

# Chapter 8

## Worked examples

This chapter provides a series of worked examples illustrating the various tree comparison, consensus, randomisation, and mapping features available in COMPONENT. Where possible the examples have been taken from the literature. By working through this chapter you should gain a feel for using COMPONENT.

All the example files are on your distribution diskette and will have been copied onto your hard disk when you installed COMPONENT.

### Consensus trees

---

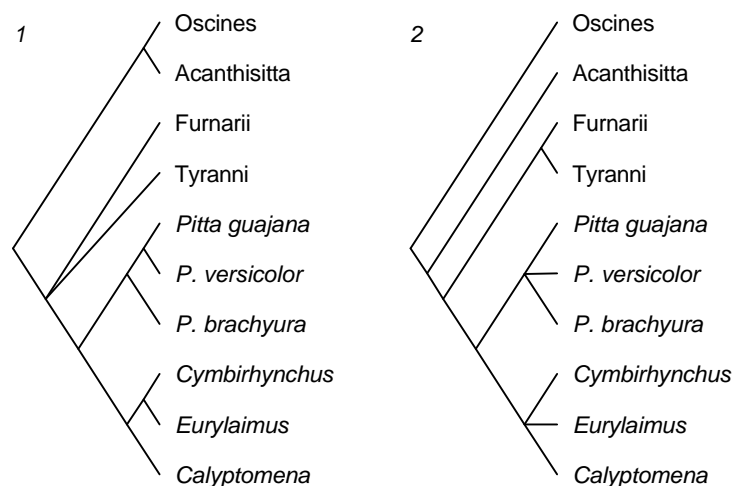
Consensus trees are a well known means for summarising information shared by a set of trees. The following examples illustrate some of the methods available in COMPONENT.

#### Semi-strict consensus trees

---

The semi-strict consensus method (Bremer, 1990) combines all the uncontradicted clusters from two or more trees. The file PASS.NEX contains two trees for passerine birds, one based on morphological data, the other based on single copy DNA hybridisation data (see Bledsoe and Raikow, 1990).

Figure 8.1  
Two cladograms for  
passerine birds



Use the **Consensus** command in the **Trees** menu to compute the strict and semi-strict consensus of these two trees. If you look at the cluster table in the output you will see that four clusters (3, 4, 7, and 9) occur in both trees, and hence are present in the strict consensus tree.

Cluster table

		1	
	1234567890		
	-----		
1	**.....	1	
2	..**.....	1	
3	...*****	2	
4	..*****	2	
5	.*****	1	
6	....**....	1	
7	....***...	2	
8	.....**.	1	
9	.....***	2	

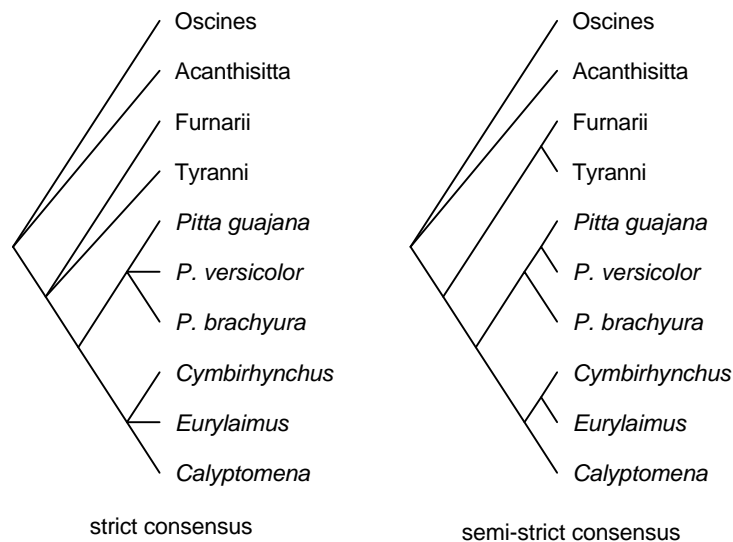
However, the compatibility matrix tells us that clusters 2, 6, and 8, while occurring in only one of the two trees are not contradicted by any other cluster:

Compatibility matrix

2	1
3	11
4	111
5	.111
6	11111
7	111111
8	1111111
9	11111111
	123456789
	-----
	.***.****

For example, cluster 6 which groups *Pitta guajana*, and *P. versicolor* is present in tree 1. In tree 2 these two taxa are part of a trichotomy with *P. brachyura*. For these three taxa the two trees are different but *consistent*. The semi-strict consensus tree comprises all such uncontradicted clusters. Here for comparison are the strict and semi-strict trees:

Figure 8.2  
The strict and semi-strict consensus trees for the two trees in Figure 8.1



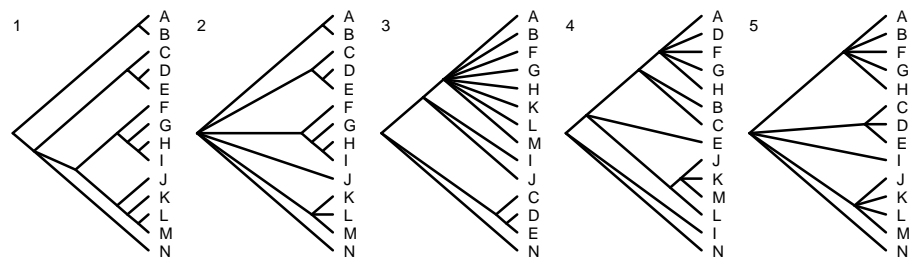
When the input trees contain polytomies it is important to distinguish between trees that differ because one is simply more resolved than the other, and trees that conflict with each other. This is especially relevant when using parsimony programs such as PAUP and Hennig86 that can collapse unsupported nodes.

### Nelson consensus trees

Strict consensus trees are often called "Nelson trees", with Nelson (1979) cited as the source. This is unfortunate, because Nelson's paper does not describe the strict consensus tree but rather an interesting extension of the semi-strict consensus tree. Nelson's method seeks the set of clusters that are most frequently replicated in a set of trees, and any clusters that are compatible with that set. If there is more than one such set, the Nelson tree is the intersection of the sets.

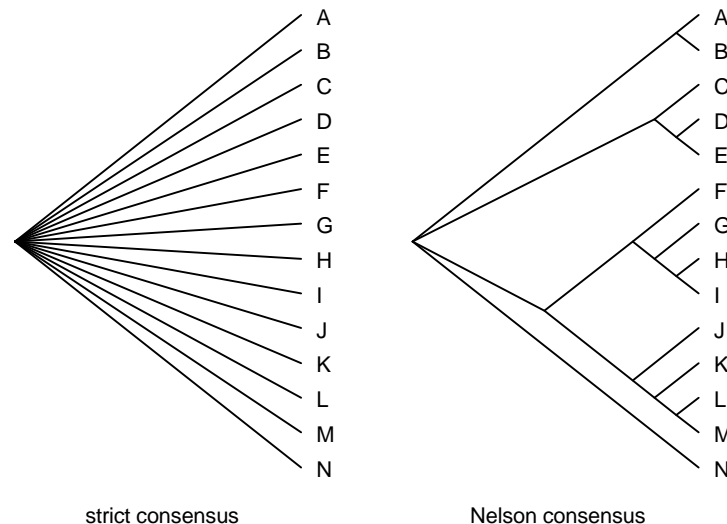
The five trees used by Nelson to illustrate his method are in the file NELSON79.NEX and are shown below:

Figure 8.3  
The five trees used by Nelson (1979) to illustrate his consensus method



Use the **Consensus** command from the **Trees** menu to compute the strict and Nelson consensus trees for these five trees. Below are the strict and the Nelson consensus trees for the five trees.

Figure 8.4  
The strict and Nelson consensus trees for the five trees in Figure 8.3



Look at the output in the Log window. Notice that there is no cluster common to all five trees, so that the strict consensus tree is a star tree, and that no cluster is uncontradicted (hence the semi-strict consensus tree will be identical to the strict consensus tree):

Compatibility matrix

```

2| 1
3| 11
4| 111
5| 1111
6| 11111
7| 111111
8| 1111.11
9| ...1...1.
10| 1...1...111
11| 1111111...
12| 1111.111...
13| 1111111... 1.
14| 11111.1... 1.1
15| 1111111... 1.11
16| 111.11.111 1.111
17| 1111111111 1.1111
18| 1111...1111 .1...11

```

```

          1 11111111
1234567890 12345678
-----
.....

```

COMPONENT also displays the largest sets of compatible clusters (i.e., cliques) that contain the most frequently replicated clusters, and classifies each cluster according to whether the cluster occurs in all the largest cliques ("True"), only some of the largest cliques ("Ambiguous"), or none of the largest cliques ("False").

Largest cliques

```

[ 1..5 7 11 13..15 17 ]
[ 1..7 11 13 15 17 ]

```

Cluster partitions

```

T:[ 1..5 7 11 13 15 17 ]
A:[ 6 14 ]
F:[ 8..10 12 16 18 ]

```

The Nelson consensus tree comprises the "true" clusters.

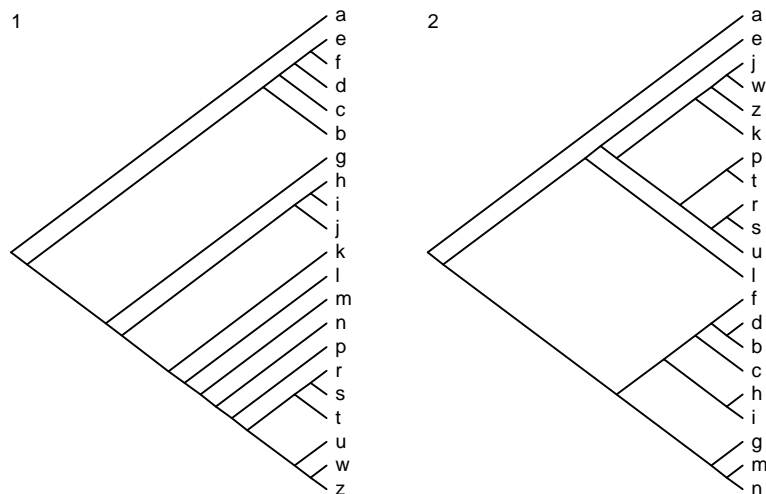
## Agreement subtrees

---

Another kind of consensus tree is the *agreement subtree* (Kubicka, et al. 1992; also known as the *common pruned tree* [Finden and Gordon, 1985]). An agreement subtree is the largest subtree in common to two trees. Unlike the other consensus methods implemented in COMPONENT, agreement subtrees can only be computed for two trees at a time, and the algorithm requires binary trees.

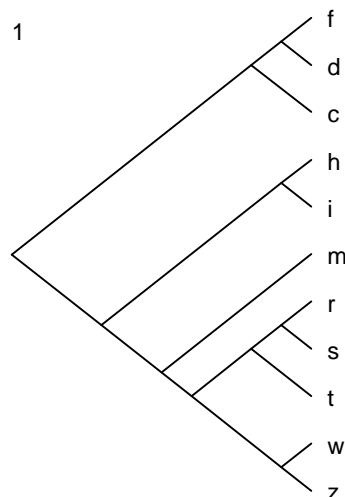
The file AGREE.NEX contains two unrooted trees taken from Kubicka et al. (1992).

Figure 8.5  
Two unrooted binary trees



- From the **Trees** menu choose the **Compare tree with** command to compute the greatest agreement subtree for these two trees. The display buffer will list the leaves pruned to arrive at an agreement subtree, and the subtree itself. In this case, 10 leaves have been removed, leaving this subtree:

Figure 8.6  
An unrooted agreement subtree for the two trees in Figure 8.5



Note that the subtree might not be unique, since there may be other subtrees of equal size. Kubicka et al.'s algorithm will find only one agreement subtree.

### Rooted versus unrooted trees

Computing the agreement subtree for these two trees may have seemed time consuming. One reason for this is that the trees are unrooted. If you root the trees using the **Rooted** command from the **Trees** menu the program takes only a tenth of the time required for the unrooted trees (for an explanation of why this is so see Chapter 5).

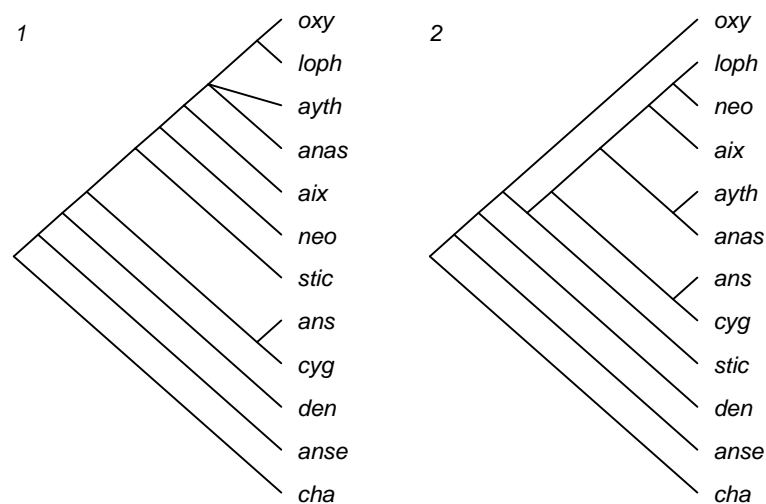
## Comparing trees

COMPONENT features several different tree comparison measures, and also allows you to compare both individual pairs of trees and whole sets of trees.

### Comparing two trees using quartets

Bledsoe and Raikow (1990) recently used Estabrook et al.'s (1985) quartet measures to compare trees constructed from molecular and non-molecular data sets. One example they gave (also discussed by Estabrook, 1992) concerned estimates of waterfowl phylogeny published by Livezey (1986) and Madsen et al. (1988), reproduced as trees 1 and 2, respectively, in the file DUCKS.NEX, and shown below:

Figure 8.7  
Two cladograms for ducks



As before, choose the **Compare tree** command from the **Trees** menu, but this time choose **Quartets**. COMPONENT will compute the selected tree comparison measures and output the results in the display buffer:

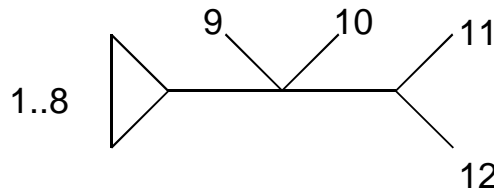
Quartets

SD	EA	DC	SJA	Q	s	d	r1	r2	u
0.415	0.424	0.392	0.405	495	285	194	0	16	0

Of the  $Q=495$  possible quartets for 12 taxa,  $s=285$  are resolved and of the same type in the two trees, and  $d=194$  are resolved but of different types. There are no quartets that are unresolved in both trees ( $u=0$ ), but  $r2=16$  quartets are resolved in tree 2 but not in tree 1.

Looking at tree 1 you can see that it contains a trichotomy. We can redraw tree 1 as:

Figure 8.8  
Tree 1 in Figure 8.7  
redrawn to show how  
the 16 unresolved  
quartets are obtained



Given this tree there are 16 quartets that contain leaves 9 and 10, and have one leaf from the set  $a=\{1,2,\dots,8\}$ , and one leaf from the set  $b=\{11, 12\}$ . Examples include (1, 9, 10, 12) and (3, 9, 10, 11). It is these 16 quartets that are unresolved.

It is worth noting that using quartets to compare trees assumes that the trees are **unrooted**. However, if the trees are **rooted** (say, if UPGMA has been used to construct the trees, or an outgroup has been designated) then triplets are more appropriate. Here are the triplet statistics for the two trees:

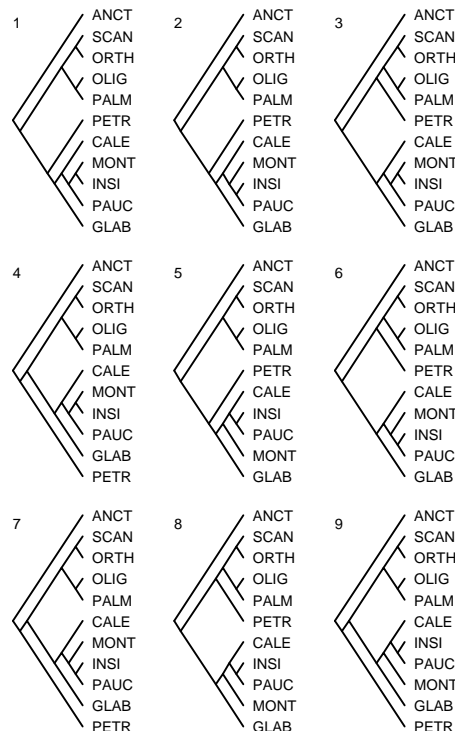
Triplets

SD	EA	DC	SJA	Q	s	d	r1	r2	u
0.201	0.205	0.195	0.197	220	175	43	0	2	0

### Comparing all trees in a profile using the NNI metric

COMPONENT's **Trees | Tree-to-tree distances** command can compute the dissimilarity between all pairs of trees within the same profile of trees. As an example, the file PEG.NEX contains the nine minimal, fully resolved trees for Bolick's (1981) *Salmea* data sets distributed with Hennig86:

Figure 8.9  
The nine most  
parsimonious trees for  
Bolick's (1981) *Salmea*  
data set.



You can explore Maddison's (1991) concept of "islands of trees" using COMPONENT's nearest neighbor interchange (NNI) metric.

- From the **Trees** menu choose **Tree-to-tree distances**. From the sub menu select the **NNI** command. A dialog box will appear displaying the three approximations available; accept the default ( $d_{ra}$ ) by clicking on the **OK** button.

COMPONENT computes and displays the lower left triangle of the pairwise distance matrix for the active trees in the profile. The program also displays a histogram of the frequency distribution of pairwise distances.

Nearest neighbour interchanges  
dra approximation

```

2| 1
3| 1 2
4| 1 2 1
5| 2 1 3 3
6| 2 1 1 2 2
7| 2 1 2 1 2 1
8| 3 2 2 3 1 1 2
9| 3 2 3 2 1 2 1 1
+-----+
      1 2 3 4 5 6 7 8 9

```

If you look carefully you will see that each tree always has another tree that is at most one NNI away from it. In other words, we need make only a single NNI rearrangement to convert one tree into another. We can represent this visually as a graph showing the nine trees connected by single NNIs:

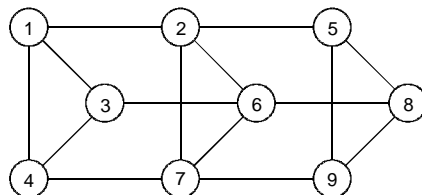


Figure 8.10  
The NNI island formed  
by the nine trees in  
Figure 8.9

In graph theoretic terminology, these nine trees form a single *connected component*, in Maddison's (1991) terminology the trees form a single NNI island. The practical significance of this is that if we were using a heuristic method to find most parsimonious trees, once we discovered any of these nine trees, just using NNI rearrangements alone would suffice to discover the other eight trees. By exploring other sets of equally parsimonious trees in this way we can gain an insight into the performance of various methods of tree rearrangement used in heuristic searches (Page, 1993b).

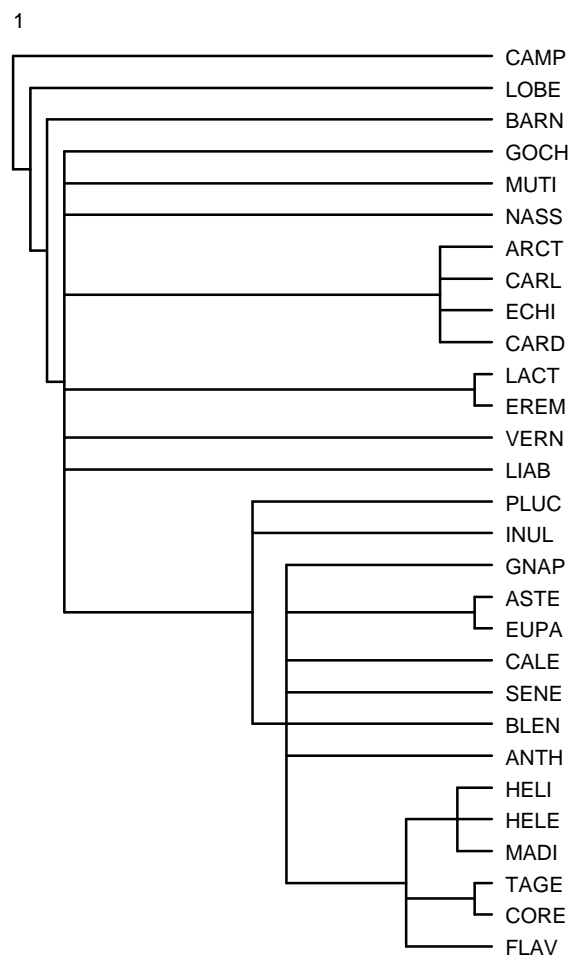
## Comparing one tree with many trees

Sanderson (1989) recently proposed using the distribution of similarity between bootstrap trees and some other tree, such a minimal tree for the original data, as a way of computing a "confidence interval" on a tree. One way to visualise this is to imagine the bootstrap trees as a roughly spherical cloud in "tree space", and we're interested in the trees nearer the centre of that cloud. Trees on the periphery are "outliers." You can use COMPONENT to explore this approach further, and to repeat Sanderson's analysis of Bremer's (1987) data.



One natural candidate for a reference tree is the majority-rule consensus tree (see Chapter 4) of the bootstrap trees. This tree has the nice property of being a median tree, that is, given a set of trees the majority rule consensus tree will be the tree most similar to the most similar, on average, to all the other trees when similarity is measured using the partition metric. You can think of it as being at the centre of the cloud of trees. The file BREMER.NEX imports 50 bootstrap trees from a Hennig86 file called BREMER.TRE. You can use the **Consensus** command to compute the majority-rule consensus tree for these trees. Make sure you save the consensus tree to a file (select the **Save trees** option in the Consensus dialog box). Once you've computed the consensus tree, load it into memory using the **Open** command from the **File** menu. Here is the majority-rule tree for the 50 bootstrap trees:

Figure 8.11  
The majority rule  
consensus tree for 50  
bootstrap trees for  
Bremer's (1987) data set

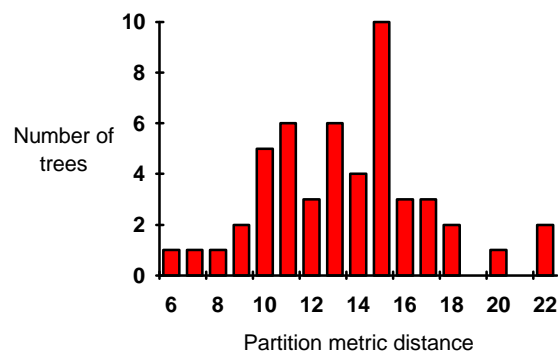


This tree is slightly different from Sanderson's (1989:fig. 1) due to the vagaries of random sampling of characters during bootstrapping.

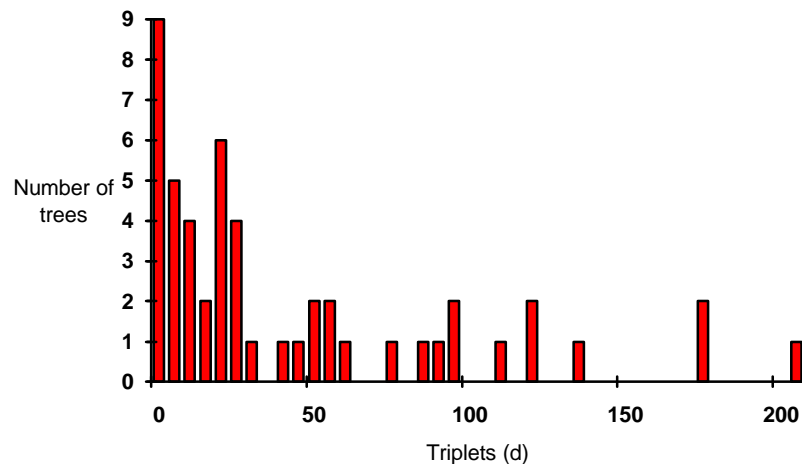
Using the **Compare with** command from the **Trees** menu, you can compare the 50 bootstrap trees with the majority-rule tree using a variety of tree comparison measures. To do this you must have the trees from the file BREMER.NEX in one Tree window, and the majority rule consensus tree in a second Tree window (this tree is stored in the file you created when you computed the consensus tree).

- Make the window containing the consensus tree the active window. From the **Trees** menu choose the **Compare with** command. A dialog box will appear listing the Tree windows that contain trees that are comparable with the consensus tree, in this case the window containing BREMER.NEX.
- After you have clicked on the **OK** button the Between Profile Tree-to-Tree Distances dialog box will appear. For the first analysis use the partition metric, so just click on **OK**. COMPONENT will now compare every tree in the first window with every tree in the second window, i.e., the consensus tree with every bootstrap tree.
- Repeat the analysis but using triplets instead of partitions. Click on **Triplets** to select the triplets measures, then click on **OK**.

Here is the result for partitions:



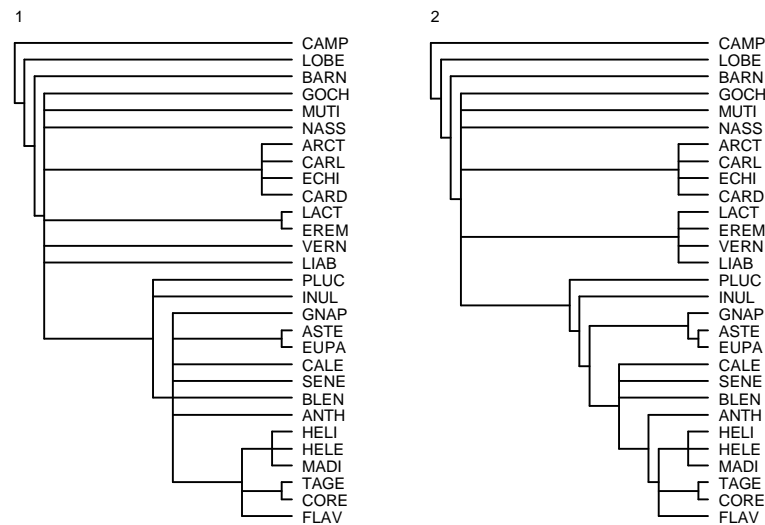
Here is the result for triplets:



Note the quite different distribution for triplets. In part this is because the triplet measure displayed is the number of triplets that are resolved and different in two trees, whereas the partition measure counts each cluster that is found in only one or other of the trees.

Six of the 50 bootstrap trees (trees 2, 26, 27, 29, 31, and 33) do not conflict with the majority-rule tree. For example, here is the majority rule tree (tree 1) and bootstrap tree 2 (these two trees are in the file BREMER.NEX).

Figure 8.12  
The majority rule  
consensus tree (tree 1)  
in Figure 8.11 and one of  
the six bootstrap trees  
that are refinements of  
the consensus tree (tree  
2)



We can compare these two trees in more detail using the **Compare tree with** command. Here are the triplet statistics for the two trees:

Triplets

SD	EA	DC	SJA	Q	s	d	r1	r2	u
0.046	0.275	0.000	0.000	3654	2648	0	2	251	753

Of the 3654 triplets for 50 leaves, the two trees share 2648 identical triplets. The consensus tree resolves 2 triplets that are not resolved in tree 2, and tree 2 resolves 251 triplets not resolved by the consensus tree. A further 753 triplets are unresolved in both trees. There are no triplets that are resolved differently in both trees, hence the two trees are mutually consistent ( $d = 0$ ). The partition distance between the two trees is 6, so that there are six clusters unique to one or other tree.

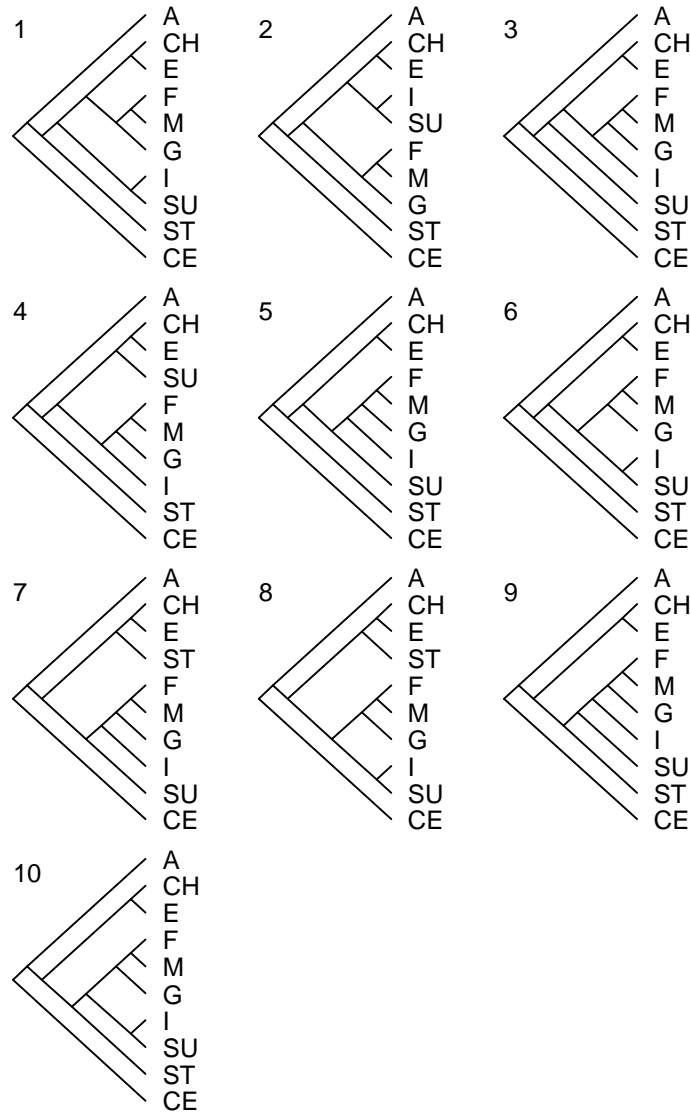
This example shows how important it is to distinguish between trees that are different and inconsistent, and trees that are different but consistent (as are the two trees in Figure 8.12 above). (How would you construct the tree that combines the information in both the consensus tree and tree 2?)

To construct a 95% confidence interval using Sanderson's (1989) method we would include the  $0.95 \times 50 \approx 47$  trees closest to the majority-rule tree. Using the triplet measure we would therefore include all trees with  $d \leq 130$ . Likewise, the 65% confidence interval would comprise the 33 trees with  $d \leq 40$ .

## Comparing all trees in two profiles of trees

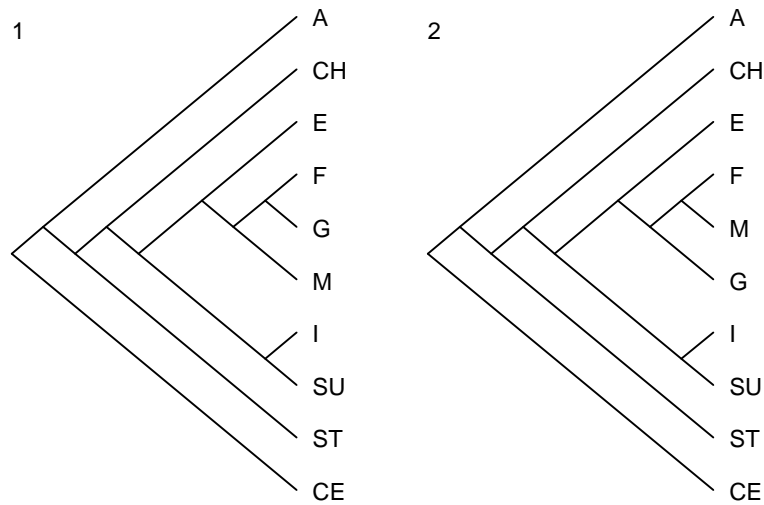
Swofford (1991) recently discussed Kluge's (1989) study of the congruence between two data sets for boid snakes belonging to *Epicrates*. One data set comprised the presence and absence of 24 skin and scent gland lipids (Tolson, 1987), the other data set comprised 53 skeletal and external characters. Here are the 10 minimal trees for the lipid data (EPIB.NEX):

Figure 8.13  
The ten minimal trees for  
the Epicrates lipid data  
set.



and here are the two minimal trees for the morphological data (EPIM.NEX):

Figure 8.14  
The two minimal trees  
for the Epicrates  
morphological data set.



There are no trees common to the two sets. Swofford (1991:319) computed an agreement subtree for all 12 trees together and found it "disheartening that obtaining a common pruned tree [=agreement subtree] requires the removal of nearly half of the taxa." However, combining all 12 trees together confounds differences between trees within the same set with differences between trees from different sets. A better approach would be to compare pairs of trees from the two different sets of trees.

Use COMPONENT's **Trees Compare with** command to compare the two sets of trees. Firstly choose the partition metric. The following table shows the partition distance between each pair of trees from the two sets:

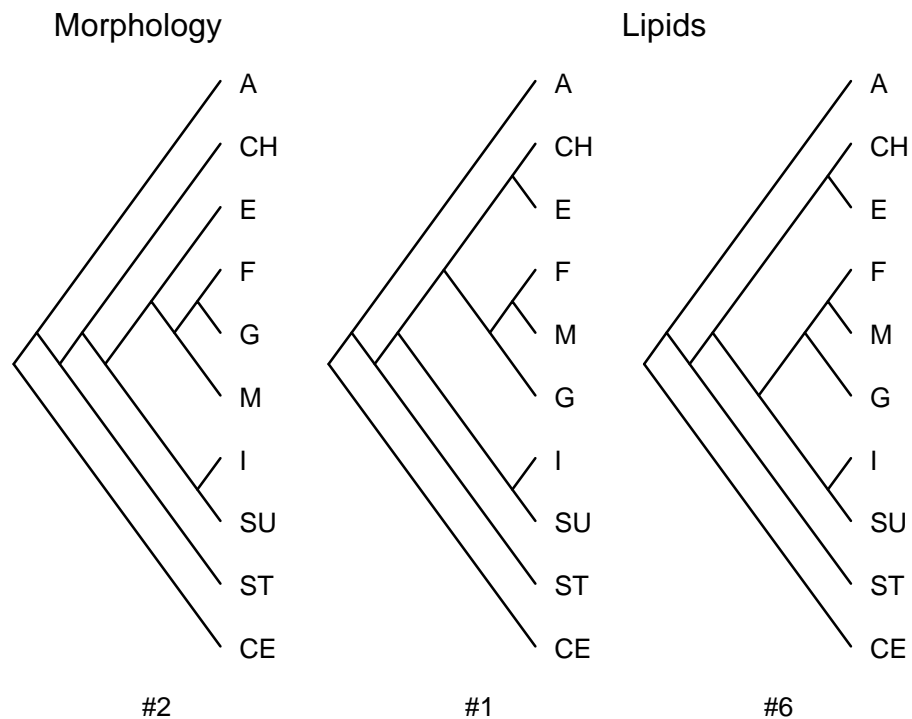
	1	2	3	4	5	6	7	8	9	10
1	6	6	8	8	8	6	10	8	10	8
2	4	4	6	6	6	4	8	6	8	6

Now use the same command, but this time choose the **Agreement subtree** option to compute the number of leaves we need to remove to arrive at the same tree:

	1	2	3	4	5	6	7	8	9	10
1	2	3	3	3	3	2	4	3	4	3
2	1	2	2	2	2	1	3	2	3	2

Note that tree 2 for the morphological data differs in the placement of just one taxon from trees 1 and 6 for the lipid data. Combining all the trees together has obscured the degree of similarity between the two sets of trees. Here for comparison is tree #2 from the morphological set, and trees #1 and #6 from the lipid set:

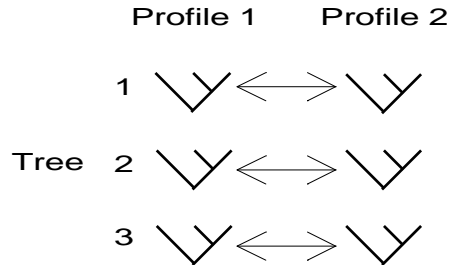
Figure 8.15  
Tree 2 for the morphological data (EPIM.NEX) and trees 1 and 6 for the lipid data (EPIB.NEX) for Epicrates



Note that by removing taxon CH from trees 2 and 1 we obtain an identical subtree, and likewise pruning taxon E from trees 2 and 6 results in the same tree.

## Comparing two sets of paired trees

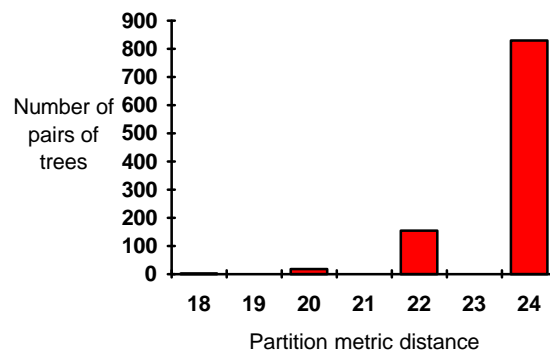
In the previous example we compare every tree in one profile with every tree in a second profile. In this next example we will treat the trees in each profile as if they were paired, like this:



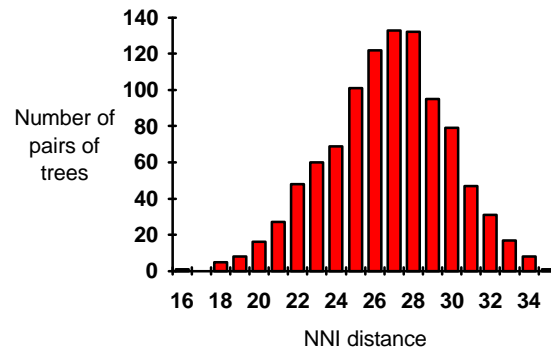
The distributions of tree comparison measures can be estimated by generating large numbers of pairs of random trees and computing the distances between each pair. The resulting distribution of tree-to-tree distances is an estimate of the probability that two trees drawn at random from a set of all possible trees will have a given value of dissimilarity.

- Choose the **Random trees** command from the **Generate** menu. COMPONENT will display the Random Trees dialog box. In the **Number of trees** input box type in the number of trees you want to generate (say 100). In the **Number of leaves** input box type in the number of leaves you want each tree to have (say 15). Click on the **OK** button. COMPONENT will generate the random trees and display them in a Tree window
- Generate a second profile with the same number of same-sized trees by repeating the above step.
- Compare the two profiles using the **Trees Compare with** command as above, but this time click on the **Pairs** radio button in the **Compare** group.

The different tree comparison measures can have quite different distributions. For example, here is the distribution of the partition metric distance between 1000 pairs of unrooted, random binary trees with 15 leaves generated using the EQUIPROBABLE model (so that all labelled binary trees are equally likely to occur):



By way of contrast, the following chart shows the frequency distribution of the  $d_{rs}$  approximation to the NNI metric computed between the same 1000 pairs of trees:



While both distributions emphasise that finding trees with this number of leaves that are very similar is highly unlikely to be due to chance alone, the NNI metric is much more discriminating between trees (there are 25 distinct values of the NNI metric, but only 4 different values of the partition metric).

## Random trees and all possible trees

---

COMPONENT implements a variety of methods for generated random trees, as well as generating all possible trees. You have already encountered random trees in the previous example above.

### Generating all possible tree shapes

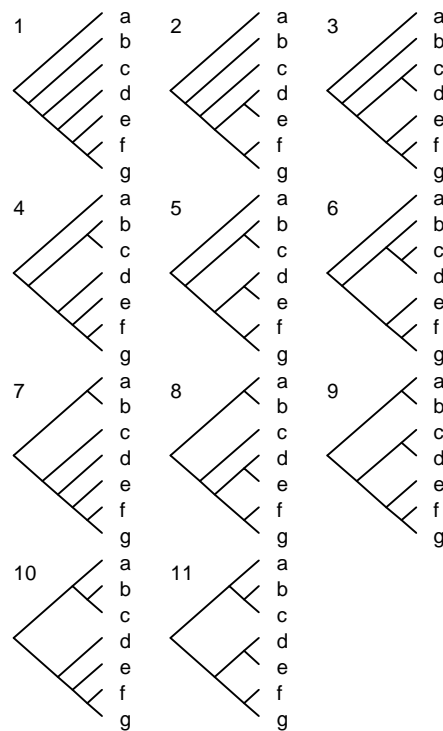
---

A novel feature of COMPONENT is its ability to generate all possible tree shapes. To illustrate, in this example we will generate all the possible rooted tree shapes for seven leaves.

- From the **Generate** menu choose the **All** command. A dialog box will appear. Click on the **Labelled** check box to remove the tick, and click on the **Leaves** spin button until the number 7 is displayed. Click on **OK**.

COMPONENT will generate all 11 possible tree shapes for rooted trees with 7 leaves:

Figure 8.16  
The 11 possible shapes  
for rooted, binary trees  
for seven taxa



In Harding's (1972) notation tree 1 has shape  $\tau_1$ , tree 2 has shape  $\tau_2$ , and so on.



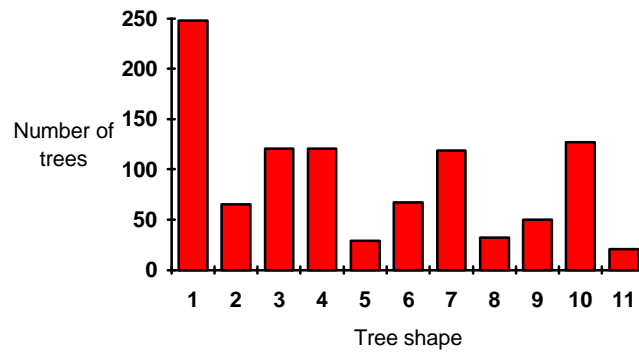
Although strictly speaking the trees are unlabelled trees COMPONENT has arbitrarily labelled the leaves with the letters a-g. This is merely a programming convenience.

## Tree statistics

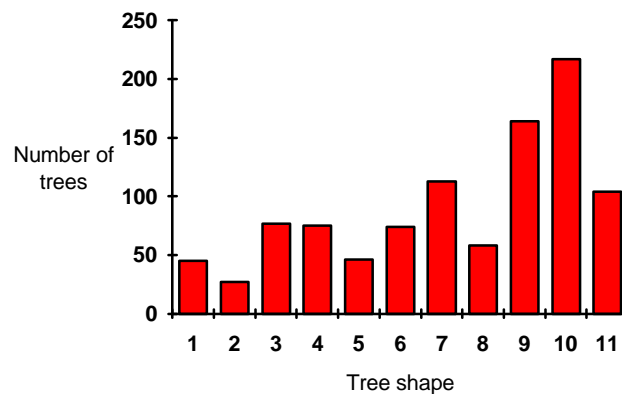
There has been a lot of interest in comparing the observed distribution trees shapes in cladograms constructed by systematists with the distributions predicted by various "null models" (e.g., Savage, 1983; Slowinski and Guyer, 1989; Guyer and Slowinski, 1991). With COMPONENT you can quickly compute the distribution of tree shapes for a set of binary trees using the **Statistics** command from the **Trees** menu.

Firstly we need some random trees. Use the **Random trees** command from the **Generate** menu to generate a set of 1000, rooted (binary) trees with 7 leaves using the EQUIPROBABLE model (the default). Then choose the **Statistics** command to compute the shape of each tree. You should get a distribution of shapes something like this:





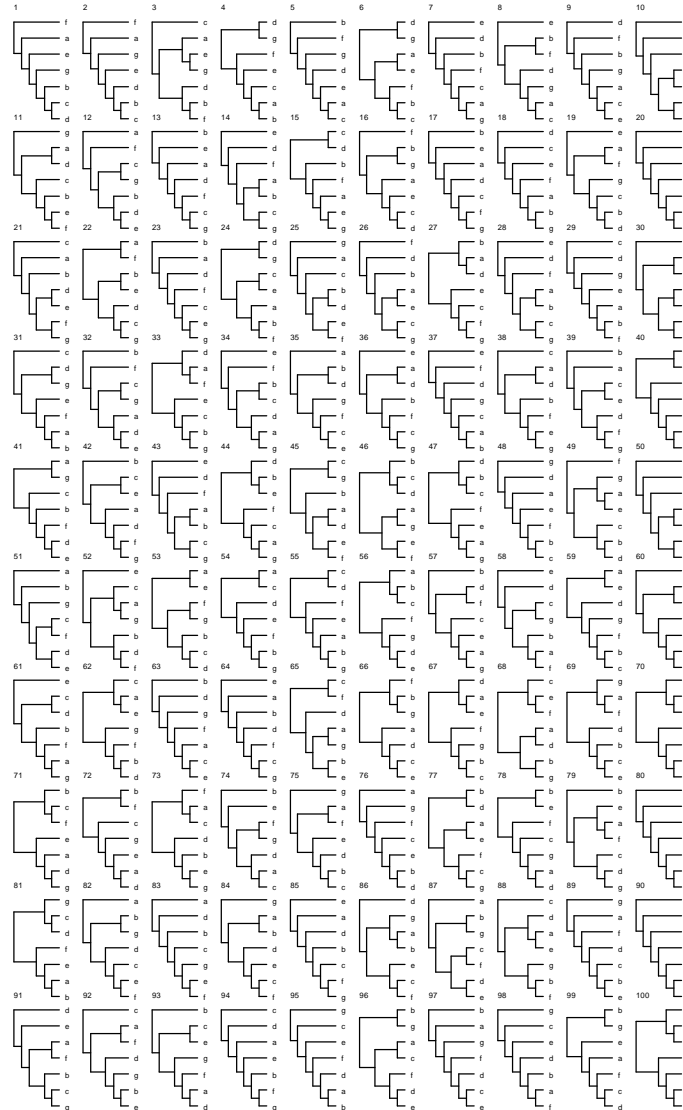
Now generate another set of 1000 trees, but this time use the MARKOVIAN model (select the **Labelled dendrogram** option in the Random Trees dialog box). You should get a distribution of tree shapes something like this:



The two distributions are quite different: unbalanced trees (such as shape 7<sub>1</sub>) are much more common under the EQUIPROBABLE model than under the MARKOVIAN model, whereas more balanced trees (such as shape 7<sub>10</sub>) are more common under the MARKOVIAN model.

Another, more laborious way of counting tree shapes is to order all the trees using the **LLR** method (see Appendix B) using the **Trees Order** command. This ensures that any tree with the same shape will look the same. Then use the **Trees Print** command to display many trees at once and simply count the number of times each shape occurs. An example is shown below:

Figure 8.17  
100 random 7-trees in  
LLR order



## Tree mapping

Tree mapping is a procedure where one tree is mapped onto another. This map defines a third tree called a "reconciled" tree which, as the name implies, reconciles any incongruence between the two trees. Tree mapping can be used to explore host-parasite cospeciation, incongruence between gene trees and species trees, and cladistic biogeography (Page, 1990a; 1990b; 1993b; submitted).

## Host and parasite trees

### Introduction

Page (1990b) used tree mapping to reanalyse Hafner and Nadler's (1988) data on pocket gophers and their parasitic chewing lice. Below are the dendrograms Hafner and Nadler obtained using allozyme data for eight gophers and their lice (these trees are in the file LICE.NEX).



The statistics for this tree and the map between the two trees are displayed in the Log window:

Reconciled tree statistics

```
Duplications = 4
Total leaves = 36
Leaves added = 26
Losses = 10
```

Map between associate and host tree

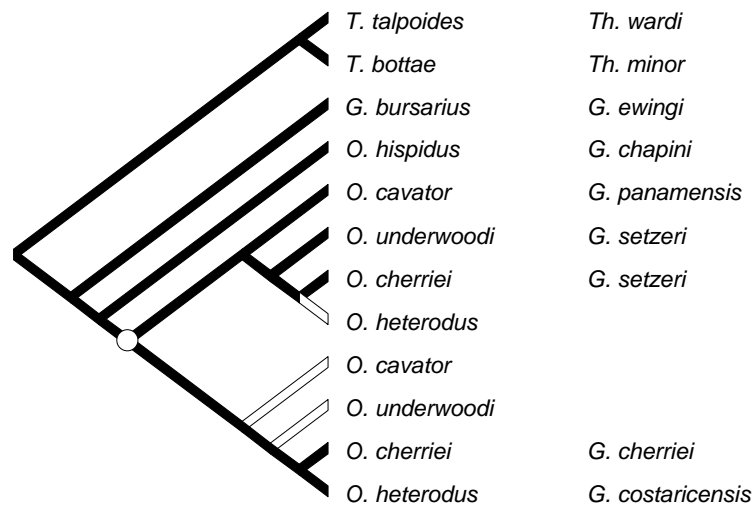
Associate		Host	Duplication?
11	->	9	YES (OVERLAP)
12	->	10	NO
Th. wardi	->	T. talpoides	-
Th. minor	->	T. bottae	-
13	->	9	YES
14	->	9	YES
G. thomomyus	->	T. talpoides	-
15	->	9	NO
G. actuosi	->	T. bottae	-
G. ewingi	->	G. bursarius	-
16	->	12	NO
G. chapini	->	O. hispidus	-
17	->	13	YES (OVERLAP)
18	->	13	NO
G. panamensis	->	O. cavator	-
G. setzeri	->	14	-
19	->	15	NO
G. cherriei	->	O. cherriei	-
G. costaricensis	->	O. heterodus	-

The reconciled tree is the most parsimonious summary of the history of the gopher-lice association under the assumption that there has been no host switching. As you can see the reconciled tree requires numerous ad hoc hypotheses to explain the incongruence between the lice and their hosts.

### Pruning parasites

There are a number of further analyses you could apply to this data. For example, you could see what happens if you allow one or more lice to have dispersed by deleting those lice then reconciling the pruned tree with the gopher tree. To do this use the **Prune or graft leaves** command from the **Trees** menu (see Chapter 2). Try pruning *Geomydoecus actuosi* and *G. thomomyus*, two lice that Hafner and Nadler (1988) suggested have dispersed. You should get a much better fit:

Figure 8.20  
The reconciled tree for  
the gopher and lice trees  
shown in Figure 8.18  
after pruning the lice  
*Geomydoecus actuosi*  
and *G. thomomys*

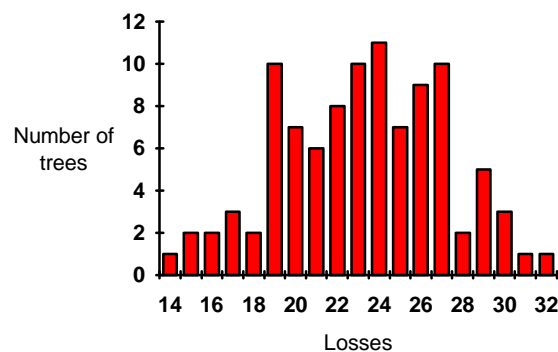


This tree requires just one duplication, which results in two sister lineages of lice parasitizing four species of *Orthogeomys*. Page (1990b; 1993a) discusses the evidence supporting this hypothesis of two sympatric clades.

You can automate the process of deleting the lice by using the **Prune each leaf** command. This deletes each associate in turn and computes the fit statistics for the corresponding reconciled tree. You can use this information to help decide which associate(s) are contributing most any incongruence between the two trees (see Chapter 7).

### Testing hypotheses of cospeciation

Page (1990a, 1990b) proposed testing hypotheses of cospeciation by comparing the measures of fit computed for the host and parasite trees with the distribution of the same measures between the observed parasite tree and a set of random host trees. To illustrate the idea, the file LICERAND.NEX uses the RANDOM command (see Chapter 3) to generate 100 random trees for the gophers. Open the file LICERAND.NEX and use the **Map onto all trees** command to compute the fit statistics between the lice tree and the random trees. This next figure shows the distribution of losses for the 100 trees.



As we can see, all 100 trees have more than the 10 losses actually observed, suggesting that the observed value is significant. Of course in practice 100 is a rather small number of trees, but suffices to illustrate the idea. More accurate estimates of the distribution can be obtained using 1000 or more trees.

## Area cladograms

Rosen's (1979) data on *Xiphophorus* and *Heterandria* have often been used to illustrate cladistic biogeographic methods. Rosen's data is on disk as ROSEN.NEX.

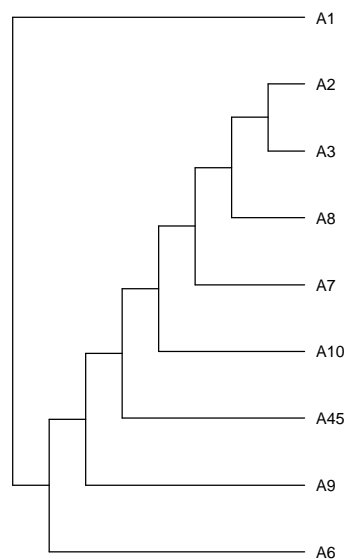
### Calculating a global area cladogram

To compute an area cladogram for both fish genera together:

- Choose the **Heuristic search** command from the **Map trees** sub menu.
- COMPONENT will display the Heuristic search dialog box (see Chapter 7). Accept the default options of using NNI branch swapping, minimising leaves added, and including all blocks (i.e., searching for the best area cladogram for both taxa).

COMPONENT will find a single area cladogram:

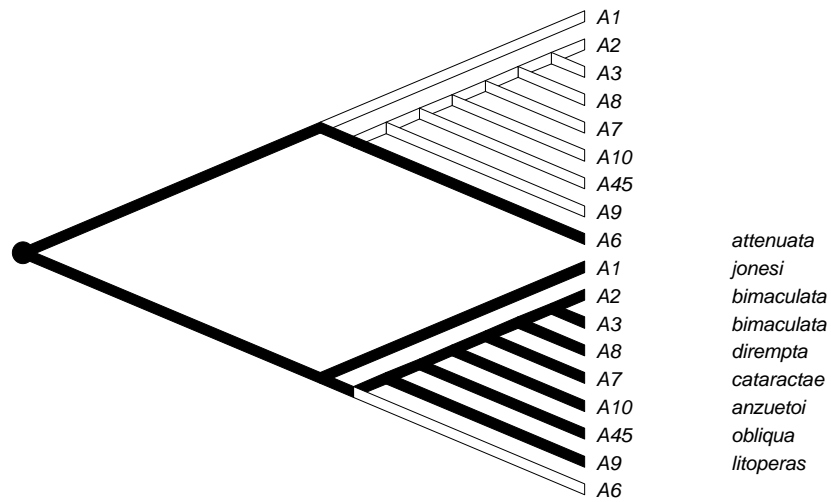
Figure 8.21  
The single area  
cladogram for  
Heterandria and  
Xiphophorus combined



The statistics for this tree are duplications=4, leaves added=32, losses=18.

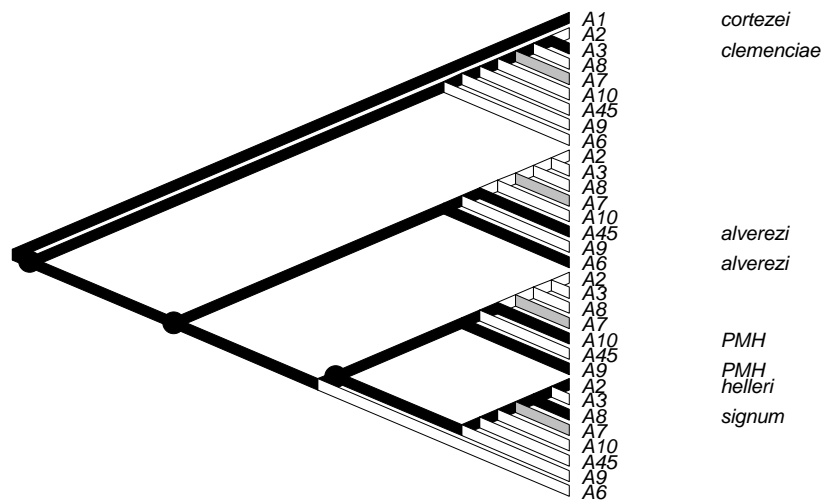
Having computed the area cladogram you can reconcile it with each taxon cladogram. Here is the reconciled tree for *Heterandria*:

Figure 8.22  
The reconciled tree for *Heterandria* and the area cladogram shown in Figure 8.21



The fit statistics are duplications=1, leaves added=10, losses=3. The reconciled tree for *Xiphophorus* is even less appealing (duplications=3, leaves added=22, losses=15):

Figure 8.23  
The reconciled tree for *Xiphophorus* and the area cladogram shown in Figure 8.21



The terminal branches for area A7 in the reconciled tree for *Xiphophorus* are shaded light grey to indicate that no *Xiphophorus* occur in that area. By default this absence from area A7 is treated as "missing data" and so does not contribute to the measures of fit between the taxon and area cladograms (see Chapter 7).

Note that the fit statistics for the combined area cladogram are the sums of the statistics for the two taxa taken separately.

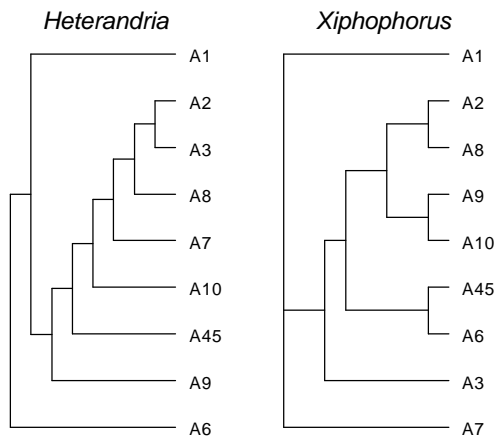
### Calculating area cladograms separately

Given that both fish have relatively poor fit statistics for the combined area cladogram, you might like to compute area cladograms separately for each fish. To do this:

- Select the *Heterandria* block.
- Choose the **Heuristic search** command as above, but check the **Current block only** radio button in the **Include** group in the Heuristic Search dialog box.

COMPONENT will now search for the best area cladogram for *Heterandria* alone. Use the **Trees Save to file** command to store the tree(s) in a separate file. Repeat the analysis for *Xiphophorus*. Below are shown the single optimal area cladogram for *Heterandria* (tree 1) and the Adams consensus tree (tree 2) of the 15 equally optimal trees for *Xiphophorus* (each of which differs solely in the placement of area 7).

Figure 8.24  
The single area cladogram for *Heterandria* and the Adams consensus of the 15 equally parsimonious area cladograms for *Xiphophorus*



Inspection reveals that the two trees agree on the relationships of areas 1, 2, 45, 8, and 10 (you could find this out by computing an agreement subtree for the two trees, see Chapter 5 and the example above). This is Rosen's original result.

### Interpreting incongruence due to widespread taxa

Ignoring area 7 (about which *Xiphophorus* is uninformative) note that the areas about which the two fish disagree (i.e., 3, 6 and 9) are all part of the range of a widespread taxon (Platnick, 1981). Geographically adjacent areas can share the same taxon due to a failure to speciate rather than close relationship (Nelson and Platnick, 1981; Page, 1989a). Nelson and Platnick developed their "Assumption 2" to cope with this possibility. This method removes from the range of a widespread taxon any areas that harbour endemics of another clade (Page, 1988); the deleted areas are allowed to "float" on the area cladogram, their position being determined by the relationships of the taxa endemic to those areas.

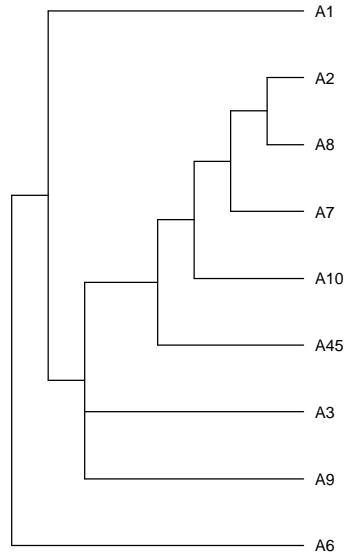
You can do an Assumption 2-like analysis of Rosen's data in COMPONENT by editing the ranges of the widespread taxa. I've already done this for you in the file PLATNICK.NEX. If you look at the file you'll see that I've simply put comment symbols ("[" and "]") around the areas I want to delete, e.g.:

```
.
alverezi      : A45 [A6],
PMH           : [A9] A10,
```



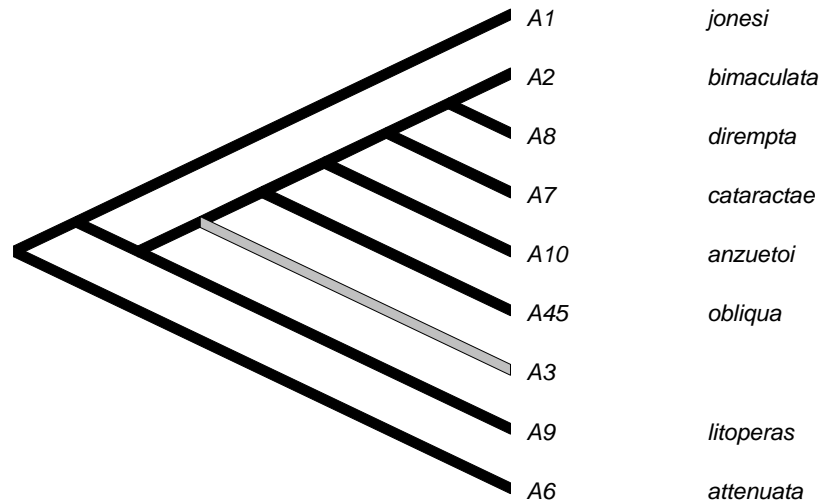
If you now search for the area cladogram(s) for the two fish combined you will obtain the three trees found by Platnick (1981) and Page (1989). The strict consensus of these trees is:

Figure 8.25  
 Strict consensus of the three optimal area cladograms for *Heterandria* and *Xiphophorus* when the widespread taxa are treated under Assumption 2



The statistics for the three trees are duplications=0, leaves added=0, and losses=0. Just as before you can reconcile each taxon tree with the area cladogram. Here is the result for *Heterandria*:

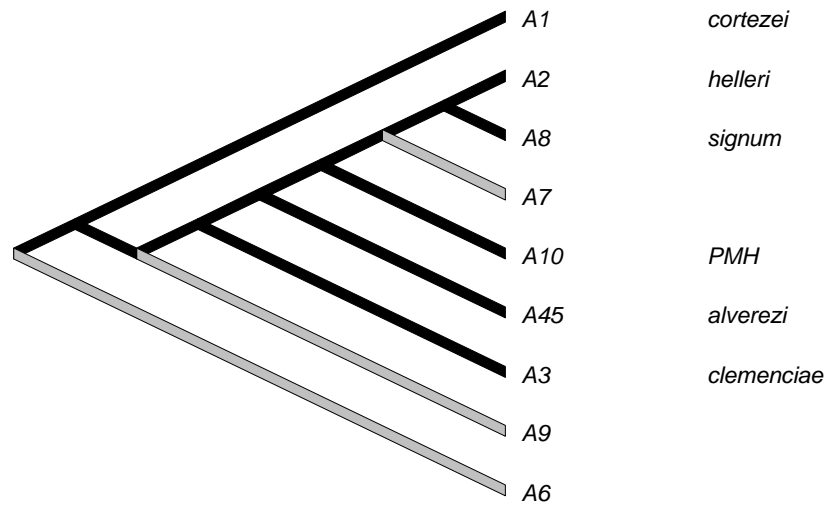
Figure 8.26  
 The reconciled tree for *Heterandria* and the area cladogram shown in Figure 8.25



Note that area 3 has no taxon because we've deleted the occurrence *H. bimaculata* from that area, hence *Heterandria* places no constraints on the relationships of area 3.

Below is the reconciled tree for *Xiphophorus*:

Figure 8.27  
The reconciled tree for  
Heterandria and the area  
cladogram shown in  
Figure 8.25



In this case we've deleted area 6 from the range of *X. alverezi*, and area 9 from the range of taxon "PMH."

Much attention has been paid to widespread taxa in the biogeographic literature. Widespread taxa are a source of ambiguity, and possibly misleading information, but the only evidence for the latter will come from comparisons with area cladograms generated for other taxa. As always, congruence is the arbiter.