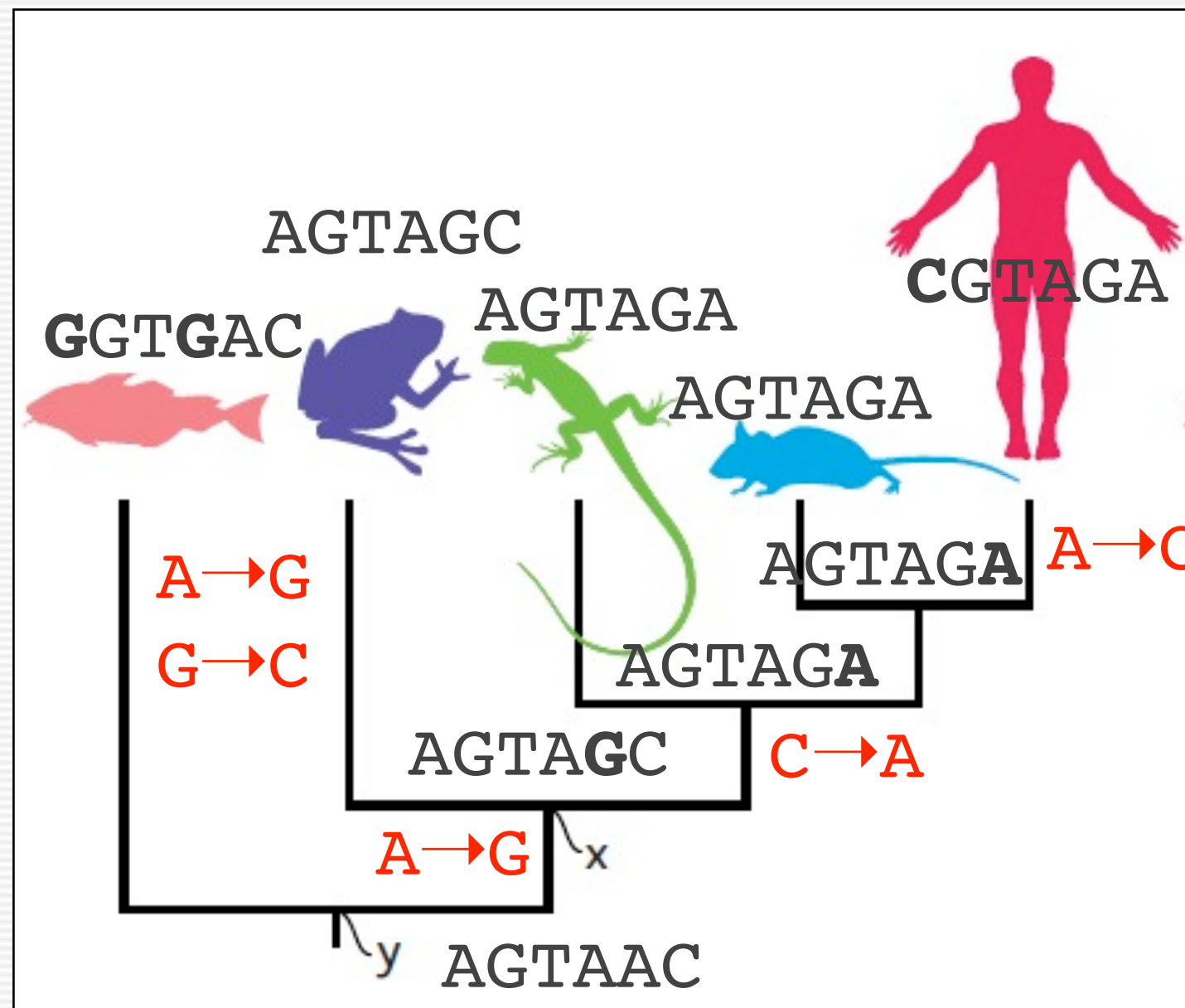


MOLECULAR EVOLUTION AND PHYLOGENETICS

- If we could **observe** evolution: speciation, mutation, natural selection and fixation, we might see something like this:



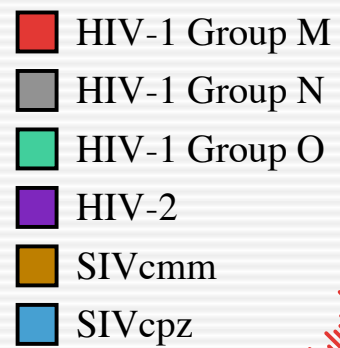
[HTTP://WWW.SCIENCEMAG.ORG/CGI/REPRINT/310/5750/979.PDF](http://www.sciencemag.org/cgi/reprint/310/5750/979.pdf)

EVOLUTIONARY INFERENCE

- In practice all we see is this:

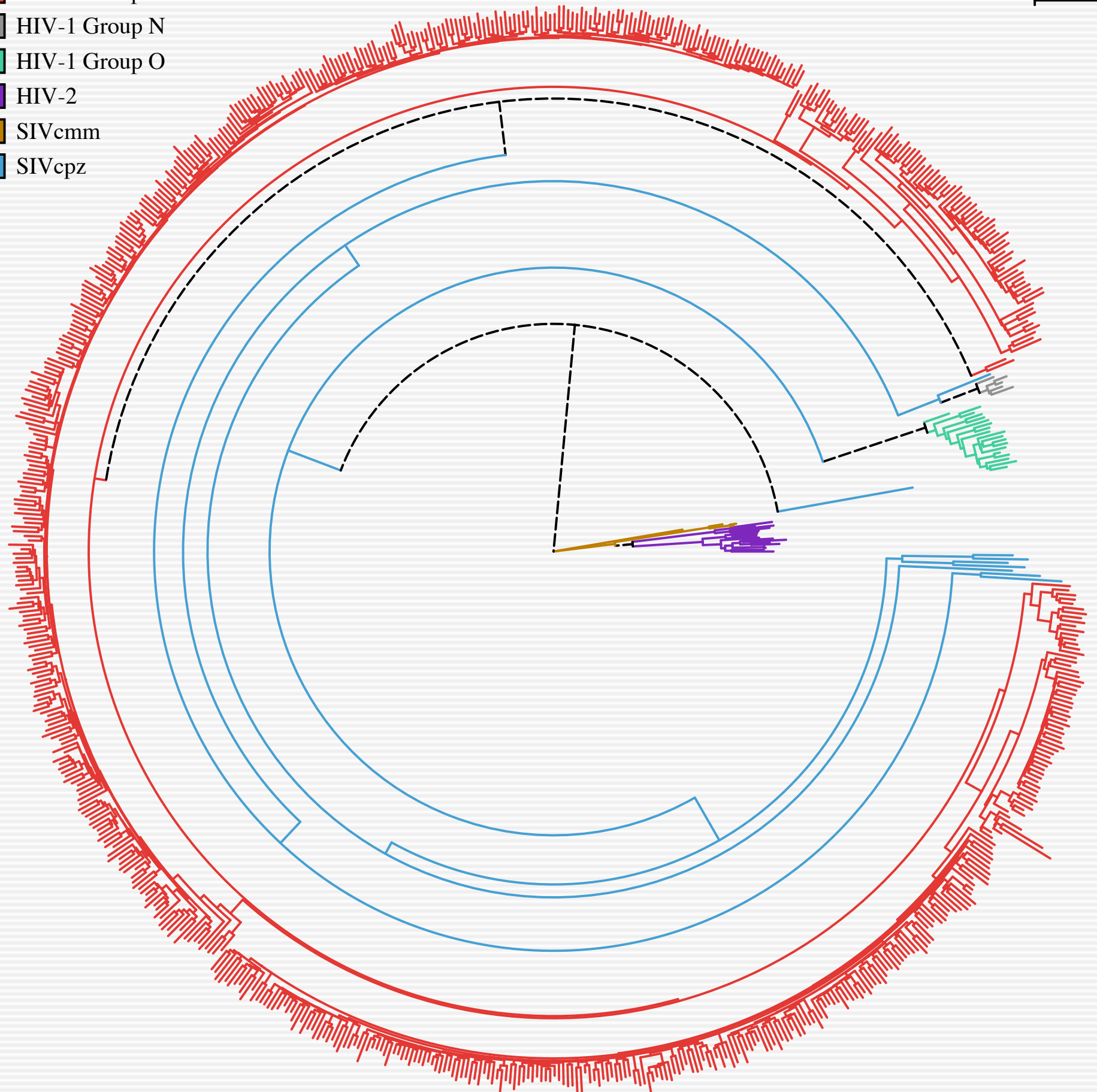
FISH	GGTGAC
FROG	AGTAGC
LIZARD	AGTAGA
MOUSE	AGTAGA
HUMAN	CGTAGA

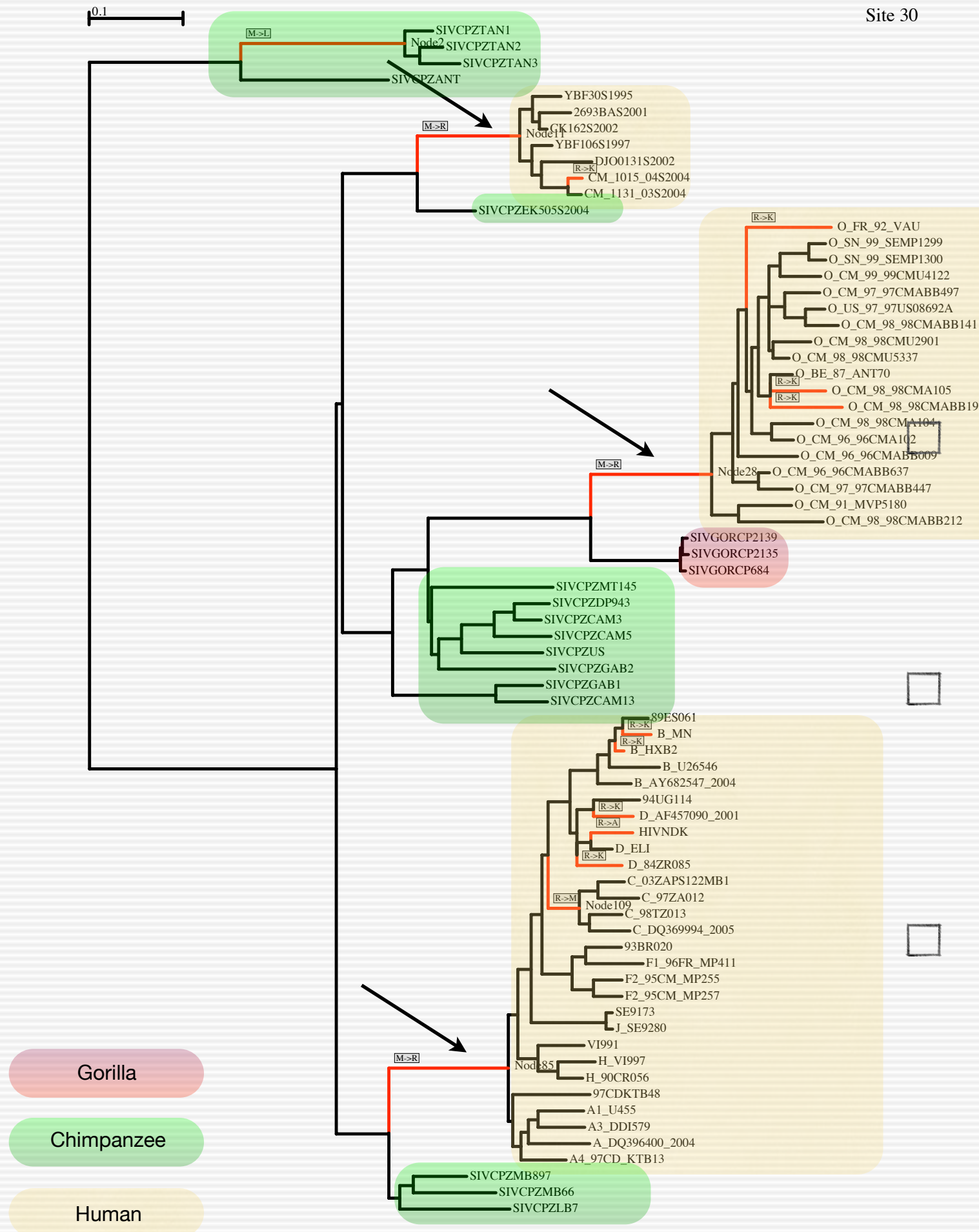
- We wish to reconstruct the evolutionary history of the sample, e.g.
 - The species tree, the ancestral sequences and mutations



0.1

- ☐ Three SIVcpz/SIVgor to human transmissions seeding three major clades of HIV-1
- ☐ At least 8 SIVsm to human transmissions, seeding HIV-2
- ☐ Circulating diversity in HIV-1 is immense: from 5-40% depending on gene



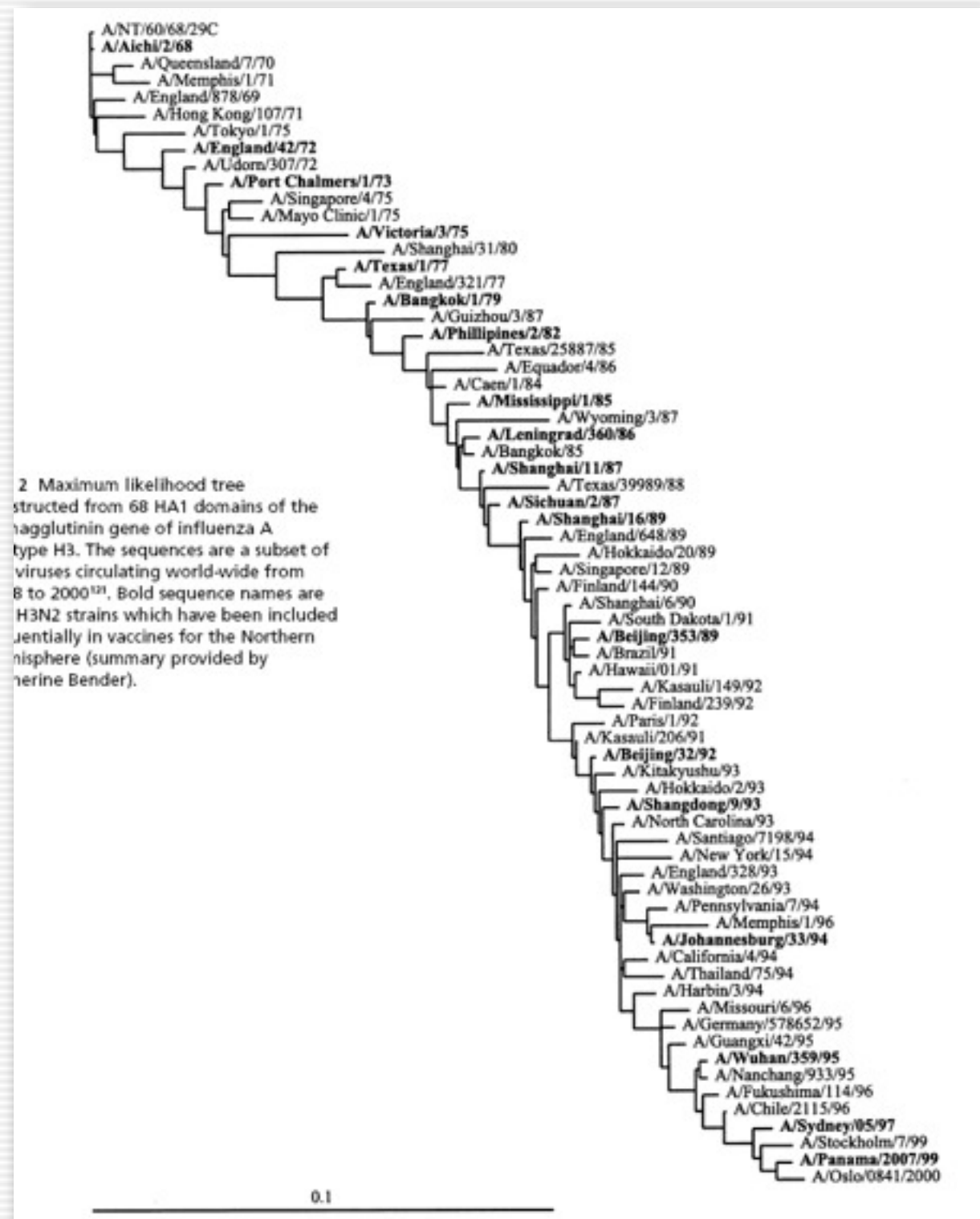


There were 3 independent introductions of HIV-1 into human hosts from SIVcpz/gor.

- ☐ Was there anything in common in what happened to the virus during the zoonosis?
- ☐ If the virus was forced to adapt to human hosts the same way, maybe we can use this information to fight it.

TREE SHAPES CAN BE INFORMATIVE

- Phylogenetic tree of Influenza A virus H₃N₂ serotype hemagglutinin sequences from 1968 to 2000
- Displays the classic ‘ladder’ shape attributable to antigenic drift



RATES OF EVOLUTION CAN BE USED TO INFER IMPORTANT DATES IN THE PAST: E.G. DATING THE ORIGIN OF HIV

An African HIV-1 sequence from 1959 and implications for the origin of the epidemic

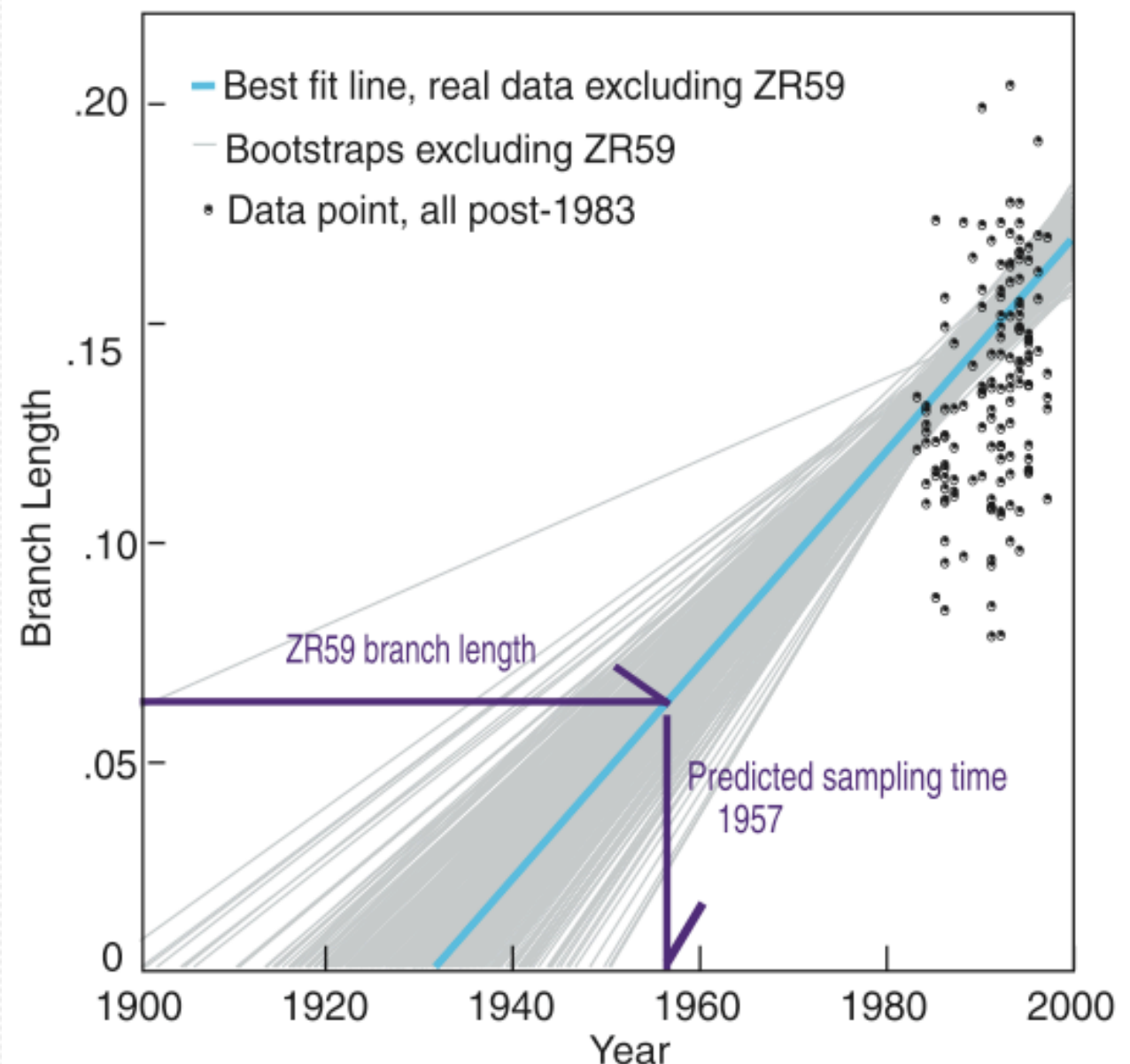
Tuofu Zhu^{*†}, Bette T. Korber^{‡§}, Andre J. Nahmias^{||},
Edward Hooper[¶], Paul M. Sharp[#] & David D. Ho^{*}

Timing the Ancestor of the HIV-1 Pandemic Strains

B. Korber,^{1,2*†} M. Muldoon,^{2,3} J. Theiler,¹ F. Gao,⁴ R. Gupta,¹
A. Lapedes,^{1,2} B. H. Hahn,⁴ S. Wolinsky,⁵ T. Bhattacharya^{1†}

B

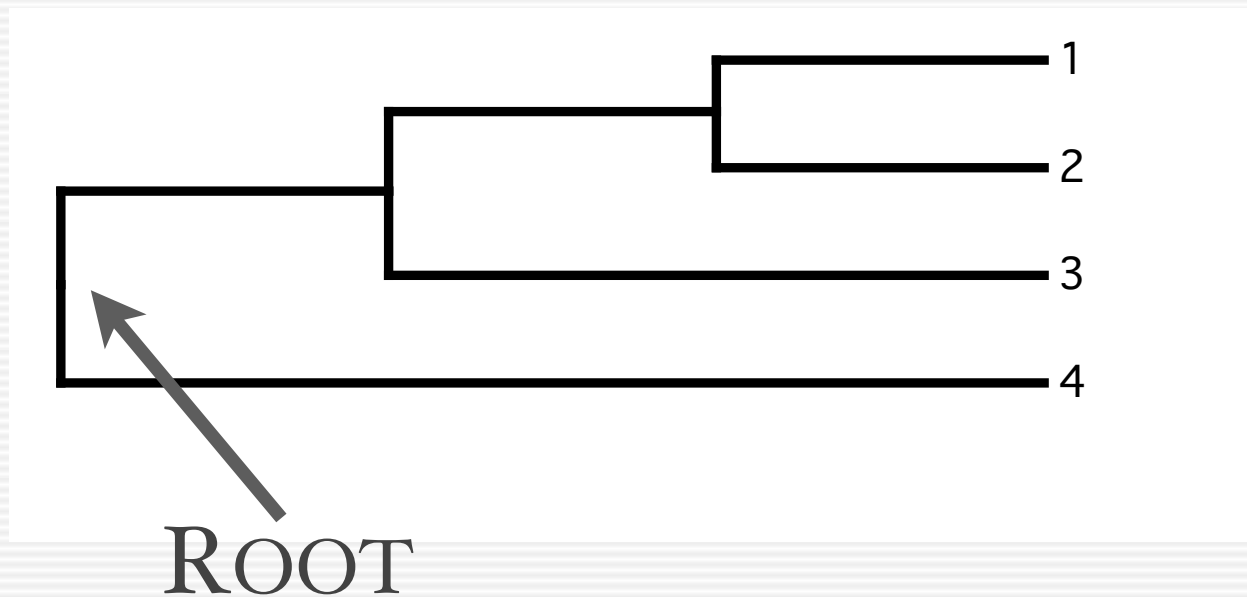
Test case: the ZR1959 sequence



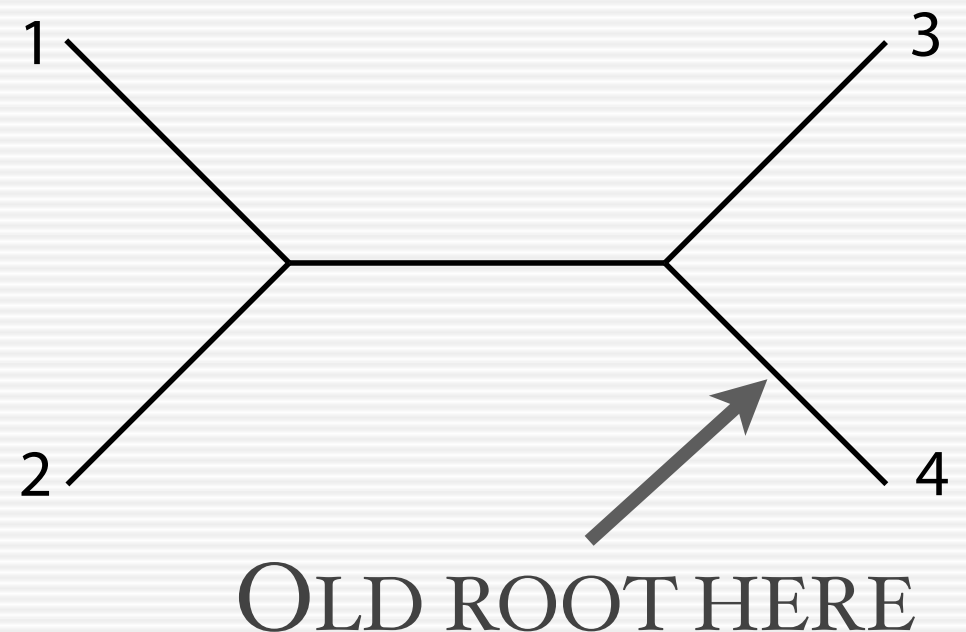
TREE TYPES

- A tree is an acyclic connected graph
- A rooted tree has a node designated as the root. This implicitly assigns a direction to the tree.
- An unrooted tree does not have a root – this is necessary to assume in some cases because directionality of evolution is not always known.
- In an unrooted binary (bifurcating) tree all interior nodes have degree 3.
- In a rooted tree, one node has degree 2 (the root)

ROOTED

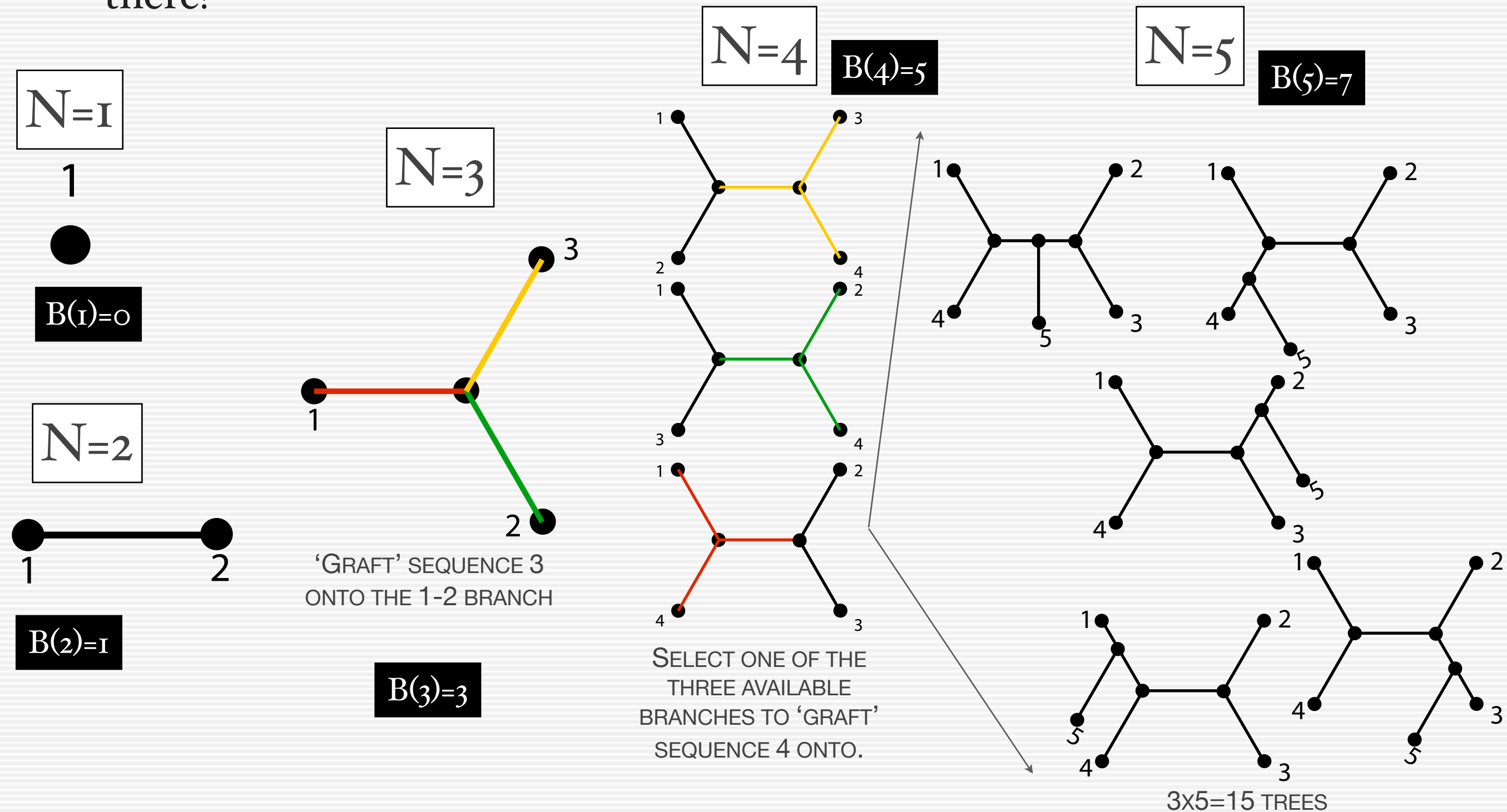


UNROOTED



COUNTING TREES AND BRANCHES

- How many branches $B(N)$ does an **unrooted tree** on N leaves have and how many different **unrooted labeled** trees, $T_u(N)$, with N leaves are there?



COUNTING TREES

BRANCH COUNT

$$B(1) = 0$$

$$B(2) = 1$$

$$B(N) = B(N - 1) + 2$$

...

$$B(N) = 2N - 3, N \geq 1$$

TREE COUNT

$$T_u(1) = 1$$

$$T_u(2) = 1$$

$$T_u(N) = B(N - 1)T_u(N - 1)$$

...

$$T_u(N) = \prod_{k=1}^{N-1} (2k - 3) =$$

$$(2N - 5) \times (2N - 3) \dots \times 3 \times 1 =$$

$$(2N - 5)!!$$

$$N \geq 3$$

THERE ARE COMBINATORIALLY MANY TREES

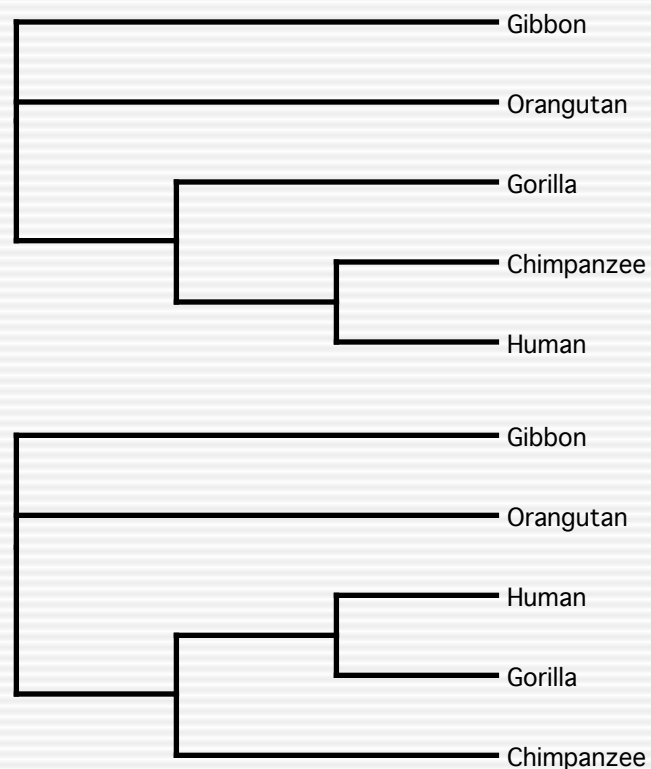
N	$T_u(N)$
3	1
4	3
5	15
6	105
7	945
8	10395
9	135135
10	2027025

N	$T_u(N)$
20	2.22E+20
50	2.84E+74
100	1.70E+182

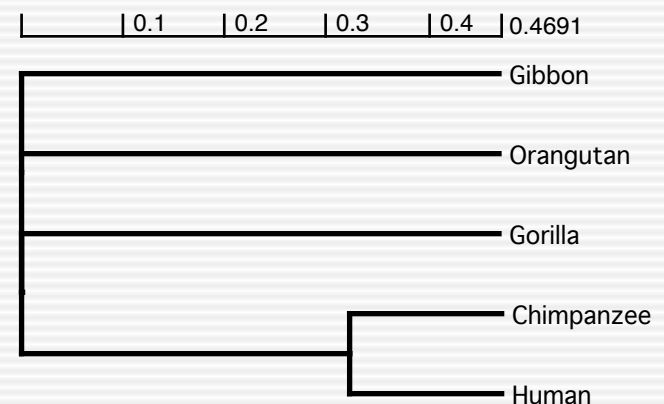
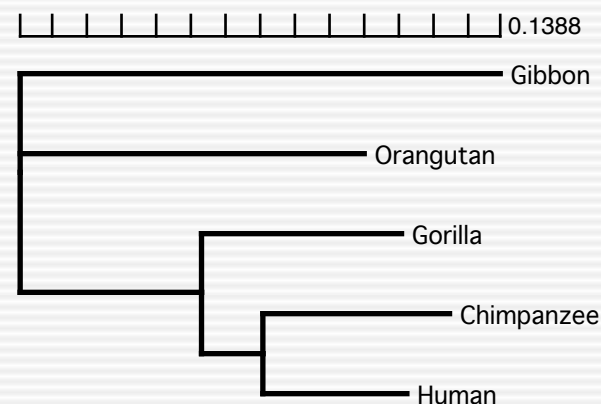
TOPOLOGY VS TREE

- Topology defines the structure of the tree (unweighted edges)
- Topology combined with branch lengths constitutes a phylogenetic tree

DIFFERENT TOPOLOGIES



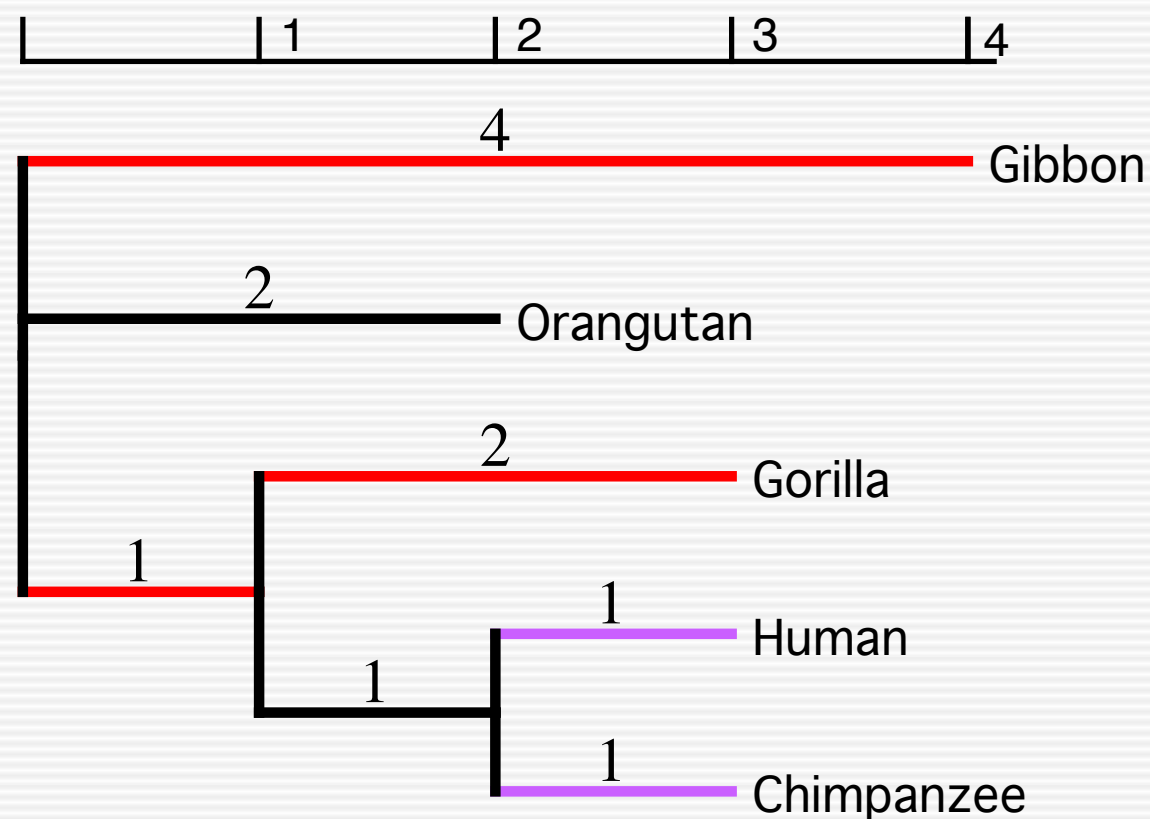
SAME TOPOLOGIES, DIFFERENT TREES



AN ULTRAMETRIC TREE

DISTANCES IN A TREE

- To measure distances between two leaves in a tree, we compute the length of the (unique) path connecting them.



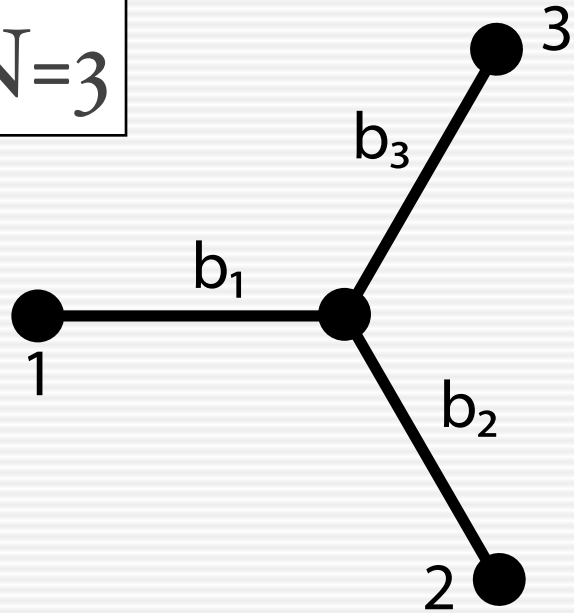
- $d(\text{Gibbon} \text{ -- } \text{Gorilla}) = 4 + 1 + 2 = 7$

- $d(\text{Human} \text{ -- } \text{Chimp}) = 1 + 1 = 2$

FITTING A DISTANCE MATRIX TO A TREE

- It is relatively easy to compute **pairwise** distances between two sequences/organisms. These distances can be derived from nucleotide sequences, morphology, allele frequencies, copy numbers etc.
- For N sequences, we start with an $N \times N$ (symmetric) pairwise distance matrix $\mathbf{D}_{ij}$
- Given a topology on N leaves and a distance matrix $\mathbf{D}$, the objective is to find branch lengths that recapitulate the distance matrix.

$N=3$



D_{ij}	1	2	3
1	-	2	5
2		-	4
3			-

$$b_1 + b_2 = d_{12} = 2$$

$$b_2 + b_3 = d_{23} = 4$$

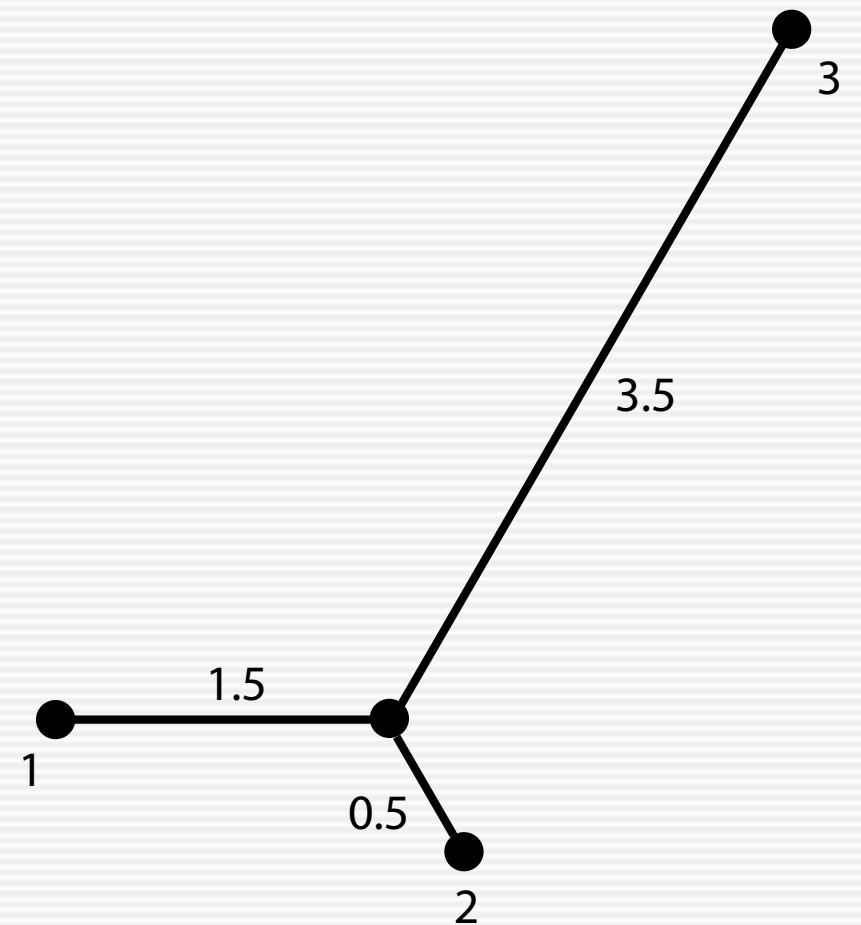
$$b_1 + b_3 = d_{13} = 5$$

$$b_1 \geq 0, b_2 \geq 0, b_3 \geq 0$$

$$b_1 = 1.5$$

$$b_2 = 0.5$$

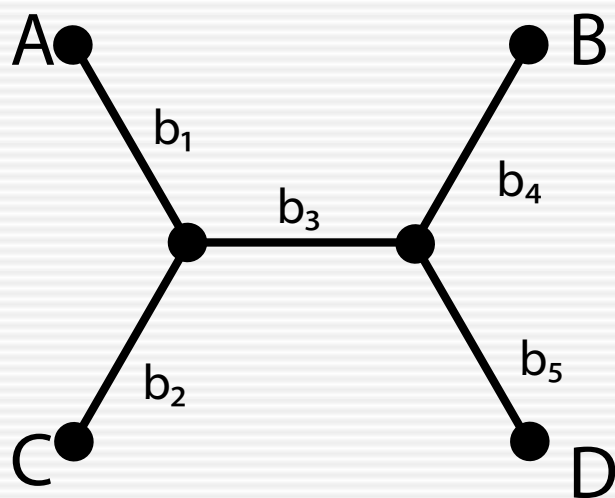
$$b_3 = 3.5$$



ASSUMING THAT THE DISTANCES OBEY THE TRIANGLE INEQUALITY, THE SYSTEM CAN ALWAYS BE SOLVED

WHAT ABOUT $N > 3$?

- For N sequences there are $N(N-1)/2$ pairwise distances and $2N-3$ branches
- This defines an **overdetermined** system of linear equations, when $N > 3$.
- This system will only have solutions if the pairwise distances satisfy certain conditions.



$$d(A, B) = b_1 + b_3 + b_4$$

$$d(A, C) = b_1 + b_2$$

$$d(A, D) = b_1 + b_3 + b_5$$

$$d(B, C) = b_2 + b_3 + b_4$$

$$d(B, D) = b_4 + b_5$$

$$d(C, D) = b_2 + b_3 + b_5$$

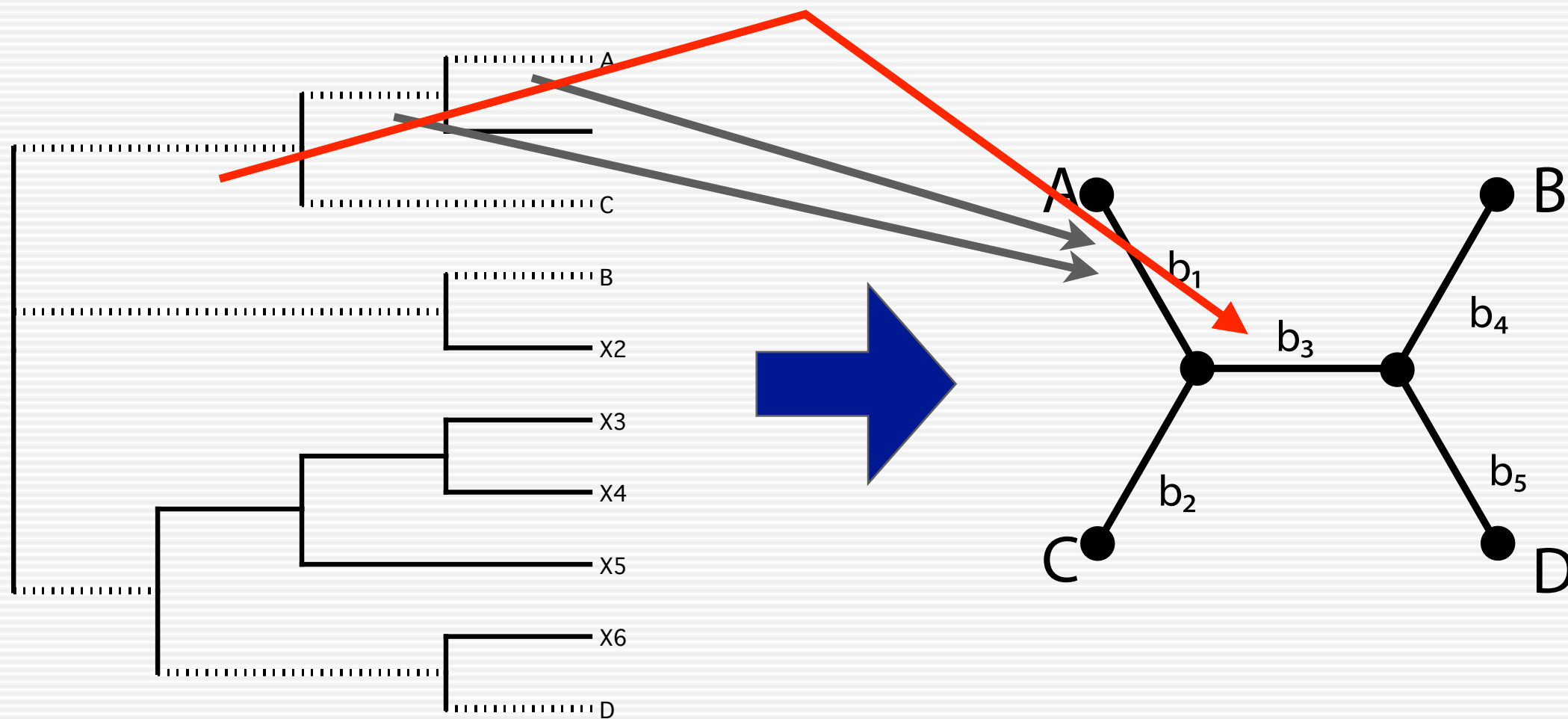
ADDITIVE DISTANCES

- If there exists a tree whose path lengths can recreate the distance matrix on $\mathbf{N}$ data points, then the distance matrix is called additive
- An additive distance matrix satisfies the 4-point condition, formulated as follows
 - For any four points A,B,C,D

$$d(A, B) + d(C, D) \leq \max (d(A, C) + d(B, D), d(A, D) + d(B, C))$$

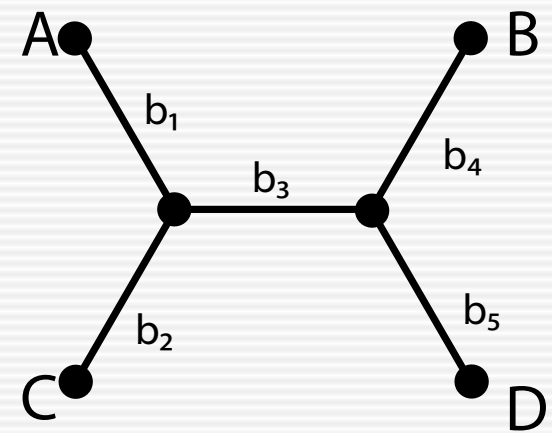
FOUR POINT CONDITION

- If the distances are additive, then there exists a tree which recreates the distance matrix via its path lengths
- Focus only on the paths connecting leaves A,B,C,D
- Collapse the tree to 4 leaves, where each branch length now contains the sum of one or more branch lengths from the original tree.



FOUR POINT CONDITION

- The remaining tree can have one of the three possible 4-leaf topologies.
- Consider this particular case (other two are analogous)



$$\begin{aligned}d(A, B) + d(C, D) &= d(A, D) + d(C, B) = b_1 + b_2 + 2b_3 + b_4 + b_5 \\d(A, C) + d(B, D) &= b_1 + b_2 + b_4 + b_5\end{aligned}$$

- The 4 point condition is satisfied:

$$d(A, B) + d(C, D) \leq \max(d(A, C) + d(B, D), d(A, D) + d(B, C))$$

FOUR POINT CONDITION

- It is a necessary condition: if a distance matrix fails it, then it is NOT additive.
- But is it sufficient?
- Note: an additive matrix obeys the triangle inequality (take $C=D$ in the four point condition)

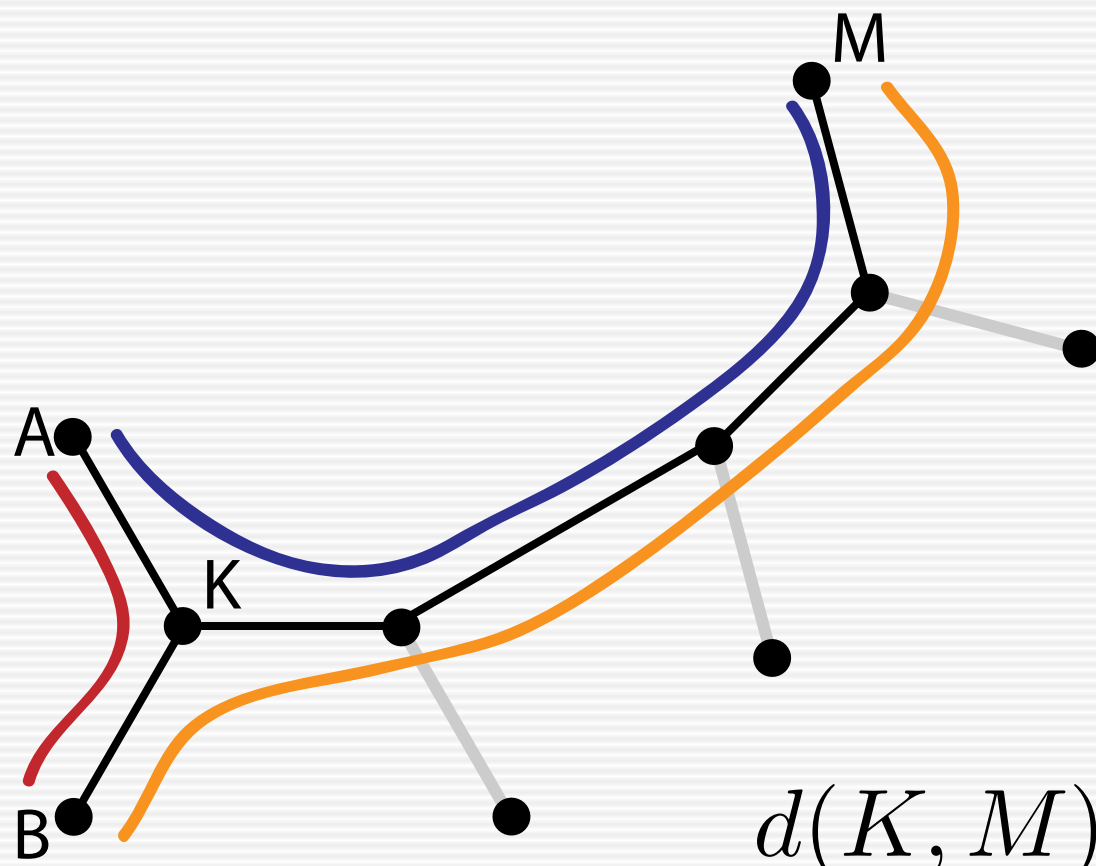
$$\begin{aligned} d(A, B) + d(C, C) &\leq \max(d(A, C) + d(B, C), d(A, C) + d(B, C)) \\ d(A, B) &\leq d(A, C) + d(B, C) \end{aligned}$$

NEIGHBOR JOINING (NJ)

- If the distances are additive, then there is a constructive algorithm that will produce a tree that recapitulates the distances.
- It is due to Saitou and Nei (1987) – the paper has been cited ~20000 times!
- The original authors did not set out to work out an algorithm for additive distances, but their idea turned out to be quite powerful indeed!
- The idea is very similar to clustering
 - Find the two ‘nearest’ sequences
 - Replace them with their parent, recompute distances and iterate until only two sequences remain

NJ IDEA 1

- Computing the distances to the parent of two neighbors.
- Consider computing the distance from two leaves with the same parent to any other leaf.

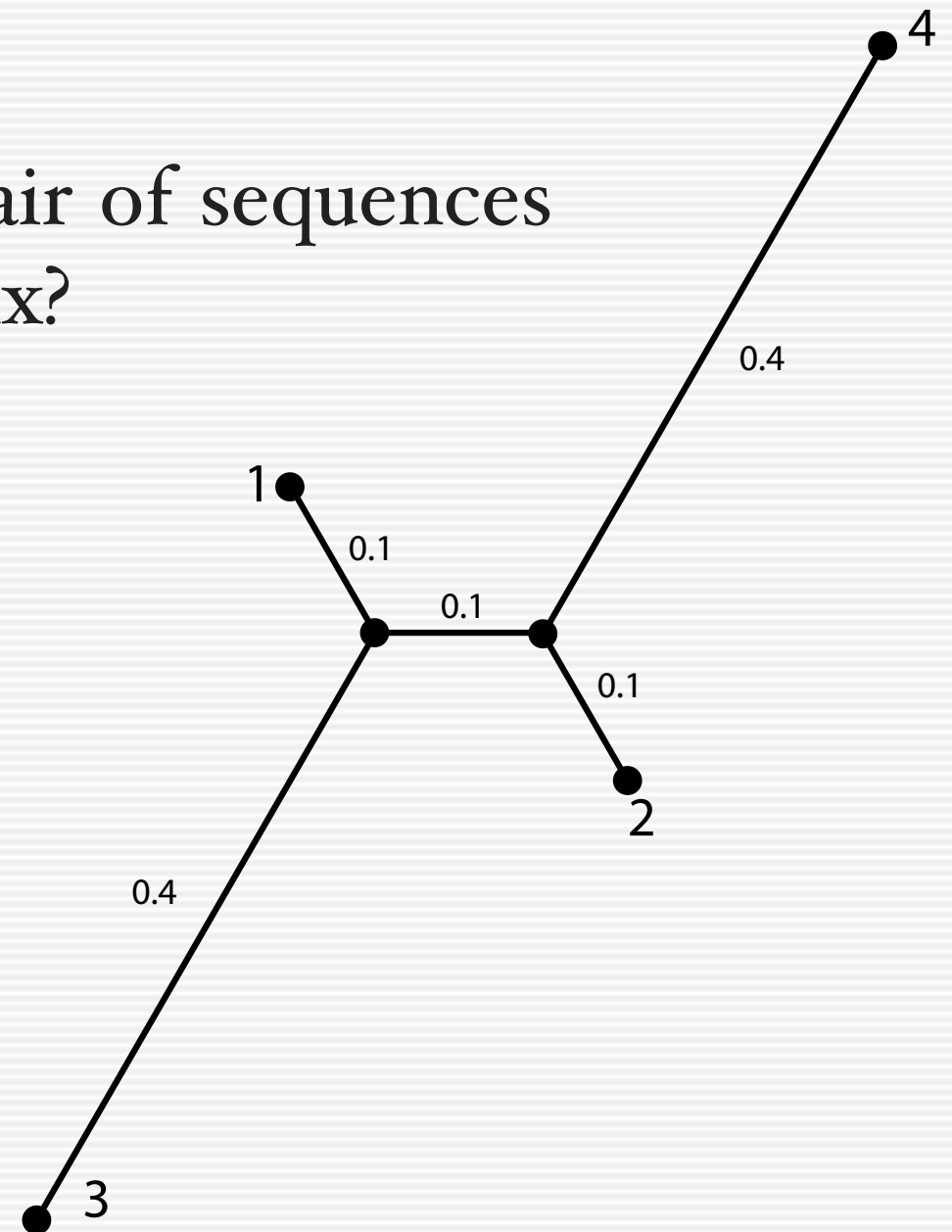


$$d(K, M) = \frac{\overbrace{d(A, M)}^{\text{blue}} + \overbrace{d(B, M)}^{\text{orange}} - \overbrace{d(A, B)}^{\text{red}}}{2}$$

IDEA 2

- How to decide which two nodes are nearest neighbors, having access to nothing but pairwise distances?
- Is it enough to simply consider the pair of sequences with the smallest distance in the matrix?

	1	2	3	4
1	0	0.3	0.5	0.6
2		0	0.6	0.5
3			0	0.9
4				0



IDEA 2 (CONT.)

- Even though it is **not** enough to look just for the shortest distance pair, it IS enough to look at the sequences that are both maximally close to each other and maximally far from the rest of the sequences.
- Define (L is the current number of leaves):

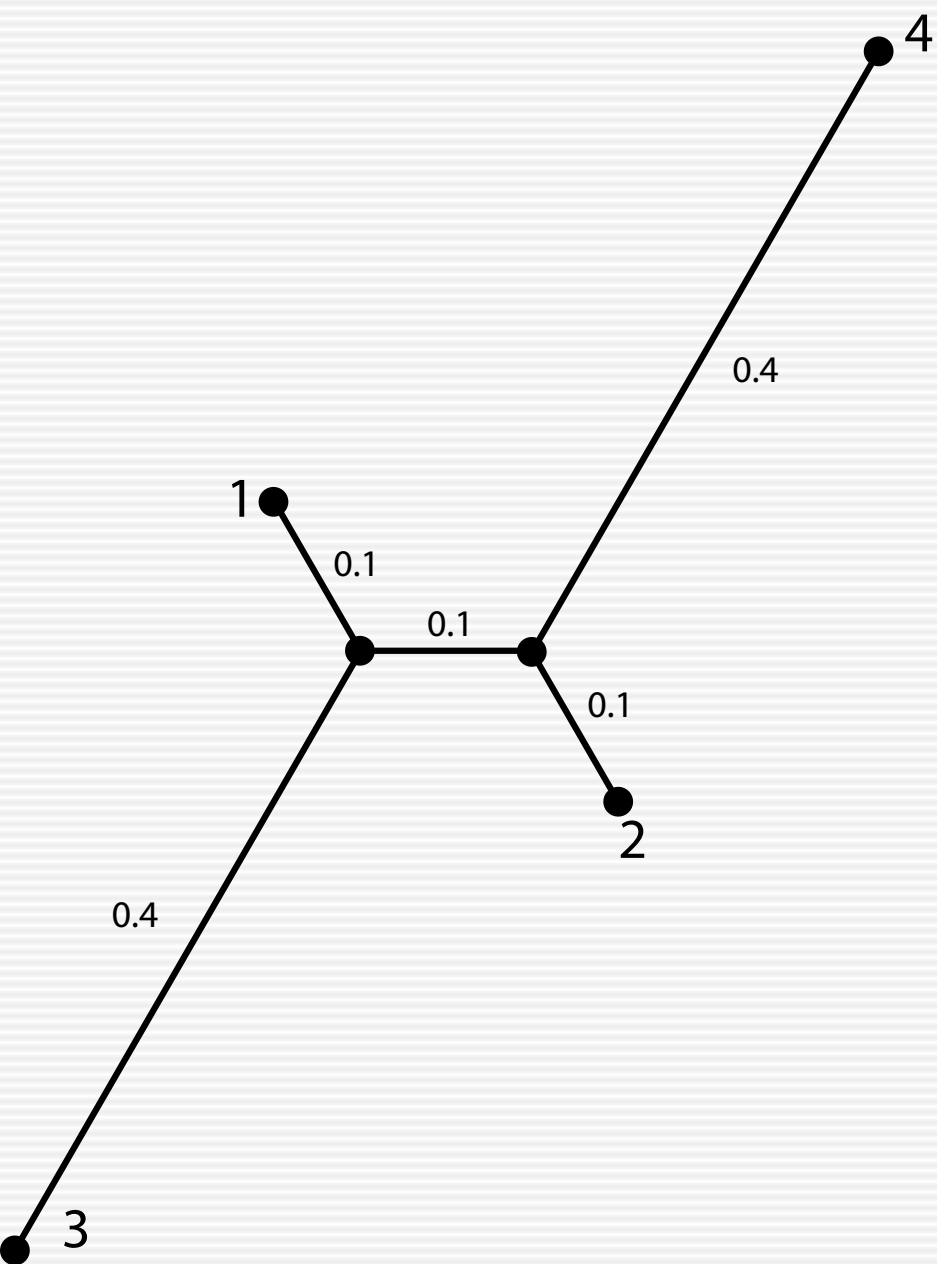
AVERAGE DISTANCE TO
OTHER BRANCHES

$$r_i = \frac{1}{|L| - 2} \sum_{k \in L} d(i, k)$$

RE-WEIGHTED DISTANCES

$$D(i, j) = d(i, j) - (r_i + r_j)$$

- The pair with the smallest $D(i, j)$ are closest neighbors. The proof is not difficult, but requires a few tricks.


 $d(i,j)$

	1	2	3	4
1	0	0.3	0.5	0.6
2	0.3	0	0.6	0.5
3	0.5	0.6	0	0.9
4	0.6	0.5	0.9	0

 R_I

1	0.7
2	0.7
3	1
4	1

 $D(i,j)$

	1	2	3	4
1	0	-1.1	-1.2	-1.1
2		0	-1.1	-1.2
3			0	-1.1
4				0

IDEA 3

- How to partition $d(i,j)$ into the branch lengths leading from the parent (k) to i and j ?

$$d(i, k) = \frac{1}{2}(d(i, j) + r_i - r_j) \quad d(j, k) = d(i, j) - d(i, k)$$

PROOF

$$\boxed{2} \quad r_j = d(j, k) + \frac{d(i, j)}{|L| - 2} + \frac{1}{|L| - 2} \sum_{m \in L, m \neq i, m \neq j} d(k, m)$$

I

$$r_i = \frac{1}{|L| - 2} \sum_{m \in L} d(i, m)$$

$$= \frac{d(i, j)}{|L| - 2} + \frac{1}{|L| - 2} \sum_{m \in L, m \neq i, m \neq j} d(i, m)$$

$$= \frac{d(i, j)}{|L| - 2} + \frac{1}{|L| - 2} \sum_{m \in L, m \neq i, m \neq j} d(i, k) + d(k, m)$$

$$= d(i, k) + \frac{d(i, j)}{|L| - 2} + \frac{1}{|L| - 2} \sum_{m \in L, m \neq i, m \neq j} d(k, m)$$

$$\boxed{3} \quad \begin{aligned} d(i, j) + r_i - r_j &= \\ d(i, k) + d(j, k) + d(i, k) - d(j, k) &= \\ 2d(i, k) \end{aligned}$$

PUTTING IT ALL TOGETHER

ALGORITHM NEIGHBOR JOIN

Data : A distance matrix on N sequences, $d(i, j)$

Result: The tree (topology + branch lengths) that recapitulates $d(i, j)$
if the matrix is additive

```
1  $L \leftarrow$  the set of all leaves;
2  $T \leftarrow$  a graph with nodes (disconnected) set to the leaf set  $L$ ;
3 while  $|L| > 2$  do
4   Pick a pair  $(i, j)$  from  $L$  for which  $D(i, j)$  is minimal (break ties arbitrarily);
5   Define a new parent node  $k$  and set  $d(m, k) = \frac{1}{2}(d(i, m) + d(j, m) - d(i, j))$  for
   all  $m \in L \setminus \{i, j\}$ ;
6   Add  $k$  to  $T$ . Join  $k$  and  $i$  with branch length  $d(i, k) = \frac{1}{2}(d(i, j) + r_i - r_j)$ . Joint
    $k$  and  $j$  with branch length  $d(j, k) = d(i, j) - d(i, k)$ ;
7   Remove  $i, j$  from  $L$ . Add  $k$  to  $L$ ;
8   Update matrix  $d$  to remove  $i, j$  and add  $k$ .;
9 end
10 Join the last two remaining nodes in  $L$ ,  $i, j$  with a branch in  $T$  using length  $d(i, j)$ .
11 return  $T$ ;
```

RUN TIME $O(N^?)$

STEP 1

$d(i,j)$

	1	2	3	4
1	0	0.3	0.5	0.6
2	0.3	0	0.6	0.5
3	0.5	0.6	0	0.9
4	0.6	0.5	0.9	0

r_i

1	0.7
2	0.7
3	1
4	1

$D(i,j)$

	1	2	3	4
1	0	-1.1	-1.2	-1.1
2		0	-1.1	-1.2
3			0	-1.1
4				0

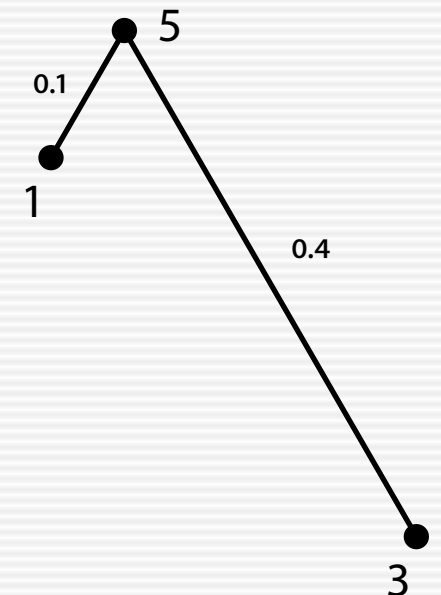
Join 1,3 at node 5

$$d(1, 5) = 1/2(0.5 + 0.7 - 1) = 0.1$$

$$d(3, 5) = 0.5 - 0.1$$

Updated $d(i,j)$

	2	4	5
2	0	0.5	0.2
4		0	0.5
5			0



STEP 2

$d(i,j)$

	2	4	5
2	0	0.5	0.2
4		0	0.5
5			0

r_i

	2	4	5
2	0.7		
4	1.0		
5	0.7		

$D(i,j)$

	2	4	5
2	0	-1.2	-1.2
4		0	-1.2
5			0

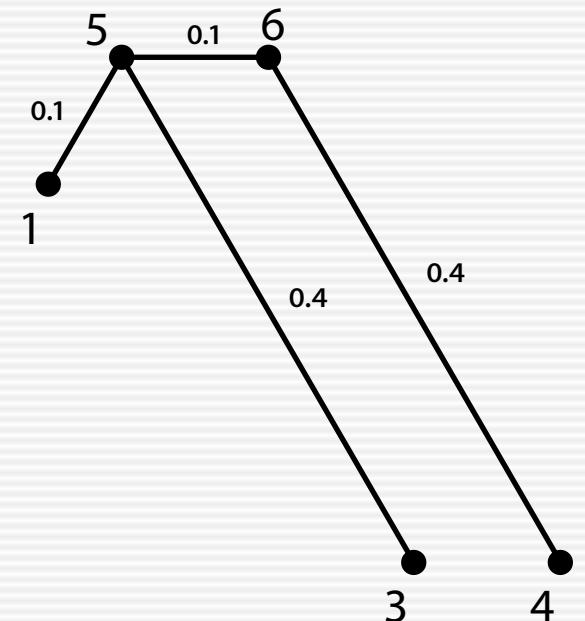
Join 4,5 at node 6

Updated $d(i,j)$

$$d(4,6) = 1/2(0.5 + 1.0 - 0.7) = 0.4$$

$$d(5,6) = 0.5 - 0.4 = 0.1$$

	2	6
2	0	0.1
6		0

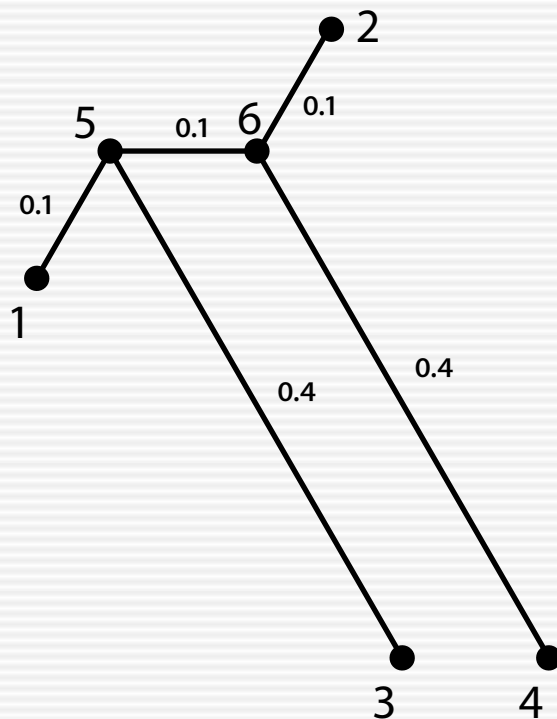


STEP 3

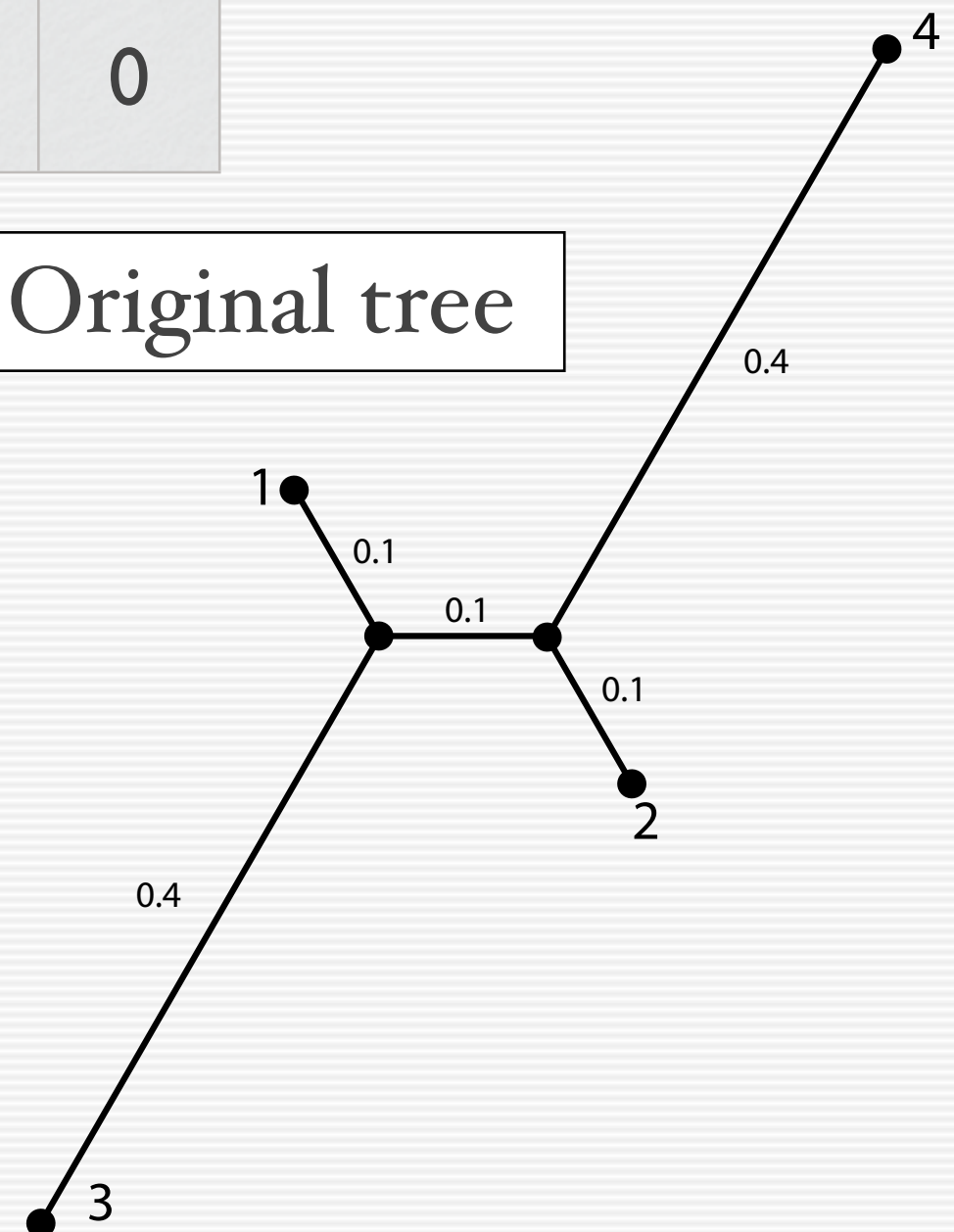
Join the remaining 2
nodes (6 and 2)

	2	6
2	0	0.1
6		0

NJ tree



Original tree



BIOLOGY IS MESSY

- Comparisons of biological sequences very rarely generate additive distance matrices
- NJ can be applied to non-additive matrices and generally performs quite well – many advanced tree search programs take NJ trees as good starting points, for example

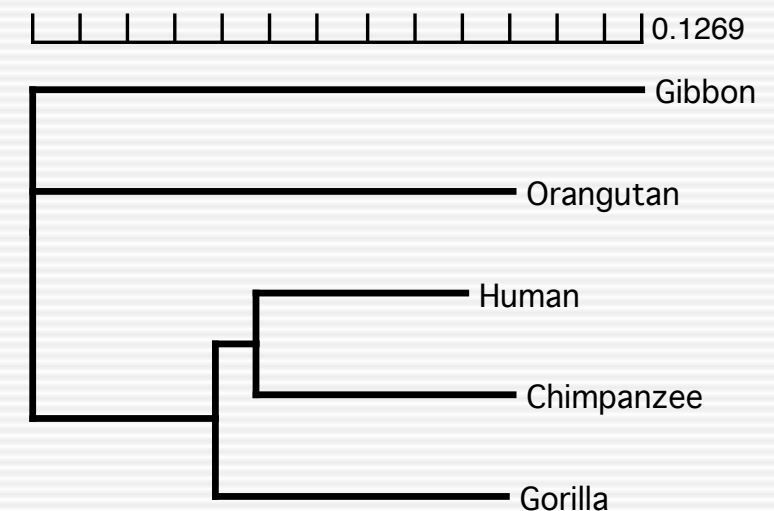
Genetic distances

	Human	Chimpanzee	Gorilla	Orangutan	Gibbon
Human	0	0.0978173	0.115615	0.187822	0.21524
Chimpanzee		0	0.12003	0.204948	0.228186
Gorilla			0	0.198166	0.227662
Orangutan				0	0.227143
Gibbon					0

NJ tree distances

	Human	Chimpanzee	Gorilla	Orangutan	Gibbon
Human	0	0.0978173	0.11296	0.190827	0.217544
Chimpanzee		0	0.122685	0.200553	0.22727
Gorilla			0	0.199555	0.226272
Orangutan				0	0.227143
Gibbon					0

NJ tree



NON-ADDITIVE MATRICES

- One can try to find the tree that minimizes an error between the distance matrix $d(i,j)$ and tree-induced pairwise distances $T(i,j)$
- For example – least squares

$$\min_T \sum_{i,j} (d(i,j) - T(i,j))^2$$

- This problem is NP-hard (need to look at all trees, potentially)
- Difficult to quantify how one tree compares to the other (e.g. if one achieves error 0.1 and the other - 0.105 – are they really that different?)

ALGORITHMIC VS OPTIMALITY BASED TREE RECONSTRUCTION

- Neighbor joining (and some other methods) are **algorithmic** – they produce a single tree from the input.
 - Advantage: fast
 - Disadvantage: have no idea how the found tree compares to the rest $(2^{N-5})!!$ -1 trees.
- Optimality based criterion search states:
 - Any candidate tree, T and be assigned a score, $s(T)$
 - We seek to minimize (or maximize) $s(T)$ over all possible trees
 - Advantage: compares many trees, gives one an idea of how good the proposed solution is
 - Disadvantage: slow (many trees), still need to explore a combinatorial set of possible solutions.

PARSIMONY

[HTTP://EN.WIKTIONARY.ORG/WIKI/PARSIMONY](http://en.wiktionary.org/wiki/parsimony)

Noun

parsimony (*uncountable*)

[[edit](#)]



Wikipedia has an article on:

[Parsimony](#)

1. Great reluctance to spend money unnecessarily.

■ 1776, Adam Smith, *An Inquiry into the Nature and Causes of the Wealth of Nations*:

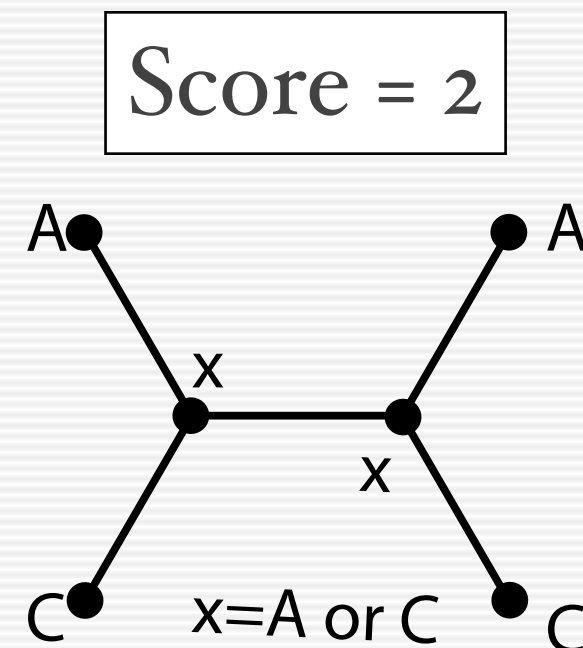
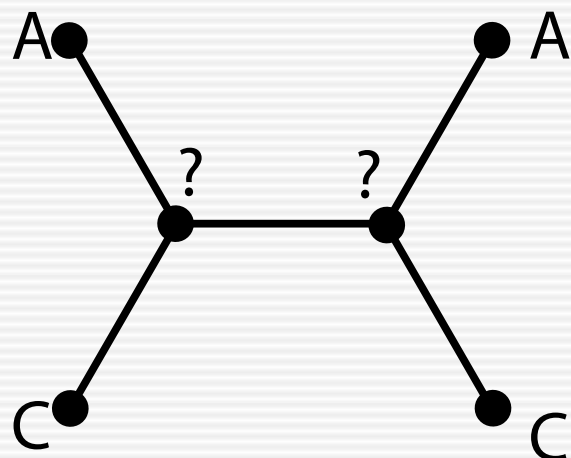
Parsimony, and not industry, is the immediate cause of the increase of capital. Industry, indeed, provides the subject which parsimony accumulates. But whatever industry might acquire, if parsimony did not save and store up, the capital would never be the greater.

2. By analogy from (1), principle of using the least resources or explanations to solve a problem.

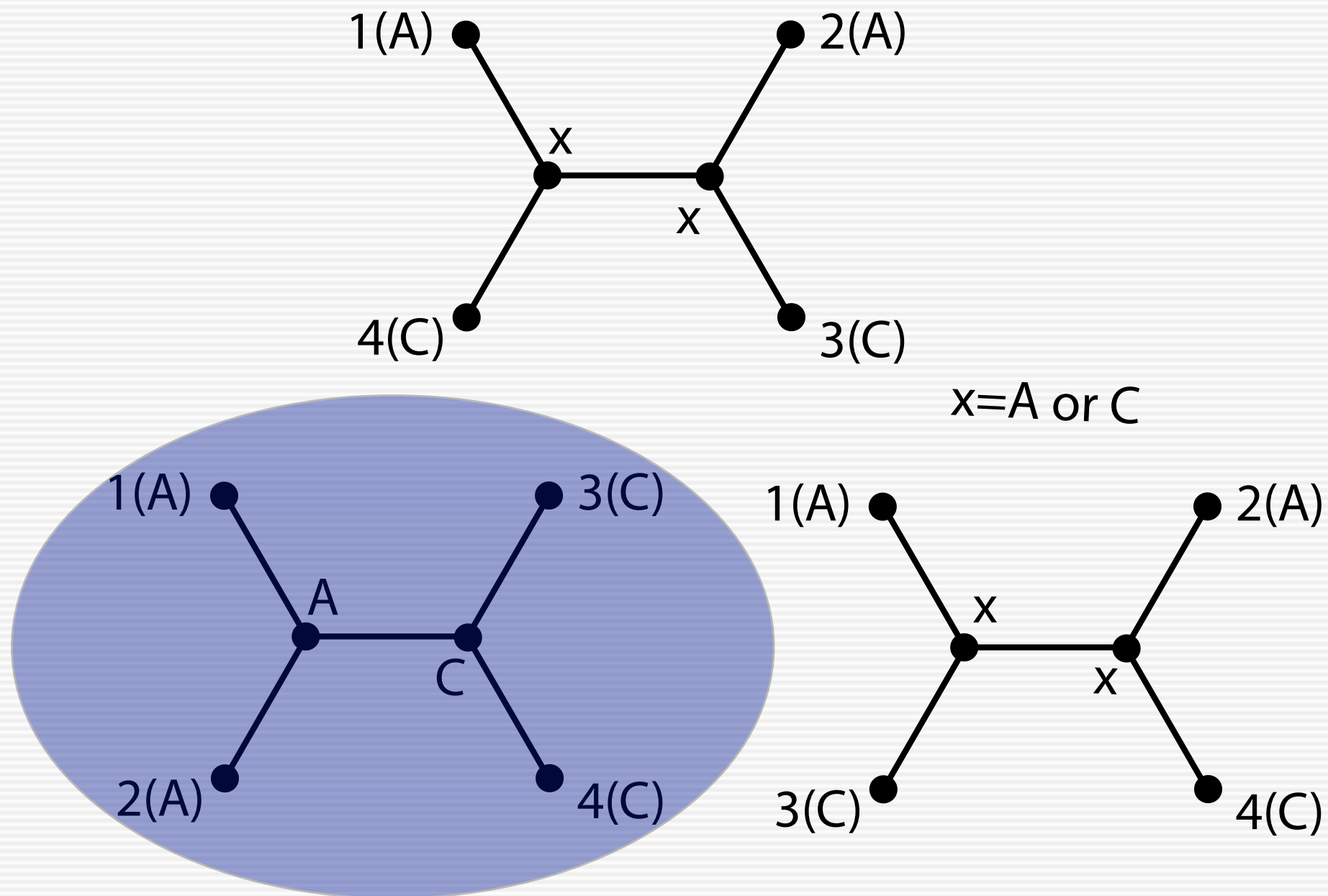
- ☐ The idea is to find the tree that explains the observed pattern of sequences in the fewest/cheapest possible sequence of steps (e.g. substitutions).
- ☐ Two issues:
 - ☐ Given the topology and leaf labels, find the minimum cost of the tree (an edit distance problem)
 - ☐ Find the topology that minimizes said cost

PARSIMONY EXAMPLES

- Let each leaf be labeled with a letter from some alphabet
 - Nucleotides, amino-acid residues, presence or absence of a trait
- Define the cost of changing one letter to another (a substitution), $c(\mathbf{x}, \mathbf{y})$
 - The simplest case $c(\mathbf{x}, \mathbf{y}) = 1$, if $\mathbf{x} \neq \mathbf{y}$, and $c(\mathbf{x}, \mathbf{x}) = 0$.
- How would you assign interior node labels to minimize the total cost of the tree below?



TOPOLOGY SEARCH EXAMPLE



□ Which topology is the best?

PARSIMONY ON MOLECULAR SEQUENCES

- Consider an alignment of nucleotide sequences

	20	30	40	50	60	70	80	90
Human	GTC	ATTCTCATAA	TCGCCCACGG	ACTTACATCC	TCATTACTAT	TCTGCCTAGC	AAACTCAAAC	TACGAACGCA
Chimpanzee	ATT	ATCCTCATAA	TCGCCCACGG	ACTTACATCC	TCATTATTAT	TCTGCCTAGC	AAACTCAAAT	TATGAACGCA
Gorilla	GTT	GTTCTTATAA	TTGCCACGG	ACTTACATCA	TCATTATTAT	TCTGCCTAGC	AAACTCAAAC	TACGAACGAA
Orangutan	ACC	ACCCTCATGA	TTGCCATGG	ACTCACATCC	TCCCTACTGT	TCTGCCTAGC	AAACTCAAAC	TACGAACGAA
Gibbon	ACC	GTCTCATAA	TCGCCCACGG	ACTAACCTCT	TCCCTGCTAT	TCTGCCTTGC	AAACTCAAAC	TACGAACGAA

- We seek the topology that minimizes the cumulative parsimony score across all sites of the alignment
- Cumulative score is simply the sum of site-by-site scores
- To score each site, we need to solve a parsimony problem (assign interior labels optimally at that site)
- Want to be able to do it for (almost) arbitrary cost functions

INFORMATIVE SITES

- For standard parsimony (score = 0 or 1 for match or mismatch) some alignment columns will have the same score for all topologies – these sites are called uninformative



The diagram shows a sequence alignment of five species (Human, Chimpanzee, Gorilla, Orangutan, Gibbon) across 10 columns (20-90). Two callouts highlight specific sites:

- Invariable sites. Score 0 for all topologies:** Points to column 40, where all species have the same nucleotide (C).
- Single difference. Score 1 for all topologies:** Points to column 90, where all species have the same nucleotide (A).

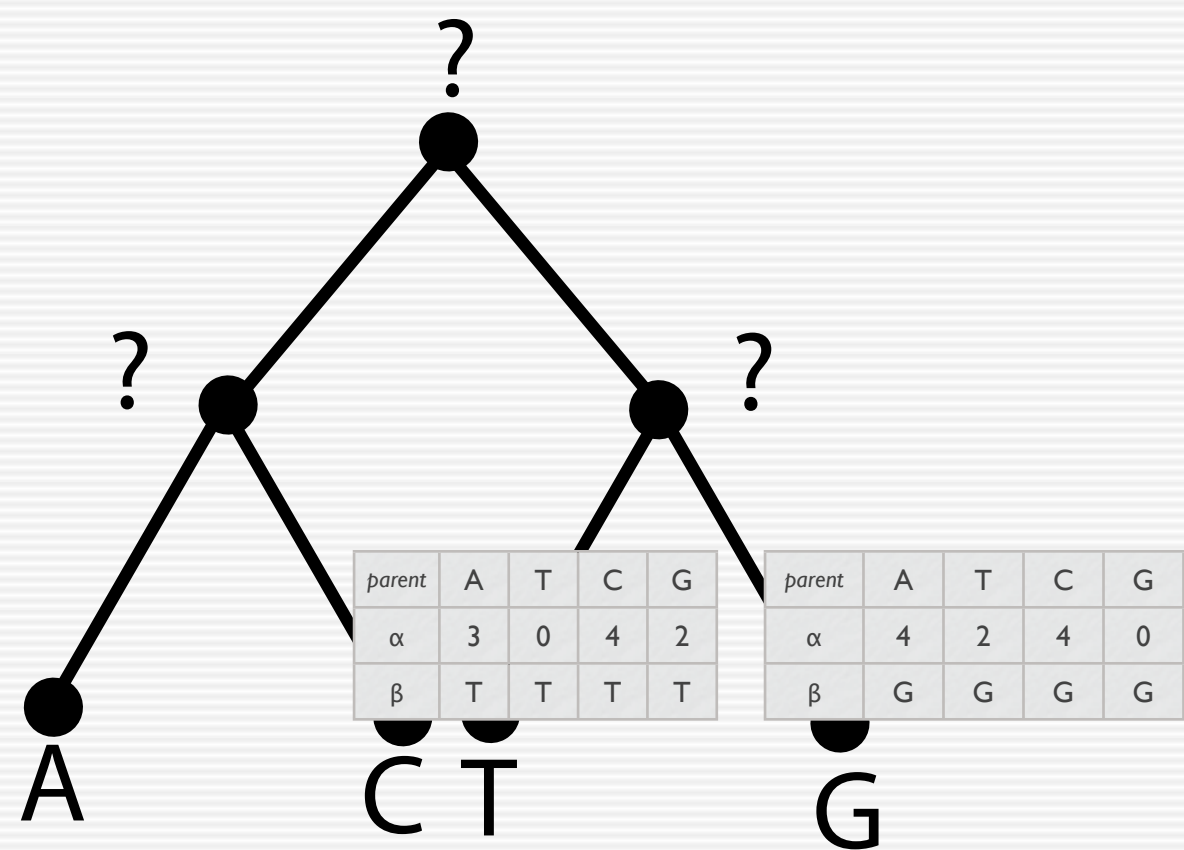
	20	30	40	50	60	70	80	90
Human	GTC	ATTCTCATAA	TCGCCCCACGG	ACTTACATCC	TCATTACTAT	TCTGCCTAGC	AAACTCAAAC	TACGAACGCA
Chimpanzee	ATT	ATCCTCATAA	TCGCCCCACGG	ACTTACATCC	TCATTATTAT	TCTGCCTAGC	AAACTCAAAT	TATGAACGCA
Gorilla	GTT	GTTCTTATAA	TTGCCCCACGG	ACTTACATCA	TCATTATTAT	TCTGCCTAGC	AAACTCAAAC	TACGAACGAA
Orangutan	ACC	ACCCTCATGA	TTGCCCCATGG	ACTCACATCC	TCCCTACTGT	TCTGCCTAGC	AAACTCAAAC	TACGAACGAA
Gibbon	ACC	GCCTCATAA	TCGCCCCACGG	ACTAACCTCT	TCCCTGCTAT	TCTGCCTTGC	AAACTCAAAC	TACGAACGAA

- An informative sites must have at least two different characters with at least two instances of each character.

SANKOFF'S ALGORITHM

- Permits, for a fixed topology, to compute the optimal interior node label assignment and the parsimony score for a user specified cost function $\mathbf{c}(\mathbf{x}, \mathbf{y})$
- Uses the fact that the score of the subtree rooted at some interior node $\mathbf{n}$ is independent of the rest of the tree given the label of $\mathbf{n}$'s parent.
- For each node $\mathbf{n}$ in the tree, the algorithm populates two arrays (of dimension equal to the size of the alphabet) for each node – leaf and internal – in the topology (except the arbitrarily chosen interior node designated as root):
 - $\alpha_n(i)$ - the optimal score of the subtree rooted at $\mathbf{n}$, given that the label of $\mathbf{n}$'s parent is $\mathbf{i}$.
 - $\beta_n(i)$ - the label at $\mathbf{n}$ that achieves score $\alpha_n(i)$
- The arrays can be computed recursively from the leaves up to the tree root
- The second pass from the root down to the leaves assigns the optimal labels

STEP 1: Traverse the tree from the leaves up (postorder) and populate cost/label arrays

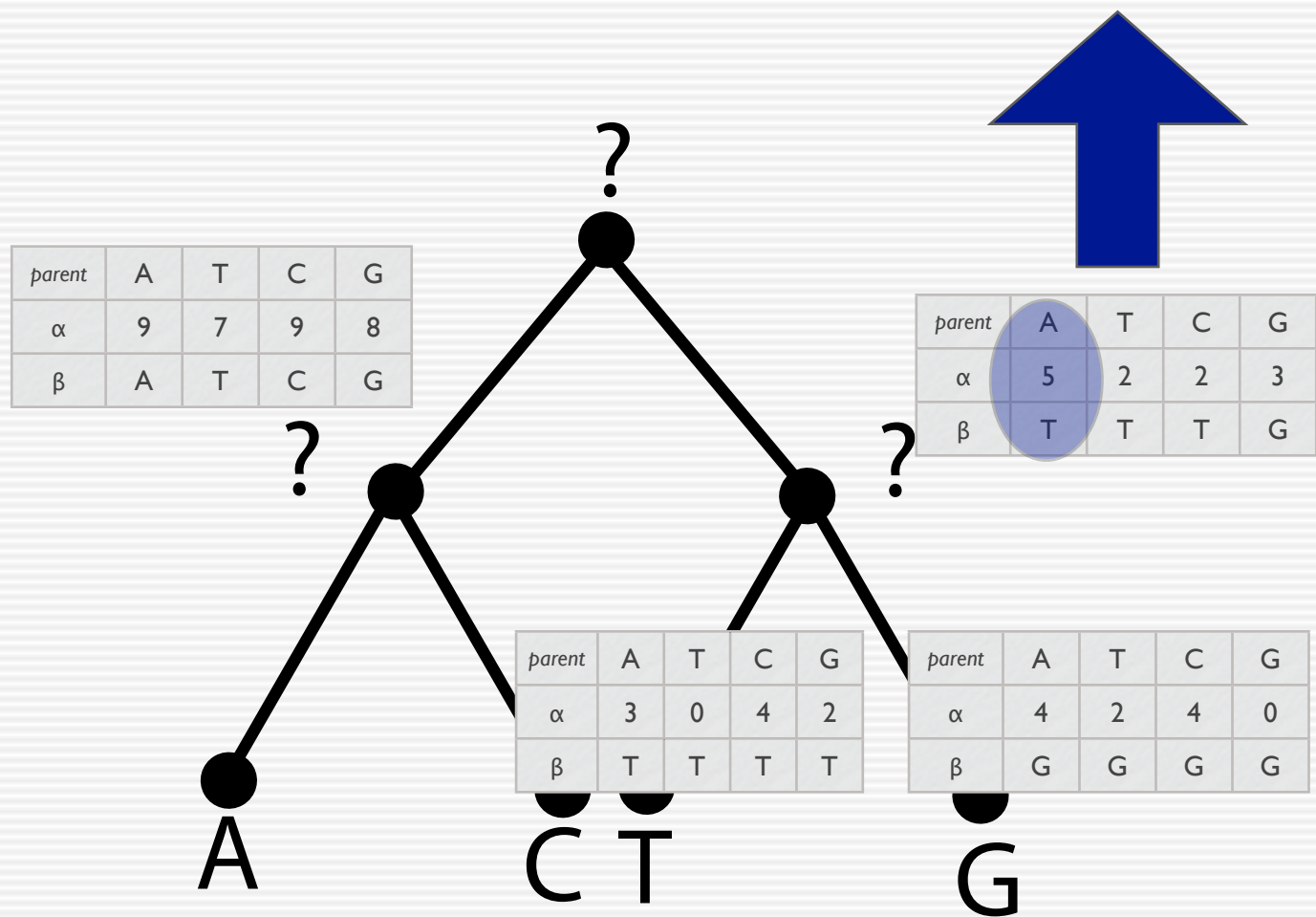


Substitution costs

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

STEP I...

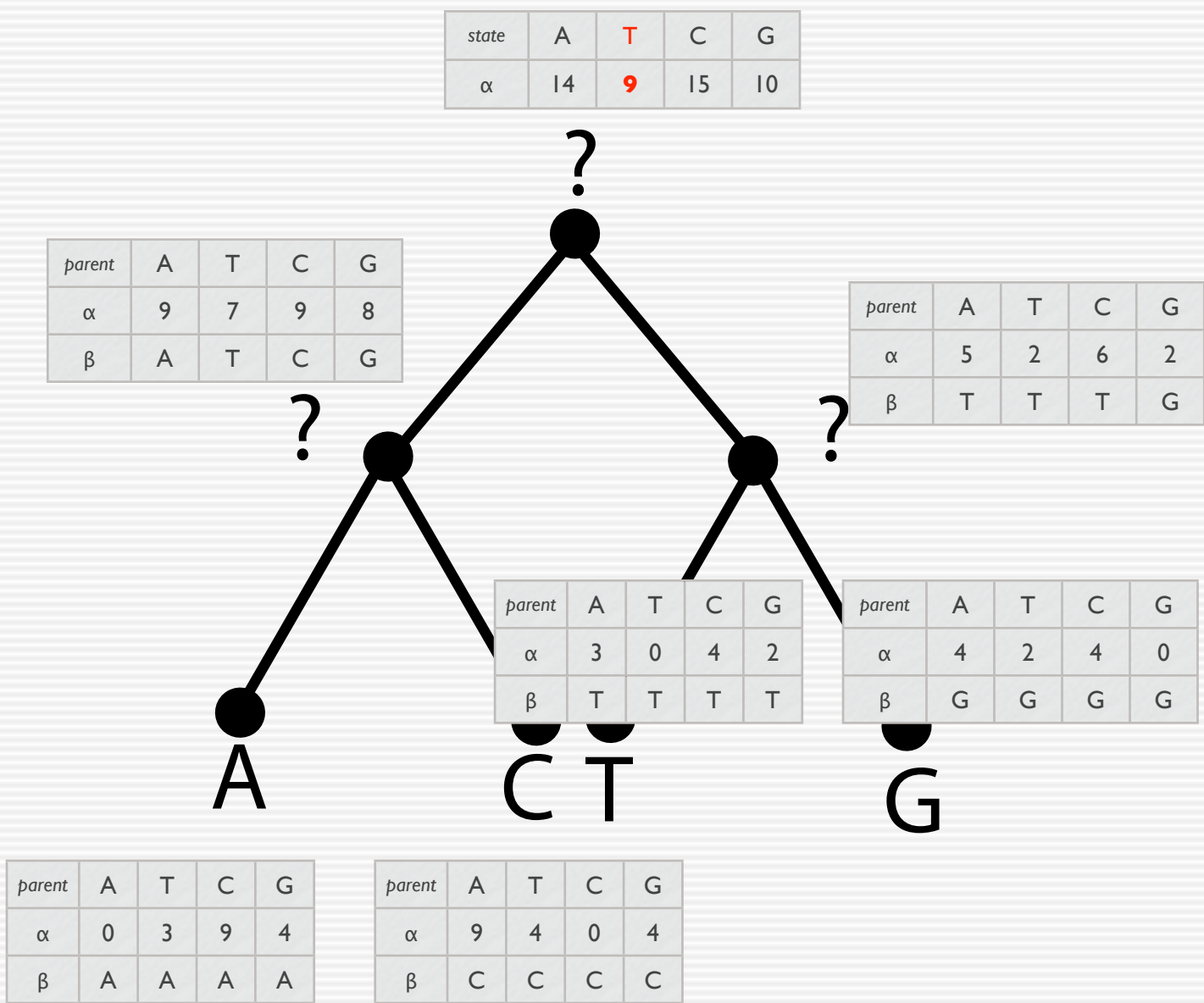
TRY A: $c(A,A)+3+4=7$
TRY T: $c(A,T)+0+2=5$
 TRY C: $c(A,C)+4+4=17$
 TRY G: $c(A,G)+2+0=6$



Substitution costs

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

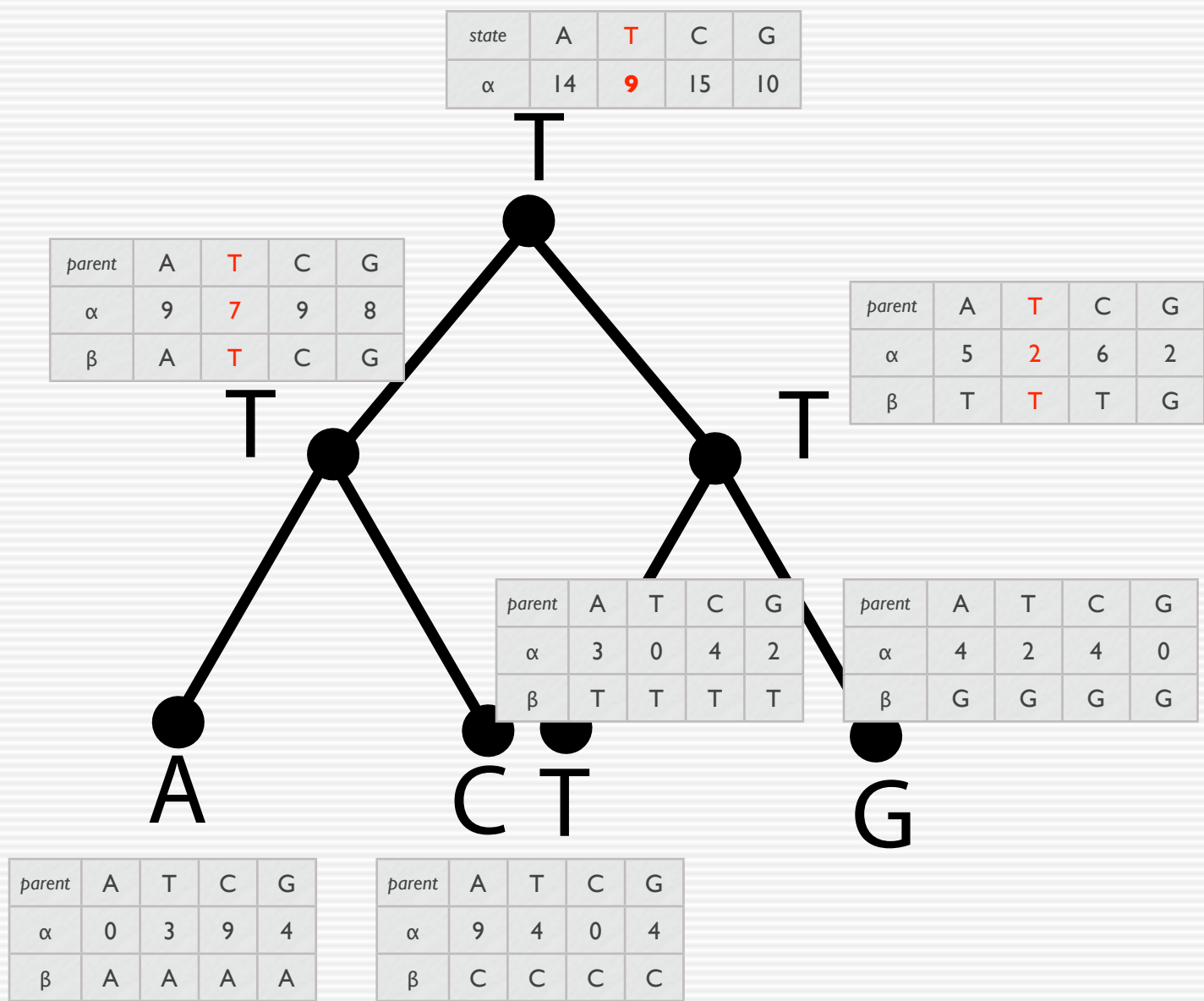
STEP 2: label the root



Substitution costs

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

STEP 3: label the rest of the tree

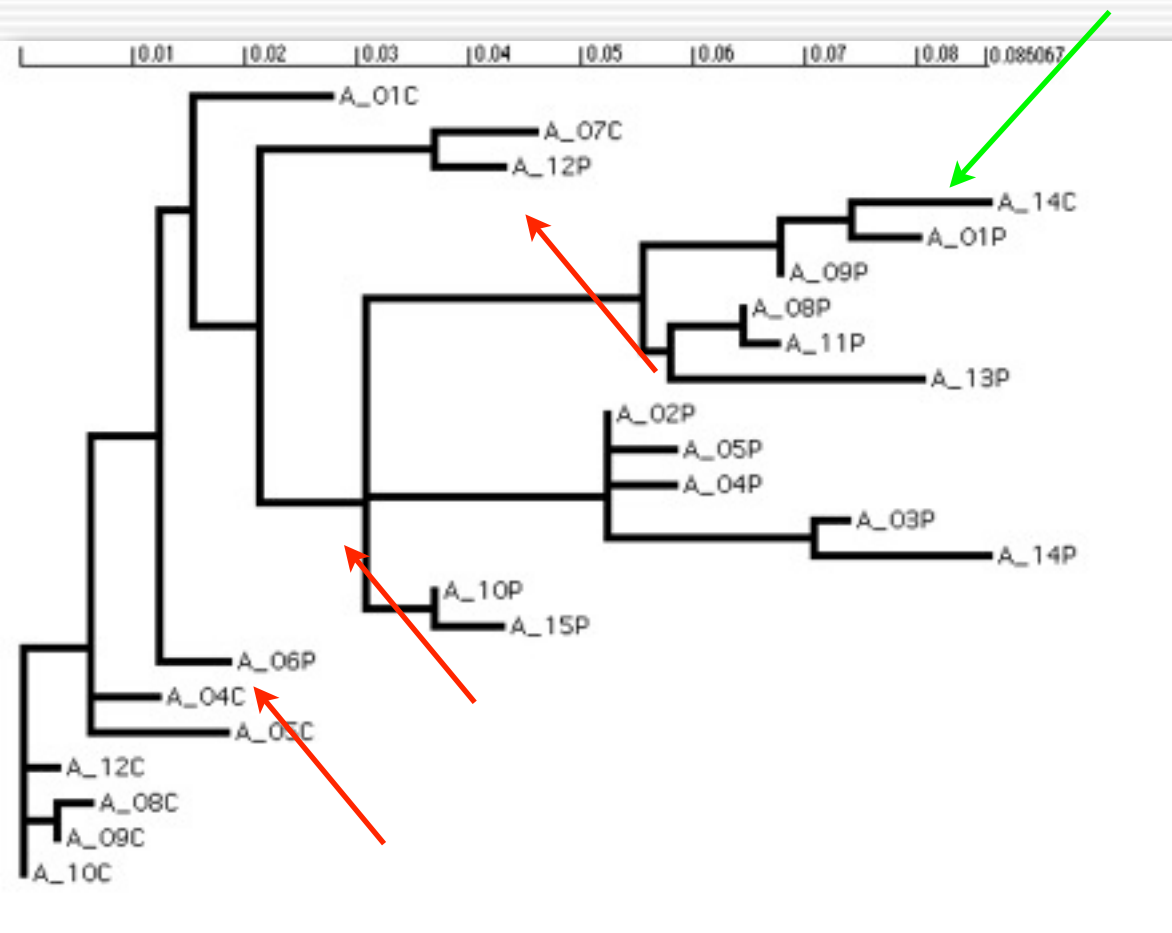


Substitution costs

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

Optimal cost: 9. Run time?

HIV COMPARTMENTALIZATION



C - CNS

P - Blood plasma

Arrows - jumps inferred by parsimony

☐ HIV can colonize different tissues or compartments

☐ Blood

☐ Central nervous system

☐ Lymph nodes

☐ Genital tract

☐ Sometimes the virus jumps compartments often, but sometimes rarely

☐ In the latter case, there are separate viral populations that can complicate treatment and lead to poor prognosis

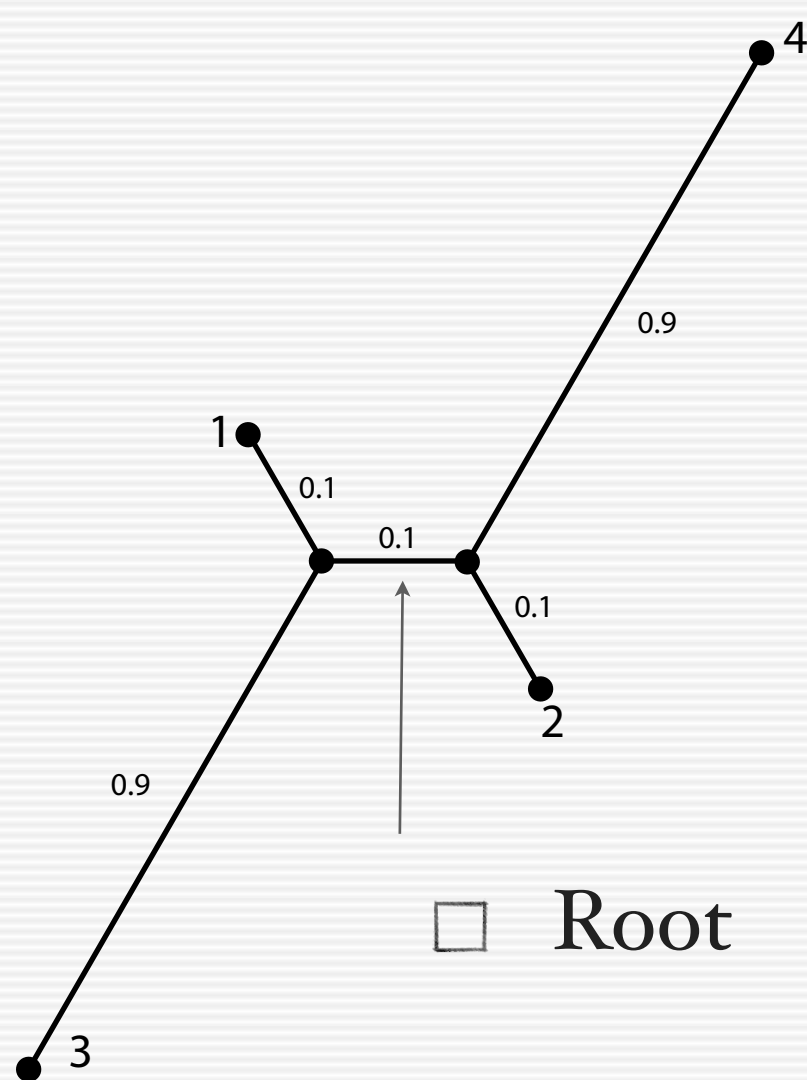
☐ We can use parsimony to map how often the virus jumps between compartments and run a statistical test to decide if its frequent or rare.

MORE ON PARSIMONY

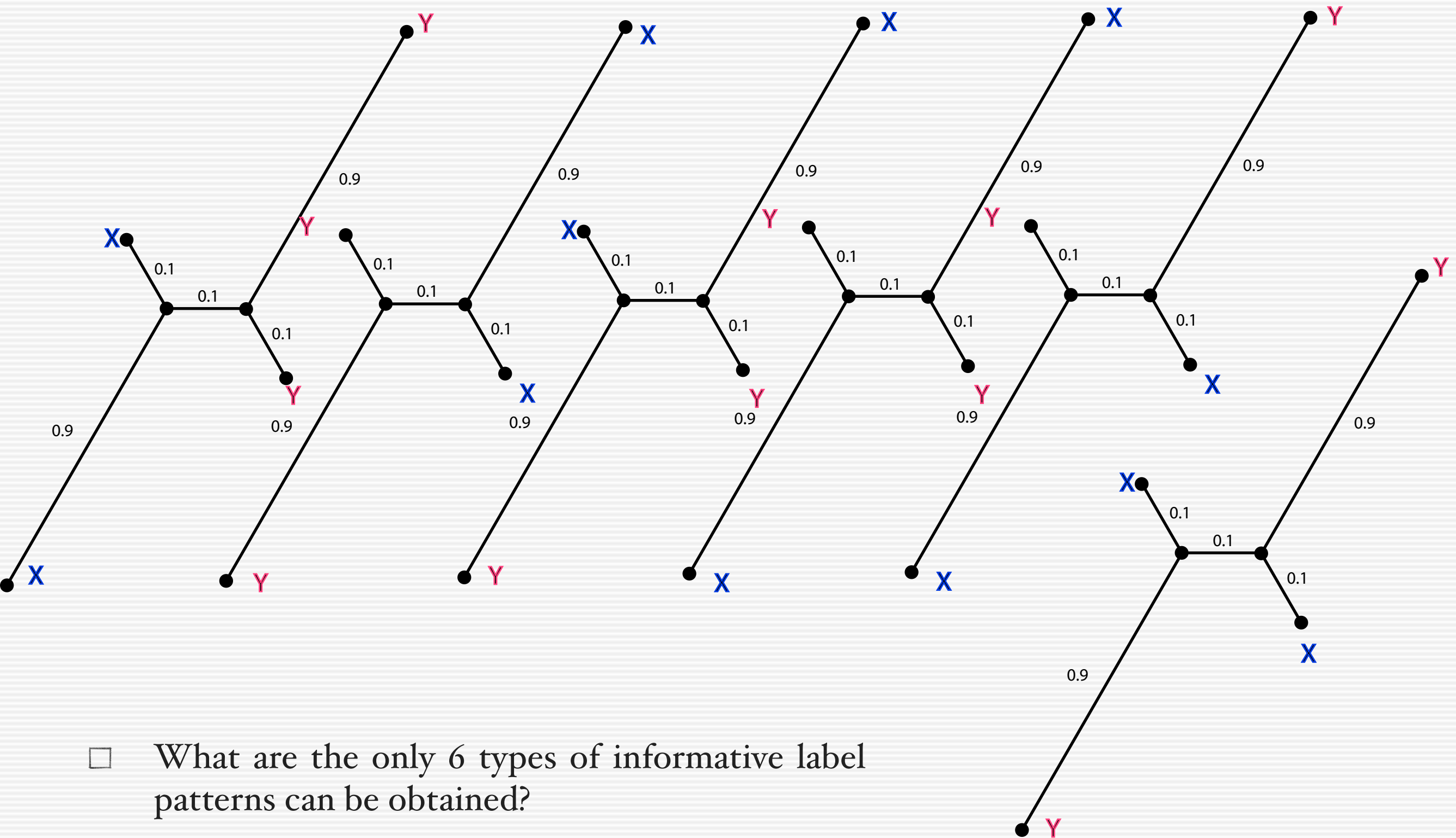
- Can be implemented very efficiently, permitting rapid screens of large sets of candidate trees
- Can be coupled with a branch and bound algorithm to exhaustively explore all topologies on ~20-30 taxa
- Works well if the assumptions of the method are not violated
 - The scoring matrix is reasonable
 - Branch lengths are short and not too different from one another

BUT...

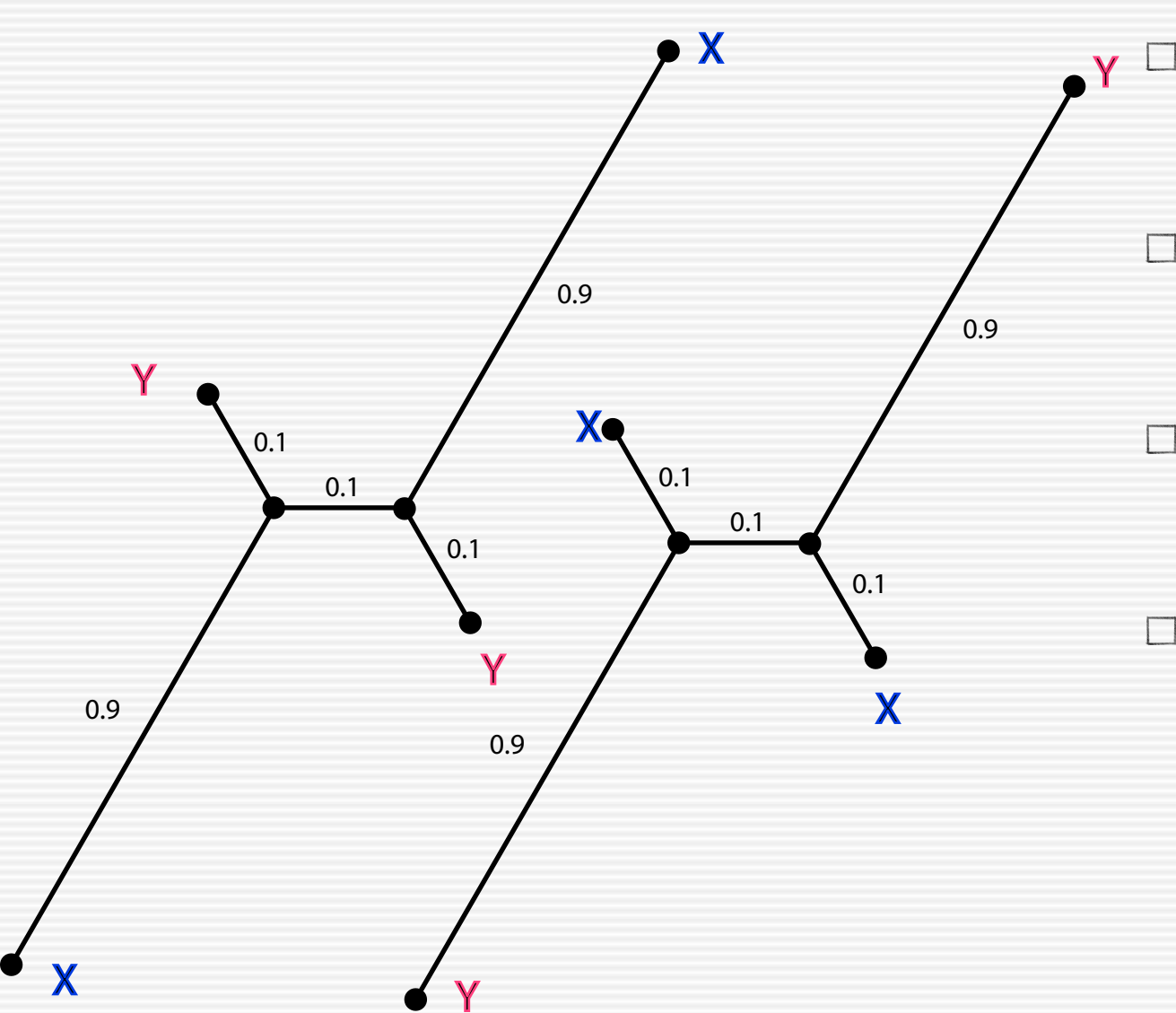
- ❑ Parsimony can also behave very poorly
- ❑ Under certain scenarios, the more data you give the method, the more certain it will be about inferring an incorrect tree
- ❑ This behavior is called positively misleading
- ❑ Example was given by Joe Felsenstein in a lead-up to his seminal work on using probabilistic models to reconstruct phylogenies.



- ☐ Consider the tree on the left.
- ☐ Treat each branch length as the probability that the sequence will mutate along this branch.
- ☐ Generate many sets of labels (alignment sites) using this model
- ☐ Reconstruct trees using parsimony (simple scoring function) from all sites.
- ☐ Which tree will parsimony tend to recover?



- ☐ What are the only 6 types of informative label patterns can be obtained?
- ☐ Which two have the highest probability of being generated?

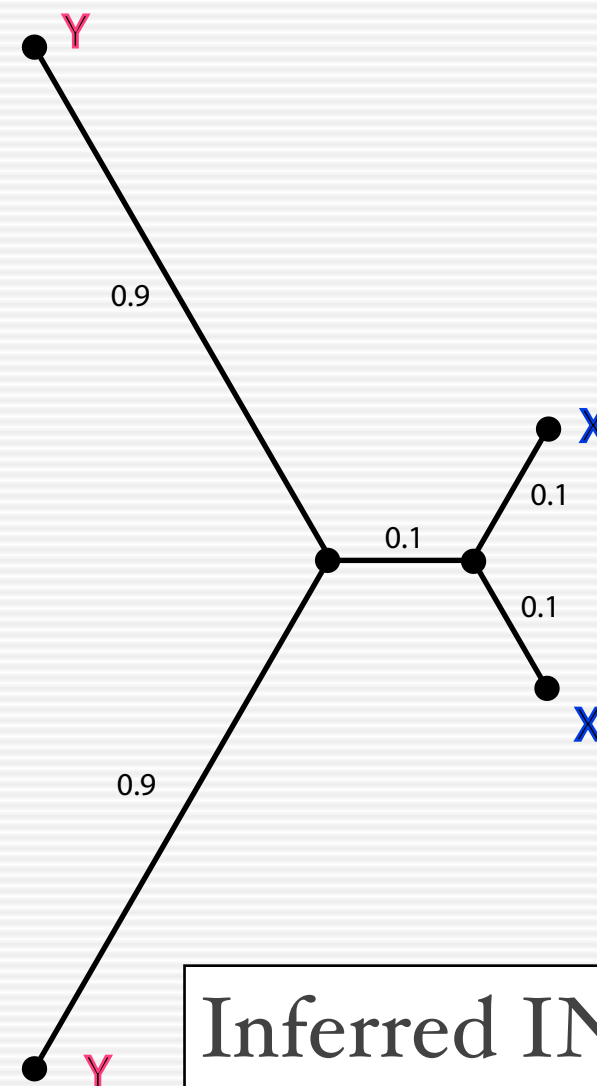


These are the two most frequent informative) patterns (by a considerable margin)

Which topology has the lowest parsimony score for these patterns?

Felsenstein termed this phenomenon: long branch attraction

Maximum likelihood phylogenetic inference does not have this issue (at least if the model is not too wrong)



Inferred INCORRECT tree