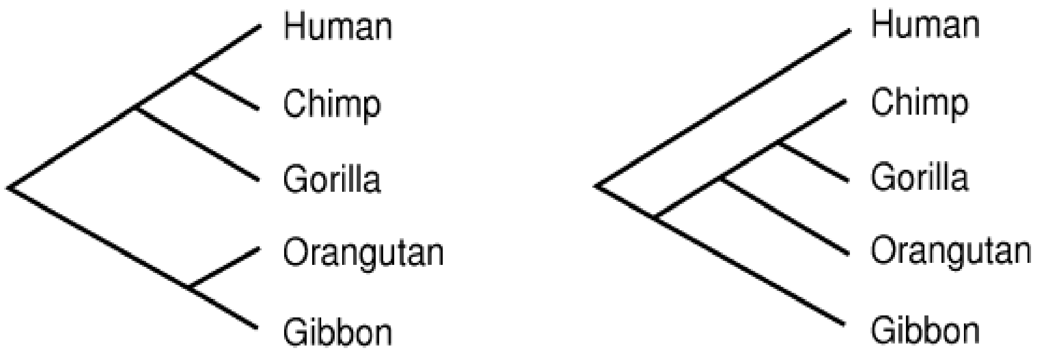
Constructing Phylogenetic Trees

Different methodologies

Relationship among various species

- What is the relationship of humans to the modern apes?
- More specifically, which of the gorilla, chimpanzee, orangutan, and gibbon are our closest evolutionary kin, or are all these apes more closely related to each other than they are to us?
- Two possible diagrams that represent more detailed versions of these competing views of hominoid evolution

Which evolutionary relationship is true?



Phylogentic tree.

Phylogeny

- *Phylogenetics is the study of the evolutionary history of living organisms.*

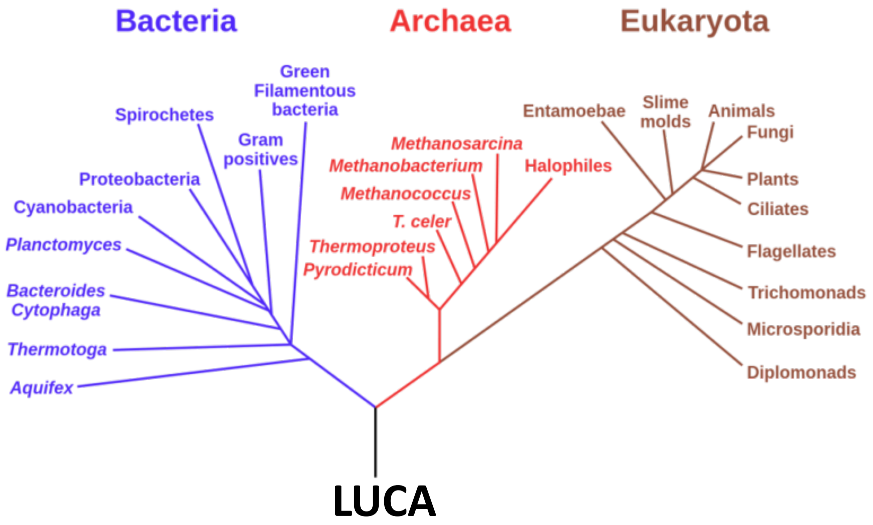
- This is mostly done using tree-like diagrams to represent pedigrees of these organisms.

- The tree branching patterns representing the evolutionary divergence are referred to as *phylogeny.*

- The purpose of *phylogeny is to reconstruct the history of life and explain the present diversity* of living creatures.

Why Phylogeny?

- **<u>Underlying principle:</u>** To group organisms according to the level of their similarity (the more similar the species are the closer they are to the common ancestor)



- Phylogenetics relies on the comparison of equivalent genes coming from several species for reconstructing evolutionary trees; it can also apply to individual genes

- Tasks that can be done using phylogenetics:
  (i) determining the closest relatives of the organism that one is interested.
  (ii) discovering the function of a gene
  (iii) retracing the origin of a gene

Phylogenetic trees--terminologies

- The source of the DNA sequence is referred to as a taxon (pl. taxa). The source may be human, ape, organtuan etc.

- An equivalent term in common use is operational taxonomic unit, usually abbreviated as **OTU**.

- The diagram consisting of line segments represents the evolutionary history of the taxa.

- Each of the line segments in the diagram is referred to as an **edge**. A diagramin which there are no loops formed by the edges, is called a **tree**.

Rooted and unrooted trees

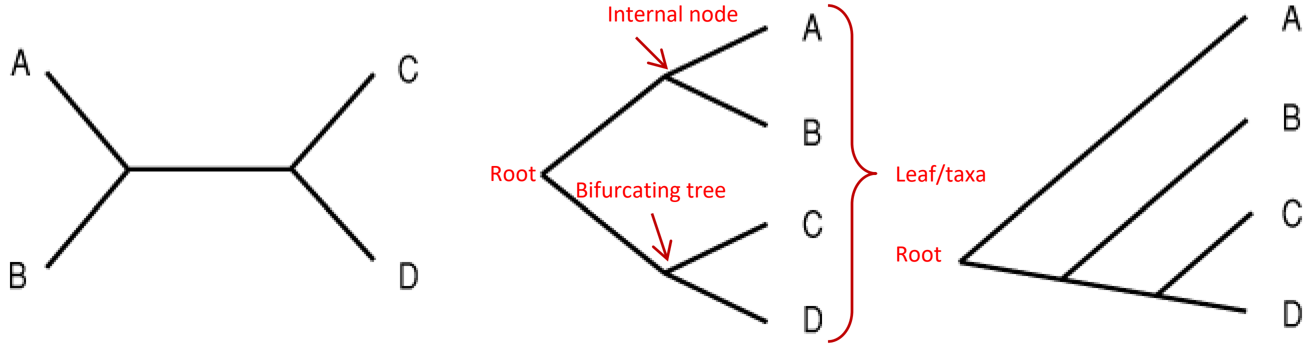
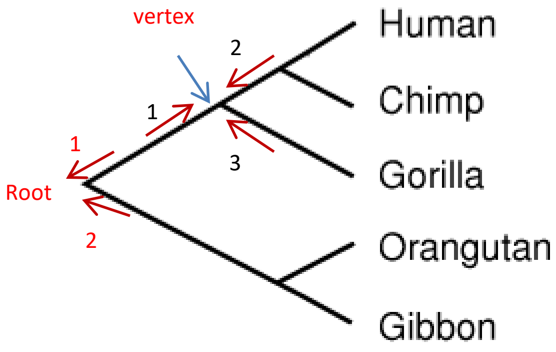


Figure 5.2. An unrooted tree (L) and two rooted versions (C,R).

Left

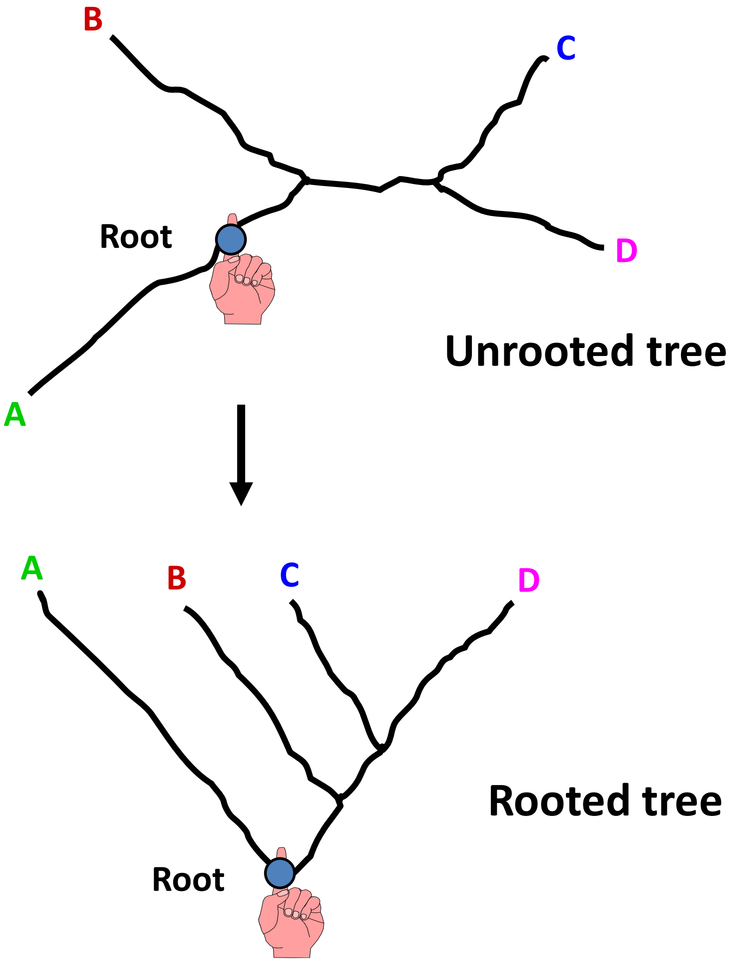
Center, Right



A tree is said to be **bifurcating** if at each interior vertex three edges meet and at the root two edges meet, as in the trees in given above.

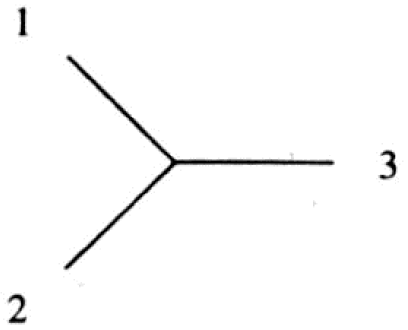
Rooting the tree:

To root a tree mentally, imagine that the tree is made of string. Grab the string at the root and tug o● it until the ends of the string (the taxa) fall opposite the root:



Note that in this rooted tree, taxon A is no more closely related to taxon B than it is to C or D.

Counting Trees



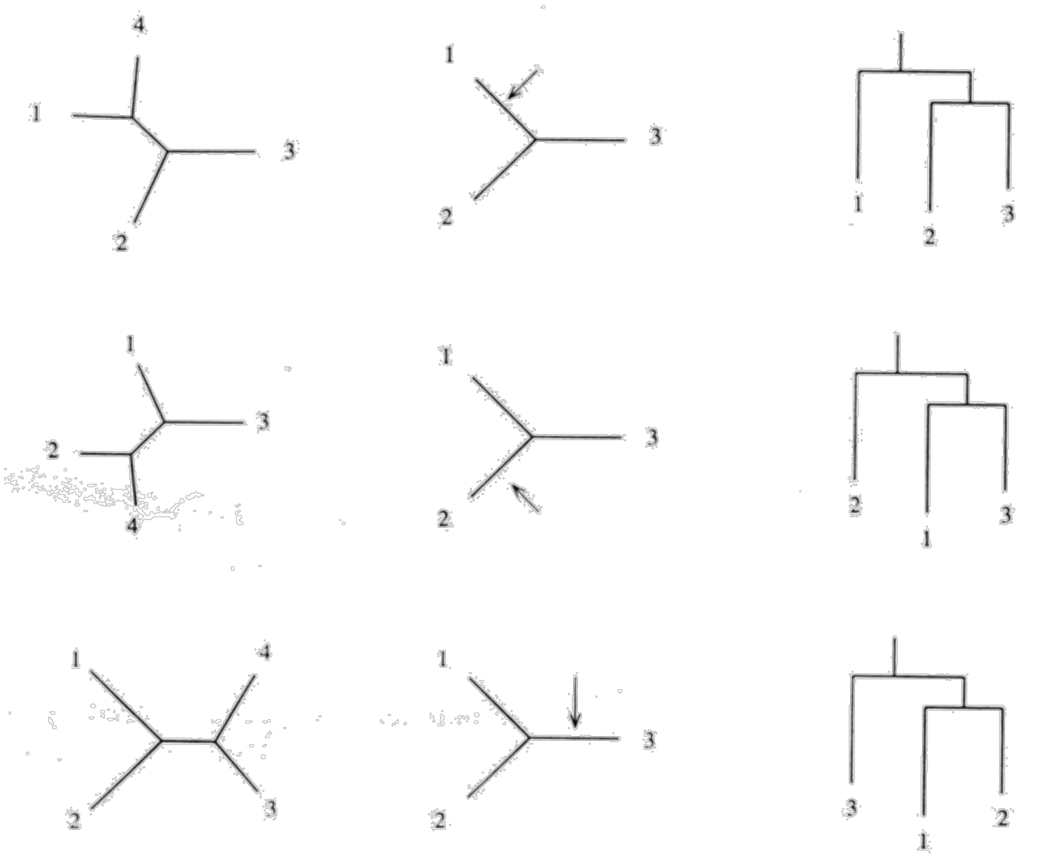
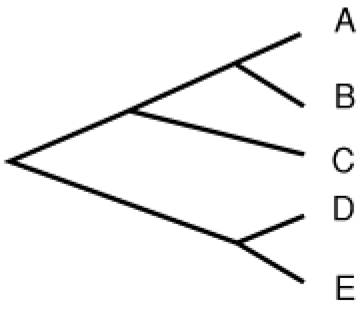
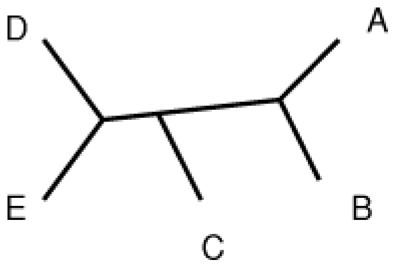


Figure 7.3 *The rooted trees (right-hand column) derived from the unrooted tree for three sequences by picking different edges as positions for the root (arrows).*

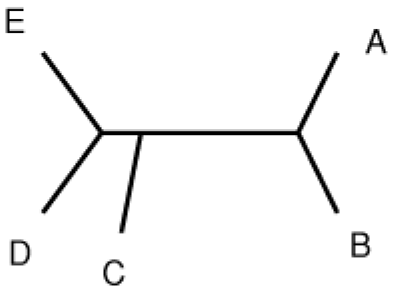
Topological trees-
do not specify the lengths of edges



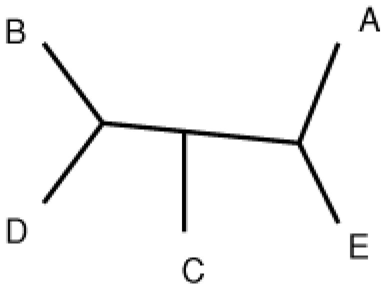
Tree T_1



Tree T_2



Tree T_3



Tree T_4

Trees T1 , T2, and T3 are all topologically the same as **unrooted Trees**.

T1,T2 and T3 could be deformed into the other ones without either cutting or gluing pieces of it together.

Tree T4 , on the other hand, is topologically distinct from T1, T2, and T3.

Metric trees
each edge assigned a certain length.

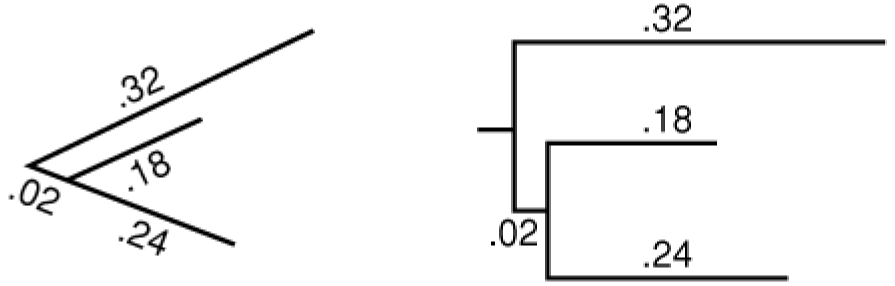


Figure 5.4. Alternate depictions of the same metric tree.

Molecular clock: It means that the mutation rate is constant for all lineages under consideration.

If μ denotes the mutation rate, measured in (base substitutions per site)/year, for instance, and t denotes a time in years, then the amount of mutation that will occur during this time is **$d = \mu t$** base substitutions per site.

Molecular clock and its importance

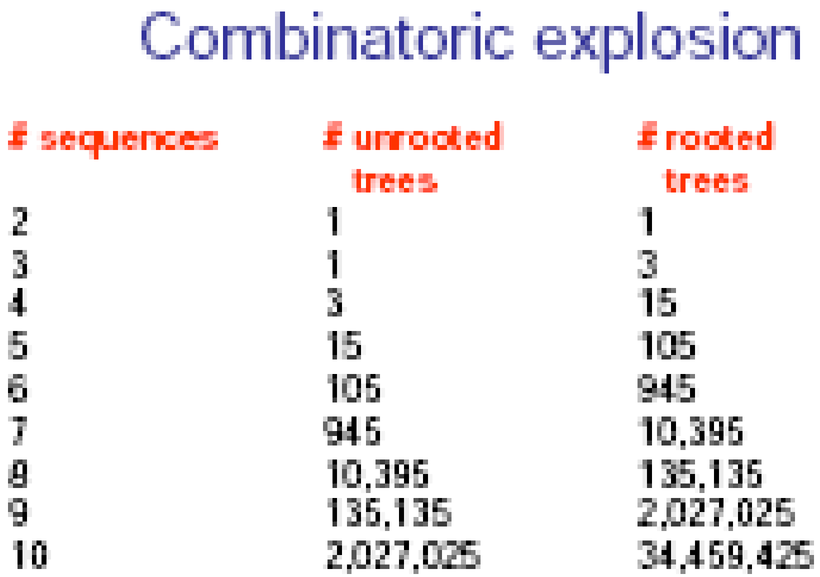
- **Molecular clock:** the mutation rate is constant for all lineages under consideration.

- A molecular clock means that the amount of mutation along **any edge is proportional to the elapsed time,** with the constant of proportionality being the constant rate of mutation.

- Under the assumption of a molecular clock, then, whether we draw edge lengths representing amount of mutation or elapsed time, we draw exactly the same figure, up to scaling by this constant.

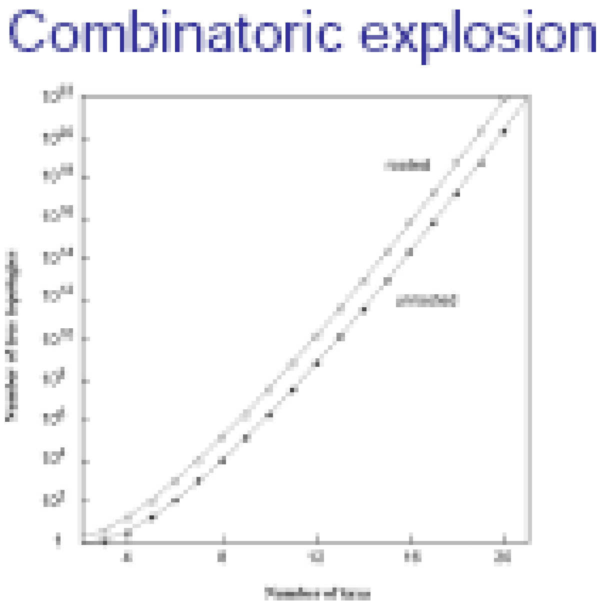
How many trees?

- *If n is the number of taxa or sequences, then for*
- *Number of unrooted trees = (2n-5)! / 2^{n-3} (n-3)!*
- *Number of rooted trees = (2n-3)! / 2^{n-2}(n-2)!*



Combinatoric explosion

# sequences	# unrooted trees	# rooted trees
2	1	1
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10,395
8	10,395	135,135
9	135,135	2,027,025
10	2,027,025	34,459,425



Combinatoric explosion

Tree Construction: Distance Methods -- Basics

- We use DNA sequences to construct the phylogeny for comparing different species.

- **Aim:** To find the evolutionary relationship of four species: S1, S2, S3, and S4 from their DNA sequences.

- Then determine the distances between the taxa. Get their table. Use appropriate (JC/K2P/K3P) model to get the distance.

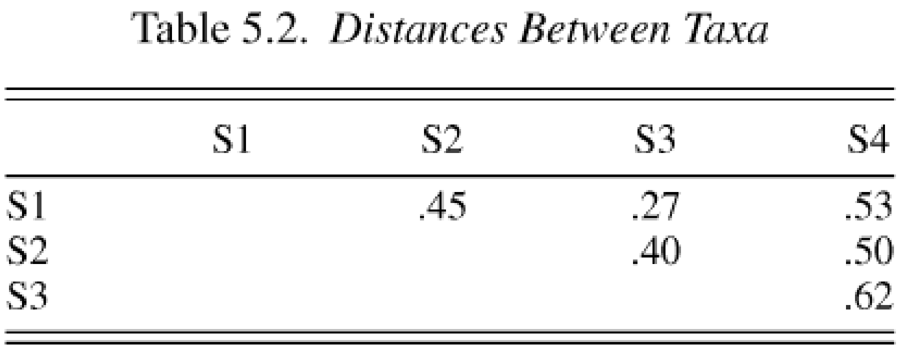
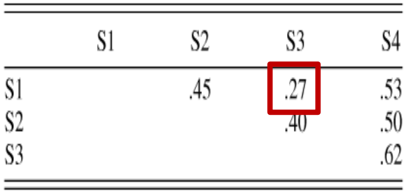


Table 5.2. *Distances Between Taxa*

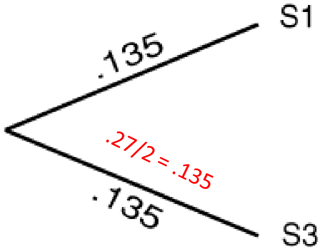
	S1	S2	S3	S4
S1		.45	.27	.53
S2			.40	.50
S3				.62

Unweighted pair-group method with arithmetic means (UPGMA)

Step-1: The shortest distance is between taxas S1 and S3 and Combine it into a group.



	S1	S2	S3	S4
S1		.45	.27	.53
S2			.40	.50
S3				.62



Step-2:

(a) Combine S1 and S3 into a group.

(b) Then **average the distances** of S1-S3 to each different taxon to get the distance from the group to that taxon.

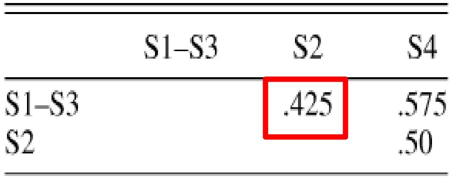
For example,

d(S1-S3), S2 = d(S1,S2) + d(S3,S2)/2
= (.45 + .40)/2 = .425,

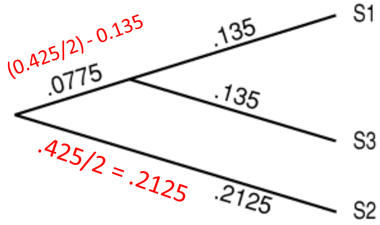
d(S1-S3), S4 = d(S1,S4) + d(S3,S4)/2
= (.53 + .62)/2 = .575.

Step-3:

In the collapsed table. the closest taxa are S1--S3 and S2, which are .425 apart .

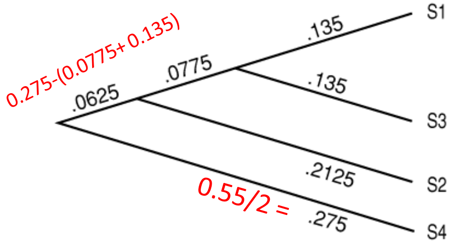


	S1–S3	S2	S4
S1–S3		.425	.575
S2			.50



Step-4: Again combining taxa, we form a group S1--S2--S3, and compute its distance from S4 by averaging the original distances from S4.

	S1-S3-S2	S4
S1-S3-S2		.5375 (0.55?)



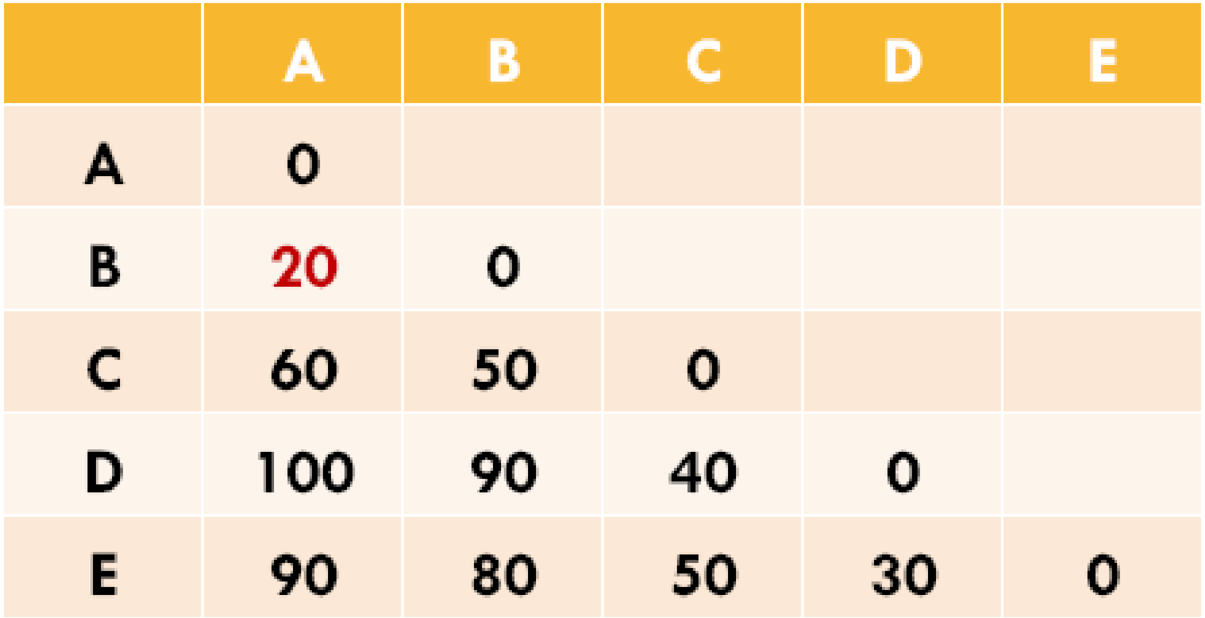
d(S1-S3,S2), S4 = d(S1-S3,S4) + d(S2,S4)/2
= (0.575 + 0.5)/2 = 0.5375

(or)

d(S1-S3,S2), S4 = (d(S1,S4) + d(S2,S4) + d(S3,S4))/3
= (0.53 + 0.5 + 0.62)/3 = 0.55

Exercise-In class

Construct the phylogenetic tree for the hypothetical taxa’s A-E given below.



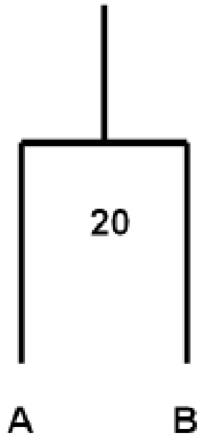
	A	B	C	D	E
A	0				
B	20	0			
C	60	50	0		
D	100	90	40	0	
E	90	80	50	30	0

https://www.pellegrini.mcdb.ucla.edu/wp-content/uploads/sites/21/2017/07/week-3c-Phylogenetic_Tree_ConstructionMai-copy.pdf

In class exercise –Solution-Step-1

Step-1

	A	B	C	D	E
A	0				
B	20	0			
C	60	50	0		
D	100	90	40	0	
E	90	80	50	30	0

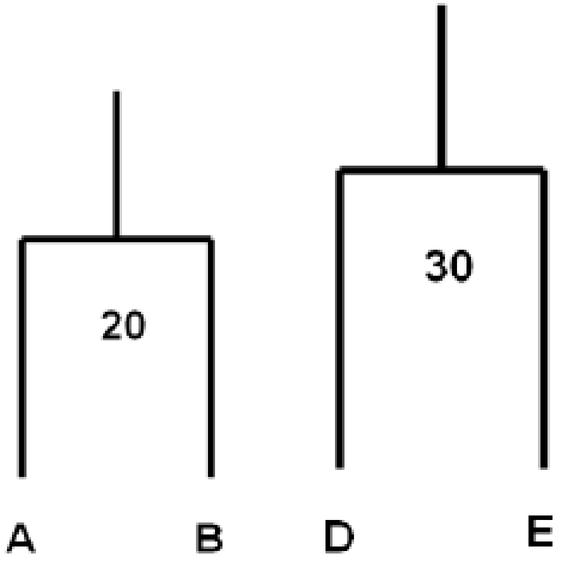


- New average distance between AB and C is:
  - C to AB = (60 + 50) / 2 = 55
- Distance between D to AB is:
  - D to AB = (100 + 90) / 2 = 95
- Distance between E to AB is:
  - E to AB = (90 + 80) / 2 = 85

Step-2: Continued…

Step-2

	AB	C	D	E
AB	0			
C	55	0		
D	95	40	0	
E	85	50	30	0

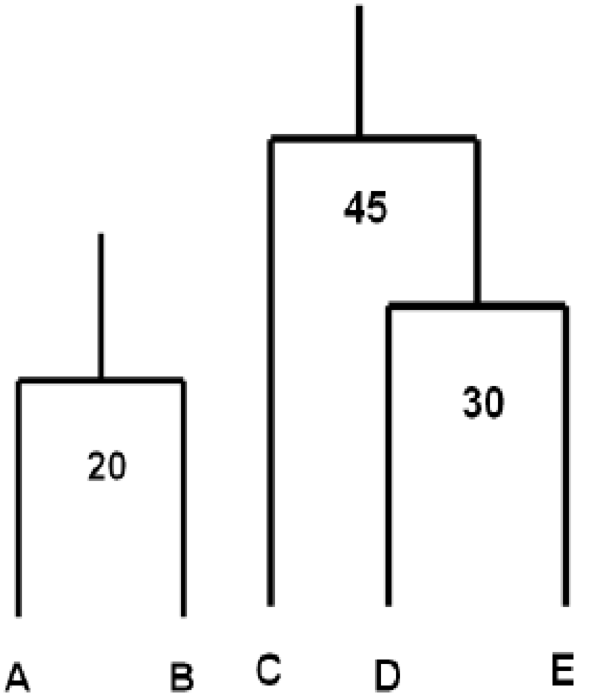


- New average distance between AB and DE is:
  - AB to DE = (95 + 85) / 2 = 90

Step-3: Continued..

Step-3

	AB	C	DE
AB	0		
C	55	0	
DE	90	45	0

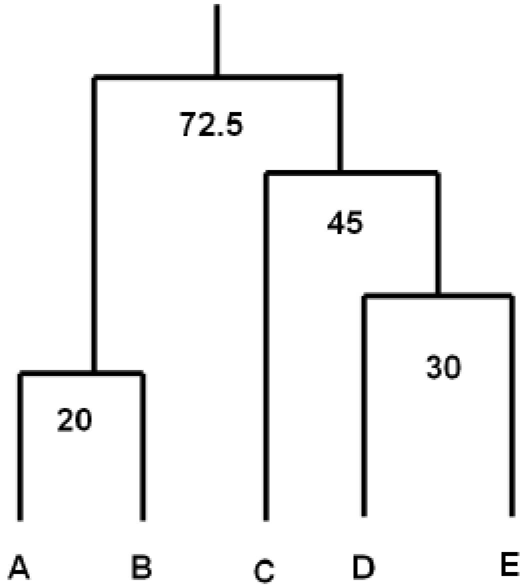


- □ New Average distance between CDE and AB is:
  - □ CDE to AB = (90 + 55) / 2 = 72.5

Step-4-Final

Step-4 and final

	AB	CDE
AB	0	
CDE	72.5	0



□ There are only two clusters. so this completes the calculation!

When UPGMA fails?

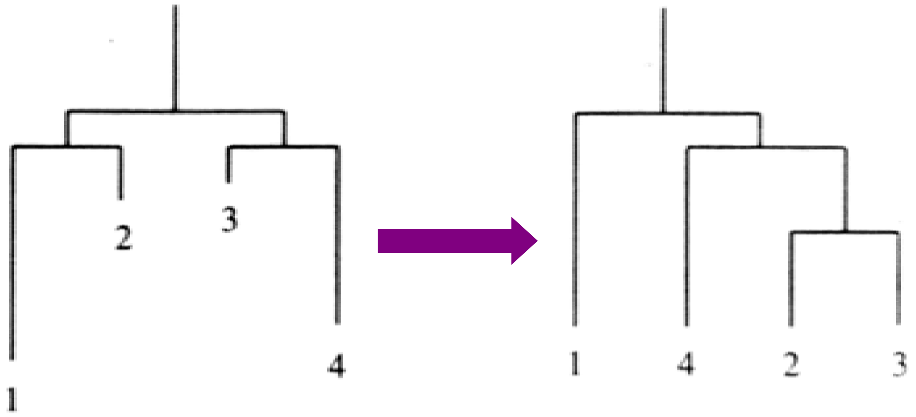


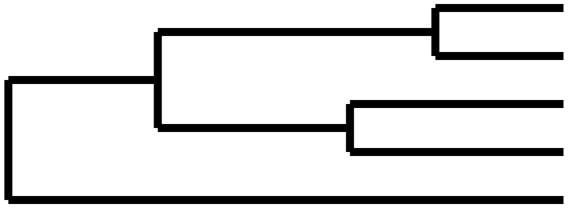
Figure 7.5 *A tree (left) that is reconstructed incorrectly by UPGMA (right).*

- The closest leaves are not neighboring leaves; they do not have a common parent node.

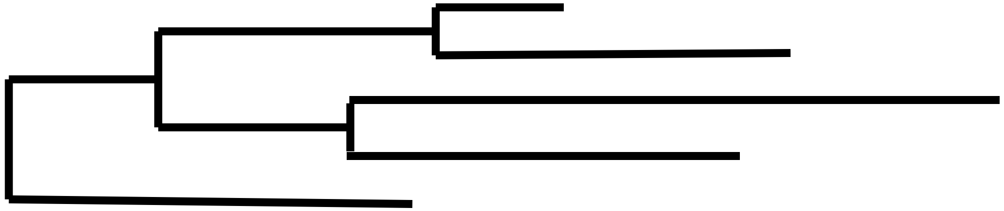
- A test of whether reconstruction is likely to be correct is the <u>ultrametric</u> condition

- A distance measures are ultrametric if
  either all three distances are equal $d_{ij} = d_{ik} = d_{jk}$ (or)
  two of them are equal and one is smaller: $d_{jk} < d_{ij} = d_{ik}$

Evolutionary clock speeds



Uniform clock: **Ultrametric distances** lead to identical distances from root to leaves

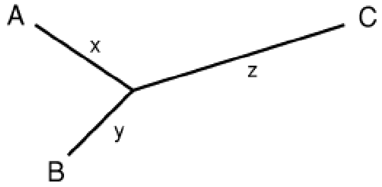


Non-uniform evolutionary clock: leaves have different distances to the root -- an important property is that of **additive trees**.

These are trees where the distance between any pair of leaves is the sum of the lengths of edges connecting them. Such trees obey the so-called **4-point condition** (Neighbour-Joining algorithm).

Fitch-Margoliash algorithm

- Consider 3 taxa on an unrooted tree



- Assign lengths (*x, y, z*) to the edges to fit the data exactly.

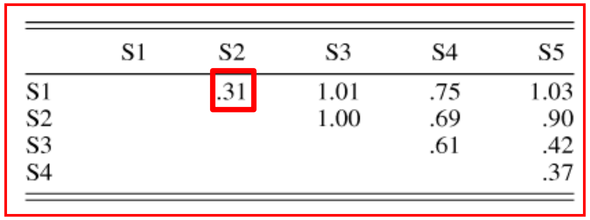
$$\begin{aligned} x + y \quad\quad &= d_{AB} \\ x + \quad\quad z &= d_{AC} \\ y + z &= d_{BC} \end{aligned}$$

- Solve these equations either by writing the system as a matrix equation and finding an inverse.

- The solution for above equation is called the **3-point formulas** for fitting taxa to the tree and its

$$\begin{aligned} x &= (d_{AB} + d_{AC} - d_{BC})/2, \\ y &= (d_{AB} + d_{BC} - d_{AC})/2, \\ z &= (d_{AC} + d_{BC} - d_{AB})/2. \end{aligned}$$

Example-Fitch-Margoliash algorithm



	S1	S2	S3	S4	S5
S1		.31	1.01	.75	1.03
S2			1.00	.69	.90
S3				.61	.42
S4					.37

- As **we did with UPGMA**, the closest pair of taxa to join i.e., from the distance table, S1 and S2 are the first pair to join.

- Reduce to the 3-taxa case by combining all other taxa into a group (leave S1 & S2). i.e., introduce the group S3--S4--S5.

- Find the distance from each of S1 and S2 to the group by averaging their distances to each

  d(S1 to S3--S4--S5) = (1.01 + .75 + 1.03)/3 = .93,

  d(S2 to S3--S4--S5) = (1.00 + .69 + .90)/3 = .863.

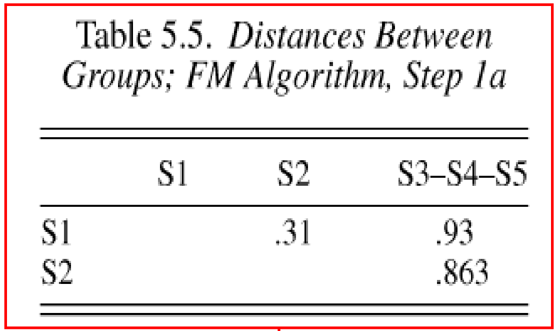
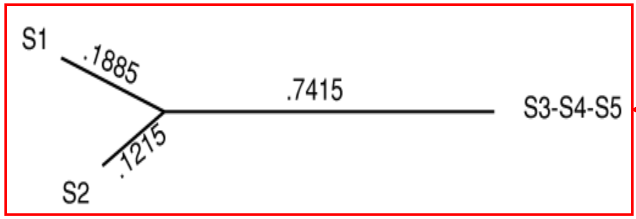


Table 5.5. *Distances Between Groups; FM Algorithm, Step 1a*

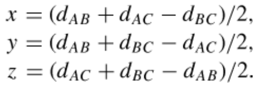
	S1	S2	S3–S4–S5
S1		.31	.93
S2			.863

- Three taxa is in this table. So exactly fit the data to the tree using the 3-point formulas to produce unequal distances



Check it !

Calculate distance using 3-point formula



$$x = (d_{AB} + d_{AC} - d_{BC})/2,$$
$$y = (d_{AB} + d_{BC} - d_{AC})/2,$$
$$z = (d_{AC} + d_{BC} - d_{AB})/2.$$

Continued….

- Keep only the edges ending at S1 and S2 in and return to the original data.

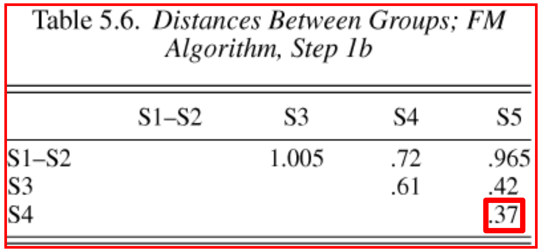


Table 5.6. *Distances Between Groups; FM Algorithm, Step 1b*

	S1–S2	S3	S4	S5
S1–S2		1.005	.72	.965
S3			.61	.42
S4				.37

- As S1 and S2 are joined, use UPGMA method to generate the distance table.
- Look for the closest pair (now S4 and S5) and join them in a similar manner.
- Then combine everything but S4 and S5 into a single temporary group S1--S2--S3
- Compute

  d(S4, S1--S2--S3) = (.75 + .69 + .61)/3 = .683 and

  d(S5, S1--S2--S3) = (1.03 + .90 + .42)/3 = .783.

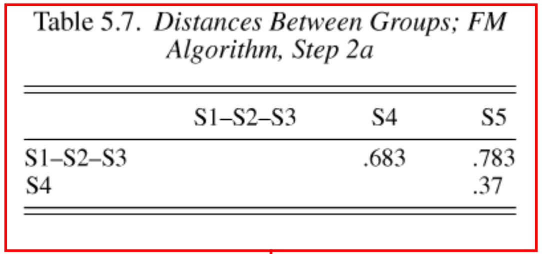
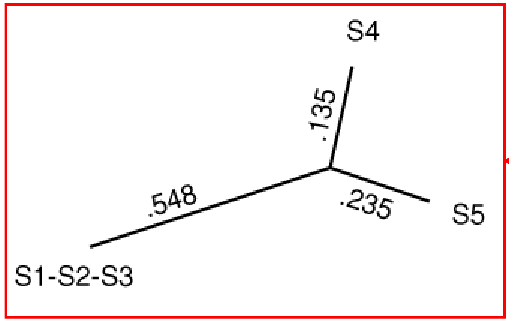


Table 5.7. *Distances Between Groups; FM Algorithm, Step 2a*

	S1–S2–S3	S4	S5
S1–S2–S3		.683	.783
S4			.37

- Applying the 3-point formulas



Calculate distance using 3-point formula

$$x = (d_{AB} + d_{AC} - d_{BC})/2,$$
$$y = (d_{AB} + d_{BC} - d_{AC})/2,$$
$$z = (d_{AC} + d_{BC} - d_{AB})/2.$$

Continued…

- Two groups are joined;

  S1--S2 and S4—S5 & S3 and S4--S5.

- Compute a new table containing these two groups.
- i.e., d(S1--S2, S4--S5) = (d(S1,S4) + d(S1,S5)
- + d(S2,S4) + d(S2--S5))/4

  = (.75 + 1.03 + .69 +.90)/4 = .8425 and

  d(S3, S4--S5) = (.61 + .42)/2 = .515.

Original table

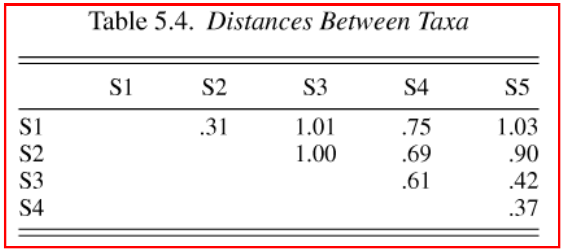


Table 5.4. *Distances Between Taxa*

	S1	S2	S3	S4	S5
S1		.31	1.01	.75	1.03
S2			1.00	.69	.90
S3				.61	.42
S4					.37

New table

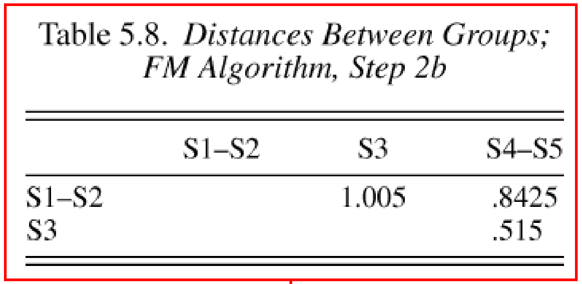
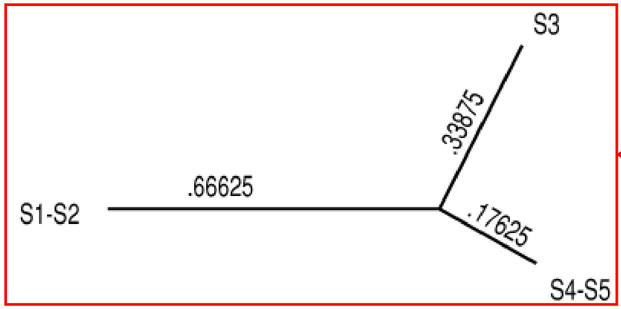


Table 5.8. *Distances Between Groups; FM Algorithm, Step 2b*

	S1–S2	S3	S4–S5
S1–S2		1.005	.8425
S3			.515

- Already d(S1--S2, S3) is computed.

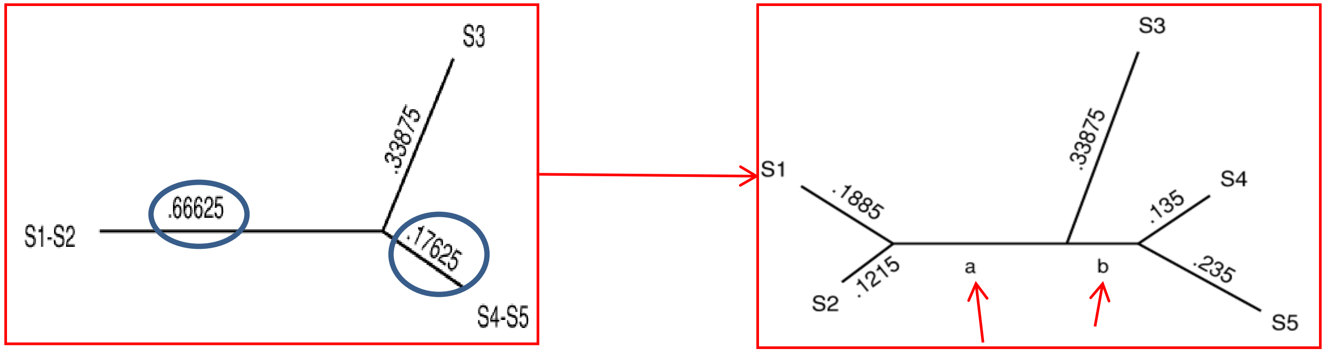


Calculate distance using 3-point formula

$$x = (d_{AB} + d_{AC} - d_{BC})/2,$$
$$y = (d_{AB} + d_{BC} - d_{AC})/2,$$
$$z = (d_{AC} + d_{BC} - d_{AB})/2.$$

Final step

- Putting together



How to compute a & b?

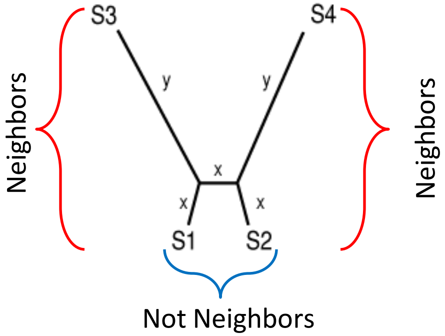
S1 and S2 are on average (.1885 + .1215)/2 = .155 from the vertex joining them
S4 and S5 are on average (.135 + .235)/2 = .185 from the vertex joining them.

Then to assign lengths to the remaining sides a & b, its done as follows:

a = 0.66625 - 0.155 = 0.51125 and
b = 0.17625- 0.185 = - 0.00875 (taken as 0, since its negative)

Neighbor-Joining (N-J) algorithm

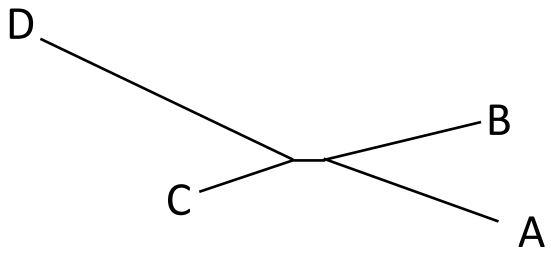
- UPGMA and the Fitch-Margoliash algorithm are seldom used for tree construction.
- There are better distance method that tends to perform better than either. One popular method is the Neighbor Joining (N-J) algorithm.
- Both UPGMA, or the Fitch-Margoliash algorithm, might be flawed in terms of determining the minimal distance between the taxa.
- For example, consider the metric tree with 4 taxa. x and y represent specific lengths, with x << y.
- Here, vertices S1 and S3 in this tree are neighbors, because the edges leading from them join.
- Similarly, S2 and S4 are neighbors, but S1 and S2 are not.
- The very first joining step will be incorrect. If we join non-neighbors, true tree will not be recovered.



- The essence of the problem is that if no molecular clock is operating, then the closest taxa by distance are not necessarily neighbors on the tree.

Neighbor-Joining (N-J) algorithm

- Another algorithm that works by clustering the sequences
- Does not assume molecular clock



- N-J trees are ***unrooted***
- N-J assumes ***additivity***

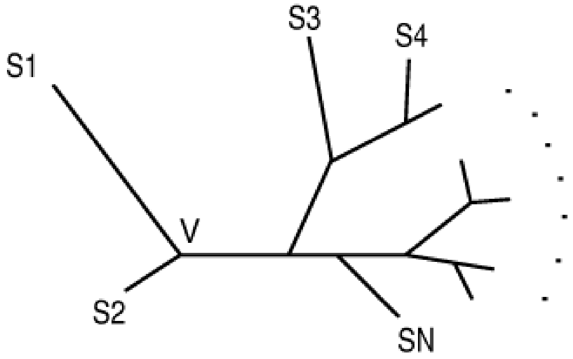
Def. Edge lengths are said to be additive if the distance between any pair of leaves is the sum of lengths of the edges on the path connecting them

- Method uses an approximate algorithm, where the tree is built by finding a pair of neighboring leaves *i* and *j* that minimize the length of the tree. Finally neighboring leaves are joined.
- Running time $O(n^2)$

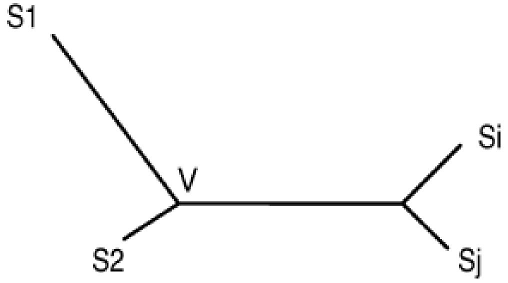
Need new distance measure

- A more sophisticated criterion required to join the taxa.
- Imagine a tree in which taxa S1 and S2 are neighbors joined at vertex V , with V somehow joined to the remaining taxa S3, S4, ...,SN , as shown below.

$$d(\mathrm{S}1, \mathrm{S}2) + d(\mathrm{S}i, \mathrm{S}j) < d(\mathrm{S}1, \mathrm{S}i) + d(\mathrm{S}2, \mathrm{S}j),$$

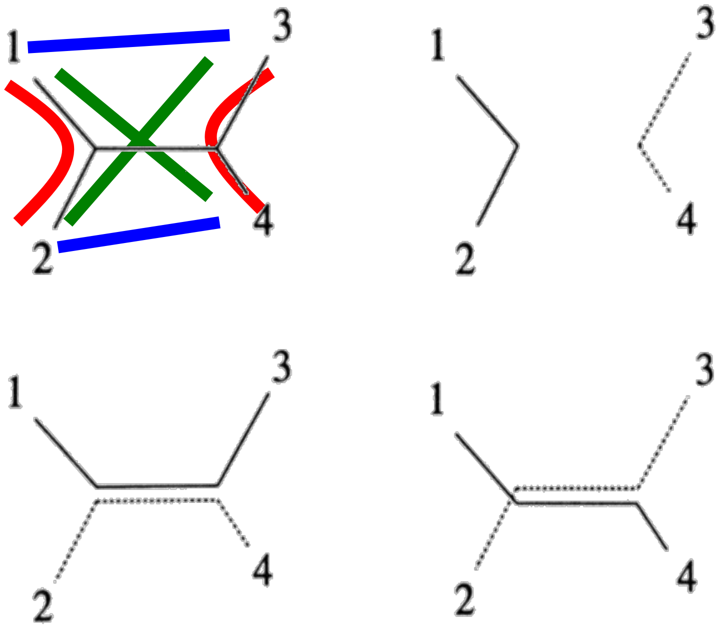


- For every *i, j* = 3, 4,...N , our metric tree would include a subtree like the one shown below



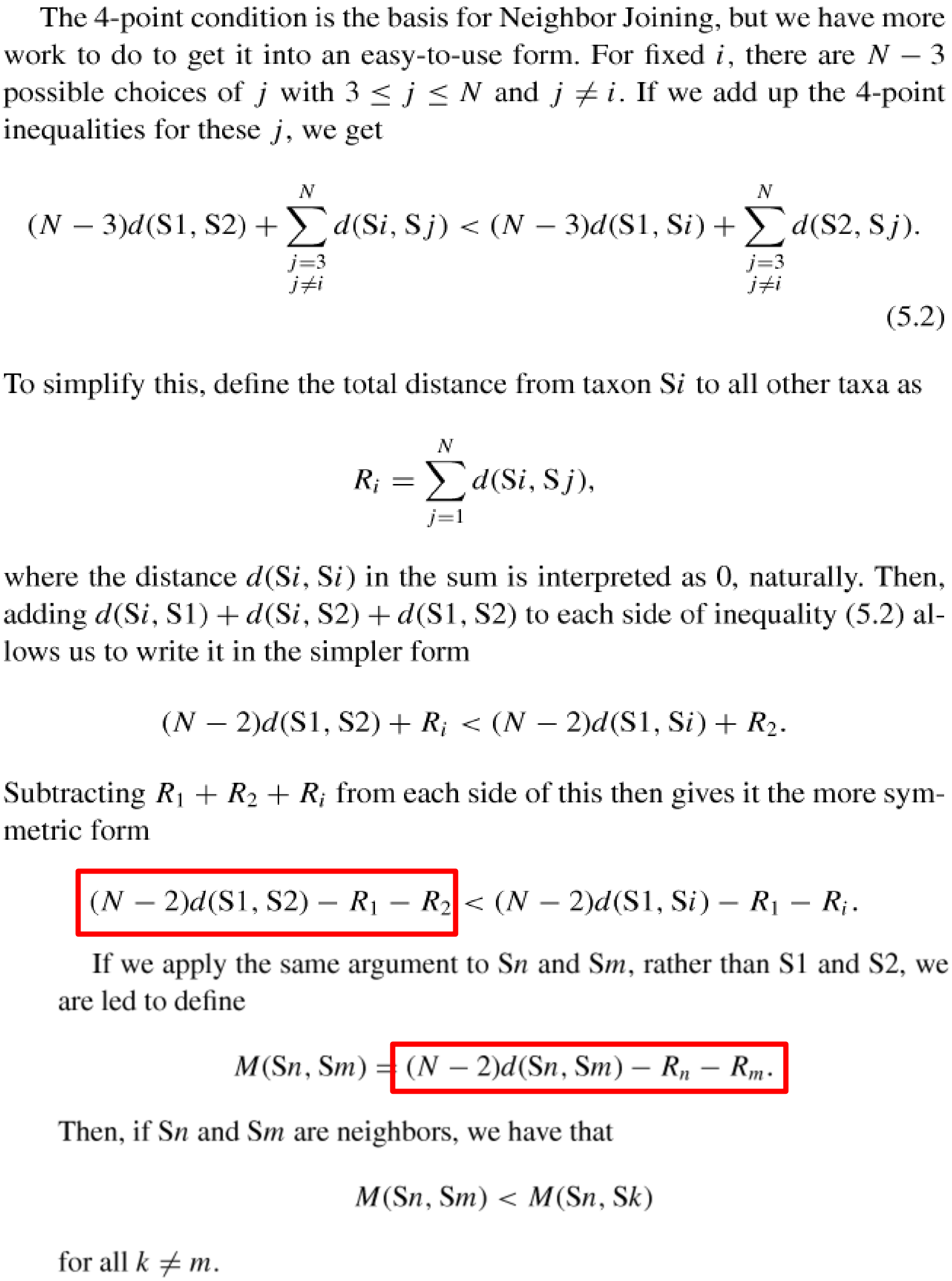
$$d(\mathrm{S}1, \mathrm{S}2) + d(\mathrm{S}i, \mathrm{S}j) < d(\mathrm{S}1, \mathrm{S}i) + d(\mathrm{S}2, \mathrm{S}j),$$

Need new distance measure 4-point condition



Additivity means that two of the summed lengths $d_{12} + d_{34}$, $d_{13} + d_{24}$, $d_{14} + d_{23}$ *must be larger than the third and equal in size. This holds if the pairwise distances are obtained by summing edge lengths, as the diagrams show.*

4-point condition- Easy-to-use-form



The 4-point condition is the basis for Neighbor Joining, but we have more work to do to get it into an easy-to-use form. For fixed i, there are $N-3$ possible choices of j with $3 \le j \le N$ and $j \ne i$. If we add up the 4-point inequalities for these j, we get

$$(N-3)d(\mathrm{S}1,\mathrm{S}2) + \sum_{\substack{j=3\\ j\neq i}}^{N} d(\mathrm{S}i,\mathrm{S}j) < (N-3)d(\mathrm{S}1,\mathrm{S}i) + \sum_{\substack{j=3\\ j\neq i}}^{N} d(\mathrm{S}2,\mathrm{S}j). \tag{5.2}$$

To simplify this, define the total distance from taxon Si to all other taxa as

$$R_i = \sum_{j=1}^{N} d(\mathrm{S}i,\mathrm{S}j),$$

where the distance $d(\mathrm{S}i,\mathrm{S}i)$ in the sum is interpreted as 0, naturally. Then, adding $d(\mathrm{S}i,\mathrm{S}1) + d(\mathrm{S}i,\mathrm{S}2) + d(\mathrm{S}1,\mathrm{S}2)$ to each side of inequality (5.2) allows us to write it in the simpler form

$$(N-2)d(\mathrm{S}1,\mathrm{S}2) + R_i < (N-2)d(\mathrm{S}1,\mathrm{S}i) + R_2.$$

Subtracting $R_1 + R_2 + R_i$ from each side of this then gives it the more symmetric form

$$\boxed{(N-2)d(\mathrm{S}1,\mathrm{S}2) - R_1 - R_2} < (N-2)d(\mathrm{S}1,\mathrm{S}i) - R_1 - R_i.$$

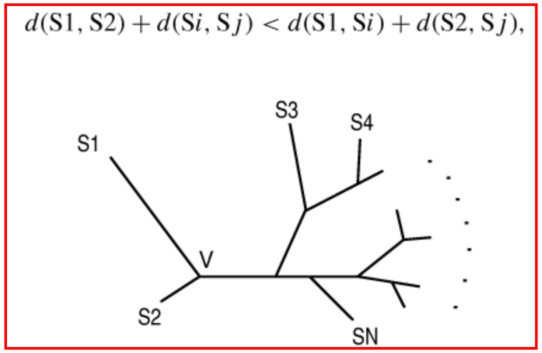
If we apply the same argument to Sn and Sm, rather than S1 and S2, we are led to define

$$M(\mathrm{S}n,\mathrm{S}m) = \boxed{(N-2)d(\mathrm{S}n,\mathrm{S}m) - R_n - R_m.}$$

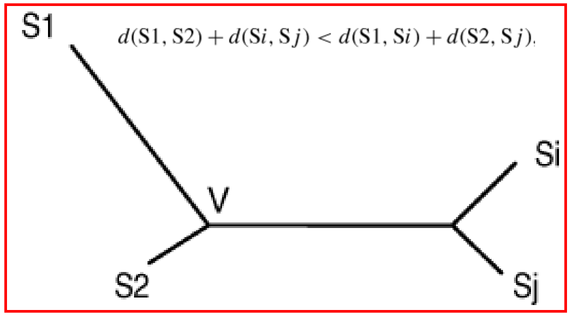
Then, if Sn and Sm are neighbors, we have that

$$M(\mathrm{S}n,\mathrm{S}m) < M(\mathrm{S}n,\mathrm{S}k)$$

for all $k \neq m$.

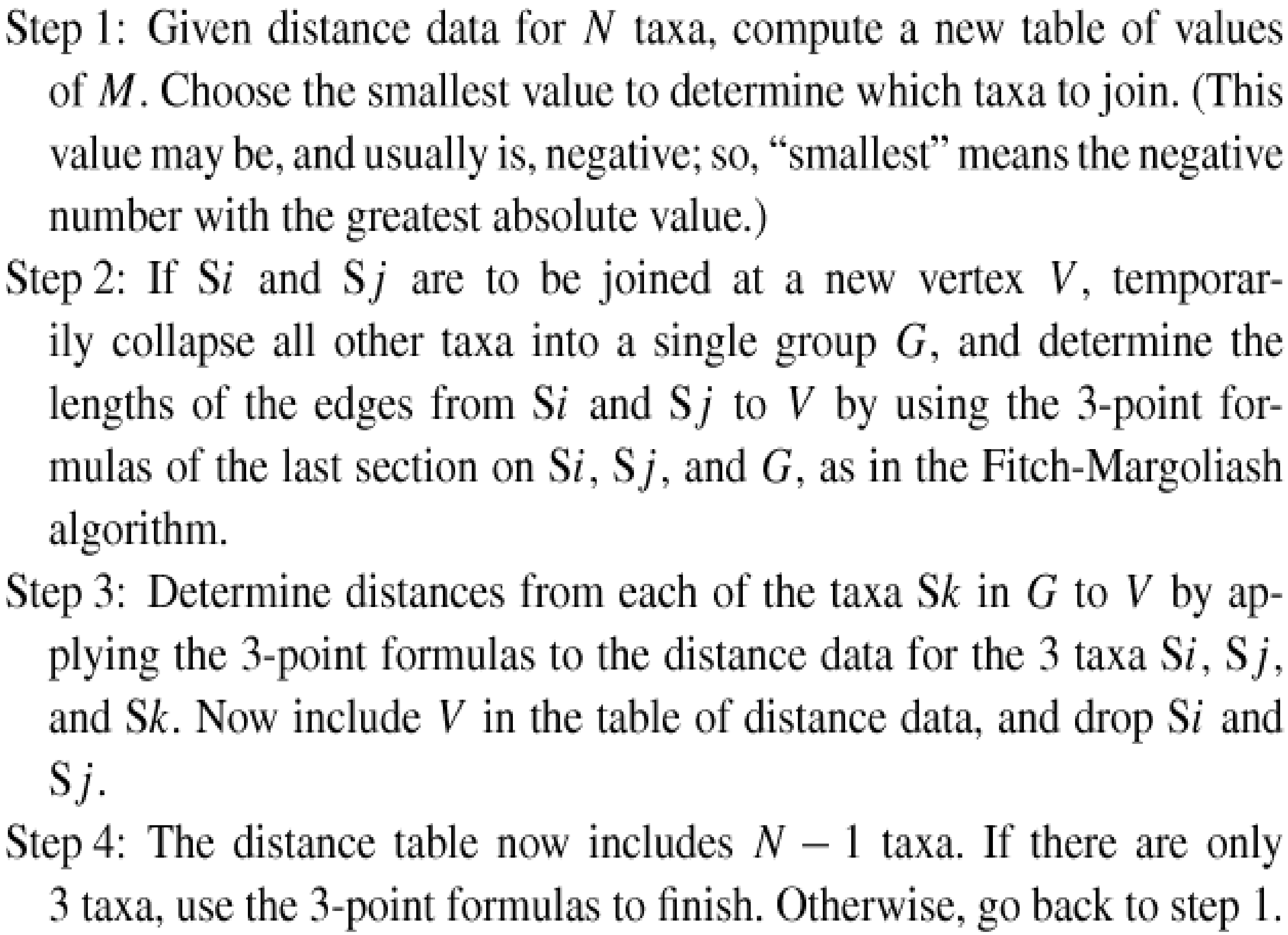


$$d(\mathrm{S}1,\mathrm{S}2) + d(\mathrm{S}i,\mathrm{S}j) < d(\mathrm{S}1,\mathrm{S}i) + d(\mathrm{S}2,\mathrm{S}j),$$



$$d(\mathrm{S}1,\mathrm{S}2) + d(\mathrm{S}i,\mathrm{S}j) < d(\mathrm{S}1,\mathrm{S}i) + d(\mathrm{S}2,\mathrm{S}j).$$

Outline of the method



Step 1: Given distance data for N taxa, compute a new table of values of M. Choose the smallest value to determine which taxa to join. (This value may be, and usually is, negative; so, “smallest” means the negative number with the greatest absolute value.)

Step 2: If Si and Sj are to be joined at a new vertex V, temporarily collapse all other taxa into a single group G, and determine the lengths of the edges from Si and Sj to V by using the 3-point formulas of the last section on Si, Sj, and G, as in the Fitch-Margoliash algorithm.

Step 3: Determine distances from each of the taxa Sk in G to V by applying the 3-point formulas to the distance data for the 3 taxa Si, Sj, and Sk. Now include V in the table of distance data, and drop Si and Sj.

Step 4: The distance table now includes $N - 1$ taxa. If there are only 3 taxa, use the 3-point formulas to finish. Otherwise, go back to step 1.

Slightly different version of algo.

- A slightly different version of algo. is provided in the coming example.

- Please check the problem-5.3.2 in the book for different way of writing the equation. Use the algo. given in the book for the coming problem.

- See whether the trees are same!!!! (**important HW!**)

An example N-J (1)

n = number of taxa =4 (A, B, C,D)

Step 1: Compute for each row in distance matrix

$$u_i = \sum_{j \neq i}^{n} \frac{d_{ij}}{(n-2)}$$

Different from given In the book.

	A	B	C	D	Step 1 - ***ui***
A	0	8	7	12	=(8+7+12)/(4-2) = 13.5
B	8	0	9	14	=(8+9+14)/(4-2)=15.5
C	7	9	0	11	=(7+9+11)/(4-2)=13.5
D	12	14	11	0	=(12+14+11)/(4-2)=18.5

Step 2: Compute (the lower-diagonal matrix) and choose the smallest (most negative). Use the form of equation below

$$d_{ij} - (u_i + u_j)$$

Different from given In the book.

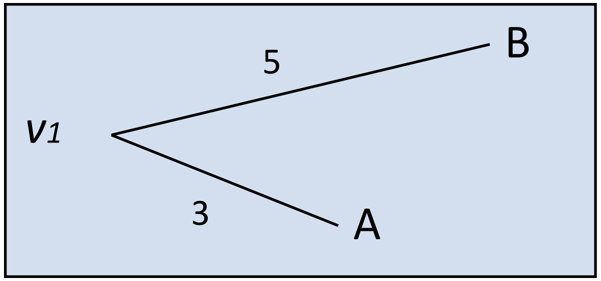
	A	B	C	D
A	0	8	7	12
B	**8-(13.5+15.5)=-21**	0	9	14
C	7-(13.5+13.5)=-20	9-(15.5+13.5)= -20	0	11
D	12-(13.5+18.5)=-20	14-(15.5+18.5)=-20	11-(13.5+18.5)=-21	0

An example N-J (2)

Step 3. Join *A* and *B* together with a new node *v1*. Compute the edge lengths, from *A* to node *v1* and from *B* to node *v1*

$$v_A = \frac{d_{AB}}{2} + \frac{(u_A - u_B)}{2} = \frac{8}{2} + \frac{(13.5 - 15.5)}{2} = 3$$

$$v_B = \frac{d_{AB}}{2} + \frac{(u_B - u_A)}{2} = \frac{8}{2} + \frac{(15.5 - 13.5)}{2} = 5$$



This is the three point formula from Fitch- Margoliash algorithm.

- **Step 4.** Compute distances between the new node *v1* and remaining items (C and D) using ***three point formula from Fitch- Margoliash algorithm***

$$d_{(AB),C} = \frac{(d_{AC} + d_{BC} - d_{AB})}{2} = \frac{7 + 9 - 8}{2} = 4$$

$$d_{(AB),D} = \frac{(d_{AD} + d_{BD} - d_{AB})}{2} = \frac{12 + 14 - 8}{2} = 9$$

An example N-J (3)

Step 5. Delete A and B from the distance matrix and replace them by new item AB

Step 6. Continue from step 1, because more than two items remain

Step 1: Compute for each row in distance matrix

$$u_i = \sum_{j \neq i}^{n} \frac{d_{ij}}{(n-2)}$$

New reduced distance matrix

	AB	C	D	Step 1 = ui
AB	0	4	9	(4+9)/1=13
C	4	0	11	(4+11)/1=15
D	9	11	0	(9+11)/1=20

Step 2: Compute and choose the smallest (the lower-diagonal matrix)

$$d_{ij} - (u_i + u_j)$$

	AB	C	D
AB	0	4	9
C	4-(13+15)=-24	0	11
D	9-(13+20)=-24	11-(15+20)=-24	0

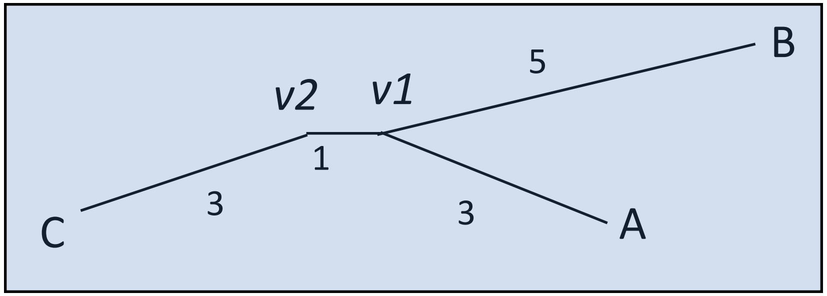
An example N-J (4)

Step 3: Join *v1* and *C* together with a new node *v2*. Compute the edge lengths, from *v1* to node *v2* and from *C* to node *v2*

	AB	C	D	Step 1 = ui
AB	0	4	9	(4+9)/1=13
C	4	0	11	(4+11)/1=15
D	9	11	0	(9+11)/1=20

$$v_1 = \frac{d_{ABC}}{2} + \frac{(u_{AB} - u_C)}{2} = \frac{4}{2} + \frac{(13-15)}{2} = 1$$

$$v_C = \frac{d_{ABC}}{2} + \frac{(u_C - u_{AB})}{2} = \frac{4}{2} + \frac{(15-13)}{2} = 3$$



Step 4: Compute distances between the new node *v2* and remaining items (D)

$$d_{(ABC),D} = \frac{(d_{ABD} + d_{CD} - d_{ABC})}{2} = \frac{9+11-4}{2} = 8$$

An example N-J (5)

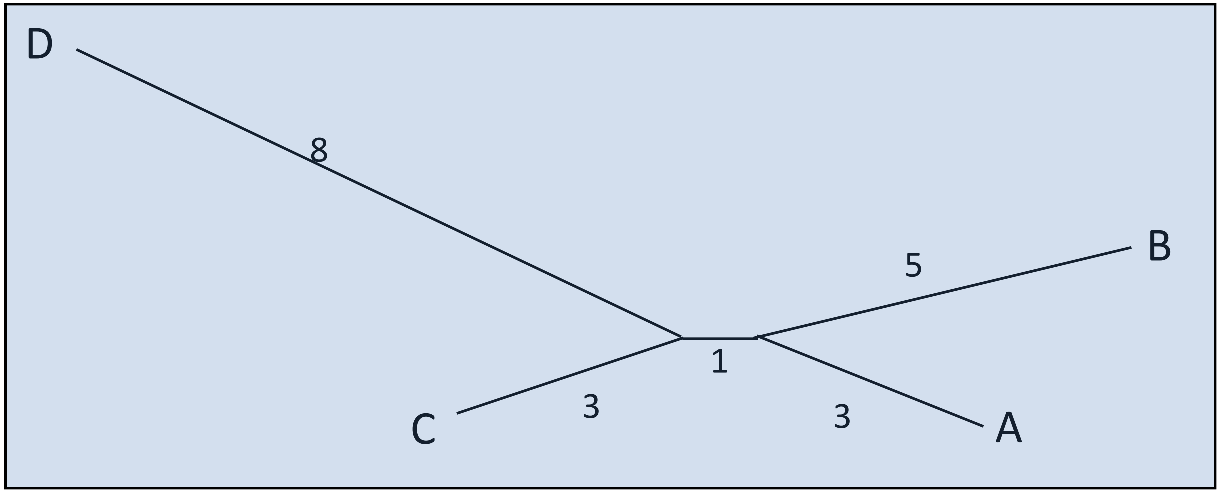
Step 5: Delete AB and C from the distance matrix and replace them by ABC

	ABC	D
ABC	0	8
D		0

Step 6 :Only two nodes remaining → connect them

Original distance matrix and final phylogenetic tree (including the edge lengths)

	A	B	C	D
A	0	**8**	**7**	**12**
B		0	**9**	14
C			0	**11**
D				0



Problem-5.3.2 (Check the difference from the earlier problem! Complete it as HW!)

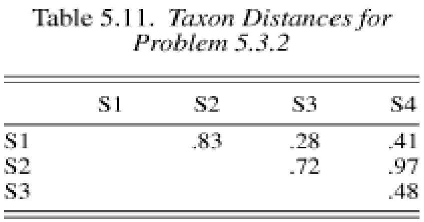
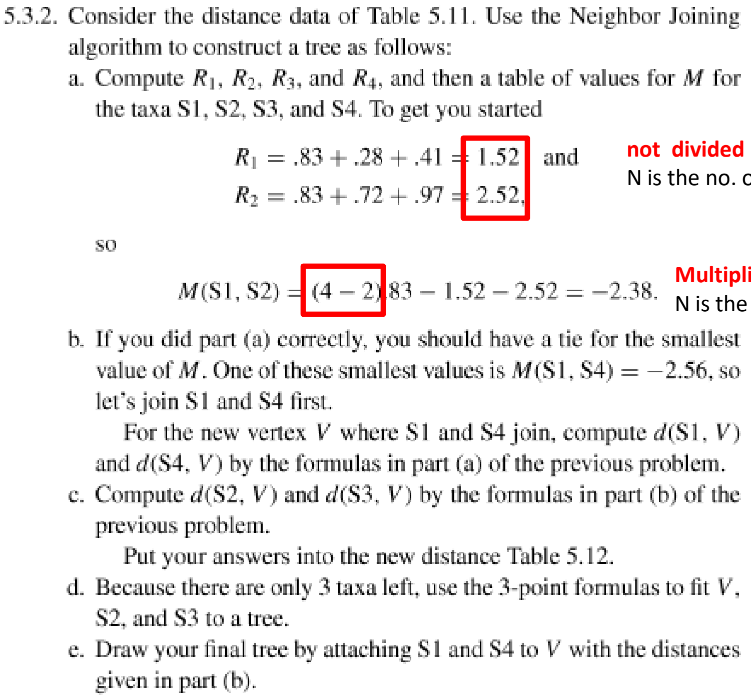


Table 5.11. *Taxon Distances for Problem 5.3.2*

	S1	S2	S3	S4
S1		.83	.28	.41
S2			.72	.97
S3				.48



5.3.2. Consider the distance data of Table 5.11. Use the Neighbor Joining algorithm to construct a tree as follows:

a. Compute R_1, R_2, R_3, and R_4, and then a table of values for M for the taxa S1, S2, S3, and S4. To get you started

$$R_1 = .83 + .28 + .41 = \boxed{1.52} \quad \text{and}$$
$$R_2 = .83 + .72 + .97 = \boxed{2.52,}$$

not divided by N-2, where N is the no. of taxa.

so

$$M(\text{S1, S2}) = \boxed{(4-2)}.83 - 1.52 - 2.52 = -2.38.$$

Multiplied by N-2, where N is the no. of taxa.

b. If you did part (a) correctly, you should have a tie for the smallest value of M. One of these smallest values is $M(\text{S1, S4}) = -2.56$, so let's join S1 and S4 first.

For the new vertex V where S1 and S4 join, compute $d(\text{S1}, V)$ and $d(\text{S4}, V)$ by the formulas in part (a) of the previous problem.

c. Compute $d(\text{S2}, V)$ and $d(\text{S3}, V)$ by the formulas in part (b) of the previous problem.

Put your answers into the new distance Table 5.12.

d. Because there are only 3 taxa left, use the 3-point formulas to fit V, S2, and S3 to a tree.

e. Draw your final tree by attaching S1 and S4 to V with the distances given in part (b).

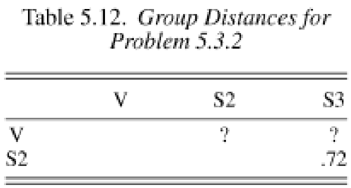


Table 5.12. *Group Distances for Problem 5.3.2*

	V	S2	S3
V		?	?
S2			.72

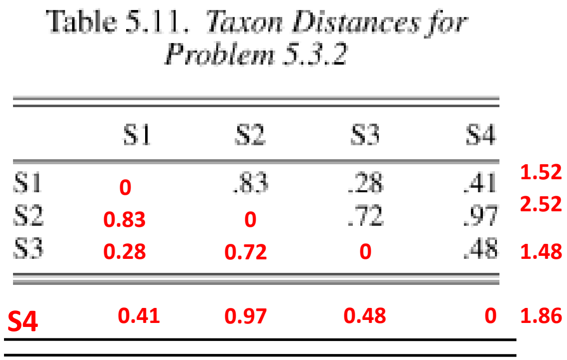


Table 5.11. *Taxon Distances for Problem 5.3.2*

	S1	S2	S3	S4	
S1	0	.83	.28	.41	1.52
S2	0.83	0	.72	.97	2.52
S3	0.28	0.72	0	.48	1.48
S4	0.41	0.97	0.48	0	1.86

Note: **(i)** Red color denotes the numbers that I added .

(ii) The summated distance in the last column **is not divided** by N-2, where N is the no. of taxa.

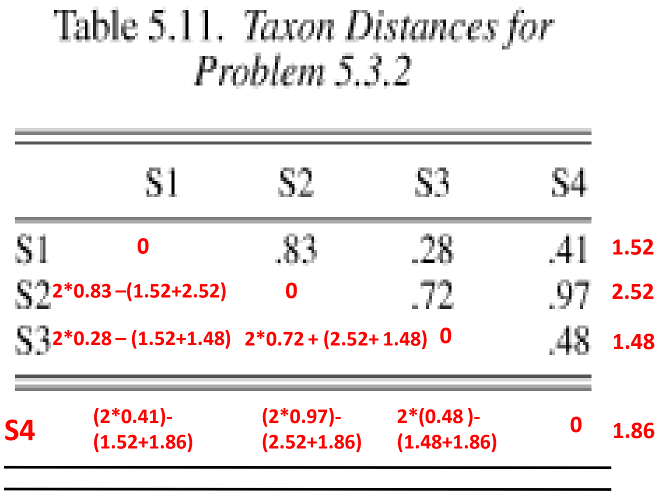


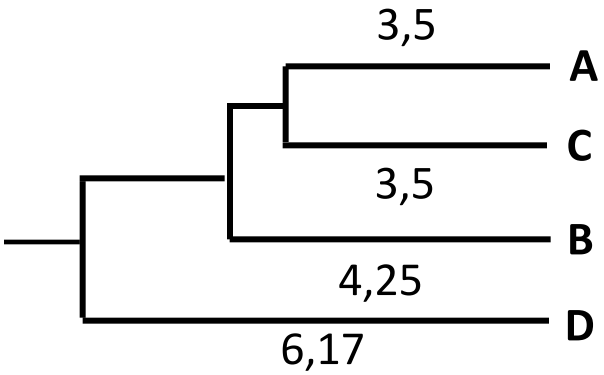
Table 5.11. *Taxon Distances for Problem 5.3.2*

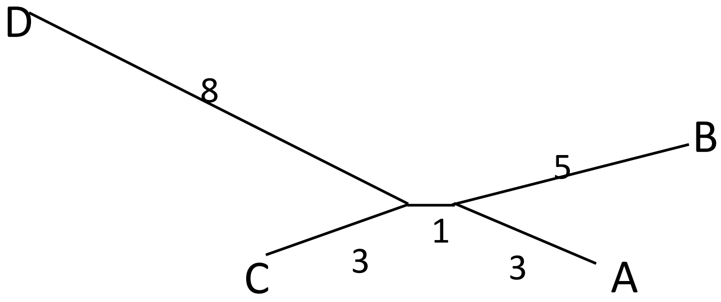
	S1	S2	S3	S4	
S1	0	.83	.28	.41	1.52
S2	2*0.83 –(1.52+2.52)	0	.72	.97	2.52
S3	2*0.28 – (1.52+1.48)	2*0.72 + (2.52+ 1.48)	0	.48	1.48
S4	(2*0.41)- (1.52+1.86)	(2*0.97)- (2.52+1.86)	2*(0.48)- (1.48+1.86)	0	1.86

Comparison

- **UPGMA**
  - The total branch length from the root up to any leaf is equal
  - Produces a rooted tree, where the root is hypothesized ancestor of the sequences in the tree
  - Suitable for closely related sequences
  - Can be used to infer phylogenies if one can assume that evolutionary rates are the same in all lineages

- **Neighbor-joining**
  - Unrooted tree, where the direction of evolution is unknown
  - Suitable for datasets with largely varying rates of evolution
  - Suitable for large datasets





Conclusion

- UPGMA method constructs a rooted phylogenetic tree correctly if there is a molecular clock with a constant rate of mutation
- UPGMA method is rarely used, because molecular clock assumption is not generally true: selection pressures vary across time periods, genes within organisms, organisms, regions within gene
- N-J method produces an unrooted tree without molecular clock hypothesis
- N-J method is one of the most popular and widely used by molecular evolutionist
- Distance methods are strongly dependent on the model of evolution used
- Sequence information is reduced when transforming sequence data into distances
- Distance methods are computationaly fast

Algorithm: Neighbor-Joining

Initialisation:

Define *T* to be the set of leaf nodes, one for each given sequence

Iteration:

Compute $u_i = \sum_{j \neq i}^{n} \frac{d_{ij}}{(n-2)}$ for each sequence, where ***n*** is the number of sequences in the distance matrix.

Pick a pair ***i*** and ***j*** (for which $d_{ij} - u_i - u_j$ is the smallest (pick randomly if several equal)

Join items ***i*** and ***j*** with a new node ***v.***

Compute the branch lengths from a new node ***v*** to items *i* and *j*

Compute the distances between new node ***v*** and remaining items

Remove ***i*** and ***j*** from the distance matrix and replace them by new node ***v***

Termination:

When only two items *i* and *j* remain, add the remaining edge between ***i*** and ***j***, with length d_{ij}

Tree Construction: Maximum Parsimony

Distance methods for tree construction begin by reducing the full DNA sequence data to a collection of pair-wise distances between taxa.

They may not use all the information in the original sequences.

Maximum Parsimony method uses the entire sequences.

Among all possible trees that might relate the taxa, it looks for the one that would require the fewest possible mutations to have occurred.

To assess the number of mutations, we never compute distances, but instead consider how mutations occur at each separate site in the sequences.

We refer to this number as the parsimony score of the tree.

One by one, consider all the trees that might relate our taxa and compute a parsimony score for each.

Then choose the tree that has least parsimony score.

Example-How to get parsimony score or the mutation count?

Suppose we look at a single site in the DNA for each of our taxa and have, for example,

S1:A, S2:T, S3:T, S4:G, S5:A.

So the parsimony Score is 3.

This is either
A or G or T

So atleast one mutation would have occurred

So mutation count here is also 1.

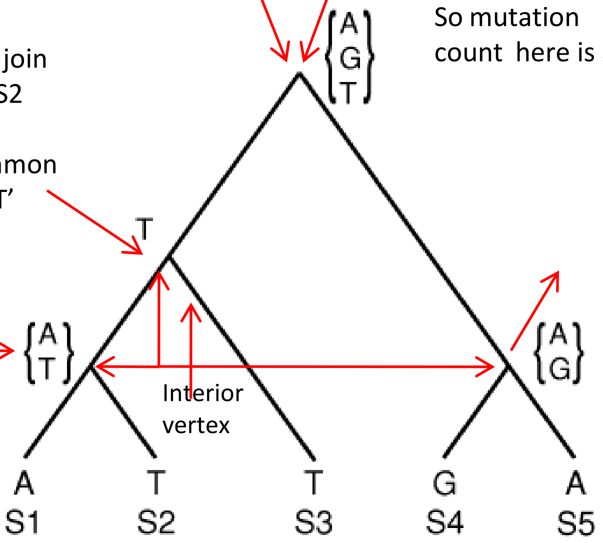
In this vertex join S3 to S1 and S2

Take the common Nucleotide ,'T'

This is either
A or T

So atleast one mutation would have occurred

So mutation count is 1



This is either
A or G

So atleast one mutation would have occurred

So mutation count is 1

Start backward:

Example for same base site with different arrangement.

What is the parsimony score here?

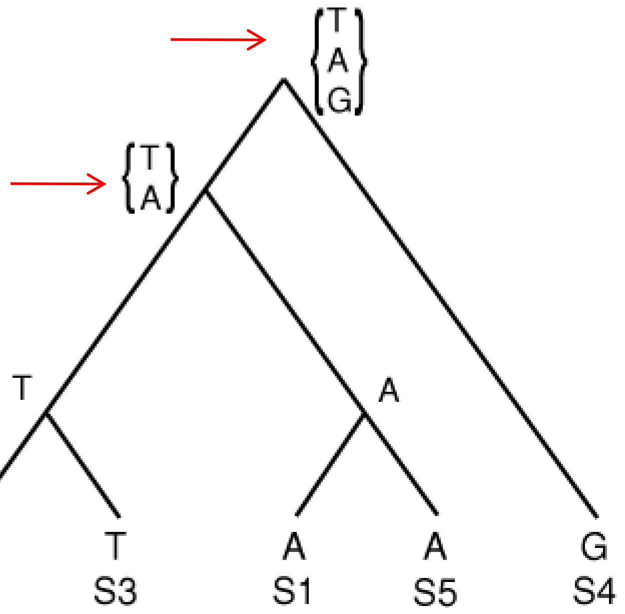
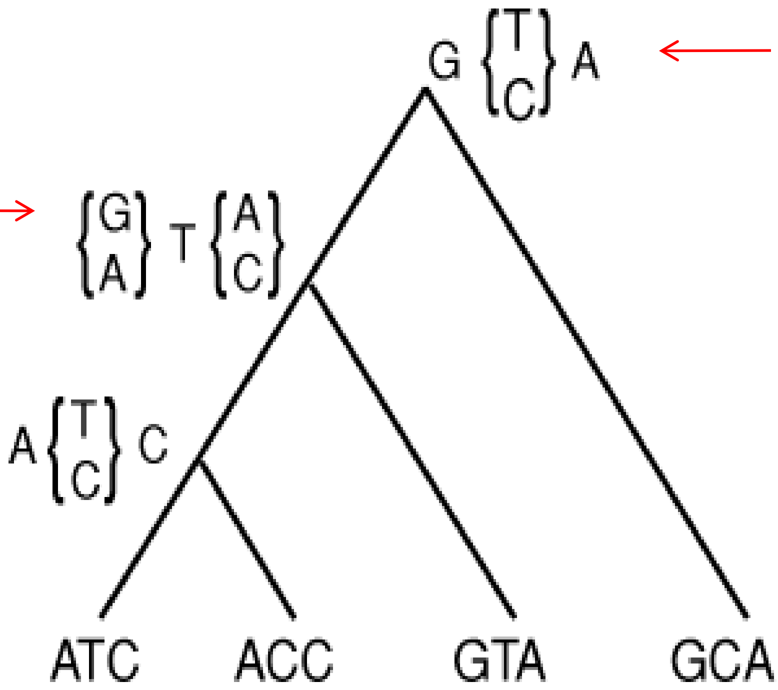


Figure 5.19. A more parsimonious tree.

Computing parsimony score for three sites.

So overall mutation count
is (2 + 1 + 1) = 4



Mutation
Count here is 1.
Since only
Second site is
Mutated.

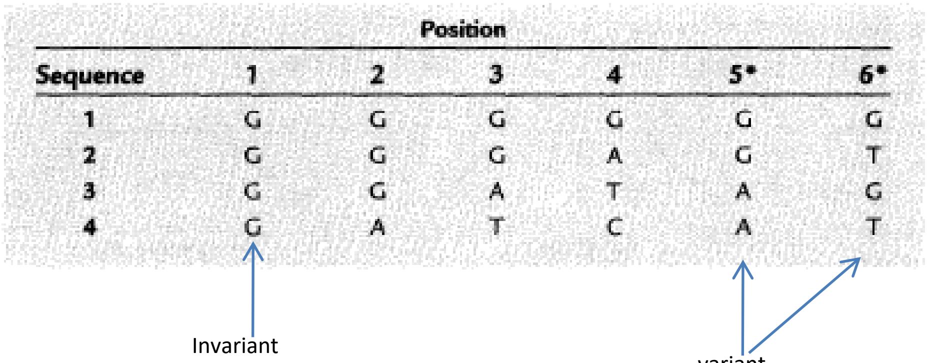
Mutation
Count here is 2.
Since two sites are
Mutated.

Mutation
Count here is 1.

Figure 5.20. Computing a parsimony score for a tree at three sites.

Informative sites

Definition. An informative site is one at which at least two different bases occur at least twice each among the sequences being considered.



	Position					
Sequence	1	2	3	4	5*	6*
1	G	G	G	G	G	G
2	G	G	G	A	G	T
3	G	G	A	T	A	G
4	G	A	T	C	A	T

Invariant
Uninformative

variant
informative

Although this is not hard to do by hand with only a few sites, as more sites are considered it quickly becomes too big a job. So computers are used.

We can save some effort in using the parsimony method if we make the observation that not all sites will affect the number of mutations needed for a tree.

So uninformative sites are not considered.

Problems

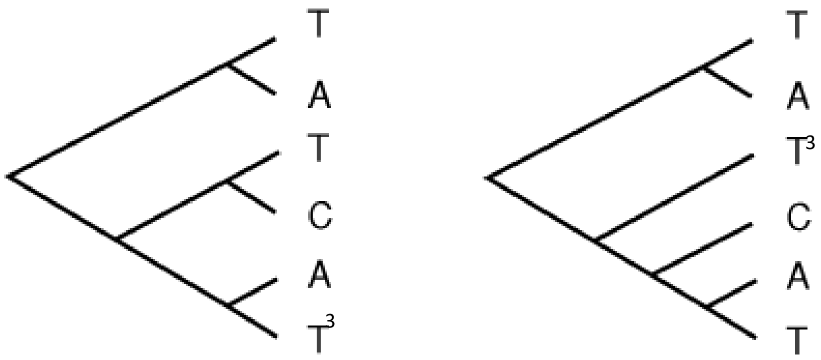
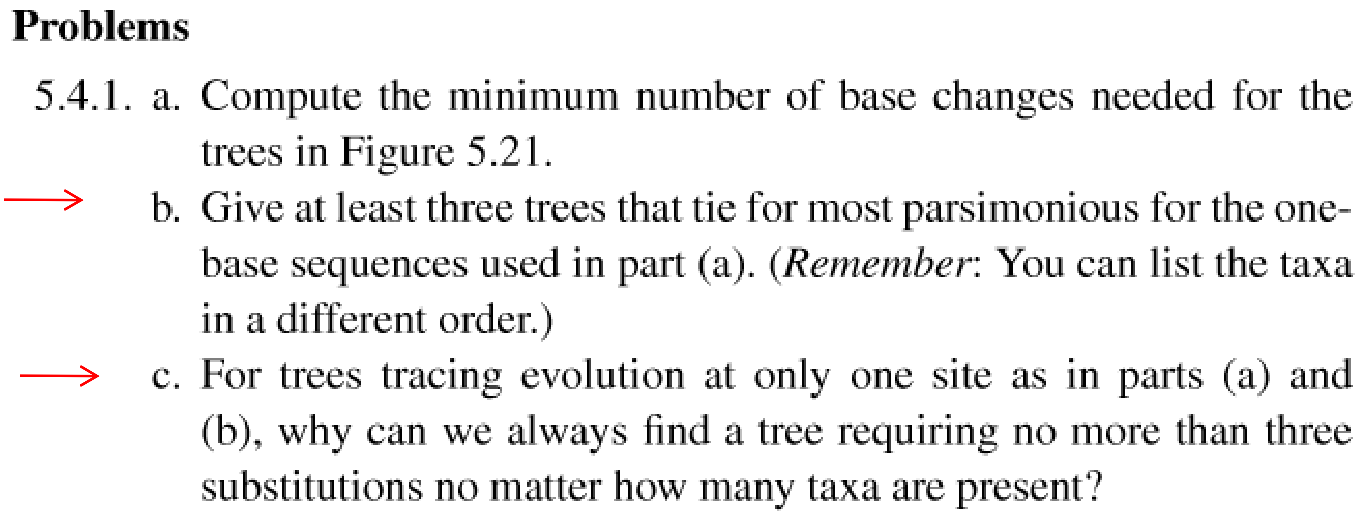


Figure 5.21. Trees for Problem 5.4.1.



Problems

5.4.1. a. Compute the minimum number of base changes needed for the trees in Figure 5.21.

HW → b. Give at least three trees that tie for most parsimonious for the one-base sequences used in part (a). (*Remember*: You can list the taxa in a different order.)

HW → c. For trees tracing evolution at only one site as in parts (a) and (b), why can we always find a tree requiring no more than three substitutions no matter how many taxa are present?

Problems

(5.4.2.7 & 5.4.2.8)

5.4.2. a. Find the parsimony score of the trees in Figure 5.22. (Only informative sites in the DNA sequences are shown.)

→ b. Draw the third possible (unrooted) topological tree relating these sequences and find its parsimony score. Which of the three trees is most parsimonious?

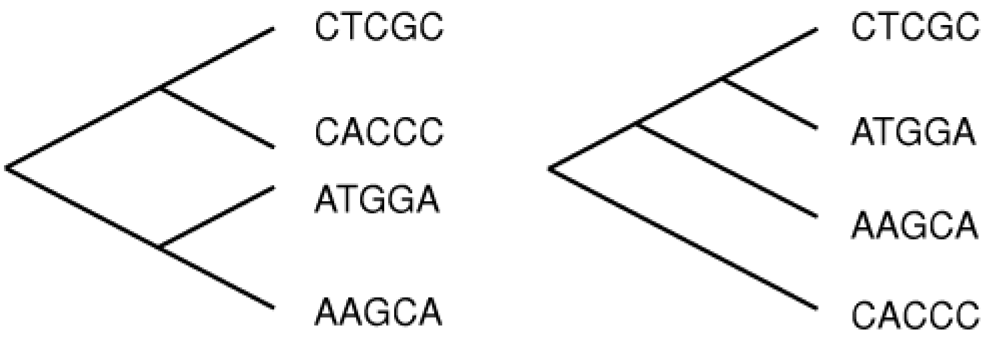
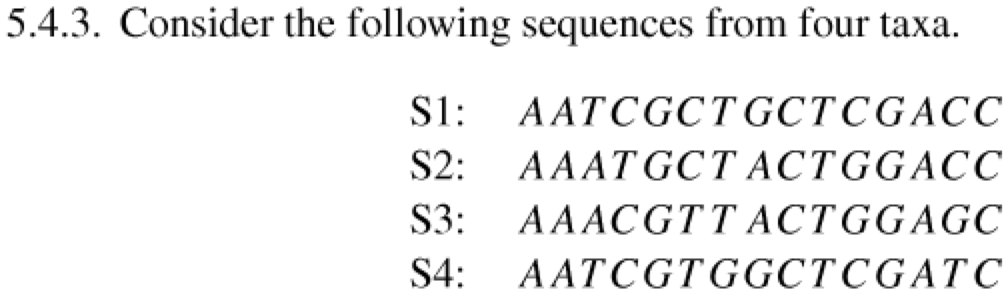


Figure 5.22. Trees for Problem 5.4.2.

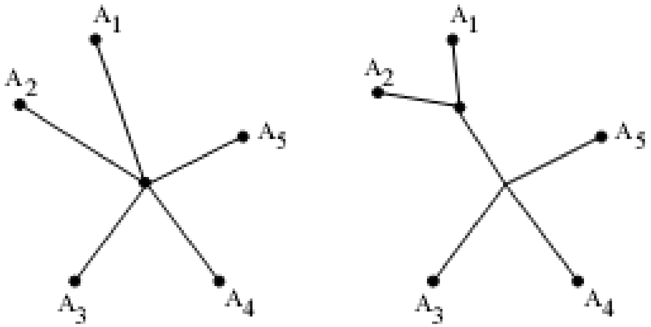
Problems



5.4.3. Consider the following sequences from four taxa.

S1: *AATCGCTGCTCGACC*
S2: *AAATGCTACTGGACC*
S3: *AAACGTTACTGGAGC*
S4: *AATCGTGGCTCGATC*

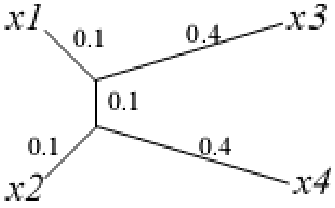
3 6 8 11



How to determine which nodes are neighbors?

It does *not* suffice simply to pick the i.e. a pair i, j with d_{ij} minimal.

Example: assume the true tree is



Given distances generated by this tree:

	x_2	x_3	x_4
x_1	0.3	0.5	0.6
x_2		0.6	0.5
x_3			0.9

x_1 and x_2 have minimal distance, but are not neighbors.

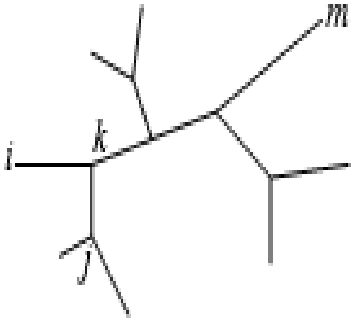
To avoid this problem, the trick is to subtract the "averaged distances" to all other leaves, thus compensating for long edges. We define a matrix $N = N_{ij}$ with

$$N_{ij} := d_{ij} - (r_i + r_j),$$

where

$$r_i = \frac{1}{|L| - 2} \sum_{k \in L} d_{ik},$$

Let i and j be two *neighboring leaves* that have the same parent node, k. Remove i, j from the list of nodes and add k to the current list of nodes. How do we have to set its distance to any given leaf m?



$$d_{im} = d_{ik} + d_{km},\ d_{jm} = d_{jk} + d_{km} \text{ and } d_{ij} = d_{ik} + d_{jk},$$

thus

$$d_{im} + d_{jm} = d_{ik} + d_{km} + d_{jk} + d_{km} = d_{ij} + 2d_{km},$$

which implies

$$d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij}).$$