1 Inferring trees from a data matrix

Consider the character matrix shown in table 1. We assume that the investigator has conducted primary homology analysis such that:

1. the characters (columns) contain codes for aspects of the organims that are thought to be comparable (we assume that the character homology statements are correct).
2. the character states are described with sufficient detail that we expect organisms with the same state to both being displaying the same evolutionary innovation (we assume that the character state homology statements are correct – satisfying Remane's "special similarity" and continuation criteria).

Table 1: Table 1 with color-coding of character types

	Character #									
Taxon	1	2	3	4	5	6	7	8	9	10
A	0	0	0	0	0	0	0	0	0	0
B	1	0	0	0	0	1	1	1	1	1
C	0	1	1	1	0	1	1	1	1	1
D	0	0	0	0	1	1	1	1	1	0

2 Clustering by distance

The most obvious way to infer a tree of taxa that describes this data is to cluster taxa based on similarity. We can produce a pairwise distance matrix for this set of taxa that reveals the proportion of characters for which any two taxa differ. This is shown in Table 4.

Table 2: The pairwise distance matrix for the characters shown in Table 1

	Taxon			
Taxon	A	B	C	D
A	-	0.6	0.8	0.5
B	0.6	-	0.4	0.3
C	0.8	0.4	-	0.5
D	0.5	0.3	0.5	-

2.1 Side-note about distance matrices

Note that the distance matrix is symmetric because the distance from taxon A to taxa B is the same as the distance from B to A. The distance matrix summarizes the amount of

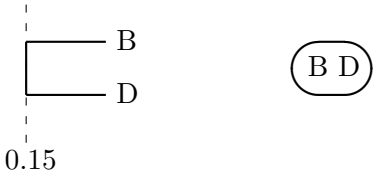
divergence in character states between two taxa, but is not a complete description of the information in the character matrix. It is possible to convert a character matrix into a distance matrix, but we can't invert this mapping. Another way to say this is to note that many different character matrices can map to the same distance matrix. The implication of this line of thought is that we may have more power to infer trees if we use the character matrix directly. This turns out to be the case, but for now we will pursue a simple method based on the matrix of pairwise distances.

2.2 UPGMA

Unweighted Pair Group Method with Arithmetic mean – a technique from numerical taxonomy (phenetics).

We expect close relatives to be similar to each other, so we could construct a tree by progressively grouping the pair of taxa that are closest – those 2 with the smallest distance. If we do this using the distance matrix in Table 4 we see that the smallest distance is 0.3 and is this is the distance between taxa B and D. This gives us the small tree shown in figure 1
Note that we set the node that connects B and D back in time a distance (0.15) that is half

Figure 1: round 1 of UPGMA from distance matrix in Table 4



the distance between B and D. This means yields a path from B to the MRCA of B and D then to D with a total length of 0.3 – which is the observed distance between the taxa.

We have a start on constructing a tree, but note that to take the next step we have to have distances between the other taxa (A and C) to the new group (B+D). In UPGMA, we create these distances by taking arithmetic averages. So

$$\begin{aligned} dist(A \leftrightarrow (B+D)) &= \frac{dist(A \leftrightarrow B) + dist(A \leftrightarrow D)}{2} \\ dist(C \leftrightarrow (B+D)) &= \frac{dist(C \leftrightarrow B) + dist(C \leftrightarrow D)}{2} \end{aligned}$$

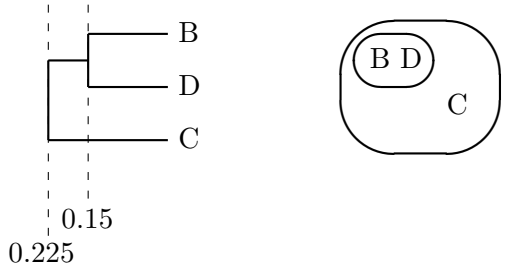
This allows us to construct a distance matrix for the second "round" of UPGMA. See table 3

At this stage the smallest distance is between C and the group (B+D). So this next group on the tree will lead to B, C, and D, and the internal node will have a depth of $\frac{.45}{2}$ (or .225). This gives us the small tree shown in figure 2
If we consider the distance matrix again, we find that we have only one distance:

Table 3: The second round pairwise distance matrix derived from perfect4dist after grouping B and D.

Taxon	Taxon		
	A	(B+D)	C
A	-	0.55	0.8
(B+D)	0.55	-	0.45
C	0.8	0.45	-

Figure 2: round 2 of UPGMA from distance matrix in Table 4



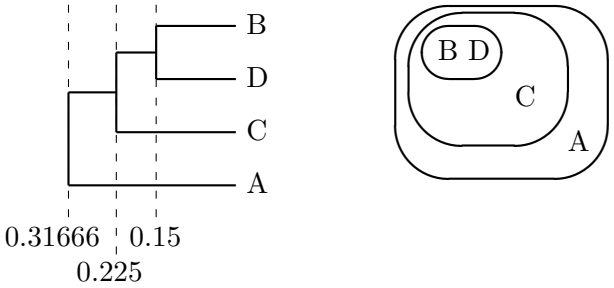
$$dist(A \leftrightarrow (B + C + D)) = \frac{dist(A \leftrightarrow B) + dist(A \leftrightarrow C) + dist(A \leftrightarrow D)}{3}$$
$$\approx 0.6333333$$

Thus we can place the root of the tree at $\frac{0.6333333}{2}$ (or 0.3166667), and attach taxon A to complete the tree, see figure 8.

problems with UPGMA

We have just constructed the upgma tree for the characters matrix shown in table 1 (via the distance matrix shown in table 4). UPGMA is a phenetic technique. Recall that the numerical taxonomists (or pheneticists) were more interested in producing clustering tools that summarize the similarity between groups. They tended to think the phylogenetic relationships were too difficult to estimate to serve as a basis for taxonomy. We can ask how well the tree (or "phenogram" in this case) represents the distances in the distance matrix.

Figure 3: the UPGMA tree from distance matrix in Table 4



The answer is that it does not do a perfect job. For instance, just looking at the tree you would expect that

$$dist(C \leftrightarrow B) = dist(C \leftrightarrow D),$$

and

$$dist(A \leftrightarrow B) = dist(A \leftrightarrow C) = dist(A \leftrightarrow D),$$

but neither of these sets of equations apply to the distance matrix in table 1. Thus has been some distortion of the distance matrix as we fit it onto a tree. This could be caused by the fact that we have a very small sample of characters – so random errors in the data matrix obscure the expected amount of character change between taxa. Another possibility is that UPGMA, while a straightforward way to construct a tree, is too simplistic and does not handle the complexities of real data. This second possibility has been demonstrated convincingly. In particular UPGMA is very sensitive to changes in the rate of character evolution – and these changes seem to be common in evolution. Thus UPGMA is not commonly used anymore.

3 Phylogenetic inference by Hennigian character analysis

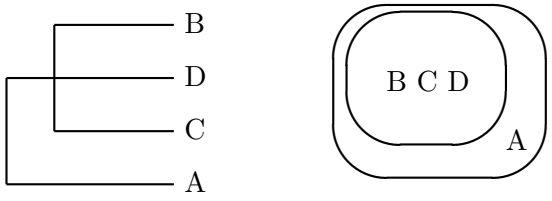
Note that in the previous section we did not use the terms "apomorphy" or "pleisiomorphy." That is because UPGMA does not come out of the phylogenetic systematics school that Hennig was crucial in developing. Recall that Hennig pointed out that synapomorphies alone are needed to recognize monophyletic groups – in particular symplesiomorphies are not helpful. But from table 1 alone we cannot determine which states are apomorphic and which are plesiomorphic.

3.1 Outgroup polarization

The most common way of determining the orientation of phylogenies (and, thus, characters) is the outgroup method. Usually we have a group of taxa that we are interested in, and we can identify organisms that are "more distant phylogenetically" than members of the group of interest. For example if we are interested in the African great ape phylogeny we might feel comfortable in assuming that human, chimp, and gorilla (the 3 African great apes) are more closely related to each other than any of them are to orangutan. Thus we can use orangutan as an **outgroup** in the analysis. In effect we treat as outgroup's character states as if they were the plesiomorphic states. We know that orangutan is not identical MRCA of orangutan and the African apes, but it turns out that this does not cause problems for phylogeny reconstruction (though it does make the interpretation of character evolution more complex – we'll discuss this later in the course). In the previous example the African apes are the **ingroup**.

In short, we assume (or use outside, independent evidence) to find an outgroup. When we do this, we **must** be confident that the *ingroup is monophyletic with respect to the outgroup.*

Figure 4: The assumption of A as the outgroup



Going back to the ape example, not long ago many scientists felt that orangutan might be the closest living relative to humans. If we still view this as an open question, then it would be inappropriate to use orangutan as an outgroup with human, gorilla, chimp as the ingroup – we would move to another taxon such as an old world monkey (e.g. a baboon). Using very distantly related outgroups can lead to problems because it may be difficult to make homology statement over long stretches of time.

Consider the data in table 1 again. If this data were being analyzed by Hennig, then he would identify an outgroup. For the sake of argument, we will say that we have identified taxon A as the outgroup. So we start the analysis "knowing" that the B, C, and D form a monophyletic group, thus the tree we start with is shown in figure 4; The polytomy (before we look at the data) is interperted as a soft polytomy (representing uncertainty about the phylogeny rather than a statement that B, C, and D were created by a single speciation event).

For convenience, I coded every character in the data matrix in table 1 such that the character stated displayed by taxon A is denoted by 0. Thus, using the outgroup (A) to polarize the characters, we assume that 0 is the plesiomorphic state and 1 is the apomorphic state. Recall that Hennig pointed out that we can only learn about the phylogeny through apomorphic states, specifically from shared, apomorphic states – synapomorphies.

4 Hennigian inference from polarized data

Recall that Hennig pointed out that synapomorphies alone are needed to recognize monophyletic groups – in particular symplesiomorphies are not helpful. Consulting table 1, we see that only one character (#10) tells us something new about the relationships between taxa.

The apomorphies in characters 1-5 identify taxa as having acquired a new state, but in all of these case the states are unshared – autapomorphies. These characters reinforce our belief that taxa A, B, C, and D are distinct (if that were an open question), but the only groups that the apomorphies support are single taxa "groups." These are trivial groupings – ones that will be found an every possible tree for these 4 taxa.

Figure 5: The tree preferred by Hennigian analysis of the data in table 1

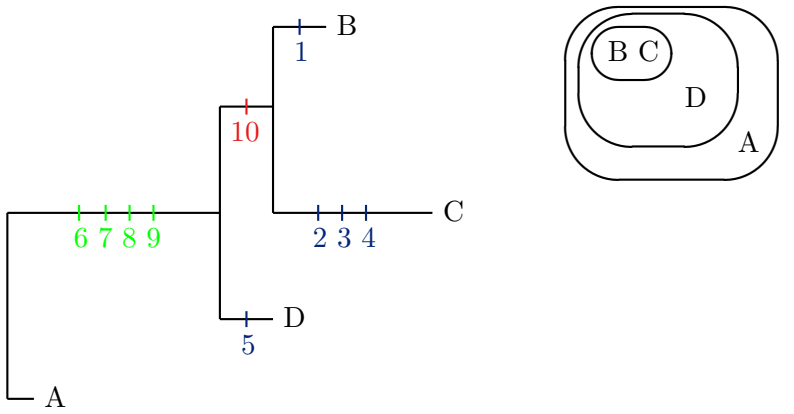
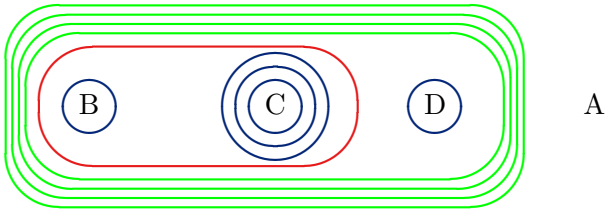


Figure 6: The tree preferred by Hennigian analysis of the data in table 1



The green characters (6-9) point to the existence of a monophyletic group B + C + D, but this was the entire ingroup. So these characters fall on the internal branch of the tree shown in figure 4. This is a branch that we already knew was in the tree.

Finally we come to character # 10. The apomorphic state for this characters points to grouping of B + C. This grouping is not present in the tree shown in figure 4, so we have learned something about the phylogeny from this character matrix (or we made a mistake making the homology statements during the construction of the matrix, and have inferred something incorrect about the phylogeny).

Figure 5 shows the tree that Hennigian inference would prefer. Tick-marks on branches are paired with numbers. These numbers indicate the number of the character(s) in the matrix that support that branch. Each character changes from the plesiomorphic state to the apomorphic state on the branch with the corresponding tick mark. Figure 6 shows a hierarchical characterization in which each character results in a line surrounding the taxa with apomorphic states. The lines are colored according to the character colors (but not labelled to avoid cluttering the figure).

4.1 Lack of homoplasy

Note that there does not appear to be any homoplasy when we map the data onto the tree shown in figure 5. Each character can be mapped on to the tree with a single transition from the plesiomorphic state to the apomorphic state. This is the example of a best case scenario. If we view the initial construction of the matrix as a primary hypothesis of homology, and the construction of the tree as the secondary test of homology, then we would say that the tree gives us no reason to question our primary homology statements.

This lack of homoplasy can be seen in the figure 6 by the fact that none of the lines have to cross each other. Each character corresponds to a grouping that is **compatible** with the grouping implied by every other character. There is a one-to-one correspondence between a (variable) character in the matrix and the branch on the tree that it "supports" – we say that a character supports a branch when the character changes state across the branch. This means implies that an evolutionary event that occurred in the lineage that the branch represents

4.2 Path lengths = character divergence

Also note that if we label the branches with the proportion of the characters in the matrix that change across each branch then we get the tree show in figure 7. A *path* on a tree is a set of branches that you have to follow when you move from one taxon to another. A path length can be defined as the sum of lengths of all of the branches along the path. Interestingly, if we use the tree in figure 7 to construct a path length matrix for all pairs of taxa, then we will obtain a matrix that is identical to the pairwise distance matrix (table 4) that we calculated directly from the character data. Recall from section 2.2 on the problems with the UPGMA tree, that there were some obvious problems with the UPGMA in terms of not being able to predict properties of the distance matrix. Even though the Hennigian-based tree did not use the distance matrix directly it was able to explain the distance matrix perfectly (while the distance-based approach fails).

We noted that UPGMA has problems as a phylogenetic inference method when the rates of evolution are not equal on all parts of the phylogeny. Note that this tree appears to be such a case. The branch leading to C is 3 times as long (3 times as many characters changed) as the terminal branch leading to taxon B or to D. In fact the path from B to D is short – these two taxa are very similar, but mainly because of sharing states which are symplesiomorphies. UPGMA does not attempt to discriminate between primitive and derived similarities, and the result is that it groups B and D because their raw distance was lower than the B to C distance.

A *path* on a tree is a set of branches that you have to follow when you move from one taxon to another. A path length can be defined as the sum of lengths of all of the branches along the path – in the context of a leaf-to-leaf path, this is referred to as a patristic distance. Interestingly, if we use the tree in figure 7 to construct a path length matrix for all pairs of

Figure 7: The tree figure 5 with branches expressed as the proportion of characters that change across the branch.

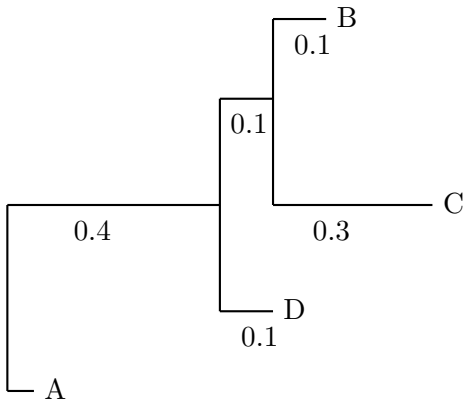
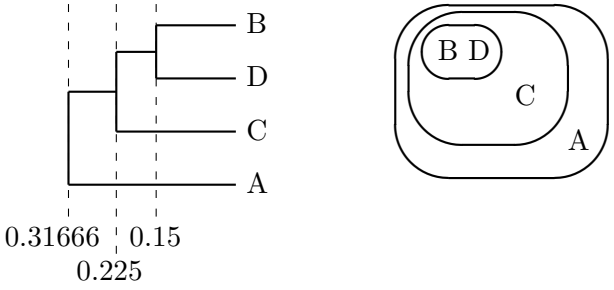


Figure 8: the UPGMA phenogram and hierarchy from distance matrix in Table 4



taxa, then we will obtain a matrix that is identical to the pairwise distance matrix (table 4) that we calculated directly from the character data.

Recall that UPGMA (Unweighted Pair Group Method with Arithmetic mean) inferred a tree by summarizing the character matrix into a "dissimilarity" (or "difference" or "distance") matrix shown in table 4. By repeatedly clustering the closest pair of taxa (or "operational taxonomic units" – OTU's), we produced the tree shown in figure 8.

Table 4: The pairwise distance matrix for the characters shown in Table 1

	Taxon			
Taxon	A	B	C	D
A	-	0.6	0.8	0.5
B	0.6	-	0.4	0.3
C	0.8	0.4	-	0.5
D	0.5	0.3	0.5	-

Even though the Hennigian-based tree did not use the distance matrix directly it was able to explain the distance matrix perfectly (while the distance-based approach fails). The patristic distances from the Hennigian tree in figure 7 perfectly return the character-based distances shown in Table 4. However if we use calculate a taxon-to-taxon distance matrix from the

patristic distances from the UPGMA phenogram shown in figure 8, we get the matrix shown in table 5

Table 5: Patristic distance matrix from the phenogram shown in figure 8

	Taxon			
Taxon	A	B	C	D
A	-	$0.633\overline{3}$	$0.633\overline{3}$	$0.633\overline{3}$
B	$0.633\overline{3}$	-	0.45	0.3
C	$0.633\overline{3}$	0.45	-	0.45
D	$0.633\overline{3}$	0.3	0.45	-

We noted that UPGMA has problems as a phylogenetic inference method when the rates of evolution are not equal on all parts of the phylogeny. Note that this tree appears to be such a case. The branch leading to C is 3 times as long (3 times as many characters changed) as the terminal branch leading to taxon B or to D. In fact the path from B to D is short – these two taxa are very similar, but mainly because of sharing states which are symplesiomorphies. UPGMA does not attempt to discriminate between primitive and derived similarities, and the result is that it groups B and D because their raw distance was lower than the B to C distance.

4.3 Accurate estimation from distances

Can we accurately predict the tree if we have good estimates of the evolutionary distance between taxa?

Yes. If our distances were directly proportional to the time to the MRCA between each pair of taxa, then UPGMA would always yield the correct phylogeny. But that is not too helpful, though because we don't observe these times – we can only observe characters data that gives us some clues about the history.

What if we knew the true number transitions that had occurred between species – would that be enough information to correctly infer the phylogeny? The distance matrix shown in table 4 represents an example of such knowledge – As demonstrated by the fact that we could recover it as the patristic matrix from the Hennigian tree in figure 7 (every change on the phylogeny in that figure contributes to an increase in the evolutionary distances between leaves of the tree).

A computer scientist named Peter Buneman (1971) showed that a dissimilarity matrix that accurately measures the evolutionary distance between taxa *is* sufficient knowledge to infer a phylogeny. Rather than just group the smallest distance in a cluster (as UPGMA did), Buneman's method works on quartets (groups of four taxa) and chooses the tree based on the sum of two distances:

Figure 9: Tree with edge lengths

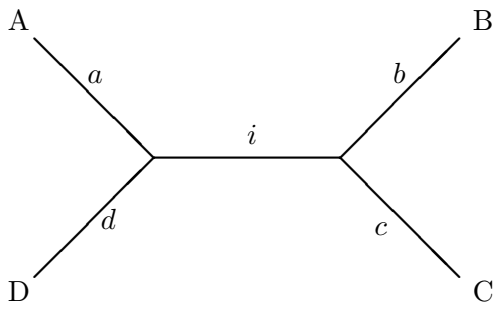


Table 6: Correspondence between tree and pairs of distances in a method based on Buneman's four point condition

Tree	distance sum	sum on tree in figure 14	from table 4
$(A+B)\|(C+D)$	$d_{AB}+d_{CD}$	$a+b+c+d+2i$	1.1
$(A+C)\|(B+D)$	$d_{AC}+d_{BD}$	$a+b+c+d+2i$	1.1
$(A+D)\|(B+C)$	$d_{AD}+d_{BC}$	$a+b+c+d$	0.9

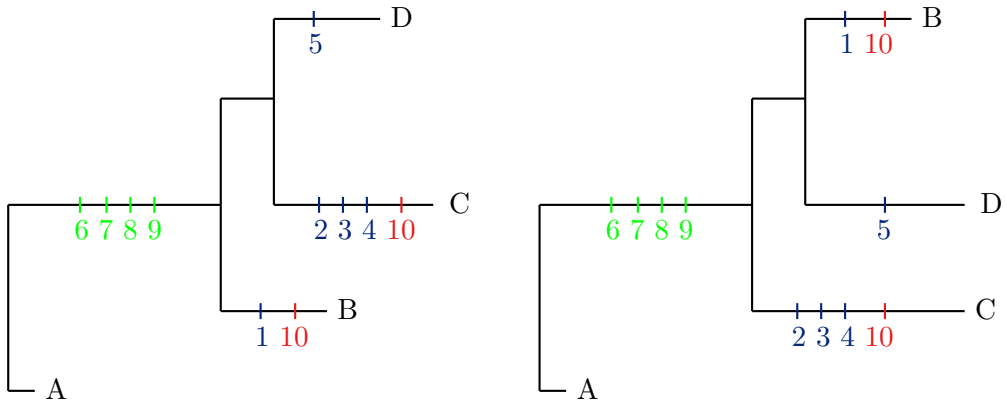
Note that when the sum associated with the true is the smallest (and the internal edge length can be estimated by half the distance between that sum and the other two).

4.4 Alternative trees require homoplasy

In this case of perfect, homoplasy-free data we were able to construct the tree by adding groupings every time we found a character that supported the grouping (in this case there was only one helpful character, but that is because it was a tree of only 4 species). We might ask whether or not we can recognize incorrect trees. The other two bifurcating trees for these taxa are shown in figure 10. Note that in both of these trees there is an internal branch with no inferred character changes – these branches have no support in this character matrix. The other point to note is that in each of these trees, character #10 (the only character with a synapormorphy that gave us information about the relationships within B+C+D) can only be explained as a homoplastic character – with two independent acquisitions of the 1 character state. The conclusion is that the analyses suggested by Hennig can not only allow us to construct the tree, but given a tree we can:

- detect unsupported groupings (branches with no inferred changes), and
- detect disagreement between the tree and character matrix (mapping characters on rejected trees requires homoplasy)

Figure 10: The bifurcating trees rejected by Hennigian analysis of the data in table 1



Tree inference from data with conflict

Consider the character matrix shown in table 7. Taxon A is the outgroup, and the table uses colors to indicate the information content of the characters. Blue for characters with only autapomorphies, green characters for with a synapomorphy that support the assumed separation between the ingroup and outgroup, and red for the character with a synapomorphy that can inform the other parts of the tree.

Table 7: Data with color-coding of character types

	Character #											
Taxon	1	2	3	4	5	6	7	8	9	10	11	12
A	0	0	0	0	0	0	0	0	0	0	0	0
B	1	0	0	0	0	1	1	1	1	1	1	1
C	0	1	1	1	0	1	1	1	1	1	1	0
D	0	0	0	0	1	1	1	1	1	0	0	1

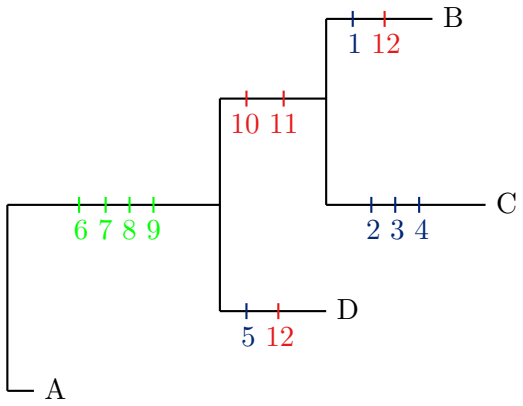
Once again we assume that A is the outgroup (see figure 4).

5 Phylogenetic inference

Figure 11 shows a tree for this dataset with characters mapped onto it. Tick-marks on branches are paired with numbers. These numbers indicate the number of the character(s) in the matrix that support that branch. Each character changes from the plesiomorphic state to the apomorphic state on the branch with the corresponding tick mark.

Note that in table 7, characters #10 and #11 support the tree that puts taxa B and C together. However on this tree character #12 displays homoplasy. Character #12 favors a

Figure 11: A tree for the data in table 7



different tree – the tree with B and D together (shown in figure 12). On *that* tree, character #12 would be inferred to have evolved along the branch that leads to the clade with B+D.

Figure 13 shows a hierarchical characterization in which each character results in a line surrounding the taxa with apomorphic states. The lines are colored according to the character colors (but not labelled to avoid cluttering the figure). Unlike previous examples (in the previous lectures) the lines demarcating groups in this example must cross each other. Specifically the oval that corresponds to character 12 (one of the red ovals), has to intersect with the lines that represent characters 10 and 11. This crossing of lines is a sign of character conflict.

In Hennigian analyses

1. characters 10 and 11 both "say" that taxa B and C are more closely related to each other than either are to taxon C.
2. character 12 says that B and D are sister taxa (because the share the apomorphic state, the should belong to a monophyletic group that excludes C and A)

But these two statements are in direct conflict – they cannot both be true. One solution is to go back to the organisms and reexamine your assumptions about character and character state homology. Recoding the data could result in a new, homoplasy free character matrix.

This process of going back and forth between character coding and phylogenetic character analysis is the testing of primary homology statements – it is often called "reciprocal illumination."

Figure 12: An alternative tree for the data in table 7

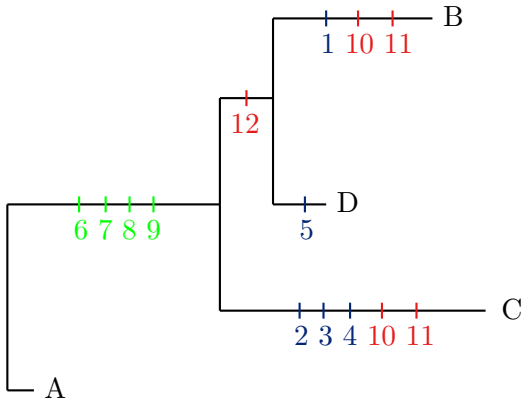


Figure 13: nesting relationships for data in table 7

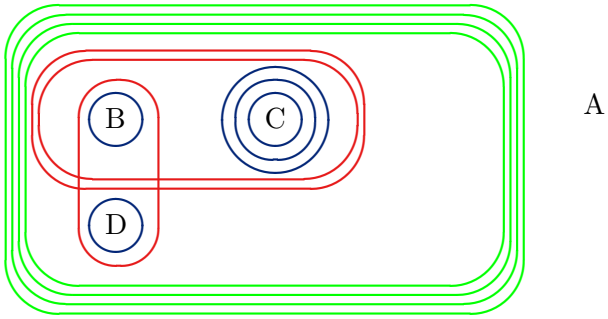


Table 8: The pairwise distance matrix (in number of characters that differ) for the characters shown in Table 7

	Taxon			
Taxon	A	B	C	D
A	-	8	9	6
B	8	-	5	4
C	9	5	-	7
D	6	4	7	-

Note (figure 14 and Table 9) that when there are homoplastic characters in the matrix, Buneman's four point condition will not necessarily hold.

Buneman's four point condition

Four any four taxa (say A, B, C, and D) there are 6 taxon-to-taxon distances that can be estimated from character data:

$$d_{AB}, d_{AC}, d_{AD}, d_{BC}, d_{BD}, \text{ and } d_{CD}$$

There are three ways that you can construct sums of 2 of these distances while making sure that each of the four taxa appear in the formulae:

$$d_{AB} + d_{CD} \quad (1)$$
$$d_{AC} + d_{BC} \quad (2)$$
$$d_{AD} + d_{BD} \quad (3)$$

Buneman's four point condition states that at least two of the sums will be identical to each other, and the third sum will be smaller. The smallest sum corresponds to the sum of distances that pair up sister groups.

Table 9 demonstrates why Buneman's condition should hold (using branch lengths that are shown in Figure 14), but we can see tha for the data shown in Table 8 the condition does *not* hold. None of the sums shown in the fourth column of the table are identical.

Buneman's condition does not hold in this case becaus the distances in table 8 are not accurate estimates of the evolutionary distance between taxa. This occurs because homoplasy masks evolutionary events. Refer to the mapping of characters in Figure 11. Note that there are six events that occur on the path from B to D (character state changes in characters 1, 5, 10, and 11 and two changes in character 12), but the homoplasy in character 12 results in B and D sharing the same state. Thus, when we calculate the pairwise distance matrix (figure 8) we miss (fail to count) both of these events and estimate a distance of 4. We underestimate the true evolutionary distance by 2 events. So homoplasy confounds distance-based approaches as well as Hennigian analysis.

Tree	distance sum	sum on Fig 14 tree	sum on Fig 15 tree	sum on Fig 16 tree	from table 8
$AB\|CD$	$d_{AB} + d_{CD}$	$a+b+c+d+2i$	$e+f+g+h$	$l+m+n+o+2k$	$8+7=\mathbf{15}$
$AC\|BD$	$d_{AC} + d_{BD}$	$a+b+c+d+2i$	$e+f+g+h+2j$	$l+m+n+o$	$9+4=\mathbf{13}$
$AD\|BC$	$d_{AD} + d_{BC}$	$a+b+c+d$	$e+f+g+h+2j$	$l+m+n+o+2k$	$6+5=\mathbf{11}$

Table 9: Failure of Buneman four point condition on data set with homoplasy

Figure 14: $AD|BC$ Tree with edge lengths (introducing the notation for Buneman's fourpoint condition

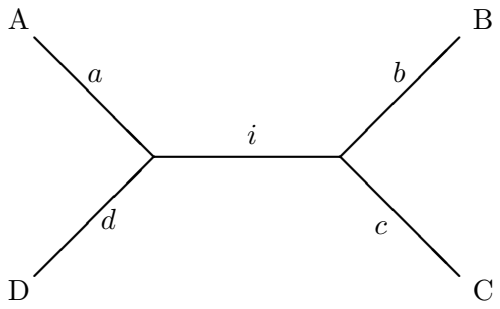


Figure 15: $AB|CD$ Tree with edge lengths (introducing the notation for Buneman's fourpoint condition

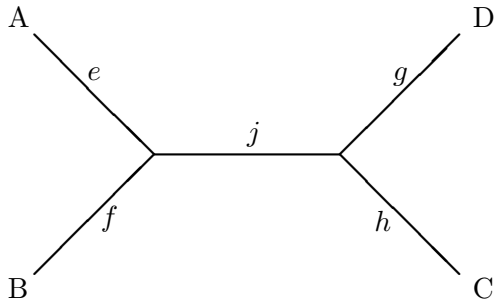
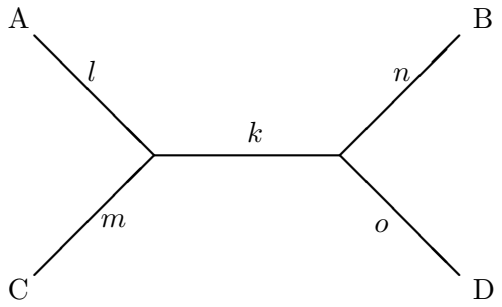


Figure 16: $AC|BD$ Tree with edge lengths (introducing the notation for Buneman's fourpoint condition



6 Inferring trees from data with character conflict

What if we don't want to return to the character coding and go back to the specimens? We may feel that we have fully examined the characters in an objective way at the beginning stage of the analysis. Returning to the character coding decisions with the intent of eliminating character conflict in our matrix could lead to arbitrary decisions of character rejection. Even if our secondary character-recoding or character-rejection decisions aren't "arbitrary" they could be very subjective and hard to replicate by other workers. Another danger is that the "second iteration" of the data matrix would look cleaner than the real data was – this could lead other scientists to put too much faith in our results.

The predominant phylogenetic techniques today try to avoid recoding the matrix. If the "reciprocal illumination" approach to recoding characters is employed, then it is done at an early stage based on experience from related groups (this leads to knowledge that a certain character is not "good" for a particular group).

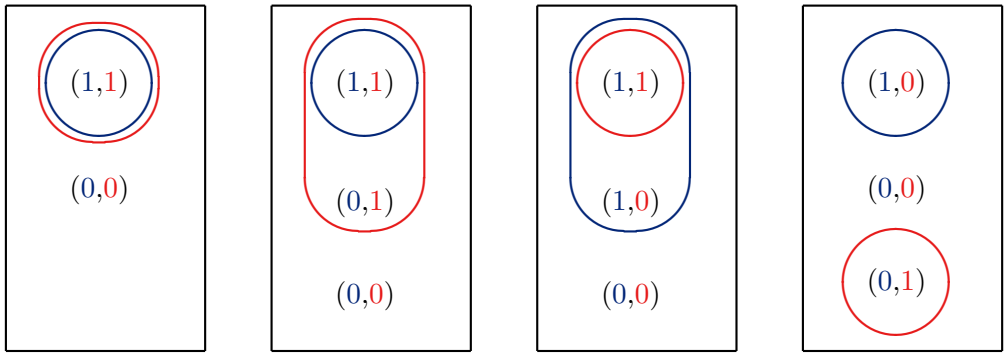
7 Compatibility

One method to infer a tree from a data set that shows incongruent signals of evolutionary relationships is to remove characters from the data matrix that seem to conflict with other characters. To make this practice objective, we could try to remove the *fewest* characters from the matrix required to yield a conflict-free matrix. This is referred to as a maximal compatibility – the determination of the largest number of characters that are all compatible with each other.

Two characters are compatible with each other if there is a tree on which both characters can be mapped without any homoplasy.

Fortunately, there is a easy method to determine if two characters are incompatible with each

Figure 17: The 4 possible nesting relationships two compatible characters



other. If we consider the possible combination that a taxon can display for two characters there are only four possibilities:

(0,0) – a character state of 0 for both characters.

(0,1) – a character state of 0 for the first character and state 1 for the second character.

(1,0) – a character state of 1 for the first character and state 0 for the second character.

(1,1) – a character state of 1 for both characters

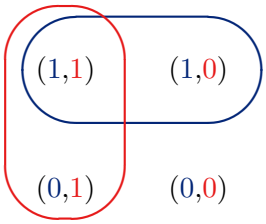
If you consider the states for a pair of characters over all of the taxa, and you see each of these four of these combinations of characters states, then the two characters are **incompatible**.

Figure 17 depicts the combinations of states that could be observed when you look at pairs of character states among compatible characters. In this figure, this first character is shown in blue – taxa with state 1 in this character can be put inside the blue oval. The second character is shown in red. Taxa with state 1 for the second character would fall inside the red ovals. As in other nesting diagrams that we have seen, the ability to draw the diagrams without making lines cross each other indicates the absence of conflict. Note that, this figure only depicts three of the four possible combinations of characters.

Figure 18 shows what happens when all four combinations (0,0), (0,1), (1,0), and (1,1) are seen when two characters are paired up. The red and blue lines must cross – indicating incompatibility.

You can think of compatibility as an attempt to explain the data, by saying that any character conflict (incompatibility between characters) reflects an error in character coding (the decision of how to represent homologous aspects of organisms as a character in a matrix). If we assume that such errors are rare, and that when they occur we should be suspicious of the entire character (column of the matrix), then a logical approach is to figure out how to reject as few of our primary statements of homology as possible to eliminate the conflict. This is exactly what the compatability method does.

Figure 18: Incompatible characters revealed by the presence of all four combinations of character state pairs.



After we have culled the matrix to remove incompatible characters, then we can use Hennigian character analysis to infer the tree.

8 Parsimony

While the compatibility approach is logical, it seems radical to throw out an entire character (column of the matrix) when we detect conflict. It may well be the case that only a few of our character state assignments were in error. As shown in figures 11 and 12, we can map any character matrix onto a tree by allowing some characters to be reconstructed with homoplasy. When we map a character onto a tree and we find that we must posit homoplasy, this means that some of the character states appear to have evolved more than once. This indicates that our homology statements in terms of character state assignments were not perfect.

In the compatibility method in which we throw the entire character out when we need to detect that at least one of our homology statements must be wrong. When we use **parsimony** criterion for phylogenetic inference, we are not this extreme. Instead we express a preference for trees that require the minimum amount of homoplasy.

It turns out that this is equivalent to preferring trees with the fewest number of inferred character state transitions. When we have changes in characters mapped to a tree, then we can simply count the number of tick marks (each mark represents a change of state $0 \leftrightarrow 1$). This number of changes or *steps* is the parsimony score of the tree. We can score each character on a tree and arrive at a parsimony score for each character if we choose. The score for the whole tree is simply the sum of the score for each character.

Table 10 shows the parsimony score for each character in Table 7. Note that only the last three characters have scores that vary between the trees. These characters have scores shown in red. They are the ones that display potential synapomorphies within the ingroup, so they are the potentially informative characters in a Henngian character analysis. They

are referred to as *parsimony-informative* because when we are using parsimony to evaluate trees, it is only characters like these that carry information for inferring the relationships between taxa.

Note that parsimony would prefer the tree that groups B+C together, because it has the smallest score (13) – the minimum number of character state changes required to explain the data. This is not too surprising: the B+C character had only one character (#12) for which we need to invoke homoplasy to explain the characters. In contrast, the B+D tree only requires one change for character #12, but requires two changes for characters #10 and #11. The C+D tree is the worst explanation, according to parsimony – no character shows a synapomorhy uniting C and D, and three characters (#10,#11, and #12) require multiple events to explain the character state distributions.

Table 10: Table 7 with tree scores

	Character #												
Taxon	1	2	3	4	5	6	7	8	9	10	11	12	
A	0	0	0	0	0	0	0	0	0	0	0	0	
B	1	0	0	0	0	1	1	1	1	1	1	1	
C	0	1	1	1	0	1	1	1	1	1	1	0	
D	0	0	0	0	1	1	1	1	1	0	0	1	
Tree	Character Score												Total Score
B+C	1	1	1	1	1	1	1	1	1	1	1	2	13
B+D	1	1	1	1	1	1	1	1	1	2	2	1	14
C+D	1	1	1	1	1	1	1	1	1	2	2	2	15

8.1 Fitch optimization

In the last section, I told you the minimal # of changes required for each character – but how did I know this number?

A brute force method would be to assign character states to all of the internal nodes of the tree. If we have a tree with internal states assignments, then the number of changes required is easy to calculate

A bit about sets

Walter Fitch's (1970 and 1971) algorithms for calculating the parsimony score of a tree use a bit of set notation. A **set** is a collection of objects. The objects are usually referred to as **elements** of the set. The common notation for a set is a pair of curly braces - {}.

The operations on sets used in Fitch's algorithm are the intersection and the union.

The **union** of two sets is a set that contains all of the elements that are in either sets. The union is denote $\cup$. For example:

$$\{2, 5, 6, 9\} \cup \{1, 5, 7, 9, 10\} = \{1, 2, 5, 6, 7, 9\}$$

The **intersection** of two sets is a set that contains only elements that are in *both* sets. The $\cap$ is used to represent an intersection. For example

$$\{2, 5, 6, 9\} \cap \{1, 5, 7, 9, 10\} = \{5, 9\}$$

An empty set, denote with a $\emptyset$, is a set that contains no elements. For example:

$$\{2, 6, 9\} \cap \{1, 5, 7, 10\} = \{\} = \emptyset$$

8.2 Fitch downpass

Walter Fitch showed that you can calculate the best possible parsimony score for a character on a tree (the fewest number of changes-of-state that explain the data), using a single pass down the tree from the leaves to the root.

1. Start with score=0
2. Use the character states observed for each leaf in the tree to create a set of character states for each leaf on the tree.
3. Repeat the following steps as you move from the leaves to the root of tree (if you move down the tree in such a way that you only encounter an ancestral node after you have determined a state-set for the node's descendants, then you can complete the algorithm in a single "traversal" from leaves to the root):
   (a) Let s_A denote the state set of the ancestral node. Since we did not observe this species, s_A is unknown.
   (b) Let s_L denote the state set of the left descendant, and let s_R denote the state set of the right descendant. s_R and s_L will be known because the taxa are observed (if the descendant nodes are leaves), or the state set was inferred from a previous iteration of the algorithm.

   If $s_L \cap s_R = \emptyset$
   then $s_A = s_L \cup s_R$ and add 1 to the current score
   otherwise $s_A = s_L \cap s_R$

You can think of the logic of the algorithm as:

1. "if the two descendants show the same states, then assume that the ancestor had the states that are in common" – this is the part in which you assign $s_A = s_L \cap s_R$ if $s_L \cap s_R \neq \emptyset$.

2. "if the two descendants have different states, then assume that the ancestor could have had *any* state that is in at least one of the children. If this is the case, then we know that we will need to have one change-of-state in this part of the tree, so we add one to the score" – this is the part in which we augment the score, and assign $s_A = s_L \cup s_R$.

Important note: this downpass of the Fitch algorithm only gives us the parsimony score of the character on the tree. It does not (necessarily) give us the most-parsimonious ancestral character state reconstructions for each node. Technically the ancestral state sets are referred to as the *preliminary* state sets for the nodes. We will not cover the algorithm to determine which are the most parsimonious evolutionary scenarios (I am happy to discuss it with you, if are interested).

To evaluate trees using parsimony we do not need to know all of the ancestral character state assignments, we just need the score. So the "downpass" of Fitch's algorithm gives us what we need to score a tree using parsimony.

Figures 19-27 give an example of how the algorithm works.

Table 11: A data matrix consisting of one character in a DNA sequence

Taxon	Character state
A	A
B	C
C	C
D	A
E	A
F	C
G	G
H	A
I	A

Figure 19: Fitch algorithm steps 1 and 2 for that DNA sequence character shown in Table 11

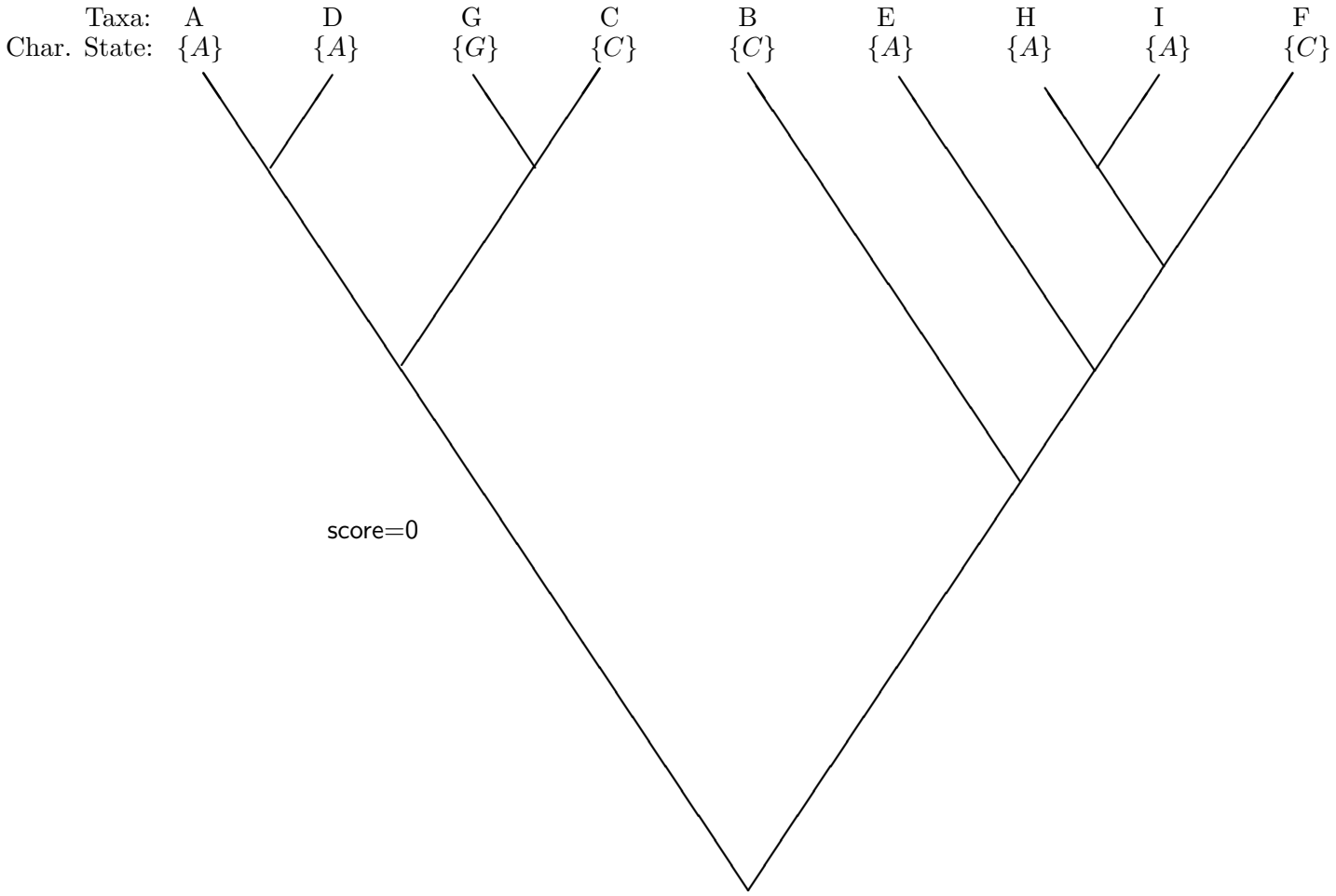


Figure 20: Fitch algorithm step 3 (first ancestral node) for that DNA sequence character shown in Table 11

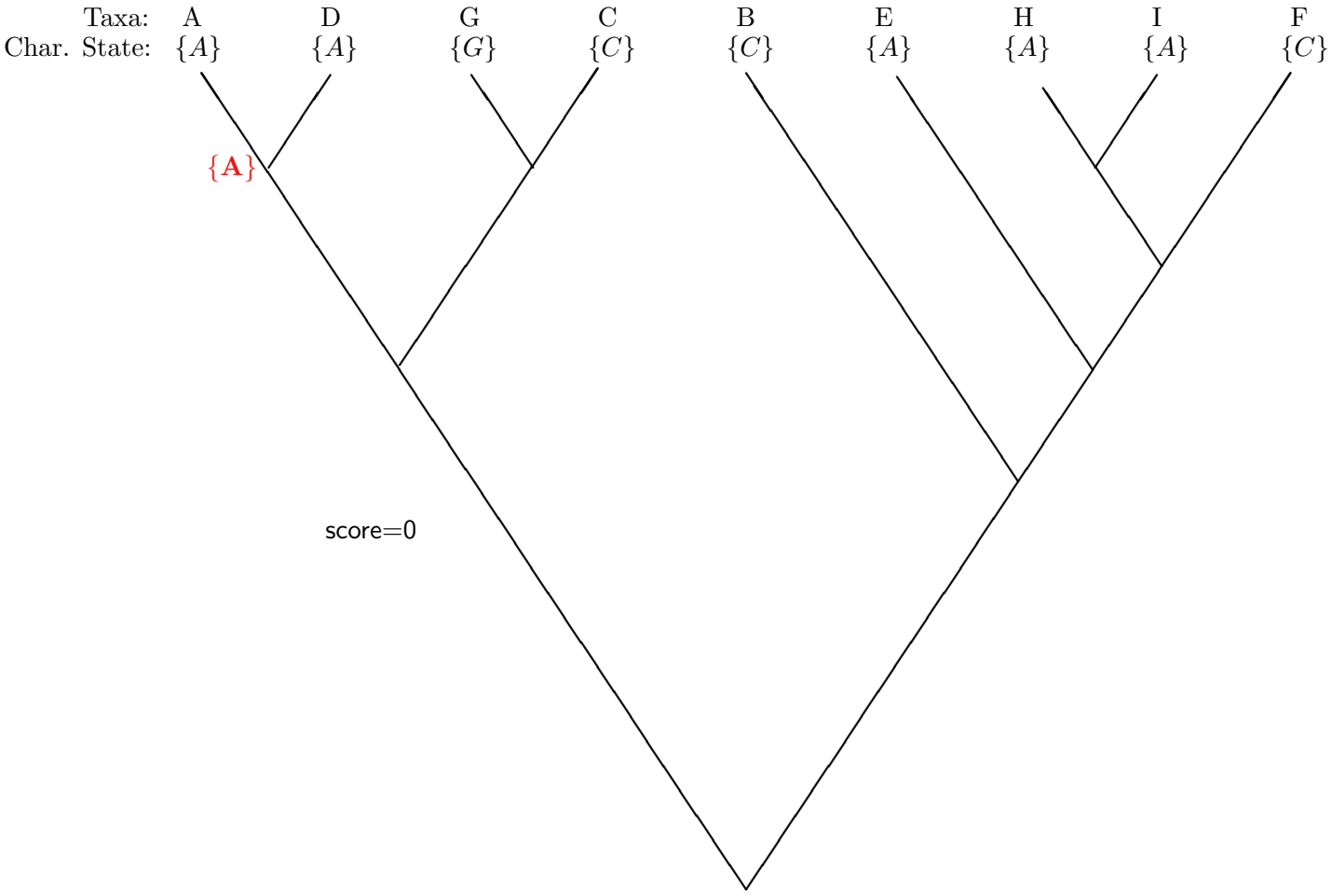


Figure 21: Fitch algorithm step 3 (second ancestral node) for that DNA sequence character shown in Table 11

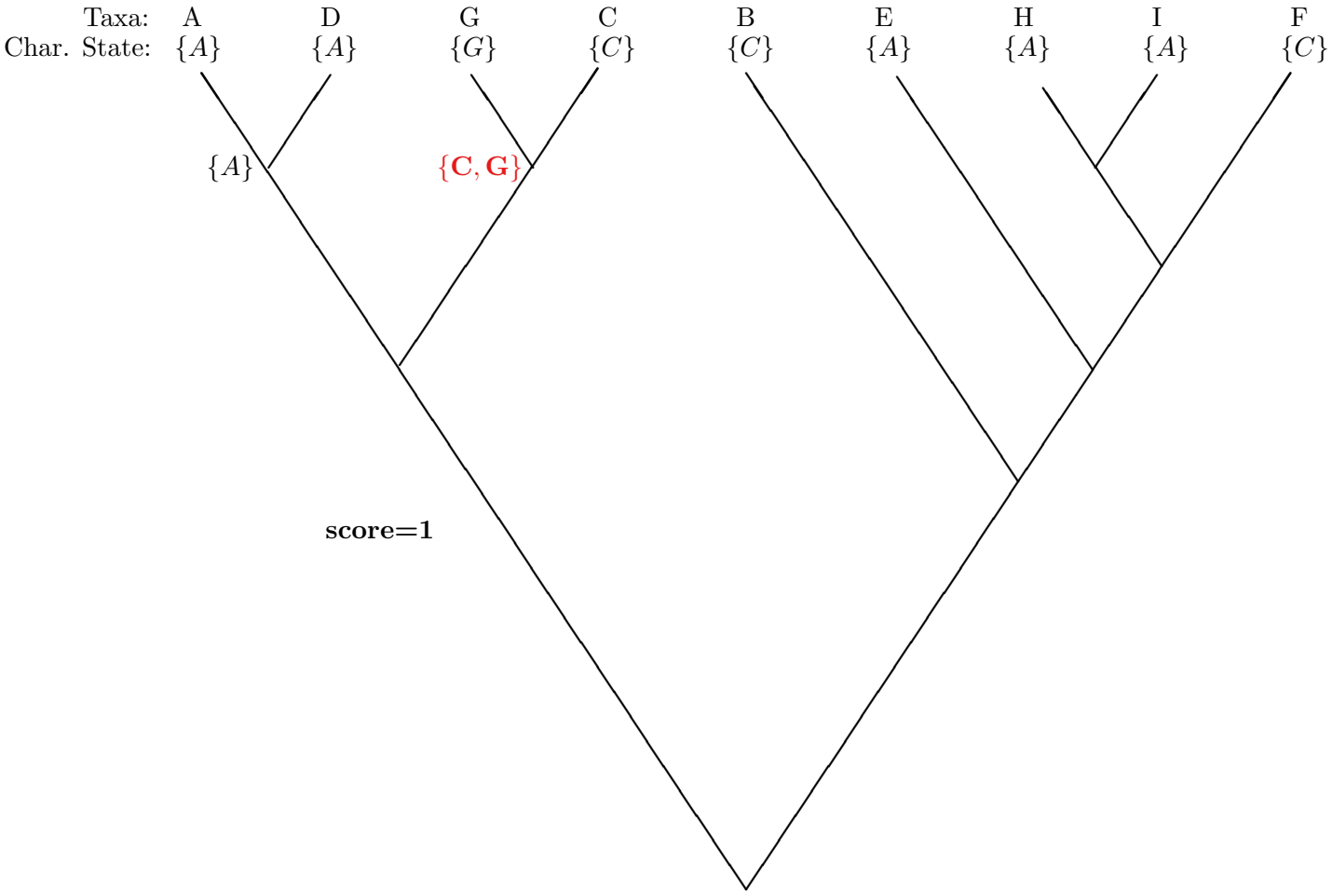


Figure 22: Fitch algorithm step 3 (third ancestral node) for that DNA sequence character shown in Table 11

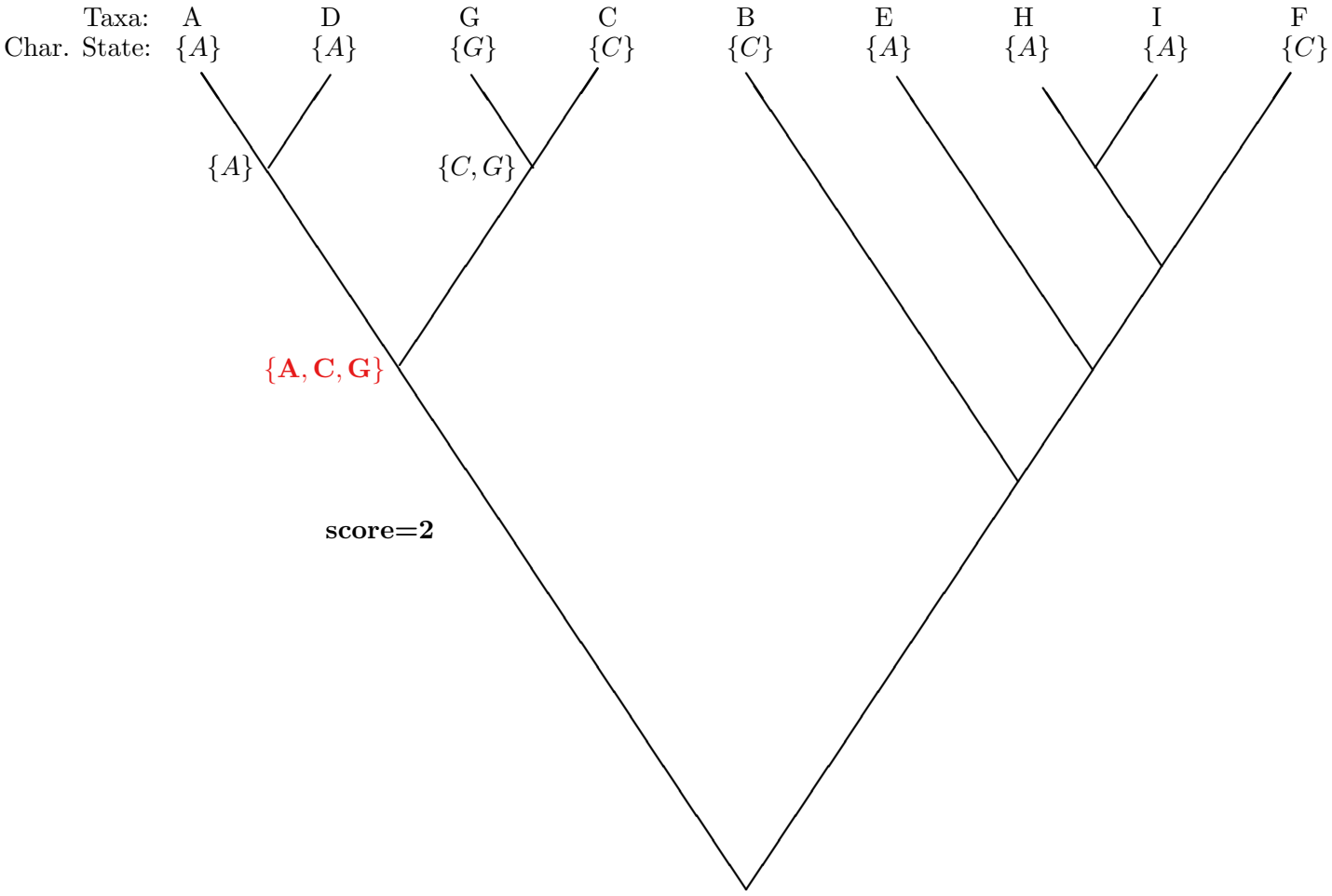


Figure 23: Fitch algorithm step 3 (fourth ancestral node) for that DNA sequence character shown in Table 11

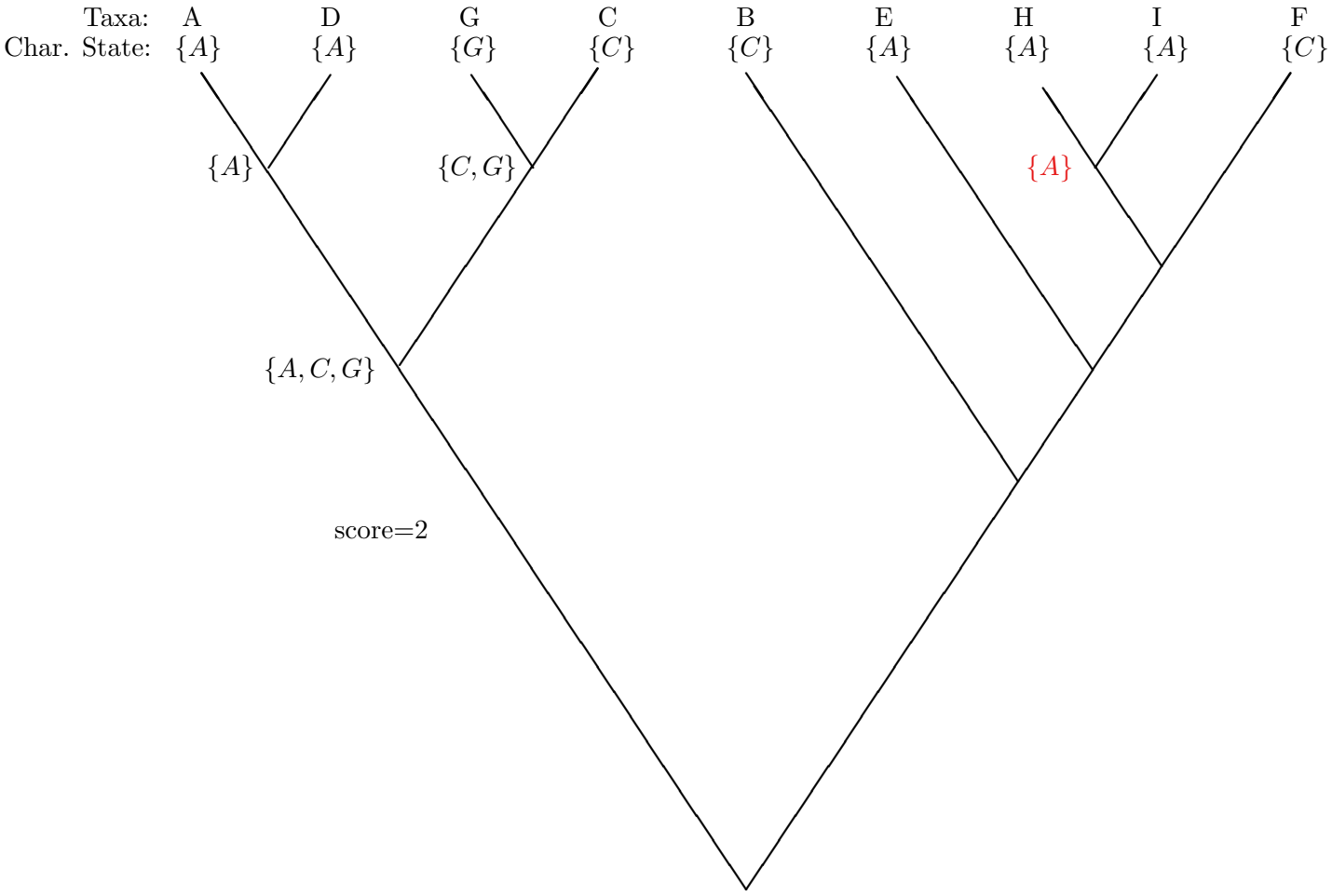


Figure 24: Fitch algorithm step 3 (fifth ancestral node) for that DNA sequence character shown in Table 11

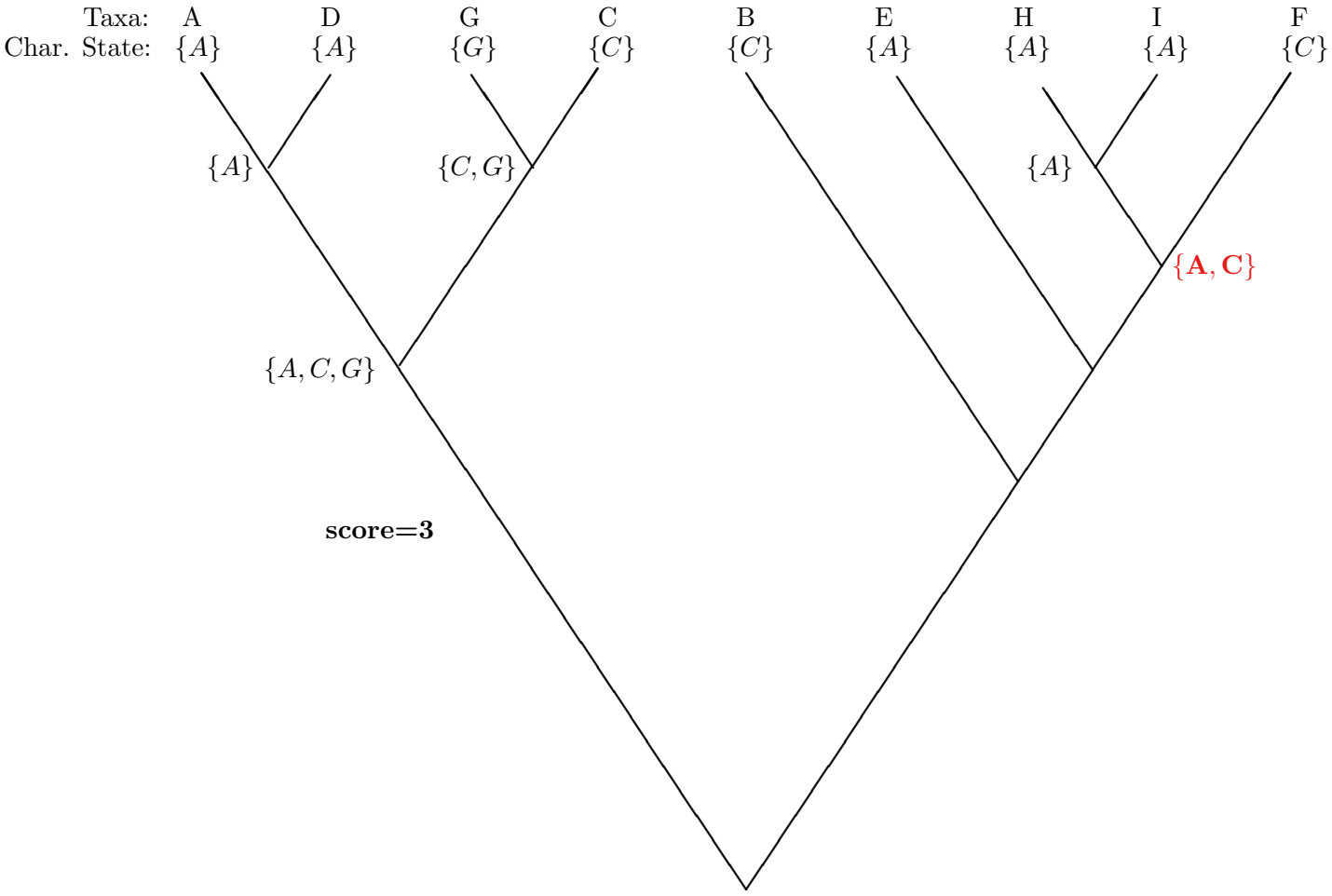


Figure 25: Fitch algorithm step 3 (fifth ancestral node) for that DNA sequence character shown in Table 11

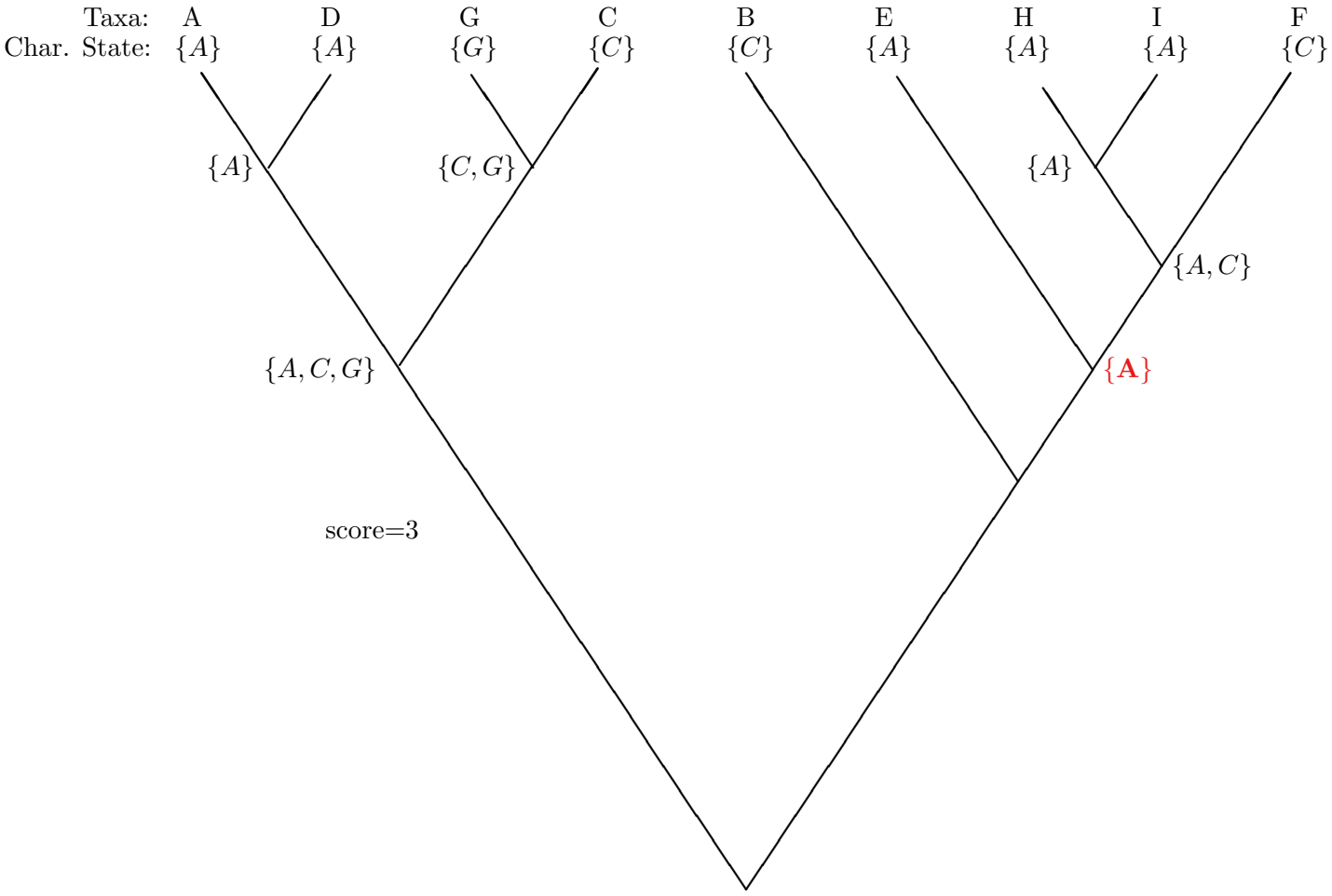


Figure 26: Fitch algorithm step 3 (sixth ancestral node) for that DNA sequence character shown in Table 11

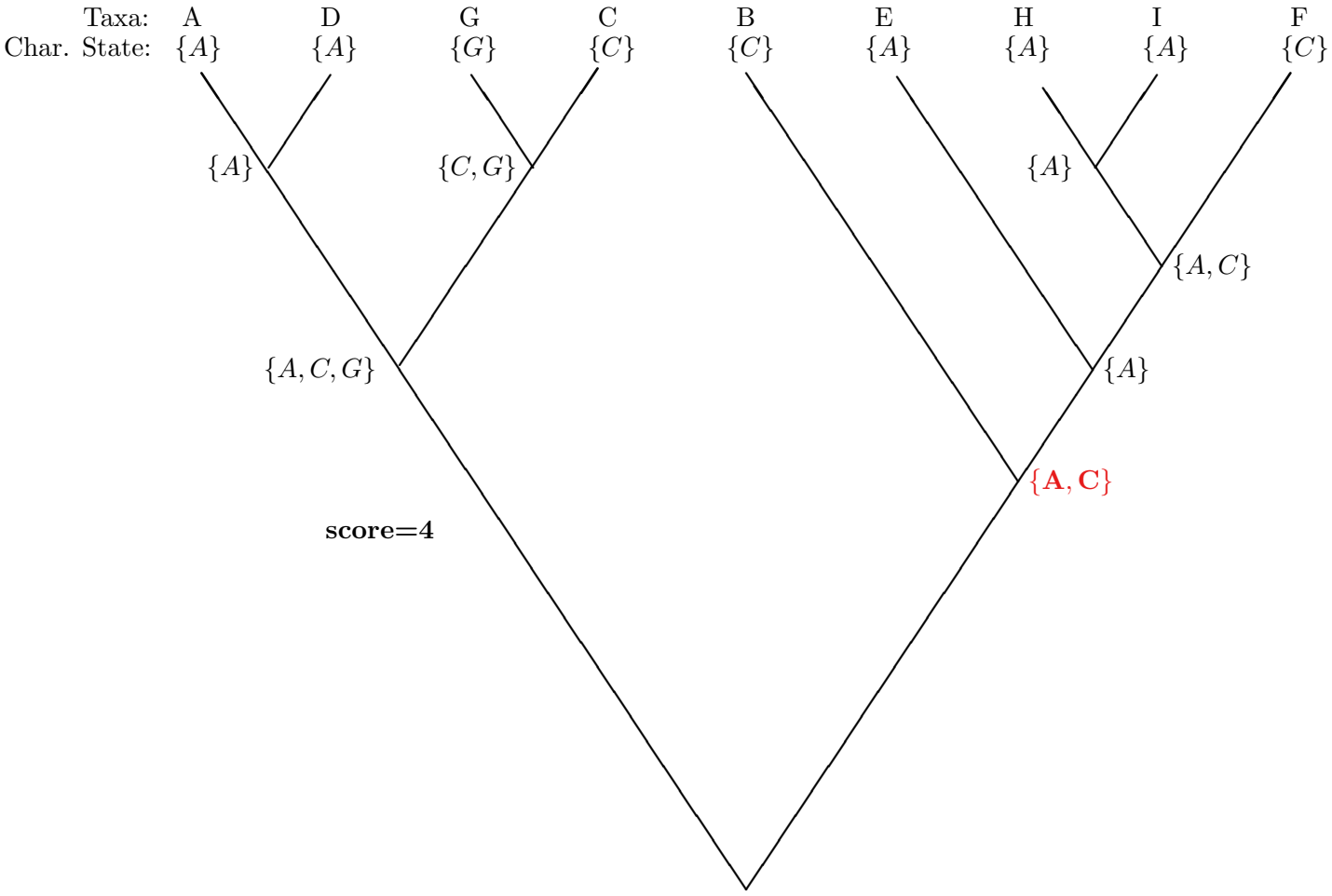


Figure 27: Fitch algorithm step 3 (seventh ancestral node – root) for that DNA sequence character shown in Table 11