In the Name of God, the Merciful, the Compassionate

Introduction to Bioinformatics
10 - Phylogenetic Tree Construction Methods and Programs

Instructor: Hossein Zeinali

Amirkabir University of Technology

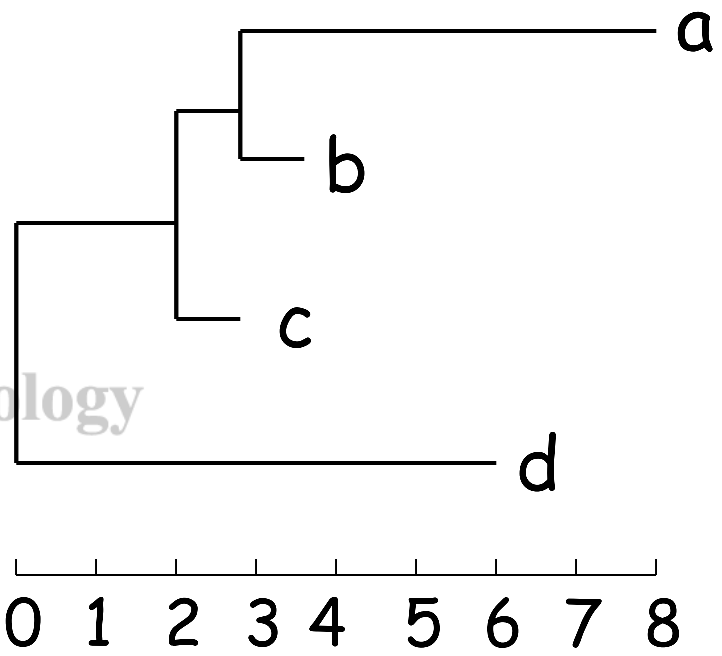
Tree Construction Methods

- There are two main categories of tree building methods.
- **Distance-based:**
  - Distance: the amount of dissimilarity between pairs of sequences, computed on the basis of sequence alignment.
  - Assumes all sequences involved are homologous and that tree branches are additive
    - The distance between two taxa equals the sum of all branch lengths connecting them.
- **Character-based:**
  - Characters are molecular sequences from individual taxa.
  - Main assumption: characters at corresponding positions in a MSA are homologous among the sequences involved.
  - Second assumption: each character evolves independently and is therefore treated as an individual evolutionary unit.
  - Consider the entire MSA

Distance-Based Methods

- Given a MSA and an evolutionary model, calculate the distance between all pairs of sequences
- Construct distance matrix
- Construct phylogenetic tree based on the distance matrix
- Two ways to construct a tree based on a distance matrix
  - Clustering
  - Optimality

	a	b	c	d
a	0			
b	6	0		
c	7	3	0	
d	14	10	9	0



Clustering-Based Methods

- E.g., UPGMA and Neighbor-Joining
- Compute a tree based on a distance matrix starting from the most similar sequence pairs.
- A cluster is a set of taxa
- Interspecies distances translate into inter-cluster distances
- Clusters are repeatedly merged
- “Closest” clusters merged first
- Distances are recomputed after merging

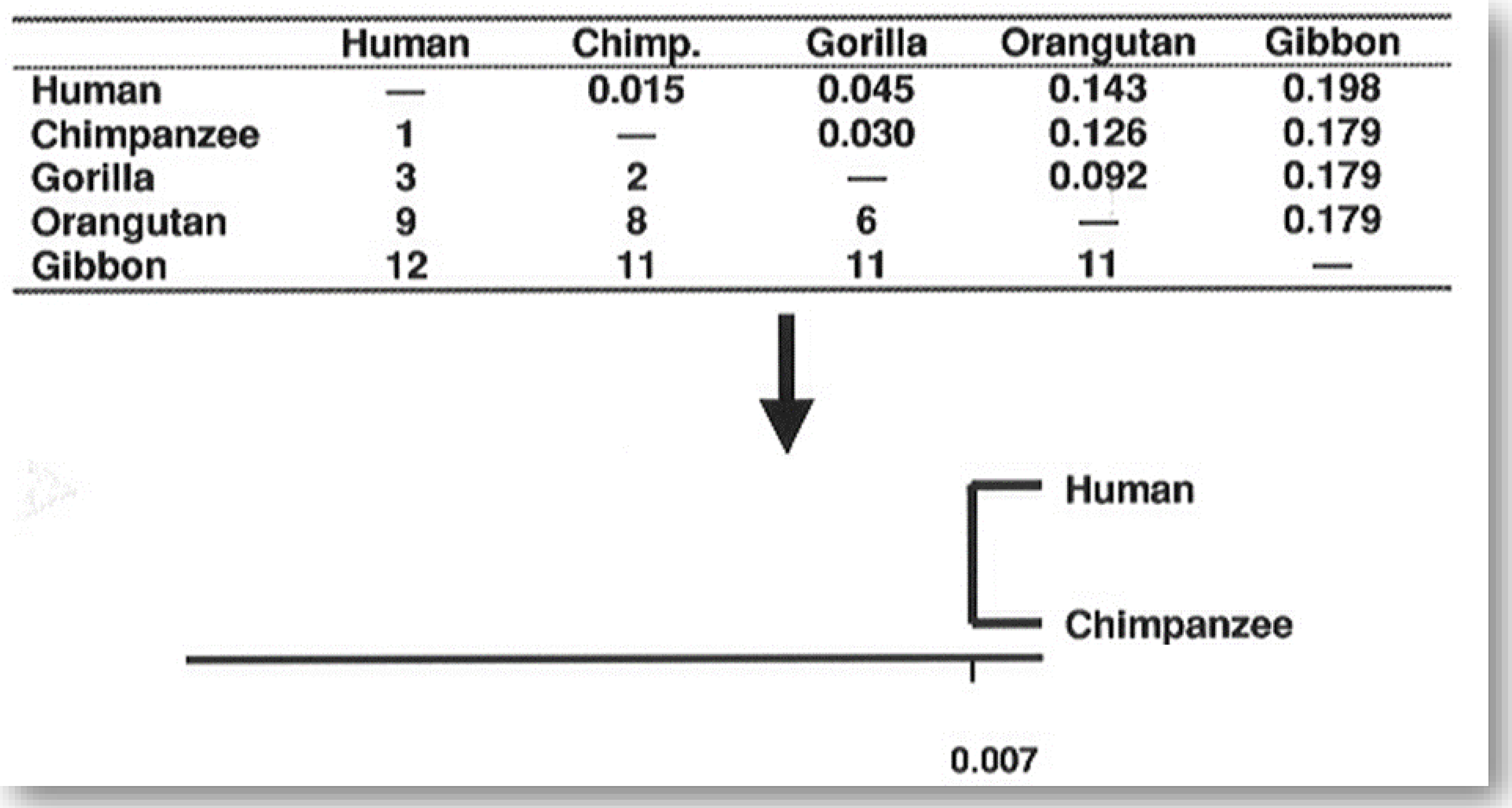
UPGMA

- UPGMA – Unweighted Pair Group Method Using Arithmetic Average
- The simplest clustering method which builds a tree by a sequential clustering method.
- Uses molecular clock assumption:
  - All taxa evolve at a constant rate and are equally distant from the root (**Ultrametric Tree**)
  - This assumption is usually wrong
    - Thus, UPGMA often produces erroneous tree topologies.
- So why use UPGMA?
  - Very fast

UPGMA Steps

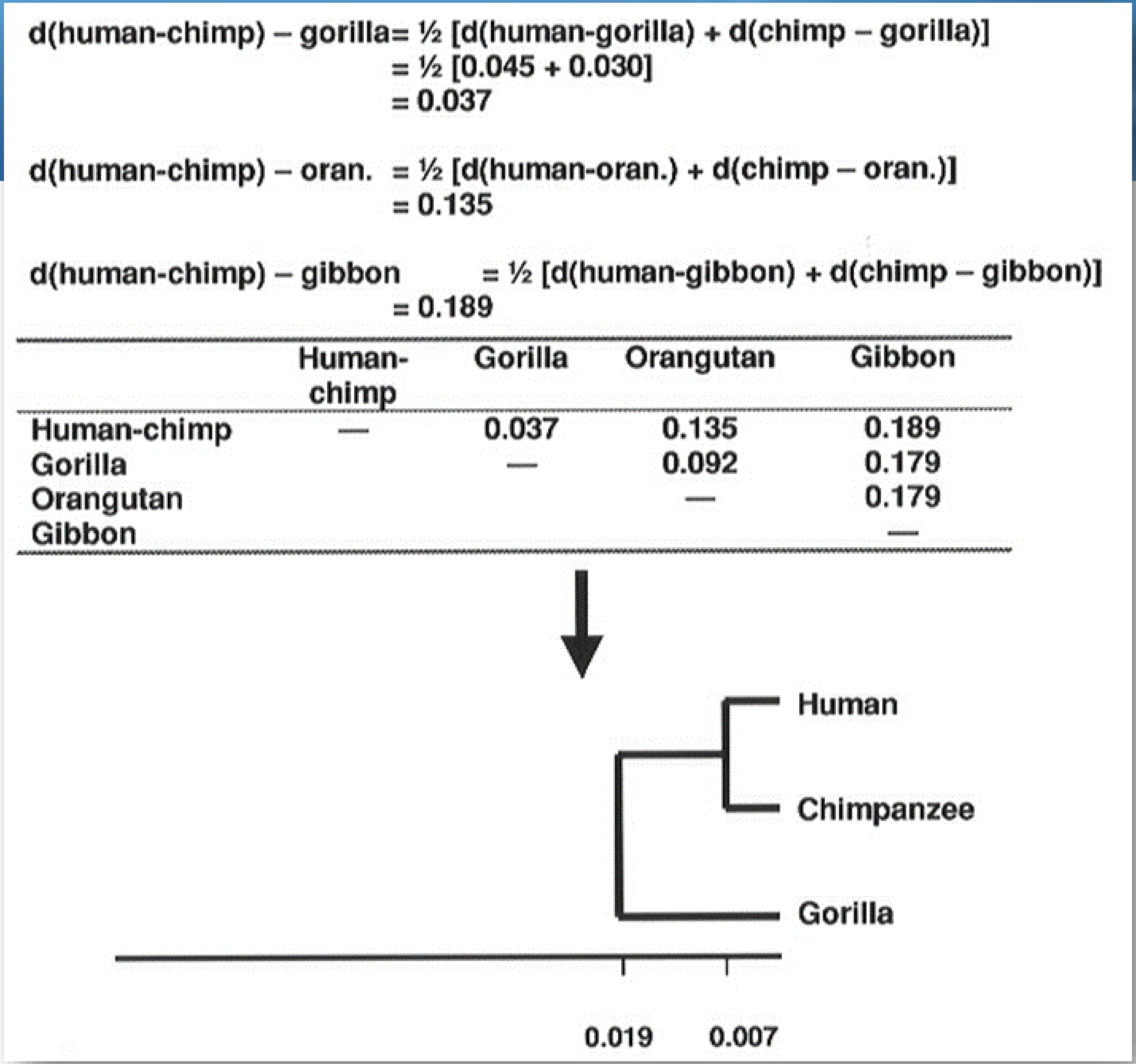
1. Given a distance matrix, it starts by grouping two taxa with the smallest pairwise distance.
   - A node is placed at the midpoint or half distance between them.
2. It then creates a reduced matrix by treating the new cluster as a single taxon.
3. The distances between this new composite taxon and all remaining taxa are calculated to create a reduced matrix.
4. The same grouping process is repeated and another newly reduced matrix is created.
5. The iteration continues until all taxa are placed on the tree.
   - The last taxon added is considered the outgroup producing a rooted tree.

UPGMA Example



	Human	Chimp.	Gorilla	Orangutan	Gibbon
Human	—	0.015	0.045	0.143	0.198
Chimpanzee	1	—	0.030	0.126	0.179
Gorilla	3	2	—	0.092	0.179
Orangutan	9	8	6	—	0.179
Gibbon	12	11	11	11	—

Step2



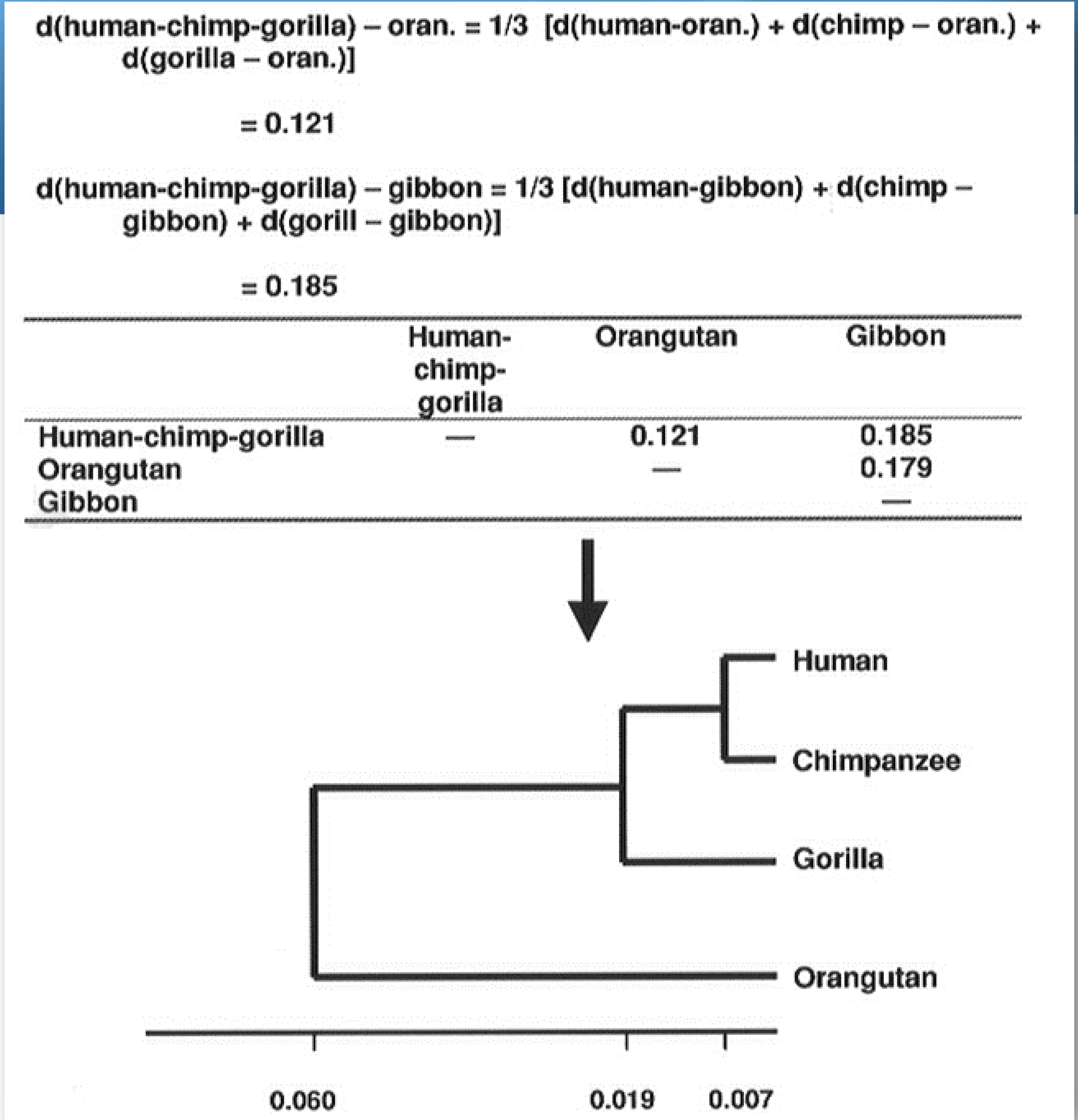
d(human-chimp) – gorilla= ½ [d(human-gorilla) + d(chimp – gorilla)]
= ½ [0.045 + 0.030]
= 0.037

d(human-chimp) – oran. = ½ [d(human-oran.) + d(chimp – oran.)]
= 0.135

d(human-chimp) – gibbon = ½ [d(human-gibbon) + d(chimp – gibbon)]
= 0.189

	Human-chimp	Gorilla	Orangutan	Gibbon
Human-chimp	—	0.037	0.135	0.189
Gorilla		—	0.092	0.179
Orangutan			—	0.179
Gibbon				—

Step3



d(human-chimp-gorilla) – oran. = 1/3 [d(human-oran.) + d(chimp – oran.) + d(gorilla – oran.)]

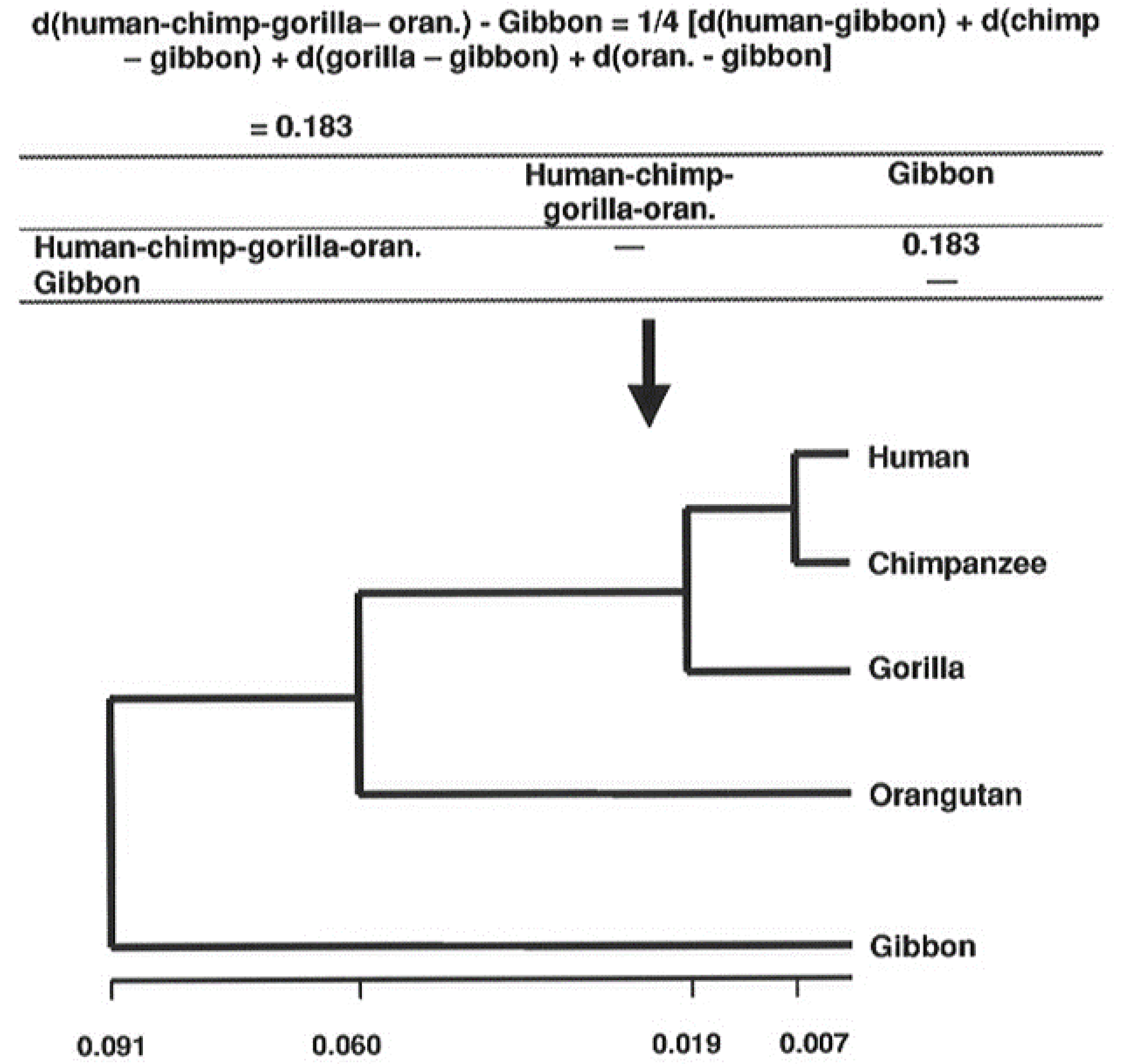
= 0.121

d(human-chimp-gorilla) – gibbon = 1/3 [d(human-gibbon) + d(chimp – gibbon) + d(gorill – gibbon)]

= 0.185

	Human-chimp-gorilla	Orangutan	Gibbon
Human-chimp-gorilla	—	0.121	0.185
Orangutan		—	0.179
Gibbon			—

Step4

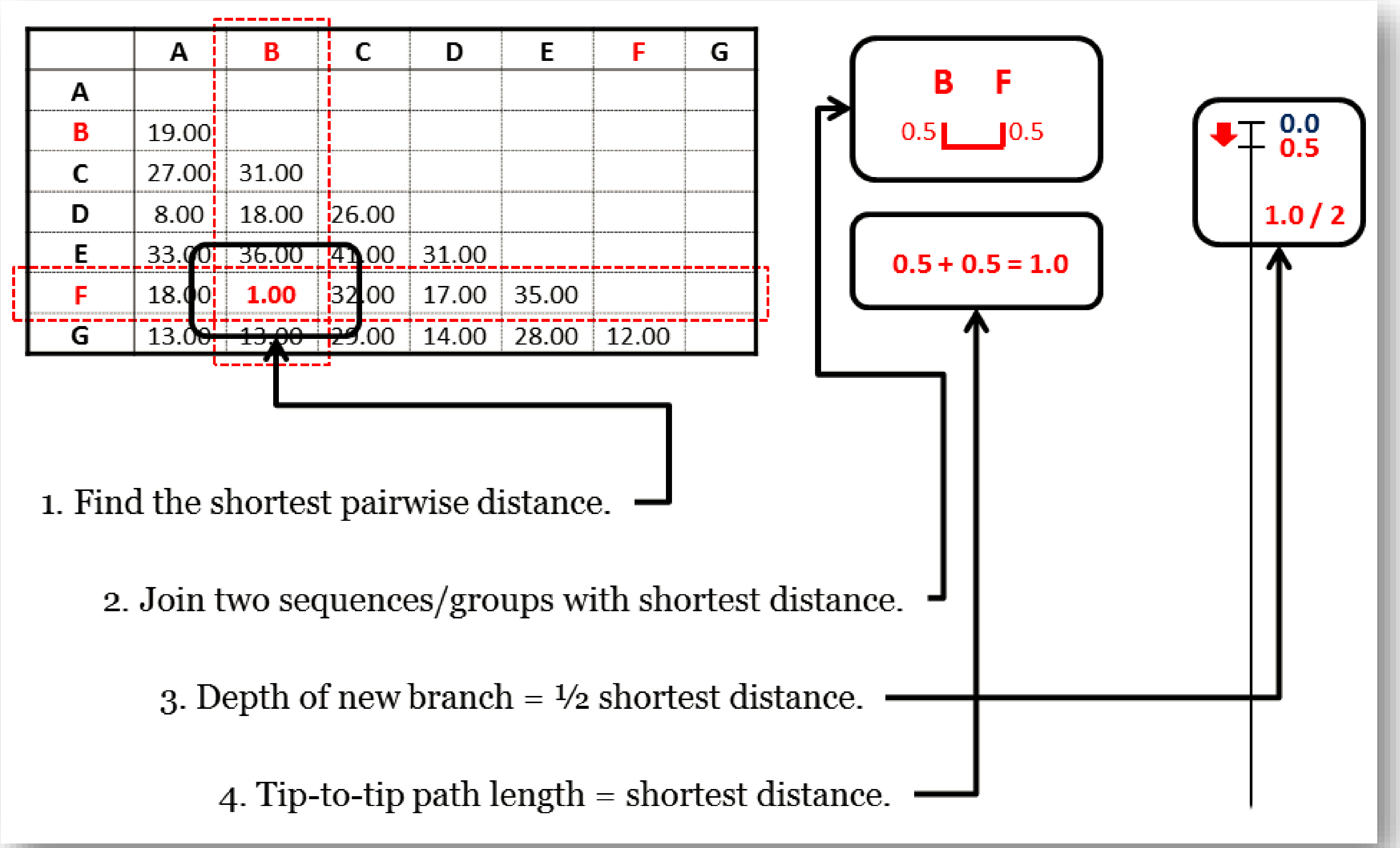


d(human-chimp-gorilla– oran.) - Gibbon = 1/4 [d(human-gibbon) + d(chimp – gibbon) + d(gorilla – gibbon) + d(oran. - gibbon]

= 0.183

	Human-chimp-gorilla-oran.	Gibbon
Human-chimp-gorilla-oran.	—	0.183
Gibbon		—

Second Example



	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

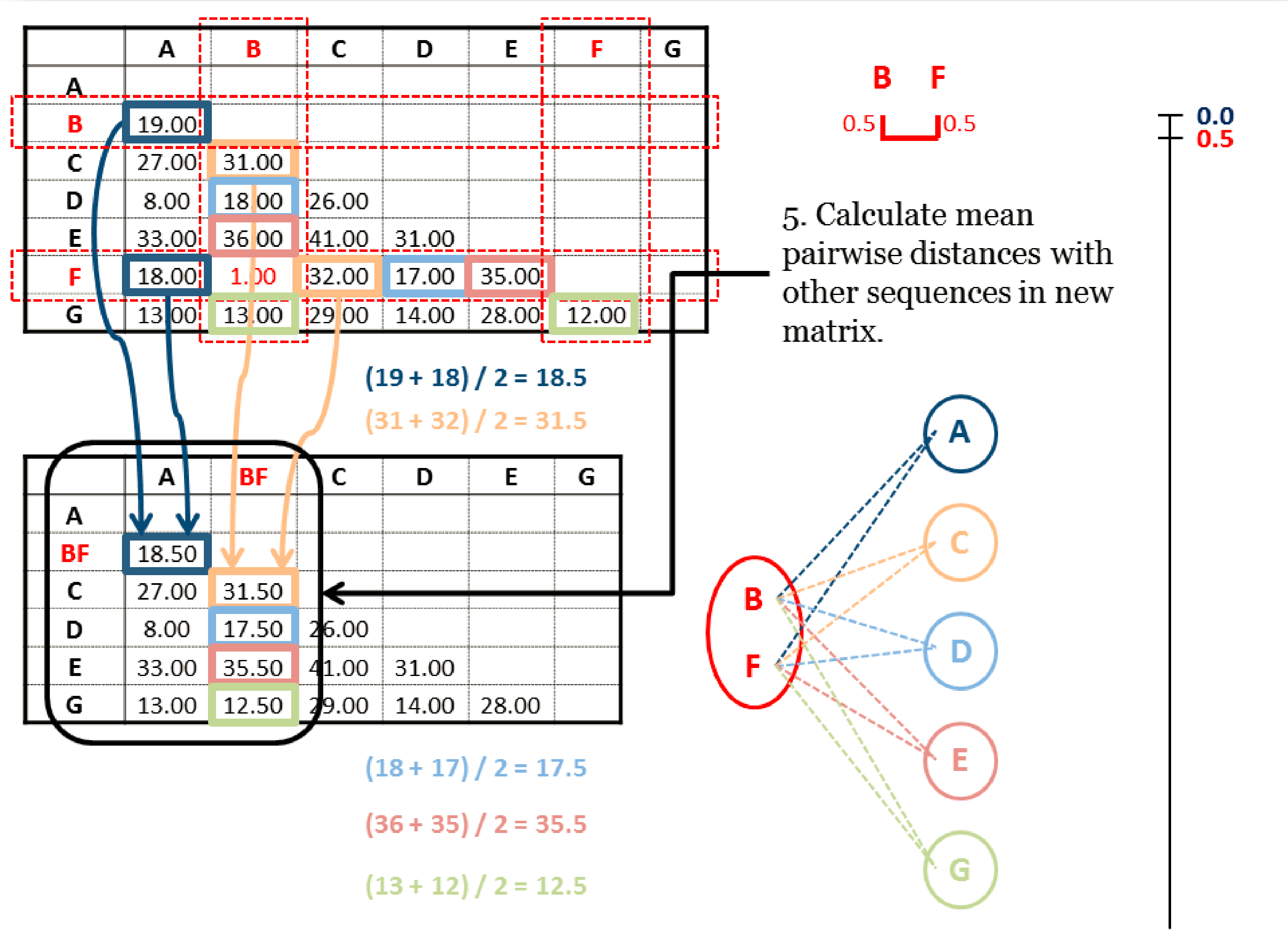
1. Find the shortest pairwise distance.

2. Join two sequences/groups with shortest distance.

3. Depth of new branch = ½ shortest distance.

4. Tip-to-tip path length = shortest distance.

Step2



	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

(19 + 18) / 2 = 18.5

(31 + 32) / 2 = 31.5

	A	BF	C	D	E	G
A						
BF	18.50					
C	27.00	31.50				
D	8.00	17.50	26.00			
E	33.00	35.50	41.00	31.00		
G	13.00	12.50	29.00	14.00	28.00	

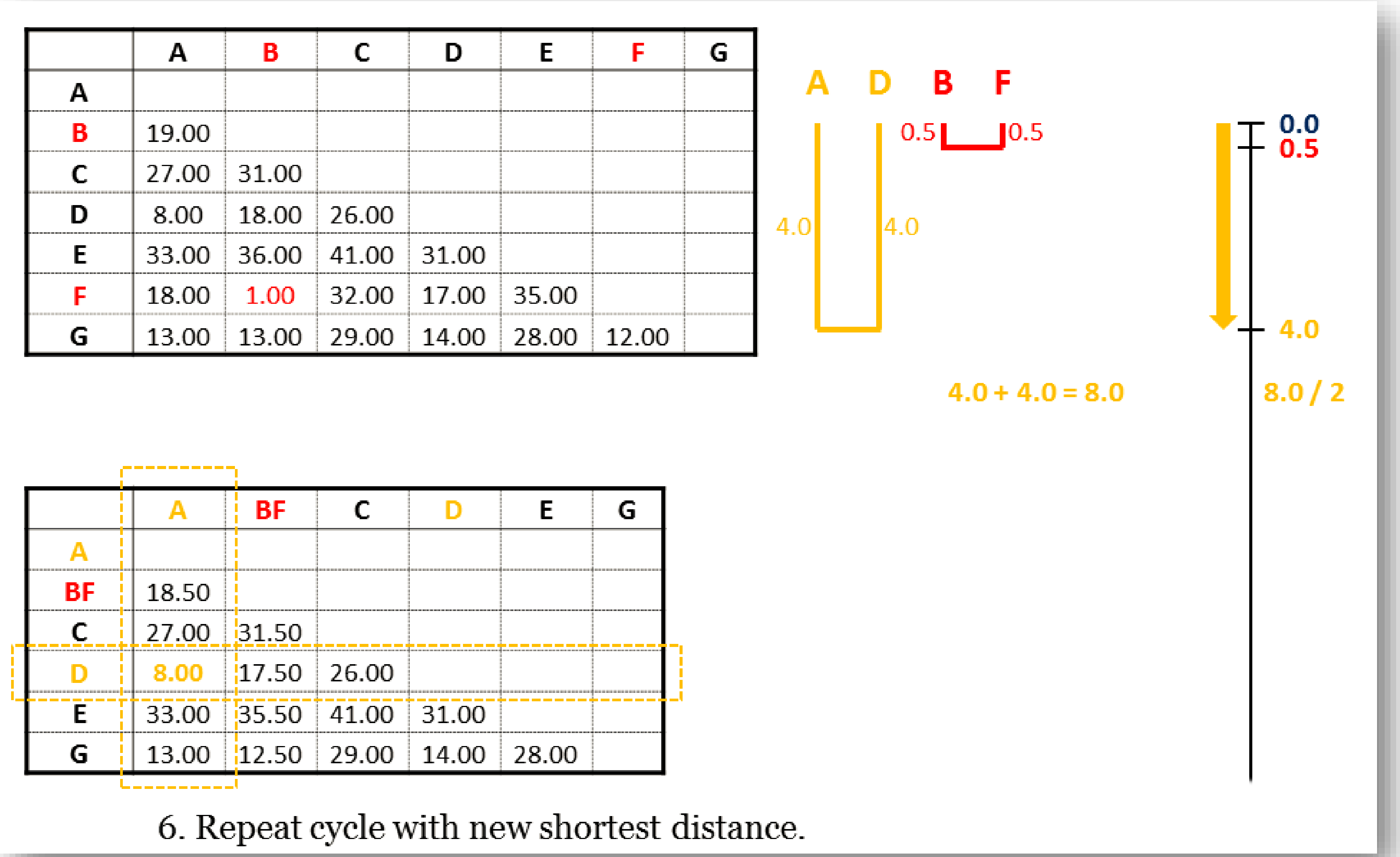
(18 + 17) / 2 = 17.5

(36 + 35) / 2 = 35.5

(13 + 12) / 2 = 12.5

5. Calculate mean pairwise distances with other sequences in new matrix.

Step3

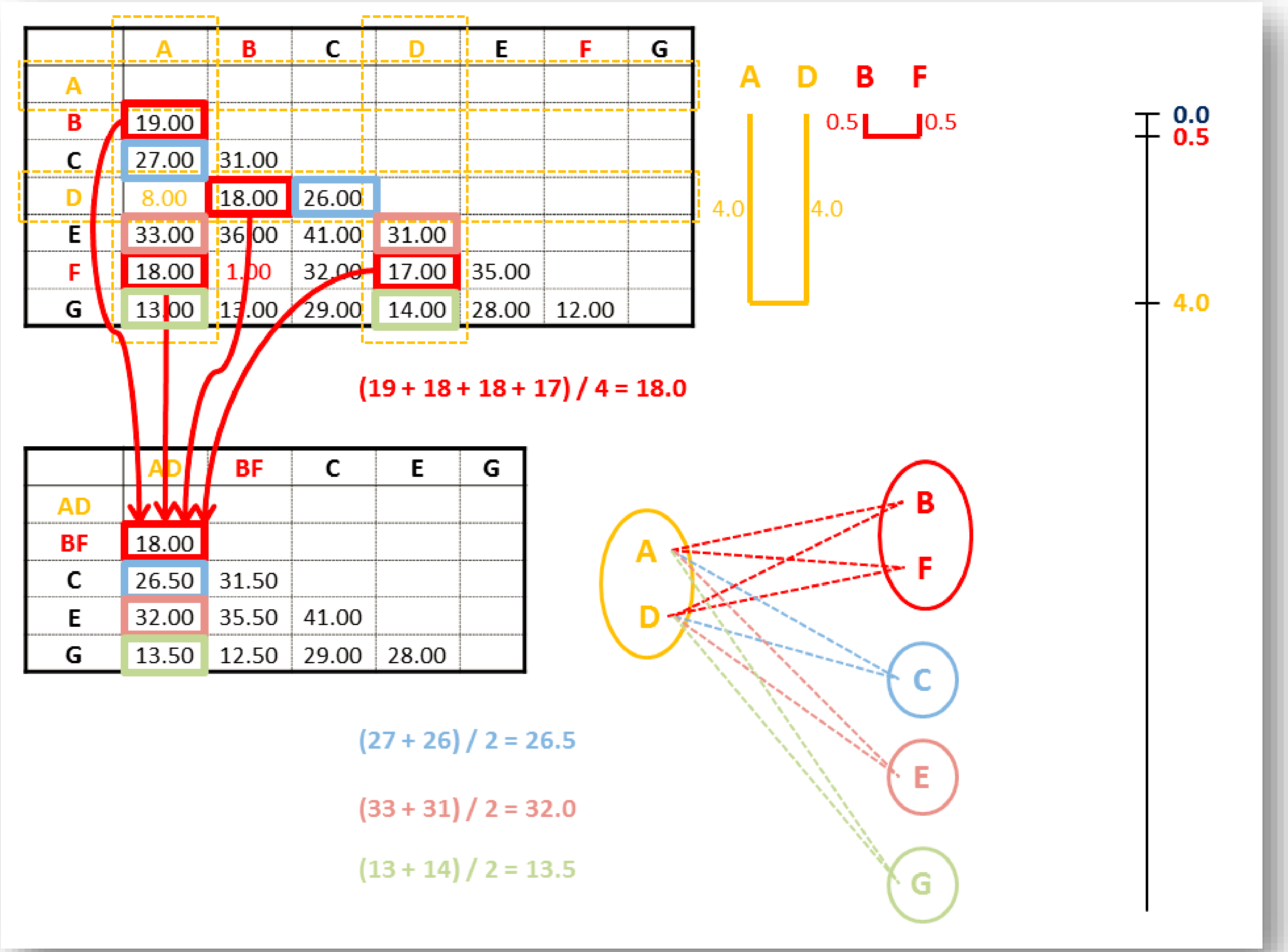


	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

	A	BF	C	D	E	G
A						
BF	18.50					
C	27.00	31.50				
D	8.00	17.50	26.00			
E	33.00	35.50	41.00	31.00		
G	13.00	12.50	29.00	14.00	28.00	

6. Repeat cycle with new shortest distance.

Step4



	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

(19 + 18 + 18 + 17) / 4 = 18.0

	AD	BF	C	E	G
AD					
BF	18.00				
C	26.50	31.50			
E	32.00	35.50	41.00		
G	13.50	12.50	29.00	28.00	

(27 + 26) / 2 = 26.5

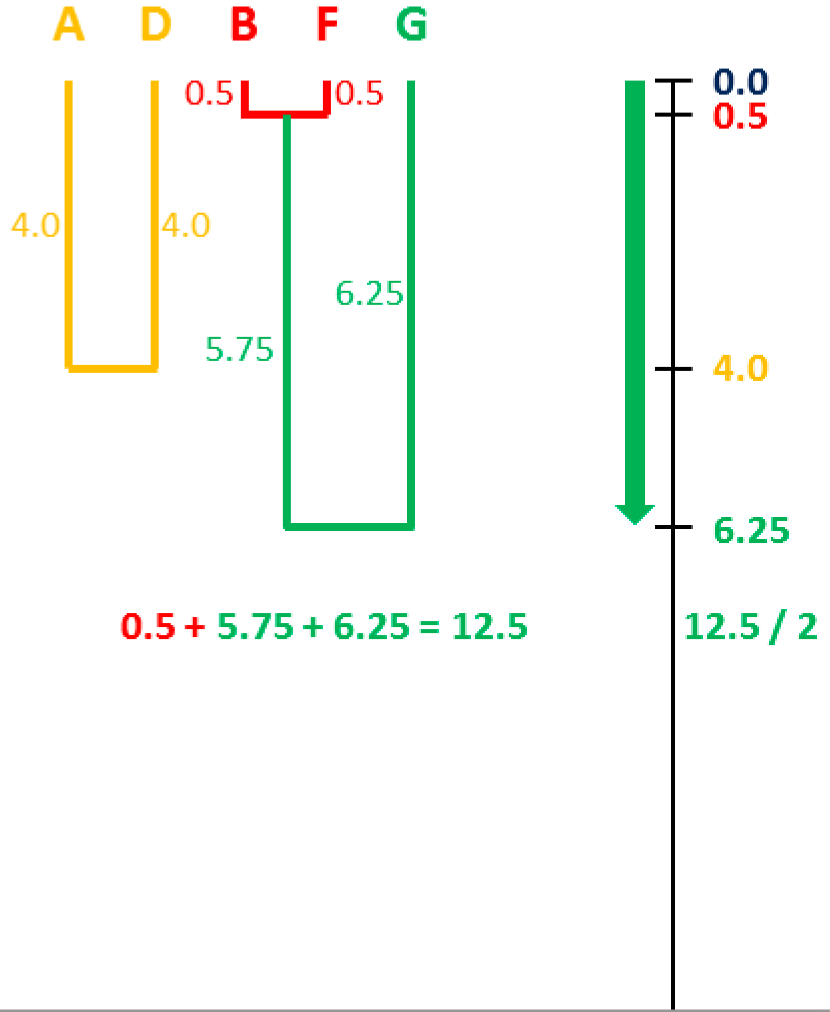
(33 + 31) / 2 = 32.0

(13 + 14) / 2 = 13.5

Step …

	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

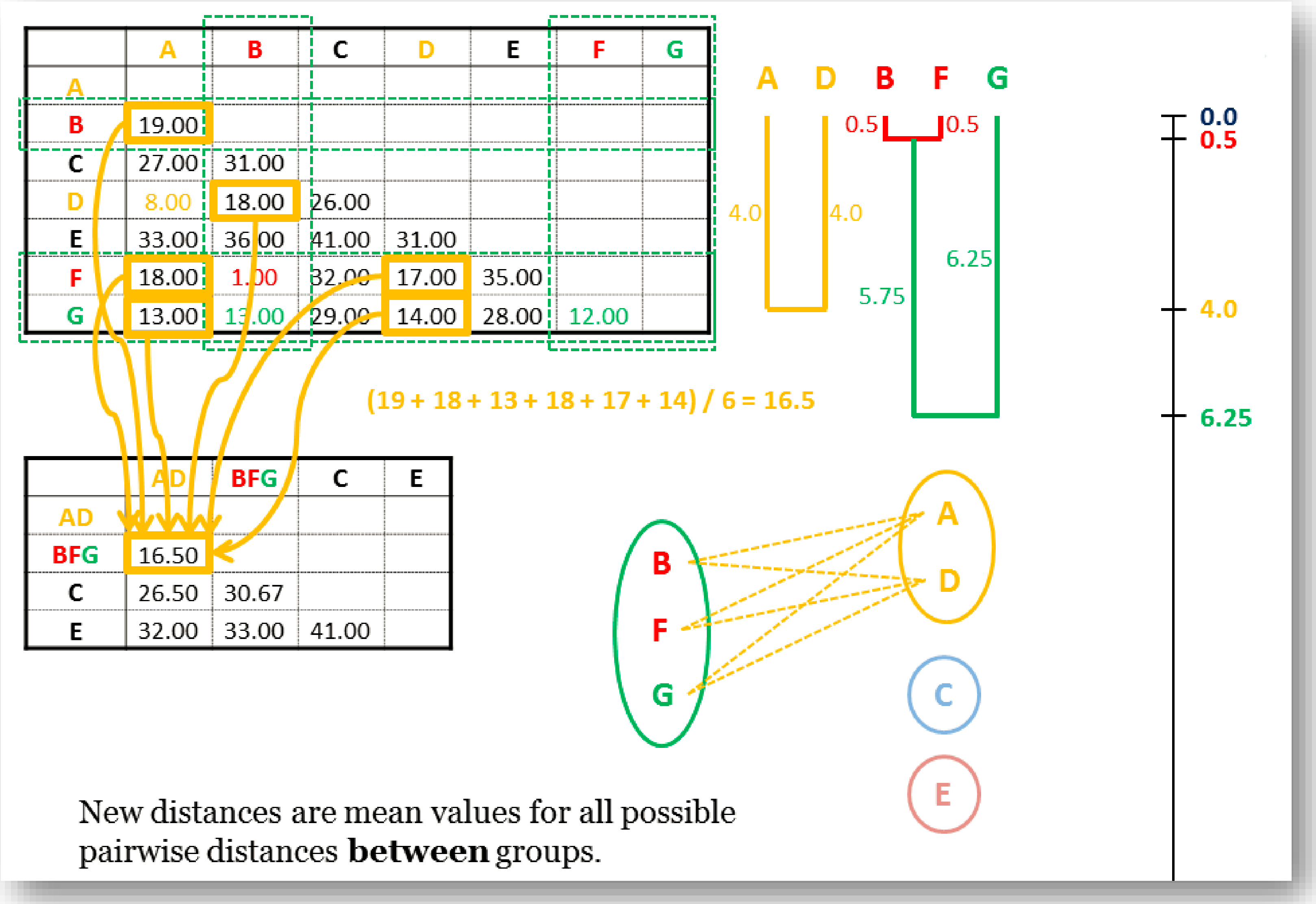
	AD	BF	C	E	G
AD					
BF	18.00				
C	26.50	31.50			
E	32.00	35.50	41.00		
G	13.50	**12.50**	29.00	28.00	



0.5 + 5.75 + 6.25 = 12.5

12.5 / 2

Step …



	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

(19 + 18 + 13 + 18 + 17 + 14) / 6 = 16.5

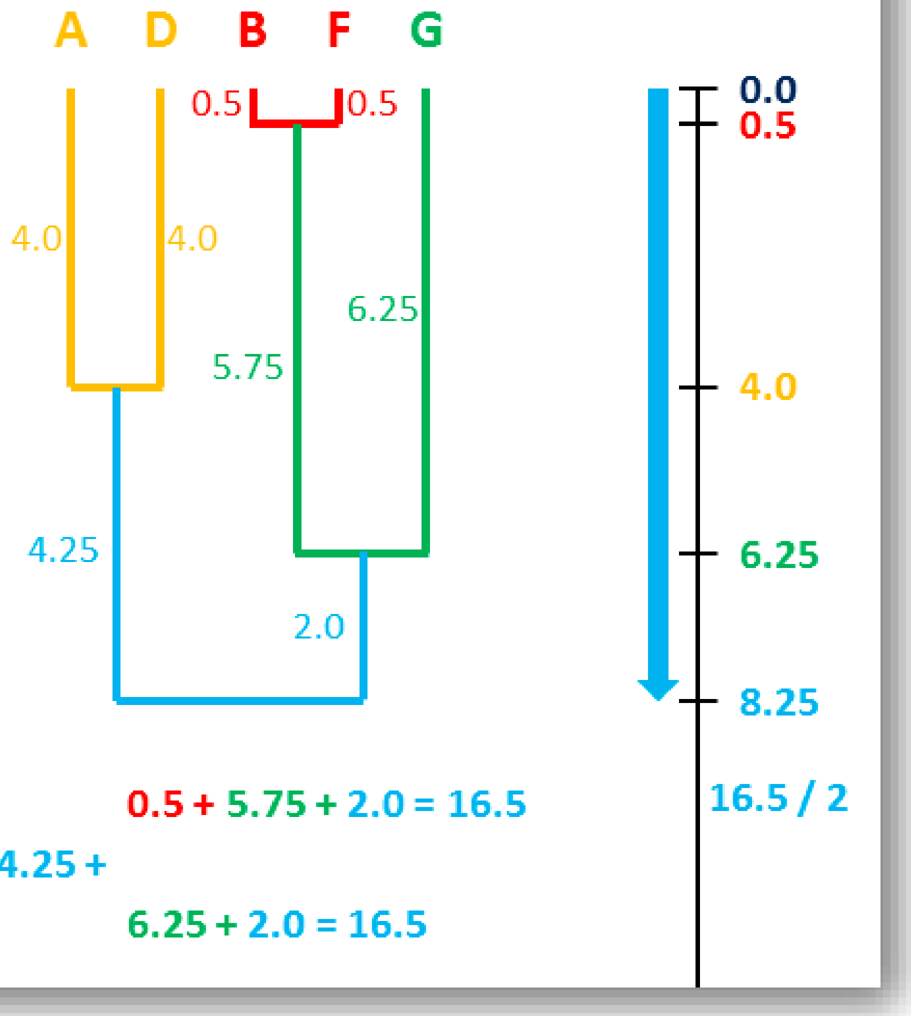
	AD	BFG	C	E
AD				
BFG	16.50			
C	26.50	30.67		
E	32.00	33.00	41.00	

New distances are mean values for all possible pairwise distances **between** groups.

Step …

	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

	AD	BFG	C	E
AD				
BFG	16.50			
C	26.50	30.67		
E	32.00	33.00	41.00	



0.5 + 5.75 + 2.0 = 16.5

4.0 + 4.25 +

6.25 + 2.0 = 16.5

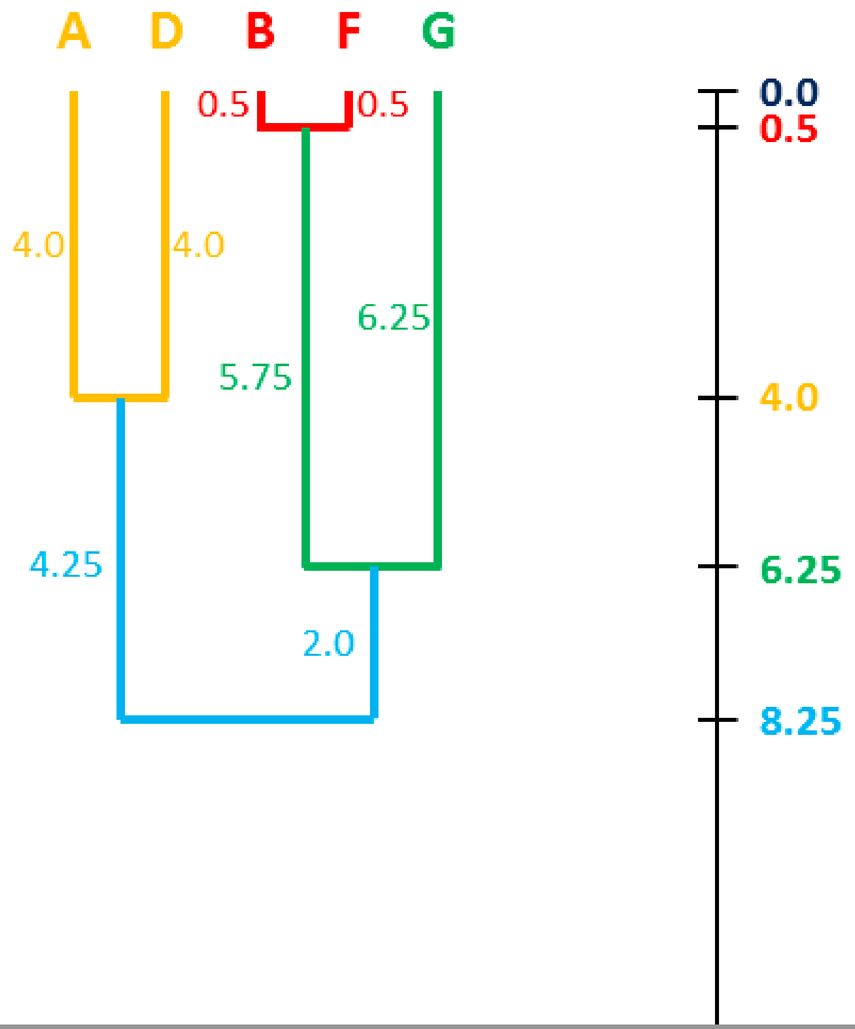
Step …

	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

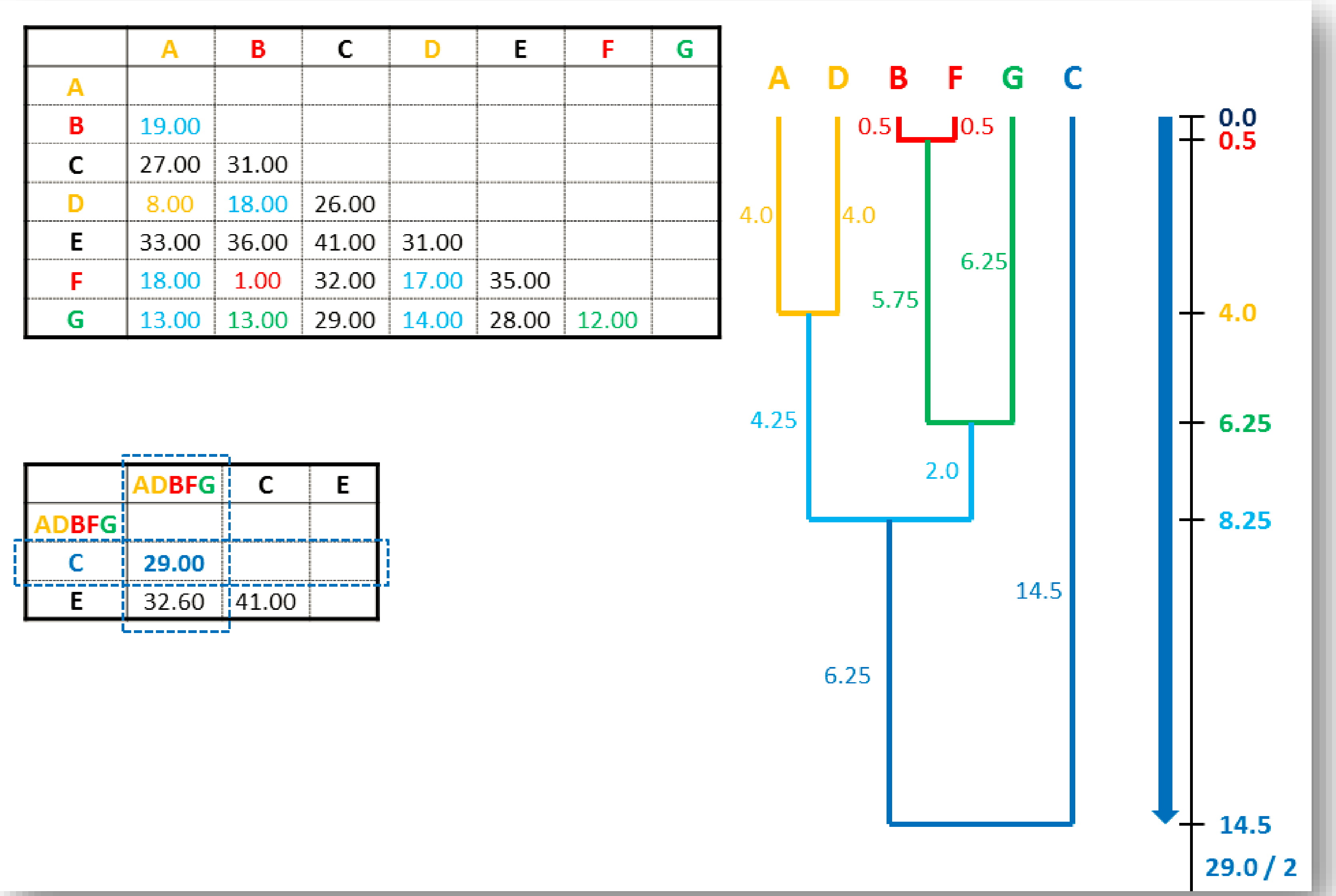
(27 + 31 + 26 + 32 + 29) / 5 = 29.00

	ADBFG	C	E
ADBFG			
C	29.00		
E	32.60	41.00	

(33 + 36 + 31 + 35 + 28) / 5 = 32.60



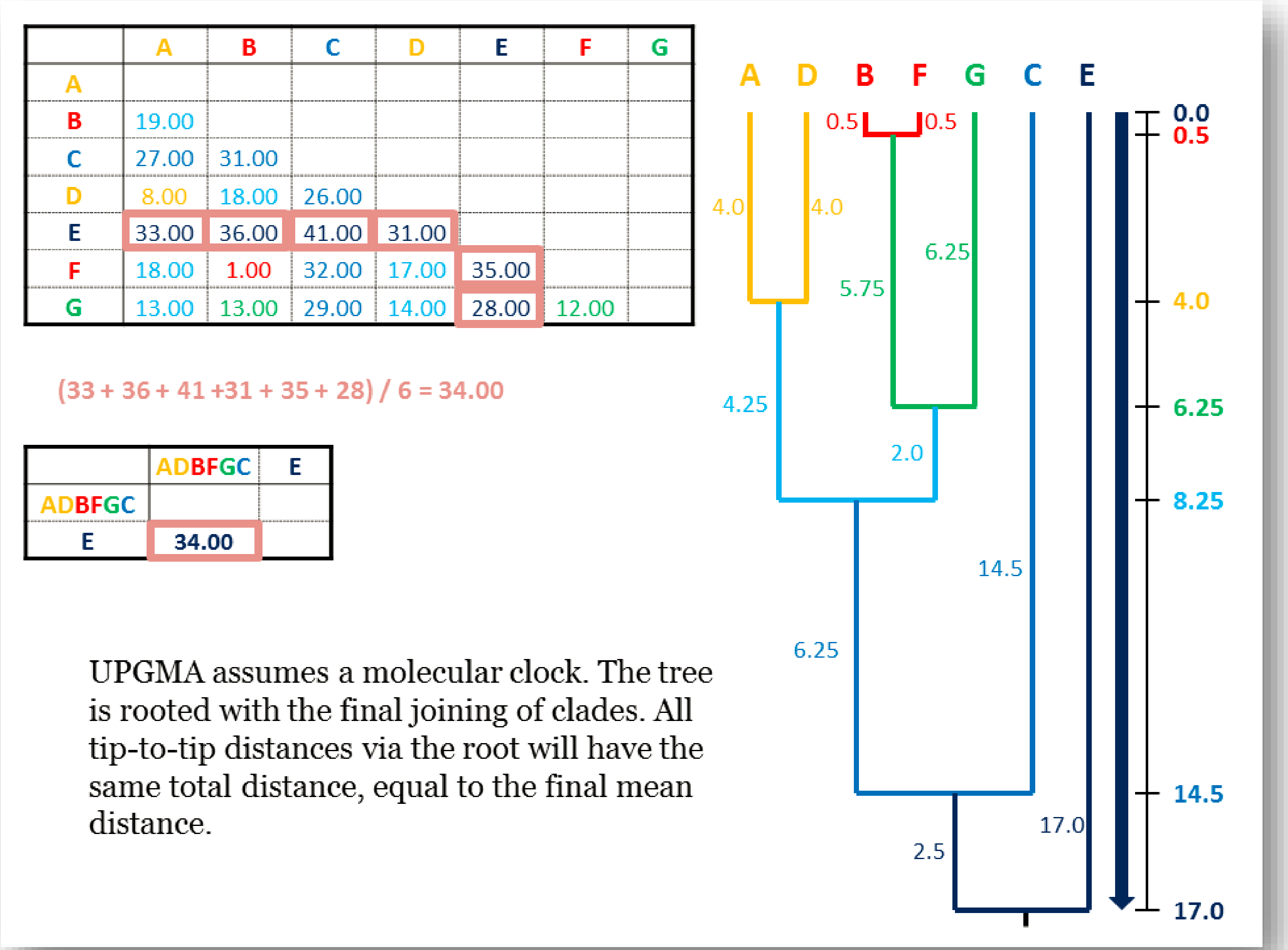
Step …



	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

	ADBFG	C	E
ADBFG			
C	**29.00**		
E	32.60	41.00	

Last Step



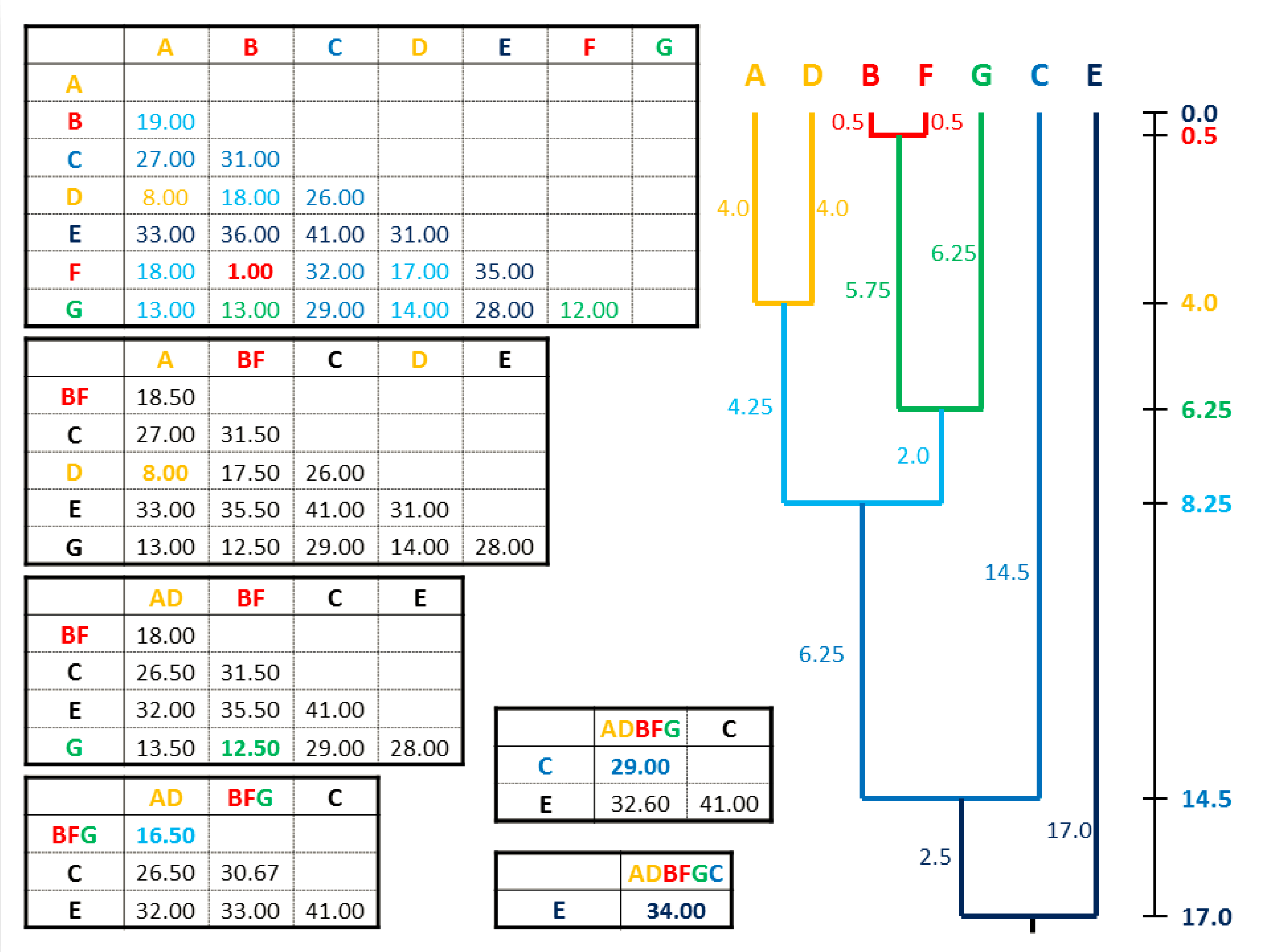
	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	1.00	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

(33 + 36 + 41 +31 + 35 + 28) / 6 = 34.00

	ADBFGC	E
ADBFGC		
E	34.00	

UPGMA assumes a molecular clock. The tree is rooted with the final joining of clades. All tip-to-tip distances via the root will have the same total distance, equal to the final mean distance.

All Steps at a Glance



	A	B	C	D	E	F	G
A							
B	19.00						
C	27.00	31.00					
D	8.00	18.00	26.00				
E	33.00	36.00	41.00	31.00			
F	18.00	**1.00**	32.00	17.00	35.00		
G	13.00	13.00	29.00	14.00	28.00	12.00	

	A	BF	C	D	E
BF	18.50				
C	27.00	31.50			
D	**8.00**	17.50	26.00		
E	33.00	35.50	41.00	31.00	
G	13.00	12.50	29.00	14.00	28.00

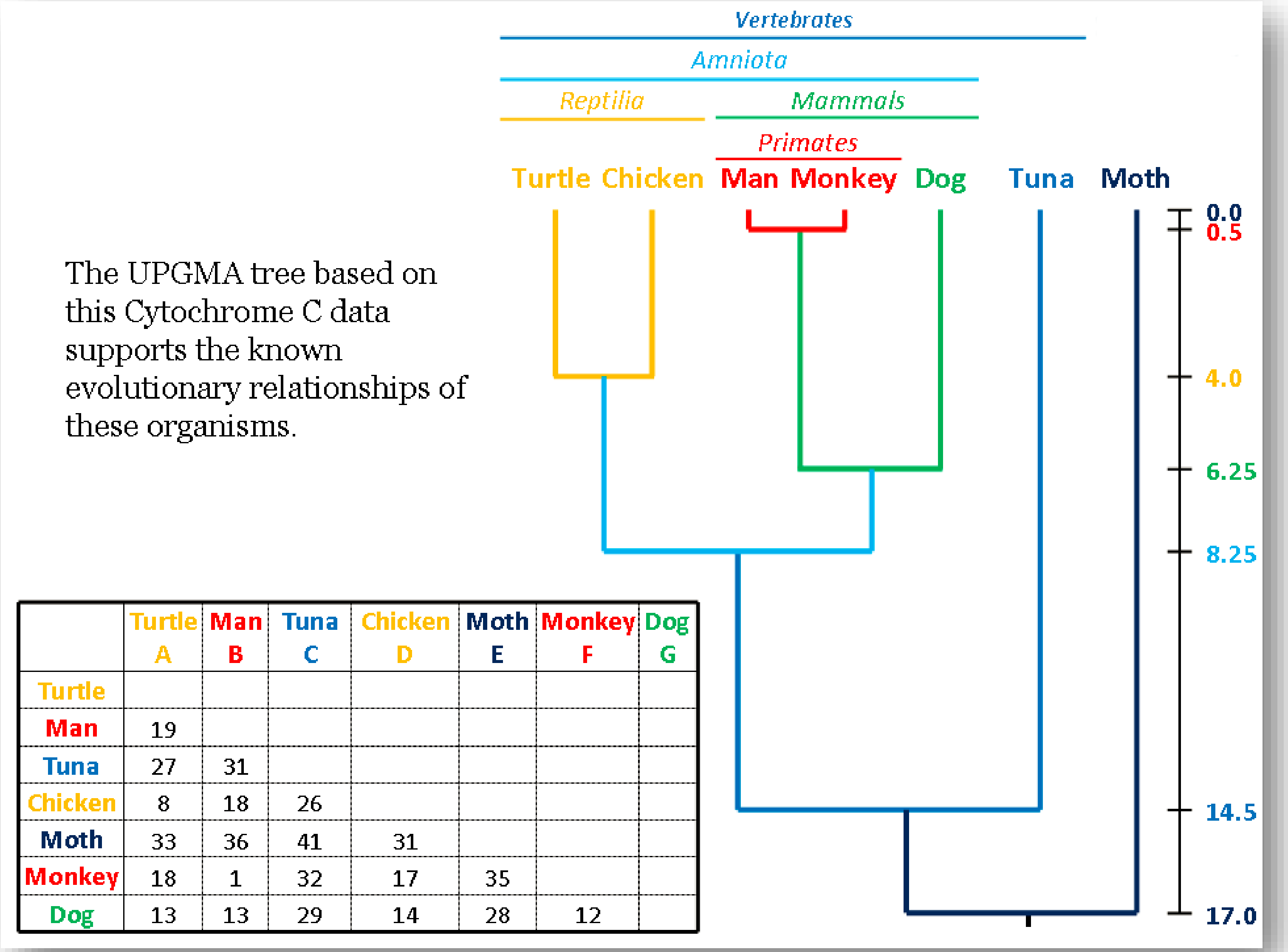
	AD	BF	C	E
BF	18.00			
C	26.50	31.50		
E	32.00	35.50	41.00	
G	13.50	**12.50**	29.00	28.00

	AD	BFG	C
BFG	**16.50**		
C	26.50	30.67	
E	32.00	33.00	41.00

	ADBFG	C
C	**29.00**	
E	32.60	41.00

	ADBFGC
E	**34.00**

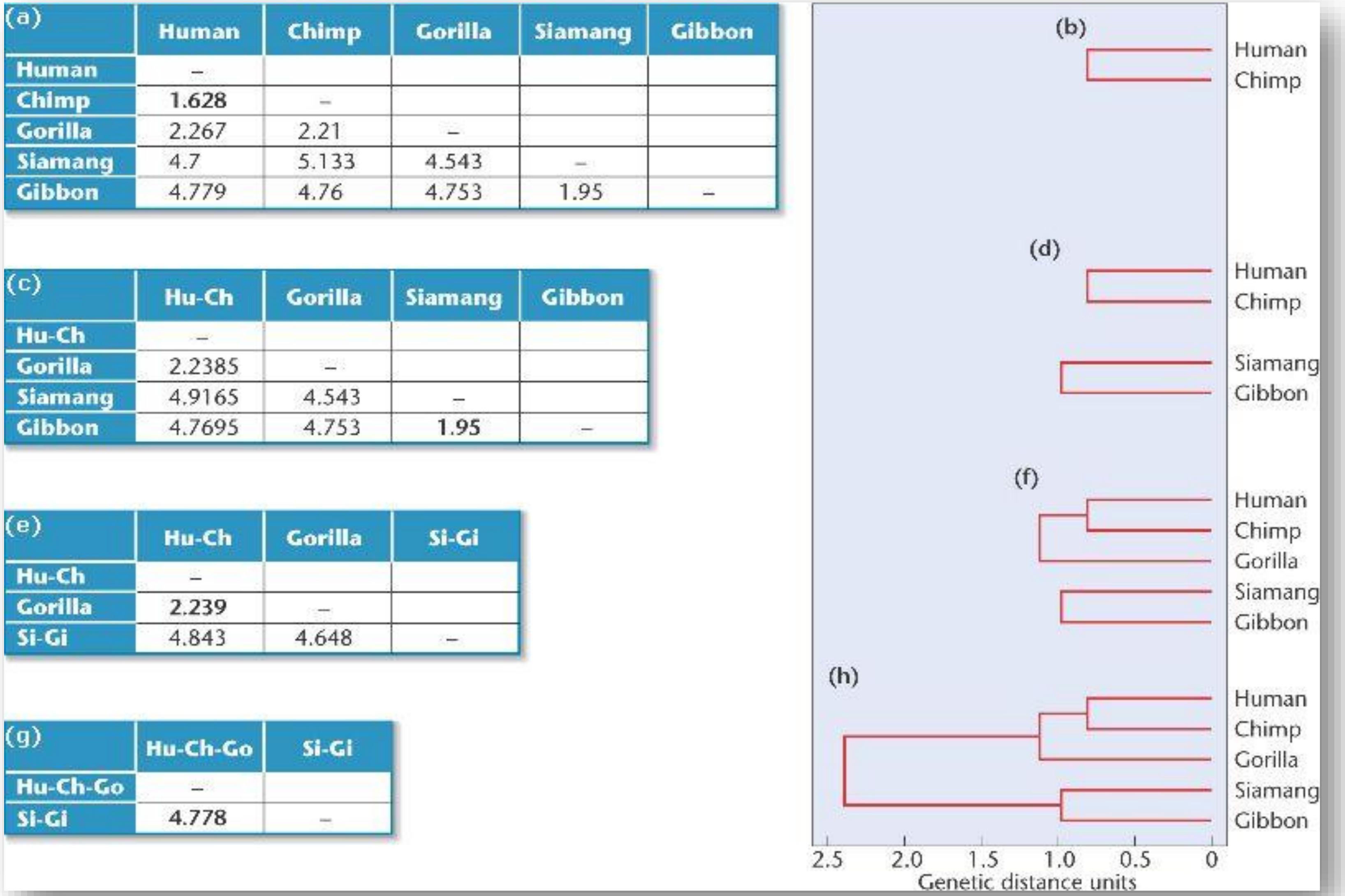
Conclusions



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

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

Last Example: All Steps at a Glance



(a)	Human	Chimp	Gorilla	Siamang	Gibbon
Human	–				
Chimp	**1.628**	–			
Gorilla	2.267	2.21	–		
Siamang	4.7	5.133	4.543	–	
Gibbon	4.779	4.76	4.753	1.95	–

(c)	Hu-Ch	Gorilla	Siamang	Gibbon
Hu-Ch	–			
Gorilla	2.2385	–		
Siamang	4.9165	4.543	–	
Gibbon	4.7695	4.753	**1.95**	–

(e)	Hu-Ch	Gorilla	Si-Gi
Hu-Ch	–		
Gorilla	**2.239**	–	
Si-Gi	4.843	4.648	–

(g)	Hu-Ch-Go	Si-Gi
Hu-Ch-Go	–	
Si-Gi	**4.778**	–

Neighbor Joining (NJ)

- The UPGMA method uses unweighted distances and assumes that all taxa have constant evolutionary rates.
- NJ Idea: Find a pair of taxa that are close to each other but far from other taxa
  - Implicitly finds a pair of neighboring taxa
- Similar to UPGMA, NJ builds a tree by using stepwise reduced distance matrices.
  - NJ does not assume the taxa to be equidistant from the root.
- No molecular clock assumption

Neighbor Joining (Cont.)

- NJ corrects for unequal evolutionary rates between sequences by using a *conversion step*.
- The conversion step requires calculation of “r-values” and “transformed r-values”
- The r-value for a sequence is the sum of the distances between sequence *i* and all other sequences:

$$r_i = \sum d_{ij}$$

- The transformed r-value for a sequence is:

$$r_i' = \frac{r_i}{n-2}$$

where n is the number of taxa

- Transformed r-values are used to determine the distance of a taxon to the nearest node.

Neighbor Joining (Cont.)

- The *converted distance* between two sequences is:

$$d'_{ij} = d_{ij} - \frac{1}{2}\left(r_i + r_j\right)$$

d_{ij} is the actual evolutionary distance.

- These converted distances are used in building the tree
- The final equation we need is for computing the distance from a new cluster to each taxa. Assume taxa i and j were merged into a cluster u. The distance from taxa i to cluster u is:

$$d_{iu} = \frac{\left(d_{ij} + \left(r'_i - r'_j\right)\right)}{2}$$

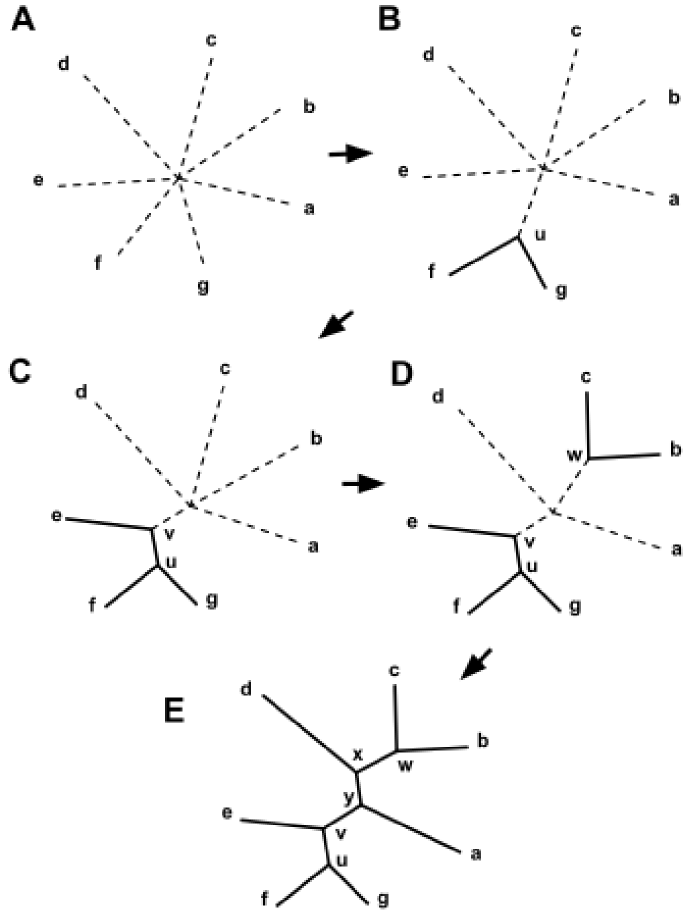
Neighbor Joining Example

	A	B	C
B	0.40		
C	0.35	0.45	
D	0.60	0.70	0.55

- Initialize tree into a star shape with all taxa connected to the center
- Step 1: Compute r-values and transformed r-values for all taxa

$$r_A = d_{AB} + d_{AC} + d_{AD} = 0.4 + 0.35 + 0.6 = 1.35$$

$$r'_A = \frac{r_A}{4-2} = \frac{1.35}{2} = 0.675$$



NJ Example (Cont.)

- Step 2: Compute converted distances

$$d'_{AB} = d_{AB} - \frac{1}{2}(r_A + r_B)$$

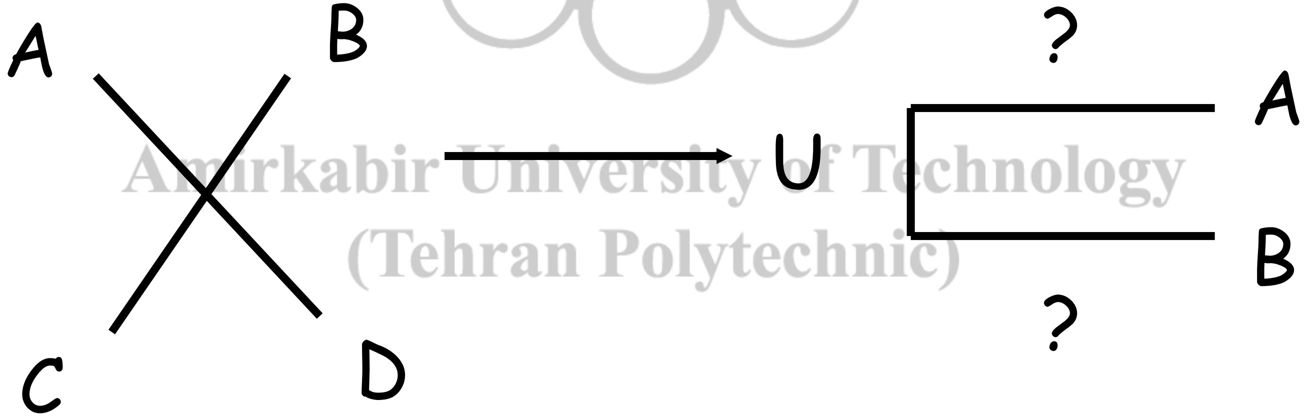
$$= 0.4 - 0.5 \times (1.35 + 1.55) = -1.05$$

- Step 3: Fill out converted distance matrix

	A	B	C
B	-1.05		
C	-1	-1	
D	-1	-1	-1.05

NJ Example (Cont.)

- Step 4: Create a node by merging closest taxa
- In this example, the distance between A and B is the same as the distance between C and D
- We can pick either pair to start with
- Let’s pick A and B and create a node called U

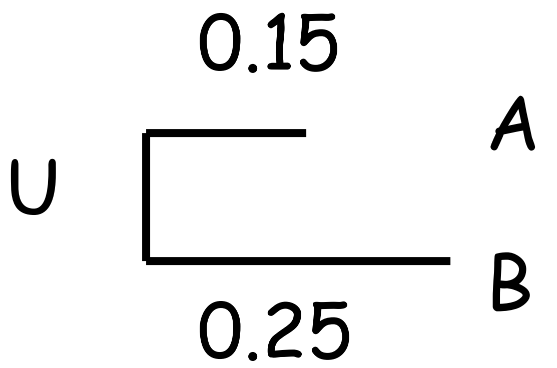


NJ Example (Cont.)

- Step 5: Compute branch lengths
- Use the equation for computing the distance from a taxa to a node

$$d_{AU} = \frac{\left(d_{AB} + (r'_A - r'_B)\right)}{2}$$

$$= \frac{\left(0.4 + (0.675 - 0.775)\right)}{2} = 0.15$$



NJ Example (Cont.)

- Step 6: Construct reduced distance matrix by computing converted distances from each taxa to the new node U
- Same as UPGMA, we simply calculated the average

$$d_{CU} = \frac{\left((d_{AC} - d_{UA}) + (d_{BC} - d_{UB})\right)}{2}$$

$$= \frac{\left((0.35 - 0.15) + (0.45 - 0.25)\right)}{2} = 0.2$$

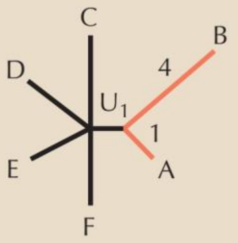
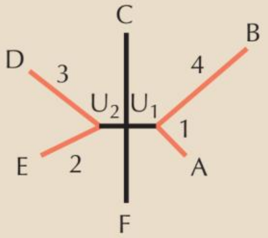
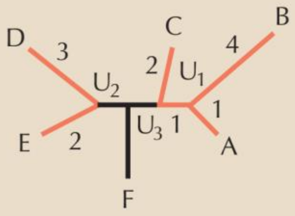
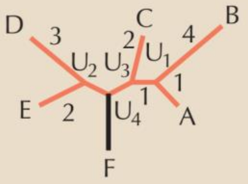
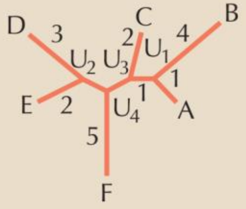
The reduced distance matrix:

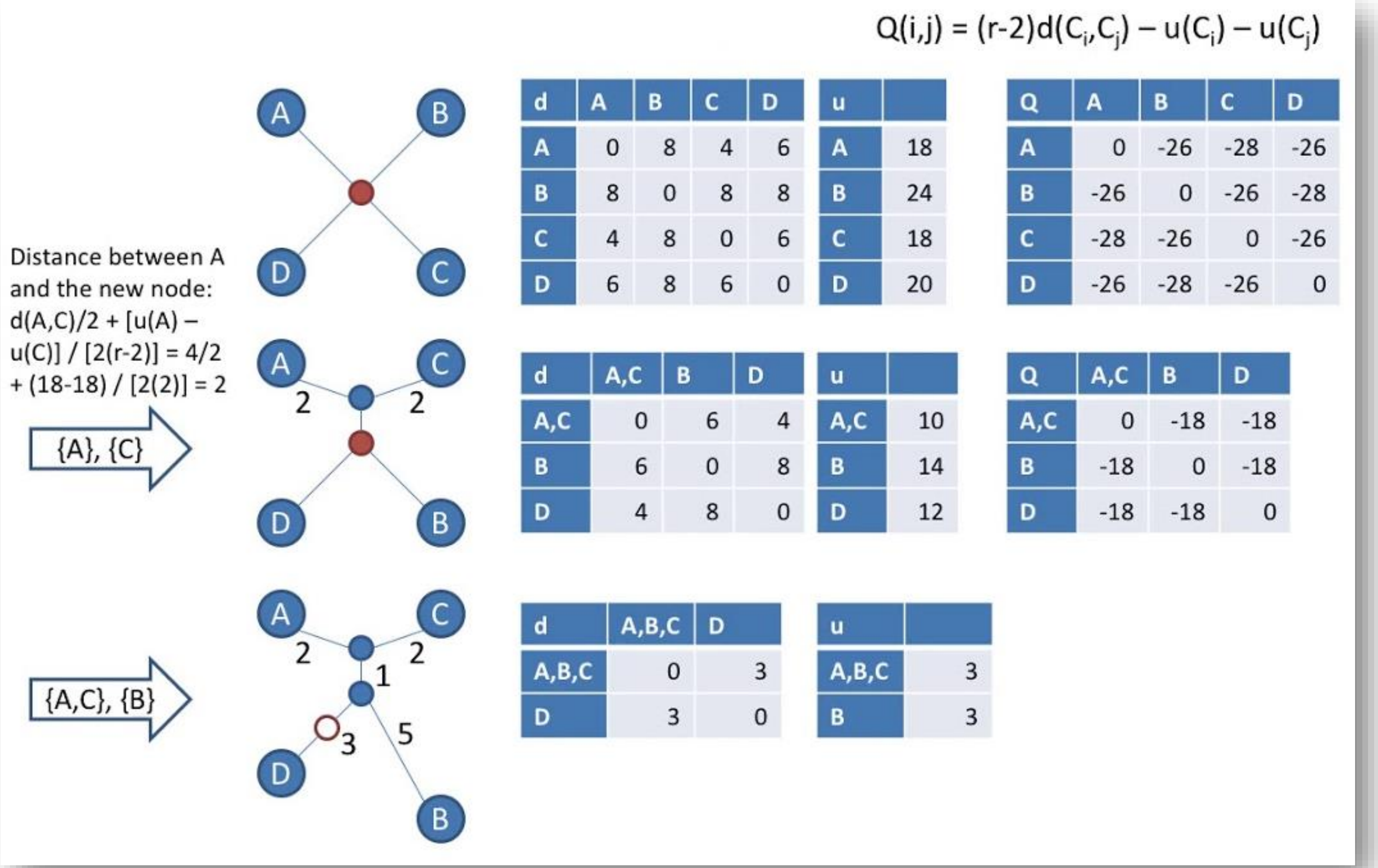
	U	C
C	0.20	
D	0.45	0.55

- From here, we go back to step 1
- Continue until all taxa have been decomposed from the star tree.

Note

- For NJ, different equations are used in reference books.
  - Also, notations differ from our text-book.
- For the next two examples just flow the algorithm and try to recognize similarities.

	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
Distance matrix	$\begin{array}{cccccc} & A & B & C & D & E \\ B & 5 \\ C & 4 & 7 \\ D & 7 & 10 & 7 \\ E & 6 & 9 & 6 & 5 \\ F & 8 & 11 & 8 & 9 & 8 \end{array}$	$\begin{array}{ccccc} & U_1 & C & D & E \\ C & 3 \\ D & 6 & 7 \\ E & 5 & 6 & 5 \\ F & 7 & 8 & 9 & 8 \end{array}$	$\begin{array}{cccc} & U_1 & C & U_2 \\ C & 3 \\ U_2 & 3 & 4 \\ F & 7 & 8 & 6 \end{array}$	$\begin{array}{ccc} & U_2 & U_3 \\ U_3 & 2 \\ F & 6 & 6 \end{array}$	$\begin{array}{cc} & U_4 \\ F & 5 \end{array}$
Step 1					
S calculations $S_x =$ (sum all D_x)/$(N-2)$, where *N* is the # of OTUs in the set.	$S_A = (5+4+7+6+8)/4 = 7.5$ $S_B = (5+7+10+9+11)/4 = 10.5$ $S_C = (4+7+7+6+8)/4 = 8$ $S_D = (7+10+7+5+9)/4 = 9.5$ $S_E = (6+9+6+5+8)/4 = 8.5$ $S_F = (8+11+8+9+8)/4 = 11$	$S_{U_1} = (3+6+5+7)/3 = 7$ $S_C = (3+7+6{=}8)/3 = 8$ $S_D = (6+7+5+9)/3 = 9$ $S_E = (5+6+5+8)/3 = 8$ $S_F = (7+8+9+8)/3 = 10.6$	$S_{U_1} = (3+3+7)/2 = 6.5$ $S_C = (3+4+8)/2 = 7.5$ $S_{U_2} = (3+4+6)/2 = 6.5$ $S_F = (7+8+6)/2 = 10.5$	$S_{U_2} = (2+6)/1 = 8$ $S_{U_3} = (2+6)/1 = 8$ $S_F = (6+6)/1 = 12$	Because $N-2=0$, we cannot do this calculation.
Step 2					
Calculate pair with smallest (*M*), where $M_{ij} = D_{ij} - S_i - S_j$.	Smallest are $M_{AB} = 5 - 7.5 - 10.5 = -13$ $M_{DE} = 5 - 9.5 - 8.5 = -13$ Choose one of these (AB here).	Smallest is $M_{CU_1} = 3 - 7 - 8 = -12$ $M_{DE} = 5 - 9 - 8 = -12$ Choose one of these (DE here).	Smallest is $M_{CU_1} = 3 - 6.5 - 7.5 = -11$	Smallest is $M_{U_2F} = 6 - 8 - 12 = -14$ $M_{U_3F} = 6 - 8 - 12 = -14$ $M_{U_2U_3} = 2 - 8 - 8 = -14$ Choose one of these ($M_{U_2U_3}$ here).	
Step 3					
Create a node (U) that joins pair with lowest M_{ij} such that $S_{IU} = D_{ij}/2 + (S_i - S_j)/2$.	U_1 joins A and B: $S_{AU_1} = D_{AB}/2 + (S_A - S_B)/2 = 1$ $S_{BU_1} = D_{AB}/2 + (S_B - S_A)/2 = 4$	U_2 joins D and E: $S_{DU_2} = D_{DE}/2 + (S_D - S_E)2 = 3$ $S_{EU_2} = D_{DE}/2 + (S_E - S_D)/2 = 2$	U_3 joins C and U_1: $S_{CU_3} = D_{CU_1}/2 + (S_C - S_{U_1})/2 = 2$ $S_{U_1U_3} = D_{CU_1}/2 + (S_{U1} - S_C)/2 = 1$	U_4 joins U_2 and U_3: $S_{U_2U_4} = D_{U_2U_3}/2 + (S_{U2} - S_{U3})/2 = 1$ $S_{U_3U_4} = D_{U_2U_3}/2 + (S_{U3} - S_{U2})/2 = 1$.	For last pair, connect U_4 and F with branch length = 5.
Step 4					
Join *i* and *j* according to *S* above and make all other taxa in form of a star. Branches in black are of unknown length. Branches in red are of known length.		 C D B 3 4 U₂ U₁ 1 E 2 A F	 D C B 3 4 2 U₂ U₁ 1 U₃ 1 E 2 A F	 D 3 C 2 4 B U₂ U₃ U₁ 1 1 E 2 U₄ A F	 D 3 C 2 4 B U₂ U₃ U₁ 1 1 E 2 U₄ A 5 F
Step 5					**Comments**
Calculate new distance matrix of all other taxa to U with $D_{xU} = D_{ix} + D_{jx} - D_{ij}$, where *i* and *j* are those selected from above.					Note this is the same tree we started with (drawn in unrooted form here).



$$Q(i,j) = (r-2)d(C_i,C_j) - u(C_i) - u(C_j)$$

d	A	B	C	D
A	0	8	4	6
B	8	0	8	8
C	4	8	0	6
D	6	8	6	0

u	
A	18
B	24
C	18
D	20

Q	A	B	C	D
A	0	-26	-28	-26
B	-26	0	-26	-28
C	-28	-26	0	-26
D	-26	-28	-26	0

Distance between A and the new node: d(A,C)/2 + [u(A) – u(C)] / [2(r-2)] = 4/2 + (18-18) / [2(2)] = 2

{A}, {C}

d	A,C	B	D
A,C	0	6	4
B	6	0	8
D	4	8	0

u	
A,C	10
B	14
D	12

Q	A,C	B	D
A,C	0	-18	-18
B	-18	0	-18
D	-18	-18	0

{A,C}, {B}

d	A,B,C	D
A,B,C	0	3
D	3	0

u	
A,B,C	3
B	3

Generalized Neighbor Joining

- One of the disadvantages of the NJ method is that it generates only one tree and does not test other possible tree topologies.
  - In the initial step of NJ, there may be more than one equally close pair of neighbors to join.
    - Select only one option may yield a suboptimal tree.
- Generalized NJ method:
  - Multiple NJ trees with different initial taxon groupings are generated.
  - A best tree is then selected from a pool of regular NJ trees that best fit the actual evolutionary distances.

Optimality-Based Methods

- Clustering methods produce a single tree with no ability to judge how good it is compared to alternative tree topologies
- Optimality-based methods compare all possible tree topologies and select a tree that best fits the distance matrix
- Two algorithms:
  - Fitch-Margoliash
  - Minimum Evolution
- The *exhaustive search* for an optimal tree necessitates a *slow computation*, which is a clear drawback especially when the dataset is large.

Fitch-Margoliash (FM)

- Selects best tree among all possible trees based on minimum deviation between distances calculated in the tree and distances in the distance matrix
- Basically, a least squares method
- d_{ij} = distance between i and j in matrix
- p_{ij} = distance between i and j in tree
- Objective: find tree that minimizes

$$E = \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \frac{\left(d_{ij} - p_{ij}\right)^2}{d_{ij}^2}$$

Minimum Evolution

- Similar to Fitch-Margoliash, but uses a different optimality criterion
- Searches for a tree with the minimum total branch length

$$S = \sum b_i$$

  where b_i is the ith branch length.
- This is an indirect way of achieving the best fit of the branch lengths with the original data.
- Analysis has shown that minimum evolution in fact slightly outperforms the least square-based FM method.

Summary of Distance-Based Methods

- Clustering-based methods:
  - – Computationally very fast and can handle large datasets that other methods cannot
  - – Not guaranteed to find the best tree
- Optimality-based methods:
  - – Better overall accuracies
  - – Computationally slow
- All distance-based methods lose all sequence information and cannot infer the most likely state at an internal node.

Character-Based Methods

- Based directly on the sequence characters in the MSA rather than overall pairwise distances
- Also called *discrete methods*
- Count mutational events accumulated on sequences
  - Avoid the loss of information when characters are converted to distances.
- Evolutionary dynamics of each character can be studied and ancestral sequences inferred
- Two popular approaches
  - Maximum Parsimony (MP)
  - Maximum Likelihood (ML)

Maximum Parsimony

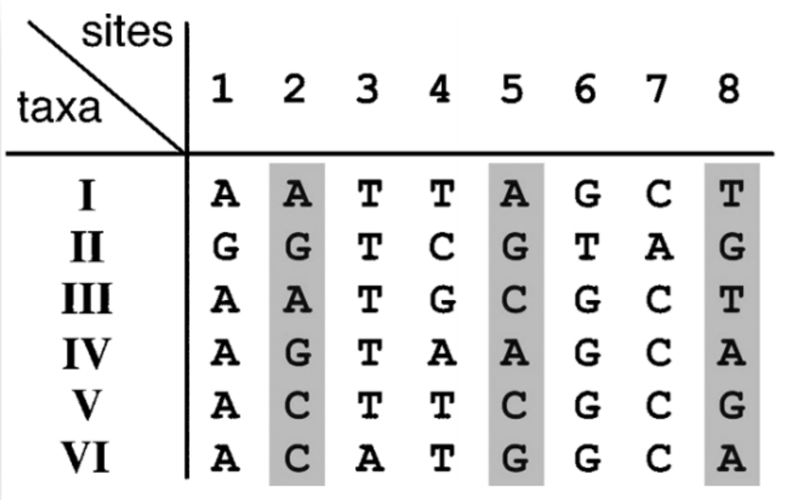
- Parsimony is based on *Occam's razor* principle
  - The simplest explanation is most likely correct
- **Goal:** choose a tree that has the fewest evolutionary changes or shortest overall branch lengths.
  - Tree with the least number of substitutions is probably the best
- **Parsimony score of a tree:** The smallest (weighted) number of steps required by the tree
- Two parsimony problems:
- **Large Parsimony problem:** Find the tree with the lowest parsimony score
- **Small Parsimony problem:** Given a tree, find its parsimony score
- Use the small parsimony problem to solve the large parsimony problem

Maximum Parsimony

- Parsimony is based on *Occam's razor* principle
  - – The simplest explanation is most likely correct
- **Goal:** choose a tree that has the fewest evolutionary changes or shortest overall branch lengths.
  - – Tree with the least number of substitutions is probably the best
- Parsimony tree building works by searching for all possible tree topologies and reconstructing ancestral sequences that require the minimum number of changes.

Maximum Parsimony (Cont.)

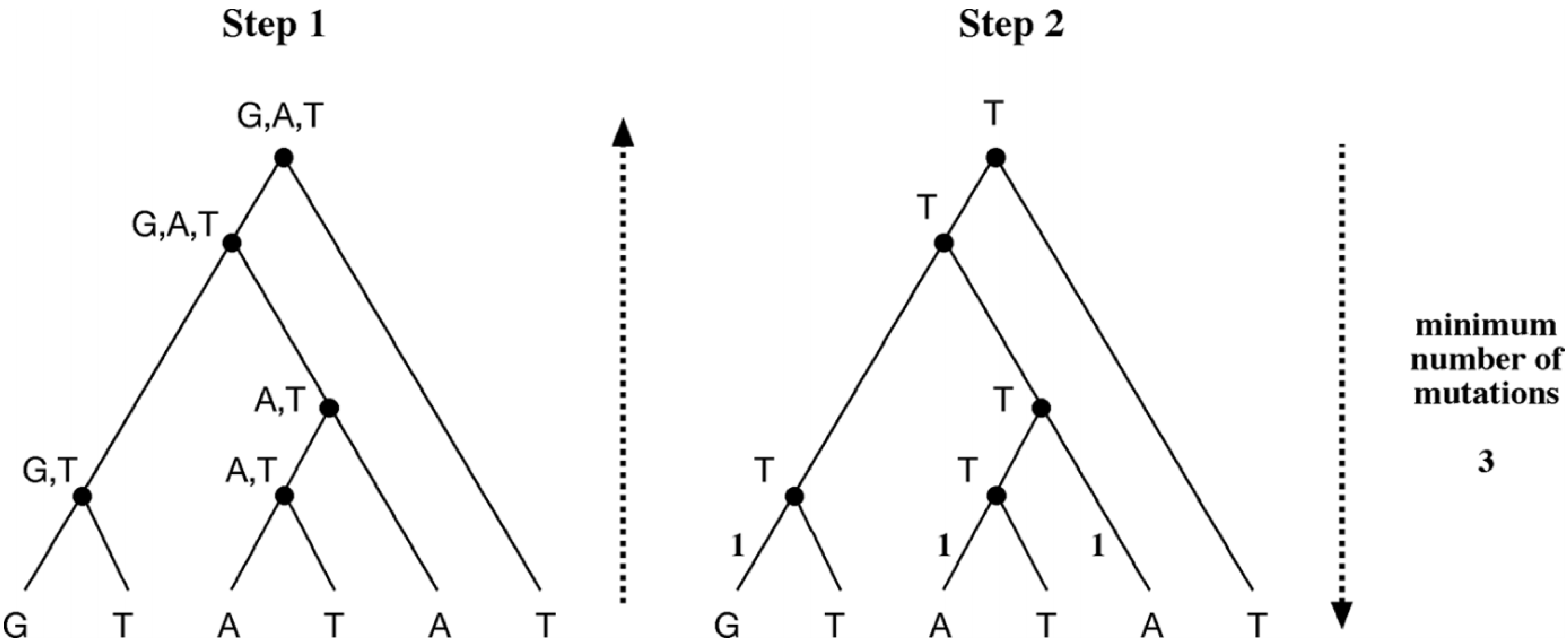
- To save computing time, the *richest phylogenetic information* sites are used:
  - Called *informative* sites
  - Sites that have at least two different kinds of characters, each occurring at least twice
  - Other sites are *noninformative*, which are *constant sites* or sites that have changes occurring only once.



sites taxa	1	2	3	4	5	6	7	8
I	A	A	T	T	A	G	C	T
II	G	G	T	C	G	T	A	G
III	A	A	T	G	C	G	C	T
IV	A	G	T	A	A	G	C	A
V	A	C	T	T	C	G	C	G
VI	A	C	A	T	G	G	C	A

- Then the minimum number of substitutions at each informative site is computed for a given tree topology.
- The tree that has the smallest number of changes is chosen as the best tree.

Predicting Ancestral Sequences at Internal Nodes

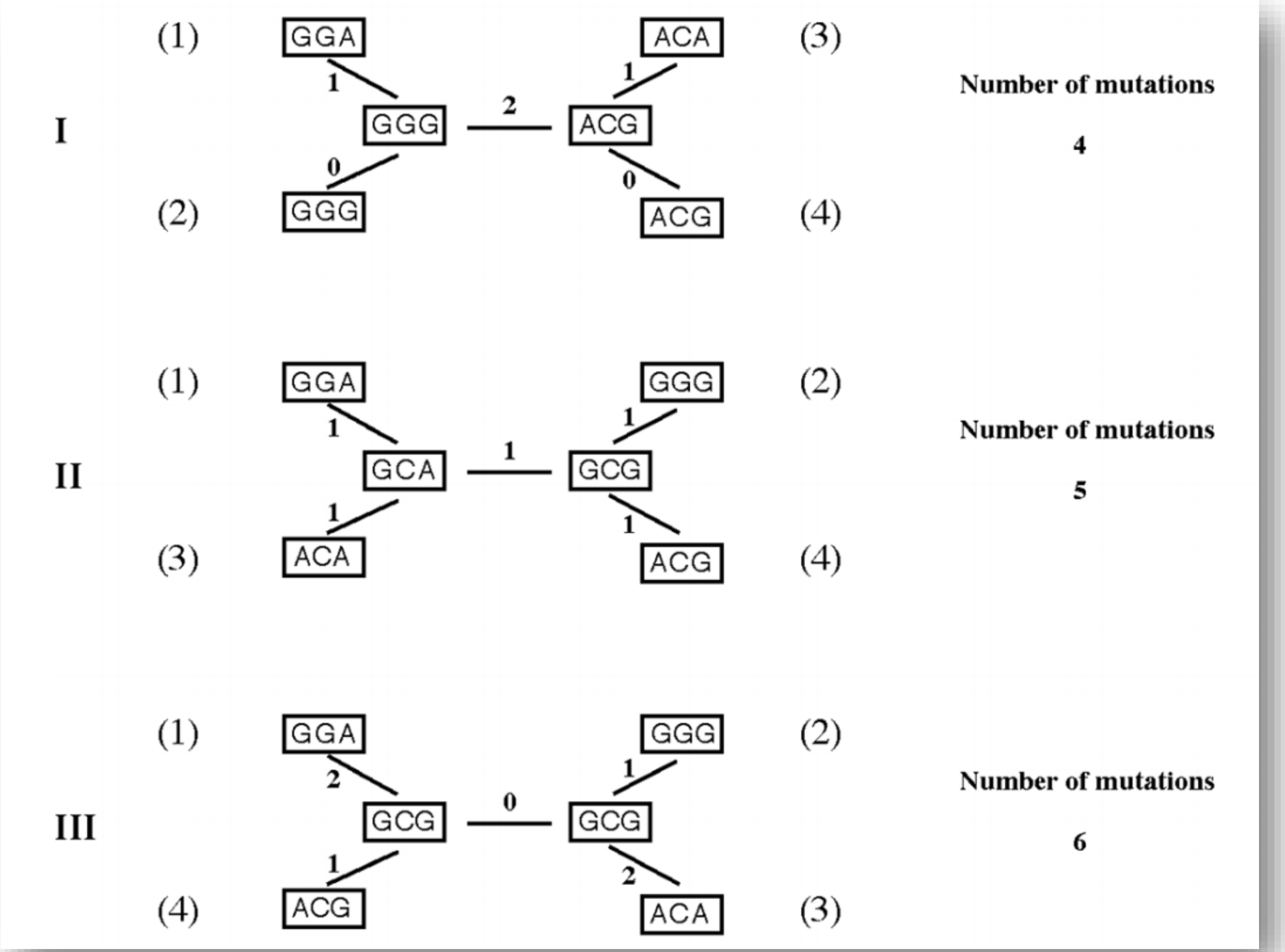


Weighted Parsimony

- The parsimony method discussed is unweighted
- The MP method that incorporates a weighting scheme is called *weighted parsimony.*
  - Transversions are more costly than transitions
- In some cases, the weighting scheme may result in different tree topologies.
- We will see a comparison example where transitions are weighted as 1 and transversions are weighted as 5.

Unweighted Parsimony Example

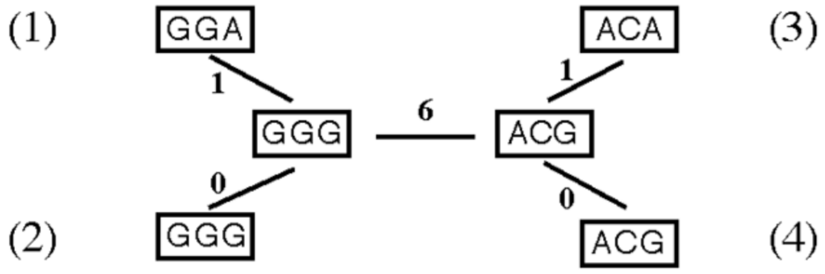
1 : GGA
2 : GGG
3 : ACA
4 : ACG



Weighted Parsimony Example

1 : GGA
2 : GGG
3 : ACA
4 : ACG

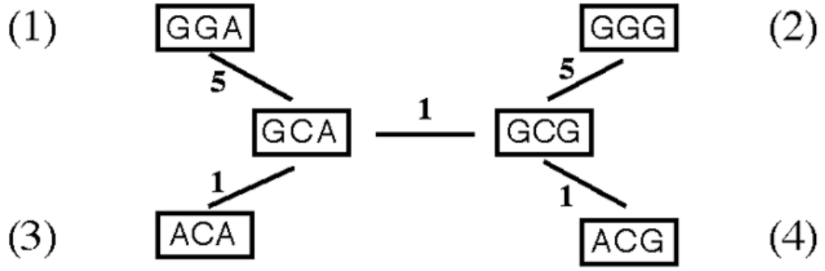
I



Number of mutations

8

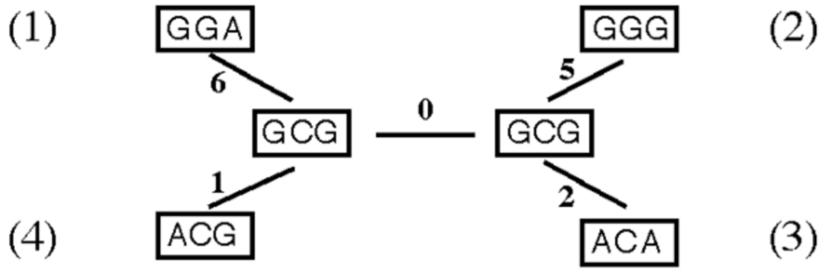
II



Number of mutations

13

III



Number of mutations

14

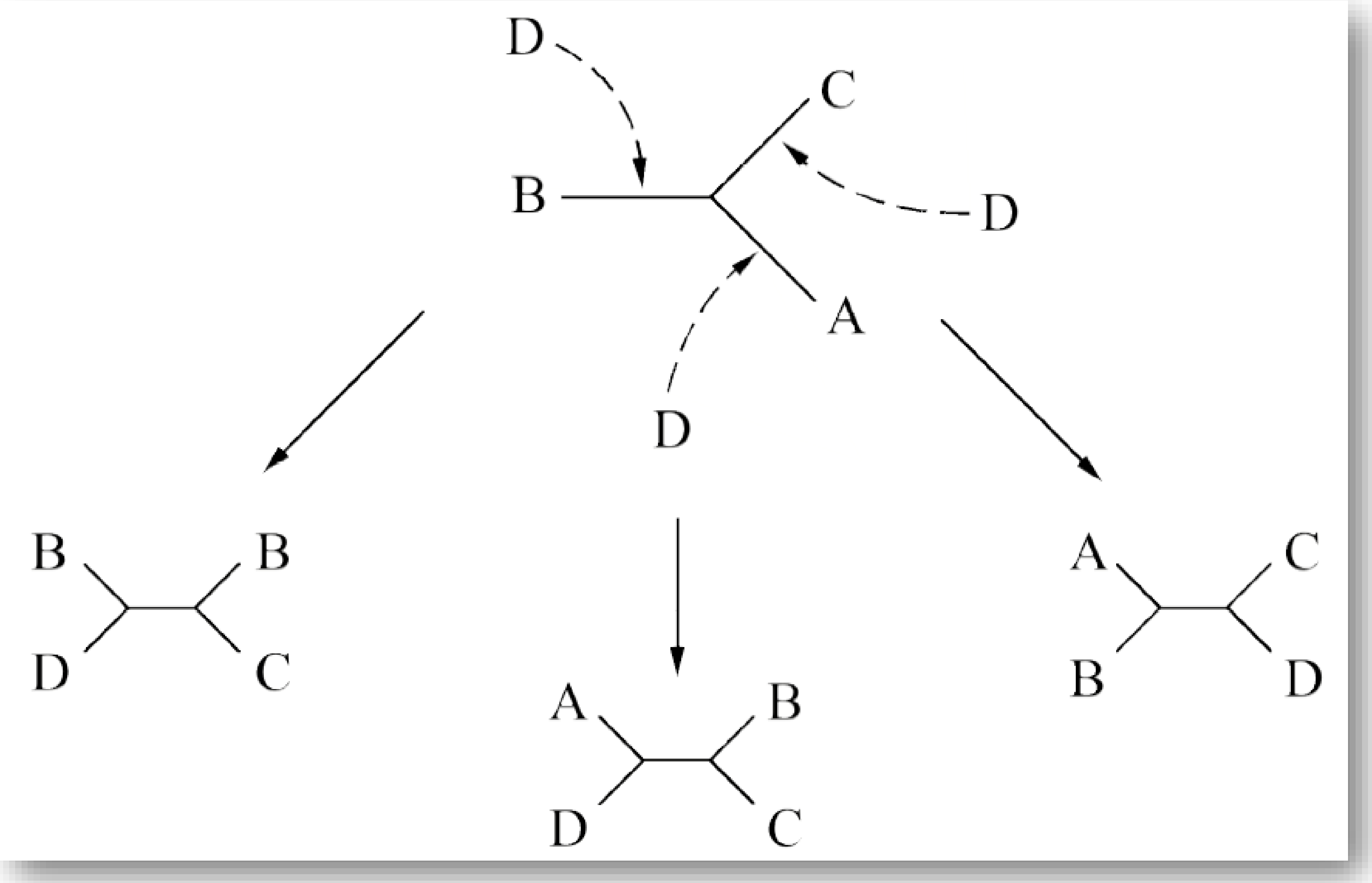
Searching for a Most Parsimonious Tree

- Solving the large parsimony problem requires searching all possible trees
  - This is an exhaustive search method.
- Searching methods:
  - Exhaustive search (exact)
  - Branch-and-Bound (exact)
  - Heuristic search methods (not exact)

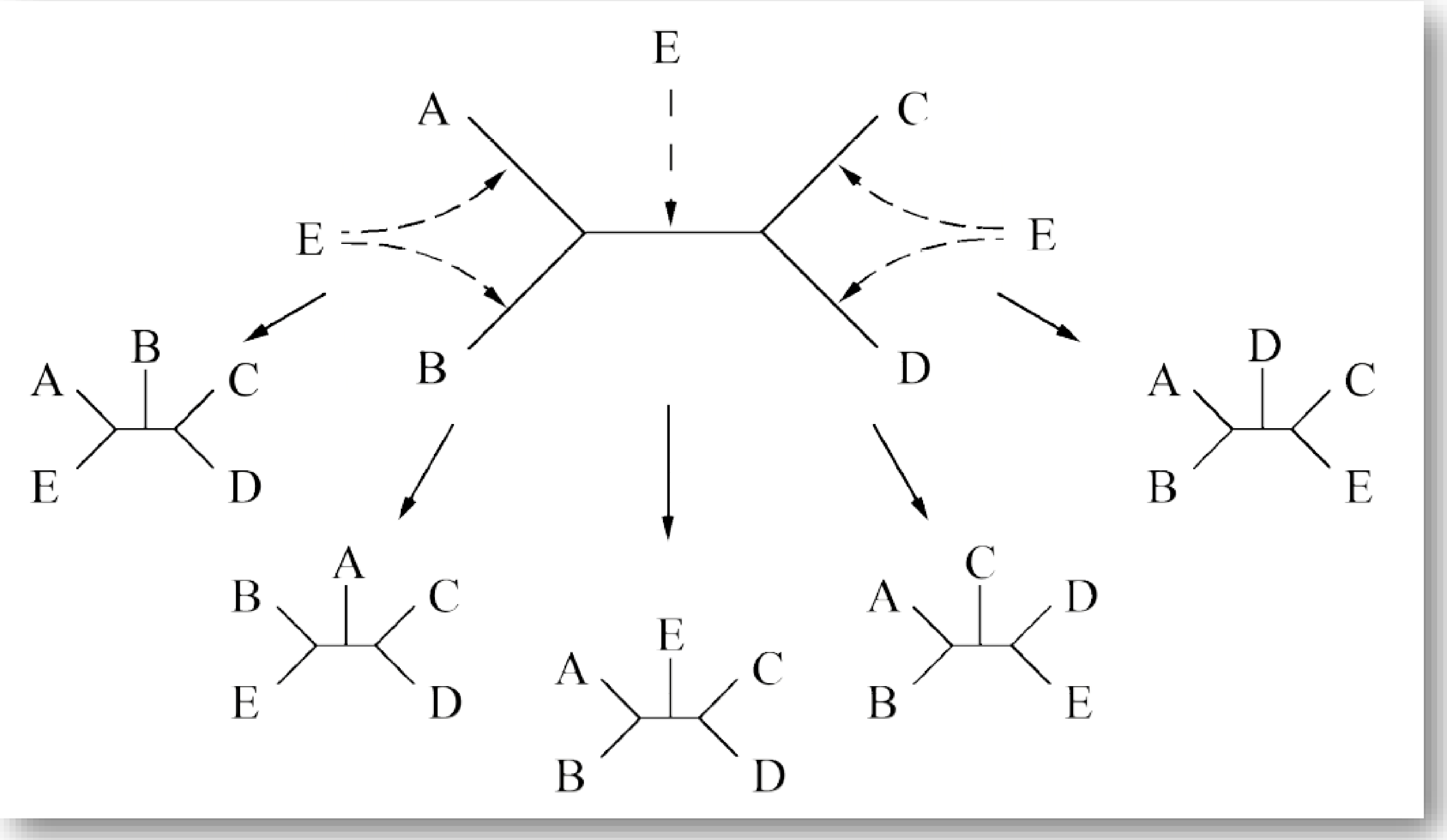
Exhaustive Search

- Build the only possible unrooted tree for three taxa (can be randomly chosen)
- Try all possible places to add the fourth taxon and score each tree
- Try all places to add the fifth taxon to the trees and score again
- Continue to add all taxa to the trees and find the best one.
- The method is computationally too demanding to use when the number of taxa is more than ten.

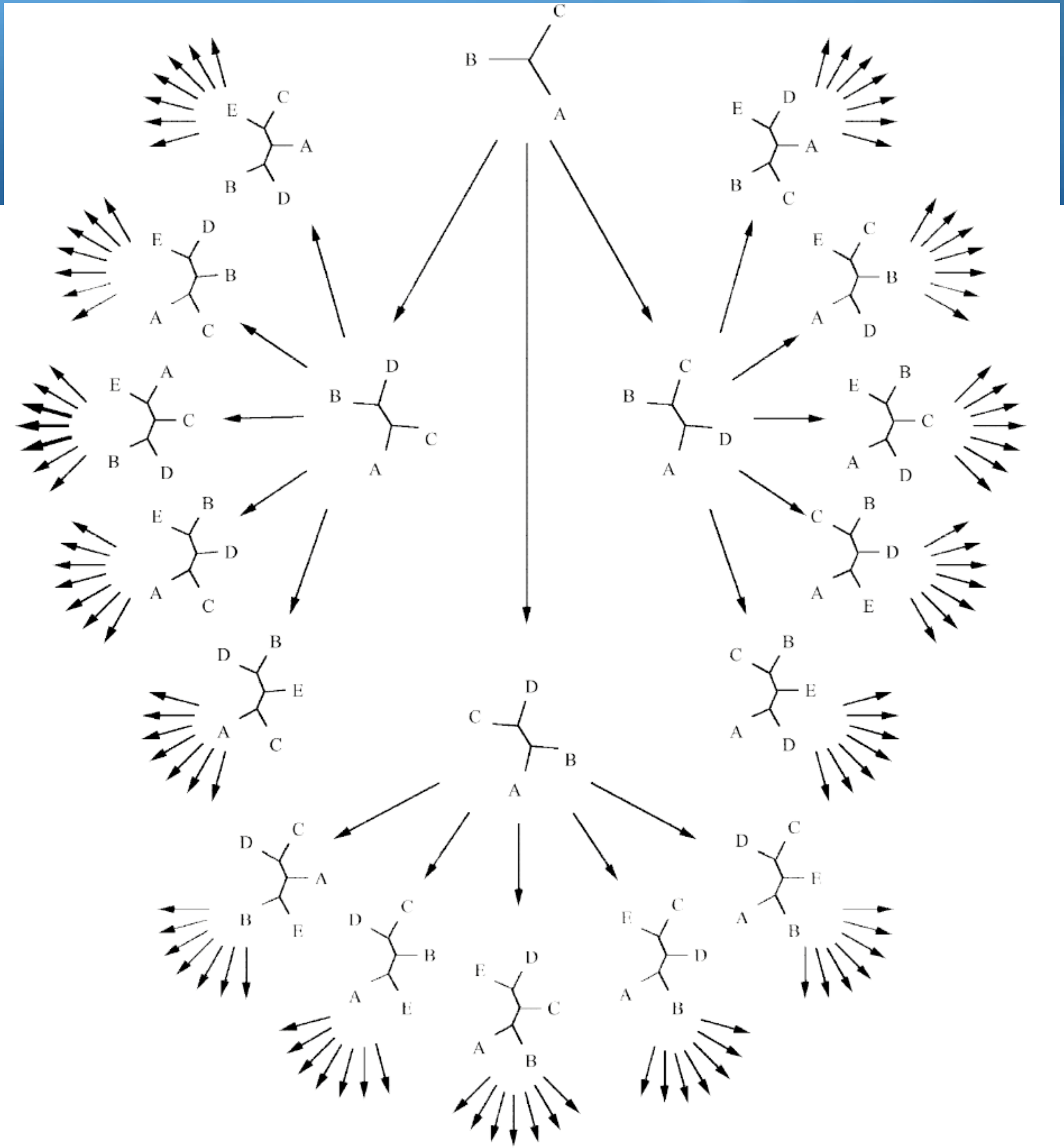
Adding the Fourth Taxon



Adding the Fifth Taxon



Add More!



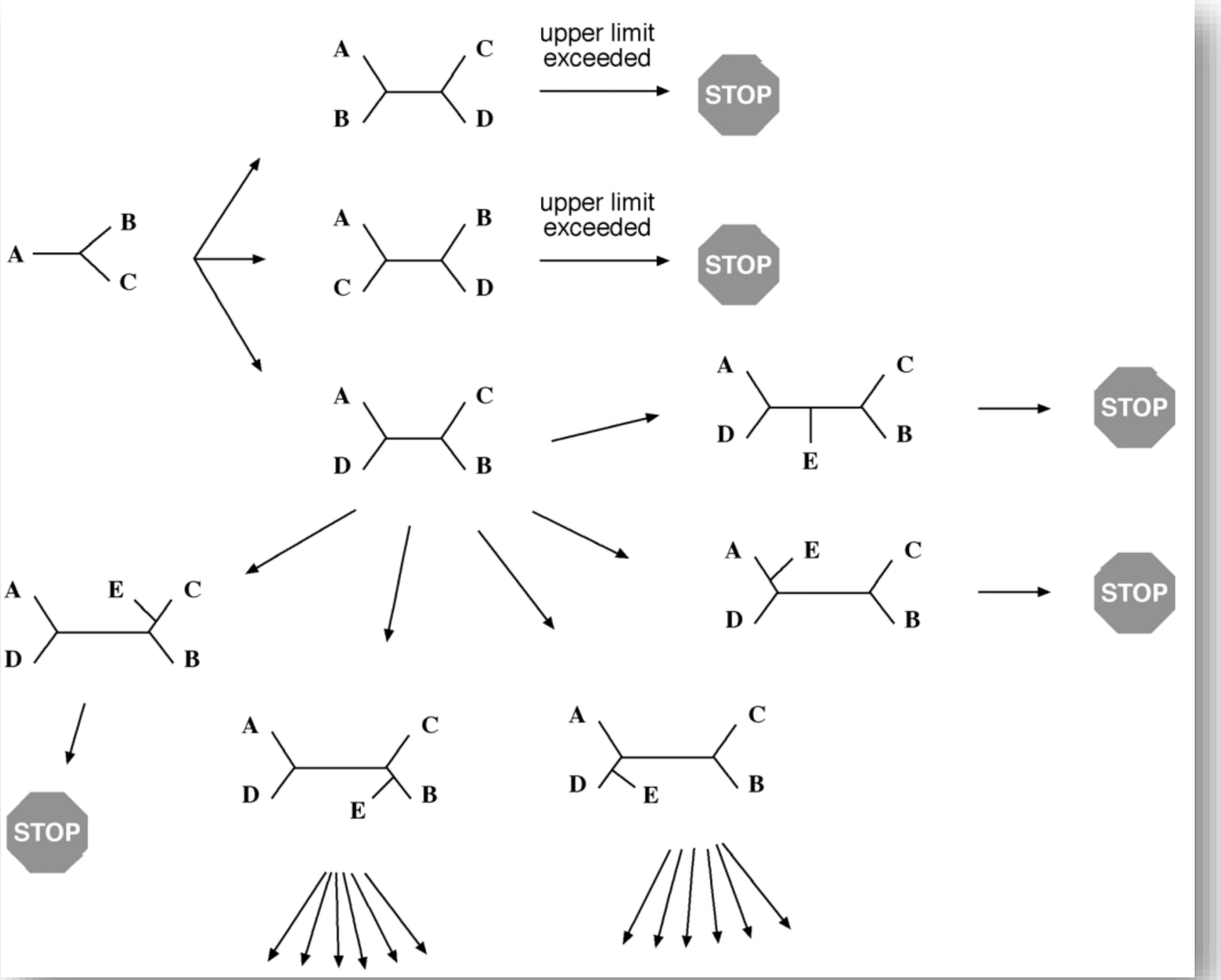
Branch-and-Bound

- It starts by building a distance tree for all taxa involved using either NJ or UPGMA.
  - Computes the *minimum number of substitutions* for this tree.
    - The result defines the **upper bound** to which any other trees are compared.
  - The rationale is that a maximally parsimonious tree must be equal to or shorter than the distance-based tree.
- Similar to exhaustive search except that we maintain the score of the best tree obtained so far which limits the tree growth.
  - When a tip of the search tree is reached the tree is either optimal (and retained) or suboptimal (and rejected)

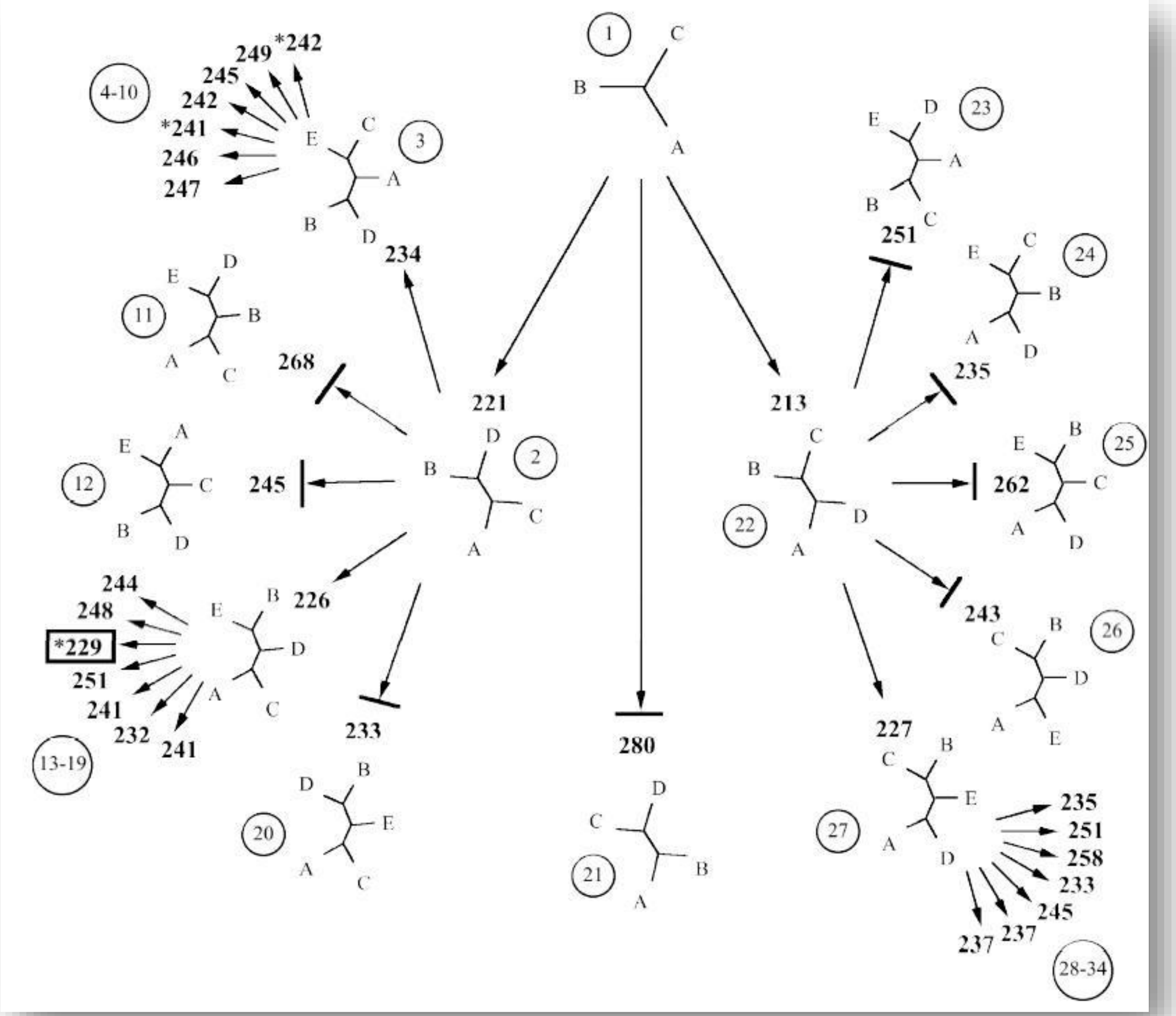
Branch-and-Bound (Cont.)

- When a tip of the search tree is reached the tree is either optimal (and retained) or suboptimal (and rejected)
- When all paths leading from the initial 3 taxon tree have been explored, the algorithm terminates, and all **most parsimonious** trees will have been identified.
- It can be used for up to twenty taxa and after that, the method becomes computationally unfeasible.

Branch-and-Bound : Breadth First Search



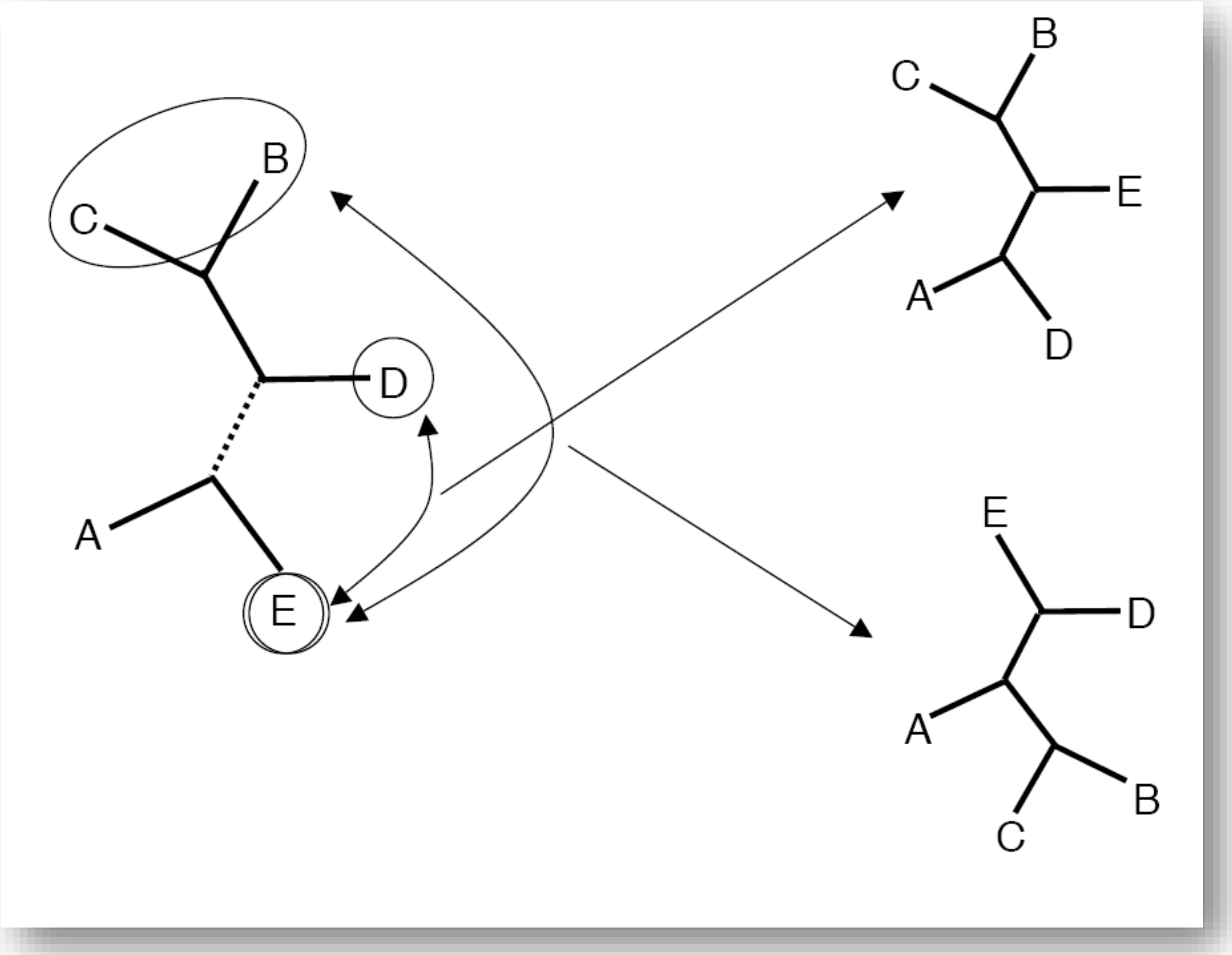
Branch-and-Bound: Depth First Search



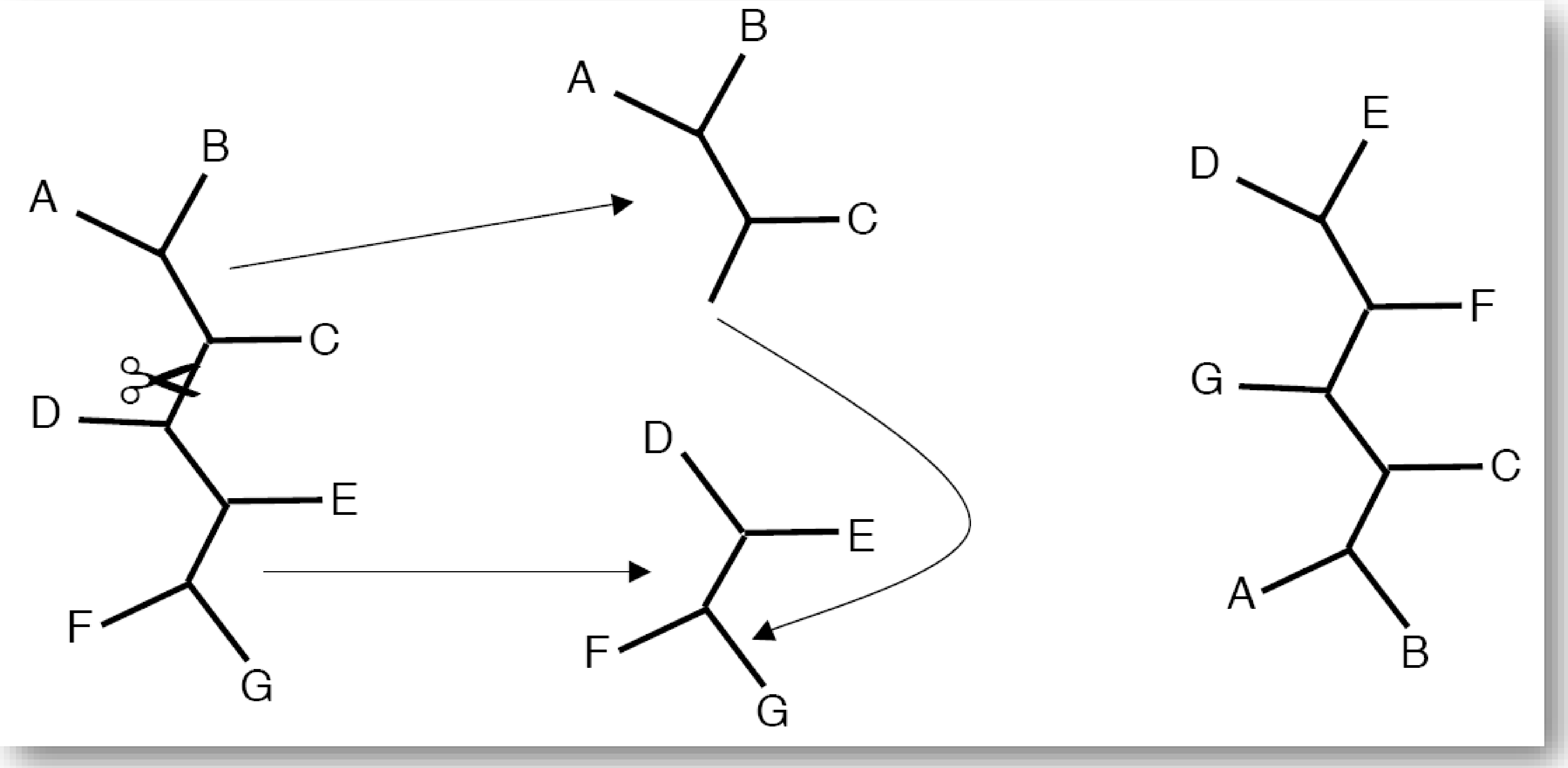
Heuristic Search

- Shortcuts have been designed to reduce the search space
- Idea: Build a tree quickly (by NJ or some other fast method) and rearrange parts of it to explore some of the possible trees
  - The total branch length for the new tree is recomputed.
  - If the tree is found to be shorter, it is used as a starting point for another round of rearrangement.
- Branch swapping
  - Nearest Neighbor Interchange
  - Subtree pruning and regrafting
  - Tree bisection and reconnection

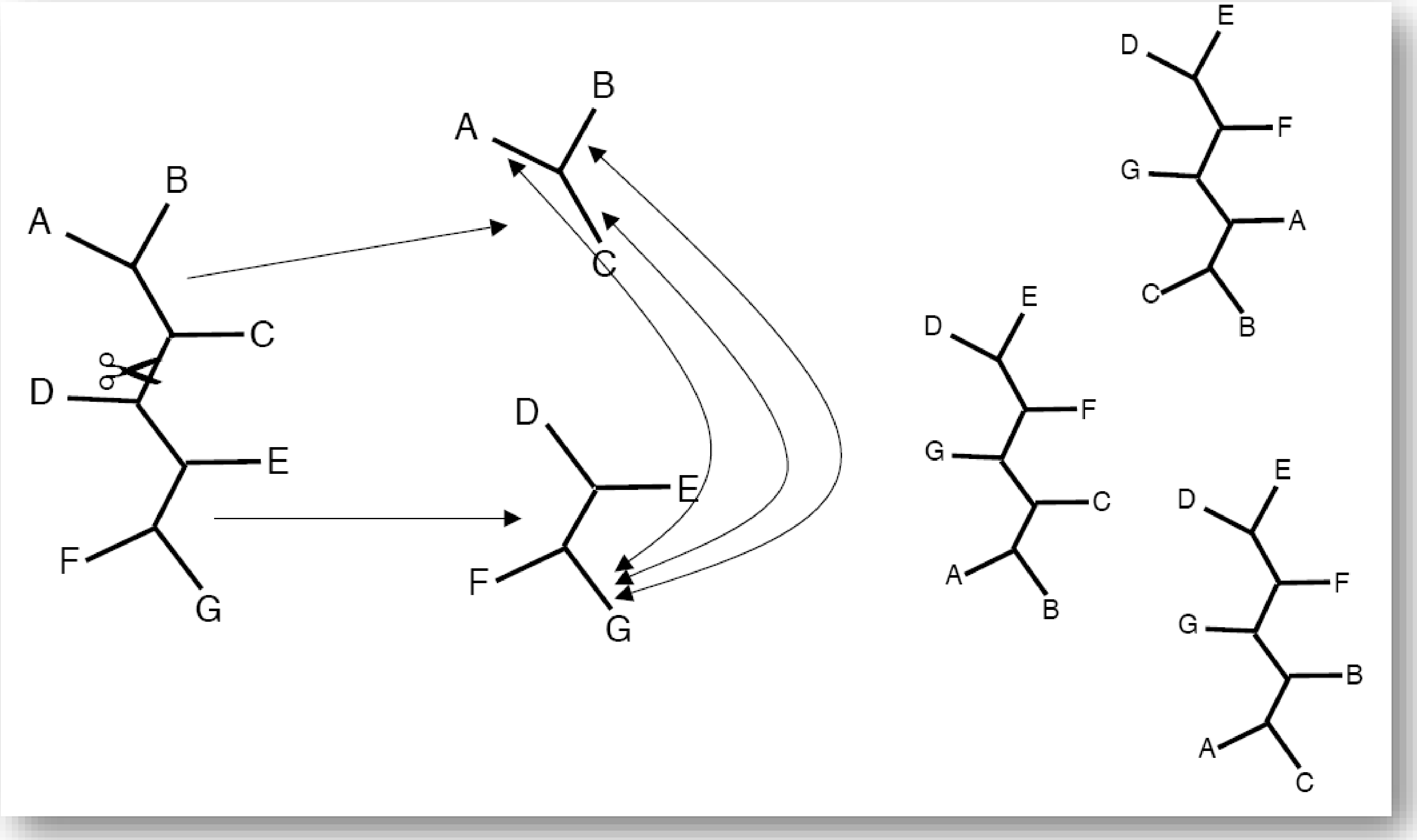
Nearest-Neighbor Interchange



Subtree Pruning and Regrafting



Tree Bisection and Reconnection



Stepwise Addition – Another Heuristic

- A greedy method
- Start with 3 taxon tree
- Add one taxon at a time
- Keep only the best tree found so far
- No guarantee of optimality, but may provide a good starting point for a search

MP Pros and Cons

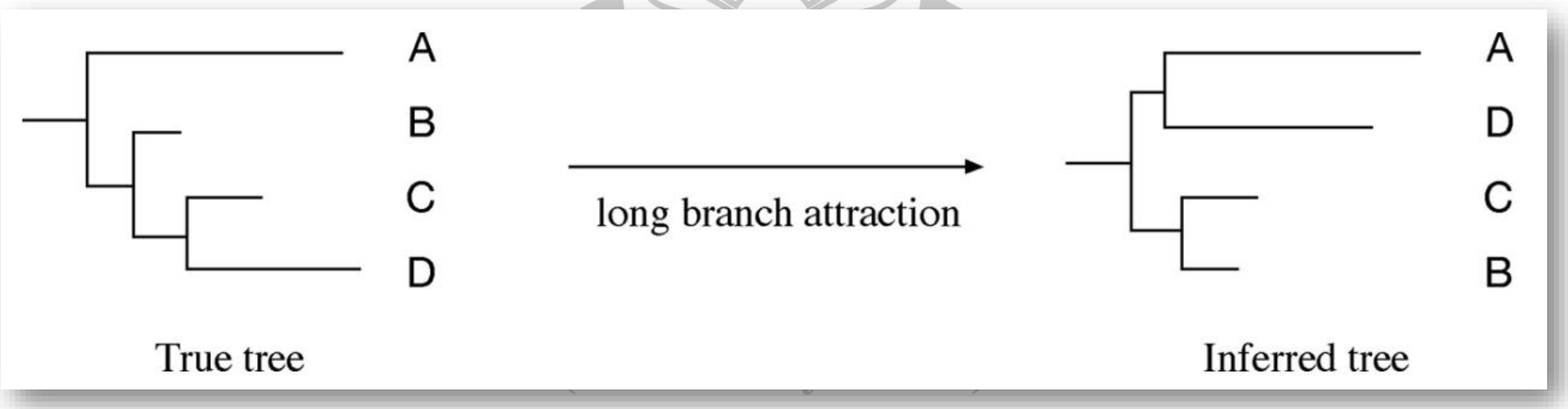
- Pros:
  - The main advantage of MP is that it is intuitive
    - Its assumptions are easily understood.
  - The character-based method is able to provide evolutionary information about the sequence characters.
  - It tends to produce more accurate trees than the distance-based methods when sequence divergence is low.
- Cons:
  - When sequence divergence is high, or the amount of homoplasies is large, tree estimation by MP can be less effective.

MP Pros and Cons

- Cons (Cont.):
  - When sequence divergence is high, or the amount of homoplasies is large, tree estimation by MP can be less effective.
  - Estimation of branch lengths may also be erroneous.
    - MP does not employ substitution models to correct for multiple substitutions.
  - MP only considers informative sites, and ignores other sites.
    - Certain phylogenetic signals may be lost.
  - MP is also slow compared to the distance methods.
  - Is very sensitive to the “long-branch attraction” artifacts.

Long-Branch Attraction (LBA)

- LBA refers to a phylogenetic artifact in which rapidly evolving taxa with long branches are placed together in a tree, regardless of their true positions in a tree.



Maximum Likelihood (ML) Method

- ML is based on a Markov model of evolution
  - Uses probabilistic models to choose a best tree that has the highest probability or likelihood of reproducing the observed data.
- It finds a tree that most likely reflects the actual evolutionary process.
- ML is an exhaustive method that searches every possible tree topology
- It considers every position in an alignment, not just informative sites.
- Its performance depends of the used substitution model

Maximum Likelihood (ML) Method

- **Observed**: The species labeling the leaves
- **Hidden**: The ancestral states
- **Transition probabilities**: The mutation probabilities
- **Assumptions**:
  - Only mutations are allowed
  - Sites are independent
  - Branches may have different lengths
- Transition probability matrix:

$$M = \left[m_{ij}\right] \quad \text{i,j } \{A,C,T,G\}$$

where m_{ij} = Prob(i -> j mutation in 1 time unit)

Maximum Likelihood (ML) Method

- ML works by calculating the probability of a given evolutionary path for a particular extant sequence.
- The probability values are determined by a substitution model.
- For Jukes–Cantor model, the probability (*P*) that a nucleotide remains the same after time *t* is:

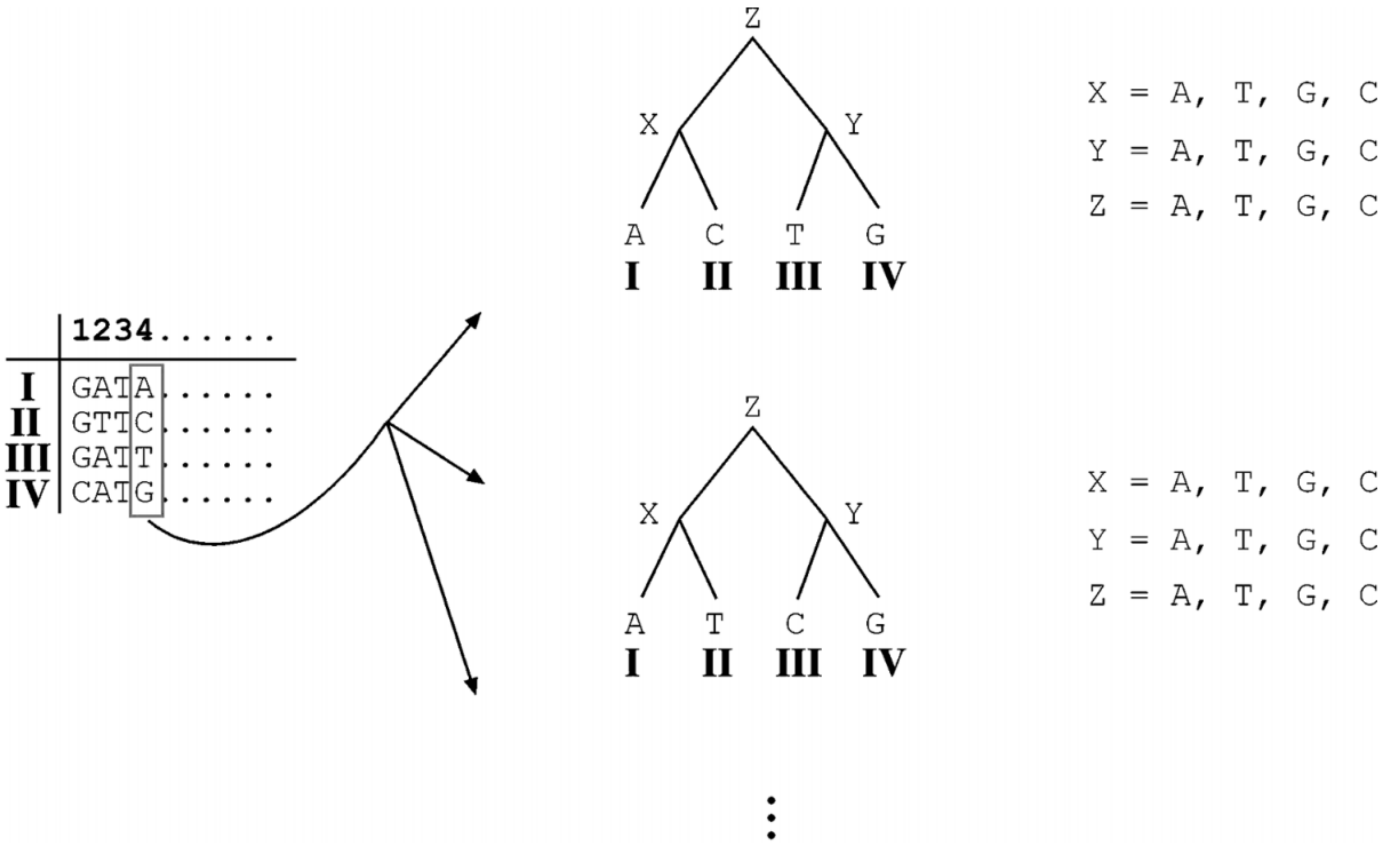
$$P(t) = 1/4 + 3/4\, e^{-\alpha t}$$

- For a nucleotide to change into a different residue after time *t*, the probability value is determined by:

$$P(t) = 1/4 - 1/4\, e^{-\alpha t}$$

The Probability of an Assignment

- For a particular site, the probability of a tree path is the product of the probability from the root to all the tips, including every intermediate branches in the tree topology.
- It is computationally more convenient to express all probability values as natural log likelihood values.



```
     1234......
I    GATA......
II   GTTC......
III  GATT......
IV   CATG......
```

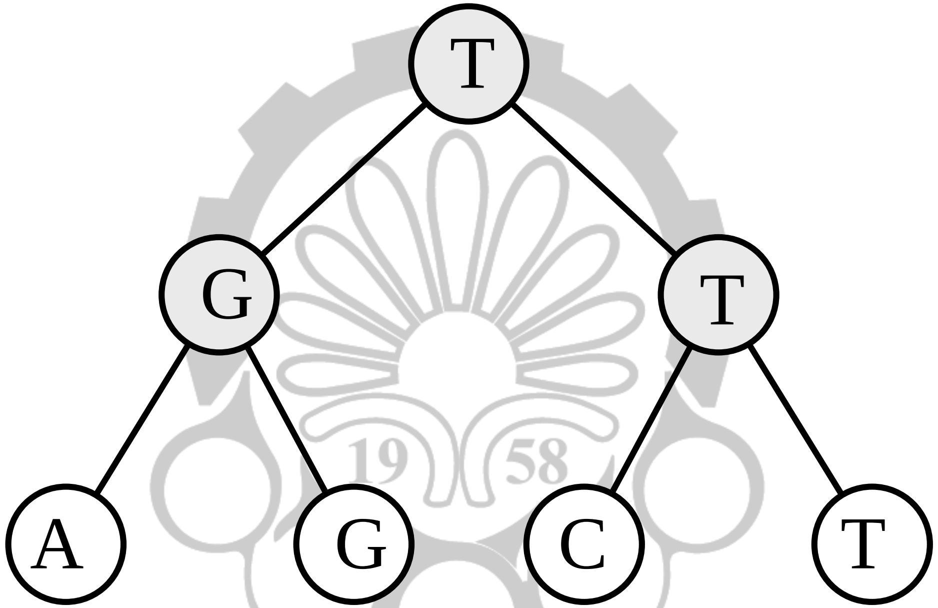
$$L_{(4)} = \Pr(Z \to X) * \Pr(Z \to Y) * \Pr(X \to A) * \Pr(X \to C) * \Pr(Y \to T) * \Pr(Y \to G)$$

$$\begin{aligned}\ln L_{(4)} &= \ln \Pr(Z \to X) + \ln \Pr(Z \to Y) + \ln \Pr(X \to A) + \ln \Pr(X \to C) \\ &\quad + \ln \Pr(Y \to T) + \ln \Pr(Y \to G)\end{aligned}$$

Maximum Likelihood (ML) Method

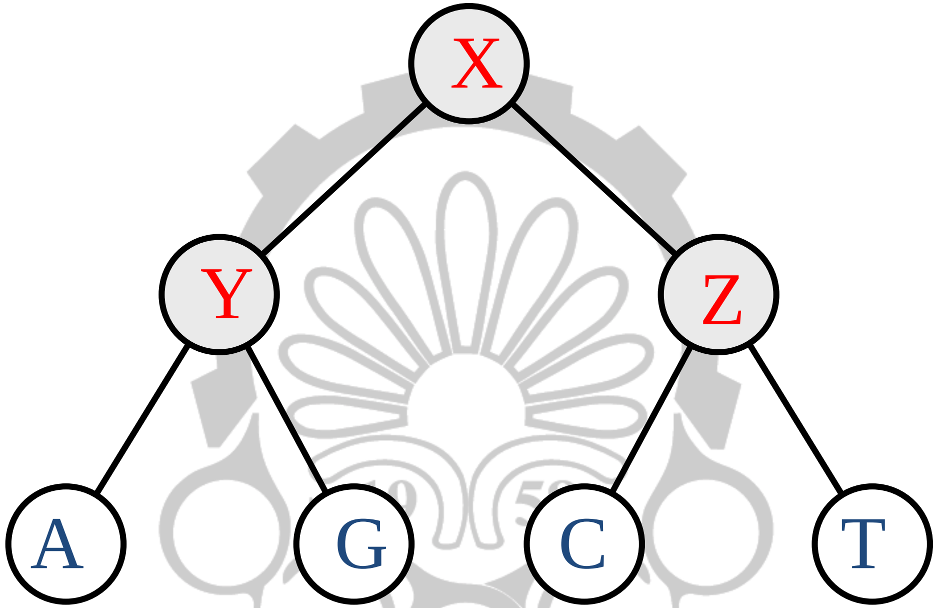
- The overall log likelihood score for a given tree path for the entire sequence is the sum of log likelihood of all individual sites.
- The same procedure has to be repeated for all other possible tree topologies.
- The tree having the highest likelihood score among all others is chosen as the best tree, which is the ML tree.
- This process is exhaustive in nature and therefore very time consuming.

The Probability of an Assignment



Probability = $m_{TG} \cdot m_{GA} \cdot m_{GG} \cdot m_{TT} \cdot m_{TC} \cdot m_{TT}$

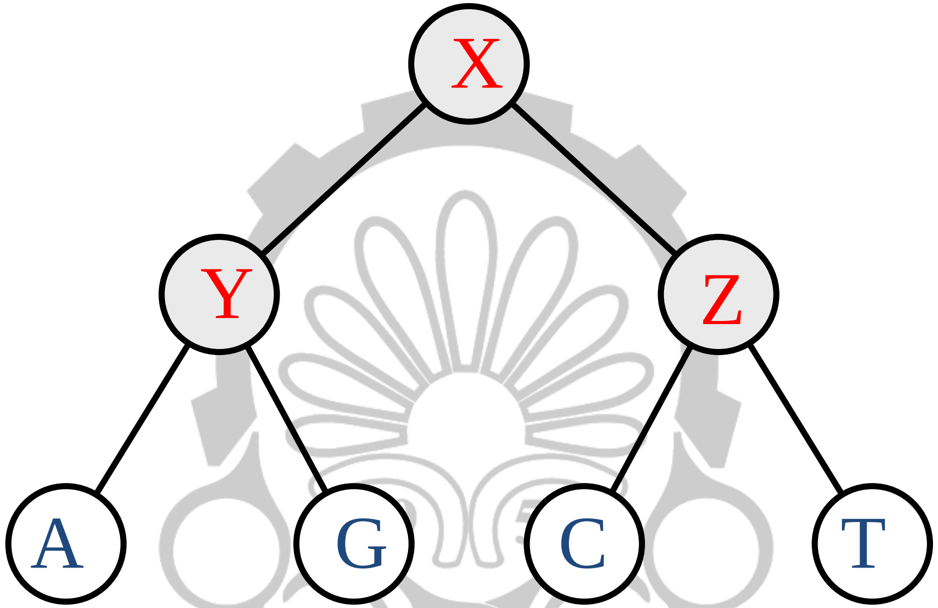
Ancestral Reconstruction: Most Likely Assignment



$$L^* = \max_{X,Y,Z} \{m_{XY} \cdot m_{YA} \cdot m_{YG} \cdot m_{XZ} \cdot m_{ZC} \cdot m_{ZT}\}$$

Compute using Viterbi algorithm

Likelihood of a Tree



$$L^* = \sum_{X,Y,Z} \{m_{XY} \cdot m_{YA} \cdot m_{YG} \cdot m_{XZ} \cdot m_{ZC} \cdot m_{ZT}\}$$

Compute using forward algorithm

Maximum Likelihood Comments

- ML is robust
- ML converges to the correct answer as more data is added
- Can put in a Bayesian statistical framework to obtain a distribution of possible phylogenies
- ML can be slow because of its exhaustive nature.
  - To overcome the problem, several heuristic or alternative approaches have been proposed which are not covered here.

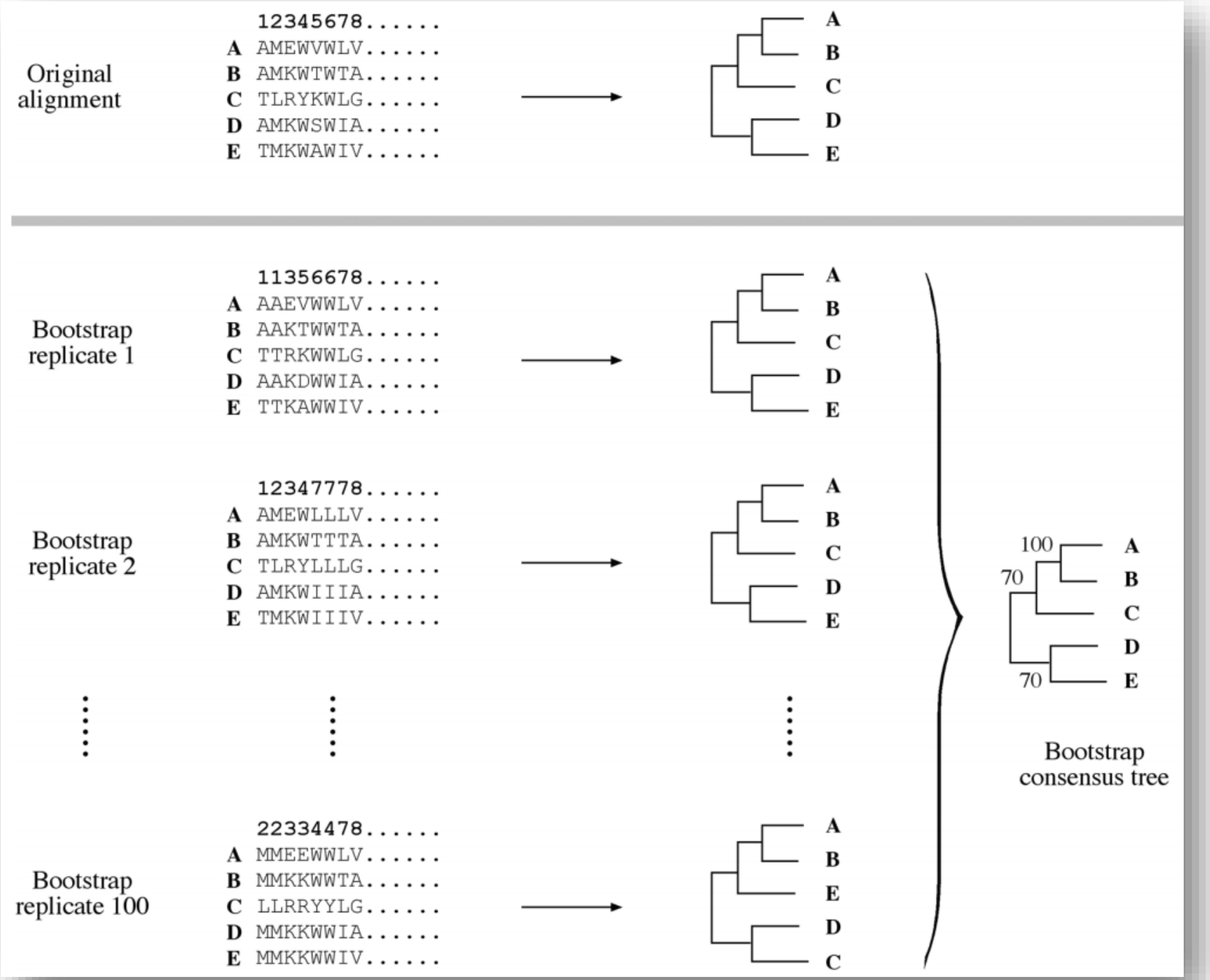
Phylogenetic Tree Evaluation

- After tree construction, the next step is to statistically evaluate the reliability of the inferred phylogeny.
- How reliable the tree or a portion of the tree is?
- Whether this tree is significantly better than another tree?
- Bootstrapping
- Jackknifing
- Bayesian Simulation
- Statistical difference tests (are two trees significantly different?)
  - Kishino-Hasegawa Test (paired t-test)
  - Shimodaira-Hasegawa Test (χ^2 test)

Bootstrapping

- *Bootstrapping* is a statistical technique that tests the sampling errors of a phylogenetic tree.
- A bootstrap sample is obtained by sampling sites randomly with replacement
  - Obtain a data matrix with same number of taxa and number of characters as original one
- Construct trees for samples
- For each branch in original tree, compute fraction of bootstrap samples in which that branch appears
  - Assigns a bootstrap support value to each branch
- Idea: If a grouping has a lot of support, it will be supported by at least some positions in most of the bootstrap samples

Schematic Representation of Bootstrapping



Original alignment

```
  12345678......
A AMEWVWLV......
B AMKWTWTA......
C TLRYKWLG......
D AMKWSWIA......
E TMKWAWIV......
```

Bootstrap replicate 1

```
  11356678......
A AAEVWWLV......
B AAKTWWTA......
C TTRKWWLG......
D AAKDWWIA......
E TTKAWWIV......
```

Bootstrap replicate 2

```
  12347778......
A AMEWLLLV......
B AMKWTTTA......
C TLRYLLLG......
D AMKWIIIA......
E TMKWIIIV......
```

Bootstrap replicate 100

```
  22334478......
A MMEEWWLV......
B MMKKWWTA......
C LLRRYYLG......
D MMKKWWIA......
E MMKKWWIV......
```

Bootstrap consensus tree

Bootstrapping Comments

- Bootstrapping strategies:
  - *Nonparametric bootstrapping*: produce perturbations through random replacement (random duplication) of sites.
  - *Parametric bootstrapping*: new datasets can be generated based on a particular sequence distribution (i.e. substitution model). Is more robust than nonparametric.
- Analysis has shown that a bootstrap value of 70% approximately corresponds to 95% statistical confidence.
- Bootstrapping doesn’t really assess the accuracy of a tree, only indicates the consistency of the data: bootstrap results should be interpreted with caution.
- To get reliable statistics, bootstrapping needs to be done on your tree 500 – 1000 times, this is a big problem if your tree took a few days to construct.

Jackknifing

- Another resampling technique
- Randomly delete half of the sites in the dataset
- Construct new tree with this smaller dataset, see how often taxa are grouped
- Advantage – sites aren’t duplicated
  - – Computing time is much shortened
- Disadvantage – again really only measuring consistency of the data

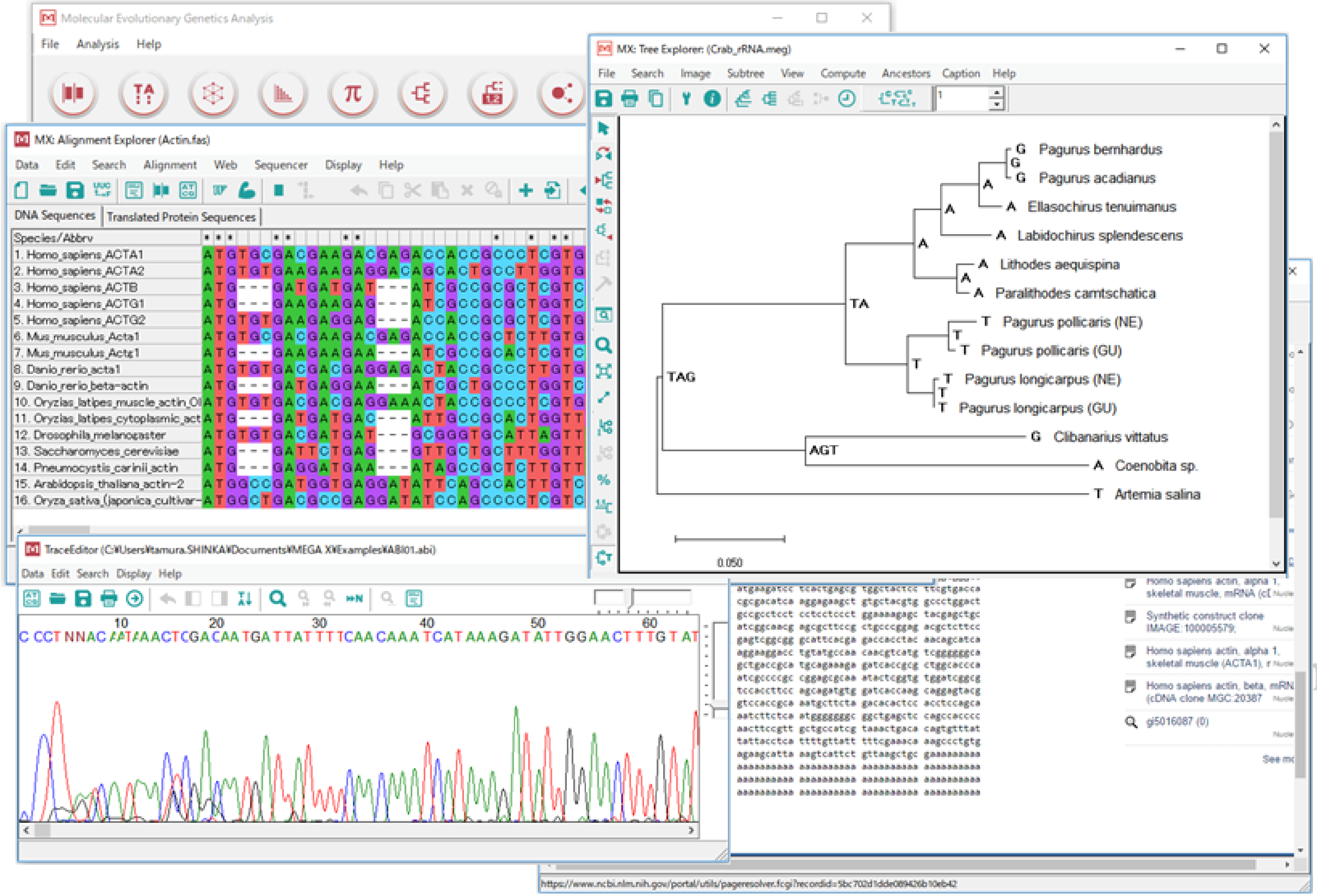
Final Comments on Phylogenetics

- No method is perfect
- Different methods make very different assumptions
- If multiple methods using different assumptions come up with similar results, we should trust the results more than any single method

Phylogenetic Programs

- Huge list at: http://evolution.genetics.washington.edu/phylip/software.html
- PAUP* - one of the most popular programs, commercial, Mac and Unix only, nice user interface
- PHYLIP – free, multiplatform, a bit difficult to use but web servers make it easier
- WebPhylip – another interface for PHYLIP online
- TREE-PUZZLE – uses a heuristic to allow ML on large datasets, also available as a web server
- PHYML – web based, uses genetic algorithm
- MrBayes – Bayesian program, fast and can handle large datasets, multiplatform
- BAMBE – web based Bayesian program

Molecular Evolutionary Genetics Analysis (MEGA)


Molecular Evolutionary Genetics Analysis
File Analysis Help
MX: Alignment Explorer (Actin.fas)
Data Edit Search Alignment Web Sequencer Display Help
DNA Sequences
Translated Protein Sequences
MX: Tree Explorer: (Crab_rRNA.meg)
File Search Image Subtree View Compute Ancestors Caption Help
G Pagurus bernhardus
G Pagurus acadianus
A Ellasochirus tenuimanus
A Labidochirus splendescens
A Lithodes aequispina
A Paralithodes camtschatica
T Pagurus pollicaris (NE)
T Pagurus pollicaris (GU)
T Pagurus longicarpus (NE)
T Pagurus longicarpus (GU)
G Clibanarius vittatus
A Coenobita sp.
T Artemia salina
TA
TAG
AGT
0.050
TraceEditor (C:¥Users¥tamura.SHINKA¥Documents¥MEGA X¥Examples¥AB101.abi)
Data Edit Search Display Help
10 20 30 40 50 60
C CCT NNAC AATAAACTCGACAATGATTATTTTCAACAAATCATAAAGATATTGGAACTTTGTAT
Homo sapiens actin, alpha 1, skeletal muscle, mRNA (cD
Synthetic construct clone IMAGE:100005579;
Homo sapiens actin, alpha 1, skeletal muscle (ACTA1), r
Homo sapiens actin, beta, mRN (cDNA clone MGC:20387
gi5016087 (0)
See mo
nology
https://www.ncbi.nlm.nih.gov/portal/utils/pageresolver.fcgi?recordid=5bc702d1dde089426b10eb42

Species/Abbrv	
1. Homo_sapiens_ACTA1	ATGTGCGACGAAGACGAGACCACCGCCCTCGTG
2. Homo_sapiens_ACTA2	ATGTGTGAAGAAGAGGACAGCACTGCCTTGGTG
3. Homo_sapiens_ACTB	ATG---GATGATGAT---ATCGCCGCGCTCGTC
4. Homo_sapiens_ACTG1	ATG---GAAGAAGAG---ATCGCCGCGCTGGTC
5. Homo_sapiens_ACTG2	ATGTGTGAAGAGGAG---ACCACCGCGCTCGTG
6. Mus_musculus_Acta1	ATGTGCGACGAAGACGAGACCACCGCTCTTGTG
7. Mus_musculus_Actg1	ATG---GAAGAAGAA---ATCGCCGCACTCGTC
8. Danio_rerio_acta1	ATGTGTGACGACGAGGAGACTACCGCCCTTGTG
9. Danio_rerio_beta-actin	ATG---GATGAGGAA---ATCGCTGCCCTGGTC
10. Oryzias_latipes_muscle_actin_Ol	ATGTGTGACGACGAGGAAACTACCGCCCTCGTG
11. Oryzias_latipes_cytoplasmic_act	ATG---GATGATGAC---ATTGCCGCACTGGTT
12. Drosophila_melanogaster	ATGTGTGACGATGAT---GCGGGTGCATTAGTT
13. Saccharomyces_cerevisiae	ATG---GATTCTGAG---GTTGCTGCTTTGGTT
14. Pneumocystis_carinii_actin	ATG---GAGGATGAA---ATAGCCGCTCTTGTT
15. Arabidopsis_thaliana_actin-2	ATGGCCGATGGTGAGGATATTCAGCCACTTGTC
16. Oryza_sativa_(japonica_cultivar-	ATGGCTGACGCCGAGGATATCCAGCCCCTCGTC

```
atgaagatcc tcactgagcg tggctactcc ttcgtgacca
cgcgacatca aggagaagct gtgctacgtg gccctggact
gccgcctcct cctcctccct ggaaaagagc tacgagctgc
atcggcaacg agcgcttccg ctgcccggag acgctcttcc
gagtcggcgg gcattcacga gaccacctac aacagcatca
aggaaggacc tgtatgccaa caacgtcatg tcgggggca
gctgaccgca tgcagaaaga gatcaccgcg ctggcaccca
atcgccccgc cggagcgcaa atactcggtg tggatcggcg
tccaccttcc agcagatgtg gatcaccaag caggagtacg
gtccaccgca aatgcttcta gacacactcc acctccagca
aatcttctca atgggggggc ggctgagctc cagccacccc
aactttccgtt gctgccatcg taaactgaca cagtgtttat
tattacctca ttttgttatt tttcgaaaca aagccctgtg
agaagcatta aagtcattct gttaagctgc gaaaaaaaaa
aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa aaaaaaaaaa
```

MEGA Molecular Evolutionary Genetics Analysis

References

- Mostly used:
  - Essential bioinformatics, Chapter 11 (Phylogenetic Tree Construction Methods and Programs)
- Second reference:
  - Bioinformatics and functional genomics, Chapter 7 (Molecular Phylogeny and Evolution)

- IP notice: some slides were selected from Drena Dobbs' and Richard Edwards' slides.

Thanks for your attention