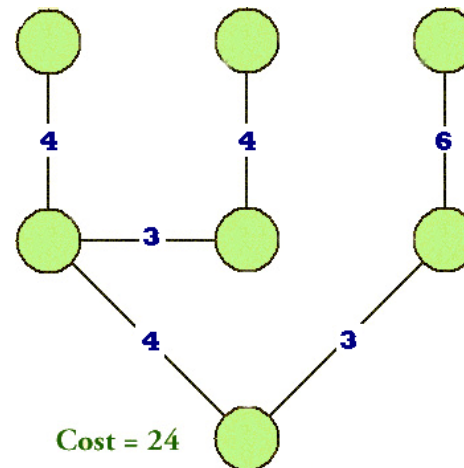
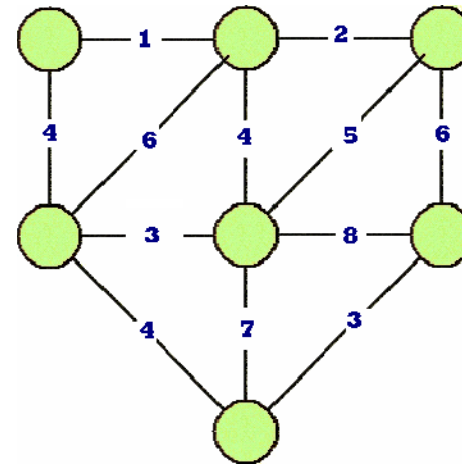
Minimum Spanning Trees

- The first algorithm for finding a MST was developed in 1926 by Otakar Borůvka. Its purpose was to minimize the cost of electrical coverage in Bohemia.
- The Problem
 - Connect all of the cities but use the least amount of electrical wire possible. This reduces the cost.
- ***We will see how building a MST can be used to study evolution of Alu repeats***



What is a Minimum Spanning Tree?

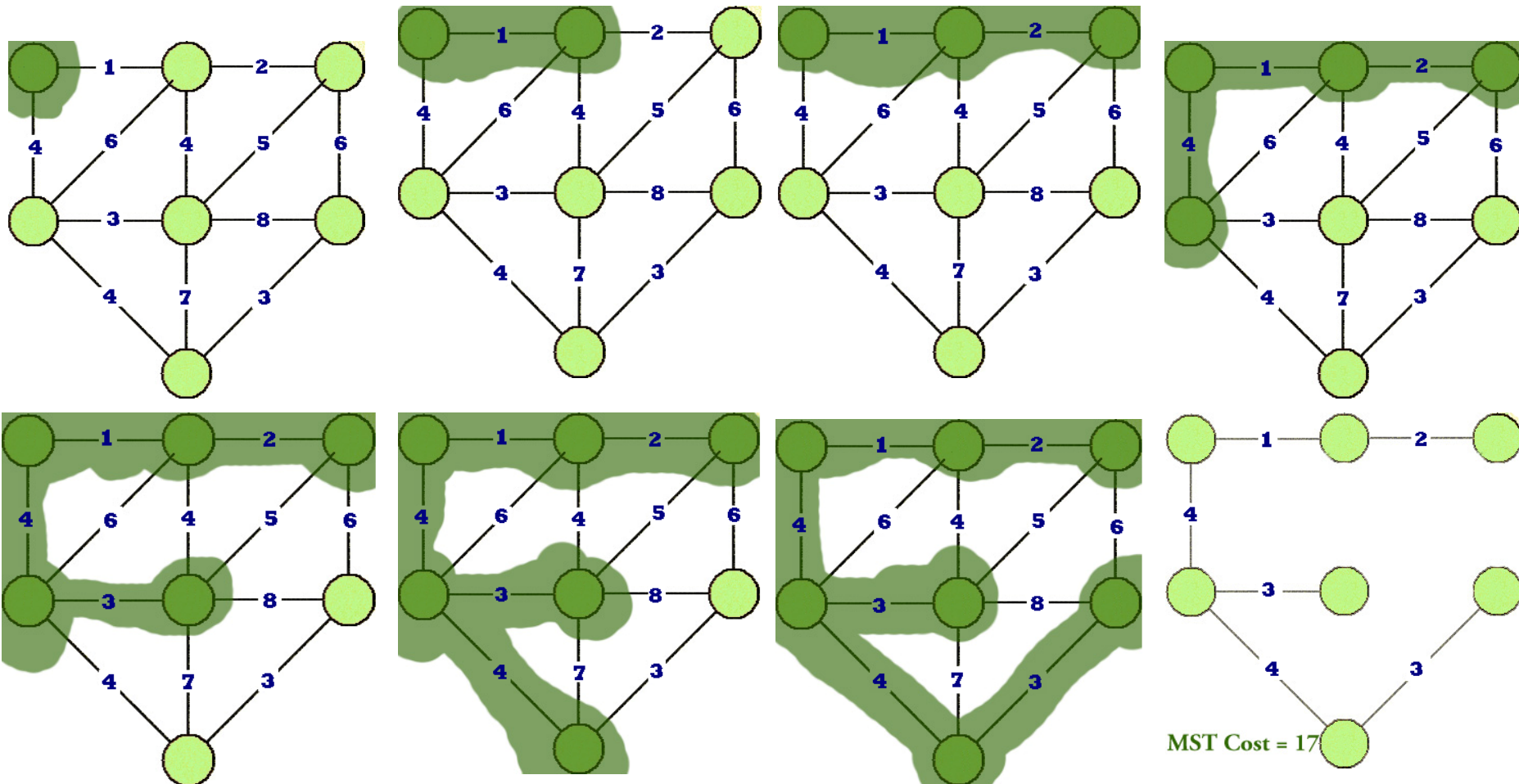
- A Minimum Spanning Tree of a graph
 - connect all the vertices in the graph and
 - minimizes the sum of edges in the tree



How can we find a MST?

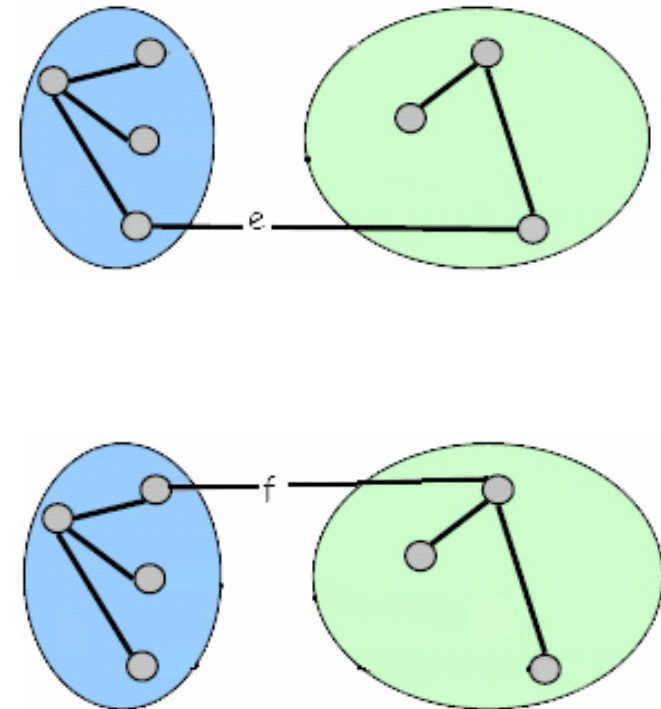
- Prim algorithm (greedy)
 - Start from a tree T with a single vertex
 - Add the shortest edge connecting a vertex in T to a vertex not in T , growing the tree T
 - This is repeated until every vertex is in T
- Prim algorithm can be implemented in $O(m \log m)$ time (m is the number of edges).

Prim's Algorithm Example



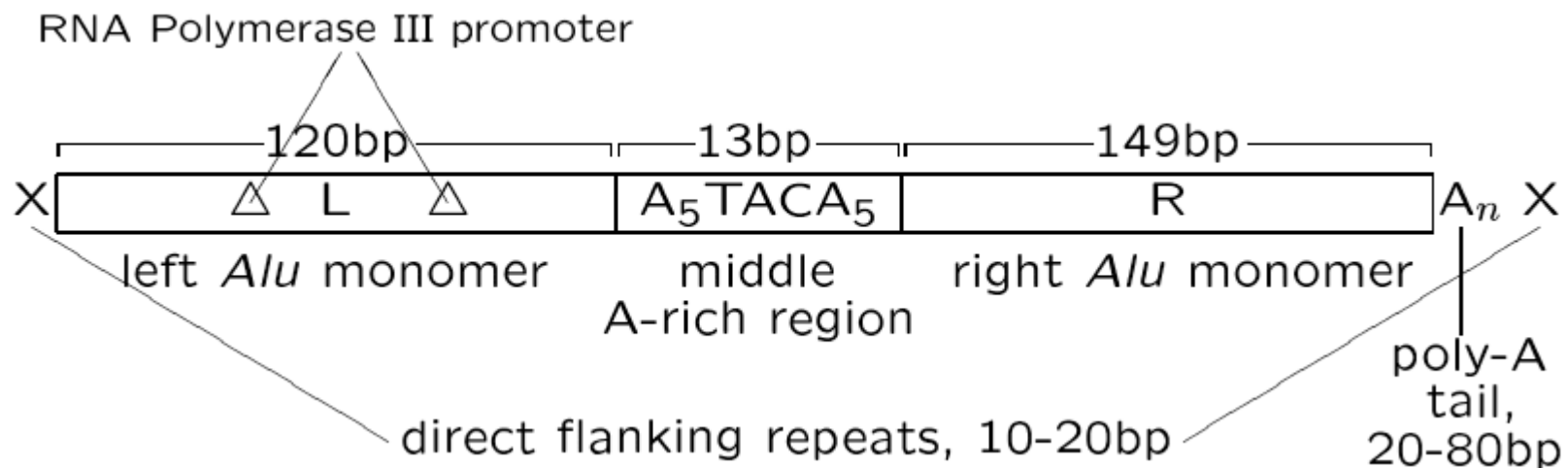
Why Prim Algorithm Constructs Minimum Spanning Tree?

- Proof:
 - This proof applies to a graph with distinct edges
 - Let e be any edge that Prim algorithm chose to connect two sets of nodes. Suppose that Prim's algorithm is flawed and it is cheaper to connect the two sets of nodes via some other edge f
 - Notice that since Prim algorithm selected edge e we know that $\text{cost}(e) < \text{cost}(f)$
 - By connecting the two sets via edge f , the cost of connecting the two vertices has gone up by exactly $\text{cost}(f) - \text{cost}(e)$
 - The contradiction is that edge e does not belong in the MST yet the MST can't be formed without using edge e



An Alu Element

- SINEs are flanked by short direct repeat sequences and are transcribed by RNA Polymerase III



Alu Subfamilies

We illustrate *Alu* subfamilies with a 40bp sample segment of AluJb, AluSx, AluY and AluYa5 subfamily consensus sequences:

```
AluJb    ...G.....A.....-...
AluSx    TGGCCAACATGGTGAAACCCCGTCTCTACTAAAAATACAAAAA-TT
AluY     ....T....C.....A..
AluYa5   C...T..A.C.....A..
```

Early analyses identified 4-6 *Alu* subfamilies.

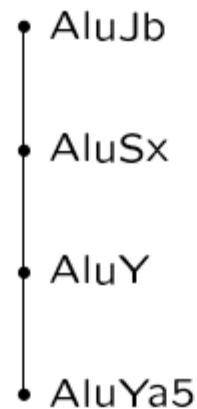
Willard *et al.*, 1987; Britten *et al.*, 1988; Deininger and Slagel, 1988;
Jurka and Smith, 1988; Quentin, 1988; Matera *et al.*, 1990;
Batzner and Deininger, 1991; Jurka and Milosavljevic, 1991; Shen *et al.*, 1991

What do *Alu* subfamilies tell us about *Alu* evolution?

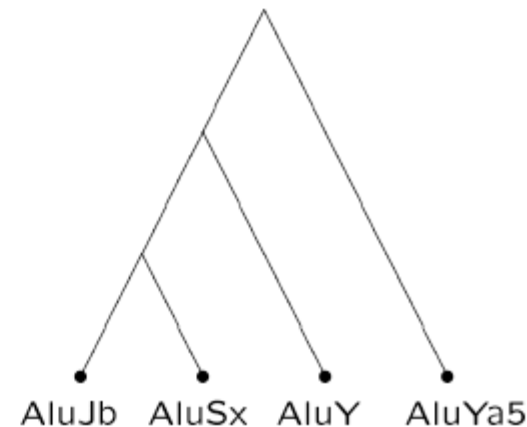
The Biological Story: Alu Evolution

What do *Alu* subfamilies tell us about *Alu* evolution?

AluJb	...G.....A.....-...
AluSx	TGGCCAACATGGTGAAACCCCGTCTCTACTAAAAATACAAAAA-TT
AluYT....C.....A..
AluYa5	C...T..A.C.....A..



EVOLUTIONARY TREE

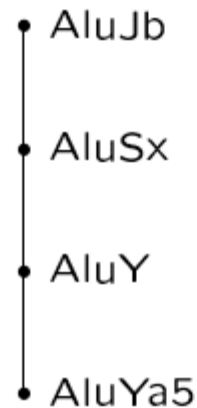


PHYLOGENETIC TREE

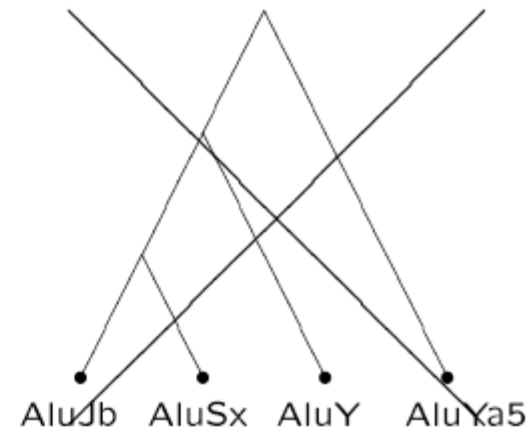
Alu Evolution

What do *Alu* subfamilies tell us about *Alu* evolution?

AluJb	...G.....A.....-..
AluSx	TGGCCAACATGGTGAAACCCCGTCTCTACTAAAAATACAAAAA-TT
AluYT....C.....A..
AluYa5	C...T..A.C.....A..



EVOLUTIONARY TREE



PHYLOGENETIC TREE
(see Cordaux *et al.*, 2004)

Alu Evolution: The Master Alu Theory

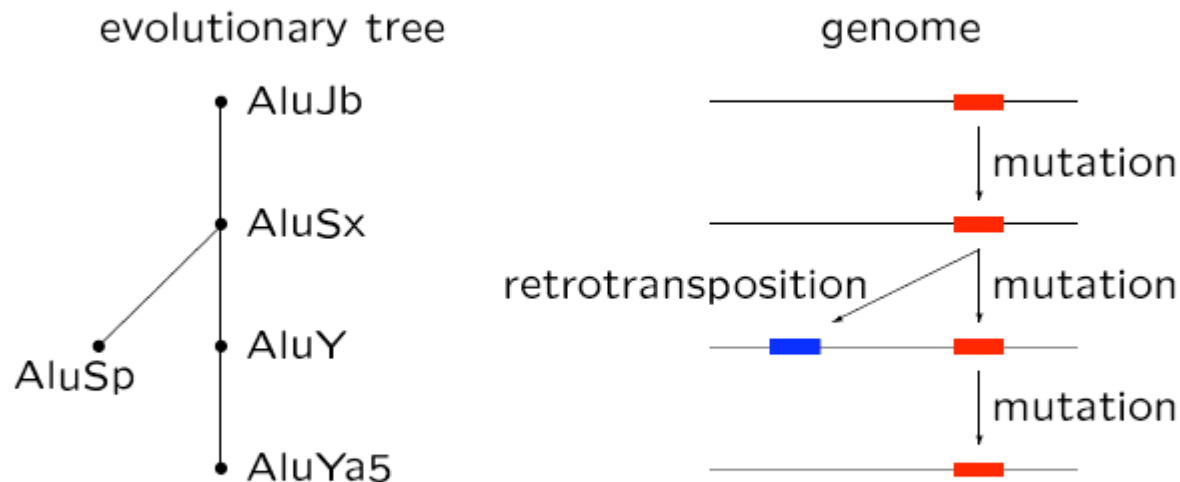


Shen *et al.*, 1991 conjectured that all *Alu* repeat elements have retroposed from “a single master gene”.

Conjecture: 4 subfamilies \longleftrightarrow linear evolution of 1 master gene.

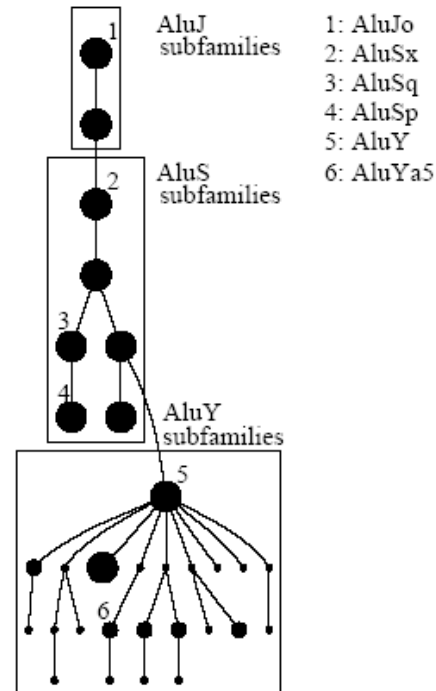
Alu Evolution: Alu Master Theory Proven Wrong

Jurka and Milosavljevic, 1991 identified additional subfamilies which do not fit the linear pattern of evolution:



The AluSp and AluY subfamily lineages must have been produced by distinct master genes – this disproves the master *Alu* theory!

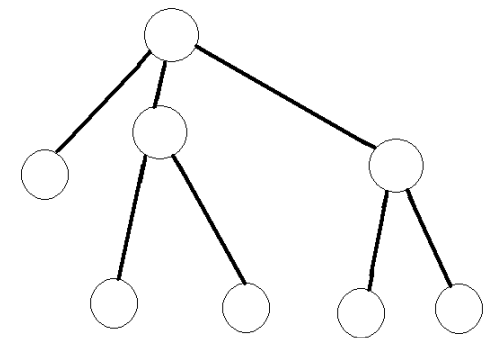
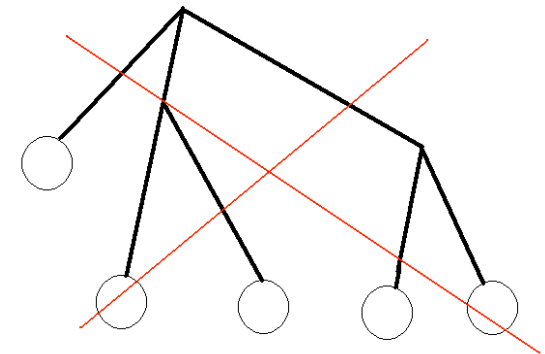
Minimum Spanning Tree As An Evolutionary Tree



The evolutionary tree of the 31 Repbase Update subfamilies, defined as their Minimum Spanning Tree (Kruskal 1956).
14 leaves in this tree \Rightarrow at least 14 *Alu* source elements.

Alu Evolution: Minimum Spanning Tree vs. Phylogenetic Tree

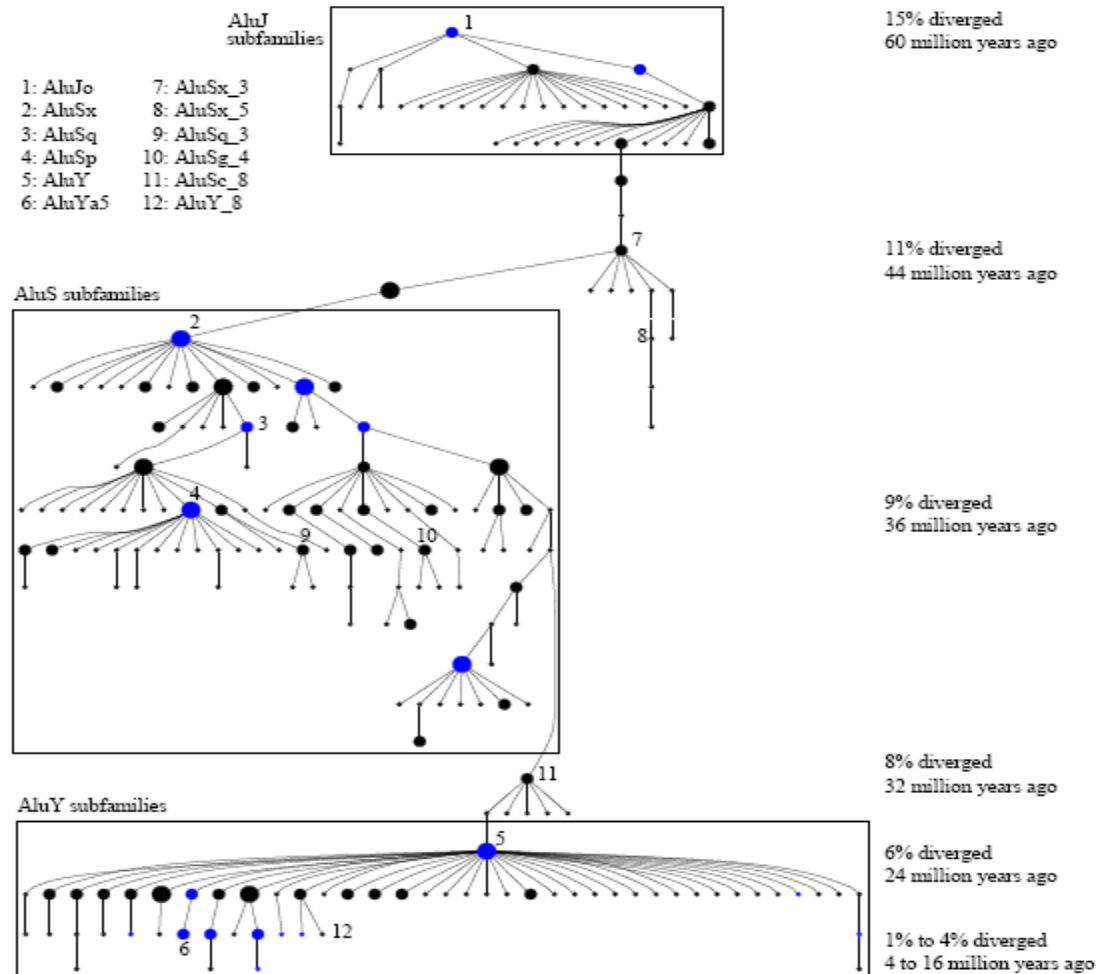
- A timeline of Alu subfamily evolution would give useful information
 - Problem - building a traditional phylogenetic tree with Alu subfamilies will not describe Alu evolution accurately
- Why can't a meaningful typical phylogenetic tree of Alu subfamilies be constructed?
 - When constructing a typical phylogenetic tree, the input is made up of leaf nodes, but no internal nodes
 - Alu subfamilies may be either internal or external nodes of the evolutionary tree because Alu subfamilies that created new Alu subfamilies are themselves still present in the genome. Traditional phylogenetic tree reconstruction methods are not applicable since they don't allow for the inclusion of such internal nodes



Constructing MST for Alu Evolution

- Building an evolutionary tree using an MST will allow for the inclusion of internal nodes
 - Define the length between two subfamilies as the Hamming distance between their sequences
 - Root the subfamily with highest average divergence from its consensus sequence (the oldest subfamily), as the root
 - It takes ~4 million years for 1% of sequence divergence between subfamilies to emerge, this allows for the creation of a timeline of Alu evolution to be created
- Why an MST is useful as an evolutionary tree in this case
 - The less the Hamming distance (edge weight) between two subfamilies, the more likely that they are directly related
 - An MST represents a way for Alu subfamilies to have evolved minimizing the sum of all the edge weights (total Hamming distance between all Alu subfamilies) which makes it the most parsimonious way and thus the most likely way for the evolution of the subfamilies to have occurred.

MST As An Evolutionary Tree



Sources

- <http://www.math.tau.ac.il/~rshamir/ge/02/scribes/lec01.pdf>
- <http://bioinformatics.oupjournals.org/cgi/screenpdf/20/3/340.pdf>
- http://www.absoluteastronomy.com/encyclopedia/M/Mi/Minimum_spanning_tree.htm
- Serafim Batzoglou (UPGMA slides) <http://www.stanford.edu/class/cs262/Slides>
- Watkins, W.S., Rogers A.R., Ostler C.T., Wooding, S., Bamshad M. J., Brassington A.E., Carroll M.L., Nguyen S.V., Walker J.A., Prasas, R., Reddy P.G., Das P.K., Batzer M.A., Jorde, L.B.: Genetic Variation Among World Populations: **Inferences From 100 *Alu* Insertion Polymorphisms**