Chapter 0
Introduction

This chapter provides an overview of COMPONENT's features, a guide to using the rest of the manual, and an explanation of some of the terminology used in the program and its documentation.

What is COMPONENT?

COMPONENT is a tool for comparing trees. While systematists are well served by an array of sophisticated programs for searching for optimal trees and mapping characters onto cladograms (e.g., Farris, 1988; Felsenstein, 1989; Maddison and Maddison, 1992; Swofford, 1990), there are few programs designed to investigate and compare trees themselves. COMPONENT is designed to address this need by providing a range of tree consensus, comparison, and randomisation measures. Possible applications include:

- comparing trees in studies of taxonomic congruence
- comparing host and parasite cladograms
- comparing gene trees with species trees
- computing area cladograms
- finding consensus trees and agreement subtrees
- investigating islands of trees
- analysing the distribution of bootstrap trees
- computing distributions of tree comparison measures
- comparing distributions of tree shapes in real and random trees
- numbering trees for use in simulations
- generating random trees for input into phylogenetic programs as user trees
- converting tree files from one format to another

Overview of features

COMPONENT is a standard Microsoft® Windows™ application with menus, dialog boxes and on-line context sensitive help. The program has built in text editing facilities, so you can create input files and correct mistakes interactively. Trees are displayed graphically, and can be pruned, rerooted, edited, deleted, and compared. Trees can be input from disk files in a variety of formats, or generated within the program. You can output trees to disk files or print them in a variety of styles.

In the next section, some of the key features of the program are highlighted.

Tree editor

COMPONENT includes a simple tree editor similar to that in MacClade. By clicking on the Tools palette the cursor changes to a tool that lets you rearrange branches, collapse clades, and reroot the tree. The editor lets you explore different tree topologies, and can simplify enter large trees described in the literature — simply enter an unresolved tree in your input file then use the editor to create the tree you want.

Support for other programs

COMPONENT uses the NEXUS file format for describing trees, hence it is compatible with PAUP 3.0 and MacClade 3.0. Tree files produced by those programs can be read directly by COMPONENT. To facilitate analysis of trees produced by other programs, COMPONENT can read and write PHYLIP and Hennig86 tree files, as well as some less common formats (CONTREE and FREQPARS). Hence, you can use the program to convert tree files written by one program into the format used by another program.

Output

The results of your analyses are stored in a display buffer which can be saved to disk, printed, or edited. In addition COMPONENT provides WYSIWYG ("what-you-see-is-what-you-get") printing of trees on any Windows supported printer. You can also copy and paste pictures of trees from COMPONENT into other Windows applications, such as word processors, or save the pictures to disk as WMF graphics files.

Consensus trees

Consensus trees are widely used to summarise the agreement among a set of trees. COMPONENT provides the strict, majority rule, semi-strict, Nelson, and Adams methods. The program can output the frequency of all clusters in the trees, as well as a pairwise compatibility matrix of the clusters. The new algorithm of Kubicka et al.'s (1992) for computing agreement subtrees is also included.

Comparing trees

COMPONENT implements a variety of tree comparison measures, including the well known partition metric (Penny and Hendy, 1985) as well as the nearest neighbour interchange metric (Robinson, 1971; Waterman and Smith, 1978) and quartet measures (Estabrook et. al, 1985). Comparisons can be made between two individual trees, a set of trees, or two sets of trees in different input files. These measures can be used to quantify the similarity between trees as part of congruence studies (e.g., Swofford, 1990) and investigate islands of trees (Maddison, 1991; Page, 1993b).

Random trees

Random trees can be generated using a variety of models. These trees can be used as the basis for statistical tests of the similarity between trees. COMPONENT can also generate all possible tree shapes for a specified number of taxa, a useful feature for exploring measures of tree shape and balance.

Statistics

COMPONENT can compute several tree statistics, including a numerical representation of the shape of a tree based on Harding's (1972) notation which can be used to find the frequency of particular tree shapes in sets of trees (Page, 1993c). This is particularly useful in studies of models of phylogenesis (e.g., Savage, 1983; Guyer and Slowinski, 1991). Other statistics include Sourdis' (1985) tree number and Rohlf's (1982) measure of resolution.

Tree mapping

COMPONENT can map one tree onto another, which is the basis of Goodman et al.'s (1979) method of reconciling incongruent gene and species trees, and Nelson and Platnick's (1981) "component analysis." You can create maps between pairs of trees, compute a "reconciled tree," and search for trees with the most parsimonious maps. These techniques can be used to explore conflicts between gene and species trees, host and parasite cladograms, and in cladistic biogeography (Page, 1993a; submitted).

Using this manual

Chapter 1 describes the installation procedure and the basics of using the program. It is here that you will find how to edit files, view output, and set up printers. If you want an overview of all the commands available in COMPONENT turn to the pictorial guide to the menus in Appendix C.

Chapter 2 outlines the basic tree operations such as reading, saving, editing, printing, pruning, and rooting. If you are new to COMPONENT this is a good place to start.

The NEXUS file format used by COMPONENT is described in Chapter 3. The commands recognised by COMPONENT are described in detail, and differences

between COMPONENT and other NEXUS programs are noted. The chapter provides some example input files.

Chapters 4-6 describe how to compute consensus trees, compare trees, and to generate random trees. As well as instructions on how to perform each operation you will find a discussion of the algorithms the program uses and how to interpret output.

Chapter 7 describes tree mapping routines designed for use in studies of gene tree/species tree incongruence, host-parasite cospeciation, and biogeography.

Chapter 8 provides a series of worked examples showing how you can use COMPONENT to perform various analyses. Data files for all these examples are on the distribution disk. This is a good place to learn about COMPONENT 's capabilities.

Appendix A describes the various program formats supported by COMPONENT (other than the NEXUS format for which see Chapter 3). Appendix B contains a technical description of two methods used to order trees. Appendix C is a pictorial guide to all the menu commands. Error messages are listed in Appendix D.

Typographic conventions

Throughout this manual the following typographic conventions are followed.

Bold face	Indicates a menu item or a dialog box control such as a button.
Italics	Indicates a key on the keyboard
`Courier`	This type face is used to indicate either input into the program (such as a command in a file) or program output.
☞	This symbol highlights an important note.
	The mouse symbol indicates actions requiring a mouse.
	The keyboard symbol indicates actions requiring the keyboard.

Contacting the author

If you run into problems or have comments or suggestions you can contact me the following address:

Roderic D. M. Page
Biogeography and Conservation Laboratory
Natural History Museum
Cromwell Road
London SW7 5BD
UNITED KINGDOM

Tel: (071) 938 9168 (International +44 71 938 9168)
Fax: (071) 938 9260 (International +44 71 938 9260)
e-mail R.Page@nhm.ic.ac.uk (Internet)

Technical matters

COMPONENT is written in object-oriented Pascal and was compiled using Borland ® Turbo Pascal® for Windows version 1.5 and Borland® Resource Workshop™ version 1.02. The on-line help file was written using Microsoft Word for Windows™ 2.0c and compiled using the Microsoft Help Compiler 3.00. The manual was drafted in Windows Write, completed using Word for Windows, and printed using a Hewlett Packard Laserjet 4.

Tree terminology

This section describes some of the terminology you will encounter in this manual. Tree terminology varies among authors (Hendy and Penny, 1984; Penny et al., 1992), particularly between the mathematical and biological literature. This section is not exhaustive; some concepts not discussed here are illustrated in the relevant chapters.

Tree

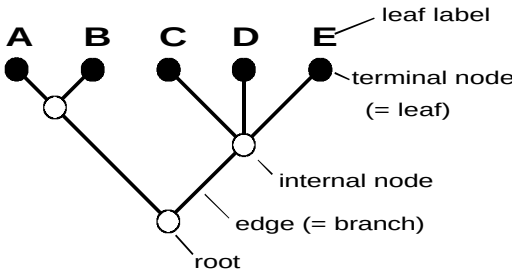


Figure 0.1
A tree

A tree consists of *nodes* connected by *edges* (also called *branches*). *Terminal nodes* (also called *leaves*) are connected to just one other node, that is they have *degree* one. Nodes connected to more than one node (i.e. with degree > 1) are *internal* nodes.

Labelled and unlabelled trees

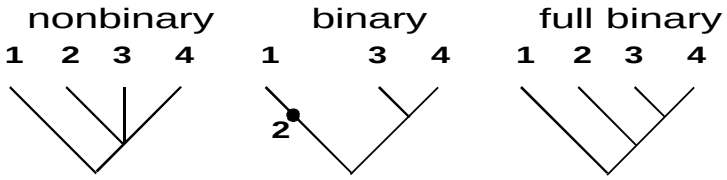
The trees used in phylogenetics and biogeography are typically *terminally labelled*, that is, each terminal node or leaf is associated with some object (such as a taxon or an area) and is labelled with the name of that object. If no node is labelled then the tree is *unlabelled*. Unlabelled trees are sometimes referred to as *shapes* or *topologies* (these terms will be used interchangeably in this manual). Here are the three possible unlabelled rooted binary trees with five leaves (the terms *rooted* and *binary* are explained below):

Harding (1972) developed a notation for tree shapes that allows you to describe the shape of any rooted binary tree. Using his system, the trees above have shapes **5**$_1$, **5**$_2$, and **5**$_3$, respectively. Furnas (1984) described algorithms for computing a unique number to describe each rooted and unrooted tree shape. His algorithms, which for rooted trees are equivalent to Harding's (1972), are implemented in COMPONENT (see Chapter 2, section 2.9, and Appendix B).

Binary tree

A tree is *binary* if none of its internal nodes has degree > 3. A binary tree is *full* if it has at most one internal node of degree two (the root). These terms are illustrated by these three trees:

Figure 0.3
A nonbinary, binary, and full binary tree

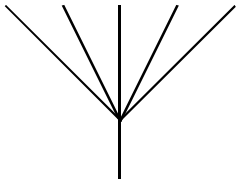


Other terms for full binary trees are *dichotomous, fully resolved*, and *strictly bifurcating*. Some of the tree comparison measures implemented in COMPONENT, such as the nearest neighbour interchange metric, require binary trees (see Chapter 5).

Polytomy

If a tree is not binary then it contains one or more nodes with more than two descendants. Such a node is called a *polytomy*:

Figure 0.4
A polytomy



Polytomies should be interpreted with care because they can be used to indicate either a multiple speciation event ("hard", Maddison 1989) or uncertainty about relationships ("soft"). Some tree comparison measures (e.g., the partition metric) use the first interpretation so that two trees which are consistent with each other, but not identical, are regarded as distinct trees. Other measures (such as those based on quartets and triplets) allow you to distinguish between trees that are consistent, differing only in degree of resolution, and trees that actually contradict each other (see Chapter 5).

Star tree

A *star tree* is a tree containing just one internal node. Synonyms include *bush* and *big-bang tree*.

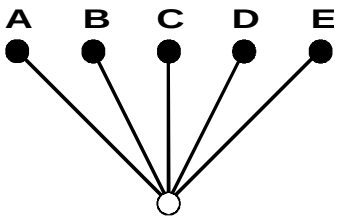


Figure 0.5
A star tree

Balance

The degree to which the internal nodes of a tree split their descendants into clusters of equal size is a measure of the *balance* (Shao and Sokal, 1990) of that tree. For example, tree shape $\mathbf{5}_3$ is more balanced than $\mathbf{5}_1$. Various measure of balance have been proposed (see Shao and Sokal, 1990). At present COMPONENT computes only the position of a rooted (unrooted) binary tree in the LLR (LLC) order of trees (see Appendix B).

Rooted tree

A tree is *rooted* if there is a special node (the *root*) that imparts a direction to the tree. A rooted tree can be also visualised as an unrooted tree with an additional leaf that has been "pulled down" to root the tree:

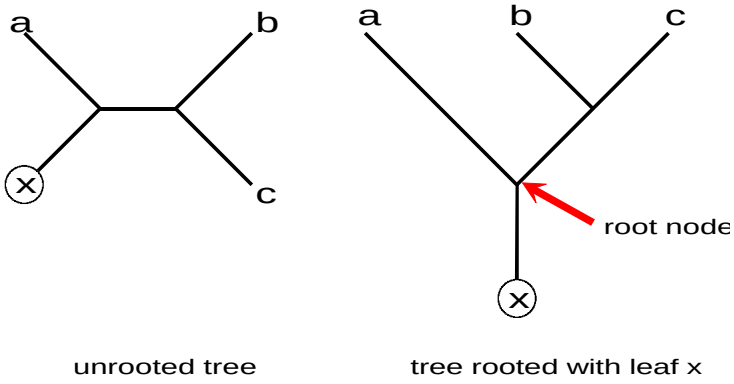


Figure 0.6
An rooted tree (right) can be regarded as an unrooted tree that is rooted by one of its leaves

Some programs (e.g., PAUP 3.0) regard this additional leaf (x in the above example) as the root of the tree. However, COMPONENT regards the basal node of the tree as the root.

Cluster

A *cluster* of a node is the set of all leaves that are descendants of that node (the cluster of a leaf is itself). Clusters correspond to "components", "clades", and "monophyletic taxa." The diagram below shows a tree and its clusters

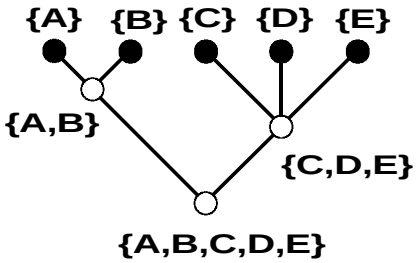


Figure 0.7
A tree and its clusters

If x and y are clusters of a tree then they must be *compatible*, that is x Ç y Î {x, y, Æ}. Hence either x and y are nested one inside the other, for examples {A, B} Ì {A, B, C, D, E}, or x and y are disjoint, for example {A, B} Ç {C, D, E}= Æ.

If a tree is unrooted then the notion of ancestors and descendants becomes inappropriate. Instead of clusters an unrooted tree has *partitions*. Each internal edge of an unrooted tree partitions the leaves of that tree into two sets, being the two subtrees (see below) that would result from deleting that edge. By convention, the smaller of the two subsets can be used to represent the partition.

n-tree

An n-*tree* is a set of subsets of the set S = {1, ..., *n*} that are mutually compatible. These subsets are the clusters of a rooted tree, so that a rooted tree is an *n*-tree. Since the cladistic information of a tree is just its clusters a cladogram is also an *n*-tree.

Subtree

A *subtree* of a tree is any tree formed by pruning one or more leaves from that tree (and removing any internal nodes that have degree two as a result of pruning a leaf). For example, given this tree:

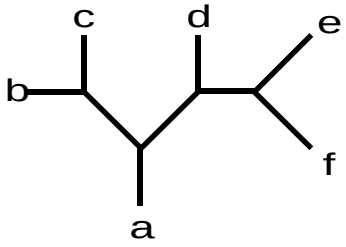


Figure 0.8
A tree

the subtree obtained by pruning the leaf d is:

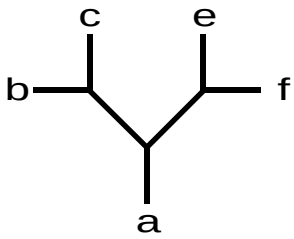


Figure 0.9
The subtree of the tree in Figure 0.8 obtained by pruning leaf d

Given two trees COMPONENT can compute the largest subtree in common to the two trees (see Chapter 5).

Ordered tree

In an *ordered tree* the nodes are assigned a fixed order (such as their left-right position on a page). Ordered trees may have the same topology and labels but are still considered distinct on the basis of this order. For example, these two trees are different ordered trees but the same unordered trees:

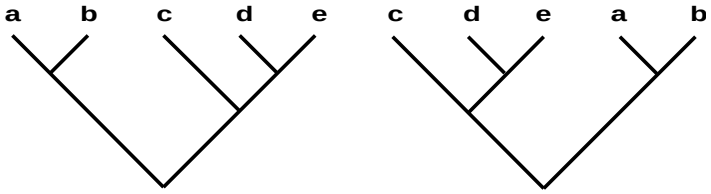


Figure 0.10
Two trees that are distinct ordered trees but the same unordered trees

Ordering rarely has biological meaning and does not affect any of COMPONENT's computations. Its principal use is to improve the appearance of a tree for output (using the **Trees Order** command), or in algorithms for generating random trees (see Chapter 6 and Appendix B) and computing a tree's shape (Chapter 2).

Random tree

A *random tree* is a tree drawn at random from a set of trees. In this manual "at random" means drawn from a *uniform* distribution in which all trees of a particular kind are equally represented and hence have an equal probability of being sampled. The various uniform tree distributions from which COMPONENT can sample are described in Chapter 6.

Dendrogram

A *dendrogram* is a rooted tree with the internal nodes ranked on the basis of their relative distance to the root (e.g., Lapointe and Legendre, 1991). For example:

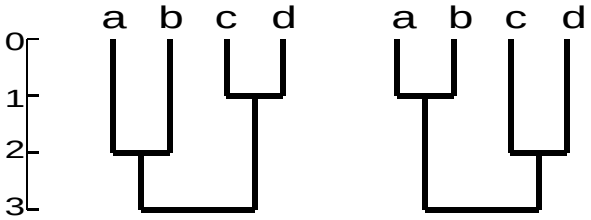


Figure 0.11
A dendrogram

These two dendrograms both have the topology ((a,b),(c,d)) and are identical as cladograms but because the ranks of their internal nodes differ they are distinct dendrograms. The well-known Markovian model for generating random trees corresponds to uniform sampling from the set of labelled binary dendrograms (Page, 1991).

Profile

A *profile* is a set of trees with the same labels. For example, here is a profile of four trees:

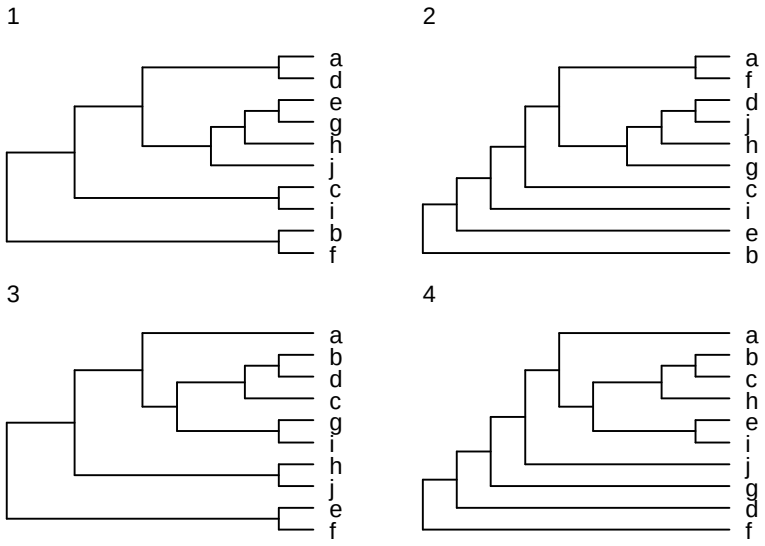


Figure 0.12
A profile of four trees

This term is commonly used in the consensus tree literature, and is often used in this manual. The set of trees in a TREES block of an input file is a profile.

Consensus tree

A *consensus tree* summarises the agreement between two or more trees. It can be regarded as a function on a profile of trees that maps those trees onto a single tree — the consensus tree. The consensus tree methods implemented in COMPONENT are described in Chapter 4.