Chapter 5
Comparing trees

This chapter describes the tree comparison measures available in COMPONENT, and the various ways you can compare trees using the program. Among the possible uses of tree comparison measures are:

- comparing trees for the same taxa computed from different data sets to measure taxonomic congruence (e.g., Penny et al., 1982; Bledsoe and Raikow, 1990).
- comparing bootstrap trees with a reference tree (e.g., Sanderson, 1989).
- identifying islands of trees (e.g., Maddison, 1991; Page, 1993b).

COMPONENT implements the partition and nearest neighbor interchange metrics, triplets, quartets, and agreement subtrees.

Overview

A tree comparison measure is some measure of the similarity between two trees, T_1 and T_2. There are two basic kinds (Boorman and Olivier, 1973). The first counts the minimum number of operations required to transform T_1 into T_2 using some method of transforming trees. The second represents the two trees as sets of simpler structures (such as clusters or quartets) and then uses various measures of similarity between sets.

Transforming one tree into another

A good example of a measure defined in terms of transforming one tree into another is the nearest nearest neighbor interchange (NNI) metric (e.g., Waterman and Smith, 1978) which measures the minimum number of NNIs required to change T_1 into T_2. In the example below, one NNI is required to convert T_1 into T_2, so $d_{NNI}(T_1, T_2) = 1$.

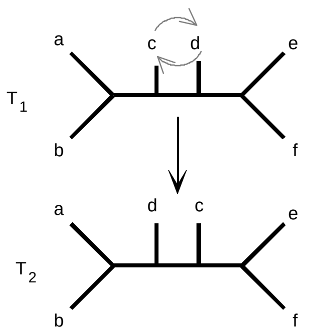
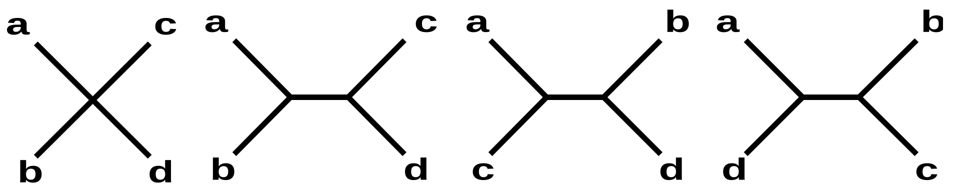


Figure 5.1 Transforming T_1 into T_2 by a single nearest neighbor interchange of leaves c nd d

Trees as sets

Estabrook et al.'s (1985) quartet measures are an example of treating trees as sets of simpler structures. The trees T_1 and T_2 in Figure 5.1 each contain 15 unrooted four-leaf subtrees called quartets. The next figure shows the four possible resolutions of the quartet abcd: abcd, ab|cd, ac|bd, and ad|bc.

Figure 5.2 The four possible quartets for four taxa



The following table lists each quartet and how they are resolved in the two trees:

Table 5.1 The quartets for the two trees in Figure 5.1

Quartet	T_1	T_2	
abcd	ab|cd	ab|cd	
abce	ab|ce	ab|ce	
abcf	ab|cf	ab|cf	
abde	ab|de	ab|de	
abdf	ab|df	ab|df	
abef	ab|ef	ab|ef	
acde	ac|de	ad|ce	†
acdf	ac|df	ad|cf	†
acef	ac|ef	ac|ef	
adef	ad|ef	ad|ef	
bcde	bc|de	bd|ce	†
bcdf	bc|df	bd|cf	†
bcef	bc|ef	bc|ef	
bdef	bd|ef	bd|ef	
cdef	cd|ef	cd|ef	

Four of the 15 quartets are resolved differently in the two trees (those marked † above). Estabrook et al. (1985) describe various measures of tree dissimilarity based on the number of quartets in common to two trees. One measure, EA, is the proportion of quartets that are resolved and identical in the two trees; in the example above $EA(T_1, T_2) = 11/15 = 0.73$. Other measures allow for quartets that may be resolved in one tree but not in the other due to polytomies (see below).

These two categories of tree comparison measure are not mutually exclusive. The well known partition metric (Penny and Hendy, 1985) belongs to both categories. It can be defined as either (1) the minimum number of "contraction" and "decontraction" operations (Borque, 1978) required to transform one tree into another (Robinson and Foulds, 1981), or (2) as the number of partitions (clusters if the trees are rooted) found in one or other but not both trees (Hendy et al., 1984).

Choosing a tree comparison measure

Choosing a comparison measure depends on what aspect of tree structure you are interested in comparing. Other considerations are computational speed and accuracy.

The partition metric is easy to compute (Day, 1985), widely available (it is also implemented in PAUP 3.0 and COMPONENT 1.5) and treats trees as sets of clusters, which is how most biologists interpret trees. However, its resolution is poor and two trees differing solely in the postion of one taxon can be maximally different (Penny and Hendy, 1985).

Because NNIs are frequently used to rearrange trees in heuristic searches for most parsimonious trees the NNI metric is particularly useful for studying islands of trees (Maddison, 1991; Page, 1993b). Its main disadvantage is that no exact, efficient algorithm for its computation is known (Brown and Day, 1984; Krivánek, 1986).

Agreement subtrees are useful for identifying trees that differ in the placement of one or more taxa but are otherwise very similar, avoiding a limitation of the partition metric.

Quartet measures of tree similarity may be most useful in comparing trees constructed using invariants defined on sets of four taxa (e.g., Lake, 1987; Sidow and Wilson, 1990; Steel, 1992).

Kinds of comparison

COMPONENT enables you to make several kinds of tree comparisons. You can

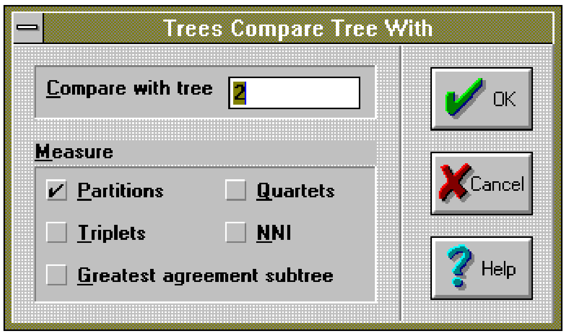
- compare any two trees in the same profile.
- compare all trees in the same profile with themselves.
- compare all trees in one profile with all the trees in another profile.
- compare all ordered pairs of trees from two profiles.

☞ *COMPONENT's tree comparison commands operate only on the active trees in the currently active block.*

Comparing two trees in the same profile

You can ask COMPONENT to compare the tree displayed in the active tree window with any other tree in the same profile. Unlike the other comparison commands **Compare Tree with** allows you to make several different kinds of comparisons at once.

- From the **Trees** menu choose **Compare tree with**. COMPONENT displays this dialog box :



Trees Compare Tree With

Compare with tree: 2

Measure

[✓ Partitions] Quartets

Triplets NNI

Greatest agreement subtree

OK Cancel Help

Figure 5.3
The Compare Tree With dialog box

- Use the **Compare with tree** input box to specify which tree you want to compare the currently displayed tree (by default the dialog box lists the next tree in the profile.) Check the boxes corresponding to the tree comparison measures you want to use. Some measures require the trees to be binary and hence will not be available if not all the trees in the profile are binary.

COMPONENT will compute the selected tree comparison measures and output the results in the display buffer.

Comparing all trees in the same profile

COMPONENT can compare every tree in the current profile with every other tree in the same profile, producing pairwise distance matrix.

- From the **Trees** menu choose **Tree-to-tree distances**. You will see a submenu listing the tree comparison measures available:

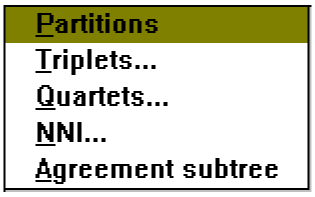


Figure 5.4
The Tree-to-tree distances submenu

If some of the trees in the profile are not binary then you will not be able to compute the NNI or agreement subtree measures (see below).

From the submenu select the tree comparison measure you want to use. If you choose the **Triplets**, **Quartets**, or **NNI** measures you will see a dialog box displaying various choices of method of computation or statistic to be computed.

NNI

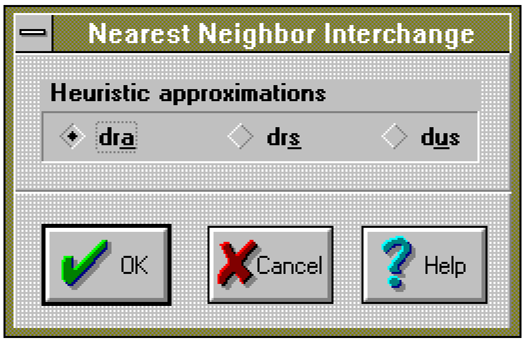


Figure 5.5
The Nearest Neigbor Interchange dialog box

For NNI measures you have a choice of three approximations to the NNI metric. The order of accuracy is $d_{ra} < d_{rs} < d_{us}$ (in other words, d_{us} is the most accurate). Not surprisingly, the order of speed of execution is the reverse. Computing d_{rs} takes twice as long as d_{ra}. Given trees with n leaves, computing d_{us} takes n times longer than d_{rs}. See below for more details.

Triplets and Quartets

When comparing trees using quartet of triplet measures a dialog box will appear offering a choice of measures for COMPONENT to output:

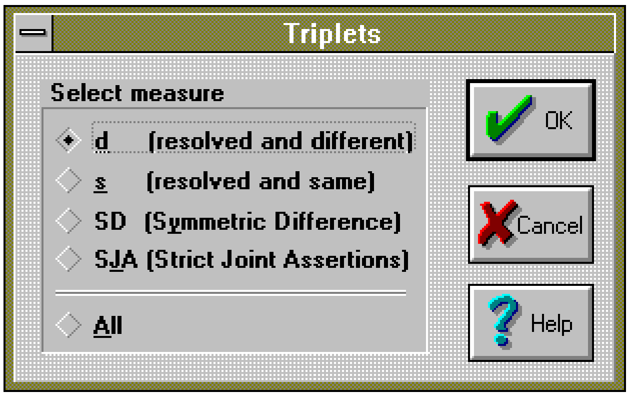


Figure 5.6
The Triplets dialog box

Triplets
Select measure
[✓ d (resolved and different)]
s (resolved and same)
SD (Symmetric Difference)
SJA (Strict Joint Assertions)
All
OK
Cancel
Help

If you select **d**, **s**, **SD**, or **SJA**, then the program will output the lower left triangle of the appropriate pairwise distance matrix. If you select **All** then the program outputs all the statistics (including d, s, r_1, r_2, and u), and each tree comparison takes up a whole line. Since COMPONENT calculates all these statistics anyway your choice will not affect the speed of computation, only the extent of the output.

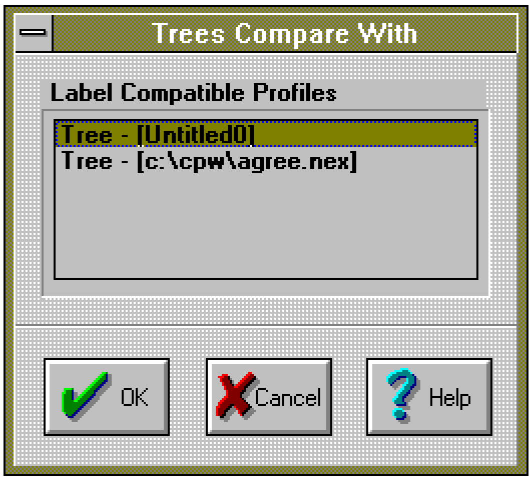
- COMPONENT computes and displays the lower left triangle of the pairwise distance matrix for the active trees in the profile. The program also displays a histogram of the frequency distribution of pairwise distances.

Comparing all trees in two profiles

COMPONENT can compare the trees in two different profiles (i.e., in two different Tree windows). The trees in the two profiles need not have the same number of leaves, but must have at least three (four if the trees are unrooted) leaves in common.

■ Choose the **Compare with** command from the **Trees** menu. A dialog box will appear listing all the Tree windows containing trees that can be compared with the trees in the currently active Tree window:

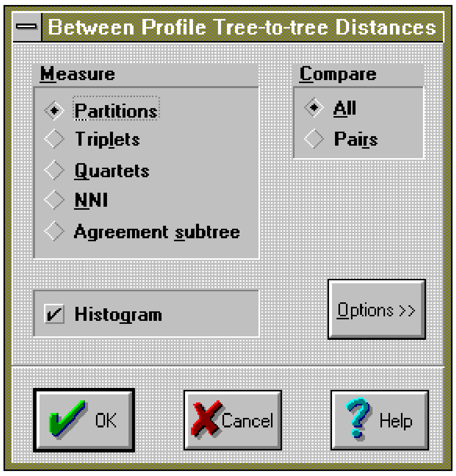
Figure 5.7
The Trees Compare With dialog box



If the **Compare with** command is grayed then there are no comparable Tree windows. To be comparable the profiles must have at least three (four if the trees are unrooted) leaves in common.

■ Once you've selected a Tree window to compare with you will be presented with a dialog box listing the available tree comparison measures:

Figure 5.8
The Between Profile Tree-to-tree Distances dialog box



Choose the measure you want to use. If you click on the **Options** button the dialog box will expand to offer choices for the **Triplets**, **Quartets**, and **NNI** options.

■ The **Compare** group box lists two choices for how you want the two profiles to be treated. The default (**All**) is to compare every tree in the first profile with every tree in the other profile. Hence tree 1 in the first profile is compared with every tree in the other profile (as shown below), then tree 2 in the first profile is compared with every tree in the second profile, and so on:

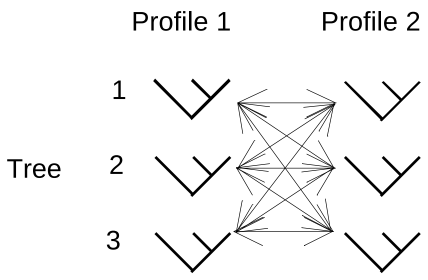


Figure 5.9
The ***All*** *option compares every tree in the first profile with every tree in the second profile.*

The other option (**Pairs**) treats the trees in two profiles as if they were paired, so that tree one in the first profile is only compared with tree one in the second profile, tree two with tree two, as shown below.

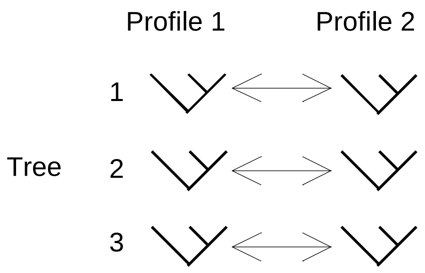


Figure 5.10
The ***Pairs*** *option treates the two profiles as paired*

COMPONENT computes and displays the pairwise distances between the trees in the two profiles. If you checked the **Histogram** check box, the program also displays a histogram of the frequency distribution of pairwise distances.

Agreement subtrees

An agreement subtree of two trees is an identical subtree that can be obtained from both trees by pruning leaves with the same label. Finden and Gordon (1985) refer to these trees as "common pruned trees." A greatest agreement subtree (GAS) is a subtree that results from pruning the fewest number of leaves. For example, T_3 below is a greatest agreement subtree of T_1 and T_2.

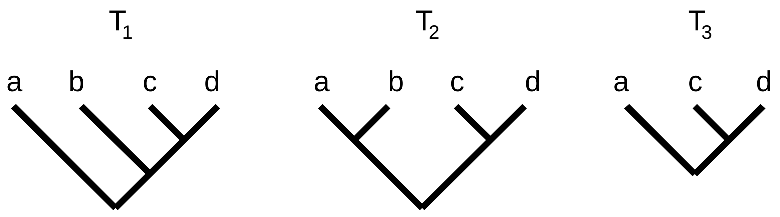


Figure 5.11
Two trees T_1 and T_2 and an agreement subtree, T_3

There may be more than one greatest agreement subtree. In the example above, (b,(c,d)) is also an agreement subtree.

Dissimilarity

Given two trees, T_1 and T_2, we can define the distance d_{GAS} (T_1, T_2) as the number of leaves removed to obtain a greatest agreement subtree.

Algorithm

COMPONENT uses Kubicka et. al's (1992) algorithm to find a greatest agreement subtree for two binary trees, and hence, $d_{GAS}(T_1, T_2)$.

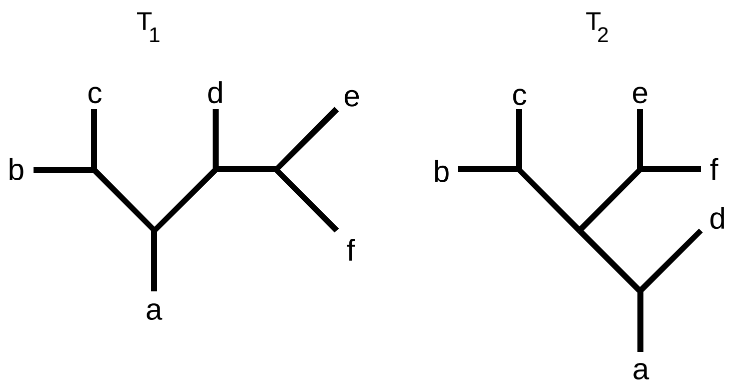


Figure 5.12
Two unrooted binary trees

Given two unrooted binary trees, such as T_1 and T_2 in Figure 5.12 Kubicka et al's (1992) algorithm uses a recursive procedure AGREE (T_1, T_2, a) to find the greatest agreement subtree for T_1 and T_2 that contains the leaf a:

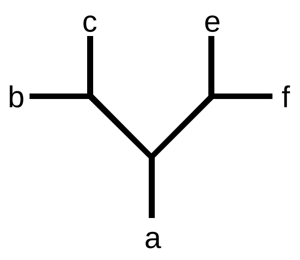


Figure 5.13
A greatest agreement subtree for the two trees in Figure 5.12

This procedure is repeated for every leaf in the tree, resulting in a collection of subtrees. The largest of these is the greatest agreement subtree for T_1 and T_2.

☞ *Kubicka et al.'s (1992) algorithm finds a single greatest agreement subtree. While the algorithm guarantees that there is no agreement subtree larger than the one it finds, there may be other agreement subtrees of equal size.*

Restrictions

The trees being compared must be binary (i.e., fully resolved).

Rooted versus unrooted trees

If the two trees are unrooted then COMPONENT computes AGREE (T_1, T_2, *i*) for all $1 \leq i \leq n$, as described above.

A rooted tree can be visualised as an unrooted tree with an additional leaf ("x") that has been "pulled down" to root the tree (see Chapter 0). Consequently to find a greatest agreement subtree for two rooted trees we need only compute AGREE (T_1, T_2, x). As a result, finding the agreement subtree for two trees takes (often substantially) less time if the trees are rooted rather than unrooted.

When comparing two profiles, unless the trees in both profiles are rooted COMPONENT will treat both sets of trees as unrooted.

Output

If you are comparing two trees with the **Compare with tree** command from the **Trees** menu then COMPONENT will list the leaves deleted from the two trees to obtain an agreement subtree and display the subtree:

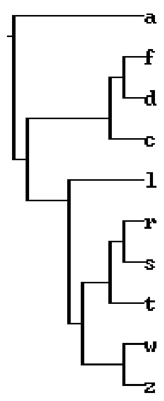
Figure 5.14 Example output for an agreement subtree

```
Greatest agreement subtree

   Leaves in subtree: 10
   Leaves pruned: 11
      e
      b
      g
      h
      i
      j
      k
      m
      n
      p
      u

Subtree (may not be unique)

TREE  =
```



Otherwise the program will output just the number of leaves that must be deleted to obtain an agreement subtree.

Nearest neighbor interchange metric

Given two unrooted binary trees T_1 and T_2, the distance $d_{NNI}(T_1, T_2)$ between those trees is the smallest number of nearest neighbor interchanges (NNI) required to transform one tree into another (Robinson, 1971; Waterman and Smith, 1978).

A **N**earest **N**eighbor **I**nterchange (NNI) is the interchanging of two of the subtrees incident to an internal edge (=branch) in a binary tree. Two such interchanges are possible for each internal edge.

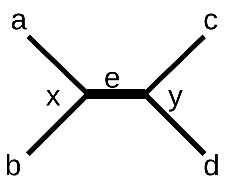


Figure 5.15 An unrooted tree for four taxa

For example, in the tree in Figure 5.15 nodes x and y are adjacent to edge e, nodes a and b are incident to node x, and nodes c and d are incident to node y. Interchanging nodes a and d (or b and c) results in the tree:

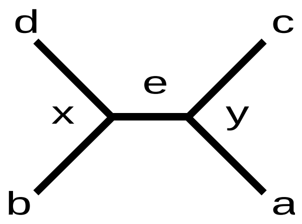


Figure 5.16 The tree produced from the tree in Figure 5.15 after interchanging nodes a and d (or b and c)

Similarly, interchanging nodes a and c (or b and d) results in the tree:

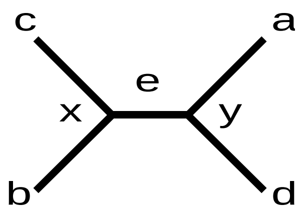


Figure 5.17 The tree produced from the tree in Figure 5.15 after interchanging nodes a and c (or b and d)

Algorithm

The computational complexity of computing $d_{NNI}(T_1, T_2)$ for labeled trees is unknown, although for unlabeled trees the problem is NP-complete (Kriváne k, 1986). Brown and Day (1984) have developed an efficient approximation which is implemented in COMPONENT.

Brown and Day (1984) describe three approximations to d_{NNI} (T_1, T_2):

1. $d_{ra}(T_1, T_2, m)$, which is an upper bound on the minimal number of NNI required to transform T_1 into T_2, where both trees are arbitrarily rooted at leaf *m*. This measure is asymmetric since $d_{ra}(T_1, T_2, m)$ does not always equal $d_{ra}(T_2, T_1, m)$.

2. $d_{rs}(T_1, T_2, m)$, which is the smaller of $d_{ra}(T_1, T_2, m)$ and $d_{ra}(T_2, T_1, m)$. This measure is symmetrical by definition.

3. d_{us} (T_1, T_2), which is smallest value of $d_{rs}(T_1, T_2, m)$ for all *m*.

COMPONENT implements all three measures. When computing just the value of $d_{ra}(T_1, T_2, m)$ or $d_{rs}(T_1, T_2, m)$, T_1 and T_2 are arbitrarily rooted with their first leaves (i.e., $m = 1$).

☞ *The trees being compared are treated as unrooted trees (regardless of their current rooting).*

Restrictions

The trees being compared must be binary (i.e., fully resolved).

Rooted versus unrooted trees

COMPONENT treats the trees being compared as unrooted trees, regardless of whether or not they are rooted.

Distribution

The table below gives the exact distribution of the probability of observing a given value of d_{NNI} for unrooted binary trees with ≤ 8 leaves (from Jarvis, et al. 1983).

Table 5.2 Distribution of the nearest neighbor interchange metric for trees with up to 8 leaves

	Number of leaves				
Distance	4	5	6	7	8
1	1.000	0.2857	0.0577	0.0085	0.0010
2		0.5714	0.1978	0.0395	0.0056
3		0.1429	0.3709	0.1201	0.0224
4			0.3407	0.2528	0.0675
5			0.0330	0.3418	0.1551
6				0.2175	0.2609
7				0.0198	0.2914
8					0.1704
9					0.0253
10					0.0004

For trees with > 8 leaves the distribution can be estimated by computing d_{NNI} for pairs of trees randomly selected from the set of all possible binary trees (Brown and Day, 1984).

Partition metric

Given two trees, T_1 and T_2, the partition metric is the number of clusters found in one or other, but not both trees. This measure has been discussed in detail by Penny and Hendy (1985).

Algorithm

COMPONENT uses Day's (1985) algorithm.

Rooted versus unrooted trees

If the trees being compared are unrooted COMPONENT arbitrarily roots the trees with the first leaf in the profile. The choice of leaf does not affect the result.

Distribution

The table below gives the exact distribution of the probability of observing a given value of the partition metric for unrooted binary trees with ≤ 8 leaves (from Hendy and Penny, 1984).

Table 5.3 Distribution of the partition metric for trees with up to 8 leaves

	Number of leaves				
Distance	4	5	6	7	8
0	0.333	0.0667	0.0095	0.0011	0.0001
2	0.667	0.267	0.0571	0.0085	0.0010
4		0.667	0.237	0.0466	0.0065
6			0.697	0.216	0.0379
8				0.728	0.200
10					0.755

Hendy and Penny (1984) have computed exact values for this measure for trees with up to 16 leaves.

Quartets

A quartet is the smallest possible informative subtree of an unrooted tree, and contains just four leaves. The two possible topologies for a quartet are:

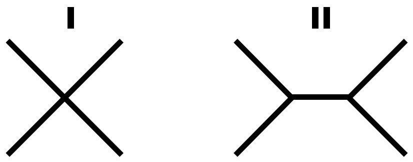


Figure 5.18 The two possible quartet topologies

Type I topologies are only found in nonbinary (i.e., incompletely resolved) trees.

An unrooted tree with *n* leaves contains Q = *n* (*n* −1)(*n* − 2)(*n* −3)/24 quartets. Each quartet (a, b, c, d) will be one of four possible types:

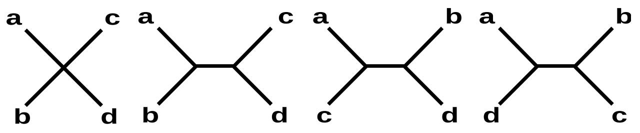


Figure 5.19 The four possible quartets

An unrooted tree can be thought of as a set of quartets. Hence one way to measure the similarity of two unrooted trees, T_1 and T_2, is to compare their quartets (Estabrook, et al., 1985).

Measures

Each pair of quartets from two trees belongs to one of five classes:

s	resolved and identical
d	resolved and different
r_1	resolved in T_1 but not in T_2
r_2	resolved in T_2 but not T_1
u	unresolved in both T_1 and T_2

Note that $Q = s + d + r_1 + r_2 + u$, and that for two binary trees, $r_1 = r_2 = u = 0$.

From these classes of quartets Estabrook et al. (1985) and Day (1986) derived a number of dissimilarity values, including:

Do not **C**onflict (DC)	$= d$
Explicitly **A**gree (EA)	$= d + r_1 + r_2 + u$
Strict **J**oint **A**ssertions (SJA)	$= d / (d + s)$
Symmetric **D**ifference (SD)	$= (2d + r_1 + r_2)/(2d + 2s + r_1 + r_2)$

Algorithm

COMPONENT uses an algorithm based on Douchette (1985) to compute quartet dissimilarity measures.

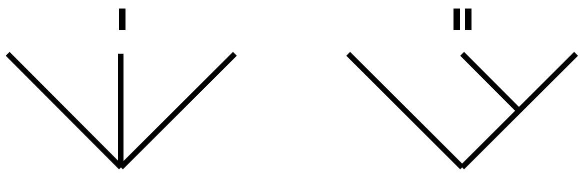
Distribution

Day (1986) has estimated the distribution of various quartet statistics.

Triplets

A triplet is the smallest possible informative subtree of an rooted tree, and is the rooted analogue of a quartet. The two possible topologies for a triplet are:

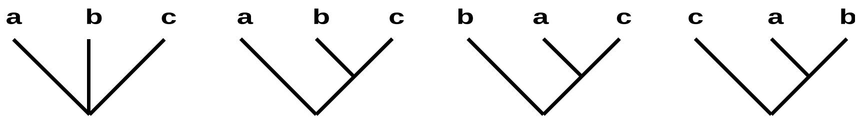
Figure 5.20
The two possible toplogies for a triplet



Type I topologies are only found in nonbinary (i.e., incompletely resolved) trees.

A rooted tree with *n* leaves contains T = *n* (*n* –1)(*n* – 2)/6 triplets. Each triplet (a, b, c) will be one of four possible types:

Figure 5. 21
The four possible triplets



Analogously with quartets, we can use the frequencies of the four possible triplets in two trees as a measure the similarity their similarity.

Measures

See the equivalent section for quartets.

Algorithm

COMPONENT uses an algorithm based on Douchette's (1985) algorithm for quartets.