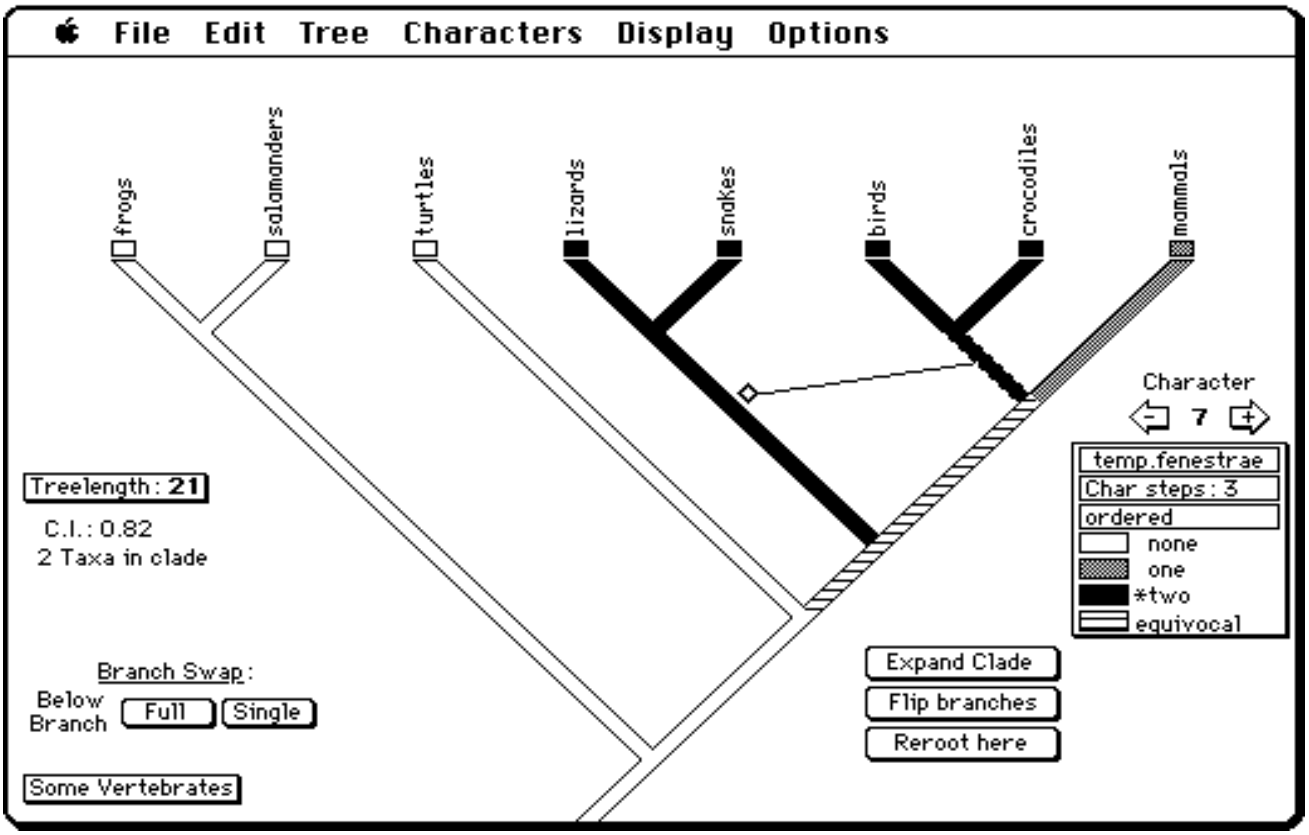
MacClade

version 2.1

Wayne Maddison and David Maddison

An interactive, graphic program for analyzing phylogenies and studying character evolution.

For the Apple Macintosh™ Computer



MacClade

version 2.1
for the Apple Macintosh ™

Copyright: MacClade version 2.1 and the MacClade manual are Copyright 1987 by Wayne Maddison and David Maddison, all rights reserved except as follows. You may copy the program and documentation, as long as (1) you do not sell the copies; (2) unaltered versions of files "MacClade" and "MacClade.help" are included on the disk; (3) the documentation is unaltered.

Distribution: You can become a registered user by writing to us. Registered users will receive notification of major new versions and any substantial, non-cosmetic bugs. The program itself is free; however, if you wish to obtain a copy from us, we would ask that you reimburse us for the cost of a disk, photocopying the manual, and mailing. (At present this amounts to about $6.00 to North America, $8 overseas; you can reimburse us by sending 3 or 4 Mac disks instead of money.) If you can obtain your copy of the program from someone else, and photocopy the manual yourself, please do, as it will save us time.

Some necessary stuff:

LIMITED WARRANTY. Even though we (Wayne Maddison and David Maddison) have tested the software and reviewed the documentation, we make no warranty, either express or implied, with respect to this software, its quality, performance, merchantability, or fitness for a particular purpose. As a result, this software is sold "as is", and you, the user, are assuming the entire risk as to its quality and performance. Should the programs prove defective, the user assumes the entire risk, including any and all necessary servicing, repair or correction, as well as any and all incidental or consequential damages.

System, Finder, and Imagewriter Driver are copyrighted programs of Apple Computer, Inc. licensed to Wayne Maddison to distribute for use only in combination with MacClade. Apple Software shall not be copied onto another diskette (except for archive purposes) or into memory unless as part of the execution of MacClade. When MacClade has completed execution Apple Software shall not be used by any other program.

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

Macintosh is a trademark licensed to Apple Computer, Inc. Apple is a registered trademark and MacWrite, MacPaint, Macintosh Pascal, Edit, and Switcher are trademarks of Apple Computer, Inc. TML Pascal is a trademark of TML Systems. Microsoft, Multiplan, and Excel are trademarks of Microsoft Corporation. Camera Desk Accessory is copyright by Keith A. Esau.

MacClade

for the Apple Macintosh

Version 2.1
March 1987
manual slightly revised January 1989

Wayne P. Maddison and David R. Maddison

Department of Zoology
University of California
Berkeley, CA 94720
(415) 643-6299

Museum of Comparative Zoology
Harvard University
Cambridge MA 02138
(617) 495-4012

First version (June 1986),
tree manipulation, reconstruction of character evolution,
and other stuff

by Wayne Maddison

Text-editor, help, printing, scrolling,
Standard Macintosh interface (menus, dialog boxes, etc.)
and other stuff

by David Maddison

CONTENTS

1. INTRODUCTION 5

2. USING MACCLADE 7

3. TREES AND TREE MANIPULATION 11

4. TRACING CHARACTER EVOLUTION 18

5. DIAGNOSTICS OF CHARACTER EVOLUTION 22

6. ASSUMPTIONS ABOUT CHARACTER EVOLUTION 27

7. FORMAT OF DATA FILES 34

8. EDITING DATA FILES 39

9. PRINTING 41

10. USES OF MACCLADE 43

11. PAST, PRESENT AND FUTURE OF MACCLADE

(including what to do about bugs in this version) 46

REFERENCES 48

APPENDICES:

1. ALGORITHMS FOR TRACING CHARACTER EVOLUTION 49

2. USING A MACINTOSH 54

3. MACCLADE DATA FILES CREATED WITH OTHER PROGRAMS 56

1. INTRODUCTION

MacClade was written to allow systematists and other evolutionary biologists to analyze character evolution and phylogenetic hypotheses in an interactive environment. It includes a number of new algorithms allowing a flexibility of assumptions not available before in phylogenetics programs. Emphasis was placed on graphic display and ease of use. MacClade can therefore serve not only as a research tool, but as a teaching device for courses introducing phylogenetic principles.

MacClade has a number of features for a flexible and interactive analysis of character evolution. MacClade displays an input cladogram and reconstructs the evolution of characters. The evolution of a character is displayed by "painting" the cladogram with different patterns. These patterns indicate all of the most parsimonious reconstructions of ancestral states according to the cladogram, but the user can examine other hypotheses of character evolution as well by fixing the state at a branch to a state of the user's choice. Besides allowing the user to explore alternative hypotheses under the same cladogram, MacClade allows quick comparisons of character evolution according to different cladograms, because the user can easily pick up and move branches on-screen. MacClade contains among the most varied and complete algorithms of any computer program for reconstructing the evolution of characters. These features illustrate one of the motivations behind MacClade: the conviction that if evolutionary biologists are to test their hypotheses of evolutionary processes against phylogenetic patterns, the techniques for analyzing character evolution must be improved and made easier.

The ability to pick up and move branches on screen is one of several features that can be used to explore different cladograms under varying assumptions. Clades can be rerooted at branches selected on the screen. Character weights and assumptions like irreversibility can be changed easily. Terminal taxa can be moved down into ancestral position, a feature useful to paleontologists wishing to examine the consequences of supposing a fossil is the actual ancestor. With each change, the new cladogram length, character evolution and other diagnostic information are shown. This quick feedback gives a good feel for how the alternative hypotheses relate to the data: which characters support which clades, why some clades are unstable with slight changes in the data or assumptions, and so on. A user familiar with the organisms studied stands a good chance of finding parsimonious cladograms, though MacClade is not designed as a powerful cladogram-finding program. MacClade's interactiveness encourages the user's exploration of the data set, and the insights gained contribute to an intelligent reconstruction of phylogeny.

Using this manual

One of the joys (and problems) of the Macintosh is that it is so easy to operate that one can quickly learn how to use a program without reading the manual. A lot of people relegate their manuals to a life on the shelf collecting dust. We suggest that even those sorts at least skim over the manual, so that they learn some of the assumptions employed by MacClade, and learn some of the features and short cuts that may not be obvious from the program itself. If you are new to the Macintosh, or need a short review about how to operate one, it would be a good idea to read Appendix 2, "Using a Macintosh", before you start the practice session. The manual includes the following sections:

Chapter 2, "Using MacClade", gives general information about MacClade, including an overview of starting, using, and quitting from the program.

Chapters 3 through 6 cover phylogenetic analysis and examination of character evolution. Technical details about some of the methods used by MacClade are outlined in Appendix 1.

Chapters 7 and 8 describe the format of data files, and editing of them using MacClade's text-editor. For those who wish to use other programs to create MacClade data files, the techniques are outlined in Appendix 3.

Chapter 9 describes the printing options of MacClade.
Chapter 10 gives some information on possible uses of MacClade in systematic research and teaching.
Chapter 11 recounts the history of MacClade features, good and bad. If your machine crashes when using MacClade, you should look in this section for advice.

Throughout the rest of this manual, and on the MacClade screen, we use "tree" in place of "cladogram". This is done for the sake of compactness, not because we feel they should be called trees instead of cladograms. Indeed, unless the Make Ancestor function has been employed, MacClade's trees would be better considered as cladograms according to recent definitions (Eldredge & Cracraft, 1980), though these definitions have been disputed (Hendy & Penny, 1984).

MacClade Specifications

The maximum number of taxa is 80; the maximum number of characters is 200. Characters can be unordered, ordered, irreversible, Dollo, or of a user-defined type. Character states integral (0-9); characters can be integrally weighted. Missing data and polymorphic terminal taxa are allowed. In combination with MacPaint and the Laserwriter printer MacClade produces publication quality cladograms.

Hardware Specifications

MacClade requires a Macintosh, Macintosh Plus, XL, SE or II with 512K or more memory. On the Mac II there are two minor problems known. The first is that the larger screen is not fully used; the second is that the number of taxa in a clade is not properly displayed on the left under the treelength box. Instead of listing the number of taxa, the space is left blank.

Distribution of MacClade

You can obtain MacClade from us or from someone else (see distribution notice on inside of front cover for details). We encourage you to obtain your copy from someone else, so that we can spend time improving MacClade rather than packing envelopes. But if you do get it from someone else, you may wish to write us so that we can put you on the mailing list for bug and update notices.

Disk contents

Included on your MacClade disk should be the following:

1. MacClade
2. MacClade.help
3. Example data files (including Vertebrates)
4. System Folder with System, Finder, and Imagewriter files.

If you are missing some of these files, then contact us (or the person you got MacClade from).

Program source code

MacClade is written in Pascal. Initial development used the Macintosh Pascal interpreter; the current version was compiled using the TML Systems TML Pascal compiler. Source code is available on disk from us.

How to cite MacClade

We hope to have a description of MacClade published soon. When this comes out you can cite this in bibliographies of papers using MacClade. Until then you can cite MacClade in the text (e.g. "Analysis of character evolution was done using the computer program MacClade (version 2.1, written by Wayne Maddison and David Maddison, Harvard University)") and/or in the Literature Cited (e.g. "Maddison, W.P. and D.R. Maddison, 1987. MacClade, version 2.1. A phylogenetics

computer program distributed by the authors."). We request that you send one of us a reprint of any paper that uses MacClade.

Acknowledgments

A number of people around the MCZ, including Bob O'Hara, Chuck Crumly, Hans Sues and Neil Shubin, helped with comments on the program or the Vertebrates data set (do not hold them responsible for any peculiarities in the data set). Many other people, from the MCZ and elsewhere, provided valuable suggestions and criticisms; we thank them all. Herbert Levi and NSF provided use of a Macintosh for part of the development of MacClade; Harvard University provided the use of facilities. We also thank David Swofford and Joe Felsenstein for fruitful discussion.

2. USING MACCLADE

This section provides a general overview of the operation of MacClade.

We suggest that before you use MacClade on any of your own data files, you play with the example data files on the disk. You might start MacClade, open an example data file such as "Vertebrates", and just start mucking around with trees, pressing buttons and selecting menu items to see what happens.

Data Files

MacClade data files can be created and edited with MacClade, or programs other than MacClade. In general, we will presume you are using MacClade to make your data files. If you wish to use other text editors like MacWrite or Microsoft Word or spreadsheet programs like Excel to create MacClade data files, please see the instructions in Appendix 3, "MacClade Data Files Created by Other Programs".

Starting MacClade

The method used to start the program from the Finder will depend upon whether you wish to (1) create a new data file to be analyzed, or (2) examine an already existing data file created earlier by MacClade. The two methods are outlined below:

•To start MacClade and *create a new MacClade data file*, double-click on the MacClade icon:

and then choose "New" from the File menu. You will be presented with an empty window into which you can type and edit text (see "Editing data files" for instructions on use of the editor). You should save the file to disk (choose "Save as..." from the File menu) before continuing.

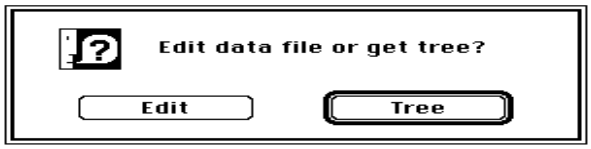
•To start MacClade and *open an already existing MacClade data file,* double-click on the icon of the data file:

This will get you directly into MacClade, using that data file.

The First Choice: Edit the data file or examine a tree

MacClade has two primary "windows". One is the tree window, in which trees are displayed and manipulated. The other is the edit window, in which you can edit the data file.

If you have started MacClade and opened an existing file, you will be presented with the following box:



You now have to choose which of the two windows will be displayed.

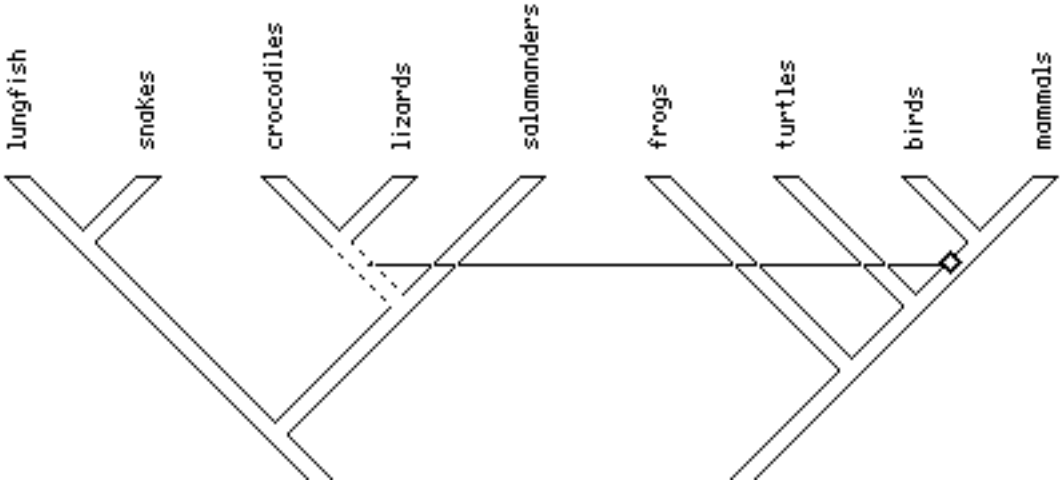
If you click on the "Edit" button, the edit window will come on the screen, and you can proceed to edit the data. (Note that if you just created a new MacClade file by choosing "New File" from the File menu, you will not be presented with the choice, as the first thing you must do is type in the data file.) Once you have typed in the data file or your changes to it, and saved your changes, you can then proceed to the tree window by choosing "Get Tree" from the "Tree" menu.

If you click on the "Tree" button, the tree window will come on the screen. You may have to make some choices before the window will appear. If there are trees stored in the file, MacClade will present a dialog box and allow you to choose the tree you wish to examine. If there are no trees stored in the data file, you will be able to choose only the "default ladder" (which is a "Hennigian Comb").

Note that you can call up the text editor and edit the data file at just about any time while using MacClade. You can quickly change back and forth between analyzing the trees and editing the data file.

The Analysis: Examining Trees and Character Evolution

The tree displayed on the screen may be rearranged. Rearranging the tree follows an intuitive method of grabbing with the mouse, dragging, and dropping. For example, the figure below shows the crocodiles + lizard group being moved over to be the sister group of mammals + birds.



Tree rearrangements are fully described in Chapter 3, "Trees and Tree Manipulation"

You can examine evolution of a character, by asking MacClade to trace its states on the tree. To do this, click on the trace button:

A dialog box will appear, allowing you to choose the character to be traced. Once you have chosen the character, its states will be painted on the tree branches, as shown on the cover of this manual. Further details on studying character evolution are given in Chapters 4 through 6.

Closing a data file

Choose "Close File" from the File menu. If the present tree, set of weightings, or transformation type assignments are not saved in the data file, MacClade will give a warning and allows you to save them before closing, or to close without saving them, or to simply cancel the Close File command.

Examining a second data file

If you are examining a data file with MacClade, you can examine another data file by closing the first data file, then choosing "Open File" from the File menu, and selecting and opening the second data file.

Quitting from MacClade

Choose "Quit" from the File menu. If the present tree, set of weightings, or transformation type sets are not saved in the data file, MacClade will give a warning and allows you to save them before quitting, or to quit without saving them, or to simply cancel the Quit command.

Memory Warning

MacClade will warn you if your memory is getting low by drawing a small box in the lower left hand part of the screen, near the title. The memory available is shown in the box. If such a warning appears, you may wish to save the current tree, weight set, or transformation type set to the data file in case MacClade crashes due to lack of memory. Lack of memory should only be a problem on a 512K machine with the largest data sets (80 taxa by 200 character s), and then only if you have been examining several data files and doing lots of editing, printing, using the Help facility, and so on. One way to clean up memory is to Quit down to the Finder then restart MacClade before examining an extra large data set.

Using MacClade with Switcher

Switcher is a program that allows you to switch between two or more programs rapidly, without having to quit from an application before moving on to another. With it you can, for example, be using MacClade, dump the picture of a tree to a MacPaint file, and then with one flick of the mouse be in MacPaint editing that image for Laserwriter output. Another possible combination is to be using a word processor, MacPaint, and MacClade all at once under Switcher, thus allowing you to quickly switch between analyzing your data, producing figures, and editing your paper.

The number and kinds of programs that can be run at once under Switcher will depend upon how much memory your Macintosh has and how much memory each of the programs require. When using Switcher, you can allocate how much memory each program is allowed to use. MacClade itself will run (using the smallest data sets) with only 256K of memory, although this is dangerous; for large data sets, 512K is necessary. We recommend allotting *at least* 300K to MacClade, even for the smallest data sets.

Switcher can be obtained from your local Apple dealer, or from your local users' group or electronic bulletin board. It's cheap (free or something around $25, depending on where you get it), and well worth it.

Getting help

MacClade has a help facility: choose "Help..." from the apple menu, and you will be presented with a window giving some basic information about MacClade. The information for this help facility is included in the file "MacClade.help", which should be included on your disk. If MacClade can't find this file (perhaps because it's off in some folder or on another disk) when you ask for help, MacClade will allow you to search for the file using the standard file-opening dialog box. Once you have found the help file, open it; MacClade will read it in and present you with the information.

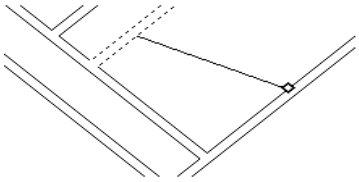
You can customize the help file yourself (it can be edited with any text-editor, including MacClade's own). But if you do, we ask that you keep a copy of the unaltered help file around, because if you wish to pass on a copy of MacClade to someone else, you must pass on an unaltered copy of the help file. Also, note that if you change the name of the file from "MacClade.help", MacClade will not recognize it!

3. TREES AND TREE MANIPULATION

MacClade will display a tree on the screen. MacClade begins with either the default tree or a tree read in from the data file. You can manipulate this tree on-screen by using the mouse to move branches around, reroot clades, etc. Most of the manipulation is left up to you. You can also let MacClade itself manipulate the tree, by asking it to search for a shorter tree. Searching for short trees is not MacClade's forte, however, and it should not be relied upon as a replacement for programs such as Swofford's PAUP, Farris & Mickevich's PHYSYS, Felsenstein's PHYLIP, etc.

Moving branches and clades around

A branch (and along with it any clade above it) can be picked up and moved to another part of the tree. Select a branch to be moved by clicking on it, and hold the mouse button down; the cursor will change from an arrow to a small diamond. Drag the diamond to the place you want the branch dropped, and drop it by releasing the mouse button:

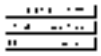


In some circumstances MacClade will refuse to move a branch because it would be inappropriate (for instance if you try to move a branch onto one of its descendants). After a branch move, MacClade will automatically recalculate treelength, consistency index, etc., for the new tree. If any branches involved in the move have their states fixed (see chapter 4), their states will be unfixed by the move and a warning given.

Other tree-manipulation commands

When you select a branch, various command buttons will appear on the screen. If you drag the diamond over to one of those command buttons, and drop it in the button by letting go of the mouse button, that command will be enacted on the selected branch.

For example, to reroot the tree at a branch, select that branch, and drop the diamond cursor onto the "Reroot Here" button.



The clade will be rerooted at that branch.

Other commands are effected by dropping the diamond in the appropriate command button. (There are a very few exceptions to this drop-in-button method of giving branch-manipulation commands; the exceptions will be outlined where appropriate.)

Undoing tree manipulations

After you make a change in tree structure, MacClade remembers the previous tree, so that you can undo the change and recover the old tree. To recover the old tree, choose the "Undo" command from the Tree menu. This does not work for all tree manipulations (for example, the cosmetic manipulations, described below, cannot be undone by the Undo

command). Only the following types of moves can be undone: branch move, rerooting, branch swapping, or getting a tree from the data file. The "Undo" command will reflect which type of move is to be undone; the menu item will read "Undo Branch Move", "Undo Rerooting", "Undo Branch Swap", or "Undo Get Tree". Calling Undo again will get back the tree you just undid.

Rerooting clades

To reroot a clade, select a branch and drop on "reroot" button:

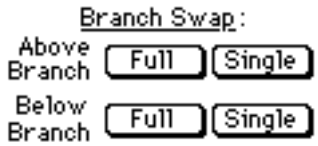
or hold down R key and click circle of reroot cursor

on branch. If the whole tree is on the screen, then the whole tree is rerooted at the branch selected. If one clade has been expanded to fill the screen, then rerooting only reroots the clade shown; the rest of the tree below the clade remains untouched. This can be useful to try different rootings of the ingroup. To do this, expand the ingroup and reroot at various branches. Rerooting a clade may unfix fixed states at nodes and will make any taxa placed as ancestral terminal again (MacClade will give warnings if these occur).

Note that rerooting the entire tree will not change the treelength unless some characters are irreversible or of a user-defined type.

Searching for short trees using local branch swapping

Select branch and drop on one of the branch swap buttons on the left hand side:



Dropping a branch on one of the "Full" or "Single" buttons on the " Above Branch" line performs local branch swapping in the clade above the branch selected (to select the whole tree, grab the root). The part of the tree below the clade is left untouched. The branch swapping algorithm visits each branch in the clade, and sees if a shorter tree would be obtained by rearranging relationships between its sister branch and its two descendant branches. The Single swap finds the single swap throughout the clade that shortens the tree the most. The Full swap does this repeatedly until it can no longer find a local rearrangement that improves length. To help you track where the swapping is taking place, each time a branch is swapped, the branch moved is shaded black (or white if already black) (branch shades are changed only if branches are of normal width). In each case the black branch has been moved onto its ancestor's sister branch. During Full swap, to temporarily suspend swapping place the mouse cursor in the Suspend button (you don't have to press the mouse button down); to abort swapping and show the tree obtained so far by swapping, put the cursor in the Abort button.

Single swap and Full swap differ in a respect other than the number of swaps they will perform. If no rearrangements will improve length, and yet there is a rearrangement which will keep the same length, Full swap will make no change in the tree but Single swap **will** make that equally parsimonious rearrangement. Thus you can use Single swap to search for equally parsimonious trees.

The Above Branch buttons are visible only when the branch selected is suitably far down from the terminal taxa.

Dropping a branch on one of the "Full" or "Single" buttons on the " Below Branch" line performs local branch swapping in the region of the tree below the branch selected, while the clade above the branch is left untouched. The Below Branch buttons are visible only when the branch selected is suitably far up from the root.

While this algorithm may help to fine-tune a tree, do not rely on it to find the shortest tree! Once we had a data matrix whose shortest tree was 36 steps, yet the local branch swapping could find no tree shorter than 72 steps, because the input tree required more than just local rearrangements.

If you are trying to find short trees, one strategy is to alternate between Single and Full swapping (since Single can find equally parsimonious alternatives which may allow you to escape from a local optimum). Note that MacClade ignores characters of user-defined and Dollo types during branch swapping; thus, you may actually end up with a longer tree after branch swapping than you had before. Note also that because the swapping algorithms visit the branches on the left side of the tree first, MacClade will preferentially make swaps on the left side.

Making a terminal taxon ancestral

To push a terminal taxon downward to be an ancestor, select the branch of that taxon and drop on "Make Ancestor" button:

(This button is visible only when a terminal taxon is selected, and only if there are no characters of user-defined types).

This command forces the selected terminal taxon to become an ancestor. The taxon is put in the place of the branch that used to be its immediate ancestor. The name of the taxon remains on top, but instead of a full branch the taxon has only a dashed line pointing to its new position as ancestor. All of the character states of the taxon are forced down onto this ancestral branch. This is useful for paleontologists, because they can see the number of extra steps in character evolution required by supposing a fossil is an ancestor. It is also useful for neontologists to determine if a species might be paraphyletic according to the tree: if the tree length goes up when it is made an ancestor, then the species has autapomorphies. Show Changes (see Chapter 5, "Diagnostics of Character Evolution") can also give a summary of autapomorphies.

Even though a taxon made ancestral has only a line coming down from the taxon's name instead of a full terminal branch, this line can still be grabbed with the mouse, though the only thing that can be done with it is to drop it onto the Make Terminal button.

A taxon fixed as ancestor remains so fixed even when another character is traced. Ancestors are automatically made terminal when branch swapping is invoked or during rerooting and certain branch moves, or if you start the text editor. The fixing as ancestor is not stored when the tree is saved to the file, nor is it remembered on Undo Get Tree.

Warning: If a terminal taxon polymorphic for some character is forced down as an ancestor, the number of steps required is likely to be underestimated. A warning to this effect is given by MacClade. For example, when a terminal taxon polymorphic for 0,1 is forced down as ancestor into a region of the tree with all state 0 surrounding, an extra step should be counted (gain then loss of 1) but isn't. The shading of branches indicating character evolution should be correct, however,

except for Dollo characters and perhaps irreversible characters, for which the shading make be slightly incorrect. Note also that for Dollo and irreversible characters you can force the assumptions of the methods to be violated (for instance if you make a taxon with state 1 ancestral to a clade with state 0 for an irreversible character). MacClade will not give you a warning if you do this.

Making a taxon terminal again

To return a taxon to non-ancestral status, select branch and drop on "Make Terminal" button:

To return all taxa to non-ancestral status, select All Terminal in Tree menu.

COSMETIC CHANGES TO TREE

The next five features allow one to change the appearance of a tree on the screen, but they do not affect the cladistic structure. Of course, these do not change treelength or character evolution reconstruction. They are features for aesthetic purposes. Cosmetic changes can't be undone using the Undo command.

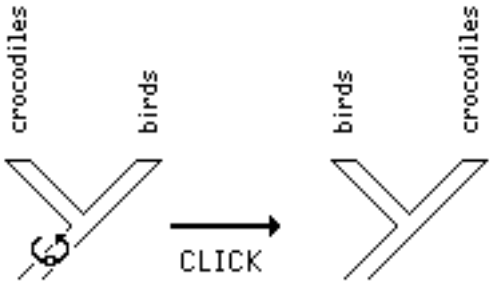
Flipping Branches

There may be times when you wish to change the order of sister groups on the screen. You can accomplish this by invoking the Flip Branches command.

To flip the basal branches in a clade, select the branch below the clade and drop on "Flip Branches" button:

or, hold down F (flip) key and click base of flip cursor:

on branch below clade. For example, to reverse the position of crocodiles and birds in the following example, hold down the F key and click on the branch below the clade:



Ladderizing clades

In the course of moving branches around, you might end up with a jagged clade, looking rather like a monkey-puzzle tree:

If you wish to tidy the clade up a bit, by reducing the jagged aspect, you could use the Flip Branches function to reorganize the clade. But this can be very time consuming. A quicker method is to use the ladderize function, which automatically rotates branches to produce the arrangement that is maximally comb-like. To ladderize a clade, hold down L key and click the central dot of ladderize cursor:

on the branch below the clade. Ladderizing the above jagged clade would yield the following arrangement:

To alter the direction of the ladderization, choose "Ladderize options..." under the Options menu; this will bring up a dialog box that will allow you to choose right-directed ladders or left-directed ladders (default is right directed ladder, as shown above, with the backbone of the ladder directed toward the right).

Expanding clades

You can expand one clade to fill the entire screen. To do this, select the branch just below the clade and drop on "Expand Clade" button:

This command is available only when a non-terminal branch is selected. Expanding a clade is especially useful when there are many taxa, and the branch density at normal viewing is too high.

SHORT CUT: double-clicking on a branch will expand the clade above that branch.

Shrinking Clades

To shrink the tree back to its normal size, choose Shrink from the Tree menu.

SHORT CUT: double-clicking on the root of the expanded clade shown on the screen will shrink the tree.

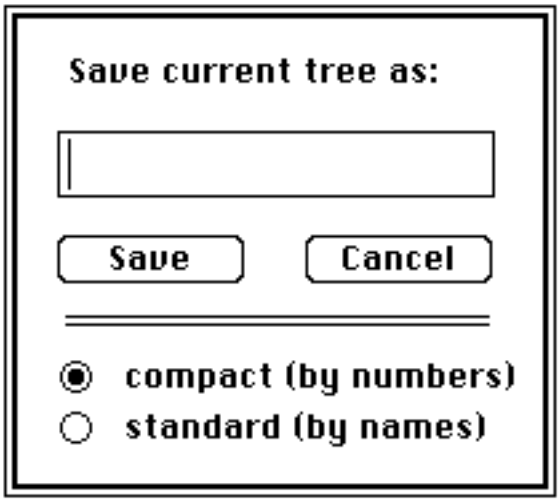
Changing thickness of branches

MacClade offers three different branch thicknesses: normal, medium, thin. The thickness desired can be chosen by making the appropriate selection in the Options menu. Note that medium and thin branches are always solid black: you can not trace a character on a thin or medium branch (if you ask MacClade to trace a character on a thinly-branched tree, MacClade will automatically switch to normal branch thickness).

This option is available mainly for output to MacPaint files or for printing; it can also come in handy when you have more than 60 taxa on the screen and you are having trouble distinguishing branches from spaces in between branches.

Saving a tree to the data file

You may save the current tree to the data file by selecting "Save Tree..." in Tree menu; a dialog box will appear:



Type in the name you wish to give the tree (you can leave the tree unnamed if you wish), and click on the save button. Note that you can save tree data in one of two forms: compact or standard. "Compact (by numbers)" means that in the tree description will be stored in the data file in a compact, MacClade-specific format. The taxa will be referred to by their positions in the data matrix:

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

"Standard (by names)" means that the taxa will be referred to by their names:

(rayfinned_fish,(((lungfish,snakes),((crocodiles,lizards),salamanders)),
(frogs,(turtles,(birds,mammals)))));

Note that taxon names must be all different for this format to work. In general the Standard format is safer, more forgiving and more portable, but it takes up a lot more room. This is explained in more detail below under "Getting a tree...". If you are going to save a lot of trees for a stable data matrix, use the Compact format; if you are going to be adding and subtracting taxa from the data matrix or moving the trees to and from other programs like PAUP and PHYLIP, then use the Standard format. Standard tree descriptions follow a format that is a subset of the standard format agreed to at an informal meeting of a number of phylogenetics programmers at the 1986 meetings of the Society of the Study of Evolution; they can be read by PAUP and PHYLIP.

Getting a tree from the data file

To examine a tree saved in the data file, choose "Get Tree" from the Tree menu. You will be presented with a dialog box listing the trees stored in the file. Select the tree you wish to examine by clicking on its name, then clicking on the Choose button (or just double-clicking on the name). The tree will be read in from the data file and displayed. The name of the tree will be shown on the screen just above the tree length, and will remain there until the tree is changed.

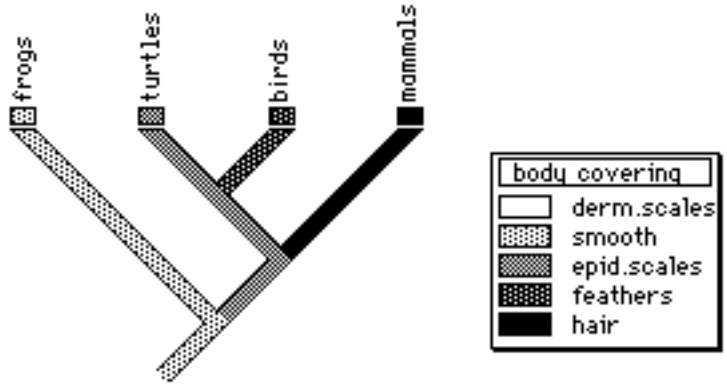
MacClade reads a tree description in the Compact (by numbers) form described above by interpreting the numbers as referring to taxa by their position in the data matrix. Thus in the example above, MacClade will place the first taxon in the matrix as the sister group to the rest, the 8th and 9th taxa will be sister groups, and so on. Thus, after saving such a description, if you rearrange the order of taxa in the data matrix, or delete or add taxa from the middle of the matrix, the description may yield a tree with the taxa in a completely unintended arrangement. For this reason it is better to save in the Standard form if you are going to rearrange taxa in the matrix. However, the Compact form is safe if the only changes are adding and deleting taxa from the end of the matrix. MacClade will then build the trees in a largely correct way (since the positions of most taxa in the matrix are unchanged) and add or subtract taxa as appropriate.

MacClade reads a Standard (by name) tree description by placing taxa according to their names. Thus in the example above, MacClade will place crocodiles and lizards as sister groups regardless of their position in the data matrix. The tree description to be read in from the data file need not include the same taxa or the same number of taxa as in the data matrix, and the ordering of taxa in the data matrix may be changed. If the tree specification has taxa not included in the data matrix, MacClade will ignore these taxa; if the data matrix has taxa not included in the tree description, MacClade will simply attach these onto the tree at the base in ladder-like fashion. MacClade gives a warning if taxa have been added or subtracted.

Because MacClade can read standard format tree descriptions, tree descriptions created by PAUP or PHYLIP can be cut and pasted into your MacClade data file for use by MacClade. When you do this, remember to add a line for the name of the tree (preceded by a '#') immediately preceding the line(s) with the tree description.

4. TRACING CHARACTER EVOLUTION

MacClade will display the evolution of any character in the data matrix by graphically tracing (painting) onto the tree the inferred states of ancestors.



Tracing a character on the tree

To bring forth the dialog box that allows you to choose a character to trace, select Trace Character or Other Character in Characters menu; **or** click on trace button:

The tracing displays the most parsimonious (simplest) reconstructions of evolution of the selected character. The tree branches are shaded to indicate what states are most parsimoniously placed on them. A box appears for each terminal taxon to show the state(s) observed in it. If the state was unknown for a taxon (coded in the data matrix as ?), no box appears. A legend is provided on the right of the screen showing what shade represents what state.

> *SHORT CUTS: There are several ways to trace a character without going through the dialog box. (1) If you hold down the Option key while you click on the trace button, a character will be traced (this will trace either character 1, or, if another character has been previously traced for this data file in this run of MacClade, that other character will be traced). (2) If character data boxes are visible, then double-click on the line of boxes of character to be traced. (3) If the diagnostics table is being shown, then double-click on the left side (in the character name or character number area) of the line of the character to be traced.*

Explanation of legend

When a character is traced, a legend appears at right, looking like this:

If the character has been given a name in the data file (in this example, "appendages"), that name will appear in a box at the top of the character legend.

"Char steps" shows the number of evolutionary steps required in the character by the tree. The number shown is simply the number of steps; it is not multiplied by the character weight. Thus if there are characters weighted differently from 1, when you add all the character steps you may not get the tree length (see Treelength, below). A '+' appears beside the number of steps when the character is of user-defined type and has a terminal taxon polymorphic for more than two states, signifying that the number of steps is underestimated (see Appendix 1).

The box below the character steps box states the assumption about character transformation (that is, whether the character is ordered, irreversible, and so on (see Chapter 6, "Assumptions About Character Evolution")).

Below these boxes is a series of smaller, shaded boxes, which show the patterns used on the tree. State 0 is always shown at the top of the list, with state 1 below it, and so on. If you have named the states in the data file, then the names of the states will appear beside the appropriate boxes; if not, only 0, 1, 2, etc. will appear.

For some characters, you may see one or two additional shadings, indicating the "polymorphic" or "equivocal" patterns. If a terminal taxon is polymorphic (possesses more than one state), the branch is shaded by the "polymorphic" pattern. If the mouse is clicked on a branch with a polymorphic pattern, asterisks appear in the right hand legend to indicate what states were observed in the taxon. If more than one state can be equally parsimoniously placed on a branch, the branch is shaded by the "equivocal" pattern, and clicking on it makes asterisks appear showing which states may be equally parsimoniously placed on the branch.

Turning the trace off

Select Trace Off in Characters menu; or click on trace button again. The trace will also automatically be turned off if you switch to thin or medium branches.

Tracing the next or previous character

Click on one of the arrows above character legend:

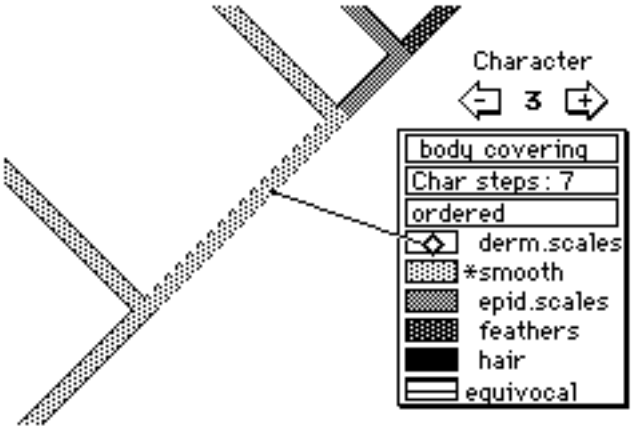
and hold down button until character number desired appears.

> *SHORT CUT: If the Option key is depressed when you click on these arrows, the character number will jump ahead or behind by increments of 10.*

Fixing the state of a branch

Initially, when MacClade traces a character on a tree, each branch will be shaded to indicate the states most parsimoniously placed on the branch. You can examine alternative pathways of character evolution other than the most parsimonious, by fixing the state of a selected branch to a desired state.

This can be done by selecting a branch, and then dropping the diamond over top of one of the shaded boxes in the right hand legend, which indicate the shading used for a character state:



The branch is then forced to have that state regardless of what it costs in terms of number of steps. In the above example, the selected branch, most parsimoniously assigned the state "smooth", is being fixed to the state "derm.scales". The surrounding branches will be re-shaded as appropriate, and the number of the steps will be recalculated, given the fixing of the branch.

This is useful when you want to see how much it would cost in number of steps to have an alternative state in a particular ancestor. It is also useful if the initial shading of the tree had equivocal areas (where more than one state could have been placed at a branch with equal parsimony). One can then see one of the resolutions of this ambiguity by fixing the branch to one of these equally parsimonious alternatives (for instance, the alternative least favorable to any evolutionary hypothesis you may be testing) after which MacClade will trace character evolution in the rest of the tree given this fixing.

The state of a fixed branch is unfixed automatically if another branch is moved onto it, or if a different character is traced, or if the character trace is turned off (even temporarily, for instance for branch swapping or Show Changes Mode), or if Get Tree is called.

Warning: if you fix the state of a terminal branch to a state other than that observed for its terminal taxon, you are really asking "what if the taxon really had this state?" Although one might count an extra step between this state and the state actually observed for the taxon, MacClade does not do so. MacClade does not write such changes in the states of taxa into the data file. As stated above, such fixings are unfixed when another character is traced or trace is turned off.

Warning: For Dollo and irreversible characters you can force the assumptions of the methods to be violated (for instance if you fix state 1 at a branch below a clade with state 0 for an irreversible character). MacClade will <u>not</u> give you a warning if you do this.

Seeing which branches are fixed

To see which branches have their character state fixed, click on the "Show Fix" button on screen:

Those branches whose states have been fixed will be highlighted. This button is visible only when the state of a branch has been fixed.

Unfixing state of a branch

To unfix the state of a single branch, select branch and drop on "Unfix" button:

To unfix states of all branches whose states have been fixed, choose Unfix All from the Character menu.

Algorithms used by MacClade

The algorithms used by MacClade to calculate these reconstructions of character evolution are briefly described in Appendix 1. The algorithm display options (choose Algorithm display in the Options menu) allow you to display the workings of MacClade's algorithms for unordered, ordered, and reversible characters. As described in the appendix, the algorithms of MacClade make a pass down the tree, and a pass up the tree before they finally decide what states to put at each branch. If Down Pass or Up Pass are selected for display, MacClade will display the states assigned to the branches on the downpass and uppass respectively, instead of the final assignment of states. These options were created for our debugging of the MacClade algorithms, but we left them in because you may find them useful to understand the algorithms better, or for teaching.

> *SHORT CUTS: If Down Pass is displayed, you can have the Up Pass displayed without going through the dialog box by holding down the Option key and clicking on the words "Down Pass" near the character legend. Similarly, if the Up Pass is displayed, holding down the Option key and clicking on the words "Up Pass" will cause the Final states to be traced. If either Down Pass or Up Pass is displayed, then double-clicking on those words will bring forth the Algorithm display dialog box.*

Note: if MacClade does not shade the basal node of an outgroup the same as would be achieved using the "outgroup algorithm" of Maddison *et al.* (1984), this does not mean that either the outgroup algorithm or MacClade is in error, for the outgroup algorithm considers the outgroup in isolation from the ingroup, whereas MacClade would consider both when the outgroup is attached to the ingroup (see Maddison *et al.*, p. 88).

5. DIAGNOSTICS OF CHARACTER EVOLUTION

Numbers of steps and Treelength

MacClade gives the length of the tree as a whole and the number of evolutionary steps required for each character. The number of steps required for each character is shown in the legend of the character traced, as described in the previous chapter, and in the diagnostics table described below. The treelength, which is the total number of (weighted) steps of character evolution are required by the tree, is given in the treelength box:

The total treelength is calculated as the sum of the number of steps for the individual characters multiplied by their respective weights. It includes steps that must have occurred within polymorphic terminal taxa, except occasionally for characters of a user-defined type. For characters of user-defined type, changes within terminal taxa polymorphic for more than two states are not counted; in this case the treelength will be underestimated and MacClade will put a '+' in the treelength box to indicate that the actual treelength is higher than the number given.

Algorithms used for counting the number of steps for each character are described in Appendix 1.

Turning tree counting off and on

To turn off tree counting, choose Suspend tree count under the Options menu; to turn the counting back on, choose Count tree under Options menu.

Turning off the count is useful with large data sets when you want to make a lot of branch rearrangements and don't need to know tree length until you're done. Having the tree counted between each branch move can slow things down considerably.

Consistency Indices

MacClade gives consistency indices (Kluge and Farris, 1969) for the tree as a whole and for the characters individually. The consistency index for the tree is given just underneath the treelength box; those for the characters are given in the diagnostics table which is described below.

The consistency index (CI) for each character is calculated as the minimum conceivable number of steps divided by the observed number of steps. Thus the CI would be 1 if the character had no homoplasy, 0.5 if there were twice as many steps as needed, and so on. The minimum conceivable number of steps for unordered characters is the number of states occurring in the data matrix minus one. For ordered, Dollo and irreversible characters, it is the maximum state minus the minimum state. For characters of user-defined types, the minimum conceivable number of steps is in general very difficult to calculate, and for this reason no consistency index is given for these characters. Consistency index is also not given for invariant characters.

The consistency index of a tree is the weighted sum of the minimum conceivable number of evolutionary steps for each character divided by the weighted sum of the observed number of steps for each character. Characters of user-defined type and monomorphic characters are not included in the CI. If the characters in the data set are perfectly congruent with each other and the tree, and thus there is no homoplasy, then the observed number of steps will equal the minimum, and the CI will be 1.00. The messier the data on the tree, and the greater the amount of homoplasy, the greater is the observed number of steps, and the more CI shrinks. CI can range from 0 to 1.

Recall that because MacClade counts length within a polymorphic terminal taxon, there may be some possibly-homoplastic length that cannot be removed by branch moves. Thus the lowest CI for characters with polymorphic terminal taxa that can be attained by moving branches may be less than one.

Version 2.1 of MacClade calculates the consistency index differently from Version 1 when terminal branches have their states fixed. Recall that fixing a terminal branch is interpreted as supposing that the taxon actually had the state fixed, because no steps are counted between the state fixed and those actually observed for the taxon. Occasionally, the only occurence of a state can be hidden by such a fixing. For instance if state 2 of a three state character (0,1,2) occurs in only one taxon, and this taxon is fixed to state 1, then the character effectively becomes a two state character. In such a situation, MacClade Version 2.1 now appropriately reduces the minimum conceivable number of steps used in calculating the CI, in this example to 1, whereas Version 1 left the minimum at 2. Thus Version 2.1 would yield lower CI's in these situations.

Character diagnostics table

The character diagnostics table, which looks like this:

Character	type	weight	states	steps	C.I.
1. amnion	r	1	2	3	0.33
2. appendages	o	2	3	3	0.67
3. body covering	o	2	5	7	0.57
4. thermoreg.	r	2	2	1	1.00
5. int.nostrils	r	1	2	1	1.00

provides information about each character.

To see this table, choose Char. diagnostics from Display menu; **or** click on the diagnostics button:

(You can get rid of this table by clicking the diagnostics button again, or by choosing Taxon Names Only from the Display menu).

The first columns (titled "Character") give the number of the character and its name (if you've given it a name).

The next two columns give assumptions about character evolution. The first of these (titled "type"), lists the transformation type of the character (whether it's assumed unordered, ordered, irreversible, etc.). The next (titled "weight"), lists the weight given the character. We will discuss the character transformation type and character weight later, in the chapter "Assumptions about Character Evolution".

The column titled "states" lists the number of states in the data matrix for that character. Thus, if data matrix contains states 0, 2, and 7 for the character, the number of states will be listed as 3.

The last two columns of the table list of the number of steps on the tree for that character (not weighted by character weight) and consistency index of that character. A '+' following the number of character steps indicates it is underestimated because for character of a user-defined type in which a terminal taxon is polymorphic for 3 or more states (see explanation in Appendix 1).

Scrolling through the diagnostics table

To scroll through the diagnostics table, place cursor in scroll bar at right of screen and click (top half of scroll bar to go backward; lower half to go forward):

SHORT CUT: For a quicker page scroll hold Option key down, and then click on scroll bar.

Expanding or shrinking the diagnostics table viewing area

In order to get an overview of more than the default number of characters, you can pull down the tree to expose more room for the diagnostics table. Do this by placing the cursor over the small, thick, upper, black, horizontal bar at right margin of screen at bottom of scroll bar (see figure above); when cursor changes to

drag the bar down. You can also raise the tree to shrink the diagnostics table section by dragging the bar up.

SHORT CUT: If you have moved the tree or taxon names out of their default positions, they can be quickly returned to the default by holding down the Option key and clicking once on the thicker upper bar.

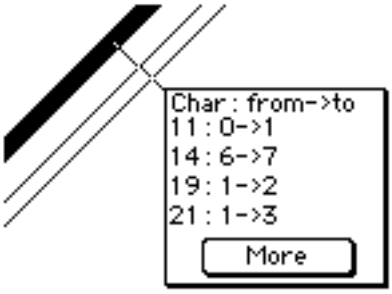
Exposing or hiding the taxon names

Normally, the diagnostics table replaces the taxon names on the screen. If you wish to see the taxon names at the same time as observing the diagnostics table, you can grab and pull the smaller, thinner horizontal bar at right margin of screen at level of top of tree (see figure above). By dragging the bar down, you will expose the taxon names; by pushing it up, you will hide them again.

Showing total number of changes on each branch: Show Changes Mode

To display the number of characters unambiguously changing on each branch, choose Show Total Changes from Display menu. This will get you into "Show Changes Mode", a mode in which the branches are shaded according to how many unambiguous apomorphies there are for each branch: the darker a branch is shaded, the more characters change on it. This is useful if you want to know how many tick marks to put on a branch to indicate apparent apomorphies of the clade. If there is ambiguity in the placement of a change -- e.g., when it is equally parsimonious to suppose two independent derivations or a gain then loss -- Show Changes will not list the change except in the one case described below. In PAUP, the number of unambiguous changes is indicated by the minimum possible length of a branch given by PAUP's BLRANGE option.

To find out what characters change on a given branch, click on the branch. The characters will be listed in the box at the lower right. A line will be drawn from the branch selected to the box. For example, for the branch selected in the following example:



character 11 changes from state 0 to 1, character 14 from state 6 to 7, and so on. The presence of the "More" button indicates that there are more changes not listed in that box; the see the other changes, click on the more button.

Even though these changes along a branch can often be considered apomorphies unambiguously supporting the clade, this is not always a good interpretation. For instance, if the character subsequently reverses in one taxon within the clade, it would be inappropriate to claim that the character supports the monophyly of the clade except in the context of other characters which place the reversing taxon within the clade. Likewise, a tree shaded darkly by Show Changes is not necessarily a good tree, for some of the unequivocal character state changes that occur on the tree might be avoided by a shorter tree.

The only ambiguous change MacClade will indicate in Show Changes is one in the placement of a difference between the right and left descendant branches of the root. If the right descendant branch of the root is reconstructed to have state 1, while the left branch has state 0, then there is ambiguity as to whether the root had 0 changing to 1 up the right side or the root had 1 changing to 0 up the left side, unless the character is irreversible. Because you might be interested in knowing about these characters which might distinguish the left side of the tree from the right, MacClade indicates these by shading the root and listing them when you click on the root branch. When there are such characters MacClade gives a warning that these changes indicated for the root actually belong on either descendant branch.

MacClade's presentation of the changes occurring may be confusing when the reconstruction of character evolution is equivocal for a branch. For instance, suppose one has an ordered character. If a branch is reconstructed as having either state 2 or 3 equally parsimoniously, and its immediate ancestor has state 1, then the change occurring along the branch is either 1 -> 2 or 1 -> 3. You might therefore expect that when you call for Show Total Changes then click on this branch that MacClade would tell you the unambiguous change occurring on the branch is 1 -> 2,3 (state 1 to either state 2 or 3). Instead, MacClade indicates 1 -> 2. MacClade does this because the change from 1 to 2 is considered unambiguous since at least that much change must occur on a parsimonious reconstruction, but the possible additional change from 2 to 3 is considered ambiguous and therefore is not listed. The change from 2 to 3 is considered additional because the character is ordered. MacClade only lists the extent of the change necessary according to the reconstruction. This can be confusing near the root. For instance if the left descendant node of the root is reconstructed as having state 3, the right descendant as having state 1 or 2, MacClade will indicate that the unambiguous change here occurring is 3 <-> 2 (if the character is ordered). The reconstruction allows state 1 at the root transforming to 3 on the left and remaining 1 on the right, but MacClade indicates the change

between the left and right as only between states 3 and 2 because at least this much change is needed; whether the change is actually greater (between 3 and 1) is ambiguous.

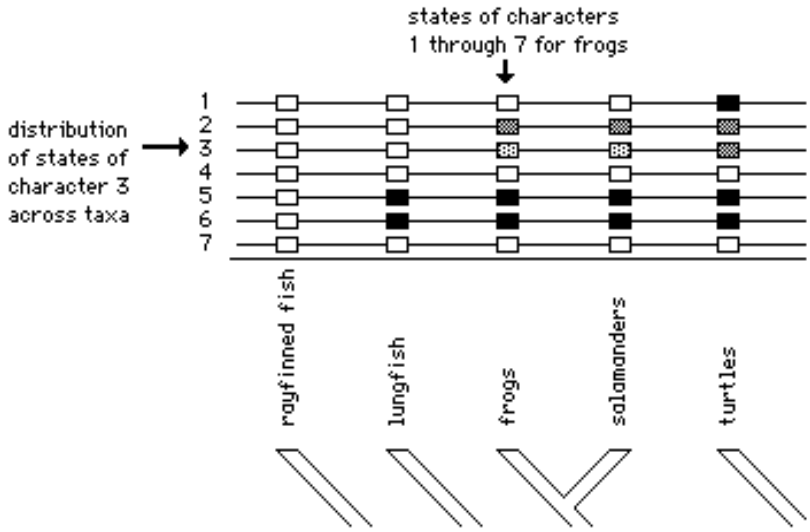
Note that Show Changes Mode is a "locked" mode; that is, you can not move branches around, use the menus, etc.; the only things you can do are click on the tree branches and click on the two or three command buttons, or dump the screen image to a MacPaint file (shift-control-3) or the printer (shift-control-4). The Macintosh isn't allowed to hear other commands you might give it.

To exit from the Show Changes Mode click on the "Changes off" button:

Character data boxes

You can get a graphic overview of the data matrix by selecting Char. data boxes from Display menu or by pressing the data boxes button:

This will display the "data boxes". (You can get rid of the data boxes by pressing the data boxes button again or by choosing Taxon Names Only from Display menu.)



Scrolling through the data boxes

To scroll through the data boxes, use the same method as described above for the diagnostics table.

If you have a large number of taxa on the screen, scrolling through the data boxes can be very slow. It can take a very long time to get from, say, character 3 to character 73. There are a number of faster scrolling methods. One quick way is: expand a clade of 2 terminal taxa to fill the screen; scroll through the data boxes to the desired spot; shrink the tree.

Expanding or shrinking the data boxes area

Use the same method as described above for the diagnostics table.

Exposing or hiding the taxon names

Use the same method as described above for the diagnostics table.

6. ASSUMPTIONS ABOUT CHARACTER EVOLUTION

The course of character evolution and treelength inferred by MacClade will depend upon the background assumptions employed in the analysis. Below we describe how you can specify assumptions about the nature of character state transformations and character weighting. Following this we describe an assumption that you cannot change, about ancestral monomorphism.

ASSUMPTIONS ABOUT TRANSFORMATION BETWEEN CHARACTER STATES

Transformation types

For a given character, various assumptions about transformation between states are possible. You may believe that the states follow a linear transformation series, or that gains are more difficult than losses, and so on. These assumptions can be incorporated into a MacClade analysis by assigning the character to a particular *transformation type*. For example, a character assumed to evolve along a linear transformation series is of an *ordered* type, a character in which it is presumed that any state can transform to any other equally easily is of an *unordered* type. Further details about transformation types are given below.

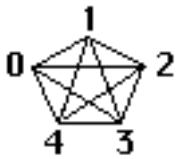
MacClade has 5 predefined transformation types. In MacClade, each of these types is given a single lower-case letter abbreviation; thus, ordered type is designated by "o", unordered by "u". The transformation types predefined by MacClade, and their abbreviations, are as follows:

- unordered (u)
- ordered (o)
- reversible (r)
- irreversible (i)
- Dollo (d).

You can also define your own transformation type (a "user-defined type"), and give it a one letter lower-case abbreviation (other than u,o,r,i, or d). The method used to do this are described below, in the section, "Defining your own transformation types", and in Chapter 7, "Format of Data Files".

The pre-defined transformation types can be characterized as follows:

unordered (Fitch Parsimony) - A change from any state to any other state is counted as one step. Thus, a change from 0 to 1, or from 0 to 8, or 7 to 4, are each counted as one step. A 5-state unordered character can be represented diagrammatically as follows, where change between any two states only involves one step (i.e., only one line has to be traversed in the diagram):



ordered (Wagner Parsimony) - The number of steps from one state to another state is counted as the (absolute value of the) difference between the numbers given to the states. Thus, a change from 0 to 1 is counted as one step, from 0 to 8 as eight steps, from 7 to 4 as three steps. A 5-state ordered character can be represented diagrammatically as follows[Begin Footnote] --- In this diagram, the number of steps in the change between any two states is equal to the number of lines on the path between the two states; thus, from 1 to 4 is 3 lines or 3 states. The diagram does not necessarily imply that for a lineage to evolve between state 0 and 2 it must pass through state 1 (although that interpretation is consistent with the diagram); the diagram simply summarizes the number of steps for each state-to-state change.--- [End Footnote] :

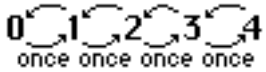
0—1—2—3—4

reversible - (for binary characters only). A change from state 0 to state 1 is counted as one step, as is the reverse.

irreversible (Camin-Sokal Parsimony) - The number of steps from one state to another state is counted as the difference between the numbers of the states, with the restriction that decreases in state number do not occur. Thus, a change from 0 to 1 is counted as one step, from 0 to 8 as eight steps, but changes from 1 to 0 or 8 to 0 are impossible. Multiple gains are allowed, no losses are allowed. A 5-state irreversible character can be represented diagrammatically as follows:

0→1→2→3→4

Dollo - The number of steps from one state to another state is counted as the difference between the numbers given to the states, with the restriction that each increase in state number occurs only once. Thus, a change from 0 to 1 is counted as one step, from 0 to 8 as eight steps, but gain of 1 or 8 can only happen once each on the tree. Multiple losses are allowed, multiple gains are not. A 5-state Dollo character can be represented diagrammatically as follows



Note that MacClade adds one more assumption: if two states of a Dollo character might be otherwise equally-parsimoniously placed at the root, MacClade will place the lower-numbered state there.

MacClade reserves the words "ordered" and "unordered" for multistate characters, and uses "reversible" for two-state (binary) characters. The reason for this is that there is no useful distinction between ordered and unordered with binary characters (see Meacham, 1984). Thus MacClade calls all two-state characters which are not irreversible (and not of user-defined type) "reversible". Indeed, if you code a two-state character as being ordered or unordered, MacClade will treat it as being of reversible type. Internally, MacClade treats these two state reversible characters by the same algorithms as unordered multistate characters, and when saving the transformation types to the file, MacClade designates two-state reversible characters by the letter 'u', so that if you later add a new state the character will be treated as an unordered three-state character.

These transformation types are essentially specifications of intra-character weightings. Thus, if you assume a character to be ordered, you are weighting some state changes more highly than others. For example, a change in an ordered character between states 0 and 1 would count

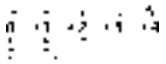
as 1 step, whereas as change between state 0 and 8 would be weighted 8 times as much (that is, it would count as 8 steps), and a change from 7 to 4 would count as 3 steps. For an ordered character then, the weight given to a change is equal to the (absolute value of the) difference between the numbers given to the states. If you assume a character to be unordered, you are weighting all state changes equally, such that a change between states 0 and 1, 0 and 8, or 7 and 4 each would count as one step.

The default transformation type for MacClade is unordered for a multistate character, reversible for binary characters. MacClade uses these types as defaults because they are considered to embody weaker, more conservative assumptions than ordered types. For instance, designating a character as being of ordered type will often increase the number of steps; it can never decrease it. Unless there is good reason to support the assumption that characters are ordered, it seems better to make the more conservative assumption and to use unordered characters.

If you want to have characters polarized, outgroups must be entered into the data matrix explicitly. State 0 (or whatever is the lowest state in the matrix) is not automatically assumed to be ancestral except for irreversible characters (and see note under description of Dollo characters, above).

Defining your own transformation types

In addition to the pre-defined transformation types discussed above, MacClade users can specify their own assumptions. You can do this by indicating in a matrix in the data file how many steps it is for a transformation from each state to each other state. For instance, suppose you had an assumption that transformation between states followed the pathways indicated on this diagram:



Thus, this supposes that for increase in character state (which might be thought of as gains) the character behaves as an ordered character, with one step between 0 and 1, three steps between 0 and 3, and so on. For any decrease in character state, there is one step. Thus, you may believe gains have to be accumulated, but any loss can be accomplished in a single step (perhaps by deletions or modifiers). You can take this diagram and translate it into the following matrix:

		To:				
		0	1	2	3	4
From:	0	0	1	2	3	4
	1	1	0	1	2	3
	2	1	1	0	1	2
	3	1	1	1	0	1
	4	1	1	1	1	0

This matrix claims that there are no steps between 0 and itself, four steps from state 0 to state 4, one step from 4 to 0, and so on.

In MacClade, each element in the matrix must be between 0 and 9 or the special infinity character ∞ (obtained by the keystroke Option-5). This special character indicates that the change is prohibited. This format for indicating distances between states and the algorithm that uses it are very flexible, and with it you can code all sorts of elaborate assumptions such as asymmetry in number of steps between gains and losses, and so on. You can even code some of the familiar transformation types in this format:

unordered	ordered	irreversible
011111	012345	012345
101111	101234	∞01234
110111	210123	∞∞∞0123
111011	321012	∞∞∞∞012
111101	432101	∞∞∞∞∞01
111110	543210	∞∞∞∞∞∞0

However, it is much better to use the built in types u, o, i, than these recoded versions, for MacClade handles the built in types much more quickly and more flexibly than user-defined types.

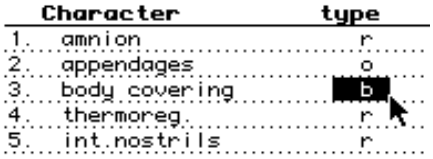
The exact methods for coding these matrices into the data file are indicated in the discussion of the format of data files. After you have coded them into the data file, you can assign these types to your characters as discussed below.

Three limitations on the use of characters of user-defined types are: (1) consistency indices are not given, (2) the number of steps due to a terminal taxon's being polymorphic for more than 2 states is not added to the counting of steps for the character and tree, and (3) branch swapping ignores these characters. The reason for (1) is explained under Consistency Index, for (2) is explained in Appendix 1, and for (3) is memory limitations on a 512K Mac.

Assigning transformation types to characters

The type currently assigned to a given character is displayed in the "type" column of the character diagnostics table, and in the second or third box of the character legend in the lower right-hand corner of the screen.

There are several ways to change the type of a character. The character diagnostics table provides one convenient way. Just click on letter indicating type of the character:



and hit the letter on the keyboard for the new type desired, then click the pointer elsewhere or hit Return key.

If a character is being traced, you can use a similar procedure, this time clicking on the name ("ordered", "unordered", or whatever) of the traced character's type in the character legend:

If MacClade thinks that the letter you typed is illegal (for example, typing an "e" for a user-defined transformation type, when no such transformation type was defined in the data file), it will beep and not enter the letter.

If you do click on an acceptible user-defined transformation type letter, or type in such a letter, the matrix of that transformation type will be displayed on the left hand side of the screen near where the tree length box usually appears:

```
Trans. type: b
01222
10111
21011
21101
21110
```

Discovering available transformation types

If you wish to see which transformation types are available (that is, the predefined types plus those defined in your data file), choose "Show Available" from the Characters menu.

> *SHORT CUT: double-click on the letter indicating the type of character in the diagnostics table, or on the name of the transformation type in the character legend.*

Saving sets of character transformation type assignments

Sets of transformation type assignments can be stored in the data file, and recalled for later use. To store the current set in the data file, choose "Save Types" from the "Characters" menu; you will be presented with a dialog box, asking you to give a name to the set of assumptions. After typing in a name, click on the "Save" button. You can then alter the transformation types again, store the new set in the data file, and so on; there is no MacClade-imposed limit to the number of sets of transformation assumptions you store.

Getting sets of character transformation types

Select Get Trans Types in Characters menu. You will be presented with a dialog box that will allow you to choose which type set you wish to use.

> *SHORT CUT: double-click on "Type" in title line of diagnostics table to bring forth the dialog box.*

Resetting to type set last loaded from file

To reset the type set to the one last loaded in from or last saved to the data file, choose Reset Types in Characters menu.

Finding the name of current type set

To see the name of the current set of transformation type assignments, click on the word "Type" in title line of the diagnostics table.

If you have changed the transformation type set from the one last read from the data file or last saved to the file, the current set is listed as "untitled".

CHARACTER WEIGHTING

In addition to the intra-character weighting provided by assigning transformation types, you can specify inter-character weights. This would be advisable when you have reason to believe that some characters will provide better evidence than others. MacClade allows only integral weights, from 0 to 9.

Changing weights

To change the weight of a character, click on the weight listed in the character diagnostics table and hit the number for the weight desired, then click the pointer elsewhere or hit Return key, in much the same manner as for changing transformation types.

Saving sets of character weights

Sets of weights can also be stored in the data file; choose "Save Weights" from the "Characters" menu; type in a name, click on the "Save" button. You can then alter the weights again, store the new set in the data file, and so on; there is no MacClade-imposed limit to the number of weight sets you can store.

You can use different weight sets to examine different character suites independently. For instance in one weight set you could weight all larval characters 1 and all adult characters 0; in another set the larval characters 0 and the adult 1. By examining the sets alternately, the contributions of the different suites of characters can be assessed.

Getting sets of character weights from data file

Select Get Weights in Characters menu. You will be presented with a dialog box that will allow you to choose which weight set you wish to use.

> *SHORT CUT: double click on "Weight" in title line of diagnostics table to bring forth the dialog box.*

Resetting to weight set last loaded from file

To reset the weight set to the one last loaded in from or last saved to the data file, choose Reset Weights in Characters menu.

Finding the name of current weight set

To see the name of the current set of weights, click on the word "Weight" in title line of the diagnostics table.

If you have changed the weight set from the one last read from the data file or last saved to the file, the current set is listed as "untitled".

An assumption you cannot change

The algorithms used for reconstructing character evolution assume that for each character there is only one state at each point on the tree's branches. That is, ancestral species are presumed to be monomorphic. The only place where polymorphism is allowed is in the terminal taxa. For ancestral branches MacClade sometimes shows an equivocal pattern representing more than one state, but this does not mean that such a branch is polymorphic. Rather, it means that any of the states represented could by itself be placed on that branch.

This assumption of ancestral monomorphism is rather harsh, but it does make life simpler, at least for the programmer. If polymorphisms were allowed, one could explain any character distribution by supposing a highly polymorphic ancestor with successive loss of states, or by supposing convergence, or in many other ways. Thus one would have to balance gains, retention, and loss in deciding which reconstruction of evolution is preferable. Effectively, assuming widespread ancestral polymorphism could be equivalent to claiming that character inheritance is little constrained by, and thus little indicative of, the branching phylogeny.

There is at least one way you can bypass these problems and allow ancestral polymorphisms for 2 and 3 state characters in MacClade if the data are recoded appropriately. Recode the data to represent populations states instead of organism states. For instance a binary character which could exist monomorphically as states 0 or 1, or polymorphically as [0,1] gets recoded so that its new states are 0 and 2 for the monomorphic cases and 1 for the polymorphic case. Then you can choose or make a transformation type which indicates your assumptions about how many steps are required between one population state and any other. Perhaps the simplest assumption is that there is one step for any gain or loss of a state and no penalty for retaining polymorphisms; the recoded character could then be treated as ordered. This logic can be extended to a 3 state unordered character . If one believed it to be one step for any gain of a state and one step for any loss, with no penalty for retaining polymorphisms, a user-defined type with the following matrix of steps could be used:

```
0221132
2021312
2203112
1130221
1312021
3112201
2221110
```

where the character states 0,1,2,3,4,5,6 represent the original conditions 0,1,2 [01], [02], [12], [012] respectively. Thus it is 2 steps to go from condition 0 to [012] but 3 steps to go from 0 to [12]. Users interested in exploring more ideas about retained polymorphism may want to investigate some of the options of Felsenstein's PHYLIP computer package.

7. FORMAT OF DATA FILES

For MacClade to be able to read a data file, the taxon names, coding of character states, and so on, must follow the format described below. Please follow the instructions carefully. (Multiplan or Excel spreadsheet files follow a different format, described in Appendix 3.)

The basic structure of a MacClade data file is as follows:

_____

Title of data set.

Names of characters and states and matrices for user-defined types (optional).

Data Matrix.

Storage of assumptions about character transformation and weights (optional).
Storage of trees (optional).

_____

Thus, the simplest possible data file would look like this:

```
A
t0
@
```

with a title, one taxon by one character matrix, and the special end of data matrix symbol @ (rather useless data matrix, of course). A slightly more useful but still very simple data file would look like this:

```
This is the title
Homo sapiens  0
Drosophila    1
Rattus rattus 1
E. coli       0
@
```

It consists of the title, a data matrix of one character for four taxa, and the @ symbol. Note that the other parts of the data file listed above (names of characters, assumptions, trees) are optional. You may want to become familiar with MacClade before you do your own editing of assumptions and trees. You can live without ever editing the assumptions and trees, for you can let MacClade save them to the file. We describe the format for these sections of the data file in fine print below, in case you want to edit them.

Before explaining the rules for each section of the file, we will explain some general rules. Except where noted, lines can be up to 255 characters long, but no longer. Comment lines are allowed anywhere after the title, except in the middle of matrices defining user-defined types, and in the tree storage area. A comment line must begin with a double quote character ("); the line is ignored by MacClade. You can use comment lines to make reading the data file easier, for instance by making headings for the data matrix or by separating the taxa in blocks of five. Blank lines are ignored except in the middle of matrices defining user-defined types.

NOTE that data file format in version 2.1 of MacClade differs from version 1 in two important ways: (1) the data matrix must end with a line beginning with @, and the * and $ lines are now optional, and (2) Tree names must begin with #.

Title:
The first line of the data file must be the title.

Character names and states (Optional):
Following the title you can add, if you want, lines that specify the names of characters and the names of character states. Begin these lines with the number of the character (if you start the line with a space, MacClade will not recognize it properly). After the number put a period and a space, then the name of the character. If you don't want to name the character states, then leave the rest of the line blank. If you want to give names to the character states as well, put a colon after the character name. Then, for every state from 0 through the highest state number, put a space, statename, semicolon. Thus, these lines take the form:

character number. character name: state 0 name; state 1 name; state 2 name;
or
character number. character name

For example, if character number 22 is eye color and has states 0 = blue, 1 = green, 2 = brown, you would type:

22. Eye color: blue; green; brown;

Each of these lines with character and state names must contain no more than 128 characters. Any text occurring after the last semicolon in each of these lines is ignored by MacClade, and can be used for comments. You needn't give names to all characters in a file.

Matrices for user-defined transformation types (optional):
After the title, but before the data matrix, you can define your own character transformation types. More information on user-defined types is given above, in chapter 6, "Assumptions about Character Evolution". To define transformation type 'a' as described in that section, type a line '%a'. This tells MacClade to read in the next several lines as a transformation matrix for type 'a'. Then, immediately following the line with %a, type in the matrix, flush left with no spaces:

```
%a
01234
10123
11012
11101
11110
```

Recall that the matrix must have only numerals 0-9 and the symbol ∞ to indicate a change is prohibited. If you find it easier to edit an existing matrix rather than starting one from scratch, you can ask MacClade to insert a template matrix (of 1's and 0's, matching that of an unordered character) by choosing Paste Matrix from MacClade's Edit menu. Note that this template matrix will be 10x10. If you will be using your defined transformation type for characters with only, for instance, 4 states, you only need to specify a 4x4 matrix. If you input a 10x10 matrix and use it for a 4 state character, the algorithm will effectively only use the upper left 4x4 corner of the matrix (unless the matrix is strangely coded). The algorithm,

however, goes through the whole matrix, and it will be slowed down if you use a matrix bigger than needed for your character. It is thus best to edit down your matrix so it is only as big as needed.

If your matrix coding has some peculiar features (non zero elements on the diagonal, violation of the triangle inequality) MacClade will give you a warning. A non-zero element on the diagonal would mean it costs steps for a character to stand still. A violation of the triangle inequality would mean that it is possible to find a shorter path from one state to another by going an indirect route through another character than by going directly. Both of these assumptions would usually be nonsensical, and MacClade's algorithm may handle them in strange ways, sometimes giving an assignment of ancestral states that is parsimonious only in a restricted sense. Violating the triangle inequality may sometimes seem to make biological sense. It may be that you believe that it is very unlikely that state 0 would evolve directly into state 1, and thus you might code the 0 to 1 change in the type matrix as 9. You may also believe that state 0 can easily change into state 2, which in turn can easily turn into state 1; you might code the 0 to 2 change as 1 step, and the 2 to 1 change as 1 step. However, this coding would be incorrect if used literally in the type matrix, for it interprets the entries in the matrix as indicating the number of steps for direct transitions between states. Instead, the entries in the matrix are supposed to indicate the number of steps required to go from one state to another (whether direct or indirect). Thus you should place in the 0 to1 position of the matrix the number of steps on the shortest path between states 0 and 1. In the example, the matrix should indicate 2 steps for a transition from state 0 to 1, not 9 as would be required for a direct transition.

It is possible also to code the matrix in a way that makes the character state transformations impossible. For example, this coding:

implies that for every state there is at least one other state it can't transform to. This means that if 0, 1 and 2 are in the data, then they cannot all be evolved. If this is the case MacClade will count the number of steps in the character as being greater than 1000, and give a warning.

Data Matrix:
The data matrix has the taxon names at the left, the character data at the right. You do not need to specify the number of taxa and characters, as MacClade reads data files in a fairly flexible way and will figure out the number of taxa and characters for you.

Taxon names: can be 1 to 16 characters in length. The following characters cannot be used in taxon names:

: [ ] ( ) ,

while @ and # cannot be used as the first characters of taxon names. Also, for the first taxon, the name must not include any numeral or ? and must not begin with ", ∞, or %. The reason for this is that MacClade figures out what line is the first line of the data matrix by looking for the first line after the title that begins neither with a number nor ", ∞ nor %. MacClade then figures out where the character information is by scanning along this line for the first occurrence of a numeral or ?. Numerals should never be used as the first character of any taxon name. Indeed, it is best not to put numerals anywhere in any taxon names, because then the taxon name may be displayed incorrectly, especially if "2" is put in the name.

Character data:

After the name of the taxon, the character states for the first character through the last are specified by numbers (integers between 0 and 9). The character state for a taxon must be aligned with that of the other taxa, i.e. if character 3 appears in the 15th column for one taxon, it must appear there for all taxa. Specify missing data by ?. You may separate sections of the matrix by totally blank columns. Tabs are not allowed in the data matrix!!

To indicate that a taxon is polymorphic for a character, add a line just below the taxon, beginning with &. In the column corresponding to the character, place the number for the second state that the taxon has. Additional & lines can be added if the taxon has more than two states for a character. A coding of polymorphic for a species may mean that several states coexist in its populations; a coding of polymorphic for a higher taxon would mean that the taxon was observed to have more than one state, but that we are ignorant of which state is ancestral for the taxon. MacClade prohibits polymorphisms to be retained in the branches below the terminal taxa, as discussed in Chapter 6, "Assumptions About Character Evolution", except if a polymorphic taxon is forced down to ancestral position (see Make Ancestor, in the Tree manipulation chapter).

If you want to have characters polarized, outgroups must be entered into the data matrix explicitly. State 0 is not automatically assumed to be ancestral except for irreversible characters.

<u>End of matrix:</u>
Following the data matrix must be a single line beginning with "@"; this signals the end of the data matrix. Note that, unlike version 1, you needn't put * and $ lines after the data matrix.

Assumptions about character transformation and weights (optional).
Following the data matrix is the storage area for assumptions about character transformation and character weighting. You can add lines recording these assumptions by hand in the data file if you wish, or your may prefer to change these assumptions in the Diagnostics table while a tree is on screen, and let MacClade save the changed assumptions. Each set of assumptions falls on a separate line; each of these lines must begin with a * (for sets of character transformation types) or $ (for sets of character weights). All $ lines must follow all * lines; they can not be intermingled.

* lines indicate whether the character is unordered, ordered, irreversible, Dollo, or of a user-defined type. See Chapter 6, "Assumptions About Character Evolution" for a discussion of transformation types. To make a * line, type * in the first space on the line, then place the letter (eg., u, o, i) designating the transformation type desired for each character in the column corresponding to the character. If you make no indication, unordered is assumed. MacClade will treat 'u' and 'o' specifications as equivalent for two-state characters and will label characters so specified as "reversible" in the legend or "r" in the character diagnostics. If you make no indication for a two-state character, reversible is assumed. Each * line might be thought of as a scheme of transformation assumptions for all the characters. You can have multiple * lines, each recording a different scheme.

$ lines indicate the weight of characters. To make a $ line, type $ in the first space on the line, then place a number from 0 to 9 indicating the weight of each character in the column corresponding to the character. If you make no indication, weight 1 is assumed. Each $ line might be thought of as a weighting scheme. You can have multiple $ lines, each recording a different scheme.

Storage of Trees (optional)
Each tree takes up two or more lines in the file:
First line has the name of the tree, which must start with # and be followed by a 16-character or less tree name.
Subsequent lines contain specifications for the tree. Only dichotomous trees are accepted by MacClade. Parenthetical notation is used for trees, as in Example 2, with a semicolon at the end of the line. In example 2 the first tree ("Typed") was typed in by us, the second tree ("Saved Compact") is the same tree, only saved in the file by MacClade using its slightly more compact form, while the third tree ("Saved By Names") is also the same tree, saved by MacClade using the standard tree descrition format. If taxa are referred to by numbers as in the first two tree descriptions, taxon numbers come from the order of the taxa in the data matrix. Thus the "Typed" tree has the SixthTaxon and SeventhTaxon as sister groups. You don't need to add right hand parentheses; they are redundant since MacClade accepts only dichotomous trees. If taxa are referred to by names, blank spaces in their names are replaced by the underline characters '_'. Tree descriptions can extend onto up to 8 lines, but a single name or number cannot be broken up on two different lines. Tree names and descriptions must occupy adjacent lines, without interspersed blank lines or comment lines. If you input a tree specification that is incorrectly formulated or longer than 255 characters, MacClade may crash or may show you a tree lacking some of the taxa.

Instead of trying to edit tree specifications yourself, you are probably better off starting with the default pectinate tree that MacClade supplies, rearranging on screen to the desired tree, then saving it to your file. Hint: when doing many rearrangements on a large tree, expand small clades and rearrange within these as much as possible because it takes less time for MacClade to draw a small clade. Also, turn off the count of treelength & tracing of characters for extra speed.

When MacClade saves trees it will save them at the end of your data file.

Polychotomous trees are not yet accepted by MacClade. In general, the character evolution and length of a polychotomous tree can be calculated exactly only if one assumes the polychotomy to be real (i.e. multiple speciation) and not an uncertainty. If one views the polychotomy as an uncertainty in resolution (as is the usual interpretation), and the polychotomy is left unresolved, then in general character evolution and tree

lengths can be only approximately determined. This makes for some difficulty. Polychotomies will be in an upcoming version of MacClade, however. For now you can try inputting one of the resolutions of the polychotomy and then rearrange on screen to try other resolutions. This indeed may be the most informative way to learn about the polychotomy, as long as there aren't too many resolutions to try.

Note on File Size:

MacClade's text editor will not work on files larger than 32 K. An 80 x 200 data matrix would take up about 19 K, but if the file has lots of information about character and state names, and lots of trees saved, the 32 K limit may be exceeded.

Example 2. A more complex datafile with 10 taxa and 10 characters.

```
My favorite group

1. eye color: blue; green; brown;
2. # of toes: 4; 5;
3. size of nose: absent; small; medium; large; xlarge;
4. wings

%a
01234
10123
11012
11101
11110

"==Data======12345 6789.
"===========+===== ====1
FirstTaxon   00000 00000
SecondTaxon  01103 0?100
&            1
ThirdTaxon   01200 11211
FourthTaxon  0?203 11003
FifthTaxon   2?118 12211
"5==========+===== ====1
SixthTaxon   21310 10001
SeventhTaxon 21410 00000
EighthTaxon  01101 00210
NinthTaxon   01101 00000
Outgroup     01018 00000
"10=========+===== ====1
@

*my beliefs    a i   oi
$my weights    2   3

#Typed
(10,(1,(9,((6,7),((2,4),(3,(8,5)))))));

#Saved Compact
(10(1 (9 ((6 7 ((2 4 (3 (8 5 ;

#Saved by Names
(Outgroup,(FirstTaxon,(NinthTaxon,((SixthTaxon,SeventhTaxon),
((SecondTaxon,FourthTaxon),(ThirdTaxon,
(EighthTaxon,FifthTaxon)))))));
```

8. EDITING DATA FILES

This section describes use of MacClade's own text editor. If you wish to use another word-processor or a spreadsheet program for editing your files, see Appendix 3.

Creating a file

To create a file using MacClade itself, start MacClade by double-clicking on the MacClade icon. If MacClade is already started, then close the currently active data file by choosing "Close" from the "File" menu (only one data file can be used at once; thus, you need to close the old one first). You should be presented with a blank grey screen. You can now create a new MacClade data file by choosing "New" from the "File" menu; MacClade will present you with a blank text-editing window. Type in the data file (see "Editing the Data File" for instructions on using the text editor). The data file can be saved using the "Save As..." command from the "File" menu (see below).

Editing a file

You can edit the data file before you display a tree on the screen, or while a tree is being displayed (by choosing Edit File from the File menu). Note that if a tree is being displayed, and you have not saved the current transformation type set or weight set to the data file, and if you then you go into the text editor and edit and save your file, your current transformation type set and weight set will be forgotten by MacClade. For this reason, MacClade will ask you before going into the text editor if you wish to save these unsaved sets to the data file.

The MacClade editor has several special features. It has two special commands (Paste Matrix and Check Data, both under the Edit menu), described below. Special characters available include the female symbol (option-f) and the male symbol (option-m). The character that normally resides at option-m, µ, has been moved to option-y. Two characters can be used to produce solid black lines for setting off regions of the data file: a series of option-period produces thin lines, a series of option-colon produces thick lines. See the help file for examples of their use. The Tab key does not insert a tab, but instead inserts 5 blanks. No "Undo" command is available in the text editor.

Otherwise, the MacClade editor follows the usual Macintosh text-editing conventions. If you have used editors such as MacWrite or Word, you needn't read the following description which is intended for users unfamiliar with the Macintosh.

Note that when you move the arrow cursor over the text, it changes into an I-beam cursor. Note also that within the text itself is a flashing vertical bar. This is called the "blinking insertion point". When you start typing, the new text will be inserted at the blinking insertion point. To move the insertion point to another location so that you can insert text there, move the I-beam cursor over the desired location of insertion, and click the mouse button.

To select a section of already typed text, move the I-beam cursor over the start of the section, click then hold down the mouse button as you move the mouse until you have highlighted (white on black) the section desired, then release the mouse button. You can also select a section by placing the blinking insertion point at the start of the section, then move the I-beam cursor to the end of the section desired and, while holding the Shift key down, click the mouse button. This will select all text between the insertion point and the second click. Once you have selected text, you can delete it, replace it, cut or copy it.

To delete a section of text, you can (1) place the blinking insertion point at the end of the section and backspace over it, (2) select the section and hit backspace, (3) select the section and chose Cut from the Edit menu.

To replace a section of text with new text, you can (1) select the section and type the new text, or (2) select the section and choose Paste. These actions will replace the old text with the new text typed in or stored in the clipboard respectively (see below).

The commands Cut, Copy and Paste (in the Edit menu) move text into and out of a special part of the Macintosh's memory called the clipboard. If you select text then choose Cut or Copy, the text is stored in the clipboard, the difference being that Cut also deletes the text from your editing window. Once text has been thus stored you can later (within MacClade or even after you've quit MacClade and started another program), choose Paste which inserts the stored text at the insertion point. For instance, to move a line from one data file to another: (1) open the first file and choose Edit, (2) select the line, (3) choose Copy to store the text, (4) close the data file, (5) open the second data file, (6) move the insertion point, (7) choose Paste to insert the stored text.

Checking the data matrix for errors

If you wish to check to see if the data matrix has any errors (perhaps you mistyped the letter o instead of a zero, or the letter L instead of a one), choose "Check Data" from the Edit menu. If MacClade finds any errors, it will beep twice and show you the first error it finds, by selecting the offending character. You can then fix that character, choose Check Data again, and so on, until MacClade finds no further errors.

Check Data also provides one other service: If there is a line in the data file longer than 255 characters (something that may cause MacClade to blow up), Check Data will give you a warning to that effect. Note that the MacClade text-editing window only scrolls 255 character to the right. Thus, if a line of yours goes off the screen to the right when the window is scrolled as far as it can go to the right, your line is longer than 255 characters.

Pasting user-defined type template into file

Choosing Paste Matrix under the Edit menu will automatically insert a 10x10 of 0's and 1's to serve as a template for user-defined character types.

Saving a file to disk

Once you have created or edited your file, and you wish to save your changes onto the disk, you can choose either Save or Save As. Save will update the current file on disk; thus you loose the preceding version. Save As allows you to save the file under a new name; the previous version will remain under its old name.

NOTE: MacClade gets data from the file on disk, not from memory. Thus, if you make any changes to the data file, and then want to go to the tree window to work on a tree, you must first save the changed file to disk if you want to use the changes you have made.

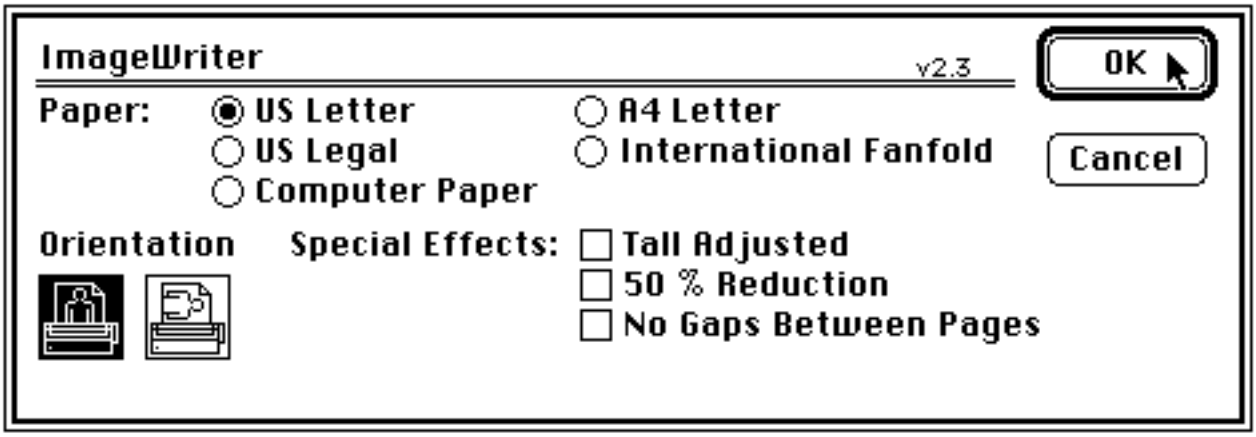
9. PRINTING

MacClade is Imagewriter and Laserwriter compatible. We're not sure about its compatibility with other printers; if anyone has had any experience, good or bad, we'd like to hear from you.

Before trying to print anything, remember that your disk must contain the printer driver file appropriate for the printer being used. To print on the Imagewriter, the Imagewriter file must be present. To print on a Laserwriter, the Laserwriter and Laser prep files must be present.

Setting the paper size, orientation, etc.

Before you tell MacClade to print something, you may wish to set up the printer for printing on different papers, printing sideways, using special effects, etc. To do this, choose "Page Setup..." from the File menu. If you are to print on an ImageWriter, a dialog box like the following will appear. You can make your selections, and click on the OK button.

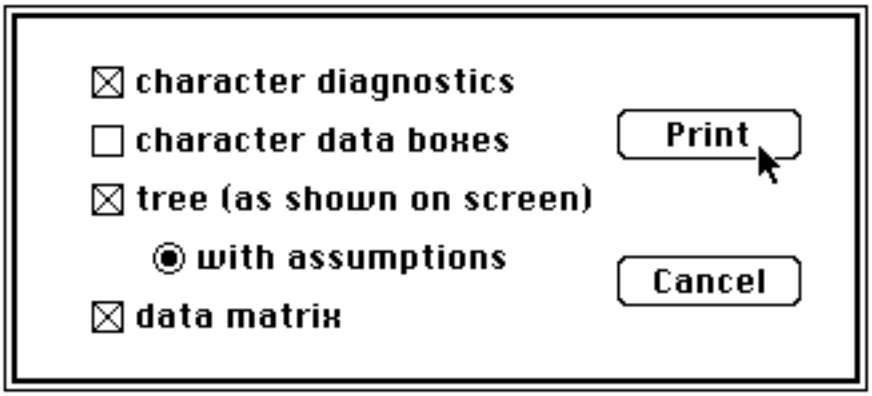


Printing data files

To print the data file, the edit window must be open. Once it is, choose "Print File" in the File menu to tell MacClade to print the file.

Printing trees and character information

When a tree is on the screen, and the edit window is not open, then choosing "Print..." from the File menu will bring forth the following dialog box:



Select the items you wish to print by clicking on the little boxes, and then click on the Print button. In the example shown above, all items except character data boxes have been selected. The different items are:

character diagnostics: if this is selected, the character diagnostics table for all characters will be printed.

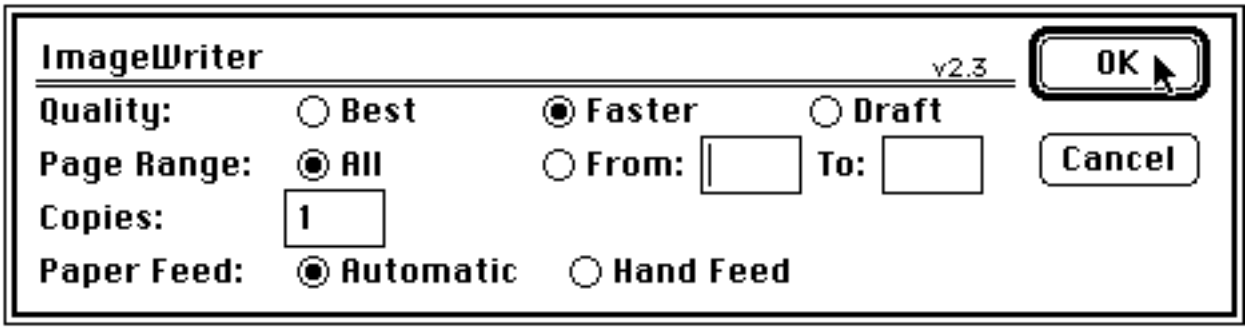
character data boxes: the character data boxes for all characters and all taxa will be printed.

tree (as shown on screen): the tree will be printed, as it appears on the screen. Thus, if the tree has been lowered, it will be printed in that way. Note that the screen will be cleaned up a bit (some of the buttons on screen will be erased). To print the screen as is on the ImageWriter, without erasing these buttons, hit shift-command-4 (see your Macintosh manual for more information on this command).If "with assumptions" is chosen, then a listing of character transformation types and character weights used will be printed.

data matrix: data matrix will be printed.

Adjusting quality of print, number of copies, etc.

Once you have selected that which you wish to print, you will be presented with a dialog box that looks something like this, if you are printing on an Imagewriter:



This box allows you to choose the quality of the printing (Best, Faster, and Draft), the range of pages to be printed, the number of copies to be printed, and whether or not you are using pin-fed (automatic) or hand fed paper.

REMEMBER: on an Imagewriter, graphics will not be printed if you select draft quality; thus, don't use it to print the tree or the data boxes!

WARNING: If you are printing the data file in Draft mode, be wary of cancelling the printing by holding down the control-period keys. MacClade may crash if you do this. MacClade may also crash if you cancel Draft printing by other means.

Printing Show Changes screen

In Show Changes Mode, click on the Print Screen button:

Printing on the LaserWriter

MacClade can print directly onto the LaserWriter, but the tree branches have nicks in them and it takes a long time for the pages to be printed. For better quality tree branches it is better to save the screen image with the tree then edit and print it using a program like MacPaint, as described in the next section.

Fixing up MacClade images for later printing

You may wish to spruce up a MacClade tree for later output on the Laserwriter for publication-quality graphics. To do this, you can save a screen image as a MacPaint file and later touch it up with MacPaint or another graphics package. To save the screen image, hit the keys

shift-command-3; this will dump the image into a MacPaint file called Screen 0, Screen 1, ... or Screen 9.

10. USES OF MACCLADE

MacClade is primarily designed as a tool that will allow systematists to explore various phylogenetic hypotheses and hypotheses of character evolution. We present here a few examples of its use. If anyone comes up with an interesting example of MacClade's use, we'd like to hear of it.

STUDYING CHARACTER EVOLUTION

MacClade's greatest strength perhaps is in analyzing the pathways of character evolution. When a character is traced, MacClade shades the branches to indicate the states that can be parsimoniously put at each branch. This can indicate number of independent derivations, and so on. These shadings can be manipulated by fixing the state at various branches. For instance, in a study of the constraints on the evolution of spider chromosomes, one of us (WPM) traced the evolution of the XXO and XXXY sex chromosome systems for a genus of jumping spiders on a tree provided by the work of C.E. Griswold. On this tree, MacClade's tracing showed 5 or 6 independent derivations of the XXXY system. But two questions remained: how good is the evidence that there were independent derivations (instead of single gain and multiple losses), and how much does this conclusion depend on the details of the tree? By fixing the states at branches, an alternative tracing on the same tree was made in which the XXXY was gained once and lost multiply. MacClade showed that this alternative was not supported, requiring 7 more chromosome fusions than the one with multiple derivations. But, if one made only a few branch moves, the difference between the multiple gain and multiple loss hypotheses was reduced to only one fusion. The ability to fix states and move branches yields much flexibility for studying alternative hypotheses of character evolution. See Felsenstein (1985) for a discussion of the importance of using phylogenetic information in studying evolutionary processes.

COMPARING TREES

You may wish to compare two or more trees for your group of organisms. For example, one of us (DRM) recently worked with G.E. Ball on a project on the ground beetle genus *Amblygnathus*, in which a tree was proposed based on standard Hennigian mental argumentation. PAUP was then used to search for most-parsimonious trees. The trees found by PAUP all differed from the mental tree in placements of a number of different groups. Using MacClade, these different placements could be compared easily. For example, whereas the mental tree placed a group of species similar to *A. suturalis* as a monophyletic group, all of the PAUP trees shattered the group into a paraphyletic assemblage. MacClade allowed exploration of these two different treatments. The appropriate clade (containing the *A. suturalis* group and the species presumed evolved from that group according the PAUP trees) was expanded on the screen; using the "Reroot" function of MacClade, trees with the root of that clade within the (therefore-paraphyletic) *A. suturalis* group and trees with the root outside of the (therefore monophyletic) group could be compared. It turned out that one character was responsible for the shattering of the group in the PAUP analysis; when this character was given zero weight, these trees differing only by placement of that group had the same length. For the other controversial placements, MacClade allowed a quick comparison of the competing placements. One placement could be studied, character by character, and then one or a few branch moves would yield the other placement for comparison. In general, a number of trees can be stored in the data file, and can be quickly compared for various aspects of interest.

FINDING PARSIMONIOUS TREES

Because MacClade's forte is not in finding the shortest trees, it would be most profitably used in combination with the more sophisticated tree-finding programs like, PAUP (D. Swofford), PHYLIP (J. Felsenstein), PHYSYS (J.S. Farris and M.F. Mickevich) and the forthcoming Hennig

(J.S. Farris). MacClade could be used initially to test favorite trees, explore the data set, explore different assumptions, and then the data set could be sent through one of the above programs for a more powerful tree search. The user could then return to MacClade to explore the newly found trees. PAUP will soon be available on the Macintosh, and PHYLIP is now available on the Mac (thanks to Willem N. Ellis, who has ported and compiled version 2.9 on the Mac. You can contact him at Institute of Taxonomic Zoology, Dept. Entomology, Plantage Middenlaan 64, 1018 DA Amsterdam, The Netherlands.)

Nonetheless, MacClade can be used to search for short trees. Perhaps the best way to proceed is to begin with a fairly reasonable tree (perhaps based on your experience with the group), and use local branch swapping on the whole tree. Try alternating between Single and Full swapping (since Single can find equally parsimonious alternatives which may allow you to escape from a local optimum). Next, examine the character state distributions (using the character data boxes), to see if any other rearrangements are suggested. After you've made some moves, try branch swapping again.

FOR PALEONTOLOGISTS

One feature in MacClade, Make Ancestor, was included with paleontologists in mind. We offer a few ideas here on the use of MacClade by paleontologists. Paleontologists, when they link a stratigraphic sequence of fossils into a phylogenetic sequence by considering older fossils to be ancestors of younger ones, effectively link so as to minimize the amount of missing fossils that need be assumed. Placing a fossil as an ancestor decreases the number of branches in a tree, thus minimizing the number of lineages for which no evidence (fossils) have been found (this point was brought to our attention by Mike Kaulbars). Minimizing number of lineages might therefore be thought of as a sort of "stratigraphic parsimony", which may be in opposition to cladistic parsimony, if by so considering the fossils as ancestors more steps are required in the tree. The effect of considering fossils as ancestors on cladistic parsimony can be determined using Make Ancestor as described elsewhere. Another aspect of stratigraphic parsimony concerns the preference to consider older fossils more basal. Placing older fossils in a more basal position in the tree minimizes the length (in time) of missing fossil lineages. One can assess the time length of missing fossils implied by a tree by including in the data matrix a character whose states are geological ages of the taxa (0 oldest, 9 youngest) and coding the character as irreversible (the character is also best set to weight 0 so it does not contribute to the cladistic character-parsimony). The tracing of such a time character indicates the minimum ages of hypothetical ancestors. Thus two aspects of stratigraphic parsimony (branch number and time length) can be examined and balanced against cladistic character-parsimony.

LEARNING PHYLOGENETIC PRINCIPLES

Because of its graphic display and ease of use, MacClade is ideally suited for introducing phylogenetic principles in the classroom. We have found students treat the search for parsimonious trees like a video game, competing against one another for shorter trees. Of course, in the process they discover that grouping by derived similarity will gain them shorter trees. The principles of phylogenetics become clear when the implications of different trees on character evolution can be seen graphically.

One feature, "Freeze Action", has been included in MacClade specifically for teaching, while the "Algorithm display options", ladderize and flip branches commands, as well as the help facility, also have specific uses in teaching.

Freezing action

To catch MacClade "in action", which can be useful for making demonstration figures, choose Freeze Action from the Options menu. MacClade will change to a mode which allowed us to print many of the figures in this manual. In this mode, MacClade will freeze after a branch is clicked on, dragged and dropped. Shift-command-4 can then be used to print the screen, shift-command-3 to save it to a MacPaint file. To unfreeze, click the mouse button. To change back to normal mode, choose Unfreeze Action from the Options menu.

This will not allow you to catch pictures of menus pulled down, dialog boxes, etc. To do this, we suggest use of the desk accessory called "Camera" (should be available from your local Macintosh user's group or electronic bulletin board).

Algorithm display options

The algorithm display options (in the Options menu) allow you to display the workings of MacClade's algorithms used in tracing character evolution, and can be used in class demonstrations. For details see Chapter 5, "Tracing Character Evolution".

Ladderize, Flip Branches

In traditional phylogenetic trees, more "primitive" groups tend to be placed near the left edge of the tree, with more "advanced" groups placed at the right. Thus, if *Homo sapiens* is included in the tree, it is generally placed to the far right, at the pinnacle of the *Scala Naturae*.

The perceived importance of the ordering of the taxa on a tree can cloud our thinking about what is actually expressed by the diagram. The tree is intended to represent branching pattern alone, yet we often color it with an arbitrary notion of evolutionary advancement. It may be instructive to see trees arranged into another arbitrary order using the Ladderize or Flip Branches commands. For example, a standard phylogenetic diagram of the vertebrates, with *Homo sapiens* to the right, can be quickly be turned into a cladistically-identical, but very different looking tree by ladderizing the tree (with the left ladderize option selected).

Help File

The MacClade.help file included on the MacClade disk is a text-only file. You can create your own help file, or edit the existing one (perhaps with instructions for specific exercises) for teaching purposes. It must be named the same, "MacClade.help". For more information about editing the Help file see Getting help in Chapter 2.

11. PAST, PRESENT, AND FUTURE OF MACCLADE

In this section we give some notes on previous releases of MacClade, including known bugs. We also discuss possible bugs in the current version, and what to do if you inadvertently find one, and what might be some causes of crashes. Finally, we describe some of our plans for the future.

Past Versions and their bugs

Version 1.0 (by Wayne Maddison) was released on 21 June 1986. It has a bug in the consistency index calculations when a character is ordered or irreversible, and state 0 is not represented in the data matrix. In this case the CI was overestimated, because MacClade 1.0 calculated the minimum possible number of steps by using the maximum state value, instead of using (maximum-minimum), as it should. Furthermore if this problem resulted in an apparent CI of greater than 1, MacClade 1.0 miswrote it as a decimal less than one (e.g., apparent CI of 1.5 was written by MacClade 1.0 as 0.150).

Version 1.01 (by Wayne Maddison) was released on 8 November 1986; it was identical to version 1.0 except that the Consistency index bug in version 1.0 was fixed.

Versions 1.0 and 1.01 also had difficulties with blank lines in the data file. This was especially a nuisance to users who edited with MacWrite then saved so as to put carriage returns both at the end of lines and paragraphs (this effectively double-spaces the whole file).

Versions 1.0 and 1.01 had another just-discovered bug, related to making terminal taxa ancestral. If a taxon polymorphic for a character was made ancestral, and then the transformation type of this character was changed to or from irreversible, the tracing of character evolution would be incorrect.

Version 2.0 (27 February, 1987) was not formally distributed, but in the event that copies of it have escaped into general use, we here note that it had some bugs in tracing the evolution of irreversible and Dollo characters when polymorphic terminal taxa were made ancestral, and in tracing Dollo characters when states at branches were fixed.

Differences between versions 1.0 and 2.1

There are numerous additions in version 2.1 of MacClade, most strikingly the addition of user-defined and Dollo transformation types, and many advances in the user interface, including addition of a text-editor, help facility, better printing facilities, menus, dialog boxes, and generally more flexible mouse-driven commands.

In addition to these new features, some of the features present in version 1 have been altered slightly. MacClade 2.1 follows different conventions from version 1 in the following respects: (1) the branch for a polymorphic terminal taxon, as long as its state is not fixed, is always shaded by the polymorphic pattern. In version 1 the branch of polymorphic terminal taxon was shaded by the minimum state in the taxon when the character was irreversible. (2) The diagnostics table lists the number of states observed for the character, in contrast to version 1 which would list the maximum state plus one for ordered and irreversible characters. (3) Consistency index no longer includes the contribution of monomorphic characters. Two important differences in the data file are (4) the data matrix must end with a line beginning with @, and the * and $ lines are now optional, and (5) Tree names must begin with #.

Errors, bugs, and crashes with version 2.1

MacClade has a fairly thorough error reporting system, which can usually warn you if you are trying to do something illegal, for instance in the data file format. However, in addition to the errors you may cause, there may be a few errors that are MacClade's fault.

The algorithms used by MacClade to count steps (Char steps and tree length) and reconstruct character evolution have been thoroughly checked and tested on a number of data sets, though some bugs may remain in some of the more complicated cases (for instance when polymorphic taxa are fixed as ancestors, states at branches are fixed and there are missing data). More likely are bugs in screen-display and giving commands, and memory management. We would much appreciate hearing of any bugs you find. Please record as much information as you can concerning the context of any bugs (if you can, send us a copy of the data file you were using, and any notes about what you were doing when things went wrong). If the system crashes, please record the ID number of the error.

If MacClade crashes with a system error just after reading in the data file, before a tree is shown on the screen, then you may have a line in your data file longer than 255 characters. Get into MacClade's text editor and use Check Data (under Edit menu) to check if this is the problem.

If MacClade crashes with a system error ID = 25 then it is quite likely that your Macintosh has run out of memory. Running out of memory also sometimes causes MacClade to crash more spectacularly, with a flickering screen and rasping speaker. See Chapter 2 for more advice on the memory.

MacClade may crash with a system error ID = 33 if you cancel draft-quality printing of the data file by holding down the option-period keys. We're not sure why.

Plans for the future

Later releases will hopefully have:
1. Polytomies.
2. Easier data-file editing (using a built-in spreadsheet-style editor).
3. Have "Show Changes" as just a different tree-coloring mode, so that one could do branch moves, etc., while "Show Changes" was in effect
4. Loading of many trees into memory at once.
5. Exclusion of selected taxa.
6. Larger data matrices.
7. Ability to share data files with other phylogenetics programs as they become available on the Mac (eg. PAUP).
8. Some of your suggestions!

REFERENCES

Camin, J.H. & R.R. Sokal 1965. A method for deducing branching sequences in phylogeny. Evolution. 19:311-326.

Eldredge, N. & J. Cracraft 1980. Phylogenetic Patterns and the Evolutionary Process. Columbia U. Press, New York.

Farris, J.S. 1970. Methods for computing Wagner trees. Syst. Zool. 19: 83-92.

Farris, J.S. 1977. Phylogenetic analysis under Dollo's Law. Syst. Zool. 26:77-88.

Felsenstein, J. 1985. Phylogenies and the comparative method. American Naturalist 125: 1-15.

Fitch, W.M. 1971. Toward defining the course of evolution: Minimum change for a specific tree topology. Syst. Zool. 20: 406-416.

Hartigan, J.A. 1973. Minimum mutation fits to a given tree. Biometrics 29: 53-65.

Hendy, M.D. & D. Penny 1984. Cladograms should be called trees. Syst. Zool. 33: 245-247.

Kluge, A.G. & J.S. Farris 1969. Quantitative phyletics and the evolution of the anurans. Syst. Zool. 18: 1-32.

Maddison, W.P., M. Donoghue, & D.R. Maddison 1984. Outgroup analysis and parsimony. Syst. Zool. 33: 83-103.

Meacham, C.A. 1984. The role of hypothesized direction of characters in the estimation of evolutionary history. Taxon 33: 26-38.

Sankoff, D. and P. Rousseau. 1975. Locating the vertices of a Steiner tree in an arbitrary metric space. Mathematical Programming 9: 240-246.

Swofford, D.L. & W.P. Maddison. 1987. Reconstructing ancestral character states under Wagner parsimony. Math. Biosci. 87: 199-229.

APPENDIX 1.
MACCLADE'S ALGORITHMS FOR TRACING CHARACTER EVOLUTION

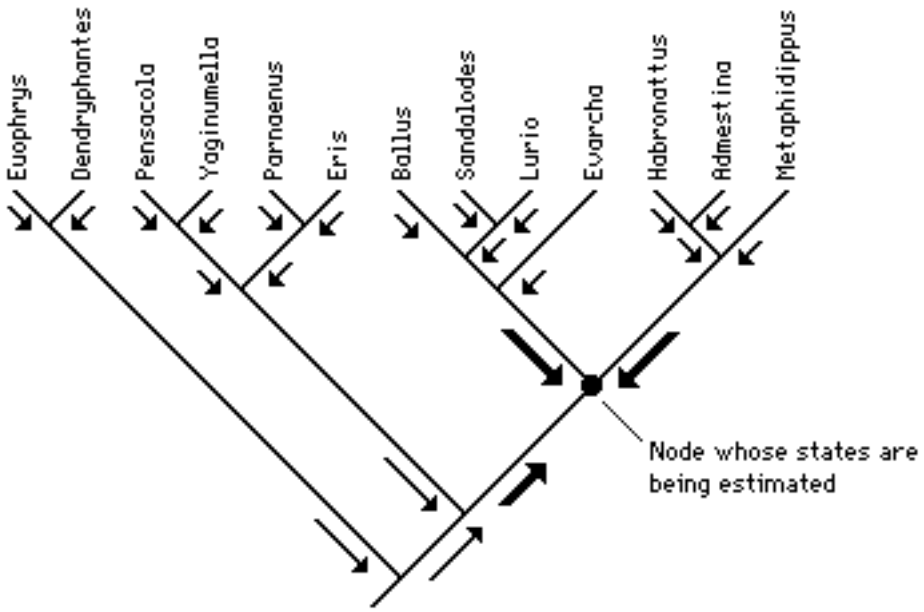
The problem of reconstructing the character states of ancestors using the principle of parsimony can be phrased as the following question: given the form of the phylogenetic tree, and given the states observed in the terminal taxa, what assignments of states to the internal nodes of the tree require the fewest evolutionary steps? In some cases it may be intuitively obvious how states can be most parsimoniously assigned to ancestors. For instance if all the extant members of one clade share state 1, then it is simplest to suppose all of the ancestral members of the clade also had state 1. In other cases it may not be so easy.

Thus a number of algorithms have been developed to reconstruct ancestral states on a tree. The different algorithms have been written to handle different sets of restrictive assumptions they impose on character evolution. For instance, the algorithm of Farris (1970), completed by Swofford and Maddison (1987), reconstructs character evolution under the assumption that states are linearly ordered.

In the following descriptions we will imagine that we are assigning states to the nodes of the cladogram. The nodes are the branching points; between them are the internodes or edges. Graphically, MacClade colors a node and the internode beneath it together as a single unit, though of course a change between one node and the next could have occurred anywhere along the internode between them.

The algorithms of MacClade are exact in that they find (barring bugs) all and only the most parsimonious assignments of character states that can be made to every node of the tree. The only exceptions known are for user-defined type characters in which some taxa are polymorphic with more than 2 states, and for Dollo and irreversible characters in which polymorphic terminal taxa are made ancestral. The problems in the former case are discussed below.

Each of these algorithms works in a similar way. Each moves through the tree, distilling information as it goes, in order to arrive at estimates for the states at each node. Information from above and below the node is distilled and combined to yield the estimate. Thus the state estimated for a node will depend on the information distilled (1) down through the clade of the node's left descendant, (2) down through the right descendant's clade, and (3) up from the part of the tree below and beside the node. These three sources of information converge at the node and indicate its state:



In practice this distilling process is done in two or three passes up and down the tree. The details of the process differ with the different assumptions used. The following description will consider first unordered and ordered characters.

Unordered & ordered

The algorithm first makes a pass down the tree from the terminal taxa to the root (we will call it the DownPass). As it proceeds down the tree, the algorithm visits each node and calculates a set of states that can be appropriately assigned to that node. To calculate this set of states, the algorithm uses the sets already calculated from the two nodes immediately above (descendant from) the node being visited. The unordered algorithm uses the following calculation: a node is assigned the set of states which are present in both state sets of the two descendant nodes if there are any such; if no states are present in both sets above, the node is assigned the union of the sets of the two descendant nodes. The ordered algorithm uses the following calculation: a node is assigned the region of overlap of the state sets of the two descendant nodes if the sets overlap; if there is no overlap, the node is assigned the states in the space in between the sets. Once the state set has been calculated for the node being visited, the algorithm can then move down the tree to the node's immediate ancestor and perform the calculation on it. The algorithm continues down until it eventually arrives at the root. The Downpass states option of MacClade displays the states assigned to the tree by this first pass of the algorithm.

The states assigned to a node on the first Downpass of the algorithm are not necessarily the most parsimonious assignments to the node according to the whole tree. When we arrive at a node on the down pass, only the nodes descendant from it have been used in the calculation of the visited node's states. Thus, the states assigned on the Downpass are most parsimonious according to the clade above it, but may not be most parsimonious according to the whole tree since the state calculations have not yet considered nodes below the visited node. There is one exception to this. By the time the algorithm gets to the root, it has considered the entire tree, and the states assigned to the root are indeed the most parsimonious states according to the whole tree.

We want to eventually know, for every node, the set of states that can be most parsimoniously assigned to it according the whole tree. Because the Downpass algorithm guarantees whole-tree-parsimony for the states assigned to the root, one way to achieve our goal

would be to reroot the tree at each node, each time performing the Downpass algorithm ending up at the new root, and thus finding the whole-tree-parsimony state sets for each node. Effectively, MacClade does this, but there is a much more efficient way to do this than actually rerooting the tree and performing a whole new Downpass each time (Swofford & Maddison, in press).

After doing a single Downpass, MacClade does another pass going up the tree (the UpPass). On this pass it uses the same calculations, but instead it calculates the state set of a node by the state sets of the node below it (its ancestor) and beside it (its sister node), using the same calculation for combining state sets as for the Downpass. MacClade's Uppass option shows the states calculated on the Uppass. If a node is shaded black this means that the node is most parsimoniously assigned this state according to the part of the tree below this node, consisting of the node's immediate ancestor, the node's sister node, and parts further away. One possibly-confusing convention in MacClade is that the Uppass state for a node actually records the opinion of the node's ancestor (not the node itself) and the part of the tree it represents.

Once this second pass has been completed, we now have information distilled from all parts of the tree surrounding a node. For each node we know the opinion of its right descendant clade as to what states the node should take (from the Downpass), we know the opinion of its left descendant clade (also from the Downpass), and we know the opinion from its ancestor (from the Uppass, representing the rest of the tree below and beside the node). These three opinions are then combined in a calculation that results in the final state set for the node. This calculation condenses three state sets into one. For unordered characters, it chooses those states occurring in the greatest number of the three sets. For ordered characters, it choses the region of triple overlap, if there is any, or otherwise it does the two-state calculation on the two most separate of the three sets, then takes the result and combines it with the third set.

Algorithms for unordered characters were first developed and proved by Fitch (1971) and Hartigan (1973). Farris's (1970) algorithm for ordered characters found only some of the most parsimonious assignments; it was completed and proved by Swofford and Maddison (1987). The ordered algorithm in MacClade is basically that of Swofford and Maddison.

Irreversible characters

For these a single pass down the tree is sufficient unless there are missing data. On this pass down the tree the state set of a node is calculated as the minimum value in the state sets of the two descendant nodes.

Camin and Sokal (1965) have discussed the use of irreversible characters. That part of the algorithm in MacClade which handles missing data is original to MacClade.

Dollo characters

For these two passes are used, one down and one up. On the down pass the state set of a node is calculated as the <u>maximum</u> value in the state sets of the two descendant nodes. On the second pass back up the tree, the final state set is assigned to the root as the <u>minimum</u> value in the Downpass state sets of its two descendant nodes. The algorithm continues to move up the tree, assigning to any node that Downpass state assigned to its two descendants the one which is closest to the state already assigned to its ancestor.

An algorithm for two-state Dollo characters was presented by Farris (1977). The algorithm in MacClade is different but basically equivalent, though it allows multistate Dollo characters. That part of the algorithm in MacClade which handles missing data is original to MacClade.

User-defined types

The algorithm for user-defined types follows a similar scheme, but differs in that assigned to each node is not a set of states, but rather a number for each possible state. This number records how many steps are required in a section of the tree represented by the node, given that the particular state is assigned to the node. On the Downpass, the number records how many steps are required in the clade above the node; on the Uppass the number records how many steps are required in the area of the tree below and beside the node. Thus for each possible state that could be assigned to a node, there is a graded scale of goodness, in contrast to the other algorithms in which a state is either included or excluded in a state set as a viable candidate. The algorithm's flexibility, as well as its slowness and other limitations, are due to its keeping records on a graded scale.

The algorithm likewise makes a Downpass and an Uppass, so that after these passes one has recorded the opinions from the parts of the tree represented by its left and right descendants and its ancestor. On the Downpass the record for a node N is calculated as follows. Recall that the information to be recorded for N consists of: for each possible state, the number of steps required in N's clade given that N is assigned this state. So, the algorithm tries assigning each possible state S to N, and for each the minimum number of steps on the left side (N to its left descendant node NLF and above) and right side (N to its right descendant node NRT and above) are added to yield the number of steps required in total in N's clade. The minimum number of steps on the left side, for instance, is calculated by finding the state T assigned to NLF that yields the lowest sum of steps between N (having S) and NLF (having T) plus the number of steps required in the clade of NLF given NLF is assigned T. This latter number of steps is available from the information already calculated for NLF. Thus the records of the two nodes NLF and NRT are combined to yield the information for N. This basic pattern of calculation is followed on the Uppass and in the final decision for any given node in which the information for three nodes is combined. After this is done, the states actually assigned to a node are those which require the minimum number of steps according to the whole tree.

All of these calculations use the user-input matrix of the distance from one state to another. When the user puts a '∞' in the matrix to indicate the change is prohibited, MacClade stores the distance as 999, which is sufficiently large for 10 state characters on cladograms of 80 taxa that the change is effectively prohibited. If MacClade indicates that a character requires more than 1000 steps on the cladogram, then you must have created impossible assumptions, for instance by constructing a distance matrix prohibiting any changes.

The algorithm is basically that of Sankoff and Rousseau (1975) modified for discrete characters, though the algorithm in MacClade expands upon their treatment by allowing asymmetries in distance for gains and losses.

Calculating lengths

So far described have been the algorithms for reconstructing ancestral states. Counting the number of steps required for a given character could be done after a reconstruction has been completed by cycling back up through the tree and counting how many changes are in the reconstruction. In the case of the Dollo characters, this is how MacClade counts length (although with fancy enough record taking on the Downpass it needn't do this). But in general there are shortcuts to counting length. For unordered, ordered and irreversible characters the number of steps can be accumulated on the Downpass. Whenever two state sets that don't overlap are combined, the distance between them is added into a counter, and by the time the algorithm gets to the root the total length required for the character has been added up. For characters of user-defined types, the records kept for each node directly contain information about length, and by the time you get to the root of the tree you have the minimum number of steps required for the tree.

Note that the reconstruction of ancestral states and the length calculated is independent of where the tree is rooted for unordered, ordered, reversible and Dollo characters.

Missing data

Missing data are generally easy to deal with in cases except for irreversible and Dollo, when special attention needs to be kept on the pass back up the tree to find when missing data causes assignments to be equivocal. We will not describe the details here. For unordered, ordered, and user-defined character types, the algorithms initially assign to taxa with missing data information that acts like an identity element -- i.e. which will leave information from other nodes untouched during the calculations.

Polymorphisms

Terminal taxa coded as having more than one state must be considered in a special way both for counting steps and reconstructing character evolution. In general, the algorithms count the minimum number of steps which must have occurred within these terminal taxa and add this to the count of steps for the character. Thus 4 steps are presumed to have occurred within a terminal taxon polymorphic for states 2, 5 and 6 in an ordered character. Similar calculations can be done for unordered, irreversible and Dollo characters, but for characters of user-defined type it is very difficult to count the minimum number of steps which must have occurred within a terminal taxon if it is polymorphic for more than 2 states. Thus the number of steps due to a terminal taxon's being polymorphic is not added to the count of steps for the character if the taxon has more than two states. In these situations MacClade puts a '+' beside the count of steps and tree length to indicate that there is some uncounted length.

When MacClade shades a branch of a polymorphic terminal taxon, it uses the polymorphic pattern and indicates the states or range of states observed in the taxon. However, for the purposes of calculations on the Downpass, a sometimes-different set of states needs to be assigned to these terminal nodes. For unordered and ordered characters, the whole set or range of states observed in the taxon is assigned. For irreversible characters, a terminal node is assigned the minimum value observed in the taxon. For Dollo characters, the node is assigned on the Down Pass the maximum observed state and on the Final Pass the closest state to its ancestor's final state. For user-defined type characters the record of number of steps required by each state assignment is set to 0 <u>unless</u>: the state is not observed in the terminal taxon, in which case it is set to 999; the state is of distance ∞ from one of the observed states, in which case the number is set to 999; the terminal taxon is polymorphic with two states, in which case for each of the states the number is set to the distance to the other state.

APPENDIX 2
USING A MACINTOSH

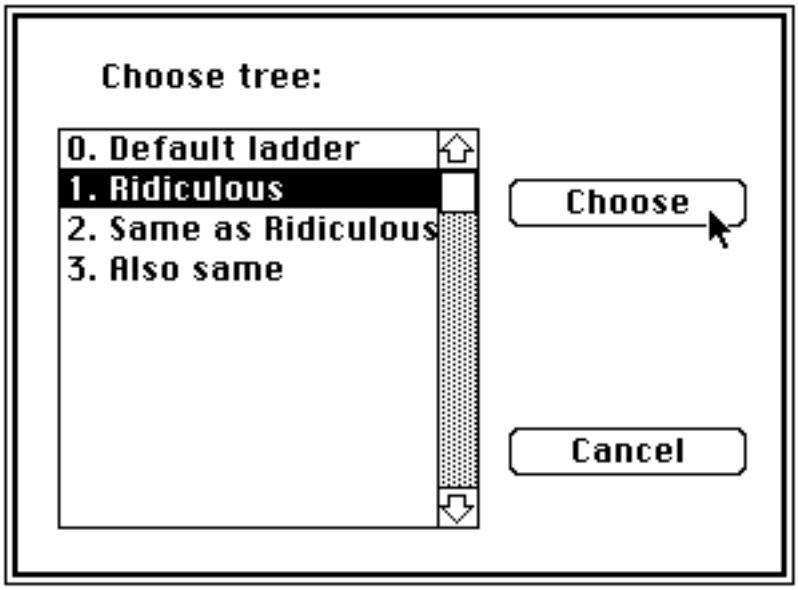
After you turn on the Macintosh and insert the MacClade disk, the disk will spin and eventually you will be presented with a grey screen containing the white bar of menus on top, at left a window displaying the files on the disk, and a trash can at lower right. In the window the files are represented by icons. The files on MacClade Disk are: MacClade, MacClade.help, some example data files (contained in the folder Data files), and various operating system files (contained in the folder System Folder). This screen is presented to you by the Finder, a program which is for the Macintosh a bit like a town square or central meeting place. From it you can start programs, copy files and disks, and so on.

Several words having special meanings should be explained. To click is to push the mouse button down; to drag is to click the mouse button down, and while holding it down, drag the mouse to the appropriate point, then let go of the mouse button. A double-click is done by clicking on the mouse button twice in close succession. To close a window you click in the little white box in the upper left-hand corner of its frame. If you select an icon, a window or some text, the object(s) selected is(are) ready to be acted upon; in general, any object on the Macintosh screen that has its image inverted (it is white on black rather than black on white) is selected. You select an object by clicking on it with the mouse. Selected objects can then be opened, deleted, changed or whatever is appropriate for that sort of object. "Open" means start or examine. Thus if you open a program, the program starts running. If you open a disk icon, a window appears showing the disks contents.

Commands in menus are chosen by the following procedure: click the mouse pointer down on the title of the menu in the menu bar, and hold the button down; the menu should drop down like a window shade and remain down; next, drag the mouse down until the command is selected, and let go of the button.

To open a file, folder, or disk, you can either (1) select its icon by moving the arrow cursor above it and clicking the mouse, and choosing "Open" from the "File" menu, or (2) double-clicking on its icon.

In several places within MacClade, you will be presented with dialog boxes that look vaguely like this:



These dialog boxes are a means for you to choose among several entities; in the example pictured, you are to choose one of the 4 trees listed. To choose one of the trees, select it by clicking once on its name (in the example the tree "Ridiculous" has been selected), and then click once on the Choose button (another way to accomplish the same thing is to double click on the name of the tree). In this case, the tree will then be displayed on the screen. You will encounter other dialog boxes within MacClade that work in a similar way; these will be for choosing files to open, choosing character weights, etc.

An introduction to the basic principles of editing on a Macintosh is given in Chapter 8.

For more information, please refer to a Macintosh manual.

APPENDIX 3
MACCLADE DATA FILES CREATED WITH OTHER PROGRAMS

MacClade data files are stored as simple text files (ASCII text files). Thus simple text files made by other text editors, such as MacWrite or Microsoft Word, can be understood by MacClade. The MacClade data files can also be created using one of two popular spreadsheet programs (Microsoft Excel or Multiplan). This section outlines the method used to create data files using programs other than MacClade.

Note that if you use another program, and wish to have male and female symbols appear on your MacClade screen, you must type option-m and option-f for these, respectively, even though they will not appear as male and female symbols in the other program.

USING OTHER WORD PROCESSING PROGRAMS

There are numerous word processing programs available for the Macintosh; two popular ones are MacWrite and Microsoft Word. If you use these or another program to write the data files, make sure you save the files as Text Only. You can do this by selecting the Text Only button in the dialog box that comes on the screen after you choose Save As from the File menu. If the files are not saved as Text Only, MacClade will not even recognize that they exist. It will be useful to use the Monaco 9 type face when editing the files because this typeface allows the characters to line up. The general text editor program Edit is useful for it naturally saves all files as text only and use Monaco 9.

USING MULTIPLAN OR EXCEL

MacClade automatically converts spreadsheet files made using the Microsoft spreadsheet programs Multiplan and Excel into MacClade data files, thus allowing you to use the special matrix-handling capabilities of these programs to conveniently edit your data. These spreadsheet programs allow you to insert and delete columns, and put character names at the tops of columns.

For MacClade to be able to read the Multiplan or Excel file, it must be saved as a SYLK file (choose SYLK in the Save As dialog box when you make your file in Multiplan or Excel). When MacClade starts to read a SYLK file, it recognizes it as such and translates it into a MacClade data file of the approprate format. This newly made MacClade data file has the same name as the original SYLK file but with ".mcl" added (or ".mcl2" added if ".mcl" already exists, and so on). After translating the SYLK file and storing the new MacClade data file on disk, MacClade then opens the new MacClade data file to begin your analysis.

When making the data file in Multiplan or Excel these rules must be followed: in the first cell (first column, first row) the title of the data must appear; in the first column the taxon names must appear; in the first row the character names must appear. The following figure of a small portion of an Excel file illustrates this:

Amblygnathus

	1	2	3	4	5
1	Amblygnathus	m.l. vent. flange	elyt. micro.	SBL	pron. hind ang.
2	seriatoporus-g	1	0	2	0
3	ellipticus-g	1	0	0	1
4	amaroides-g	1	1	0	1
5	ruficollis-g	0	1	0	2
6	iripennis	0	1	0	1
7	mexicanus	0	1	0	0
8	interior	0	1	0	0
9	evansi	0	1	0	0
10	panamensis	0	1	0	1
11	fricki	0	1	0	1
12	woodruffi	0	1	0	1
13	puncticollis	0	1	0	0
14	tikal	0	1	0	2

Thus, each row (below the first), corresponds to a single taxon, and each column (after the first) corresponds to a character. The first taxon name must not include a number and the character names must not include colons (:). The data themselves must be input as integers for observed character states or ? for missing data. Single digit integers (from 0-9) indicate a monomorphic taxon; multiple digit integers indicate polymorphisms. Thus 3 indicates the taxon has just state 3; 10 indicates it has states 1 and 0, while 201 indicates it has states 2, 0 and 1. Polymorphisms of more than 5 states are not allowed: 10736 is allowed (5 digits) but 216304 (6 digits) is not. Do not leave any cells blank! - use ? for missing data. If there are 50 taxa and 30 characters, then every cell must be filled in the upper left 50x30 block of the spreadsheet. The SYLK conversion facility in MacClade may not work if you use fancy formatting or cell protection in your spreadsheet, so be simple. You may use Freeze Titles.

There are some disadvantages using Multiplan and Excel: character state names, tree data, character weights, and transformation type definitions and assignments cannot be included in the spreadsheet. You may find it most convenient to construct the initial data matrix in Multiplan or Excel, translate it using MacClade, then edit it further within MacClade to add the character state names and so on. Note that if you make a Multiplan file named, for instance, Example, then use MacClade to translate it into a MacClade data file Example.mcl, then you modify Example and translate it once again, it will translated into a separate MacClade data file (which might be called Example.mcl2). Thus, if you have worked with the first Example.mcl, saved trees in it, edited it using a word processor, and so forth, those trees and other changes will not be transferred to the new MacClade file.

Multiplan constructs its SYLK files in a strange way when rows are copied or cut and pasted. If you select a row (taxon), copy or cut it and then paste it somewhere else in order to duplicate or move it, MacClade will not read the data matrix correctly (it will give an error that all characters are miscoded and the tree will appear with some extra taxa with strange names). Therefore, please

type in entries into Multiplan for each taxon by hand. These problems with copy or cut and paste do not occur with Excel as far as testing has shown. Thus if you use Excel, feel free to paste rows and columns!

USING FILES CREATED BY OTHER PROGRAMS

If you have created a file by another program and now wish to use it in MacClade, you cannot simply double click on the file's icon in the Finder and expect MacClade to start up and use the file. The reason for this is that the Macintosh thinks the file belongs to that other program, and will attempt to start up that other program instead of MacClade. Therefore, you have to go into the house and let the file in through the back door, so to speak:

•To *use an already existing data file created by a program other than MacClade*, double-click on the MacClade icon:

and then choose "Open File" from the File menu; MacClade will present you with a dialog box presenting choices of files to open. If the name of that file does not appear in the listing in the dialog box, then the file was improperly saved earlier. Remember, the file must be saved as a TEXT-ONLY file (if created with a word-processor) or as a SYLK file (if created with a spreadsheet); otherwise, MacClade will not even know the file exists.