

# **MESQUITE 2**

**Wayne Maddison  
&  
David Maddison**

<http://mesquiteproject.org>

Copyright ©2007 W. Maddison & D. Maddison

This Mesquite manual is current as of version 2.0, October 2007.  
It may not be the most current version; please check the web site at:  
<http://mesquiteproject.org>

Also, this manual covers only the core modules of Mesquite.  
It does not cover separately distributed packages such as PDAP,  
Cartographer, TSV, Stratigraphic Tools, CoMET, and so on.

# Mesquite: A modular system for evolutionary analysis, version 2.0

## **Principal Authors:**

Wayne P. Maddison, University of British Columbia

David R. Maddison, University of Arizona

## **Development Team:**

Wayne Maddison

David Maddison

Peter Midford (2006-)

Danny Mandel (2006-7)

Jeff Oliver (2007-)

This manual introduces you to the Mesquite system for analysis in evolutionary biology. It explains the basic operation of the system (the links above) and some standard analyses (the links at left).

New features and bug fixes since version 1.0 are described on the page on [new features and history of changes](#). Version 2 is incompatible with modules from version 1, and thus you will need to find new versions of any third-party modules you have from version 1.

Mesquite's analyses currently include phylogenetic analyses (parsimony, likelihood, comparative method, simulations and randomizations of characters and trees) and population genetics analyses (coalescence). You can also use Mesquite as an editor for phylogenetic data files. Mesquite is not primarily designed to infer phylogenetic trees, but rather for diverse analyses using already inferred trees. Mesquite's web site contains an outline of [Mesquite's features](#).

Because many analyses are possible and those available depend on what modules are installed and loaded, a comprehensive manual cannot easily be written. For this reason, many analyses available in Mesquite are not included in this manual. You are encouraged to explore, and invent new analyses. We have a special page on [how to learn](#) about Mesquite's features.

Mesquite's web site is [here](#); check it for updates. We recommend you sign up to the [Mesquite discussion e-mail list](#) to learn about updates and to discuss features.

## **Getting started**

To begin using Mesquite on the Mac OS, Unix/Linux or Windows, see the [installation instructions](#). We encourage you to explore the example files in the "examples" directory of the Mesquite\_Folder.

See the [Help](#) page to learn how best to learn Mesquite.

The website has a list of some [third-party modules](#) you can install. See the page on [modules](#) for instructions on where they can be installed.

## Reporting Bugs

We encourage you to report bugs or misbehaviour of Mesquite. You may find a problem that seems so obvious to you that you expect we must have seen it. Please report it anyway, because perhaps it occurs only in particular circumstances that we didn't test, or it occurs with a combination of options that we haven't tested recently. The [Support](#) page has more details on reporting bugs.

## Conventions of this manual

For compactness, we will use a special convention to refer to menu items. For example, [File>Save File](#) refers to the Save File menu item of the File menu. [\(Tree Window\)Drawing>Tree Form>Diagonal Tree](#) refers to the Diagonal Tree menu item of the Tree Form submenu of the Drawing menu that is associated with the Tree Window. By referring to a menu as "associated with" a window, we mean that that menu is present in the menu bar at the top of the screen when the window is frontmost (on the MacOS), or that the menu is embedded within the window (on the Windows OS and most other operating systems).

## Information for Developers

Mesquite is modular, and can be extended by adding modules into the folder "mesquite" in the folder "Mesquite\_Folder". Developers interested in writing modules should contact us at [develop@mesquiteproject.org](mailto:develop@mesquiteproject.org). Information on development for Mesquite is also available [here](#), although it is out of date.

---

# Mesquite Manual Table of Contents

Click on a topic to expand its contents [[Expand All](#) / [Collapse All](#)]

## Part I: Mesquite Basics

- [Introduction](#)
- [Learning how to Use Mesquite](#)
- [Why Mesquite was made](#)
  - [Overview](#)
  - [Practical Answer](#)
  - [Poetic Answer](#)
  - [Mesquite versus MacClade](#)
- [How Mesquite works](#)
  - [Overview](#)
  - [The tree of modules](#)
  - [Duties](#)
  - [Flexibility](#)
- [Installation](#)
  - [Mac OS X](#)
  - [Mac OS 9](#)
  - [Linux/Unix](#)
  - [Windows](#)
- [Basic concepts and vocabulary](#)
  - [Taxa](#)
  - [Characters](#)
  - [Trees](#)
- [Files](#)
  - [Making, opening, and saving data files](#)
  - [Projects](#)
  - [Opening versus including versus linking files](#)
- [Menus](#)
  - [Overview](#)
  - [Learning about menus](#)
  - [Standard menu items](#)
    - [File](#)
    - [Edit](#)
    - [Characters](#)
    - [Taxa & Trees](#)
    - [Analysis](#)
    - [Window](#)
    - [Help](#)

- [Where to find menu items](#)
- [Mesquite's Windows](#)
  - [Overview](#)
  - [Common features of windows](#)
    - [Overview](#)
    - [Information bar](#)
    - [Mini scrolls](#)
    - [Tool palettes](#)
    - [Annotation area](#)
    - [Explanation area](#)
    - [List windows](#)
- [Charts](#)
  - [Overview](#)
  - [Selection of objects in charts](#)
  - [Auto-recalculation](#)
  - [Bar & Line Charts](#)
    - [Overview](#)
    - [Example: Robustness of estimated bias in character evolution](#)
    - [Example: Compositional bias along a sequence](#)
    - [Calculating and formatting options](#)
  - [Scattergrams](#)
    - [Overview](#)
    - [Example: Canonical Variates Analysis](#)
    - [Example: Correlation between variability and hydrophobicity](#)
    - [Calculating and formatting options](#)
- [Scripting and Macros](#)
  - [Overview](#)
  - [Macros](#)
  - [Learning about scripting](#)
  - [Commands within scripts](#)
  - [Scripts](#)
  - [Variables: Integers, Strings, and Objects](#)
    - [Overview](#)
    - [Numerical variables](#)
    - [String variables](#)
    - [Object variables](#)
    - [The variable "It"](#)
  - [Flow and command control](#)
  - [Debugging and technical details](#)
- [Modules](#)
  - [Managing Modules](#)
  - [Installed versus loaded versus running modules](#)
  - [Where modules are installed](#)
  - [Modules currently in use](#)
  - [Information pages and Manuals for modules](#)

## Part II: Characters, Taxa, and Trees

- [Characters](#)
  - [Overview](#)
  - [Creating a character matrix](#)
  - [Deleting and renaming character matrices](#)
  - [The Character Matrix Editor](#)
    - [Overview](#)
    - [Adding, deleting, renaming, and sorting taxa and characters](#)
    - [Entering character data](#)
    - [Selecting taxa, characters and cells of the matrix](#)
    - [Copy/paste](#)
    - [Editing Names of Characters and States](#)
    - [Annotations](#)
    - [Coloring cells of the matrix](#)
    - [Alterations and Transformations](#)
  - [Changing the attributes of characters](#)
  - [Charting information about Characters](#)
- [Taxa](#)
  - [Overview](#)
  - [Creating and managing taxa blocks](#)
  - [Managing taxa](#)
  - [Renaming taxa](#)
  - [Assigning group membership](#)
  - [Associating two taxa blocks](#)
- [Trees](#)
  - [Overview](#)
  - [Characteristics of trees](#)
  - [Analyzing trees](#)
    - [Overview](#)
    - [Tree to tree similarity measures](#)
    - [Consensus trees](#)
    - [Fits of trees into trees](#)
    - [Taxon instability among trees](#)
  - [Stored trees and tree blocks](#)
  - [Tree sources](#)
  - [Filtering trees](#)
  - [Modifying trees automatically](#)
  - [Editing trees](#)
    - [The Basic Tree Window](#)
    - [Tree source](#)
    - [Moving from Tree to Tree](#)
    - [Storing trees](#)
    - [Tools](#)

- [Alter/Transform menu items](#)
- [Analyses](#)
- [How trees are drawn](#)
- [Printing trees and saving graphics files](#)
- [Dependent Tree Window](#)
- [Mirror Tree Window](#)
- [Multi Tree Window](#)

## Part III: Analyses

- [Character Evolution](#)
  - [Overview](#)
  - [Visualization of Character Evolution](#)
    - [Reconstruction of Ancestral States](#)
      - [Overview](#)
      - [Tracing the history of a character](#)
      - [Tracing all characters](#)
      - [Tracing character over multiple trees](#)
    - [Reconstruction Methods](#)
      - [Parsimony](#)
        - [Parimony models](#)
        - [Assigning a model](#)
        - [Creating & editing stepmatrices](#)
      - [Likelihood](#)
        - [Overview](#)
        - [Likelihood models](#)
        - [Making, editing, and applying probability models](#)
        - [Optimization settings](#)
        - [Reporting of results](#)
      - [Stochastic Character Mapping](#)
    - [Comparisons with other programs](#)
      - [NEXUS-based programs](#)
      - [MacClade](#)
      - [Pagel's Discrete & Multistate](#)
    - [Acknowledgements](#)
    - [References](#)
  - [Analyses of Character Evolution](#)
    - [Processes of character evolution](#)
    - [Estimating parameters for single characters](#)
    - [Correlations among characters](#)
      - [Visualizations](#)
        - [Mirror tree window](#)
        - [Plot tree 2D](#)
        - [Plot tree 3D](#)

- [Taxa scattergram](#)
- [Pairwise comparisons](#)
- [Felsenstein's Independent Contrasts](#)
- [Pagel's Correlation Method](#)
- [Use with Pagel's Discrete & Multistate](#)
- [References](#)
- [Diversification](#)
  - [Simulating speciation and extinction](#)
  - [Lineages through time](#)
  - [Speciation and extinction rates](#)
  - [Character-dependent Speciation and extinction rates](#)
- [Simulations & Randomizations](#)
  - [Overview](#)
  - [Simulating and randomizing characters](#)
    - [Character Simulations and Randomizations for statistical tests](#)
      - [Overview](#)
      - [Using results of simulations & randomizations](#)
      - [Simulations of character evolution](#)
      - [Viewing results of simulations](#)
      - [Randomizing characters](#)
    - [Simulating DNA sequence Evolution, with examples of hypothesis testing](#)
      - [Overview](#)
      - [Simulating DNA sequence evolution on a tree](#)
      - [Building more complex models](#)
      - [Batch Architect: Simulating, exporting, and analyzing multiple matrices](#)
      - [Examples of tests of phylogenetic hypotheses](#)
        - [Example I: Testing monophyly of a group of beetles](#)
        - [Example II: Are strepsipterans related to flies?](#)
    - [Simulating coalescence and sequence evolution within populations](#)
      - [Introduction](#)
      - [Importing and Exporting Data](#)
      - [Single populations](#)
        - [Coalescence Simulations](#)
        - [Example: Chart of coalescence depths](#)
      - [Multiple populations or species](#)
        - [Analyzing gene trees for multiple populations or species](#)
        - [Establishing the association between genes and species/populations](#)
        - [Editing already-created taxa associations](#)
        - [Simulating coalescence within a species/population tree](#)
        - [Reconstructing gene history within a species/population tree](#)
        - [Visualizing gene history in a species/population tree](#)
        - [Measuring fit between genes and species/population trees](#)
          - [s of Slatkin & Maddison](#)

- [Deep coalescences](#)
- [Gene duplications/extinctions](#)
- [Example: Effect of population divergence time on species tree](#)
- [Inferring the population or species tree](#)
  - [Tree search](#)
  - [Cluster analysis](#)
- [Simulating sampled gene sequences](#)
  - [Introduction](#)
  - [Generating a single matrix of sequences](#)
  - [Generating a series of matrices](#)
  - [Example: Multiple simulations of sequence samples](#)
  - [References](#)
- [Simulating and randomizing trees, including species trees and gene trees](#)
  - [Introduction](#)
  - [Using results of simulations & randomizations](#)
  - [Evolutionary Simulations of Trees](#)
    - [Species Trees](#)
    - [Gene Trees](#)
  - [Random Trees](#)
  - [Randomized Trees](#)
- [Molecular Data](#)
  - [Introduction](#)
  - [Editing molecular data](#)
    - [Format](#)
    - [Alter/Transform Tools](#)
    - [Alignment](#)
    - [Finding sequences](#)
    - [Display of sequences](#)
  - [Managing molecular matrices](#)
    - [Overview](#)
    - [Include Taxa from File](#)
    - [Fused Matrix Export](#)
    - [Archiving Taxon Names](#)
    - [Translation to Protein](#)
  - [Interactions with GenBank](#)
  - [Simulating DNA sequence evolution](#)
  - [Statistics for DNA sequences](#)
    - [Overview](#)
    - [ACGT Compositional bias](#)
    - [Character Compositional bias](#)
    - [GC bias coloring of matrices](#)
  - [DNA Distances](#)
  - [Statistics for Protein sequences](#)
    - [Site hydrophobicity](#)
    - [Amino Acid hydrophobicity](#)

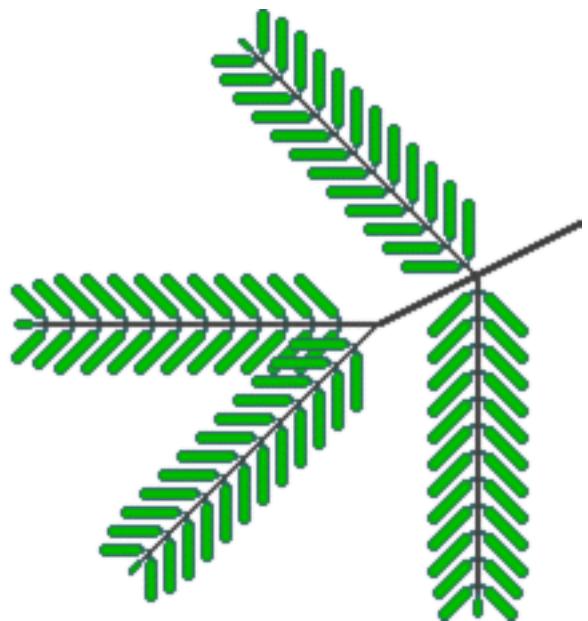
- [Visualizing protein tertiary structure](#)
- [Sequence Data within populations](#)
- [Reconstructing ancestral states](#)
- [Continuous Data](#)
  - [Continuous characters overview](#)
  - [Editing continuous data](#)
  - [Reconstructing ancestral states](#)
  - [Plotting trees](#)
  - [Simulating character evolution](#)
  - [Ordinations](#)
  - [Tree reconstruction](#)
  - [Felsenstein's independent contrasts](#)
  - [Geometric morphometrics](#)

## Part IV: Miscellaneous Debris

- [Interactions with other programs](#)
  - [Why?](#)
  - [Importing/exporting files](#)
    - [Overview](#)
    - [MacClade](#)
    - [PAUP](#)
    - [NDE](#)
    - [CLUSTAL](#)
  - [MrBayes interactions](#)
  - [CLUSTAL \(broken link?\)](#)
- [Publishing Results](#)
  - [Important information](#)
  - [How to cite Mesquite?](#)
    - [General](#)
    - [Analyses](#)
    - [What to cite](#)
- [Support & Advice](#)
  - [If you think you've found a bug...](#)
  - [Suggesting new features](#)
  - [Why is this bug here?](#)
  - [Help system](#)
  - [Reporting bugs & requesting advice](#)
  - [Mesquite Listserv](#)
  - [Java bugs that affect Mesquite](#)
- [Programming in Mesquite](#)
  - [Mesquite source code](#)
  - [Obtaining Mesquite source code](#)
  - [Using Eclipse for Mesquite development](#)

- [Installing Eclipse](#)
  - [Installing Subclipse subversion plugin](#)
  - [Subversion checkout of Mesquite source](#)
  - [Creating a Mesquite launch configuration](#)
  - [Developing your own modules](#)
  - [Contributing changes to the official Mesquite code](#)
  - [Mesquite development mailing list](#)
  - [Development Reference](#)
  - [Help](#)
    - [Why Mesquite is complex](#)
    - [Documentation](#)
    - [How to remember or document what you've done](#)
    - [MesquiteFAQ](#)
    - [Programming in Mesquite](#)
    - [Two Examples:](#)
      - [Example I: Testing monophyly of a group of beetles](#)
      - [Example II: Are strepsipterans related to flies?](#)
  - [Credits & Acknowledgments](#)
-

# MESQUITE 2



*Basics*

Search Online Manual

# Mesquite Help: Learning how to use Mesquite

## Table of contents

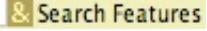
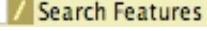
- [Documentation and help facilities](#)
- [How to remember what you've done](#)
- [Why is Mesquite complex? \(Understanding this may help you know what to do...\)](#)

## Mesquite's documentation and other learning aids

Because of Mesquite's complexity we have provided various aids to help you discover and learn features:

**Manual:** This manual provides an introduction to Mesquite, including instructions for accomplishing some analyses. It resides in the **docs** subdirectory of the **Mesquite\_Folder** directory, and is also available from the Help menu while Mesquite is running.

**Search Features:** Most windows have a small text box (usually near the top of the window) beside the words "Search Features" or "Search Data". When set to **Search Features** mode (which you can request by clicking on the little symbol until it shows "&" or "/"), you can use this to find features and how to use them. Enter text there and hit Return, and the text will be sought. There are several modes, marked by different symbols:

-  **& Search Features** — Search Features, AND: searches through information about Mesquite's features and returns a description of those features matching ALL of the search terms
-  **! Search Features** — Search Features, OR: searches through information about Mesquite's features and returns a description of those features matching ANY of the search terms

(The Search Data feature  searches the current data file or project and returns objects (e.g. taxa, characters, sequences) that match the search string.)

The results of the search are shown in the main Mesquite window (shared by the Log window). The results may list modules, tools or manual pages that match the search terms. If you touch on the name of a manual, you may get a list of the contexts in which this module can be used. (**Please:** we realize that the search results can sometimes be long and difficult to follow. Please give us your feedback as to what you would most want to know in response to a search, so that we can improve this feature.)

**Example files:** One of the best ways to learn about analyses is via the example files, which

are distributed with the main Mesquite package and with various of the add-on packages. The example files are present in the "examples" folder of Mesquite\_Folder. Some additional examples are outlined on the [Studies](#) page.

**Explanation areas:** [Explanation areas](#) at the bottom of each window may describe the window, its contents, or the function of a selected button or other object. If you hold down the Shift key as you select a menu item, an explanation for it will (usually) appear in the explanation area of the frontmost window. Explanations areas are also present in some dialog boxes.

**Web Search:** This menu item uses a web browser and the Google Search engine to search for terms entered by the user. Users may restrict their search to the Mesquite Manual if desired.

**Mesquite FAQ:** There is a preliminary [Frequently Asked Questions](#) page.

**Window information bar:** The [small tabs](#) near the top of each window allow you to select a view: Graphics, Text, Parameters, Modules and Citations. The Modules view in particular can help you learn about Mesquite calculations. It shows the set of modules currently active in the analysis and or graphics shown in the window. This is shown as a tree of modules employing other modules (a bureaucratic hierarchy!). If you pass the cursor over a module name, an explanation for it appears in the explanation area at the bottom of the window. If you touch on the name of a module, a menu will appear with choices to show more information (if available). If there is a special manual for the module, an extra label will appear that will link you to the manual.

**Additional web pages:** Mesquite surveys modules for information and composes a set of web pages. These web pages list all the modules loaded, brief explanations as to what these modules do, and the scripting commands these modules respond to. These Mesquite-composed HTML pages are in a directory called "[Mesquite\\_Prefs](#)" within a directory [Mesquite\\_Support\\_Files](#), which may reside in different places depending on your operating system. If you can find these pages, you might want to store a bookmark, or alias, or shortcut to one of them so that you can find them again without going through Mesquite.

**Menu & Control Explanations:** The menu item [Window>Menu & Control Explanations](#) causes Mesquite to compose a HTML page summarizing the menu items or buttons for the current window, and to show it to you.

**Keyword Search:** This menu item in the Help menu of Mesquite provides a currently-primitive facility to search among the names and explanations of all of the installed and loaded modules to find a keyword. You could, for instance, search for "simulat" to find all of the modules that might have to do with simulations.

Thus, if you want to learn about:

- **Modules:**

- Use the menu items in the **Help** menu to go to the appropriate HTML pages for modules or packages. You can also get to the HTML pages of packages by touching the banners in the Mesquite startup window (the window named "About Mesquite").
  - Use the **Search Features** facility in each window
  - Open the **Modules** view of the window of concern (using the tab in the information toolbar) and move the cursor over the names of the modules to see explanations in the explanation area, or touch on their names to go to their information pages composed by Mesquite.
- **Menu items**
    - Hold down Shift as you select a menu item to make an explanation for it appear in the explanation area of the frontmost window.
    - Use the **Menu & Control Explanations** menu item of the **Window** menu to make a web page summarizing the functioning of the menus and buttons for the foremost window.
  - **Buttons and tools:**
    - Touch on a tool in the tool palette to make an explanation for it appear in the explanation area of the frontmost window.
    - Use the **Menu & Control Explanations** menu item of the **Window** menu to make a web page summarizing the functioning of the menus and buttons for the foremost window

## How to remember or document what you have already done?

With an interactive program having as many options as Mesquite, it can be difficult to remember what options are currently in effect. Three facilities help you keep track of what you've done.

- Information bar of windows. The [information bar](#) has various tabs that control alternative views of a window's contents. Some of these give information about the current calculations, the parameters in use, and details about the modules in use by the window.
- The Log Window, available by selecting **Mesquite Log** in the **Window** menu, records commands given and messages relayed to the user. This text is also saved automatically to a file called "Mesquite Log" within the directory **Mesquite\_Support\_Files**.
- Auto-scripting for file saving. When Mesquite saves NEXUS files, it automatically constructs a script that attempts to return an analysis to its current state. This not only allows a user to save a snapshot of an analysis, but the script itself can also be inspected to determine current parameters (in case that's not evident otherwise). Snapshot scripts can also be seen for individual windows, by selecting the appropriate item in the **Scripting** submenu of the **Window** menu.

## Why Mesquite is complex

Mesquite has many options depending on what modules are installed and loaded. Its modularity and flexibility allow for many possible analyses, but also create challenges for the user (and the manual writer). There are too many possible analyses for us to have yet written instructions specifically for each. If the user wants to perform some particular analysis, he or she may have to use his or her puzzle-solving ability to figure out how to achieve the analysis by combining Mesquite's various functions.

Why are so many choices? Why do some analyses assault you with a gauntlet of many dialogs? These are consequences of Mesquite's flexibility. We have tried to protect you from complexity as much as possible.

You may find a slightly different rhythm of thinking will be needed when confronting Mesquite. For instance, suppose you wanted to see how much the likelihood for a character varies if random noise is added to the branch lengths of the phylogeny. If you were lucky, there would be a macro or some menu item that would build your desired analysis directly, but for this question there isn't currently such a shortcut. What you'd like to see is a frequency distribution of likelihoods calculated for the character over a series of trees, each tree derived from some given tree by adding random noise to the branch lengths. This sounds like a bar chart (histogram), but which bar chart? It seems to concern characters, so perhaps it is a Characters bar chart? No, each sample point is a tree, and thus you should ask for a Trees bar chart. When a dialog box asks you what value to calculate for the trees, what do you say? The likelihood concerns a character, and thus you respond "Tree value using character". In the next dialog, you can choose character likelihood. You will also be asked what source of trees. The trees are randomly modified by adding noise to the branch lengths; therefore, choose "Randomly Modify Tree" and for the particular modification, "Add noise to branch lengths". Although this may seem complex, it does allow you to design the analysis exactly as you wish. You may use the example files, instructions in this and other manuals, and the [Mesquite discussion list](#) in order to learn how to do the analyses you need.

Features that help reduce Mesquite's complexity to the user are:

- **Primary Choices:** Submenus and dialogs boxes with lists of alternative choices may have a small number of choices listed then an item "Other Choices..." (if a submenu) or a check box "Show Secondary Choices" (if a dialog box). If these are selected, a larger number of choices is offered, including the "secondary" choices. This division of choices into the primary choices (the ones we expect will be most frequently selected) versus the secondary choices helps keep the commonly seen array of choices small. (This division can be turned off in the Defaults menu of the Project and Files window or the Log window.)
- **Macros:** Mesquite can be instructed by a [scripting language](#). Macros can therefore be written or automatically generated to script complex calculations. Macros appear in submenus in the appropriate menu.
- **Configurations:** Even if many packages of Mesquite modules are installed, you can ask Mesquite to load only a subset of them. This allows Mesquite to startup more quickly and to present a simpler interface (i.e., with fewer options). You can control

configurations using the submenu File>Activate/Deactivate Packages>.

---

# Why Mesquite was made

We give two answers, the [practical](#) and the [poetic](#), and a comment on the relationship between [MacClade and Mesquite](#).

## The practical answer

Mesquite represents a new approach to computing for evolutionary biology. In recent years there has been a proliferation of computer programs for phylogenetic analysis, each designed for some particular analysis (e.g., see [Felsenstein's compilation](#) of programs). As these often involve unique file formats and user interfaces, it is difficult for users to move from one to another. Users tend to become constrained to a few familiar analyses, since any given program can't do everything, and each program has costs in learning. As a programmer one would like to respond by making a program that does everything, but there are now too many analyses available or conceivable for a single programmer or programming team to keep up. We have seen the impact of these constraints with MacClade: some users perform particular analyses in MacClade not because they are the most appropriate analyses for their questions, but simply because they are available in a familiar program. We would like to add more flexibility to MacClade, but in a monolithic program this can be difficult to do, and even if easy, there are more proposed methods than we could maintain in MacClade.

Hence, our goal was to design a general system for phylogenetic computing to which different programmers could contribute modules. Bringing different analytical tools into a common system increases possible analyses more than additively. In the end, the system has grown beyond being strictly phylogenetic, including capabilities for calculations involving characteristics of many organisms (e.g. population genetics and morphometrics) that need not involve phylogeny.

A second goal of Mesquite is to provide a graphical user interface that will operate, more or less without modification, under different operating systems (being written in Java).

## Modularity and Flexibility

"Modularity" in computer programming might follow different models. It could follow the "[Mr. Potato Head](#)" model, in which there is a central program to which different peripheral calculations can be attached in specific places. This allows useful, but limited, flexibility. Or, modularity could follow the "[Lego](#)" model, in which building blocks are attached to other building blocks, and so on indefinitely. This allows nearly unlimited flexibility. Mesquite's modularity is somewhat of a hybrid between these: there is a (small) central starting point to which modules attach, but from there modules can be attached to modules attached to modules, indefinitely, leading to considerable flexibility in the analyses that can be constructed.

To give an idea of the flexibility, consider the calculation of the parsimony score of a tree, the treelength. A treelength calculating module takes as input information a tree, and responds by returning its length. Such a module belongs to the general class of modules that return a number when passed a tree. Other modules belonging to this class ("NumberForTree") could return the likelihood of the tree, or a measure of the asymmetry of the tree's branching, or a measure of the tree's discordance with a containing species tree. A Tree Legend module can be written (and has been) that displays the treelength in a legend in the tree window, but the Legend module is designed so that the user can choose to display any other number for the tree, such as its likelihood, asymmetry, or discordance. If a programmer creates a new module to calculate a number for a tree such as the longest branch-length path from root to tip, and a user installs the module, then the longest path measurement would automatically become another option for the tree legend.

The Tree Legend is not the only place where analyses could use numbers for trees. A charting module could display the numbers calculated for a whole series of trees, or a tree search module could use the numbers to find a tree with minimum or maximum values for the number. When such modules are made, they can automatically have access to whatever NumberForTree modules are available. Thus, the chart could show treelength, or likelihood, or asymmetry, or discordance, or longest path. Likewise, the tree search module could seek to optimize any of those. If a programmer makes a new module to analyze numbers for trees, then suddenly all existing NumberForTree modules have a new context in which they can be analyzed. If a new NumberForTree module is made, it will appear as a new option under each of the modules making use of NumberForTree. Hence the number of alternative analyses rises as the product of numbers of modules of different interacting types.

Of course, the trees used had to come from somewhere. One module might supply the trees stored in a file, another might simulate trees using a simple markovian model of speciation and extinction, another might simulate trees as gene trees coalescing within a species tree. Characters likewise might come from a stored matrix, or might be simulated by a stochastic module of evolution, or might represent reshufflings of existing characters. This means that any calculations using trees or characters can either do their calculations on observed data and reconstructed trees, or can derive null distributions under stochastic models. The calculations don't have to do anything special to achieve this flexibility; they simply let the user choose the sources of trees and characters.

(For more details about modularity, see [How Mesquite works](#))

## A community of programmers

Our hope is that building-block style of the Mesquite system will encourage programmers to write modules for their own favorite analyses. Another attraction of the Mesquite system is that many of the details of reading and writing of files, user interface and graphical display are already taken care of, and the programmer might worry only about a single calculation. The system is built in Java and is therefore platform independent. It is also possible for programmers to link in code written in C, C++, or some other language.

We have attempted to design the system so that a programmer's efforts can be recognized as an independent, citable contribution. Modules or suites of modules can have their own names, own manuals, be distributed and cited separately. They simply run within the Mesquite system.

Mesquite source code is available for [download](#). This allows other programmers to modify existing source to create new modules.

---

## The poetic answer

The goals of Mesquite are these:

**To change the economics of imagination in evolutionary biology** – There are three ways we envision Mesquite stimulating imaginative ideas and their successful spread:

- Stimulating the creation of ideas: analyses. With multiple alternative modules available for various parts of an analysis, and with modules specializing in questions from various branches of evolutionary biology (e.g., phylogenetics, molecular evolution, population genetics, geometric morphometrics) the diversity and scope of analyses that can be constructed by combining different modules is great. Individual users can carry their imaginations through to an analysis that no one has tried previously. Indeed, Mesquite, by offering the alternatives to be combined, doesn't merely provide analytical tools for questions that have existed: it suggests and provokes new questions.
- Stimulating the creation of ideas: biology. As does [MacClade](#), Mesquite has an emphasis on visualization and exploration. An idea – whether a particular hypothesis about the evolutionary history of a group, or a stochastic model of a process – can be followed through to its consequences, and visualized. A biologist can ask "What if this were the phylogenetic tree?" and a character's evolution can be reconstructed or simulated on this tree, and the results visualized. A biologist can ask "What if the population had population sizes fluctuating in this way?", and coalescence can be simulated, and the results visualized. In providing users with the tool to ask "What if?" questions, Mesquite provides an extension of the imagination. Such tools are vital in a field whose ideas have consequences that are difficult to predict or grasp without the aid of a computer.
- Enhancing the efficient distribution of ideas: programs. The imagination of theoreticians and programmers has produced many valuable ideas for approaches and methods, and many valuable programs to implement them. However, some of the ideas haven't been translated to programs, and many of the programs haven't been as much explored and used as would have been good. We don't know, as a field, how many important ideas will lie unused for decades until they are rediscovered. By allowing the programmer to focus on the precise idea proposed (Mesquite providing much of the housekeeping code for the programmer), Mesquite may allow some ideas, that might never have been implemented, to be realized as tools. By providing a fairly user-friendly context in which modules can operate, Mesquite may encourage some programs to be used more broadly and more easily than otherwise.

**To continue to promote a phylogenetic perspective in evolutionary biology** – The last few decades have seen the realization of the importance of viewing organismal diversity and evolution in the light of phylogeny. This revolution is analogous to and as fundamental to its field as the revolution in cosmology from a Newtonian view of space to an Einsteinian view of space (Maddison and Pérez, 2000). Just as mass curves space, phylogeny has curved the space of biological diversity, providing a distortion on the distribution of traits of organisms we see around us. MacClade and Mesquite are both designed to provide a corrective lens, to help us to see organisms and their traits in their natural orientation within this curved space along the phylogeny. Mesquite's modularity allows this perspective to be extended to fields such as morphometrics, in which a phylogenetic perspective has relatively recently begun to suffuse the field.

---

## Which to use, Mesquite or MacClade?

[Version 4 of MacClade](#) (Maddison & Maddison, 2000) was released in October 2000, and the MacOS X compatible version 4.04 in July 2002. (It is now at version 4.08.) The reader might wonder why we have been working on two different programming efforts, and whether they are intended for different uses. Although Mesquite's extensibility means that it has many more features than MacClade, MacClade has some features that Mesquite lacks. Calculations and functions of MacClade's tree window not currently available in Mesquite include particular charts (e.g., Changes and Stasis), equivocal cycling, the concentrated changes test, some of the parsimony options (irreversible, stratigraphic, Dollo), a detailed Trace All Changes mode, and some options for tree printing (e.g., saving a tree to the clipboard). For some users the most significant advantages of MacClade 4 over MacClade 3 are in the data editor, where editing of molecular sequences is more sophisticated, with better tools for manual sequence alignment and on-the-fly translation to amino acids. We cannot predict when these features will be available in Mesquite. MacClade also has a simpler interface, which some users might prefer. We imagine that in the long-term future MacClade will give way to Mesquite as Mesquite matures. For the next several years, however, the two will coexist and be complementary.

## References

Maddison, D.R. and W.P. Maddison. 2000. MacClade version 4: Analysis of phylogeny and character evolution. Sinauer Associates, Sunderland Massachusetts.

Maddison, W. and T. Pérez, 2000. Biodiversidad y lecciones de la historia. In: Enfoques contemporáneos para el estudio de la biodiversidad [Hernández, H.M., A. García Aldrete, F. Álvarez and M. Ulloa, editors]. Instituto de Biología, UNAM, Mexico. Pp. 201-220.

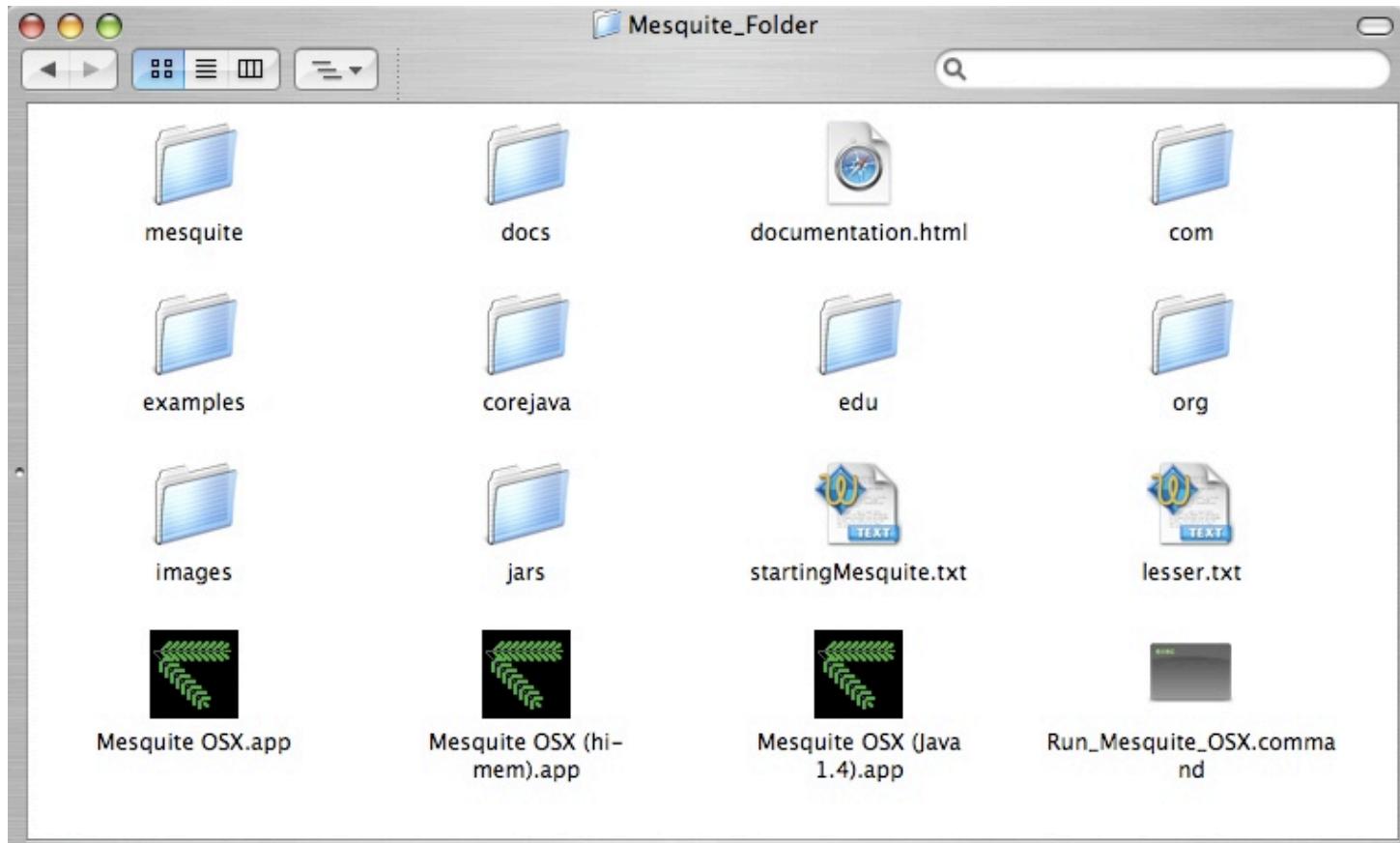
---

# Mesquite installation for MacOS X

(Please email us ([info@mesquiteproject.org](mailto:info@mesquiteproject.org)) with questions or comments about downloading Mesquite).

Instructions:

1. Download [mesquite.dmg](#).
2. If the mesquite.dmg doesn't mount into a disk image automatically, double click it to mount. You will see a Mesquite\_Folder. Drag this Mesquite\_Folder to your hard drive (e.g., your Applications folder). Its contents should look like this:



3. Double click one of the Mesquite icons to start Mesquite. We suggest you try first Mesquite OSX. This starts the most current version of Java on your computer. The other versions start Java 1.4 or allocate more memory. You might want to try these if you are having problems with the normal version. We recommend the most current versions of OS X and Java.
4. You will also notice the file "Run\_Mesquite OSX.command". This contains the command needed to start Mesquite as if from the command line. You can edit this command (using a text editor) to fine-tune the amount of memory allocated to Mesquite, and perhaps to modify startup conditions.
5. To change the **memory allocation** of the "Mesquite OSX (hi-mem)" control-click (right click) on the application, and from the drop down menu choose Show Package Contents. Inside, go to Contents/info.plist. Open this file with a text editor, and in the line "<string>-Xmx512m -Djava.library.path=lib</string>" change the number 512 either higher or lower (do not introduce spaces!).

Requirements: OS X 10.4 or higher recommended; Java 1.4 or later.

NOTE: If you have a **previous version** of Mesquite installed in your system, we advise that you delete it before installing the new version. Remember to save

More details of issues of using Mesquite under Mac OS X are described in the [Support](#) page.

---

Copyright © 2002-2007 W. & D. Maddison

# Mesquite installation for Windows

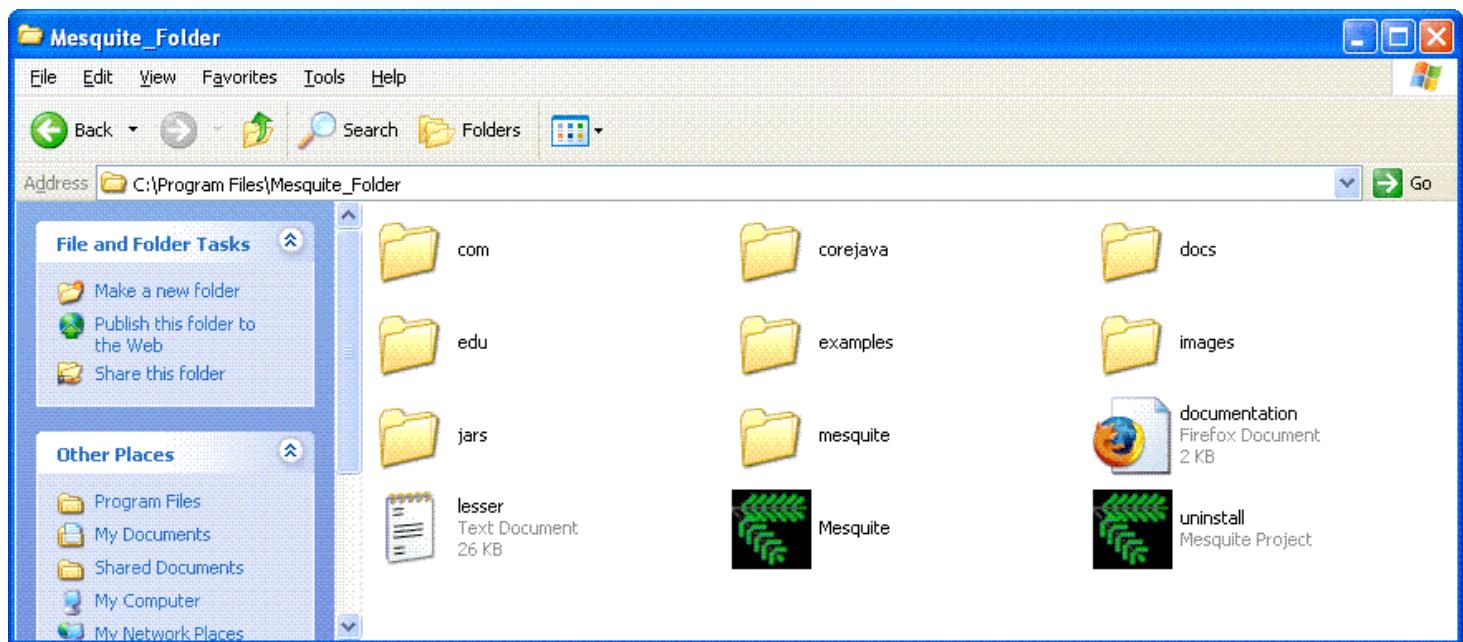
(Please email us ([info@mesquiteproject.org](mailto:info@mesquiteproject.org)) with questions or comments about downloading Mesquite).

NOTE: If you previously installed an older version of Mesquite, we recommend you delete the **Mesquite\_Folder** directory before installing the new version. First, however, you may want to take from Mesquite\_Folder/mesquite any third party packages (e.g., PDAP, TreeSetViz) before deleting the directory.

In order to run Mesquite, you must have Java 1.4 or newer installed on your computer. We recommend you install the latest version of Java before installing Mesquite ([java.sun.com](http://java.sun.com)). [Java Standard Edition 5.0](#) is suitable (JRE 5.0).

Instructions:

1. Download the [Mesquite Installer](#). There is also a special installer for a high-memory version of Mesquite [here](#). You might want to use the high memory version if you are using many trees, large trees, or large matrices.
2. Run the installer. It runs as a wizard. Once it is done, you should see a Mesquite folder something like this:



3. The file "Mesquite" with the icon is the Mesquite executable. Double click this to start Mesquite.
4. You will notice a file called "Run\_Mesquite\_Windows.bat" in the Mesquite\_Folder. This is a batch file that contains the command line instruction to start Mesquite. If you double click this file, Mesquite will start. We include this file because you can modify the command line to change how Mesquite runs when you double click the batch file. In particular, you can change the number "400", the memory allocation, to higher numbers if you need more memory (don't introduce spaces!).

More details of issues of using Mesquite under Windows are described in the [Support](#) page.

---

Copyright 2002-2007 W. & D. Maddison

# Mesquite installation for UNIX/LINUX

(Please email us at [info@mesquiteproject.org](mailto:info@mesquiteproject.org)) with questions or comments about downloading Mesquite).

Requirements: JRE or JDK 1.4 or better ([java.sun.com](http://java.sun.com)). (It is best to get Sun's Java VM; Mesquite does not run on some third-party virtual machines.) See notes on the [support](#) page, especially regarding issues with window and dialog size and placement.

Instructions:

Download and decompress this file:

- [.tgz file](#)

It will create a directory called "Mesquite\_Folder" in which all the relevant files reside. To start Mesquite, execute the run\_mesquite.sh script from the command line:

```
./run_mesquite.sh
```

This should automatically identify your java installation. If you have trouble, you may need to set your JAVA\_HOME environment variable. In the uncommon case that you want a java VM to run Mesquite that is different from the VM identified in JAVA\_HOME, you may set the MESQUITE\_JAVA\_HOME environment variable.

Once Mesquite has finished loading, go to the File menu to open a file.

If the current user directory is not the Mesquite\_Folder, and Mesquite has not been run before, Mesquite may ask you to find the file "manual.html" which resides in the **Mesquite\_Folder/docs/mesquite/** directory. This will help Mesquite find and remember where its files are.

## If run\_mesquite.sh fails

In the event that the shell script fails, it may still be possible to run Mesquite manually. The main class file is mesquite.Mesquite. To start Mesquite you can use jre or java. For instance you could use:

```
java -cp /home/myuser/Mesquite_Folder mesquite.Mesquite
```

where /home/myuser/Mesquite\_Folder could be replaced by whatever is the path to the Mesquite\_Folder. On some systems the java virtual machine is started by "jre" instead of "java", and thus the command would be:

```
jre -cp /home/myuser/Mesquite_Folder mesquite.Mesquite
```

Depending on your configuration, you may need to give the explicit path to the java virtual machine, as in:

```
/usr/java/jre1.4.2/bin/java -cp /usr/local/Mesquite_Folder mesquite.Mesquite
```

If you've changed the name of Mesquite\_Folder to, for instance, "Mesquite2", you may need to use a command like:

```
/usr/java/jre1.4.2/bin/java -cp /usr/local/Mesquite2 mesquite.Mesquite
```

If you've had to do all this because the shell script isn't working, please let us know what happened, and what you did to make Mesquite work. That way, we can make it easier for the next person.

## Window Managers

Mesquite attempts to place windows in particular places on the screen for ease of use, using standard Java calls. Some window managers override this, resulting in haphazard placement of windows. You may need to change your window manager if this sort of thing is happening.

More details of issues of using Mesquite under Linux/UNIX are described in the [Support](#) page.

---

Copyright 2002-2006 W. & D. Maddison, T. Wheeler

## Basic concepts and vocabulary

Mesquite was designed to explore the evolution of organisms and their characteristics. Here is some basic vocabulary that relates the biology to the implementation in Mesquite.

### Taxa

- **Taxon** - One of the basic entities being studied, usually a species or a gene sequence. In Mesquite (as in other programs), the word "Taxon" has shifted from its traditional meaning to become more or less equivalent to "terminal taxon" (the smallest unit of analysis of relationships).
- **Block of taxa** - A collection of taxa grouped together for analysis. In most programs dealing with phylogeny, there is only a single block of taxa allowed in a file (e.g., "Primates", consisting of 20 species of primates). In Mesquite, multiple blocks are allowed (perhaps to study associations among them, in the style of Rod Page's programs), and so it is useful to refer to different blocks of taxa. A block of taxa is equivalent to a TAXA block in a NEXUS file.
- **Group Membership (Taxa partition)** - A subdivision of a taxa block into disjoint subsets. Thus, a taxa block for Mammals may have its taxa partitioned into one of three groups, Carnivore, Ominvore and Herbivore. A taxa partition is equivalent to a TAXAPARTITION in a NEXUS file.

---

### Characters

- **Character** - A variable or characteristic measured or observed for a block of taxa. It could be from molecular data ("Nucleotide at position 367 of the 18S gene"), or it could be phenotypic ("length of wing"). Characters in Mesquite typically exist only within character matrices.
- **Character model** - a model of the evolution of a character. This could be a model for use in parsimony calculations (e.g., "unordered", a cost matrix) or in likelihood calculations or stochastic simulations (e.g., HKY85, GTR).
- **Character matrix** - A matrix representing the character states of various characters for a block of taxa. Each character matrix in Mesquite must be homogeneous with respect to type of character (e.g. nucleotide sequences can't be intermingled with morphometrics data).
- **Character partition** - A subdivision of a characters into disjoint subsets. Thus, a character matrix for insects may have its characters partitioned into one of three groups, larval, pupal, and adult. A character partition is equivalent to a CHARPARTITION in a NEXUS file.
- **Character inclusion set** - A specification of what characters are to be included and what excluded from analyses. This is equivalent (in inverse form) to an EXSET in a NEXUS file.

- **Character weight set** - A specification of what weights are to be applied to characters in summed calculations (especially for parsimony calculations). This is equivalent to a WTSET in a NEXUS file.
  - **Parsimony model set** - A specification of what parsimony model of character evolution (i.e., what "transformation type") is to be applied to each character of a matrix. This is equivalent to a TYPESET in a NEXUS file.
  - **Probability model set** - A specification of what probabilistic model of character evolution is to be applied to each character of a matrix.
- 

## Trees

- **Tree** - A tree representing the relationships of the taxa. If the taxa are species, the tree represents phylogeny; if the taxa are gene sequences, the tree represents a gene genealogy.
  - **Block of trees** - A collection of trees. This might be a set of trees saved by the user, or output by a program like PAUP\*.
-

# Making, opening and saving data files

Mesquite is currently designed for data files following the NEXUS format (Maddison, D.R., D.L. Swofford, and W.P. Maddison. 1997. NEXUS: An extensible file format for systematic information. *Systematic Biology* 46: 590-621) although it can import and export files of other formats. Thus, you could create your data file with a text editor or word processor if you followed NEXUS conventions. However, you'll probably find it easier to use Mesquite's data matrix editors, tree windows, and so on, to specify the information in the data file, and let Mesquite handle the formatting issues.

Mesquite can read NEXUS files created with [MacClade](#), and can save files that MacClade understands. For information on importing and exporting other formats, see the page on [using Mesquite with other programs](#).

## Creating a new data file

To create a new blank data file, choose [File>New](#) (i.e., the New menu item in the File menu). You'll be presented with a dialog box in which you enter the name of the set of taxa (e.g., "Drosophila") the initial number of taxa, whether or not you want to show a tree window, and whether to make a character data matrix. (The taxa could be species, or sequences, or whatever are your "terminal taxa", "Operational Taxonomic Units", or evolutionary units.) (You can just leave the name of the set of taxa as "Untitled" if you wish, but that may become confusing if you ever have more than one set of taxa in the same file.) You can later add more taxa using [\(List of Taxa\)List>Add Taxa](#) or [\(Character Matrix Editor\)>Matrix>Add Taxa](#), or by using the Add Taxa tool in the [Character Matrix editor](#).

When you make a new data file, you'll be presented with a list of taxa or perhaps a tree window. The taxa are automatically named "taxon 1", "taxon 2", and so on. You can rename a taxon name (e.g., "D. melanogaster", "D. willistoni", and so on) by selecting the I-beam tool in the List of Taxa window and touching it on the taxon name. There is a submenu, [\(List of Taxa\)List>Alter Taxon Names>](#), that might offer other ways to edit taxon names. Taxon names can also be edited in the [Character Matrix editor](#), either manually or using the Taxon Utilities and Taxon Names submenus of the Matrix menu.

A new data file does not automatically include a matrix of character data unless you request it. To add new matrices, see the section on the [Character Matrix editor](#).

## Opening an existing data file

To open an existing data file, use [File>Open>File](#). If Mesquite detects that the file is not a NEXUS file, it will ask to you to choose its file format for importing.

## Saving a data file

Save a data file using [File>Save File](#) or [File>Save File As](#). You can also export to other formats using [File>Export](#).

You can indicate that a character matrix should be written in the NEXUS file as INTERLEAVE

(or not) by going to the Character Matrix Editor window and selecting the menu item Matrix>Current Matrix>Write Interleaved

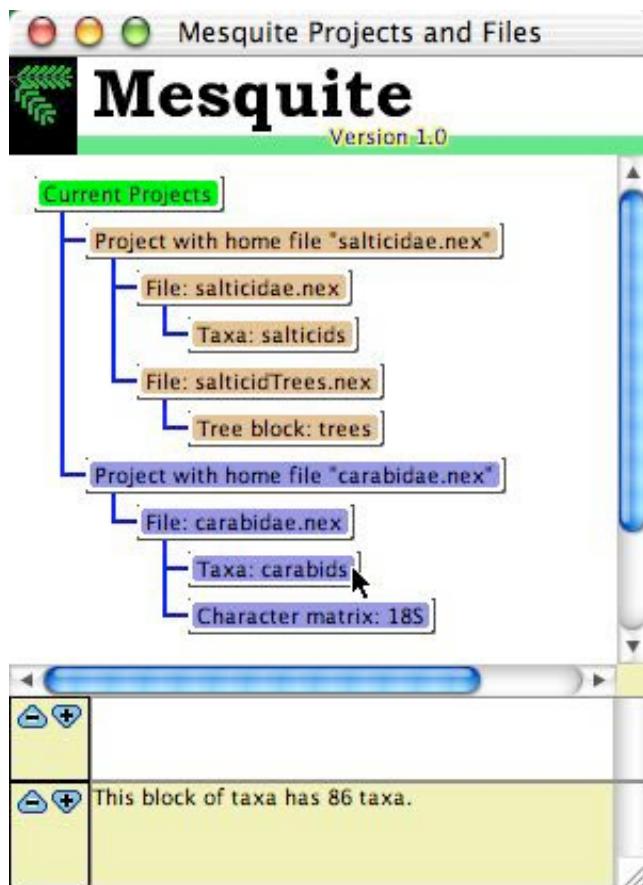
## Projects and files

Mesquite is not restricted to considering only one file at a time, but instead can collect information from various files and consider it together. Such a collection of files sharing information is called a **project**. Implicitly when you first open a file, a project is created, one which contains only a single file. Other files can be linked into the project using the Link commands.

Since Mesquite can accumulate and analyze a more or less indefinitely large collection of elements of information (several sets of taxa, data matrices, and so on), Mesquite doesn't need to respect the boundaries of files. That is, it could read a TAXA block from one file on the disk, and read a data matrix for those taxa from another file on disk, and a set of trees from another file. While other programs can handle external treefiles or command files, Mesquite can handle external character matrices, assumptions, and so on.

Mesquite therefore makes a distinction between the collection of elements of information that are currently interacting with one another in Mesquite's calculations, and the physical files on disk or server. The former collection, which may include information gathered from several files, is called a project. The set of files to which the elements of information in a project belong are said to be **linked**.

Mesquite shows a list of the projects and files currently active in the Projects and Files window. In the example shown here, there are two projects currently open. The first, marked by a sandy color, includes two linked files, "salticidae.nex" with a block of taxa and "salticidTrees.nex" with trees. The second project, marked in blue, includes one file "carabidae.nex" with a block of taxa and a character matrix. The windows (tree windows, data editors, etc.) belonging to these two projects will also be distinguished by these different background colors. **Information is not shared between projects**, so that when you are using the windows of "salticidae.nex", the character matrix "18S" will not be available for use.



## Opening versus Including versus Linking files

In Mesquite there are three ways to read a file: Open, Include and Link.

1. **Open** – If a file is to be opened up independently of any other open files, and not share information, it should be opened using the Open File... menu item. A file opened in this way is treated as belonging to a project separate from any other files open at the time. In the graphic above, Open File... was called twice, once to open salticidae.nex, second to open carabidae.nex.
2. **Include** – To read in the contents of a file and merge them into an existing file, so that all of its information becomes part of the existing file and is saved into that existing file, use the Include File... menu item.
3. **Link** – To read in the contents of a file and add its information to a collection of information in a project in use, but to maintain the file separate for purposes of writing to the disk, use the Link File... menu item in the File menu. A file opened in this way will become part of the project. Because of the interdependencies among elements of information that can exist (list of taxa in a data file matches list of taxa in a tree file), it is possible that editing information that belongs to one file will also change information in a linked file. In the graphic above, Link File... was called once, to link salticidaeTrees.nex with salticidae.nex.

# Mesquite's menus

Mesquite's menus differ from window to window and vary depending on the calculations being performed. In operating systems such as Windows and Linux, each window has its own set of menus embedded at the top, thus making it easy for the user to understand the context – a particular window – in which the menu item exists. Users of the Macintosh OS will see the screen menu bar change depending on what window is at the forefront. On the other hand, there is considerable duplication of menu items from window to window, which reflects the fact that windows share context to varying degrees, but which may be confusing to users of operating systems with window-specific menus. We will not here take sides about which menu paradigm is better, but note that Mesquite's menu are firmly organized so as to reflect the hierarchical structure of the calculations performed.

1. [Learning about menus](#)
2. [Standard menu items](#)
3. [Where to find menu items](#)

## Learning about menus

Mesquite menus include a few constant ones (File, Edit, Window) and various other menus that change as different modules are used. This situation could be confusing to the user. For instance, menu items for particular modules may appear in different places depending on the context.

Below is a list of the standard menu items that should appear in most configurations of Mesquite. There will likely be other menus and menu items in addition to these.

Mesquite can automatically generate documentation for the current menu configuration and present it to the user as a web page. This can be done by selecting the **Menu & Control Explanations** menu item from the **Window** menu of Mesquite. Also, if you hold down the Shift key while selecting a menu item, an explanation of the menu item will appear in the explanation area of the foremost window.

## Standard menu items

### File

- **New** – [Make a new file \(as a new project\)](#)
- **New Linked File** – [Make a new file \(linked to the current project\)](#)
- **Open File...** – [Open file as a separate project \(not linked to current project\)](#)
- **Open Other**
  - **URL...** – [Get file from web server](#)
- **Link File...** – [Open file and incorporate into the current file \(or home file of current](#)

- project)
- **Include File...** – Open file as linked (and hence sharing information with) the indicated project
- **Close File** – Close the file
- **Close All** – Close all projects
- **Close Window** – Close the foremost window
- **Save File** – Save the file to disk.
- **Save File As...** – Save the file to disk under another name.
- **Get Info...** – Shows what is the current file and its location.
- **Export...** – Save the file to disk under another name and with a format other than NEXUS.
- **Save Window As Text...** – Save the text version of the window (visible by touching on the Text tab of the window).
- **Print Window...** – Print the foremost window.
- **Print Window to Fit Page...** – Print the foremost window so as to fit within a page.
- **Save Window as PDF...** – Save the foremost window as a PDF image.
- **Activate/Deactivate Packages** – Change the configuration of modules loaded. These menu items affect what modules, among those installed in your copy of Mesquite, are actually loaded at startup. Most configurations load only a subset of menu items, for instance to simplify the interface.
  - **Use All Installed modules** – On next startup, load all installed modules
  - **Choose Configuration** – Choose the configuration of module packages loaded at next startup.
  - **Delete Configuration** – Delete user-defined configurations
  - **Define Configuration** – Define a new configuration of modules to be loaded at startup
- **Macros**
  - **Show Macro List** – Show list of Macros available, and where available explanations as to what each does.
  - **Edit Macro Information** – Change name and explanation for user-defined macros.
  - **Show Macro List** – Show list of Macros available, and where available explanations as to what each does.
- **Rename Log file** – Renames the current log file.
- **Reset Menus** – Forces rebuild of all menus. Useful on Windows because of a bug in the Java virtual machine that occasionally scrambles menu names
- **Force Quit** – Forces Mesquite to quit without the usual clean-up and checking whether the file has unsaved changes. Should be used only in emergency situations (e.g. if some calculation appears to have crashed)
- **Quit Mesquite** – Quit Mesquite

## Edit

- **Undo** – Minimally functional. Undos most recent change to tree in tree window. Not available for Character Matrix Editor and other windows
- **Cut, Copy, Paste** – Have their standard uses, though currently only deal with text.

- **Font** – Change the default font for the foremost window.
- **Font Size** – Change the default font size for the foremost window.
- **Add AUTHORS block to File** – Adds an AUTHORS block to the file, important for collaborative projects.
- **New Generic Nexus Block** – Make and edit a NEXUS block for the file.
- **Edit Mesquite script** – Edit a Mesquite script contained in the file
- **Edit Comment** – Edit the comment about the file.

## Characters

- **Character Matrix Editor** ▶ – Show a data editor for a character matrix
- **List of Characters** ▶ – Show the list of characters for a particular character data matrix
- **New empty matrix...** – Make a new, blank character data matrix
- **Make New Matrix from** ▶ – Make a new character data matrix by copying an existing matrix or filling the matrix from a special source (e.g., simulated characters)
  - sources of matrices listed here
- **List of Character Matrices** – Show the list of available character data matrices
- **List of Character Models** – Show the list of available models of character evolution
- **New Character Model...** ▶ – Make a new model of character evolution
  - types of character models listed here
- **Edit Character Model...** ▶ – Edit an existing model of character evolution
- **Lists**
  - **Parsimony model sets** ▶ – Show a list of the sets of assignments of parsimony models for a particular character data matrix
  - **Probability model sets** ▶ – Show a list of the sets of assignments of probability models for a particular character data matrix
  - **Character sets** ▶ – Show the character sets for a particular character data matrix
  - **Inclusion sets** ▶ – Show the character inclusions sets for a particular character data matrix
  - **Character Partitions** ▶ – Show the character partitions for a particular character data matrix

## Taxa&Trees

- **New Tree Window** ▶ – Show a new tree window for a particular set of taxa (multiple tree windows can be shown)
- **Current Tree Window** ▶ – Bring an existing tree window to the front
- **List of Taxa** ▶ – Show a list of taxa for a particular set of taxa
- **New Block of Taxa...** – Make a new set of taxa
- **List of Taxa Blocks** – Show a list of the different sets of taxa
- **List of Trees** ▶ – Show a list of trees in a tree block in the file
- **New Empty Block of Trees...** – Make a new empty block in which to store trees
- **Make New Trees Block from** – Make a new block of trees by copying an existing block or filling the block from a special source (e.g., simulated trees)

- available sources of trees listed here
- **List of Tree Blocks** – Show a list of tree blocks in the project
- **Multi Tree Window** – View several trees at once in window.
- **List of Taxa Partitions** – Show a list of taxa partitions that exist
- **List of Taxon Sets** – Show a list of the Taxon sets that exist

## Analysis

- Menu items to make new charts and other analyses are listed here

## Window

- **Menu & Control Explanations** – Causes Mesquite to compose a web page listing the menu items currently in the menu bar for this window, and where available gives explanations for what they do. Also lists the buttons currently in the window, and where available gives explanations for what they do.
- **Clone Window** – Clones the foremost window. Useful to reproduce an existing analysis then change it slightly. Cloned window will appear exactly above original window; thus the clone should be moved to view the original.
- **Save Window As Macro** – Save foremost window as a macro. This macro can later be used to reproduce the window's appearance and analyses.
- **Macros** – Execute the selected macro.
- **Scripting** – The "Show Snapshot" item displays the script that would be required to return the foremost window to its current states; "Send Script" sends a script to the foremost window.
- **Bring All Windows To Front** – Brings all Mesquite Windows to the front.
- **Current Windows** – Allows you to select current Mesquite windows to bring them to the front.
- **Mesquite Startup Window** – Makes the Mesquite startup window visible.
- **Mesquite log** – Makes the Mesquite log window visible.
- **Mesquite Projects and Files** – Makes the Mesquite projects and files window visible.
- **Display License** – Shows the license under which Mesquite is distributed.

## Help

- **Mesquite Manual** – Show home page of Mesquite manual
- **Modules Loaded** – Show web page of modules that are installed and loaded
- **Keyword Search** – Search for exact keyword among the names and explanations of loaded modules.
- **Web Search** – Search for exact keyword using the Google search engine.
- **Scripting Commands** – Show web page of available commands for scripting
- **Active Module Tree** – Show the entire employee tree of modules
- **List Active Modules** – Write to the log a list of the entire employee tree of modules
- **List Prerelease Modules** – Write to the log a list of any modules loaded that are

labelled as prerelease and substantive

- Show Developer Statistics – Write to the log some information useful for debugging.

## Where to find menu items

Other menus come and go depending on which window is in front and what calculations are in use. Thus, the standard Tree Window has a Tree menu.

As the user requests calculations and output of results, the menus change. The reason for this is that different menus are appropriate in different circumstances. Thus, if the user requests the Balls and Sticks method to draw trees, a menu item "Spot size" appears in the Tree menu. Spot size controls the size of the balls drawn at the nodes of the tree.

Exactly where a menu item appears may at first be confusing to the user, but there is a certain logic to it that hopefully won't be too obscure (technical details are available in the [developer's documentation](#)). The "Spot size" menu item belongs to the module that draws trees in the Balls and Sticks fashion, since that module wants the user to be able to control the size of the balls. That menu item appears within the menu of the modules that employ it to draw trees. Since the Drawing menu belongs to the module that coordinates tree drawing, and it is this module that (indirectly) is employing Balls and Sticks to draw the tree, the menu items belonging to Balls and Sticks appear in the Drawing window.

A module can also have its own menu, as does the tree drawing coordinator. Thus, the menu items of a module (usually) appear in the menu belonging to its closest employer module that has its own menu. We say "closest" because sometimes a module will have a menu item, but its immediate employer won't have a menu to put it in. The menu item drifts upward in the employment hierarchy until it finds an employer that has a menu that can house it.

This system means that menu items appear in an appropriate context, but that that context is not fixed because it depends on what employer has a menu in which the item can be housed.

---

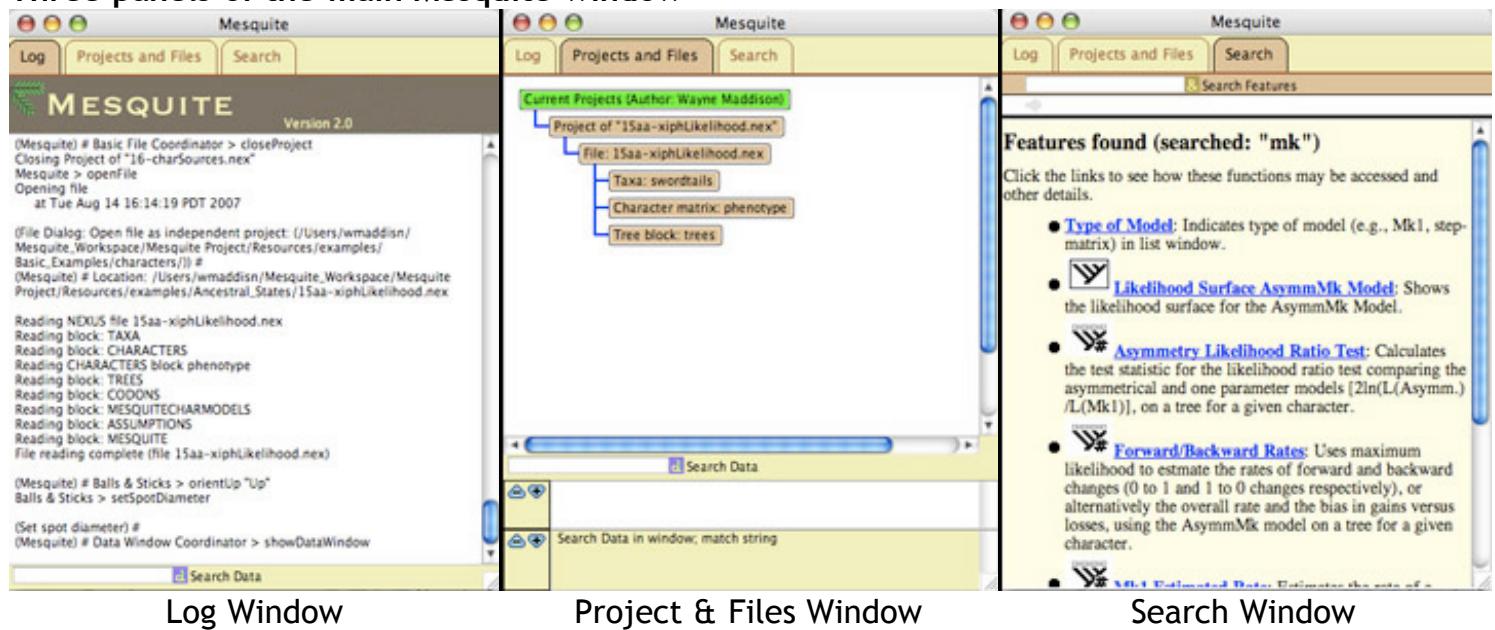
# Mesquite's Windows

With version 2 of Mesquite, separate windows can be bundled together as tabbed panels in the same window. Because most of these panels can exist as separate windows, we will refer to them as "windows" even though you may rarely see them as separate windows.

Some basic Mesquite windows are:

- **Startup window** - the colorful window with the mesquite leaf, the progress bar indicating modules loading, and the banners at right for installed and loaded packages.
- **Main Mesquite window** - this has three panels, the Log Window, the Project and Files Window, and the Search window. (See images below.) One important aspect of the Log window is that it has the Defaults menu for changing some important preferences.
- **Tree Window** - the window to view and edit trees
- **Character Matrix Editor** - a spreadsheet editor for character matrices
- **List Windows** - show lists of characters, taxa, trees, models and other objects.

## Three panels of the main Mesquite window

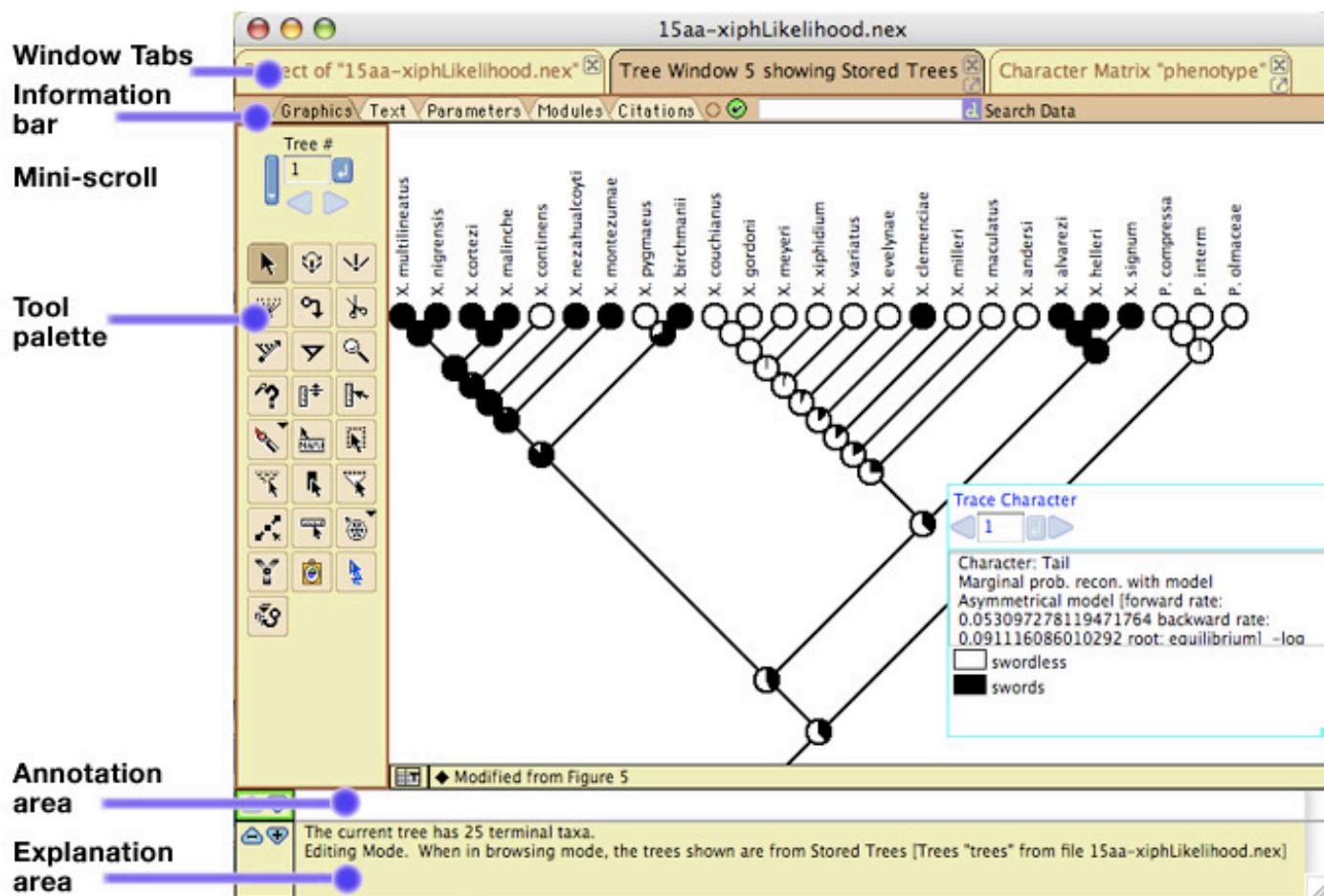


On this page are explanations of the general features of windows, as well as issues specific to list windows.

You can output a window by printing it, saving it as a PDF file, or saving a text version. To print a window, select from the Print menu items in the File Menu. You can also save the window as a PDF document using Save Window as PDF, or save it as text using Save Window as Text.

## Common features of windows

Windows in Mesquite have a consistent structure, with these components commonly appearing:

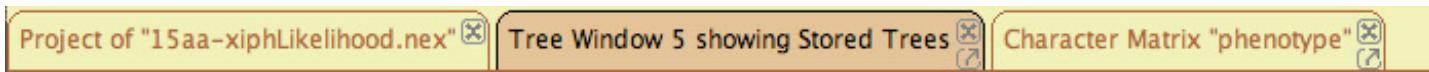


Here are brief descriptions of some of these:

- [Window Tabs](#)
- [Information bar](#)
- [Mini -scrolls](#)
- [Tool palettes](#)
- [Annotation area](#)
- [Explanation area](#)

## Window Tabs

As noted above, windows can exist either separately or bundled within a single window as tabbed panels. When bundled, the window will have a series of tabs at the top that permit you to choose which panel (i.e., which window) you want to see:



The tab that shows dark is the current window shown.

Tabs can be closed using the . By touch this you are asking the window represented by the tab to be closed.

Windows can be "popped out" as independent windows using the pop out button (  ). A popped out window can be returned to being a tabbed panel within the project window using the pop in button (  ) at the upper left of the popped out window.

You can turn off window bundling by deselecting Open Windows as Tabs in the Defaults menu of the Log window.

## Window information bar

Windows can have an information bar that allows you to access information about the window, its calculations, and the modules involved in them. If the information bar is not showing and you wish to see it, select "Show Information bar" in the Window menu.



The tabs in the information bar choose the different modes of display of the window. The first two of these (the graphics tab, and the text version tab), cause the window to display its basic output (e.g., the spreadsheet data editor, the tree drawn in a tree window) in either a graphics form or a textual form. The remaining tabs give information of other sorts.

The window modes are:

- **Grapical version of output** – The standard output of the window showing results and analyses.
- **Textual version of output** – A textual version of the results and analyses.
- **Parameters of modules** – A list of the current settings and parameters of the modules involved in producing the window and its contained results.
- **Tree of Modules** – Shows the employee tree of all modules participating in the window
- **Citations for modules** – Gives citations for some of the modules participating in the window.

## Mini scrolls

Mini scrolls are used to scroll among trees, characters and other items. The small blue button right of the text edit box, when hit, tells the scroll to enter and use the number within the edit box. Otherwise you can use the arrows to scroll forward or back. If the change is not acceptable (e.g., because you are already at the minimum or maximum value) the arrow is dimmed.



## Tool palettes

In various windows are tool palettes that look something like this:



The button of the current tool is shown darker. Usually in the explanation area of the window, an explanation is shown for a tool when you click on it. Some tools (whose buttons may show a small inverted triangle) when double clicked, right-clicked or control-clicked, show a menu in which options can be chosen.

## Annotation area

The white annotation area shows footnotes or other annotations stored for taxa, trees, characters or other objects. These annotations are sometimes editable. At its left are two buttons, a minus and a plus arrow. These reduce and increase the height of the annotation area.



## Explanation area

The explanation area, below the annotation area, shows explanations for tools, modules, objects in files, and so on. Its height can be controlled by the minus and plus arrows.

## List Windows

In the Characters and the Taxa&Trees menus are available various windows that give lists of objects with information about each, including the List of Characters and the List of Taxa Windows. Other List Windows present lists of tree blocks, taxa blocks, character matrices, character models, parsimony model sets, and so on. These List Windows have a consistent interface: you can select rows and columns, show and hide columns, and possibly rename and delete objects.

Characters "Bembidion matrix"				
Character	In States	Parsimony Model	Probability Model	Group
50 sac membrane	✓ 2	Ordered	Mk1 (est.)	Male Genitalia
51 ventral bulge	✓ 3	Ordered	Mk1 (est.)	Male Genitalia
52 aedeagal apex	✓ 2	Ordered	Mk1 (est.)	Male Genitalia
53 vaginal plate	✓ 2	Ordered	Mk1 (est.)	Female Genitalia
54 apex spermatheca	✓ 2	Ordered	Mk1 (est.)	Female Genitalia
55 S1 setae	✓ 4	Ordered	Mk1 (est.)	Female Genitalia
56 S2 length	✓ 4	Ordered	Mk1 (est.)	Female Genitalia
57 S2 dors.lat. setae	✓ 7	Ordered	Mk1 (est.)	Female Genitalia
58 stern. X setae	✓ 2	Ordered	Mk1 (est.)	Female Genitalia
59 autosome #	✓ 5	Ordered	Mk1 (est.)	Chromosomes
60 Y chrom size	✓ 3	Ordered	Mk1 (est.)	Chromosomes
61 L1 frontale	✓ 3	Ordered	Mk1 (est.)	Larval Structure
62 L1 head color	✓ 3	Ordered	Mk1 (est.)	Larval Structure
63 post FR3 micro.	✓ 4	Ordered	Mk1 (est.)	Larval Structure
64 FR3 microsculpture	✓ 4	Unordered	Mk1 (est.)	Larval Structure
65 eggbursters	✓ 4	Ordered	Mk1 (est.)	Larval Structure
66 pron. microtuber	✓ 6	Ordered	Mk1 (est.)	Larval Structure

At left of the List Window is a tool palette. The **arrow tool** allows you to select columns or rows. The **I-beam tool** allows you to edit the name of an object. The **magic wand tool** selects all rows that have the same or similar value in the column touched. The magic wand tool has alternative selection criteria that can be chosen by clicking and holding the cursor on the tool in the tool palette. The **sort tool** reorders the objects listed according the column on which you touch. Thus, if you touch on the leftmost column showing names, the objects would be sorted alphabetically. If you touch on a column showing a numerical result, the objects are sorted in ascending or descending order of the number. The sort tool sorts in ascending order by default, but by descending order if the option or ALT key is held down.

Each List Window has a Columns menu, with which you can show or hide columns. Some columns merely inform you of some quality of the object. Others may represent analyses. For instance, the column may show the value of some statistic for each of the characters listed in the List of Characters window.

For some of the List Windows, you can ask to delete the objects whose rows are selected by choosing "Delete Selected" from the List menu.

# Mesquite's Charts

Charts compile and display values for a series of objects or items, whether taxa, trees, characters, or matrices. Mesquite has two primary styles of charts:

- **Bar & Line Charts** - a bar or linechart summarizing values for each of many objects; includes histograms to show frequencies of values. (Called "Histogram" in earlier versions of Mesquite.)
- **Scattergram** - a plot showing each object's value in two variables

Most of Mesquite's charts are available through the first few submenus of the Analysis menu.

We suspect the greatest challenge to the user will be learning what chart to choose and how to set it up. For instance, how can one create a chart summarizing the estimated rate of a character's evolution according to each of a series of trees? Does one choose New Bar & Line Chart For>Characters because the value concerns a character? No, one chooses New Bar & Line Chart For>Trees because the numerous objects being summarized are trees, as the question concerns just one character but many trees. For each tree, what is being calculated is a value that relates to a character.

## Contents

- [Chart Wizard](#)
- [Selection of objects in charts](#)
- [Auto-recalculation](#)
- [Bar & Line Charts](#)
  - Example: [Robustness of estimated bias in character evolution](#)
  - Example: [Compositional bias along a sequence](#)
  - [Calculating and formatting options](#)
- [Scattergrams](#)
  - Example: [Canonical Variates Analysis](#)
  - Example: [Correlation between variability and hydrophobicity](#)
  - [Calculating and formatting options](#)

## Chart Wizard

The first item in the Analysis window, Chart Wizard, helps you make charts. It asks you a series of questions to determine what type of chart you want. We provide this because it is sometimes difficult to decide how to start building a chart. Alternatively, you can use the New Bar & Line Chart for and New Scattergram for submenus to build a chart.

## Selection of objects in charts

Most bar & line charts and scattergrams depict the values of objects— characters, taxa, trees – that can be selected. If you select these objects elsewhere in Mesquite, for example by selecting a column (character) in the Character Matrix Editor, then this selected objects will be highlighted in the chart. You can select the objects directly in the chart by clicking and dragging with the arrow cursor.

When objects are selected in the chart, and Copy is selected, then a list of the selected data points is copied to the clipboard. Otherwise if no objects are selected, then Copy puts the list of all data points into the clipboard.

## Auto-recalculation

By default, charts are recalculated whenever Mesquite detects that the data or assumptions underlying the chart have changed. If the chart calculation takes a long time, then this can lead to many delays if you need to make many changes in the data or assumptions. You can temporarily turn off the automatic chart recalculation by deselecting the Auto-recalculate menu item in the Chart or Scattergram menu. You can also request to Force Recalculation in the Chart menu.

## Bar & Line Charts

The bar & line charts available via the Analysis menu are:

- **Characters** – the items whose values are summarized are characters. These could be all of the characters of a matrix, or a series of simulated or randomized characters.
- **Character Matrices** – the items whose values are summarized are whole character matrices. These could be all of the matrices stored in a file, or a series of simulated or randomized matrices.
- **Taxa** – the items whose values are summarized are taxa within a taxa block.
- **Tree blocks** –the items whose values are summarized are blocks of multiple trees. That is, each item is a block of several trees. These could be all of the tree blocks stored in a file, or a series of simulated or randomized tree blocks.
- **Trees** – the items whose values are summarized are trees. These could be all of the trees within a tree block, or a series of simulated or randomized trees.

## Tables as output

In addition to the graphical chart, you can obtain a text table representation of the chart in several ways:

- Select Copy in the Edit menu of the chart to copy a list of point values to the clipboard; you can then copy it in to your favorite text or spreadsheet program
- Select Save Window as Text from the File menu to save a full text representation of the chart
- If Record Table to File in the Chart menu is selected, then a text file of the results

will be saved each time a chart is recalculated. This is particularly useful for some calculations that result in more than just a single value per item charted; in this case you will be asked when you start the chart whether you want to save the record.

## Examples

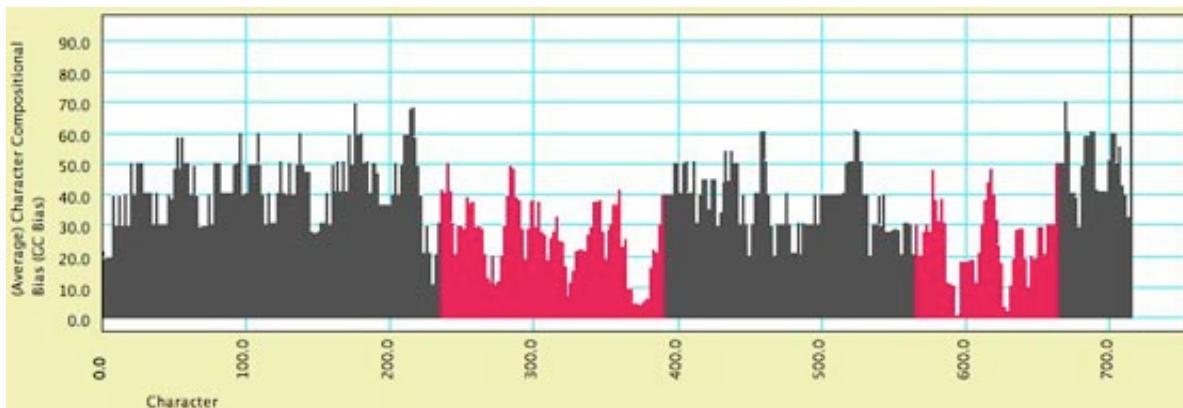
**Robustness of estimated bias in character evolution** – Suppose a biologist estimated the bias in the rates of gains versus losses in a character's evolution on a given tree. How might the estimate depend on accuracy of the tree's branch lengths? To answer this, one could see how the estimate varies when noise is added to the branch lengths of the given tree. First have a tree window available with the given tree showing. Then:

- Select the menu item Analysis>New Bar & Line Chart For>Trees
- A dialog box with heading "Source of trees (Trees chart)" should appear. Choose **Randomly Modify Current Tree**.
- In the dialog box "Random modifier of tree", choose **Add Noise to branch lengths**. You will then be asked to indicate the variance of the noise, and the number of trees to chart.
- In the dialog box "Value to calculate for trees" should appear. Choose **Tree Value using character**, because the desired value depends on the tree and a character.
- In the dialog box "Value to calculate for tree with character", ask to show secondary choices then choose **Forward/Backward rates**
- In the dialog box "Source of characters (for Forward/Backward rates)", choose **Stored Characters** (presuming you already have your character of interest entered in a data matrix).
- You may be asked other questions, depending on whether your data file includes multiple matrices. Then, a chart should appear to answer your query. (The calculation may take while.)

**Compositional bias along a sequence** – A biologist with DNA sequence data wants to see how the relative frequencies of A, C, G and T vary along the length of the sequence. To see this:

- Select the menu item Analysis>New Bar & Line Chart For>Characters
- In the dialog asking for a source of characters, choose **Stored Characters** (assuming you have your DNA matrix in the file).
- In the dialog "Value to calculate for characters (for Character Values Chart)", ask to show secondary choices then choose **Character Compositional Bias**
- The chart may initially appear uninteresting, but adjust as follows:
  - Select Chart>Orientation>Values (Y) by Items (X) to cause the chart to show the characters lined up, in sequence, along the X axis.
  - The chart will probably be set to automatically group into categories along the X axis. Try a moving window analysis by selecting Chart>Grouping on X>Moving Window..., and indicating the width of the moving window and the offset between adjacent window positions. The defaults are 5 and 1 respectively, but you could also try 10

and 2 to smooth further.



Above is an example of how the chart may appear. Some sections of the chart are red because those characters were indicated as belonging to a distinct [character group](#) or partition. In this chart, the introns (marked in red) have a stronger AT bias.

## Calculation and Formatting options

The following menu items can be found in the Chart menu:

- **Orientation**
  - **Number of Items (Y) by Values (X)** – This is the typical "histogram" where the vertical axis shows how many items have the various values arrayed along the X axis. Thus, the X axis represents the value, the Y axis the number of items.
  - **Value (Y) by Items (X)** – This displays the data with items arrayed in sequence along the X axis, and the Y axis representing the value for each item. This may be appropriate for items, like characters (sites in a DNA sequence), which have a natural ordering to them.
- **Grouping on X** – The X axis may be grouped into categories, such that the values falling within a range on the X axis are summarized in a single bar. This submenu controls any grouping.
  - **Automatic** – Mesquite chooses automatically how to group on the X axis
  - **No grouping** – The X axis is not grouped, and thus each item or object (each character, tree, etc.) is represented by a separate bar.
  - **Fixed number of groups** – The X axis is divided into a specified number of groups.
  - **Fixed Group width** – The X axis is divided into groups of a chosen width.
  - **Moving Window** – The X axis is divided into overlapping groups of a chosen width. This serves to smooth the chart by averaging over adjacent values. There are two parameters to set: the width of the moving window, and its increment. The increment is the offset between the starting edge of adjacent moving window positions.
- **Analysis** – In this submenu could be various analytical tools. Two standard choices are **Display Mean** and **Percentiles...**, which display the mean value and tails of the

distribution.

- **Show Average For Group** – When the chart is in the orientation Values by Items, and there is grouping along the X axis, then each bar may represent several objects (characters, etc.). This menu item allows you to choose whether the Y axis should show the sum of the values of those objects, or their average value.
- **Show Individual Points in Text** – When the chart has grouping along the X axis, then the text view of the window by default shows sums or averages of the groups. If instead you want the text view to give all of the values for the original objects shown by the chart, then select this menu item.
- **Show as Bar Chart** – By default, the chart is shown as a bar chart. However, if this option is turned off, the chart will be shown as a series of dots connected by a thin line.
- **Cumulative Mode** – By default, the chart is not shown in Cumulative mode. However, by choosing items in the Cumulative submenu of the chart, Mesquite will present the cumulative values at each point:
  - **Off** – Not cumulative
  - **Simple Cumulative** – The value of each item is the sum of the values of all previous items
  - **Average Cumulative** – The value of each item is the average of the values of all previous items
  - **Reverse Simple Cumulative** – The value of each item is the sum of the values of all following items
  - **Reverse Average Cumulative** – The value of each item is the average of the values of all following items

## Scattergrams

The scattergrams available in the Analysis menu are:

- **Characters** – each point in the plot represents a character. These could be all of the characters of a matrix, or a series of simulated or randomized characters.
- **Taxa** – each point in the plot represents a taxon within a taxa block.
- **Trees** – each point in the plot represents a tree. These could be all of the trees within a tree block, or a series of simulated or randomized trees.
- **Nodes** (Available under New Chart for Tree when a tree window is foremost) – each point in the plot represents a node in the tree of the tree window.

These scattergrams show the values of two variables for the objects of concern. For some scattergrams, a choice is given as to whether the **Same** or **Different** calculations should be shown on the two axes. By "Same" is meant that the same calculation is done but with a different parameter value. For instance, if the scattergram is a Taxa scattergram, it could show the character state in continuous character 1 on the Y axis, and the state in character 2 on the X axis. These represent the same calculation (continuous character state value),

differing only in the character used. By "Different" is meant an entirely different calculation, such as the asymmetry of a tree on the Y axis and its likelihood score for a character on the X axis.

## Examples

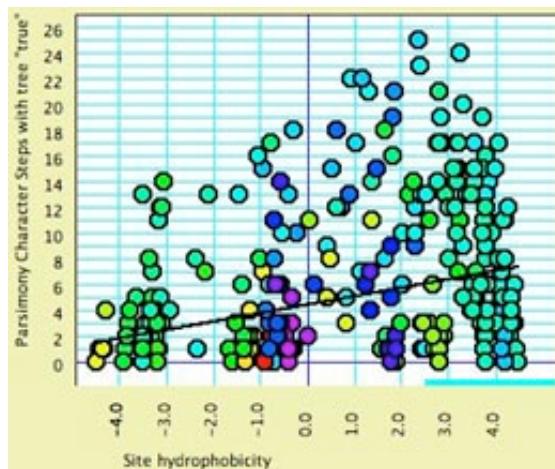
**Canonical Variates Analysis** – For a sample of specimens measured for a series of variables, how can the measurements be combined to best distinguish predefined groups? Multivariate analyses such as these can be done using modules in the Rhetenor package. Each specimen will be treated as a taxon. A continuous data matrix of the measurements should first be entered, and the taxa be assigned to [groups](#). Next:

- Select [Analysis>New Scattergram For>Taxa](#)
- In response to the query about same or different calculations, chose **Same**
- If asked, indicate you want to value for taxa to be **Continuous States of Taxon**. Otherwise, in response to the dialog box "Source of characters (for Continuous States of Taxon)", ask to show secondary choices and choose "**Characters from Ordinations**".
- In the dialog box "Source of matrices to be ordinated", select **Stored Matrices**.
- In the dialog box "Ordination (for Characer Source)", select **Canonical Variates Analysis**. (You may need to ask for secondary choices)

**Correlation between variability and hydrophobicity** – Do amino acid positions in proteins tend to evolve more quickly or more slowly depending on how hydrophobic they typically are? Mesquite does not yet have direct way to estimate rate for protein characters, but we can approximately compare relative rates by comparing the number of parsimony steps for characters on a tree. First, begin with a file containing a protein data matrix and an open tree window showing a tree. Next:

- Select [Analysis>New Scattergram For>Characters](#)
- In the dialog box "Source of characters (For Characters scattergram)" choose **Stored Characters**.
- In response to the query about same or different calculations, chose **Different**
- On the X axis we will put hydrophobicity. Thus, in the dialog box "Values for X axis", ask for secondary choices and choose **Protein Site Property**.
- In the dialog box "Property of Amino Acid" choose **Kyte & Doolittle Hydrophobicity**.
- In the dialog box "Values for Y axis", choose **Character Value with Current Tree**. In the dialog box "Value to calculate for character" chose **Parsimony Character Steps**. Indicate you want **Current Parsimony Models** to be used in the parsimony calculations.

The following scattergram shows the results of such an analysis, with two additions. First, the dots are colored by a third variable, the mean molecular weight of amino acids at that site. This can be done by selecting Color by Third Value from the Color menu, and in the dialog box "Values by which to color spots in the scattergram" asking for secondary choices.

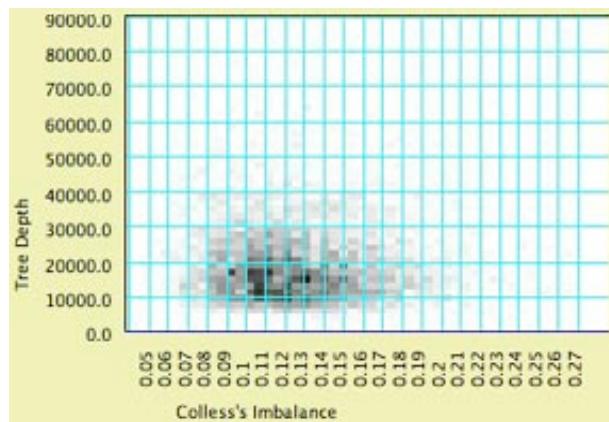


Second, the analysis assistant Scattergram Regression Diagnostics (part of the [PDAP](#) package) is in use, and shows the regression line. The text view of the window shows the details of the analysis. The correlation is highly significant.

## Calculating and formatting options

The following menu items can be found in the Scattergram menu:

- **Marker Size** – allows you to choose the size of the dots of the scatterplot
- **Analysis** – In this submenu could be various analytical tools. For instance, if [PDAP](#) is installed, you can request regression and correlation analysis under [Scattergram>Analysis>Other Choices](#).
- **Special Effects**
  - **Show Density** – this shades the background of the chart according to the density of points. To see this, you may want to turn off "Show dots" (see below). Here is an example:



- **Join the Dots, Join Last to First, Thick Joints** – These control whether and how a line is drawn between dots of the scattergram. These options are used to indicate molecular structure as in the cytochrome B example in Mesquite\_Folder/examples/Molecular/06-cytochromeB.nex

- **Show dots** – determines whether or not the dots representing objects in the scatterplot are drawn individually

In addition, if the scattergram is of characters, a **Color** menu will appear that allows you to color the dots according to a third value of the characters.

---

# Scripting and Macros

Mesquite's scripting system allows modules and other objects to be scripted using text commands. Mesquite uses scripting by text commands as follows:

- When a menu item, button, or tool is used, Mesquite sends a text command to the appropriate object, informing it of the use. This means that most user interface commands can also be executed in scripts and macros.
- When Mesquite re-reads a NEXUS file, it attempts to return windows and analyses to approximately the same condition as when the file had been saved. It does this by reading scripting instructions in a **MESQUITE** block that Mesquite had written into the file when saved. Thus, Mesquite writes scripts called "Snapshots" to later instruct itself.
- Macros are script files which, if placed in the recognized directories, appear in submenus as selectable menu items. When selected, the macro is executed. Mesquite recognizes directories called "macros" within the Mesquite\_Folder directory and the "Mesquite\_Prefs" subdirectory of the Mesquite\_Support\_Files directory.
- Not all scripts in **MESQUITE** blocks need to be written by Mesquite. Users can write **MESQUITE** block scripts by hand.
- Users can send a script to a window using [Window>Scripting>Send Script](#). This page gives some [simple examples](#) for the tree window.

The text commands are sent to modules or other objects. The particular text commands to which a module or other object responds will depend on the object. When Mesquite shows a web page for the first time, it attempts to compile automatically documentation for scripting commands that is available. This includes commands predefined by the scripting language, and commands belonging to particular modules. This compiled documentation is available from the web page linked from the "Scripting Commands" menu item in the Help menu when Mesquite is running.

Mesquite's scripting language is not as human-friendly as it might be at present, especially in its handling of variables and aspects such as the lack of "else" statements. Since the vast majority of Mesquite scripts won't be written by humans but by Mesquite itself, that is not such a liability as it might seem. We hope to reform the scripting language in the future.

## Macros

Scripts performing some special function can be written and distributed by programmers to end users as "Macros". For some calculations or display functions, the number of different analyses a user might like to do are too many to be easily supported by a graphical user interface. For instance, the user might like to print the ancestral state reconstruction of a character using a series of different stepmatrices. One can invent many such scenarios in which something repetitious is needed, and a module would be hard-pressed to maintain each as an option. Thus, small scripts using Mesquite's scripting language can be written and placed in the

"macros" folder. These macros will appear as options in an appropriate place, depending upon where they are applicable. When the user selects the menu item, the script is executed.

Macros can also be automatically composed by Mesquite. As noted above, Mesquite composes a script to place in NEXUS files that it saves. This script applies only to the particular saved file. In some contexts, you can ask Mesquite to save a script as a macro file. This is currently available in only a few contexts. Save Window as Macro in the Analysis menu and Window menu save macros that attempt to reproduce the condition (including analyses) of the foremost window. This is useful to reproduce a complex chart with a different data file, for example. Save Macro For Tree Drawing in the Drawing menu of a Tree Window creates a macro by which you can later reproduce the current appearance of the tree drawing (background color, font, tree form, orientation, and so on). Save Tree Analysis As Macro in the Tree menu of a Tree window creates a macro by which to reproduce a particular analysis with the tree. Macros so created are stored in *Mesquite\_Support\_Files/Mesquite\_Prefs/macros*, and could be (for instance) shared by users.

## Learning about scripting

Users can learn about the scripting language from this web page, but also by inspection of existing scripts. Some sources of scripts to examine are:

- The **MESQUITE** block within NEXUS files saved by Mesquite. If you want to know how to script a tree window to trace a character, for instance, open a tree window, turn on character tracing, and save the file. You can look at the file with a text editing program to see how such a script would be written
- If there are any **macros** installed, you can examine the files that specify them. These files are simple text files with scripts. Macros would be found in folders called "macros" within the *Mesquite\_Folder* directory, and in the macros folder of the "*Mesquite\_Prefs*" directory of the *Mesquite\_Support\_Files* directory.
- The menu item Windows>Scripting>Show Snapshot shows a **snapshot** for the window – the script that would be needed to set the window to its current state.

Another way to learn about the language and particular scripting commands is to look at the web pages linked from the "**Scripting Commands**" menu item in the Help menu when Mesquite is running. In particular, examine the page on universal commands, which shows basic commands available regardless of the object being scripted, including the basic flow control and variable-defining commands of the system. (We cannot provide a link here to this web page because this web page is created by Mesquite when it runs, and is placed in a location that depends on your particular computer. After running Mesquite once, you may want to find some of these files and save a bookmark/alias/shortcut to them – the file on basic scripting commands is named 'puppeteer.html').

## Commands within scripts

In general, a command found within a script takes the following form:

```
commandName argument1 argument2 ...;
```

The commandName is a single word (token); the arguments can be multiple tokens, so long as the module knows how to interpret them. Typically each argument is a single token (though this may be string of multiple tokens converted to a single token by quotation).

## Scripts

A script consists of a series of commands. At each stage in the script, there is an implicit recipient of the command given (for instance, a module or a window). For instance, here is a script within the MESQUITE block of a NEXUS file. To the right of the commands are comments to explain them.

```
Begin MESQUITE;
getNumberOfDataSets;           [file coordinator queried for number of data sets]
Integer.dataSets *it;          [storing number of data sets in integer variable 'dataSets']
getEmployee 'Data Window Coordinator'; [querying for reference to Data Window Coordinator module]
tell It;                      [commands to follow will be sent to the Data Window Coordinator]
  Integer.dataNum 0;           [Define integer variable 'dataNum' and assign it 0]
  for *Integer.dataSets;       [for loop; cycle as many times as there are data sets]
    showDataWindow *Integer.dataNum; [commands to make a data window for dataset]
    tell It;                   [commands to follow will be sent to module that makes data window]
      showWindow;              [tells the module to make the data window visible]
    endTell;                  [finished sending commands to the module that makes the data window]
    increment.dataNum;         [add 1 to the variable 'dataNum']
  endFor;                     [end of the for loop]
endTell;                      [finished sending commands to the Data Window Coordinator]
getNumberOfTaxes;             [file coordinator queried for number of sets of taxa]
Integer.taxaSets *it;          [storing number of sets of taxa in integer variable 'taxaSets']
getEmployee 'Tree Window Coordinator'; [querying for reference to Tree Window Coordinator module]
tell It;                      [commands to follow will be sent to the Tree Window Coordinator]
  Integer.taxaNum 0;           [Define integer variable 'taxaNum' and assign it 0]
  for *Integer.taxaSets;       [for loop; cycle as many times as there are sets of taxa]
    makeTreeWindow *Integer.taxaNum; [commands to make a tree window for set of taxa]
    tell It;                   [commands to follow will be sent to module that makes tree window]
      getTreeWindow;            [queries the module to return a reference to the tree window itself]
      tell It;                 [commands to follow will be sent to the tree window]
        newAssistant 'Trace Character History'; [the tree window is asked to hire a module]
      endTell;                 [finished sending commands to the tree window]
      showWindow;              [tells the module to make the tree window visible]
    endTell;                  [finished sending commands to the module that makes the tree window]
    increment.taxaNum;         [add 1 to the variable 'taxaNum']
  endFor;                     [end of the for loop]
endTell;                      [finished sending commands to the Tree Window Coordinator]
END;
```

This MESQUITE block causes Mesquite to show a data window for each of the data sets, and a tree window for each of the Taxa blocks; the tree windows are shown with a character traced.

This script illustrates some of the features of Mesquite's scripting language:

- There is a **current object** to which commands are sent. Which object is being commanded can be changed by a "tell" command. Initially, the implicit recipient of the commands is the FileCoordinator of the file in question.
- Any given command may **return an object** that is stored within the variable referred to by "It". Thus, in the above script, the makeTreeWindow returns the module hired to

supervise the tree window. The immediately following command, "tell It" thus shifts the recipient of commands to this tree window module.

- The scripting language has **variables**. They are defined or their values set by commands beginning, for instance, Integer.myIntegerName or Object.myObjectName. Their values can be utilized by prefixing their names by an asterisk, for instance \*Integer.myIntegerName.
- The scripting language has **control flow**, including **if**, **for** loops, and **while** loops.

Some of the lines of this script are commands unique to particular commandable objects (getNumberDataSets, getEmployee, showDataWindow, etc.); others are predefined by the scripting language (e.g., Integer, tell, for). More details on particular commands can be found in the web pages linked from the "Scripting Commands" menu item in the Help menu when Mesquite is running. Some details are found below.

## Variables: Integers, Strings and Objects

Three sorts of variables are supported. Reference to each requires the type of variable with name appended, as in "Integer.numberOfCharacters" or "Object.treeDrawCoordinator". The generic variable "it" refers to the object last returned by a command.

When the variable is passed as an argument for a command, it should be preceded by an **asterisk**. This allows the system to know that a variable, and not a constant string, is being passed.

### Numerical variables

Two sorts of numerical variables are supported: Integer and Number. The former contain whole numbers. The latter can contain whole or decimal numbers. Reference to each requires the type of variable with name appended, as in "Integer.numberOfCharacters" or "Object.treeDrawCoordinator". The generic variable "it" refers to the object last returned by a command.

When the variable is passed as an argument for a command, it should be preceded by an asterisk. This allows the system to know that a variable, and not a constant string, is being passed. The commands concerning variables are:

- **Integer.[name] [number];**— this declares an integer variable of name "name" and assigns it the value given by the number. If the variable already exists, its value is replaced by the number. The number may be represented by a constant, as in "Integer.counter 5", by an integer variable, as in "Integer.counter \*Integer.previousCount", or by a String variable that contains a string that can be parsed into an integer, as in "Integer.counter \*String.countString". If the number is indicated as "random", a random integer will be placed in the variable.
- **increment.[name of integer];** — this adds one to the integer's value
- **decrement.[name of integer];** —this subtracts one from the integer's value.

- **Number.[name] [number];**— this declares a numerical variable of name "name" and assigns it the value given by the number. If the variable already exists, its value is replaced by the number. The number may be represented by a constant, as in "Number.rate 0.5", by an integer variable, as in "Number.rate \*Number.previousRate", or by a String variable that contains a string that can be parsed into a number, as in "Number.rate \*String.rateString". If the number is indicated as "random", a random number between 0 and 1 will be placed in the variable. If the number to be placed into the Number variable is preceded by a '+', the number doesn't replace the existing value of the Number variable, but is added to it (similarly for '-').

## String variables

One sort of variable contains a string of text. The command to define and assign values to a string variable is:

- **String.[name] [string];** – this declares a String variable of name "name" and assigns it the value given by the string. If the variable already exists, its value is replaced by the string, unless the string passed to it is preceded by "+" in which case it is appended to the existing string. The string passed may be represented by a literal string, as in "String.name John A. MacDonald", by an String variable, as in "String.name \*String.name.firstPM", or by an Object variable, in which case the name of the Object will be used.

## Object variables

One sort of variable contains objects (such as modules, or windows). The command to define and assign values to an object variable is:

- **Object.[name] [reference to object];** – this declares an Object variable and assigns it the object indicated. If the variable already exists, its value is replaced. The reference may be "it", as in "Object.thisModule \*It", or an Object variable, as in "Object.thisModule \*Object.storedModule".

## The variable "It"

Standard Mesquite Commands to modules return an object. In the scripting language, this returned object is stored in the variable "It". Thus, after a command "getNumberDataSets" to the FileCoordinator, the FileCoordinator returns an Integer variable containing the number of data sets. This can be stored in an Integer variable by following the command by "Integer.numDataSets \*it". Likewise, "tell" often makes use of "it".

## Flow and command control

As noted above Mesquite's scripting language has flow control as well as control of the object to be commanded.

## Using "tell" to direct commands

Commands are directed toward commandable objects, including modules, windows, and others. Since different objects might use the same command names, the object to which a command is directed must be indicated. In the scripting language, at any point there is an implicit object being commanded. Subsequent commands are directed to a different object using the "tell" command, which must be balanced by "endTell". At the root level, the FileCoordinator is being commanded.

## Flow control

Flow control statements include "if", "for", and "while". Others are available (such as ifnot, stop, exitTell). Details on these can be found via the web page shown by selecting **Scripting Commands** from the **Help** menu while Mesquite is running.

- **if** [integer or integer variable]; ... **endif**; – the statements between if and endif are executed if the integer variable is non-zero
- **for** [integer or integer variable]; ... **endFor**; – the statements between for and endFor are executed as many times as the initial value of the integer variable.
- **while** [integer or integer variable]; ... **endWhile**; – the statements between while and endWhile are executed repeatedly as long as the value of the integer variable is non-zero.

## Debugging

There are a number of commands that are useful for debugging. For instance, if the Command "debug" is placed in the block, a debugging mode will be enabled which reports in the console more details about the commands as they are executed. More information about such commands can be found by selecting the "**Scripting Commands**" menu item under the Help menu when Mesquite is running.

## Examples

Here are three simple example scripts that you can send to the Tree window using Window>Scripting>Send Script. Open a tree window, turn on Trace Character History, and then paste one of these scripts into the Send Script dialog box. **The first script scrolls from tree to tree, for each recording in a file "results.txt" the reconstruction of ancestral states.**

```
String.resultsFile 'results.txt';
saveMessageToFile *String.resultsFile 'RESULTS with different trees';
appendReturnToFile *String.resultsFile;
getWindow;
tell It;
    getNumTrees;
    Integer.numReps *It;
    ifNotCombinable *Integer.numReps;
        Integer.numReps 10; [in case indefinite number of trees]
    endIf;
```

```

endTell;

Integer.count 0;
for *Integer.numReps;
    increment.count;
    getWindow;
    tell It;
        setTreeNumber *Integer.count;
    endTell;
    getEmployee #mesquite.ancstates.RecAncestralStates.RecAncestralStates;
    tell It;
        getLastResult;
        String.result *It;
        appendMessageToFile *String.resultsFile *String.result;
        appendReturnToFile *String.resultsFile;
    endTell;
endFor;

```

**This second script scrolls from character to character, for each recording the ancestral states in a file "results.txt":**

```

String.resultsFile 'results.txt';
saveMessageToFile *String.resultsFile 'RESULTS with different characters';
appendReturnToFile *String.resultsFile;
getEmployee #mesquite.ancstates.RecAncestralStates.RecAncestralStates;
tell It;
   getNumHistories;
    Integer.numReps *It;
    ifNotCombinable *Integer.numReps;
        Integer.numReps 10; [in case indefinite number of characters]
    endif;
endTell;

Integer.count 0;
for *Integer.numReps;
    increment.count;
    getEmployee #mesquite.ancstates.TraceCharacterHistory.TraceCharacterHistory;
    tell It;
        setCharacter *Integer.count;
    endTell;
    getEmployee #mesquite.ancstates.RecAncestralStates.RecAncestralStates;
    tell It;
        getLastResult;
        String.result *It;
        appendMessageToFile *String.resultsFile *String.result;
        appendReturnToFile *String.resultsFile;
    endTell;
endFor;

```

**The third script scrolls from tree to tree, printing each one.**

```

getWindow;
tell It;
    getNumTrees;
    Integer.numReps *It;
    ifNotCombinable *Integer.numReps;
        Integer.numReps 10; [in case indefinite number of trees]
    endif;
endTell;
for *Integer.numReps;
    getWindow;
    tell It;
        setTreeNumber *Integer.count;
        printToFit;
    endTell;
endFor;

```

## Technical Details

(See the [developer's documentation](#).) The Puppeteer is in charge of defining the basic language as it relates to variables and flow control. For an object to be scriptable, it must be of the Commandable interface. Commands internally in Mesquite are remembered in objects called MesquiteCommands.

---

# How Mesquite works

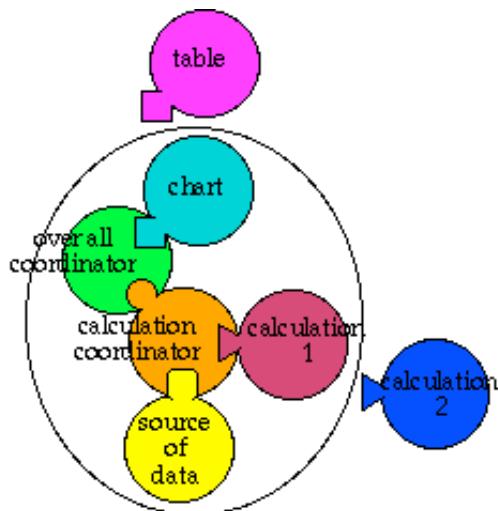
Mesquite operates via the cooperation of modules, each of which performs some function or duty. There is one core module, the **Mesquite trunk module**. When this module starts up, it finds all of the available modules. (Mesquite does this by looking into the subdirectories of the **mesquite** directory.) From each module, it gathers information, including its name and the functions the module promises to perform.

A programmer's account of Mesquite's modular architecture is given in the [developer's documentation](#).

## The tree of modules

Modules are "hired" by other modules as employees to perform certain tasks. Thus, the core Mesquite module hires a "File Coordinator" module to manage most activities, including coordinating the reading and analyzing of data files. The File Coordinator itself hires other modules to do the actual file reading and analyses. The file reading module may hire other modules to handle particular parts of the data file, one for the data matrix, another for the trees, and so on. With modules hiring modules hiring modules, the modules in use at any time are linked together in employee-employer relationships that as a whole take the form of a **tree**, with the root module being the core Mesquite trunk module. (To see the tree of modules active at any time, choose the menu item "Active Modules" from Mesquite's Help menu, or by touching the Modules tab in a window's information bar.)

Here is a diagram of a section of the module tree. The five modules in the oval are engaged in calculating and presenting some result to the user. (The two modules outside the oval are patiently waiting as alternatives.) The over-all coordinator module has hired a calculation coordinator module, which in turns uses one module to supply data and another to do the actual calculation. The over-all coordinator also hires a module to display the results (in this case, a charting module).



## Kinds of modules: Duties

A module will decide which other modules would be appropriate to hire for a particular task according to the **duties** that the various modules promise to perform. Each module belongs to a particular kind of module, and these kinds amount to a claim as to what duties will be performed. (Technically, these "kinds" are subclasses of the general Mesquite module class, and a particular module can choose which kind it is by what subclass it extends.) For instance, one "kind" of module is a DrawTree module. By belonging to this kind, a module promises that, when given a tree, it will draw it. There may be multiple DrawTree modules available -- one may make square trees ("phenograms"), another diagonal trees ("cladograms"), another circular trees, another may plot the nodes of the tree in a three dimensional space, and so on. The fact that a module is of the DrawTree kind guarantees to any employer module that it will perform a task in a predictable way, and be interchangeable with other modules of the DrawTree kind. If a module needs a tree drawn, it could choose any one of these DrawTree modules to hire. Modules can easily fire an existing employee, and hire a new one, to respond to a user's request to change the calculation or display.

## Consequences to the user: Flexibility and choice

This system of modules provides great **flexibility** for the user in designing calculations and output. The user can typically choose which particular module of a kind is used, and thus mix and match modules to construct an analysis. In the example of the illustration, the user might substitute the other calculation module (calculation 2) and the other graphical display module (table) and thus arrive at a different calculation and output. If, in the future, someone writes a new sort of calculation module (calculation 3), it can also be plugged in to the system as a new alternative. It can be used not only in the calculation diagrammed above, but in any other analysis in which a calculation of its basic kind is needed. Likewise, if someone writes a new sort of graphical presentation module, it can also be plugged in. This means that the number of alternative analyses and display modes available to the user rises multiplicatively as more modules of different kinds are available.

The diversity of analyses that can be done with Mesquite is valuable in many respects, not the least of which is that it helps overcome the constraining effect that computer programs tend to have on a user's work. However, it also forces the user to make **choices**: to think about exactly what he or she is doing. This may be an advantage or disadvantage. Of course, one could write an artificial intelligence module that advises the user what to do....

It is not just the diversity of analyses that makes Mesquite unusual. It is the fact that we cannot predict what features it will have, since its features depend on the modules currently installed and loaded. Thus, a simple manual for it is not possible. In a normal program we could say "If you want to trace a character, go to the Analysis menu and choose Trace Character. The branches will be shaded to show the parsimonious ancestral states". With Mesquite, the branches might not be shaded, because the reconstructed states might be

displayed in some other way, according to whatever display module is in use. The states might not be reconstructed by parsimony. The Trace Character command might not appear in the Analysis menu, since the Trace Character module might be hired by various other modules, and each of these employer modules might choose a different place (e.g., somewhere other than a Trace menu) to give the user the opportunity to choose analyses. We have attempted to describe standard analysis via the links at the left hand side of this page. Beyond those, however, the user can use the examples files for more information about setting up analyses.

---

## Managing modules

If you are installing third party modules, you may want to go first to the explanation of [possible places](#) to install the modules. There is a [complete list of modules](#) in the current version of Mesquite; this may be useful for searches to find whether a feature exists in Mesquite.

Users will be able to use Mesquite more effectively if they become familiar with the concepts of modules and how they interact. An understanding of the modular system will help the user more effectively give commands to Mesquite via [menus](#) and [scripting](#), and interpret the results obtained.

The page on [how Mesquite works](#) introduces modules and how they cooperate. In brief, modules link together to perform calculations and display the results to the user. The linkage of modules is in the form of a tree, with the root module being the "trunk" Mesquite module itself. Which modules are currently active depends on the files the program has read and the calculations the user has requested.

### Installed *versus* loaded *versus* running modules

An **installed module** is one available to be used: it is sitting as a file or a series of files on your disk or web server. Mesquite can find these modules and, if appropriate, can get them started up. Among the installed modules might be some that you never use, because you don't happen to need their calculations.

Not all installed modules are available to a user during a Mesquite run. The reason is that, although a module might be physically installed, it might not be **loaded** by Mesquite at startup. An installed module might not be loaded if, for instance, the user has chosen a configuration to control the loaded packages (see Activate/Deactivate Packages in the File manue). The configuration file might indicate that only some of the packages are to be loaded and available.

When Mesquite starts up, or when the user requests new windows, calculations, and so on, some of the loaded modules are started up to do the work. These are the **running modules**.

This characterization is a bit misleading, however, since many active copies of a single module on disk can be running at once. That is, the installed module on disk can be thought of as a blueprint for a machine, and each time Mesquite needs to start up a module, the machine is built and started. If Mesquite needs another machine of the same sort, it can build a new one from the blueprint. Many machines (running modules) from a single blueprint (loaded module) can be built and be running simultaneously.

(To the programmer, the loaded modules are the available classes that are subclasses of the basic module class; an active module is an object instantiated from one of these classes.)

### Where modules are installed

To be installed and found by the Mesquite system, a module must be within one of four places. We will describe each under the assumption that a package "mypackage" with a single module "MyModule" is being installed.

- the **mesquite** directory of **Mesquite\_Folder**: this is the primary location for modules. This is the only location modules can be located for use on Mac OS 9. If installed here, our example module would be at: Mesquite\_Folder/mesquite/mypackage/MyModule/MyModule.class
- in **Mesquite\_Folder/additionalMesquiteModules/mesquite**: this is an optional location for modules. If you are installing 3rd party modules, you might want to install them here, so that you can easily find them again and reinstall them when you upgrade to a newer version of Mesquite. If installed here, our example module would be at:  
Mesquite\_Folder/additionalMesquiteModules/mesquite/mypackage/MyModule/MyModule.class
- in **Mesquite\_Support\_Files/classes/mesquite**: this is an optional location for modules. It can be useful for users who don't have permission to modify the main Mesquite\_Folder but may want to install new modules. If installed here, our example module would be at:  
Mesquite\_Support\_Files/classes/mesquite/mypackage/MyModule/MyModule.class
- at the locations specified in **Mesquite\_Folder/classpaths.xml**. This file should have a format as follows:

```
<?xml version="1.0"?>
<mesquite>
    <classpath>../MyExtraModules</classpath>
</mesquite>
```

This tells Mesquite to look outside of Mesquite\_Folder, and inside MyExtraModules for the root of the classpath. If installed here, our example module would be at:  
MyExtraModules/mesquite/mypackage/MyModule/MyModule.class.

Furthermore, the active configuration must specify that the module's package be loaded. By default, all installed packages are loaded.

Thus, if you open up the Mesquite\_Folder, you might see the following

- com
- corejava
- docs
- documentation.html
- edu
- examples
- images
- Jama
- JSci
- lesser.txt
- pal
- mesquite. In this directory, you might see:
  - ancstates
  - assoc
  - basic

- o **batchArch**
- o **categ**
- o **charMatrices**
- o **charts**
- o **coalesce**
- o **configs**
- o **cont**
- o **distance**
- o **genesis**
- o **io**
- o **jama**
- o **jsci**
- o **lib**
- o **lists**
- o **Mesquite.class**
- o **minimal**
- o **molec**
- o **ornamental**
- o **pairwise**
- o **pal**
- o **parsimony**
- o **rhetenor**
- o **search**
- o **stochchar**
- o **treefarm**
- o **trees**
- o **trunk**

Each of the items in bold above is a directory (folder) that contains modules. Within **minimal**, you might see:

- **BasicFileCoordinator**
- **Defaults**
- **DrawHierarchy**
- **InterpretNEXUS**
- **ManageForeignBlocks**
- (and so on)

Each of these directories is for one module. For a module to be found and used by Mesquite, it must be present in a directory whose name is the same as the class file for the module (e.g., the class file for the module "Basic File Coordinator", **BasicFileCoordinator.class**, must be in a directory named **BasicFileCoordinator**).

## Modules currently in use

While Mesquite is running you may want to learn what modules are currently running. In the **Help** menu the **Active Module Tree** menu item displays a window that displays the modules currently

running. You can also see the modules responsible for any window by touching on the Modules tab in the window's information bar.

If you touch on the name of the module in these module tree displays, a menu will drop down that allows you to find out information about the module.

## Information pages and Manuals for modules

Mesquite gathers information about each module and summarizes it in an **information page** for each module. These are linked from the page available via the **Modules Loaded** menu item in the **Help** menu while Mesquite is running. As well, some individual modules have their own **manuals** in the form of web pages. These are also linked from the Modules Loaded web page.

---

## Mesquite's standard modules

The following is a list of the modules installed by default in this version of Mesquite, along with the explanations of what they do.

- ACGT Compositional Bias – Returns continuous characters which are the proportion G+C, and separately A, C, G, T, for each taxon in a DNA data set.
- Add item... – Adds an item in each cell of a continuous matrix.
- Add Noise (Gaussian) – Adds normally distributed noise to existing states. Only the first item of a continuous matrix is modified.
- Add Noise to Branch Lengths – Adds noise to branch lengths of tree. Noise is Normally distributed, with variance as selected. By default the variance is proportional to current branch length, so that if branch length is 10 and you've indicated a variance multiplier of 0.1, the noise added will have a variance of 1.0. Negative branch lengths are not allowed, and are changed to zero.
- Add Note On Tree – Draws editable notes in a tree display.
- Add Taxa To Tree – To a tree with some taxa excluded, adds taxa.
- Add text to taxon names – Adds text to taxon names.
- Add value – Alters continuous data by adding a value. All items of the matrix are similarly modified.
- Add/Delete Data – Adds and deletes characters and taxa in a Character Matrix editor. This can be done through the Add characters..., Add taxa..., and Delete Selected menu items and via the tools that add taxa and characters where touched.
- Adjust lineage width – Provides a tool with which to adjust lineage widths a tree window.
- Alignent Score To Ref. – Reports the Alignment Score for a taxon in a data matrix. The score for each taxon is the cost of aligning that taxon's sequence against the reference taxon using Mesquite's default pairwise aligner. Lower scores mean a better alignment.
- Align Package – Align is a Mesquite package providing tools for the alignment of sequence data.
- Align Score/RC Align Score To Ref. – Reports the Alignment Score / Reverse Complement Alignment Score for a taxon in a data matrix. The score for each taxon is the cost of aligning that taxon's sequence against the reference taxon using Mesquite's default pairwise aligner, divided by the score for aligning the reverse complement of the sequence against the reference. Scores greater than one mean that alignment is better if one of the sequences is reverse complemented .
- Align Sequences – Sends the selected sequence to be aligned.
- Align To Dropped – Supplies an alignment tool that can be used on a set of sequences. Sequences dropped by this tool on another sequence will be aligned to that other sequence (pairwise).
- All Branch Lengths to 1.0 – Assigns a value of 1.0 for branch length for all of a tree's branches.
- All Rerootings – Reroots tree at all nodes.
- All Unassigned Branch Lengths to 1.0 – Assigns a value of 1.0 for branch length for all of a tree's branches that have not yet had a branch length assigned.
- Alter Branch Lengths – Alters branch lengths of selected trees.
- Alter Data – Manages data-transforming modules.

- Alter Trees – Alters selected trees.
- Alternative Names – Lists the alternative names for the taxon.
- Amino Acid Molecular Weight – Returns molecular weight of amino acid
- Amino Acid Properties – Returns property of amino acid
- Among-group PCA – Performs among-group principal components analysis on a continuous-valued matrix.
- Ancestral States All – Reconstructs ancestral states at nodes for all characters supplied by character source.
- Ancestral States Reconstruction Package Introduction – Serves as an introduction to the ancestral states reconstruction package for Mesquite.
- Annotate Branches – Supplies a tool for tree windows to attach and view footnotes for branches.
- Annotation Panel module (list of characters) – Provides tools with which to attach notes (including images) to characters and show them.
- Annotation Panel module (list of taxa) – Provides tools with which to attach notes (including images) to characters and show them.
- Annotation Panel module – Provides tools with which to attach notes (including images) to cells of the matrix and show them.
- Append numbers – Appends to each taxon name the number of the taxon. This will ensure that each taxon name is unique.
- Arbitrarily Resolve State Ambiguities – Alters categorical data by arbitrarily resolving ambiguous sites to one of the possible states.
- Assign All Branch Lengths – Assigns a value for branch length for all of a tree's branches.
- Assign Selected Branch Lengths – Assigns a value for branch length for all of a tree's selected branches.
- Assign taxon names – Assigns to each taxon name a string followed by a number. For example, if the string is "g", then taxon 3 will be assigned the name g3.
- Assigned Colors – Provides a tool with which to color cells of a matrix.
- Associated Taxa – Lists what other taxa (e.g. contained or containing) are associated with these.
- Asymmetrical 2-param. Markov-k Model – Defines and maintains simple Markov k-state asymmetrical 2-parameter stochastic models of character evolution.
- Asymmetry Likelihood Ratio Test – Calculates the test statistic for the likelihood ratio test comparing the asymmetrical and one parameter models  $[2\ln(L(\text{Asymm.})/L(\text{Mk1}))]$ , on a tree for a given character.
- Attachment – Displays an item attached to trees.
- Augment Tree Randomly – Augments tree by random placement of excluded taxa. There are three modes: adding only to original branches, ignoring lengths; adding to any branch, considering branch length; adding to any branch, ignoring branch length. With lengths considered, probability of placing taxon on a branch is proportional to the branch's length.
- Average of Several Distances – Average of several distances among taxa, e.g. average from several data matrices.
- Average Taxon Value among Trees – Averages among trees a value calculated for a taxon using a tree.
- Average Tree Value – Calculates the average of some value for trees in a tree block.
- Balls & Sticks – Draws trees with spots and the nodes and thin lines for branches.
- Bar & Line Chart – Helps make bar & line charts.

- Bar & Line Chart for Character Matrices – Shows chart displaying values (such as treelength, etc.) for a series of character matrices.
- Bar & Line Chart for Characters – Shows chart displaying values (such as likelihoods, parameter values, parsimony counts, etc.) for a series of characters.
- Bar & Line Chart for Taxa – Makes a chart displaying values (such as proportion of gaps in a sequence, etc.) for each of a series of taxa.
- Bar & Line Chart for Trees – Makes a chart displaying values (such as likelihoods, parsimony scores, imbalance statistics, correlations,etc.) for a series of trees.
- Basic Draw Names for Tree Display – Draws taxon names on a tree. Chooses orientation of names according to orientation of tree.
- Basic File Coordinator – Coordinates the reading, maintenance and linkages of files, which include Taxa, Trees, CharacterData, and other objects. This module is hired directly by the Mesquite trunk module.
- Basic Tree Draw Coordinator – Coordinates the drawing of a tree by maintaining the basic TreeDisplay and by hiring a DrawTree module.
- Batch Architect Package Introduction – Serves as an introduction to the Batch Architect package for Mesquite.
- Batch File Template Manager – Manages templates used in saving batch files.
- Birth/Death Process Trees – Generates tree by simple birth/death model with a constant rate of speciation (birth) and of extinction (death).
- BiSSE Ln Likelihood Difference – Calculates the difference in log likelihoods between two BiSSE speciation/extinction models.
- BiSSE Net Diversification Likelihood (Calculator) – Calculates likelihood with a tree of a species diversification model whose rates ( $r = \text{spec.} - \text{ext.}$ ;  $a = \text{spec./ext.}$ ) depend on the state of a binary character (BiSSE model, Maddison, Midford & Otto, 2007).
- BiSSE Net Diversification Likelihood – Calculates likelihood with a tree of a species diversification model whose rates ( $r = \text{spec.} - \text{ext.}$ ;  $a = \text{spec./ext.}$ ) depend on the state of a binary character (BiSSE model, Maddison, Midford & Otto, 2007).
- BiSSE Speciation/Extinction Likelihood – Calculates likelihood with a tree of a species diversification model whose speciation and extinction rates depend on the state of a binary character (BiSSE model, Maddison, Midford & Otto, 2007).
- BiSSE Speciation/Extinction Likelihood Calculator – Calculates likelihood with a tree of a species diversification model whose speciation and extinction rates depend on the state of a binary character (BiSSE model, Maddison, Midford & Otto, 2007).
- BiSSE Trees & Characters – Generates tree by a speciation/extinction model in which a character controls rates of speciation/extinction. Time to next event drawn from negative exponential distribution.
- BLAST in Web Browser – BLASTs selected data.
- Boolean for Tree (in List of Trees window) – Supplies booleans for trees for a trees list window.
- Boolean Tree Value – Gives a tree a value of 1 if it satisfies the criterion, 0 otherwise.
- Bootstrap Resample Matrix – Resamples characters from a matrix for use in bootstrap methods.
- Branch Information – Provides a tool that shows information about branches.
- Branch Length from Root – For each taxon, calculates the sum of the branch lengths from that taxon to the root of the tree.
- Branch Lengths Adjust – Provides a tool to adjust branch lengths of trees.
- Brownian Model – Initializes Brownian motion character model for likelihood and other probability calculations.

- Calculated value for taxon – Supplies values for taxa reinterpreted as if characters.
- Canonical Variates Analysis – Performs canonical variates analysis on a continuous-valued matrix.
- Cell Wand (data) – Provides a tool with which to select cells in a matrix automatically.
- Character Compositional Bias – Calculates the percent of taxa with particular nucleotides (GC bias, or individual frequency of A, C, G or T) for a character.
- Character Inclusion – Shows current character inclusion in character list window.
- Character Likelihood – Calculates the negative log likelihood of a tree for a given character.
- Character List – Makes windows listing characters and information about them.
- Character Loadings – Returns loadings of a character in an ordination
- Character Matrices List – Makes windows listing character data matrices and information about them.
- Character Matrices Package Introduction – Serves as an introduction to the basic character matrices package for Mesquite.
- Character Matrix Editor – Makes editor windows to edit character data.
- Character Matrix manager – Coordinates the management of character data matrices.
- Character matrix selection coordinator – Coordinates selecting taxa and characters in data matrix.
- Character Models List – Makes windows listing character models and information about them.
- Character Reference Strip – Shows the strip at bottom of matrix with character reference.
- Character selection coordinator – Coordinates character selection.
- Character Sort (data) – Provides a tool with which to sort characters automatically.
- Character Source – Coordinates the supply of characters from various sources of characters.
- Character Source – Coordinates the supply of characters from various sources of characters.
- Character value using other character – Coordinates the calculation of a number for a character based on another character (e.g., a correlation between the two characters).
- Character value with current tree – Coordinates the calculation of a number for a character based on a current tree.
- Character value with respective tree – Coordinates the calculation of a number for a character based on a respective tree.
- Character value with tree – Coordinates the calculation of a number for a character based on a tree.
- Character Wand (data) – Provides a magic wand tool with which to select characters automatically.
- Character-Associated Diversification – Coordinates analyses for the effect of a character on diversification (speciation/extinction).
- Character/State Information – Exports character and state information for categorical matrix.
- Characters from Matrix Source – Supplies characters from source of matrices.
- Characters from Ordinations – Supplies a character from an ordination of an existing matrix.
- Characters Scattergram – Makes a scatterplot displaying two values (such as likelihoods, parsimony scores, etc.) for a series of characters.

- Chart from Instruction File – Displays a chart summarizing data contained in files, as directed by an instruction file.
- Circular tree – Draws trees in circular form.
- Clade Match with Another Current Tree – Calculates a number relating a clade and another current tree (e.g., a tree-to-tree distance metric).
- Clustal (DNA/RNA) – Imports Clustal files that consist of DNA or RNA sequence data.
- Clustal (protein) – Imports Clustal files that consist of amino acid sequence data.
- Clustal Align – Sends the selected sequence to Clustal to align.
- Cluster analysis – Supplies trees obtained from cluster analysis on distance matrices.
- Coalescence Contained within Current Tree – Generates tree by a simple coalescence model of a neutral gene with constant population size, within a current species tree from a Tree window or other tree context. Branch lengths are assigned according to generation of coalescence. The species tree used is a current tree found in a Tree Window or other tree context.
- Coalescence in Current Tree with Migration – Generates tree by a simple coalescence model of a neutral gene with constant population size, within a current species tree from a Tree window or other tree context. Branch lengths are assigned according to generation of coalescence. The default population size is 10000. The species tree used is a current tree found in a Tree Window or other tree context. Migration is allowed; the default probability is 0.00001 per individual generation. Migration can be isolated to a particular generation, or spread throughout the depth of the species/population history.
- Coalescence Package Introduction – Serves as an introduction to the coalescence package.
- Coalescence Simulated within Current Tree – Generates tree by a simple coalescence model of a neutral gene with constant population size, within species trees. Branch lengths are assigned according to generation of coalescence. The species tree used is a current tree found in a Tree Window or other tree context.
- Coalescent Trees – Generates tree by coalescence within a single panmictic population.
- Codon Position Rates Model – Defines and maintains models that specify rates for codon positions.
- Collapse Gaps to Left – Collapses gaps in the selected block of cells to yield unaligned sequences.
- Collapse Gaps to Right – Collapses gaps in the selected block of cells to the right, to yield unaligned sequences.
- Collapse Zero-length Branches – Collapses zero-length branches to yield polytomes.
- Colless's Imbalance – Calculates Colless's Imbalance statistic for tree asymmetry, normalized by maximum asymmetry. Polytmous nodes are ignored in the calculation.
- Color Branches – Provides a tool with which to color branches in a tree window.
- Color Branches by Partition – Colors the tree's branches by the taxon partition.
- Color By Amino Acid – Colors the cells of a DNA matrix by the amino acids for which they code.
- Color By Cell Value – Colors the cells of a character matrix according to a value for that cell or a moving window of cells.
- Color By Character Value – Colors the cells of a character matrix according to a value for the character or a moving window of characters.
- Color By Footnote – Colors the cells of a character matrix by whether or not they have a footnote.

- Color By State – Colors the cells of a character matrix by their contained character states.
- Color By Taxon Value – Colors the cells of a character matrix according to a value for the Taxon or a moving window of taxa.
- Color For Aligning – Colors the cells of a character matrix for easy of alignment.
- Color If Excluded – Colors the excluded characters gray.
- Compare Bases with Taxon – Compares nucleotide bases of selected taxa with those of a chosen taxon.
- Compare Other Matrix – Compares matrix in data editor with another.
- Compare with Another Current Tree – Calculates a number relating a tree and another current tree (e.g., a tree-to-tree distance metric).
- Compare with Other Trees – Calculates a number relating one tree to another in a sequence (e.g., a tree-to-tree distance metric), such that the i'th tree is compared to the i'th other tree in the sequence.
- Composite DNA Simulation Model – Defines and maintains simple stochastic models of DNA evolution for simulations.
- Concatenate Other Matrix – Concatenates matrix onto one in data editor. Assumptions like weights and character models are NOT transferred. For categorical data, state names are not included. For continuous data, new items may need to be created to accommodate differences in items between the matrices.
- Concatenate tree blocks – Concatenates selected tree blocks into a new tree block.
- Condense taxon names – Condenses taxon names, for example for use by programs that cannot handle long names or names with punctuation.
- Consensus Sequence Strip – Displays a consensus sequence as in info strip in a character matrix editor.
- Consensus State for Character –
- Consensus trees – Supplies consensus trees.
- Constellation – Draws trees with spots at the nodes in a more or less unrooted way.
- Contained Associates – Draws wide trees containing trees of associates (e.g., species tree containing gene tree)
- Continuous state of taxon – State of a continuous character.
- Convert Negative Branch Lengths to Zero – Adjusts a tree's branch lengths so that negative branch lengths are zero
- Convert Polymorphisms to Uncertainties – Alters categorical data by converting polymorphic entries to uncertainties.
- Convert Uncertainties to Polymorphisms – Alters categorical data by converting uncertain entries to polymorphisms.
- Correl Package – Correl is a Mesquite package for examining character correlation.
- Correlation Analysis – Performs a Correlation analysis for categorical characters.
- Current Codon Positions – Supplies current codon positions applied to characters for character list window.
- Current Genetic Codes – Shows current genetic codes applied to characters in character list window.
- Current Genetic Codes – Supplies the currently assigned genetic code (e.g., "standard") for a character.
- Current Parsimony Models – Shows current parsimony models applied to characters in character list window.
- Current Parsimony Models – Supplies the currently assigned parsimony model (e.g., "unordered") for a character.

- Current Probability Models – Supplies current stochastic models applied to characters for character list window.
- Current probability models – Supplies the currently assigned stochastic (likelihood, probability) model for a character.
- Current taxa partition – Supplies the current taxa partition.
- Current Tree – Supplies a single tree currently shown in a tree window.
- Current Weights – Supplies current weights applied to characters for character list window.
- Curvogram – Draws trees with curved branches (as PHYLIP's 'Curvogram').
- Data Painter – Supplies the paint can tool and dropper tool for filling cells in a character data editor.
- Data type for model – Indicates data type of model in list window.
- Data Window Coordinator – Coordinates the creation of basic data windows.
- Deassign Branch Lengths – Sets lengths of a tree's branches to unassigned.
- Deep Coalescences (gene tree) – Counts W. Maddison's (1997) number of extra gene lineages ("deep coalescences") for gene tree within species tree
- Deep Coalescences (species tree) – Counts W. Maddison's (1997) number of extra gene lineages ("deep coalescences") for species tree implied by a contained gene tree
- Deep Coalescences Multiple Loci – Counts W. Maddison's (1997) number of extra gene lineages ("deep coalescences") for species tree containing a series of gene trees. The gene trees must be represented by a single taxa block.
- Default Order (characters) – Shows default order of character.
- Default Order (taxa) – Shows default order of taxa.
- Default Trees – Supplies simple default trees (bush, ladder).
- Defaults – Supervises some Mesquite-wide defaults
- Definite Tree Source – Supplies trees from from a tree source, arranged to be a definite number
- Delete Reticulations – Deletes secondary parents of nodes with more than one immediate ancestor. There is no control over which parents get deleted.
- Dependent Contained Tree Window – Displays a single tree (the same as contained within a tree in a tree window).
- Dependent Tree Window – Displays a single tree (the same as in a tree window).
- Diagonal tree – Draws trees with standard diagonal branches ("cladogram")
- Difference in steps in two characters – Calculates the difference between two characters in parsimony steps for a given tree.
- Difference in two values for character – Coordinates the calculation of the difference in two numbers for a character.
- Display Character Loadings – Shows character loadings in ordination.
- Display Mean – Calculates and shows the mean value.
- Distance between taxa – Counts the difference between two taxa.
- Distance of Contained Taxa – Distances among taxa according to distances among contained taxa (e.g., genes).
- Distances from Character Matrix – Distances calculated from a character matrix.
- Diverse Package – Diverse is a Mesquite package for examining speciation and extinction.
- Diversification (Char. Indep.) – Coordinates analyses of diversification (speciation/extinction).
- DNA Empirical State Frequencies Model – Defines and maintains model of state frequencies that match those found in an empirical matrix.

- DrawHierarchy – Draws project and module trees
- E-mail Tree – Calls mailto: to email a text image of the tree.
- Edit State Names – Edits state names of a categorical data matrix.
- Equiprobable Trees – Generates trees randomly so that each possible labelled tree topology is equally likely.
- Evolutionary Principal Components – Performs Maddison and Dyreson's evolutionary principal components analysis on a continuous-valued matrix.
- Evolve Categorical Characters – Simulates character evolution for categorical characters on a tree.
- Evolve Continuous Characters – Simulates character evolution for continuous characters on a tree.
- Evolve DNA Characters – Simulates evolution of DNA sequences on a tree.
- Evolving Speciation Rate (Continuous Character) – Generates tree by a speciation model in which the speciation rate evolves by a Brownian motion model.
- Examples Navigator – Provides a Examples Navigator window with explanation and buttons to link to other files
- Export Distance Matrix – Generates distance matrices in the manner of PDDIST (Garland, T., Jr., and A. R. Ives. 2000)
- Export Matrices & Batch Files – Saves copies of matrices to separate files for subsequent batch analysis (e.g., parametric bootstrapping).
- Export NEXUS for MrBayes – Exports NEXUS files for use by MrBayes.
- Extract tree block – Extracts selected trees and places them in a new tree block.
- F81 distance – F81 (Felsenstein, 1981) distance from a character matrix.
- F84 distance – F84 (Felsenstein, 1984) distance from a DNA matrix.
- FASTA (DNA/RNA) – Imports and exports FASTA files that consist of DNA/RNA sequence data.
- FASTA (protein) – Imports and exports FASTA files that consist of amino acid sequence data.
- Fetch & Add GenBank Sequences – Fetches GenBank nucleotide sequences given their GenBank accession numbers and adds them to the matrix.
- File Comment – Manages the main file comment in a NEXUS file.
- Fill – Fills cells with a uniform state in a character data editor.
- Filter Trees from Other Source – Filters trees from another source.
- Find Sequence – Finds the next occurrence of a sequence in a matrix of molecular data.
- Forward/Backward Rates – Uses maximum likelihood to estimate the rates of forward and backward changes (0 to 1 and 1 to 0 changes respectively), or alternatively the overall rate and the bias in gains versus losses, using the AsymmMk model on a tree for a given character.
- Frequency of State – Calculates the frequency of a state for a categorical character
- Fused Matrix Export (NEXUS) – Exports NEXUS files with matrices fused.
- Gamma Invar Rates Model – Defines and maintains Gamma rate variation models with a proportion of invariant characters.
- Gamma Rates Model – Defines and maintains Gamma rate variation models.
- Gaps to Missing – Converts gaps to missing data in selected region.
- GC – Returns 1 for G or C, 0 for A or T.
- Gene Duplication-Extinction (gene tree) – Counts the number of duplication and extinction events for gene tree implied by a containing species tree
- Gene Duplication-Extinction (species tree) – Counts the number of duplication and

- extinction events for species tree implied by a contained gene tree
- Genesis Introduction – Serves as an introduction to the Genesis package, which implements models of character evolution.
- Go-to Window – Provides a window to jump to another file
- Group Membership (characters) – Lists and allows changes to group membership in the current partition of characters, for List of Characters window.
- Group Membership (taxa) – Lists and allows changes to group membership in the current partition of taxa, for List of Taxa window.
- GTR Rate Matrix Model – Defines and maintains six-parameter general time-reversible (GTR) rate matrix models.
- Heuristic (Add & rearrange) – Searches for optimal trees by adding taxa then rearranging the tree.
- Highlight Apparently Slightly Misaligned – Colors aligned sequences to emphasize sections that seem slightly misaligned.
- Import and Export Package Introduction – Serves as an introduction to the import/export package for Mesquite.
- Include Taxa from File – Includes a file and concatenates its taxa to an existing taxa block, for instance to add seqences.
- Initialize DNA Submodels – Initializes default DNA submodels.
- Initialize Genetic Code – Initializes default genetic codes.
- Initialize Parsimony – Initializes default character models for parsimony.
- Initialize predefined categorical parsimony models – Initializes the predefined categorical parsimony models.
- Initialize predefined continuous parsimony models – Initializes the predefined continuous parsimony models.
- Insert Node – Provides a tool with which to insert nodes along a branch.
- Interpolate Character Selection – Selects characters that are surrounded by characters selected.
- Item values chart – Manages chart of values for items.
- Items bi-plot – Makes a chart comparing two values for each of many items.
- iText – iText is a package for save PDF files within Mesquite.
- JAMA Introduction – Serves as an introduction to the JAMA library used in Mesquite.
- Jukes Cantor distance – Jukes Cantor distance from a character matrix.
- K2P distance – K2P (Kimura 2-parameter) distance from a character matrix.
- Kyte & Doolittle Hydrophobicity – Returns hydrophobicity of amino acid using the scale of Kyte, J & R.F. Doolittle (1982) J. Mol. Biol. 157:105-142. Numbers from <http://www.whatislife.com/reader/protein/aa.html>
- Label Branch Lengths – An assistant to a tree display that labels branches to show their lengths.
- Label states – A module that displays character states on tree using labels. This is a display-only module, and would be hired by another module that organizes assigning states to the nodes.
- Landmark Drawings – Places drawing of landmarks at each node in tree.
- Likelihood Ancestral States – Coordinates reconstruction of ancestral states by maximum likelihood. Currently each node is estimated independently (i.e., corresponding to PAUP's marginal reconstruction).
- Likelihood in Character – Calculates the negative log likelihood of a tree with respect to a single character.
- Likelihood Surface AsymmMk Model – Shows the likelihood surface for the AsymmMk

## Model.

- Lineages Through Time – Displays a window showing plot of lineages through time averaged over many trees.
- List of Character Inclusion Sets – Makes windows listing character sets.
- List of Character Partitions – Makes windows listing character partitions.
- List of Character Sets – Makes windows listing character sets.
- List of Character Weight Sets – Makes windows listing character weight sets.
- List of Codon Positions Sets – Makes windows listing codon positions sets.
- List of Genetic Code Sets – Makes windows listing genetic code sets.
- List of Parsimony Model Sets – Makes a window listing parsimony model sets.
- List of Probability Model Sets – Makes windows listing probability model sets.
- List of Taxa Partitions – Makes windows listing taxa partitions.
- List of Taxon Sets – Makes windows listing taxon sets.
- Log transform – Alters continuous data by log transforming values.
- Lower Case – Returns 1 if state symbol is a lower case letter.
- Maintain Clipboard Match – Finds the first occurrence of the sequence in the clipboard, within a designated taxon, and maintains that match as the clipboard changes.
- Maintain Target Match – Finds the first occurrence of the sequence in the text area below the matrix, within a designated taxon, and maintains that match as the text area changes.
- Make Rerootings of Clade – Presents a tool by which you can touch on a tree; a tree block is made consisting of trees representing all rerootings of the clade of the node touched.
- Manage alternative taxon names – Manages (including NEXUS read/write) alternative taxon names.
- Manage ASSUMPTIONS blocks – Manages ASSUMPTIONS block in NEXUS file.
- Manage Attached Notes – Manages (including NEXUS read/write) notes attached to taxa, characters and cells of matrices.
- Manage AUTHORS blocks – Manages AUTHORS block in NEXUS file.
- Manage categorical character matrices – Manages categorical data matrices (including read/write in NEXUS file).
- Manage character inclusion sets – Manages (including NEXUS read/write) character inclusion sets (EXSETS).
- Manage Character Models – Manages character models.
- Manage character partitions – Manages (including NEXUS read/write) character partitions.
- Manage character sets – Manages (including NEXUS read/write) character sets (CHARSETS).
- Manage character weight sets – Manages (including NEXUS read/write) character weight sets.
- Manage codon positions – Manages (including NEXUS read/write) codon position sets.
- Manage CODONS blocks – Manages CODONS block in NEXUS file.
- Manage Continuous char. matrices – Manages continuous data matrices (including read/write in NEXUS file).
- Manage DNA/RNA matrices – Manages DNA/RNA data matrices (including read/write in NEXUS file).
- Manage Foreign Blocks – Manages unrecognized blocks in a NEXUS file.
- Manage genetic code sets – Manages (including NEXUS read/write) genetic code sets

(CODESETs).

- Manage Geographic character matrices – Manages data matrices of geographic data (including read/write in NEXUS file).
- Manage hyperlinks – Manages (including NEXUS read/write) hyperlinks.
- Manage LABELS blocks – Manages LABELS block in NEXUS file.
- Manage MESQUITE block – Manages MESQUITE block (including compose snapshot of current state of file in MESQUITE block "Auto" in NEXUS file).
- Manage MESQUITECHARMODELS blocks – Manages MESQUITECHARMODELS block in NEXUS file.
- Manage NOTES blocks – Manages footnotes and pictures attached to taxa, characters and data points (including read/write NOTES block in NEXUS file).
- Manage parsimony model sets – Manages (including NEXUS read/write) parsimony model sets (TYPESETs).
- Manage pictures – Manages (including NEXUS read/write) pictures.
- Manage probability model sets – Manages (including NEXUS read/write) probability model sets (PROBMODELSETs).
- Manage Protein matrices – Manages Protein data matrices (including read/write in NEXUS file).
- Manage SETS blocks – Manages character sets and other sets of various kinds (including read/write SETS block in NEXUS file).
- Manage stored character orders – Manages (including NEXUS read/write) character orders.
- Manage TAXA blocks – Manages sets of taxa (including read/write TAXA block in NEXUS file).
- Manage taxa partitions – Manages (including NEXUS read/write) taxa partitions.
- Manage TaxaAssociation blocks – Manages TaxaAssociation blocks in NEXUS file.
- Manage taxon sets – Manages (including NEXUS read/write) taxon sets.
- Manage TREES blocks – Manages tree blocks (including read/write TREES block in NEXUS file).
- Map Continuous – Supplies a reconstruction of continuous-valued ancestral states on a tree. Differs from the basic reconstruction methods in that it supplies simply numbers for nodes, instead of a CharacterHistory
- Matching Sequence – Finds the next occurrence of a sequence in a matrix of molecular data. Allows a certain number of mismatches.
- Matrices from Ordinations – Supplies character matrices from ordinations of existing matrices.
- Matrix from Clipboard – Makes a character matrix from text in the clipboard.
- Matrix Info – Shows Information about the data in each taxon.
- Matrix Source (indep.) – Coordinates the supply of matrices to other modules. It acts independently, in that it decides what matrix to supply (e.g., by presenting an interface to choose).
- Matrix Source (obed.) – Coordinates the supply of matrices to other modules. It acts obediently, in that it does not present an interface to choose which matrix, but relies on employer to decide.
- Matrix value with current tree – Coordinates the calculation of a number for a character matrix based on a current tree.
- Maximum likelihood reconstruct (Generic categorical) – Assesses likelihood for categorical characters, and reconstructs ancestral states by the maximum marginal probability ("MLE") criterion

- Mean value of character (linked matrix) – Calculates the mean observed trait for continuous character in a linked matrix.
- Mean value of character – Calculates the mean observed trait for continuous character.
- Merge Taxa – Merges selected taxa and their character states.
- Mesquite – This is the central module for the Mesquite system. In it is the main method that starts up the Mesquite application. This module loads information for all of the other modules, and hires FileCoordinator modules as needed to deal with open files. Thus, in the tree of employees that active modules make, this module is at its root.
- Mesquite – This is the central module for the Mesquite system. In it is the main method that starts up the Mesquite application. This module loads information for all of the other modules, and hires FileCoordinator modules as needed to deal with open files. Thus, in the tree of employees that active modules make, this module is at its root.
- mesquite.jsci Package Introduction – Serves as an introduction to the portions of JSci used by Mesquite.
- Mirror Tree Window – Displays a single tree (the same as in a tree window) twice, in mirror image.
- Missing to Gaps – Converts missing data to gaps in selected region.
- Mk1 Estimated Rate – Estimates the rate of a character's evolution under the simple Mk1 model.
- Mk1 Model (Markov 1 parameter) – Defines and maintains simple Markov k-state 1-parameter stochastic models (Lewis, 2001) of character evolution.
- Model Paradigm – Indicates paradigm of model in list window.
- Modify Current Tree – Modifies current tree.
- Most Pairs – Chooses taxon pairings regardless of contrast in a character, so as to maximize the number of pairs that are phylogenetically independent.
- Move block – Allows one to manually blocks in a sequence, and split the blocks.
- MrBayes Max. A-Post. Tree – Supplies tree from MrBayes with highest posterior probability (MAP tree).
- MrBayes Score – Supplies posterior probability score from MrBayes
- MrBayes Trees – Supplies trees from MrBayes
- MRP Matrices from Trees – Supplies matrices which represent trees for MRP (Matrix Representation with Parsimony) supertree analyses.
- Multi Tree Window – Displays a special tree window with many trees simultaneously.
- Multiply by value – Alters continuous data by multiplying by a value. All items of the matrix are similarly modified.
- Multistate sites in taxon – Reports the number of polymorphic/partially uncertain sites in a taxon for a data matrix.
- NBRF/PIR (DNA/RNA) – Imports and exports NBRF files that consist of DNA/RNA sequence data.
- NBRF/PIR (protein) – Imports and exports NBRF files that consist of amino acid sequence data.
- Neutral Coalescence – Performs neutral coalescence within a population by a simple model of a neutral gene with constant population size. Remembers a tree of nodes, with branch lengths assigned according to generation of coalescence. The default population size is 10000.
- NEXUS Blocks List – Makes windows listing NEXUS blocks.

- NEXUS Defaults – Sets whether TITLE and LINK commands are to be suppressed where possible in saving NEXUS files.
- NEXUS file interpreter – Coordinates the reading and writing of NEXUS files.
- NNI Rearranger – Rearranges a tree by nearest neighbor interchange (NNI). Does not handle trees with polytomies.
- No Color – Turns off cell coloring.
- Node Count (Speciation Patristic) Distances implied by Tree – Distances among taxa implied by counting nodes along branches of tree from one taxon to another (Speciation Patristic distances).
- Node Depth – Calculates the depth of each node from the highest terminal in its clade.
- Node Locations (2D plot) – Calculates the node locations for a tree plotted in a two dimensional space.
- Node Locations (3D plot) – Calculates the node locations for a tree plotted in a three dimensional space.
- Node Locations (circle) – Calculates the node locations for a tree drawn in circular fashion, with root at center.
- Node Locations (oval) – Calculates the node locations for a tree drawn in oval fashion, with root at center.
- Node Locations (standard) – Calculates the node locations in a tree drawing, for use with vertical or horizontal tree drawers (e.g., the standard diagonal or square trees).
- Node Namer – Provides a tool to name the nodes of a tree
- Node Position Adjust – Provides a tool to adjust the depth of a node in the tree
- Nodes Scattergram – Shows values for nodes of tree via a scattergram.
- Nucleotide Complement – Alters nucleotide data to its complement.
- Number for Character (in List of Characters window) – Supplies numbers for characters for a character list window.
- Number for Matrix (in List of Character Matrices window) – Supplies numbers for character matrcies for a character matrices list window.
- Number for Nodes using Character – Supplies numbers, based on a character, for each node of a tree.
- Number for Nodes using Matrix – Supplies numbers, based on a matrix, for each node of a tree.
- Number for Taxon (in List of Taxa window) – Supplies numbers for taxa for a taxa list window.
- Number for Tree (in List of Trees window) – Supplies numbers for trees for a trees list window.
- Number for Tree Block (in List of Tree Blocks window) – Supplies numbers for tree blocks for a tree blocks list window.
- Number of characters in character set – Indicates number of characters in character set in list window.
- Number of characters in inclusion set – Indicates number of characters in inclusion set in list window.
- Number of characters of data matrix – Indicates taxa of data matrix.
- Number of groups in taxa partition – Indicates number of groups in taxa partition in list window.
- Number of Stops – Reports the number of stops coded by nucleotides in a taxon for a data matrix.
- Number of Taxa – Counts the number of taxa in a tree.
- Number of taxa in set – Indicates number of taxa in list window.

- Number of taxa in taxa set – Indicates number of taxa in taxa set in list window.
- Number of trees in tree block – Indicates number of trees.
- Number of Unambiguous Sites in Sequence – Reports the number of unambiguous sites.
- Number of Taxa with Non-Gap, Non-Missing – Calculates the number of taxa in the character that contain applicable, non-missing data.
- Numbers for Nodes from Character Reconstructions – Supplies numbers, based on a character reconstruction, for each node of a tree.
- Occasionally Randomly Modify – With specified probability, will ask random tree modifier to modify current tree.
- Open URL – Opens a file on the web as if it were a local data file
- Original File (import source) – Shows the original file from which a fuse/imported taxon came.
- Other Tree Blocks – Supplies blocks of trees from various sources
- Pagel format (ppy) file for Discrete – Imports and exports files formatted for Pagel's Discrete program.
- Pagel format (ppy) file for Multistate – Imports and exports files formatted for Pagel's Multistate program.
- Pagel's 1994 test of correlated (discrete) character evolution – A statistical test, described in Pagel(1994), for nonindependent evolution of two discrete, binary characters
- Pairs for one character – Chooses taxon pairings so as to maximize the number of pairs that are phylogenetically independent, subject to the constraint that each pair shows a contrast the states of a binary character.
- Pairs for two characters – Chooses taxon pairings so as to maximize the number of pairs that are phylogenetically independent, subject to the constraint that each pair shows a contrast the states of both of two binary characters.
- Pairwise Comparison – Performs pairwise comparison character correlation tests. Phylogenetically independent pairs are chosen, and the states of two binary characters are compared to see if they are correlated among these pairs.
- Pairwise Comparisons Package Introduction – Serves as an introduction to the pairwise comparisons character correlation package for Mesquite.
- PAL Introduction – Serves as an introduction to the use of the PAL library in Mesquite.
- Parameters Explorer – Provides a window to show values returned when parameter settings are varied
- Parsimony Ancestral States – Coordinates the parsimony reconstruction of ancestral states.
- Parsimony Character Steps – Calculates the number of parsimony steps in a character.
- Parsimony Linear – Reconstructs the ancestral states of continuous characters so as to minimize the sum of absolute values of changes (linear, Wagner, Farris or Manhattan parsimony). If the continuous character has multiple items, then length is reported on only the first item.
- Parsimony Ordered – Reconstructs the ancestral states of categorical characters using parsimony, under the assumption that states are ordered (ordered, Farris or Wagner parsimony; additive). Also counts parsimony steps.
- Parsimony Package Introduction – Serves as an introduction to the basic parsimony package for Mesquite.
- Parsimony Squared-change – Reconstructs the ancestral states of continuous characters so as to minimize the sum of squared changes (squared-change or least squares parsimony; Brownian motion likelihood).

- Parsimony Stepmatrix – Reconstructs ancestral states of categorical characters using a stepmatrix (cost matrix).
- Parsimony Unordered – Reconstructs the ancestral states of categorical characters using parsimony, under the assumption that states are unordered (unordered or Fitch parsimony; nonadditive). Also counts parsimony steps.
- Patristic distance correlation – Calculates the product moment correlation coefficient among the off-diagonal elements of the patristic distances matrices of two trees. Ignores unassigned and negative values in the matrices. If employed as a distance, converts scores from 1 to -1 to scores from 0 to 2
- Patristic Distances implied by Tree – Distances among taxa implied by path-length distance along branches of tree from one taxon to another (Patristic distances). Unassigned branches are treated as of length 1.0.
- Phytip (categorical data) – Imports and exports Phytip matrices that consist of basic categorical data with just two states. Exported data will consist of default symbols (0 and 1).
- Phytip (DNA/RNA) – Imports and exports Phytip matrices that consist of DNA/RNA sequence data.
- Phytip (protein) – Imports and exports Phytip matrices that consist of amino acid sequence data.
- Phytip (trees) – Imports and exports Phytip trees.
- Picture Window – Displays a picture in a window.
- Plot Tree – Draws trees plotted in a two dimensional space.
- Plot Tree 3D – Draws trees plotted in a three dimensional space.
- Polytomies Present – Indicates whether polytomies are present.
- Polytomy Assumption – Indicates whether the tree assumes that any polytomies, if present, are hard or soft. If not, the default assumption is used.
- POY (DNA/RNA) – Exports POY files of DNA/RNA sequence data.
- Principal Components Analysis – Performs principal components analysis on a continuous-valued matrix.
- Projector – Supplies a projector tool for tree windows that allows pictures attached to taxa to be displayed.
- Proportion Gaps In Character – Calculates the proportion of gaps (inapplicable codings) in a character across taxa. Does not include missing (unassigned) data.
- Proportion Gaps in Taxon – Reports the proportion of gaps (inapplicable codings) in a taxon for a data matrix.
- Proportion Internal Gaps – Calculates the proportion of internal gaps (inapplicable codings) in a character across taxa. Does not include missing (unassigned) data.
- Proportion Invariant Model – Defines and maintains models that specify that a certain proportion of characters are invariant.
- Proportion lower case codings in taxon – Reports the percentage of lower case codings in a taxon for a data matrix.
- Proportion Missing – Calculates the proportion of missing data in a character across taxa.
- Proportion missing data in taxon – Reports the proportion of missing data in a taxon for a data matrix.
- Proportion Polymorphic – Calculates the proportion of taxa coded as polymorphic or partially uncertain in a character.
- Proportion Terminal Gaps – Calculates the proportion of Terminal gaps (inapplicable codings) in a character across taxa. Does not include missing (unassigned) data.

- Protein Site Property – Calculates the mean value of amino acid properties at a site across taxa.
- Quick Key Entry – Provides a tool with which to quickly enter data. If this tool is active, then typing a key will cause that value to be entered into all selected cells.
- Random Branch Moves – Rearranges tree by random branch moves.
- Random Fill (Categorical) – Fills cells with a randomly-chosen state. For DNA data, states A, C, G, and T are chosen with equal probability; for other data, states up to and including the maximum state value are chosen with equal probability.
- Random Fill (Gaussian/Normal) – Fills cells with a random state, using a Normal distribution.
- Random Fill (Uniform) – Fills cells with a random state, uniformly.
- Random n characters – Selects n characters randomly.
- Randomly Modify Current Tree – Modifies current tree by random changes.
- Randomly Modify Matrix – Supplies character matrices that are randomly modified from an existing matrix.
- Randomly Modify Respective Matrices – Supplies character matrices that are randomly modified from a series of existing matrices. The i'th modified matrix is derived from the i'th original matrix.
- Randomly Modify Trees – Modifies each of a series of trees by random changes; the i'th tree from this module comes by modifying the i'th tree from the original source of trees.
- Randomly Resolve Polytomies – Randomly resolves polytomies in tree. All possible resolutions are equiprobable. Thus, if the tree is a polytomyous bush, the resulting resolved trees will be distributed equivalently to that from the Equiprobable Trees module.
- Rarefy Characters – Deletes characters randomly to rarefy matrix.
- Rarefy Tree – Rarefies tree by randomly excluding taxa. If some taxa are selected, random exclusion is limited to the selected taxa.
- Raw Nexus block editor – Edits blocks in a NEXUS file.
- Read DATA blocks – Coordinates the reading of a DATA block in NEXUS file.
- Read DISTRIBUTION blocks – Coordinates the reading of a DISTRIBUTION block in NEXUS file.
- Rearranged tree – Supplies trees that are rearrangements of a given tree. The original tree is NOT included among the rearrangements.
- Recode Characters – Recodes categorical data (e.g., state 1 to state 0).
- Reconstruct Ancestral States – Reconstructs ancestral states on the nodes of a tree.
- Reconstruct Deep Coalescence – Reconstructs a contained tree within a containing tree so as to minimize the amount of deep coalescence (failure of lineage sorting). The contained tree is assumed to be rooted.
- Remove from Taxon Names – Removes a specified number of characters from taxon names.
- Remove Gaps-Only Characters – Removes all characters that are gaps only.
- Remove item... – Removes an item in each cell of a continuous matrix.
- Rename item... – Renames an item of a continuous matrix.
- Replaces text in taxon names – Replaces text in taxon names.
- Reshuffle Character – Supplies characters that are reshufflings of an existing character.
- Reshuffle States Within Characters – Shuffles (permutes) character states among taxa within each character.

- Reshuffle States Within Taxa — Shuffles (permutes) character states among characters within each taxon.
- Reshuffle Terminal Taxa — Shuffles (permutes) the taxa among the terminal nodes.
- Reshuffle Within Characters (Taxa Partitioned) — Shuffles (permutes) character states among taxa of each taxa partition, within each character.
- Reshuffle Within Taxa (Char. Partitioned) — Shuffles (permutes) character states among characters of each character partition, within each taxon.
- Resolve Polytomies (to 0-length branches) — Resolves polytomies arbitrarily and assigned the resulting new branches zero length.
- Reverse Complement — Reverses the DNA sequence and complements the bases.
- Reverse Sequence — Alters data by reversing sequence of states.
- Rhetenor Package Introduction — Serves as an introduction to the Rhetenor morphometrics package.
- Right-Ladderize Tree — Sets the tree to its ladderized (right) version.
- Root tree — Sets the tree to rooted.
- Root tree with selected taxa as outgroup — Roots the tree between the selected taxa and the remainder, if possible.
- Rooting status — Indicates whether the tree is rooted or unrooted.
- s of Slatkin & Maddison — Calculates 's' of Slaktin & Maddison 1989 for tree of genes associated with given populations
- Sample Trees Directly From File — Supplies randomly-sampled trees directly from a file, without bringing the contained tree block entirely into memory. This allows much larger blocks of trees to be used within constraints of memory, but will make some calculations slower. This module does not know how many trees are in the file, and hence may attempt to read files beyond the number in the file.
- Save matrix copies — Provides for the saving of copies of matrices to separate files. This is available under the Characters menu.
- Scale All Branch Lengths — Adjusts a tree's branch lengths by multiplying them by an amount.
- Scale Selected Branch Lengths — Adjusts lengths of a tree's selected branches by multiplying them by an amount.
- Scattergram — Helps make scattergram charts.
- Search Data — Manages data-searching modules.
- Select Branches — Provides a tool and menu with which to select branches in a tree window.
- Select by Matrix Comparison — Selects cells of the matrix that differ compared to another matrix
- Select By Search (Find all) — Selects cells of the matrix according to whether their text contains a given string
- Select Lowercase Ends — Selects ends of a DNA sequence up to the first block of N sites in a row with uppercase (confident) symbols (the user chooses N). Assuming that lowercase letters are used for less certain base calls, this can be used to select and then trim (by painting with gaps) poorly-sequenced terminal regions.
- Select Matching Taxa — Selects taxa according to whether their sequence of states matches the sequence of states in the selected taxon
- Select Same Distribution — Selects cells of the matrix according to whether their distribution of states across taxa in a character is the same as that in the selected character or block of cells
- Select Same Subsequence — Selects cells of the matrix according to whether their

- sequence of states matches the sequence of states in the selected taxon or block of cells
- Select Taxa – Provides a tool with which to select taxa in a clade in a tree window.
  - Selected Taxa Convex in Tree – Determines if the selected taxa are convex in a tree. That is, does there exist a rooting of the tree in which the selected taxa form a clade?
  - Selected Taxa Form Clade – Determines if the selected taxa form a clade in a tree.
  - Sequence Length – Reports the length (total minus gaps) of a molecular sequence in a taxon.
  - Set Author – Sets the author for this machine and account.
  - Set Branch Lengths to Numbers For Nodes – Assigns a value of branch length for all of a tree's branches based upon the numbers calculated by a Numbers for Nodes.
  - Set Polytomy Assumption to Default – Sets assumption that any polytomies in the tree are whatever is current default (which can be set by going to project and files or Log window and choosing "Defaults>Set Polytomy Default...").
  - Set Polytomy Assumption to Hard – Sets assumption that any polytomies in the tree are hard.
  - Set Polytomy Assumption to Soft – Sets assumption that any polytomies in the tree are soft.
  - Set to Lower Case – Alters nucleotide data to lower case coding (e.g, to indicate less certain).
  - Set to Upper Case – Alters nucleotide data to upper case coding (e.g, to indicate more certain).
  - Set Tree Defaults – Sets the default state for polytomies & other aspects of tree handling.
  - Shade numbers – A module that displays numbers at tree nodes using labels. This is a display-only module, and would be hired by another module that organizes assigning numbers to the nodes.
  - Shade states – Shows the states at nodes by shading branches or nodes using colors, black and white, or shades of gray.
  - Shared Clades – Calculates the number of shared clades between two trees (excludes the clade consisting of all taxa).
  - Shared History Distances implied by Tree – Distances among taxa implied by length of shared history (distance from root to most recent ancestor). Unassigned branches are treated as of length 1.0.
  - Shared Partitions – Calculates the number of shared partitions between two trees. If used as a distance, then converted by subtracting shared partitions from the maximum possible, i.e. the number of partitions in the subtree of shared taxa in the tree with the most partitions.
  - Shared Proportion Distances implied by Tree – Distances among taxa implied by proportion of total branchlength (root to tip) that is shared (Shared Proportion distances). Unassigned branches are treated as of length 1.0.
  - Shift Other To Match – Shifts other sequences to match the one selected.
  - Show Employee Tree – Shows the window listing the tree of employees of the module of a given window.
  - Show List with Genetic Codes – Shows the List of Character window with the genetic codes column on.
  - Show Node Numbers – Shows the node numbers on a tree.
  - Show Percentile – Calculates and shows percentile boundaries. Finds the upper (right) or lower (left) tail of the distribution that contains the specified percentage of the

distribution. If there is not a boundary that exactly matches the specified percentage, then the closest boundary that does not exceed the percentage is shown.

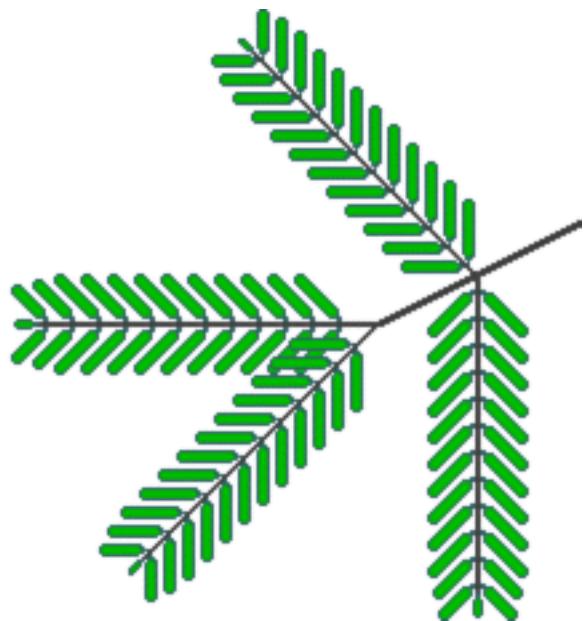
- Show text – Shows text in a window
- Show Y = X – Shows the Y=X line on a scattergram
- Shuffle states among taxa – Alters data by shuffling states among taxa within a character.
- Side Pusher – Pushes all sequences on one side.
- Simple (categorical data) – Imports and exports simple matrices that consist of basic categorical data. Exported data will consist of default symbols (0, 1, 2, ...).
- Simple (DNA/RNA) – Imports and exports simple matrices that consist of DNA/RNA sequence data.
- Simple (protein) – Imports and exports simple matrices that consist of amino acid sequence data.
- Simplified NEXUS – Exports NEXUS files, for instance of old fashioned style (using DATA block).
- Simulate Ancestral States – Simulates ancestral states on the nodes of a tree.
- Simulated Characters – Supplies simulated characters.
- Simulated Characters on Trees – Supplies simulated characters each from a respective tree.
- Simulated Matrices on Current Tree – Supplies simulated character matrices.
- Simulated Matrices on Trees – Supplies simulated character matrices, each evolved on a different of a series of trees.
- Simulated Tree Blocks – Creates a tree block of simulated trees.
- Simulated Trees – Supplies trees from simulations.
- Single Linkage – Supplies trees obtained from Single Linkage clustering.
- Sister Diversification – Compares sister clades and returns the P value of the one-tailed null hypothesis that uniform clades of state 1 are smaller than uniform clades of state 0 among those with different values (P value calculated by Binomial probability).
- Sort Taxa (data) – Provides a tool with which to sort taxa automatically.
- Speciation/Extinction Likelihood – Calculates likelihoods using a speciation/extinction model reduced from the BiSSE model (Maddison, Midford & Otto 2007)
- Speciation/extinction Likelihood Calculator – Calculates likelihoods using a speciation/extinction model reduced from the BiSSE model (Maddison, Midford & Otto 2007)
- SPR Rearranger – Rearranges a tree by subtree pruning and regrafting (SPR).
- Sprinkle Missing – Randomly converts entries in the character matrix to missing data, with a certain probability (i.e., it "sprinkles" missing data around the matrix).
- Square tree – Draws trees with standard square branches ("phenogram")
- Standardize – Alters continuous data by standardizing to have mean = 0 and unbiased estimate of variance = 1. Modifies only the first item of a multi-item matrix
- State Names Strip – Shows the strip at bottom of matrix with state names.
- States Information (in List of Characters window) – Supplies basic character state information for characters in character list window.
- Stepmatrix – Supplies editor for and manages stepmatrices (cost matrices).
- Steps in Character – Calculates the parsimony steps in a character for a given tree.
- Stochastic Character Evolution Package Introduction – Serves as an introduction to the basic stochastic character evolution package for Mesquite.
- Stochastic Character Mapping (Categorical) – Coordinates reconstruction of ancestral states using stochastic character mapping.

- Stored Character Order – Supplies current order applied to characters for character list window.
- Stored Characters – Supplies characters from data files (as opposed to simulated characters, for example).
- Stored Genetic Code – Supplies a user-specified genetic code, stored in the file.
- Stored Matrices – Supplies character matrices from data files (as opposed to simulated characters, for example).
- Stored Parsimony Model – Supplies a user-specified model of character evolution, for parsimony calculations, stored in the file. This can be different from the model assigned to the character as current in the List of Characters window.
- Stored Probability Model – Supplies a user-specified model of character evolution stored in the file.
- Stored Probability Model for Simulation – Supplies a user-specified model of character evolution stored in the file for simulation.
- Stored Taxa – Supplies taxa stored, for instance in a file.
- Stored Taxa Associations – Supplies associations between taxa that are stored, for instance in a file.
- Stored Taxon Pairs – Supplies taxon pairs stored, for instance in a file.
- Stored Tree Blocks – Supplies lists of trees stored, for instance in a file.
- Stored Trees – Supplies trees stored, for instance in a file.
- Sum of Branch Lengths – Calculates the sum of branch lengths of the tree, treating unassigned lengths as 0. The length of the root is counted.
- Tab-delimited categorical data file – Imports simple tab-delimited files of categorical data.
- Tab-delimited continuous data file – Imports and exports simple tab-delimited files of continuous data.
- Taxa Association Package Introduction – Serves as an introduction to the taxa association package for Mesquite.
- Taxa blocks list – Makes windows listing blocks of Taxa and information about them.
- Taxa of data matrix – Indicates taxa of data matrix.
- Taxa of tree block – Indicates taxa of tree block.
- Taxa partition from Taxa associations – Constructs a taxa partition from an association.
- Taxa Scattergram – Makes a scatterplot displaying two values (such as proportion of gaps in a sequence, etc.) for a series of taxa.
- Taxon Instability Among Trees – Calculates for each taxon the degree to which its implied unweighted patristic distances between that taxon and others differs among trees. For each taxon  $i$  this sums over all tree pairs  $x$  and  $y$  and over all other taxa  $j$ :  $|D_{ijx} - D_{ijy}| / (D_{ijx} + D_{ijy})^{z/2}$  where  $D_{ijq}$  is the distance between taxa  $i$  and  $j$  on tree  $q$ . Close relationships are emphasized if the exponent  $z$  is higher (default is 2).
- Taxon Link – Supplies a tool for tree windows that jumps to hypertext links for taxa.
- Taxon List – Makes windows listing taxa and information about them.
- Taxon Pairs Histogram – Makes a chart showing some value for each of many pairs of taxa.
- Taxon Pairs Scattergram – Makes a chart comparing two values for each of many pairs of taxa.
- Taxon selection coordinator – Coordinates taxon selection.
- Taxon Wand (data) – Provides a tool with which to select taxa automatically.
- Terminal ? to Gaps – Converts terminal missing data codings to gaps.

- Terminal Gaps to ? – Converts terminal gaps to missing data.
- Text Window – Displays Text in a window.
- Ti/Tv Rate Matrix Model – Defines and maintains two-parameter rate matrix models with a transition and transversion rates.
- TNT, NONA, Hennig86, PiWe, WinClada – Imports and exports TNT/NONA/Hennig86/PiWe/WinClada files.
- Top BLAST Matches – Does a blast search on selected data and returns the top blast matches for each sequence selected.
- Trace All Characters – Summarizes for each node the reconstructions of the states at all characters of the tree.
- Trace Character History – Traces a history of character evolution on the nodes of a drawn tree.
- Trace Character Over Trees – Summarizes at each node reconstructions of the states of a character over a series of trees. The summary is shown on the current tree; if you want to show it on a consensus of the trees, make sure that the current tree is that consensus.
- Transform Trees from Other Source – Transforms trees from another source.
- Translate DNA to Protein – Translates a stored DNA matrix to protein.
- Tree Block Directly From File – Supplies a block of trees directly from a file
- Tree Block Values – Makes a chart showing some value for each of many tree blocks.
- Tree Blocks from Files Listed In File – Supplies tree blocks in files listed in a file.
- Tree Blocks List – Makes windows listing tree blocks and information about them.
- Tree Depth – Calculates the deepest path (in branch length) from terminals to the root, treating unassigned lengths as 0. The length of the root is not counted.
- Tree Farm Package Introduction – Serves as an introduction to the Tree Farm package for Mesquite.
- Tree Legend – Makes the legend in a tree window to display the tree name and other information about it.
- Tree of context – Supplies a single tree from the nearest tree context (e.g., an available tree window).
- Tree Search – Supplies trees resulting from a search to optimize some value.
- Tree Search Package Introduction – Serves as an introduction to the tree search package for Mesquite.
- Tree Utility Coordinator – Coordinates use of tree utilities in tree window
- Tree Value Satisfies Criterion – Determines if the tree has a value either greater than, less than or equal to one specified.
- Tree value using 2 characters – Coordinates the calculation of a number for a tree based on 2 characters.
- Tree value using character – Coordinates the calculation of a number for a tree based on a character.
- Tree value using character matrix – Calculates a value for the tree using a character data matrix.
- Tree Window – Makes a basic tree window, which contains a tool palette. Hires assistants for the tree window (e.g., Trace Character).
- Tree Window Coordinator – Coordinates the creation of basic tree windows.
- Treelength – Calculates the parsimony treelength of a given tree and matrix.
- Treelength – Calculates the parsimony length of a tree.
- Trees & Diversification Characters – Evolves a series of trees, each tied to a single character controlling diversification rates.

- Trees Directly From File – Supplies trees directly from a file, without bringing the contained tree block entirely into memory. This allows much larger blocks of trees to be used within constraints of memory, but will make some calculations slower. This module does not know how many trees are in the file, and hence may attempt to read files beyond the number in the file.
- Trees List – Makes windows listing trees and information about them.
- Trees Package Introduction – Serves as an introduction to the basic trees package for Mesquite.
- Trees Scattergram – Makes a scatterplot displaying two values (such as likelihoods, parsimony scores, imbalance statistics, correlations, etc.) for a series of trees.
- Trim Terminal Gap Characters – Trims characters at edges of matrix that are gaps only.
- Truncate taxon names – Truncates taxon names.
- Type of character data matrix – Indicates type of character data matrix (e.g., continuous, DNA, etc.).
- Type of Model – Indicates type of model (e.g., Mk1, step-matrix) in list window.
- Ultrametricize – Adjusts a tree's branch lengths so that the distances among terminal taxa are ultrametric (i.e. like a molecular clock, all tips reaching to same level). This is not done with any sophisticated smoothing algorithm; rather, branches are just stretched until they reach to same level.
- Unambiguous Changes – Counts number of unambiguous changes along branch below based upon the character reconstruction.
- Uncorrected (p) distance (DNA) – Uncorrected (p) distance from a DNA matrix.
- Uncorrected Distance (General) – Uncorrected (p) distance from a character matrix.
- Uniform speciation (Yule) – Generates tree by simple uniform probability speciation (a Yule process) as done by Harding (1971). The chance of speciation is equal for all tips.
- Uniform Speciation with Sampling – Generates tree by simple uniform probability speciation (a Yule process), with subsequent sampling of species.
- Unique ID (characters) – Shows unique id assigned to character.
- Unique ID (taxa) – Shows unique id assigned to taxon.
- Unique IDs default – Sets the preferences for storing unique ids.
- Unroot tree – Sets the tree to unrooted.
- UPGMA – Supplies trees obtained from UPGMA clustering.
- User-Specified Nucleotide Frequency Model – Defines and maintains user-specified nucleotide frequency models.
- Utilities – Provides a menu for utilities
- Value for 2 characters with current tree – Coordinates the calculation of a number for two characters based on a current tree.
- Value from Tree – Calculates a value for each taxon using a tree.
- Values for Nodes – Shows on a drawn tree various possible numbers at the nodes.
- Variable among taxa – Selects characters that are variable among the selected taxa
- Web page link – Provides a window to jump to a web page
- Whole or Submodel – Indicates whether model is a complete or partial model of character evolution.
- Window Holder – Helps other modules by holding their windows.
- With probability p – Selects characters randomly with probability p.
- Within-group PCA – Performs within-group principal components analysis on a continuous-valued matrix.

# MESQUITE 2



*Characters,  
taxa  
& trees*

# Characters and Character matrices

Characters and their character states are means by which to describe the features of organisms. Mesquite supports characters whose states are **categorical** (discrete and not necessarily ordered) or **continuous**. Special versions of categorical characters exist for DNA, RNA and protein sequence data. For more details specific to those types of characters, see the pages on [molecular](#) and [continuous](#) characters. Categorical characters other than molecular characters can have 55 states (by default symbols 0 - 9, A - H, K - N, P - Z, a - h, k - n, p - z).

Characters can exist within **matrices** that are stored in a data file. Thus, one matrix may store a series of categorical characters to describe phenotypic features. A separate matrix may store continuous characters describing measurements taken from the organisms, while a third matrix may store DNA sequence data, in which each aligned site is treated as a character (Mesquite currently treats unaligned data as if it were aligned, though the alignment can be ignored). Each matrix may have only a single type of character, but a data file may contain more than one matrix.

Characters can also exist outside of matrices. For instance, characters may be created by [simulations and randomizations](#) and used directly in calculations, without at any point being captured in a matrix and stored in a file.

- [Creating a character matrix](#)
- [Deleting and renaming character matrices](#)
- [The Character Matrix Editor](#)
  - [Undo](#)
  - [Adding, deleting, renaming and sorting taxa and characters](#)
  - [Entering character data](#)
  - [Selecting taxa, characters and cells of the matrix](#)
  - [Searching the matrix](#)
  - [Copy/paste](#)
  - [Editing Names of Characters and States](#)
  - [Annotations](#)
  - [Coloring cells of the matrix](#)
  - [Alterations and Transformations](#)
- [Changing the attributes of characters](#)
- [Charting information about Characters](#)

## Creating a character matrix

There are several ways to create a character matrix to be stored in the file. Most simply, you can create a blank (empty) matrix by choosing [Characters>New Empty Matrix](#). In the dialog box that appears, name the character matrix and specify the number of characters. You will

also need to choose the sort of data the matrix will contain (standard categorical, DNA (or RNA) sequence data, continuous, or protein sequence data). Normally, you will create an empty matrix if you are about to start entering observations about organisms.

It is also possible to create character matrices that are already filled with character states. For instance, if you want to make a duplicate of an existing character matrix, select [Characters>Make New Matrix From>Stored Matrices](#). If you want to create a matrix from the contents of the clipboard, select [Characters>Make New Matrix From>Clipboard](#). Other choices available under [Characters>Make New Matrix From>](#) allow you to make and store matrices resulting from simulations of character evolution, randomizations of existing matrices, or other sources.

Character matrices can also be read from files, including those in NEXUS and other formats that can be imported.

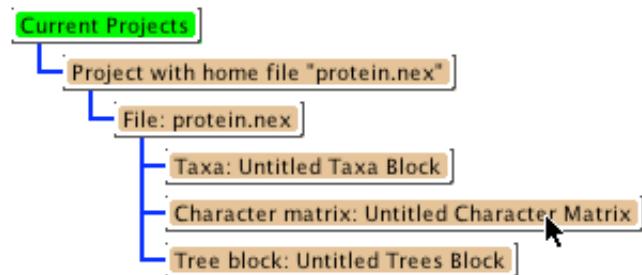
## Deleting and renaming matrices

There are three places you can rename and delete character matrices: in the Character Matrix Editor, in the List of Character Matrices window, and in the Projects and Files window.

In the Character Matrix Editor the Current Matrix submenu at the top of the matrix menu has menu items for renaming or deleting the matrix shown in the window.

In the List of Character Matrices window (available in the Characters menu), you can rename a matrix by editing its name directly. To delete matrices, select the rows corresponding to the matrices to be deleted, and select [List>Delete Selected Character Matrices](#).

To rename a matrix from the Projects and Files window, touch on the "Character Matrix" box:



A drop-down menu will appear with the option to rename the matrix. You can also use this drop-down menu to delete a matrix.

## The Character Matrix Editor

Once you have a character matrix, you may edit it using Mesquite's Character Matrix Editor, available at the top of the Characters menu. This is a spreadsheet editor, similar in style to

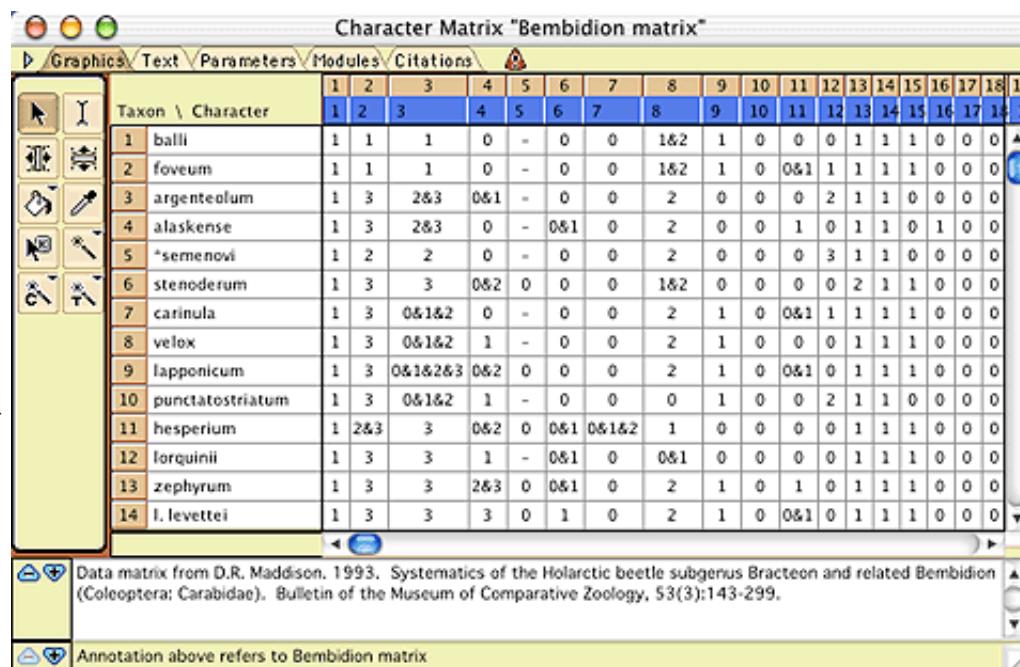
MacClade's. While the Mesquite editor can handle continuous data and has special utilities, for instance to compare matrices, it lacks MacClade's sophisticated features for viewing and alignment of molecular sequence data. Because MacClade and Mesquite share the NEXUS file format, for most data files you will be able to edit matrices in either program to use in the other. Below are instructions as to how to edit a character matrix. Most of the editing can be done in the Character Matrix Editor, but some changes can be made in other windows.

The Character Matrix Editor is controlled by the Matrix and Select menus, and by the tools in the palette at left. The Matrix menu contains items to change column widths (the Display submenu), change cell coloring, and to alter the character data.

You can have more than one Matrix Editor visible for working on the same matrix. To get a second Editor, choose Extra Matrix Editor from the Characters menu. This may be useful if you want to have the editors set to different views (e.g., one on Birds eye view, or colored as translated to protein).

*Note that currently most changes you make to a character matrix cannot be undone!*

The  
Character  
Matrix Editor  
showing  
morphological  
data



The screenshot shows the Mesquite Character Matrix Editor window titled "Character Matrix 'Bembidion matrix'". The interface includes a toolbar with various icons, a menu bar with "Graphics", "Text", "Parameters", "Modules", and "Citations", and a palette on the left containing icons for selection, text, parameters, modules, and citations. The main area displays a data matrix with 14 rows of taxa and 19 columns of characters. The rows are numbered 1 to 14 and labeled with beetle species names: balli, foveum, argenteolum, alaskense, \*semenovi, stenoderum, carinula, velox, lapponicum, punctatostriatum, hesperium, forquinii, zephyrum, and I. levettei. The columns are numbered 1 to 19 and labeled with character numbers. The matrix values are represented by numbers (e.g., 1, 0, -) and symbols (e.g., &1&2, 2&3). A status bar at the bottom indicates: "Data matrix from D.R. Maddison. 1993. Systematics of the Holarctic beetle subgenus Bracteon and related Bembidion (Coleoptera: Carabidae). Bulletin of the Museum of Comparative Zoology, 53(3):143-299." and "Annotation above refers to Bembidion matrix".

## The Character Matrix Editor showing DNA sequences

Taxon \ Character	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46									
1 Homo sapiens	T	C	A	T	A	T	C	G	C	C	C	A	C	G	G	C	T	T	A	C	A											
2 Pan	T	C	A	T	A	T	C	G	C	C	C	A	C	G	G	A	C	T	T	A	C	A										
3 Gorilla	T	T	A	T	A	T	T	G	C	C	C	A	C	G	G	A	C	T	T	A	C	A										
4 Pongo	T	C	A	T	G	A	T	T	G	C	C	C	A	T	G	G	A	C	T	C	A	C										
5 Hylobates	T	C	A	T	A	A	T	C	G	C	C	C	A	C	G	G	A	C	T	A	A	C										
6 Macaca fuscata	T	T	A	T	G	A	T	C	G	C	T	C	A	C	G	G	A	C	T	C	A	C										
7 M. mulatta	T	C	A	T	G	A	T	T	G	C	T	C	A	C	G	G	A	C	T	C	A	C										
8 M. fascicularis	T	T	A	T	A	A	T	C	G	C	C	C	A	C	G	G	G	C	T	C	A	C										
9 M. sylvanus	T	T	A	A	T	T	T	C	C	C	C	A	T	G	G	C	T	G	A	T	C	A										

There are buttons at the lower left of the Character Matrix Editor to open the List of Characters window ( ) and the List of Taxa window ( ). Reciprocally, the List of Characters window has a button ( ) to show the Character Matrix Editor.

## Undo

Mesquite's character matrix editor has some ability to undo the last change made, depending on what that change was. You can request Undo in the Edit menu. Currently, deleting character cannot be undone, nor can deleting taxa. We are working to expand the scope of what can be undone. (If inability to undo concerns you, you may want to turn on automatic NEXUS backup in the Defaults menu of the Log window, and save frequently.)

## Adding, deleting, renaming, merging and sorting taxa and characters

There are several methods for adding taxa or characters to an existing matrix. To add taxa, either choose (Character Matrix) Matrix>Add Taxa... or (Taxa) List>Add Taxa... to add taxa to the end of the matrix, or use the Add Taxa tool ( ) in the character matrix to add taxa at the point in the matrix that is touched. To add characters, either choose (Character Matrix) Matrix>Add Characters... to add characters to the end of the matrix, or use the Add Characters tool ( ) in the character matrix to add characters at the point in the matrix that is touched.

To delete existing taxa, either select the taxa in the Taxa List Window (by touching on the taxon's number at the far left), and choose (Taxa) List>Delete Selected Taxa, or select the entire row for the taxa to be deleted (by touching on the taxon's number at the far left) in the Character Matrix and choose (Character Matrix) Matrix>Delete Selected.

To delete existing characters, either select the characters in the List of Characters Window (by touching on the character's number at the far left), and choose (Characters) List>Delete Selected Characters, or select the entire row for the characters to be deleted (by touching on the characters's number at the far left) in the Character Matrix and choose (Character Matrix) Matrix>Delete Selected.

To rename taxa or characters, choose the I-beam tool ( in the Taxa List Window, List of Characters Window, or Character Matrix Window, select the name to be edited, and type the new name.

Taxa can be merged using **Merge Taxa** in the Taxon Utilities submenu of the Matrix menu. This will also merge their character states in any matrices. If the two taxa have the same states, this state is used for the fused taxon. If one of the two taxa has missing data or a gap (inapplicable), but the other has a state, the other's state is used (e.g., ? + A = A). If one has a gap and the other missing data, the result is missing data. If the two taxa have differing states which are single states or polymorphic, a polymorphism results(e.g., A + G = A&G). If the two taxa have differing states and at least one is ambiguous, an uncertainty results, unless the ambiguity is entirely contained within the polymorphism of the other (e.g., A&C + C/T = A/C/T; A&C&T + C/T = A&C&T).

To change the order of characters, you can select and drag entire characters in the List of Characters Window or the Character Matrix Editor. You can also us the sort tool () to sort characters automatically in alphabetical or numerical order of the column or row on which you touch in these windows.

## Entering character data

You can enter character data either one cell at a time, or using tools that allow entry of multiple cells at once. Tools available are shown in the following table.

Tool	Action
	I-beam Selects individual cell and allows you to edit the contents of the cell as you would any standard text.
	Key Touch If this tool is active, typing a key will cause that single value to be entered into all selected cells. Cells can be selected with this tool; by holding down the Shift or Command keys, multiple cells can be selected.
	Paint Bucket This tool will quickly fill a block of cells with a particular state. The state ("paint") can be chosen by touching the Eye Dropper on a cell of the appropriate state, or by choosing Set Fill States from the Paint Bucket's drop-down menu.
	Eye Dropper This tool, when touched on a cell in the matrix, sets the Paint Bucket's "paint" to the states in that cell.

## Selecting taxa, characters and cells of the matrix

Mesquite has several tools for selecting taxa, characters, or data cells, as described in the

following table.

Tool	Action
 Arrow	Selects individual or multiple cells. To add or subtract cells to an existing selection, hold down the Command (or Apple) key as you touch on a cell. To extend a selection to encompass a solid block of cells, hold down the Shift key as you touch on a cell.
 Wand	By default, selects all cells possessing the same valued state as the cell touched. That is, if you touch it on a cell with state "1", all cells in the entire matrix with state "1" will be selected. However, using the drop-down menu, you can ask it to choose all cells with a value greater than that touched, or less than. By default, this tool selects cells throughout the entire matrix. Using the drop-down menu, you can ask it to restrict the select to a single taxon, or a single character. Holding down the Shift key will add the new cell to the existing selection. Holding down the Command (or Apple) key will add the new cells to the existing selection if you touch on a cell that is not selected, and will remove the cells from the existing selection if the cell is already selected.
 Taxon Wand	By default, selects all taxa possessing the same state within the character touched as that in the cell touched. However, using the drop-down menu, you can ask it to choose all taxa with a value greater than that touched, or less than. Holding down the Shift key will add the new taxa to the existing selection. Holding down the Command (or Apple) key will add the new taxa to the existing selection if you touch on a taxon that is not selected, and will remove the taxa from the existing selection if the taxon is already selected.
 Character Wand	By default, selects all characters possessing the same state within the character touched as that in the cell touched. However, using the drop-down menu, you can ask it to choose all characters with a value greater than that touched, or less than. Holding down the Shift key will add the new characters to the existing selection. Holding down the Command (or Apple) key will add the new

characters to the existing selection if you touch on a character that is not selected, and will remove the characters from the existing selection if the character is already selected.

Taxa, characters and cells can also be selected using items in the Select menu of the Character Matrix Editor. These allow you to select variable characters, to select stretches of sequence matching a currently selected stretch, to reverse the current selection, and perform other changes to the selection.

## Searching the matrix

The contents of the matrix can be searched in two ways. First, the Search area at the top of the window can be set to **Search Data** (set by clicking on the little icon until it shows as ). Then, you can enter a search string and hit return.

Second, you can search the matrix using the Find commands in the Edit menu, and the Select by Search in the Select menu. In the Edit menu, Find String... selects the first instance cell in the matrix containing the given string of text. It searches first taxon names, the character names, the character states within the matrix. You can find subsequent instances using the Find Again command. Find All selects all those cells containing a given string. Find Footnote operates like Find String, except that it highlights cells that contain footnotes with the given text. (To find text within the more elaborate Annotations, you will need to call up an annotations window using the Annotate (pencil) tool in the character matrix editor, then choose Find Annotation in the Notes menu.)

## Copy/paste

You can copy taxon and character names from one region of the matrix to another and from one matrix to another. You can also copy one or more cells in the matrix to the Clipboard, and paste them into another region of the matrix, or into another matrix. Mesquite allows you to do with discontinuous selections, as long as the number of cells selected in the first taxon that is selected while copying is the same as the number of cells selected in the first taxon that is selected while pasting, and the same for subsequent taxa. That is, if you select two cells in taxon 3, one cell in taxon 5, and four cells in taxon 7, and copy this to the Clipboard, then when you paste, you must select two cells in the first taxon in which to paste, one in the next, and four in the last.

Mesquite will not let you paste a block of cells into the matrix while you have selected a differently shaped block in the matrix. However, if you attempt to do that, Mesquite will offer to change the selection so that covers the same number of cells as in the selection. You may then attempt again to paste.

## Editing Names of Characters and States

Character names can be assigned either by editing the column headings in the Character Matrix Editor, or by editing the character names directly in the List of Characters window.

The State Names Editor, available by choosing [\(Character Matrix\) Matrix>Edit State Names](#), allows you to name the states of categorical characters. It will not be available if your matrix is specified as nucleotide or protein data. You can change the orientation (states by characters, or characters by states) of the State Names Editor by touching the double arrow at the top left of the window. Footnotes can be attached to particular states by selecting the state and typing the footnote in the annotation area at the bottom of the window.

## Annotations

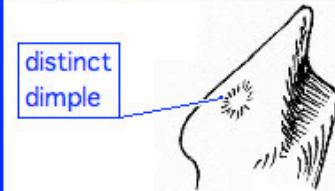
You can annotate the character matrix by attaching simple footnotes, more elaborate annotations, or colors to the taxa, characters or cells of the matrix. Simple footnotes can be attached by selecting the cell with the arrow or I-beam tool, then going to the white annotation area at the bottom of the window and entering the footnote. More elaborate annotations and colors can be attached using the Annotations Panel, available by selecting **Show Annotations Panel** in the Matrix menu. Currently the footnotes and annotations systems are separate in Mesquite – the footnotes appear in the annotation area at the bottom of many windows; the elaborate Annotations appear in a panel embedded in the Character Matrix Editor. An example data file with annotations is at Mesquite\_Folder/examples/Basic\_Examples/characters/11a-annotations.nex

The annotations panel appears at the right side of the window, as follows:

**Character Matrix "Structure & Color"**

Taxon \ Character	1	2	3	4
<b>Fubonidia</b>	<b>knobble</b>	gizmoid	zibinator shape	zibinator setae
1 Fubonidia	dimpled*	pointy	smooth	double sided
2 Snorolopa	dimpled	pointy	edgy	double sided
3 Quidnubia	flat	spatulate	smooth	single sided
4 Zorabynyx	flat	spatulate	smooth	double sided

**Annotations Panel:**

- Comment:** This is a distinct dimple, unlike others among the fubonidians. Quite pretty, actually.
- Reference:** Illustration from specimen AMNH 6788975
- Image:** 
- Author & Modification Dates:** Created Fri Apr 29 18:1 Modified Fri Apr 29 18:1

**Footnotes:**

- This striking dimple was first noticed by Snerp (1965)
- [t.1 c.1 s.1] knobble: (0) flat; (1)\* dimpled; [in taxon "Fubonidia"]

The annotations panel above shows the annotations (if any) associated with a given taxon, character or cell in the matrix (depending on what was selected with the annotation (pencil) tool). Notes can be added or deleted using the (+) or trash buttons at the top left. One image can be added to each note, and labels can be added to the images using the I-beam cursor. To control appearance of these labels, right click or control-click on the label to get a drop down menu adjusting the font, color and other properties of the label. The label's pointer can be relocated using the Adjust Pointer tool. Other features of the annotations can be accessed using the Annotations submenu of the Matrix menu. You can search for annotations containing text using the Find Annotation menu item of the Annotations submenu.

## Coloring cells of the matrix

Cells or their text can be colored. The default background cell color is chosen in the (Character Matrix) Matrix>Background Color submenu. Colors to distinguish different cells can be specified using the items in the (Character Matrix) Matrix>Color Cells and (Character Matrix) Matrix>Color Text submenus. These submenus specify the color to be used for the background of the cell, or for the text within the cell, according to the following criteria:

- Character value – A cell is colored according to a value for the entire character, such as parsimony character steps.
- Cell value – A cell is colored according to a value for that particular cell. For instance, with DNA sequence data, the cells can be colored blue if the site is G or C, white if A or T. By selecting (Character Matrix) Matrix>Moving Window (for colors)..., you can set the size of the moving window over which GC content is averaged. Other cell values are available for amino acid properties (e.g., hydrophobicity)
- Excluded – A cell is colored gray if its character is excluded.
- Footnote present – A cell is colored green if it has a footnote.
- Character State – A cell is colored according to the character state (e.g. different colors for A, C, G, T)
- Annotation attached – A cell is colored green if annotation (not footnotes, but the full complex annotations) are attached to it.
- Assigned Colors – A cell is shown with color as assigned by the paintbrush tool (). To assign a color to a cell, click on the cell with the paintbrush. Touch and hold the button in the tool palette to obtain a menu to select the color used, remove colors or color all selected cells.

When cells are colored, you may request a legend for the colors by selecting **Show Color Legend** in the Matrix menu, or by touching the small button() at lower left of the Matrix Editor (beneath the taxon names). If you double click on a color in the matrix, the editor will move to a cell with that color.

## Alterations and Transformations

The following are available in the Alter/Transform menu of the Matrix menu to modify the cells of a matrix:

- Filling selected cells with a specified state: Choose [\(Character Matrix\) Matrix>Alter/Transform>Fill](#)
- Filling selected cells with random states with equal frequency for all states: Choose [\(Character Matrix\) Matrix>Alter/Transform>Random Fill](#)
- Randomly reshuffle the states within a character among the selected taxa: Choose [\(Character Matrix\) Matrix>Alter/Transform>Other Choices...>Shuffle states among taxa](#)
- For nucleotide sequence data, convert the entries in each cell into their complement: Choose [\(Character Matrix\) Matrix>Alter/Transform>Other Choices...>Nucleotide complement](#)
- Reversing a selected molecular sequence: Choose [\(Character Matrix\) Matrix>Alter/Transform>Other Choices...>Reverse Sequence](#)

## Changing the attributes of characters

A character, in addition to having states assigned in each of the terminal taxa, may also have other attributes. For instance, a character is marked as included or excluded, and it has assumptions attributed to it, such as a weight and a parsimony model of evolution. These attributes are used in various calculations. They may be assigned in the List of Characters windows, available in the Characters menu.

Character	In States	Parsimony Model	Probability Model	Group
50 sac membrane	✓	2	Mk1 (est.)	Male Genitalia
51 ventral bulge	✓	3	Mk1 (est.)	Male Genitalia
52 aedeagal apex	✓	2	Mk1 (est.)	Male Genitalia
53 vaginal plate	✓	2	Mk1 (est.)	Female Genitalia
54 apex spermatheca	✓	2	Mk1 (est.)	Female Genitalia
55 S1 setae	✓	4	Mk1 (est.)	Female Genitalia
56 S2 length	✓	4	Mk1 (est.)	Female Genitalia
57 S2 dors.lat. setae	✓	7	Mk1 (est.)	Female Genitalia
58 stern. X setae	✓	2	Mk1 (est.)	Female Genitalia
59 autosome #	✓	5	Mk1 (est.)	Chromosomes
60 Y chrom size	✓	3	Mk1 (est.)	Chromosomes
61 L1 frontale	✓	3	Mk1 (est.)	Larval Structure
62 L1 head color	✓	3	Mk1 (est.)	Larval Structure
63 post FR3 micro.	✓	4	Mk1 (est.)	Larval Structure
64 FR3 microsculpture	✓	4	Unordered	Mk1 (est.)
65 eggbursters	✓	4	Mk1 (est.)	Larval Structure
66 pron. microtuber	✓	6	Mk1 (est.)	Larval Structure

In the List of Characters window, columns refer to inclusion, parsimony model and probability model (for likelihood calculations). Other columns can be requested for Group Membership, Weight, and (for DNA data) Codon Position. You can ask to show a column using the Columns menu. For each of these columns, the assigned attribute can be changed by first either selecting the characters to be changed (if only some characters are to be altered) or selecting the attribute's column (if all characters are to be altered). Then, by touching the name of the column (where an inverted black triangle should appear), a drop

down menu appears that allows you to make the appropriate specification.

For each of the attributes other than group membership, the bottom three menu items are to store to the file the current specification as a named specification set (like saving a typeset or weightset in MacClade), to replace an existing specification set with the current one, or to load a stored specification set to become the current.

The following are the options specific to each column:

- Inclusion - Include, Exclude, and Reverse allow you to change character inclusion. Reverse changes excluded to included and vice versa. Characters that are excluded don't participate in treelength and many other calculations. Exclusion is not universally respected by the calculations, for some calculations use even characters that are excluded.
- Parsimony model - The Model submenu allows you to select a parsimony model to assign to the characters, for use in parsimony calculations.
- Probability model - The Model submenu allows you to select a probability model to assign to the characters, for use in likelihood calculations and simulations.
- Group Membership - The image above shows Group Membership in the last column. To assign characters to groups, you must first create groups using New Group. For instance, you could create a group Adult Characters and another group Larval Characters. Then, you can assign character to the group using the Set Group submenu. You can also edit the color of the group, and rename the group. The group color is useful to distinguish characters of different groups, for instance in charts or in the Character Matrix editor.
- Weight - With the Set weight menu item you can set the weight assigned to the character. This is used currently on in treelength calculations.
- Codon Positions – This column is available for DNA data. The drop down menu allows you to assign positions.

## Charting information about Characters

Informative statistics and values for characters can be viewed or [charted](#) in various windows. In the Analysis menu a **Bar & Line Chart** can be requested to show the distribution of a value for a series of characters. For instance, the number of parsimony steps in the characters on a current tree can be charted. The **Scattergram** available in the Analysis menu plots characters in a two dimensional space with the X axis being one particular value (e.g., the character's likelihood under one tree), the Y axis another value (e.g. the character's likelihood under a different tree). Values for characters can also be viewed in the **List of Characters** window, where columns can be added (in the List menu) to show selected statistics for each of the characters.

# Taxa

Taxa are the fundamental entities in Mesquite: they represent the species or gene copies whose characteristics are recorded in character matrices, whose relationships are summarized in trees. Our use of the word "taxa" isn't the traditional one, which views a taxon as a formally recognized group at any level (e.g., genus, family, order) in a taxonomic classification. Rather, as with other computer programs, Mesquite uses "taxon" as a shorthand for "terminal taxon" (the smallest unit of analysis of relationships, equivalent to "Operational Taxonomic Unit" or "Evolutionary Unit"). Higher level groups are referred to as clades.

Taxa in Mesquite currently must belong to taxa blocks, which are collections of taxa. Thus, the taxon "Homo sapiens" would belong to a taxa block, for instance one called "Mammalia" which may also contain other taxa such as "Mus musculus" and "Ornithorhynchus anatinus".

## Creating and managing taxa blocks

To add taxa to a data file, a taxa block must first be created to contain them. Mesquite automatically asks you to create a taxa block when you make a new file. You can later create a new taxa block by selecting Taxa&Trees>New Block of Taxa.... A dialog box will ask you the name of the taxa block and the initial number of taxa (you can add or delete taxa later). You will be then shown the List of Taxa window, in which you can rename the taxa (you can also rename them in the Character Matrix Editor).

Mesquite allows more than one taxa block to exist in a file. Thus, there is a List of Taxa Blocks window which shows you all of the taxa blocks defined. To rename a taxa block, edit its name directly in this window. To delete a taxa block, select its row in this window and choose List>Delete Selected Taxa Blocks. Deleting a taxa block may cause character matrices and tree blocks that depend on it to be deleted.

## Managing taxa

You can add and delete taxa in the List of Taxa window and in the Character Matrix editor. In the List of Taxa window, taxa can be added by selecting List>Add Taxa.... In the Character Matrix Editor, taxa can be added either by using the Add Taxa tool or by selecting Matrix>Add Taxa....

Taxa can be deleted in the List of Taxa window by selecting their rows and choosing List>Delete Selected Taxa. Taxa can be deleted in the Character Matrix Editor by selecting their rows and choosing Matrix>Delete Selected.

Taxa can be renamed by editing their names directly in the List of Taxa window or in the Character Matrix editor, or by using the Name tool in the Tree Window.

To change the order of taxa, you can select and drag entire taxa in the List of Taxa Window or the Character Matrix Editor. You can also use the sort tool () to sort taxa automatically in alphabetical or numerical order of the column on which you touch in these windows.

Taxa can be merged, with their character states merged, using [Merge Taxa](#) of the Taxon Utilities submenu.

Different blocks of taxa can be merged during [file import](#) or [file export](#). Otherwise there are not yet methods to merge taxa blocks within Mesquite.

## Renaming taxa

Taxa can be renamed by editing their names directly in the List of Taxa window or in the Character Matrix editor, or by using the Name tool in the Tree Window.

In the List of Taxa window and Character Matrix Editor, the Taxon Names submenu permits you to make changes to taxon names. **Condense Taxon Names**, for instance, assigns shorter names with punctuation removed. This may be useful for converting names to a format for other programs.

## Alternative Taxon Names

This feature allows you to have several alternative naming schemes for taxa, e.g. one set of names that includes your specimen codes, another set that uses formal taxonomic names (for publication), another set whose names are condensed for exporting to programs that can handle only short names, and so on. By switching among naming schemes, you can change how your trees appear, what tree files you can read in, and so forth. (Prior to version 2 this feature used to be called Archived Names and permit just one alternative list of names; it now permits multiple lists and has a different interface.)

You can see and manage alternative naming schemes for taxa in the List of Taxa window. First, in the List of Taxa window choose [Columns>Alternative Names](#). A column will appear showing alternative names (if any). To create a list of alternative names, you can type the names directly in the Alternative Names column. Or, if you want to base the alternative names on existing names, you can touch on the column heading; a drop down menu will appear. Select Replace Alternatives by Taxon Names in order to put the current taxon names into the list of Alternatives.

Once you have a list of Alternative Names, you can save it as a stored alternative naming scheme by choosing Store Alternatives... from the column heading menu. Mesquite can store multiple alternative naming schemes in the file that you can later load, replace, or delete using items from the column heading drop down menu.

For instance, suppose you want to truncate taxon names for export, but you don't want to lose your original names. In the List of Taxa window, with Alternative Names showing, touch on the column heading and select Replace Alternatives by Taxon Names. Then, choose Store Alternatives... and enter the name "Original Names". Now, you've successfully archived the original names. Truncate the taxon names (for instance using [List>Taxon Names>Truncate...](#)). Now, export the file as you wish. Once you're done, you might want to archive the truncated names also. From the column's menu, choose Trade Taxon Names with Alternatives to put the truncated names into the column, then choose Store Alternatives... to store the alternative naming scheme for the truncated names. The file should now have two stored naming schemes, the original and the truncated.

To set the current taxon names to be those in an archived alternative naming scheme, you must first load the alternative scheme into the Alternative Names column of the List of Taxa window, then select Trade Taxon Names with Alternatives or Replace Taxon Names by Alternatives in the column heading's menu.

## Assigning group membership

Taxa can be assigned to groups to form a partition of the taxa. Thus, taxa could be assigned to groups according to a traditional taxonomic scheme (some taxa in the group "Vertebrates", others "Invertebrates"), or according to some quality such as distribution ("Neotropical", "Holarctic"). The reason to assign taxa to groups is not to constrain trees or do formal analyses (these groups in most circumstances don't get involved in formal analyses) but rather for reasons of graphics and interface. Groups can be assigned colors, and thus taxa can be highlighted by their group's color wherever they appear (e.g., in charts, in the Tree Window). (One analytical advantage is in multivariate analysis, where Canonical Variates uses the assigned groups as the prior grouping.)

To assign taxa to groups, go to the List of Taxa window. Select Columns>Group Membership (Taxa) to show the column indicating group membership. If you click on the column heading "Group", a menu will drop down with menu items to manage group membership. You must first create groups using New Group. For instance, you could create a group Neotropical and another group Holarctic. Then, you can assign taxa to the group using the Set Group submenu. You can also edit the color of the group, and rename the group.

## Associating two taxa blocks

Associating the taxa in one block with taxa in another block is useful in two contexts. First, you may want to indicate how genes are assigned to species or parasites to hosts for studies of gene tree/species tree or parasite tree/host tree relationships. Second, you may want to indicate equivalence of taxa in different blocks, e.g., sequence matrices of different gene loci with slightly different taxon names but pertaining to the same set of specimens or species. This can be helpful for fused export into a single matrix.

To set up an association between two taxa blocks, follow the instructions [here](#).

---

# Phylogenetic trees

Phylogenetic trees represent the branching history of descent linking taxa, whether these taxa are species or genes. In Mesquite, a tree refers to the taxa in a particular [taxa block](#) and once created cannot be transferred to refer a different taxa block. (As explained in the page on [taxa](#), "taxon" here is used as shorthand for "terminal taxon" or "OTU".)

## Characteristics of trees

Mesquite typically treats trees as rooted, although it is possible to de-root trees. Trees may contain polytomies (multifurcations) and also unbranched internal nodes. A tree in Mesquite does not need to contain all of the taxa in the taxa block, and indeed can contain as few as one taxon. Unlike MacClade, Mesquite does not support trees with observed taxa fixed in ancestral position. Mesquite can read, edit and write branch lengths in trees. In addition to length, a branch may have various other attributes such as width (e.g. for effective population size in population genetics) and color (for display purposes).

Polytomies in trees are interpreted either as "soft" (uncertainty in resolution) or "hard" (simultaneous divergence). The default interpretation is specified in the Defaults menu of the Startup Window, the Project and Files window, or the Log window. A change in this default applies to all projects and files. Individual trees can be marked as using a specific assumption, thus overriding the default (e.g., by using the Set Polytomy Assumption menu items in the Alter/Transform Tree submenu of the Tree menu of the Tree Window). The appropriate assumption for most phylogenetic studies is "soft", but calculations using soft polytomies are extremely difficult, and most Mesquite calculations yield results only with dichotomous trees and those with hard polytomies.

## Analyzing trees

Trees can be visualized in various tree windows, and statistics about them presented in [tree windows](#), the [List of Trees](#) window, and in [charts](#). We will not attempt to summarize all of the options here, which are discussed elsewhere, in particular in the analysis links at left.

Tree comparison methods include the following:

- **Tree to tree similarity measures** – The Shared Partitions module measures the number of partitions between taxa shared by two trees. The separately-available [TSV](#) package includes other measures such as the Robinson-Foulds metric.
- **Consensus trees** – This is not available standardly in Mesquite, but can be added with the TSV package, which includes strict consensus trees.
- **Fits of trees into trees** – The fit of a contained tree (e.g. gene or parasite) into a containing tree (e.g. species or host) can be measured by [Deep Coalescence](#) in the coalesce package.

- **Taxon instability among trees** – This module measures for each taxon how variable are its relationships among a set of trees. Taxa that are particularly unstable, i.e. whose placement is especially variable from tree to tree, score high on this index. Taxon instability calculations are illustrated in the example file at Mesquite\_Folder/examples/Basic\_Examples/tree\_manipulation/13-instability.nex

## Stored trees and tree blocks

Trees may exist within **tree blocks** that are stored in the data file. A tree block is a collection of trees stored as a unit. A single data file may contain more than one tree block, each containing from one to many trees.

Mesquite calculations can use trees other than those stored in tree blocks in a data file. Most calculations can use trees that are temporarily created using simulations or randomizations specifically for the purpose of the calculation and then discarded. Also, a recently edited tree in the Tree Window might not be stored in the file if the user has not explicitly stored it in the file using "Store Tree" or "Store Tree As". More details on storing trees in tree blocks in the file are given [below](#).

To create a new, empty tree block into which to store trees, select [Taxa&Trees>New Empty Block of Trees...](#) A new block of trees is automatically created if you ask in the Tree Window to store a tree and no block has yet been created to receive trees.

It is also possible to create tree blocks that are already filled with trees. For instance, if you want to make a duplicate of an existing tree block, select [Taxa&Trees>Make New Trees Block from>Stored Trees](#). Other choices under [Taxa&Trees>Make New Trees Block from>](#) allow you to create tree blocks filled with trees resulting from simulations or randomizations, or other tree sources.

Trees stored in a tree block can be renamed or deleted in the **List of Trees window** (available in the Taxa&Trees menu). To rename, edit the tree's name in the list directly. To delete a tree, select the tree's row in the list window and choose [List>Delete Selected Trees](#). Statistics about trees can also be viewed in the List of Trees window by adding the appropriate columns using the List menu.

Tree blocks stored in a file can be renamed or deleted in the **List of Tree Blocks window** (available in the Taxa&Trees menu). To rename, edit the tree block's name in the list directly. To delete a tree block, select the tree block's row in the list window and choose [List>Delete Selected Tree Blocks](#).

For tree files with too many trees to be held in memory simultaneously, Mesquite can read the trees in directly from the file, one at a time, and discard each from memory after it is used (for instance, in charts or in the Tree Window). To do this use Trees Directly from File as your tree source.

## Tree Sources

Trees for viewing and analysis can come from various sources, including tree blocks in the current file, or [simulations](#). Some tree sources modify other trees, such as **Randomly Modify Current Tree**. **Trees Directly From File** is useful for large treefiles. Although slow at times, it does not store all the trees in Mesquite's memory, but instead gets them from the file one at a time as needed. Another tree source, [MrBayes Trees](#), obtains trees directly from a tree file stored by MrBayes. Mesquite can updated its calculations as new trees come in from MrBayes.

## Filtering Trees

Trees used in charts, the tree window or to make a new tree block can be filtered. To do this, select Filter Trees from Other source as your source of trees. You will be asked for the original source of trees, and the filtering method. There are currently three filtering methods:

- Selected Taxa Form Clade - retains only those trees in which the currently selected set of taxa form a clade on the tree
- Selected Taxa Convex in Tree - retains only those trees in which the currently selected set of taxa can be form a clade by rerooting the tree
- Tree Value Satisfies Criterion - retains only those trees whose value is greater than, less than or equal to a particular target value. For "value", one can use any of the available calculations that assign numbers to trees, such as treelength, tree depth, likelihood in character, tree asymmetry, and so on. The user is also asked what is the target value, and whether the criterion is to be greater than, less than or equal to the target.

## Modifying trees automatically

Trees can be edited by hand, as described [below](#), or the user can request Mesquite to automatically modify trees using the "Alter/Transform" modules. These modules can be used:

- in the Tree menu of the Tree Window, under the Alter/Transform Tree submenu, or the Alter/Tranform Branch Lengths submenu. Selecting an item in these submenus performs the transformation on the tree in the Tree Window.
- in the List menu of the List of Trees window, in the Utilities submenu, in the Alter Trees or Alter Branch Lengths submenus. Selecting an item in these submenus performs the transformation on all selected trees, or on all trees if none is selected.
- to transform trees on the fly that are being supplied for a chart or tree window or making a new tree block for a file. To do this, select "Transform Trees from Other Source" whenever a source of trees is required. You will be asked for a source of trees to be transformed or filtered, and then for a means by which to transform the trees.

The Alter/Transform modules include:

- Root tree with selected taxa as outgroup
- Resolve Polytomies (to 0-length branches)
- Collapse Zero-length Branches
- Convert Negative Branch Lengths to Zero
- All Branch Lengths to 1
- Scale All Branch Lengths
- Ultrametricize

To use "Root tree with selected taxa as outgroup", you first must select the outgroup taxa in, for example, the Taxa List window. Then, choose the menu item "Root tree with selected taxa as outgroup" in the Alter/Transform Tree submenu of the Tree menu of the tree window, and in the Alter Trees submenu of the Utilities submenu of the List menu of the List of Trees window. This module will reroot the taxa in between the selected outgroups and the remaining taxa, if this is possible. If the tree cannot be rerooted so that the remaining taxa form a clade, then the tree will not be rerooted, and a warning will be given (perhaps visible only in the log). If the tree can be rerooted, then both the outgroup and ingroup will be monophyletic on the rerooted tree. In addition, if the tree has branch lengths specified, the root will be positioned at the midway point along the branch on which it is placed. That is, the length of the branch from the root to the ingroup will be equal to the length of the branch from the root to the outgroup. Future versions may allow user-choice of the branch length distribution. Adjusting of the length of the two basal branches is NOT done if the tree is already rooted between the chosen outgroups and ingroup.

## Editing trees: The Basic Tree Window

The Tree Window shows a tree that can be edited and used in analyses. A new Tree Window can be requested by selecting Taxa&Trees>New Tree Window. (Mesquite allows multiple tree windows, and so if you select this menu item a second time, you'll get a new tree window.) Many of the Tree Window's fundamental functions are controlled by its Tree menu. This menu contains items to select the tree source, store trees, and alter the tree. A second important menu is the Drawing menu, which controls the appearance of the tree. Additional menu items related to the Tree Window are found in the Analysis menu.

### Tree source

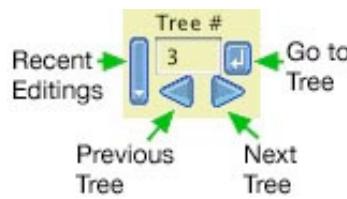
The Tree Window shows trees from a particular source (although the tree being shown could differ from those in the tree source, if you've recently modified the tree using the tools). The source of trees might be a tree block in the data file. In this case, as you used the blue arrows at the upper portion of the tool palette, you would be browsing through the stored trees, scrolling from one stored tree to another. Alternatively, the source of trees might be a coalescent simulation, in which case you would be browsing through gene trees representing replicate simulations of the coalescent process. Other tree sources may be available, and can be selected when you request the Tree Window for the first time or by

using the submenu Tree>Tree Source>. One tree source, "Default Trees", offers a simple pectinate tree ("Default Ladder"), a full polytomy ("Default Bush"), and a symmetrical tree. It is available merely as a last resort, in case other tree sources are unavailable.

We expect a common confusion will be that users will be unable to find the trees that they recently stored in the file. For example, when a new file is created, there are no stored trees, and hence the Tree Window would be forced to use another tree source (for example, "Default Trees"). If a user then stores a tree in the file (see next section), he or she might hit the blue scroll arrows of the Tree Window expecting to browse the stored trees, only to be shown one of the Default Trees. The problem is that the Tree Window is using Default Trees as the tree source, not Stored Trees. To see the trees stored in the file, select Tree>Tree Source>Stored Trees to change the tree source to Stored Trees.

## Moving from Tree to Tree

To select which tree to view, use the tree scroll at the upper left of the Tree Window:



The left and right blue arrows take you to the previous and next trees, respectively, in the tree source. The Go To button takes you to the tree whose number is entered in the text area. The Recent Editings button offers you a drop down menu by which you can return to recently edited trees. The menu lists only trees that you have edited; it does not list the trees that came directly from the tree source, and is therefore not a complete history of recent trees. The number of little white triangles in the button indicates the number of recently edited trees stored.

## Storing trees

Although the Tree Window can be used merely to browse existing trees, it will commonly be used as a tree editor, allowing the user to build a tree according to some prior hypothesis, or to modify trees to explore the effect of changes in the tree.

Once a tree has been edited, the user may want to store this new tree in the file. Exactly how that is done depends on the tree source being used by the Tree Window:

- If the tree source is Stored Trees, then two menu items are available in the Tree menu to store trees, "Store Tree" and "Store Tree As...". For instance, imagine that you scroll through the stored trees until you get to tree number 5. If you edit it using the tools of the tool palette, then you select Store Tree, your newly modified tree will replace the original tree number 5 in the tree block. If instead you choose Store Tree As, you

will add your modified tree as a new tree at the end of the tree block, leaving the original tree number 5 untouched. Thus, Mesquite assumes that you are potentially editing the original tree in the tree block whenever you edit the tree. However, the original copy is not replaced by your edited version until you select "Store Tree". Until you select this, your modified tree is maintained as a temporary tree associated with the tree window, and not stored in the tree block.

- If the tree source is not Stored Trees then you can't modify the original trees, because they are merely temporary trees, produced by something like a simulation. Thus, in this case there is no Store Tree menu item available. Instead, there is a "Store Copy of Tree As" menu item. This acts more or less like Store Tree As, in that the tree is added to a tree block.

You can tell that a tree has been edited to be different from that in the tree source when a black diamond appears in the message area at the lower left side of the Tree Window. The message area turns green when the tree is an unsaved, edited tree and the tree source is not Stored Trees.

## Tools

At the left side of the Tree Window is a **tool palette**, containing tools that you can use to interact with the tree. Exactly which tools are available will depend on the modules installed and loaded. In the description of tools below, it will be assumed that a basic set of modules is installed and loaded.

Some tools act when they are touched on a branch of the tree; others act when a branch is touched then dragged and dropped. Some tools behave differently if a key such as shift or control is held down when the tool is used. These details are explained below. Some explanation of the tool is also given in the window's Explanation Area when the tool is selected in the palette. The currently selected tool has its button darker than the rest in the tool palette.

Some tools have options that can be set. If so, then the button for the tool in the tool palette will have a small black triangle indicating the availability of a drop down menu. If you touch the button and hold down the mouse for a moment, the menu will appear to allow you to make choices.

The following tools **change the topology** of the tree (the fundamental relationships among taxa implied by the tree). Some of these tools might not be available if their controlling modules are not installed or loaded.

Tool	Action
 Arrow	If touched on a branch, dragged and dropped on another branch, the former branch is reattached to the latter. Also can be used to select taxa. If touched on a branch, dragged and dropped on

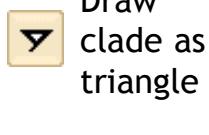
 Interchange branches	another branch, the two branches exchange positions. Can be used to rearrange a polytomy, but can also be used for branches distant on the tree.
 Collapse branch	Collapses branches to yield polytomy. Also removes unbranched internal nodes.
 Collapse all	Collapses all branches in clade. If option or ALT is held down, collapses all branches below node touched.
 Reroot	Reroots tree at branch touched.
 Prune clade	Cuts clade out of tree. Taxa cut out remain within the data file, but are not included in this particular tree.
 Insert unbranched node	Inserts node along branch. Currently used primarily in combination with assigned lineage widths to indicate population fluctuations for coalescent simulations.

The following tools change the **branch lengths or widths** of the tree.

Tool	Action
 Assign branch length	Brings up a small editable area in which to enter a branch length.
 Stretch branch	Click and drag on branch to stretch its length.
 Adjust node position	Stretches branch lengths above and below node to allow node to change position without changing positions of other nodes.
 Assign lineage width	Assigns a width to the branch. Currently used to assign effective population sizes for population genetics calculations.

The following tools affect the **appearance** of the tree, or change attributes of its branches, but do not change the topology or branch lengths. Thus the changes these tools make will not affect most calculations.

Tool	Action
 Ladderize	Rotates nodes to bring the largest clades on the right side of the tree. Does not affect topology of tree. If option or ALT key held down, ladderizes to left.



Draw clade as triangle  
Draws clade as small triangle to save space in drawing the tree. The full relationships within the clade remain, but are hidden. Clicking again will expand the clade to be viewed again. This drawing mode may have bugs.



Magnify clade  
Magnifies the image of the tree so that only the clade touched is shown. Touching again on the node shows the whole tree



Branch info  
Shows available information about the branch and calculations concerning it.



Color branch  
Colors the branch. Control-click colors clade. Repeat click erases. Shift-click shrink wraps color. Touch and hold the button to obtain a menu to select color used. This is not equivalent to the Fix States tool of MacClade.



Name node  
Gives a name to the node. This name must be unique, and different from the names of terminal taxa.



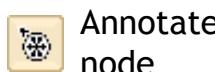
Select branch  
Selects a branch. Control-click extends selection. Shift-click shrink wraps selection.



Select branches in clade  
Selects all the branches in a clade. Control-click extends selection. Shift-click shrink wraps selection.



Select taxa in clade  
Selects all the taxa in a clade. Control-click extends selection. Shift-click shrink wraps selection.



Annotate node  
Attaches a note to a node. An asterisk appears over the node. The note can be seen by passing the Annotate cursor over the branch, or by turning on "Show Notes On Tree" using the menu that drops down from the Annotate node button in the tool palette.



Show picture  
Shows image attached to taxon. Click again to hide. Control-click to attach picture.



Hyperlink  
Shows web page or another data file. Shift click to enter link explicitly. Control-click to select local file.

## Alter/Transform menu items

In the Tree menu of the Tree Window are two submenus by which you can change the tree, "Alter/Transform Tree" and "Alter/Transform Branch Lengths". These provide various utilities to change the tree, including its branch lengths. Some, such as "Scale All Branch Lengths", affect all branches of the tree simultaneously. See the section on [automatic modification](#) for more details.

## Analyses

The Tree Window has, in its **Analysis** menu, menu items that yield analyses using the current tree in the Tree Window. Which analyses are available depends on what modules are installed and loaded. Typically the following will be available at least:

- Trace Character History - Illustrates a history of character evolution on the tree. Described in detail in the chapter on [reconstructing ancestral states](#).
- Tree Legend - adds a small legend to the Tree Window. The legend can describe information about the tree, including the results of calculations done using the tree. When the legend is in the Tree Window, an additional menu appears, the Legend menu, by which you can choose to show or hide lines of information.
- Values at Nodes - shows values associated with nodes, either by coloring the nodes or labeling them. These values describe information about the nodes, such as a statistic calculated over the clade above the node. At the moment there are few such values available (notably, the reconstructed character state in a continuous character, also available through Trace Character History).

## How trees are drawn

There are many options for the appearance of trees as drawn in the Tree Window and other windows displaying trees. Trees may have diagonal, square or circular branches; they may be drawn black on white or green on blue; they may be drawn so that apparent branch length is proportional to assigned branch length. These options are controlled by the Drawing menu.

Some of the menu items in the Drawing menu are:

- Tree Form - Chooses general form of the tree. Some are merely graphical, others involve analyses such as PlotTree and PlotTree3D (which plot the tree in a 2D or 3D space) and Contained Associates (which shows gene trees within species trees, or parasites within hosts).
- Set Current Form As Default - Sets as default the tree drawing style currently in use (e.g. Diagonal Tree, Balls & Sticks etc.). It does not capture all of the current parameters chosen for that drawing form (background color, spot sizes, etc.).
- Background Color - Sets the color of the field on which the tree is shown.
- Branch Color - Sets the color of the branches. These colors are not retained with the tree, but are for graphical purposes only. In this regard it is different from the Color Branch tool, whose colors are retained with the tree.

- Size to Window (Only in the basic Tree Window) - Shrinks the tree to fit into the window. If this is turned off, then the tree exists in a panel that may be much bigger than the tree window itself. To see various parts of this panel, you can use the scroll bars or the Bird's Eye View box that appears at the lower left corner of the Tree Window.
- Drawing Size (Only in the basic Tree Window) - When Size to Window is turned off, this item allows you to set the size of the panel in which the tree is drawn.
- Float Legends (Only in the basic Tree Window) - When Size to Window is turned off, this item allows you to choose whether the legends such as the Tree Legend and Trace Character Legend are anchored to the panel in which the tree is drawn or, anchored to the window itself.
- Font, Font Size, Font Color - Sets the font characteristics of the taxon names and node names.
- Names - Determines what names are shown and whether taxon names are colored by the [taxon group](#).
- Orientation (Only for some tree forms) - Chooses whether the root to terminal orientation is up (classic phylogeneticist or paleontologist), to the right (molecular evolutionist), bottom (population geneticist or mathematician) or left (?)
- Branches Proportional to Length - Determines if the drawn lengths of branches are proportional to their assigned lengths. Otherwise, the tree drawing routine chooses node positions at its own convenience.
- Save Macro for Tree Drawing - Saves the current tree drawing specifications (form, color, orientation, etc.) as a macro. The macro will then appear under the Macros For Tree Drawing submenu, and can be used later to set other tree windows to use the same specifications.

## Printing trees and saving graphics files

Mesquite's Tree Window offers two menu items for printing trees: Print Tree and Print Tree to Fit Page. The former prints the tree in its current size, even if that requires it to be placed over multiple pages. The latter automatically reduces or enlarges to tree image to fit a single printed page. It attempts to choose landscape or portrait mode to maximize the size of the fitted image.

You can also use Save Tree As PDF in the file menu to save the tree image to a PDF file. This works well except that legends (such as the character trace legend) are not included in the image because of an apparent bug in the library we are using.

An alternative way to generate image files is to ask to print the tree image, then use your operating system's Print dialog box to direct the output to a file (e.g., postscript or PDF) instead of a printer. Printing to a PDF file is built into the Print Dialog box of Mac OS X and therefore is easy to do. On Windows, printing to a pdf or postscript file is not so straightforward. To do this, use a postscript printer and print to file. The ".prn" file created might in fact be a ".ps" postscript file on the inside. If you are using Windows but don't have a postscript printer then you can add a virtual printer via the Control Panel. Choose a

postscript printer driver (e.g. a Laserjet marked as "PS"). Postscript and pdf files are editable by programs such as Adobe Illustrator.

## Dependent Tree Window

This window shows the same tree as in the Tree Window (hence it is "dependent"). It is available in the **Tree** menu of the Tree Window. It is useful when analyses or graphics are desired for the Tree Window, but which would conflict visually with currently running analyses or graphics. Thus, the Dependent Tree Window gives additional space on which to display graphics and analyses. As the tree in the Tree Window is changed, the tree in the Dependent Tree Window is also changed. Note: If the Tree Window is closed, the Dependent Tree Window will close also.

Users may find the menus of the Tree Window and Dependent Tree Window somewhat confusing when the latter is in use, as both windows include some menus of the other. This is a consequence of Mesquite's automatic menu arrangement; we realize it is not optimal in this case.

## Mirror Tree Window

This window, like the Dependent Tree Window, shows the same tree as in the Tree Window. However, the Mirror Tree Window shows it twice, in mirror image. This allows you to compare, for instance, two different traced characters visually, as in [this example](#). The analysis shown on the left side is controlled by the Left side submenu; that on the right by the Right side submenu.

Users may find menus confusing when this window is in use; see comments under Dependent Tree Window.

## Multi Tree Window

This window, available in the **Taxa & Trees** menu, shows multiple trees simultaneously. It is not dependent on a Tree Window, but gets its trees from an available tree source.

# Publication Quality Trees

Mesquite is not designed for rigorous tree-estimation procedures, such a Maximum Parsimony, Maximum Likelihood, or Bayesian tree estimation. However, trees generated from programs such as PAUP, PHYLIP, and MrBayes can be read into and manipulated in Mesquite. This guide will help you perform these manipulations to create a tree usable for scientific publication. Users should begin by familiarizing themselves with the [Mesquite file structure](#).

## Step 1: Get a tree

- **Within Mesquite** – Mesquite can provide trees from simulations, some basic tree searches, or default trees. If you do not have trees from other sources, you can open a tree window from the Taxa&Trees menu by selecting [New Tree Window](#). If you have multiple taxa blocks in the file, you will be prompted to choose among the blocks included in the file. You will then be asked to choose a tree source. There are at least 13 different sources for trees, but we will only deal with the first two here. "Stored Trees" refers to trees that are stored in the Mesquite file. If you haven't stored any trees in the file yet, this option won't be available. Selecting "Default Trees" will open a Tree Window corresponding to a Tree Block with three trees: a fully resolved symmetrical tree (Default symmetrical), a completely unresolved tree (Default bush) and a ladderized tree (Default ladder) (the names of trees are shown in the panel beneath the the tree). You may manupilate any one of these trees, but be sure to store a copy of the tree in the Mesquite file ([Tree > Store Copy of Tree As...](#)), or it will be lost.
- **Other sources** – When importing trees from other files, be sure that the taxon names from the tree file match the taxon names in the Mesquite File. Although importated trees may contain fewer taxa than are in the Taxa Block, they cannot contain taxa that are *not* included in the Mesquite file. If the trees you wish to import contain taxa that are not represented in the Taxa Block you should either (1) add those taxa to the correspondidng taxa block in Mesquite ([List > Add Taxa...](#) from Taxa Window) or (2) delete those taxa from the tree file before importing them into Mesquite. We recommend the former (adding Taxa to file), in order to reduce the probabiltiy of inadvertently altering tree structure.

Importing trees from files can be done through [Taxa &Trees > Get File with Trees...](#) menu option. To include the trees in the Mesquite file, select "Include Contents...".

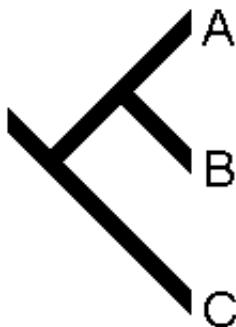
Alternatively, you may open a Tree Window corresponding to trees from external sources from [Taxa&Trees > New Tree Window](#) by selecting "Trees Directly From File" as the source of trees for the new Tree Window.

**Note:** if you are importing trees from PAUP, and would like to retain the branch lengths, you must be sure to save the branch lengths from within PAUP. From the command line in PAUP, use the `brlens = yes` option in the `savetrees` command. If you

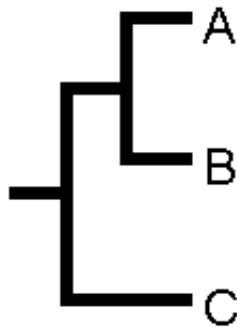
are using PAUP with menus, you can save the branch lengths when saving the tree (**Trees > Save Trees to File...**) by clicking on the Options... tab and checking the box **Include branch lengths**. If you want Mesquite to display bootstrap values generated by PAUP, then in the **savetrees** command use **savebootp = brlens** option; in the version of PAUP with menus, use the Options... tab and ask to save bootstrap proportions as branch lengths.

## Step 2: Choose Tree Form

Some examples of Tree Forms:



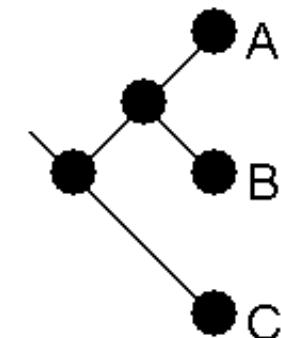
Diagonal



Square



Curvogram



Balls N Sticks

Most trees are published as "Square Trees", which you can select from the Drawing menu in the Tree Window. There are a variety of other formats available, which we encourage you to explore, but some of the following instructions will only apply to square trees (this is noted where it applies).

## Step 3: Branch Lengths

To show branch lengths of a tree, select Drawing > Branches Proportional to Lengths. The branches will now reflect lengths, and a scale is shown. You can toggle the scale off/on from the Drawing menu (Drawing > Show Scale). Note: if the branch lengths for a tree are unassigned, and Branches Proportional to Lengths is selected, the tree will be drawn with arbitrary branch lengths (usually so that each branch has a length of 1.0) and the scale shown. If you don't know if the tree has branch lengths assigned, see the instructions for viewing the tree in newick format, below. Alternatively, you can check branch lengths by using the "Adjust

Branch Length" tool (); if a branch does not have a branch length assigned, the box that opens when you use the Adjust Branch Length tool will show a "?".

## Step 4: Branch Colors

There are two approaches to coloring branches, the first colors all branches of the tree, and the second applies different colors to different branches (there is a third means of coloring branches, according to Character Reconstructions, but for this step, we only discuss the first

two). To color all the branches of a tree, select Drawing > Branch Color and choose from the list of available colors.

To color individual branches, you can use the Branch Coloring tool () from the Tree Window tool palette on the left-hand side of the Tree Window. This tool can be used to add colors for cosmetic effect to a tree (these colors play no role in analyses, and are not dictated by character states per se. See "Character Reconstructions" under step 6 below for branch coloring that based on character reconstructions). To set the color, right/control-click on the Branch Coloring tool icon, select Branch Paint Color, and choose from the list of available colors. Clicking on a branch with this tool will color that branch, control-clicking on a branch will color that branch, and all the descendant branches.

You can remove colors from branches by right/control-clicking on the Branch Coloring tool icon, and selecting Remove Color. The Remove Color tool works the same way as the branch coloring tool on single branches (click) and entire clades (control-click). You can also remove all colors from a tree by selecting Remove All Color from the Branch Coloring Tool icon options. It is important to note that colors added with the Branch Coloring tool will always be on top - any color changes applied from the Drawing > Branch Colors will not be visible on branches that have been colored using the Branch Coloring tool.

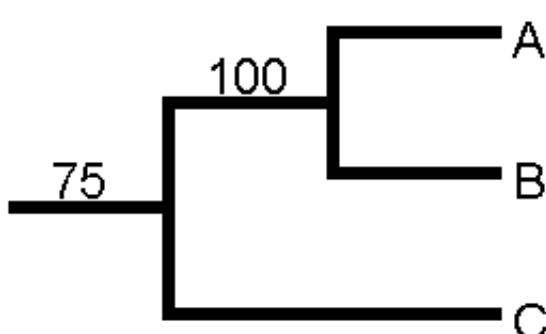
## Step 5: Taxon Name Colors

To color taxon names, you can partition the taxa into groups (groups are not the same as associations). To set a group, first go to the Taxa window for the Taxon block that contains the Taxon names you'd like to color (this will be the taxon block that corresponds to the tree window you are looking at). There should be a "Group" column on the list. If not, you can show the column by selecting Columns > Group Membership (taxa). First, select all the taxa in your first group. Using the small black triangle at the top of the column, select New group... (clicking once on the black triangle will bring up the Group menu). In the New Taxon Group dialog, type a name for this Group, and select a color. From the Group menu (accessible from the top of the Group column), you can create additional new Groups, edit current Groups, and set Group membership. For the latter, select the taxa you wish to place in the group, and select Set Group > from the Group menu (the list of groups is shown). Each taxon can only be assigned to a single Group. The taxon names in the Tree Window should be the same color as assigned to the Group. If they are not, you can color them so by selecting, in the Tree Window, Drawing > Names > Color by Taxon Group. Alternatively, you can color the background of the taxon text by taxon group by selecting Drawing > Names > Background Color by Taxon Group. Of course, you'll likely want to turn off Color by Taxon Group, or the names will be the same color as the background, and thus won't be visible! Because taxa can only be assigned membership to one Group, if you would like to draw more than one tree, which differ only in Taxon Name coloration (topology and branch lengths being the same), it would be easiest to draw the tree first with one set of Group memberships, then save the file under a different name, and in the new file, change the Group memberships. This will make it easier to assign different Group memberships without overwriting other desired memberships.

## Step 6: Extras

- **Support Values as Branch Labels** - If you would like to add support values to your tree, there are different ways to do so, depending on how you initially got the trees. Some programs save values as branch labels, then this section applies (if they program saves values as branch lengths, then see the next section). Bayesian trees from MrBayes usually contain the posterior probabilities, so they should be visible when you load the trees into Mesquite. If they are not, and the posterior probabilities are in the tree in Newick format, you can turn them on by selecting [Drawing > Names > Show Branch Names](#). If you're not sure if the support values are in the Newick format tree, click on the "Text" tab in the Tree Window. You will see the line "Tree Description", which shows the tree in Newick format. If the support values are present, they will be evident as a string of characters immediately following the right parenthesis of a clade.

For example, this tree,



in newick format, lacking branch lengths, would be:

$$((A, B)100, C)75$$

The same tree, if it had branch lengths of 10 and 5 for clades AB and ABC respectively:

$$((A, B)100:10, C)75:5$$

If the tree does not show support values, you will need to enter them using the Node Namer tool ([NAME](#)). The default position for support values is on the node of the tree; however, published phylogenetic trees usually have support values centered on the branches. If you are using a Square Tree, you can center the support values on a branch by selecting [Drawing > Names > Center Branch Names](#) (note: this option is currently only available for Square Trees). If the support values overlap (this will be an inevitability as the number of taxa & support values increases), you will need to save the tree in pdf format (step 7 below) and use a graphics editing program (Illustrator, Photoshop, etc.) to correct the overlap.

- **Support Values as Branch Lengths** - If support values are saved as branch lengths (for example, if you used the `savebootp = brlens` option in the `savetrees` command to save the tree resulting from a bootstrap analysis in PAUP), then you can see the values on

that tree in the tree window by choosing [Drawing > Branch Length Labels > Show Labels](#). The [Branch Length Labels](#) submenu allows you to adjust how the values are displayed, including the number of decimal places (for standard bootstrap values, you might wish to change this value to zero), whether or not the terminal branches have values displayed, and the color of the label.

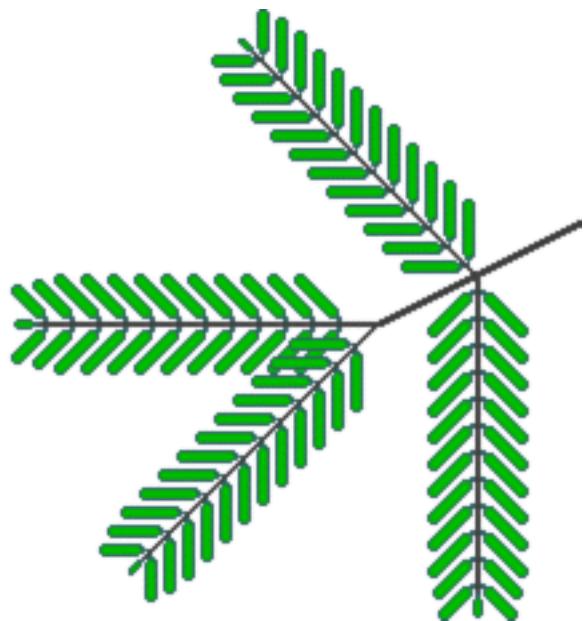
- **Character Reconstructions** - You can trace character reconstructions on trees by selecting [Analysis > Trace Character History](#). Most likely, you will be tracing Stored Characters. Note that tracing a character history will "fill" the branches with a particular color, so that any colors added using the Branch Coloring tool or the Branch Color menu item in the Drawing menu will only outline the branch. [Studying the History of Character Evolution](#) has more detailed information regarding ancestral character reconstruction, and additional instructions may be found in the [Mesquite FAQ](#) page.

## Step 7: Exporting the Tree

When you're satisfied with the tree in the Tree Window, select [File > Save Tree as PDF...](#) (you can add the extension ".pdf" when you save the tree. Mesquite will not do this automatically, but applications capable of reading pdf files will still be able to open it without the extension).

---

# MESQUITE 2



*Analyses*

# Character Evolution

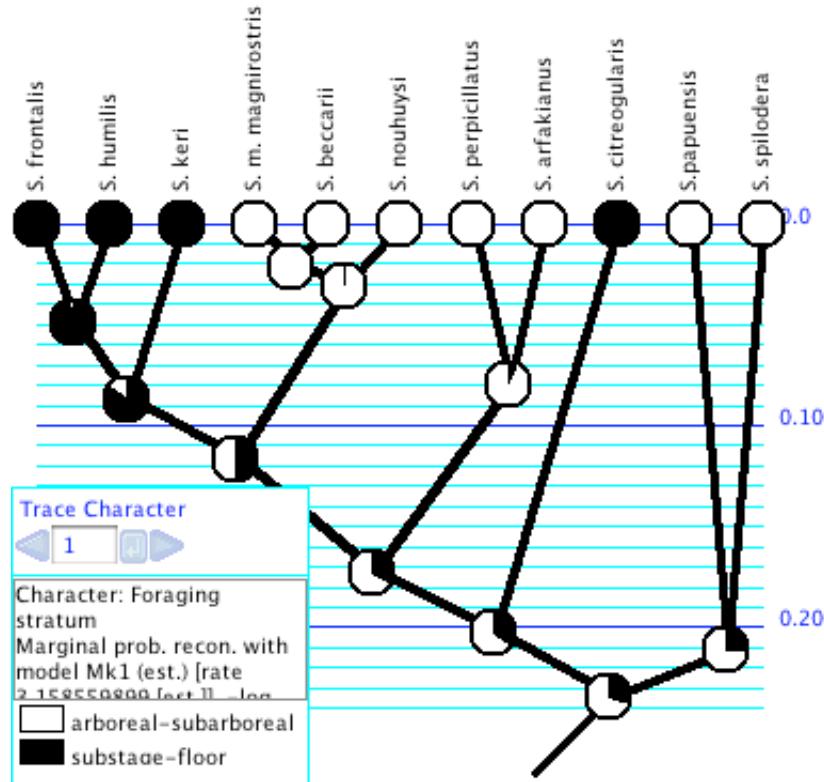
The evolution of characters (measurements, observations) among taxa (species, genes) can be studied both to infer history and interpret processes of change.

- [History: Reconstruction of ancestral states](#) (Parsimony methods for molecular, categorical and continuous data; Likelihood methods for categorical data; Stochastic character mapping)
  - [Processes of character evolution: parameters and correlations](#) (Estimating parameters of one and two parameter models for categorical characters; Pagel's 1994 correlation test; Felsenstein's Independent Contrasts; Pairwise comparisons tests of character correlation; Visualizations to explore character correlations)
-

# Studying the History of Character Evolution

With a phylogenetic tree and a distribution of character states in the observed (terminal) taxa, Mesquite can attempt to reconstruct the character states at ancestral nodes. Two separate issues to consider are the method by which the reconstruction is done, and how its results are displayed to the user. Mesquite currently can use either parsimony, likelihood or Bayesian methods to reconstruct ancestral states, and has several display methods, including "Trace Character History" which paints the branches of the tree to show the reconstruction.

We recommend highly that you examine the **example files** provided in the folder "Ancestral State Examples". The minimal configuration to use with these examples is "Ancestral States" (indicate this configuration under File>Activate/Deactivate Packages>Choose Configuration), but you can also leave Mesquite in its default All Installed Modules mode.



## Trace Character History

The Trace Character History facility graphically represents a history of character evolution on the tree. It is available under the Analysis menu of a tree window (e.g., the basic Tree Window, Dependent Tree Window, Mirror Tree Window, Multitree Window). If you select this you will probably be asked for a **source of characters** (e.g., stored characters) and a **reconstruction method** (e.g., [parsimony](#), [likelihood](#), [stochastic character mapping](#)). The tree will be painted to show ancestral states, and a trace legend will appear. The **Trace Legend** contains an important text area that gives details of the current ancestral state

tracing. You can also see details of the reconstruction by switching the window to Text mode using the tabs at its top.

For categorical and molecular data, you can change the colors used in Trace Character by double clicking on the color rectangle in the Trace Legend. Revert to Default Colors is available in the Trace menu.

For parsimony reconstructions, any tree drawing style will suffice. For likelihood reconstructions, we recommend the Balls&Sticks style (Drawing menu, Tree Form) with Line Style "Square". This permits you to see the relative likelihoods and branch lengths. For stochastic character mapping, we recommend the Square Tree style in order to display the changes within a branch.

The Trace menu gives menu items to control the character history and its display. Some important ones are:

**Character history source:** Typically you may use Trace Character History to reconstruct the ancestral states of an observed character. Alternatively, you can trace a simulated history ("Simulate Ancestral States"). The reconstructed states need not be based on actual data, but could be based on simulated data. "Simulate Ancestral States" shows the "actual" history of character evolution branch by branch as it occurred in the [simulation](#), not as it was reconstructed, and thus may show ancestral states that would be unreconstructable, obliterated by subsequent changes.

**Next, Previous, Choose Character History:** Usually, this will allow you to choose which character to view. You can also scroll through characters using the blue arrows in the trace legend.

**Trace Display Mode:** With [Trace>Trace Display Mode>Shade States](#) and [Trace>Trace Display Mode>Label States](#) you choose whether to see the branches painted, or have the states indicated in labels. If painted, you can also ask that states be indicated by labels by choosing [Trace>Label States](#).

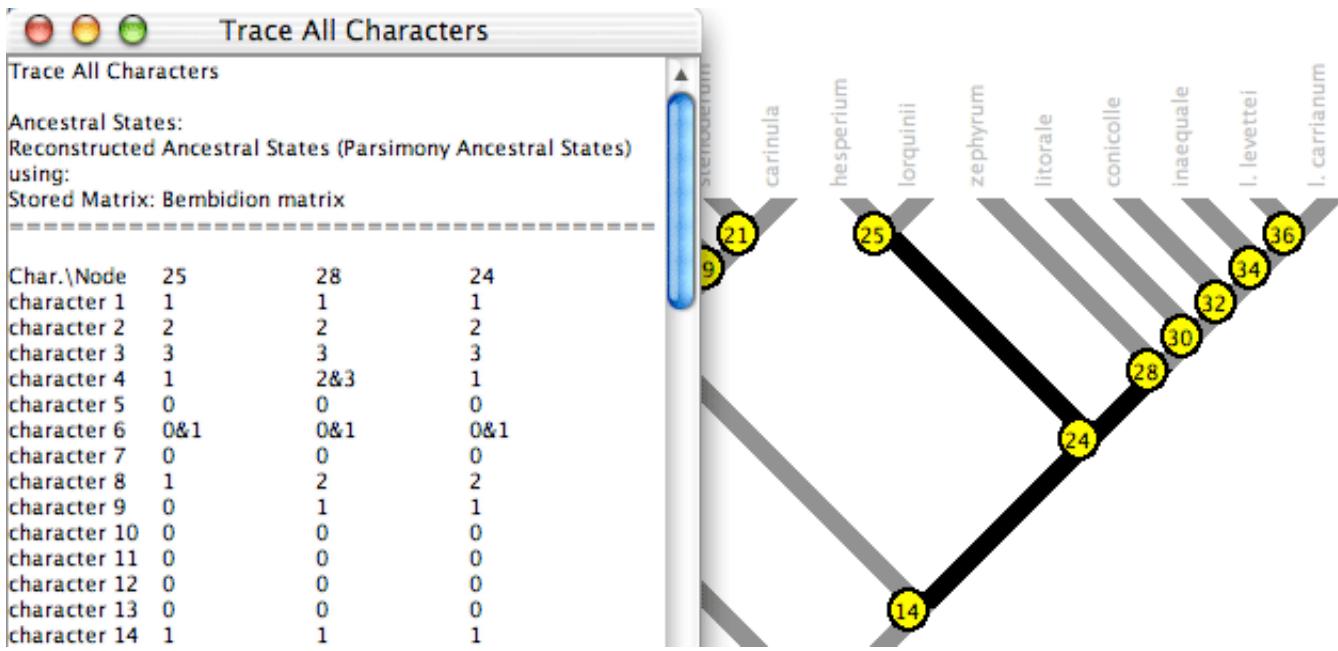
You can also see details of the reconstruction at a node by holding the cursor over the branch. A description of the reconstructed states will appear at the bottom part of the Trace Character Legend. Another method is to use the Text view of the window (touch on the Text tab at the top of the tree window) and scroll down – a text version of the trace should appear.

**Reconstruction Method:** For more details on reconstruction methods, see the sections on [parsimony](#), [likelihood](#), and [stochastic character mapping](#).

## Trace All Characters

Trace All Characters summarizes ancestral state reconstructions of many characters simultaneously. To request it, choose [Choose \(Tree Window\)Analysis>Trace All Characters](#). A text

window like that shown below will appear, listing the ancestral states reconstructed at each node for each character. Node numbers show up in red on the tree. (Alternatively, spots showing node numbers in the figure below can be turned on in the Tree Window's Drawing menu by selecting Show Node Numbers.)



By default only the selected nodes are listed. (Nodes can be selected using tools in the Tree Window.) You can request to show all nodes by turning off Show Selected Nodes Only in the Trace\_All menu. By default all characters are listed; this can be changed using the Show Selected Characters Only menu item.

The ancestral state reconstruction can be controlled in the Trace\_All menu of the tree window.

The table is either listed by characters or by nodes; you can switch from one to the other using the Rows are Characters menu item

Columns in the table in the text window may not appear perfectly aligned, but it is presented as a tab-delimited table, so you should be able to copy the text and paste it in to a text file to read in to your favorite spreadsheet program.

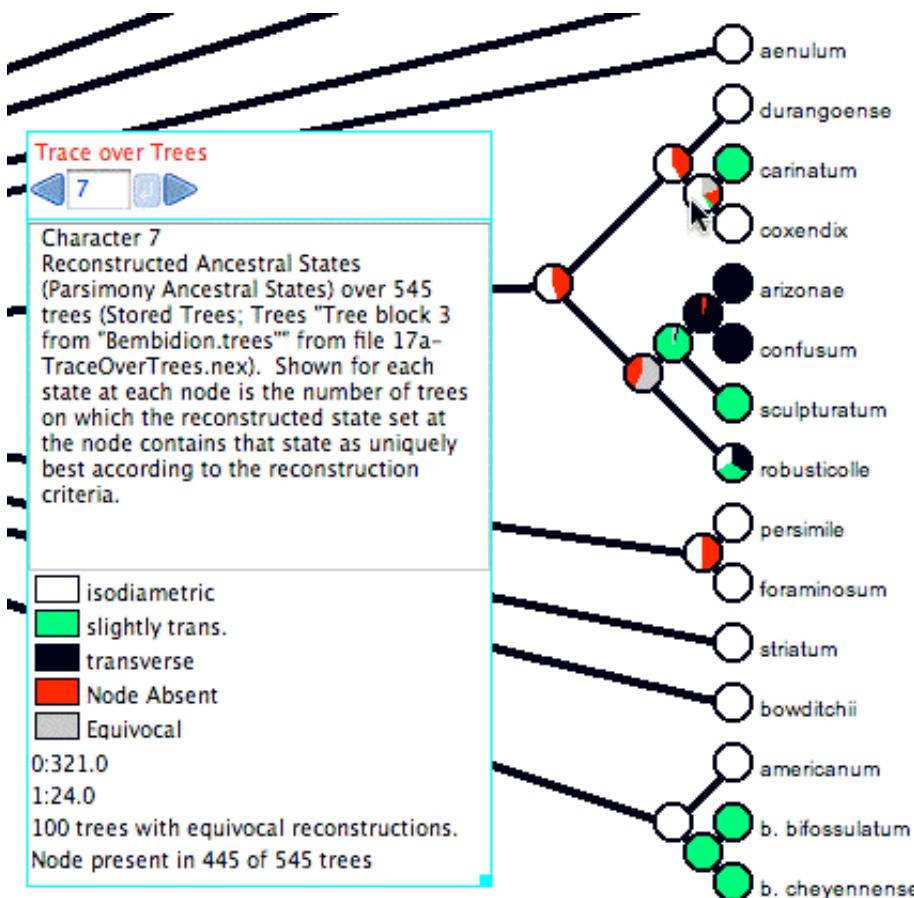
## Trace Character Over Trees

The Trace Character Over Trees facility summarizes ancestral state reconstructions over a series of trees. This is useful to understand how ancestral state reconstructions vary over a series of trees, for instance if there is uncertainty in the tree. It works for **categorical characters** only. Also, Trace Character Over Trees **does NOT calculate a consensus tree** for you. As with all other analyses in the Tree Window, it works with the tree that is given to it by the Tree Window. If you want to make your summary on a consensus tree, then you need to put the consensus tree into the Tree Window first and then request Trace Character Over

## Trees.

Choose [\(Tree Window\)Analysis>Trace Character Over Trees](#). This examines a series of trees, and for each examines a character's ancestral states on that tree. For each node in the tree in the tree window, it attempts to summarize what ancestral states are reconstructed for that same clade in the series of trees (as long as the same clade exists in the other trees). For example, imagine the tree in the tree window includes the clade Tetrapoda. Each of the series of trees is examined, and if that tree includes the clade Tetrapoda, then its reconstructed ancestral states are examined. If the tree doesn't include Tetrapoda, then it is ignored for the sake of summarizing the tetrapod ancestral states. The tree in the tree window is then decorated to summarize what ancestral states are reconstructed for each of the clades.

Here is an example of Trace Character Over Trees in action:



The cursor is over a node (the most recent common ancestor of *carinatum* and *coxendix*), and thus the legend shows a summary. The node (i.e., the clade it represents) is present in only 445 of the 545 trees examined. For this reason, 100/545 or 18.3% of the pie chart for that node is shown in red, as the node is not present in that fraction of the trees. In addition, 100 of the trees with that node have an equivocal reconstruction at that node; those trees are shown in gray in the pie chart. Of the 345 trees with the node and an unequivocal reconstruction, 321 trees have "isodiametric" reconstructed at the node (shown

in white), and in 24 trees have state "slightly trans." at that node (shown as a very thin sliver of green). In this example, "Count Trees with Uniquely Best States" is selected in the Calculate submenu of the Trace\_Over\_Trees menu. With this option a tree is counted as having a state at a node only if the state is the only optimal state. What is considered optimal depends on the reconstruction method. With parsimony, states are considered equally optimal if they are equally parsimonious. With likelihood, the Decision Threshold is used to decide whether states are good enough to be considered within the optimal set.

There are two alternatives to "Count Trees with Uniquely Best States". One is "Count All Trees with State". With this option for counting, a tree is counted as having a state at a node if the state is within the optimal set, whether or not there are other states within the optimal set. When the "Count All Trees with State" option is used, the sum of tree counts for the states at a node can more than the total number of trees with the clade, for a tree can get counted multiply at a node, under each state in an equivocal assignment. The other calculation option is "Average Frequencies Across Trees". This option is only available for reconstruction methods such as likelihood that yield a frequency or probability for each state at each node. The value presented for a node for a state is then the average frequency of that state across all of the trees possessing that node. All trees with the node are included in this calculation, even if the frequencies are very low.

An important option is what trees to examine. If the tree in the tree window is a consensus tree, then the trees examined might be the original set of most parsimonious trees that built the consensus. Trace Character Over Trees could then show how the ancestral state reconstruction varies among the most parsimonious trees. The trees examined might also be derived from a Bayesian analysis (perhaps subsampled using the Sample Trees Directly from File tree source), and the ancestral states obtained by likelihood, to do an analysis in Lutzoni's style. The trees might be random resolutions of an unresolved tree, or trees with random noise added to branch lengths, and so on. This would allow you to see how the results would vary if the tree changed. See the submenu Trace Over Trees>Tree Source for options.

## Parsimony Reconstruction Methods

Parsimony reconstruction methods find the ancestral states that minimize the number of steps of character change given the tree and observed character distribution. They can use different assumptions (models of evolution). For **categorical** characters, the **unordered** states assumption is that one step is counted for any change. The **ordered** states assumption is that the number of steps from state  $i$  to state  $j$  is  $|i-j|$ . Thus, the number of steps from state 2 to state 5 is 3 steps. A **stepmatrix** explicitly specifies the number of steps from state to state by a matrix. Mesquite does not yet do parsimony calculations for **irreversible**, **Dollo** and **character state tree** assumptions, although these models are listed in menus and you can assign them to particular characters. For **continuous** characters, the **linear cost** assumption is that the cost of a change from state  $x$  to state  $y$  is  $|x-y|$ . The **squared** change assumption is that the cost of a change from state  $x$  to state  $y$  is  $(x-y)$  squared.

Mesquite's parsimony calculations attempt to match MacClade's. Some differences remain in special cases of polymorphic terminal taxa with stepmatrices. Mesquite allows hard polytomies in the tree when stepmatrices are used.

**Assigning a parsimony model:** The parsimony model used for a character's calculations is the model assigned to it, if the character is one stored in a matrix in a file. A parsimony model can be assigned in the List of Characters window. Select the row(s) corresponding to the desired character(s), and then touch on the column heading "Parsimony Model". A drop-down menu contains a submenu that allows you to select the models to apply. You can also change the parsimony model assigned to the character being traced in Trace Character History using the Parsimony Model submenu of the Trace menu. (Recall that Mesquite cannot yet do calculations with irreversible and Dollo models.)

If the characters used in parsimony reconstruction are not stored in a matrix but rather come directly from another source of characters such as simulations, a single parsimony model can be chosen to be applied to all of the characters coming from this source. Thus, for instance, when using Trace Character History, the Parsimony Model submenu of the Trace menu can be used to assign the model to be used.

**Creating and editing stepmatrices:** To create a stepmatrix, select Characters>New Character Model>Stepmatrix. A window will appear in which you can edit the cost of i to j transitions. The number of states allowed is initially 10 (0 through 9), but you can change the number of states under (Edit Stepmatrix)>Step matrix>Set maximum state. The maximum number of states for a categorical character is 56; the maximum state value is therefore 55. This stepmatrix editor does not do triangle inequality checking (see discussion in manual of MacClade, which does check the triangle inequality).

The parsimony calculations are used also for Treelength and Character Steps.

## Likelihood Reconstruction Methods

Likelihood reconstruction methods find the ancestral states that maximize the probability the observed states would evolve under a stochastic model of evolution (Schluter et al., 1997; Pagel, 1999). The likelihood reconstruction finds, for each node, the state assignment that maximizes the probability of arriving at the observed states in the terminal taxa, given the model of evolution, and allowing the states at all other nodes to vary. (In fact, this considers all possible assignments to the other ancestral states.) This is equivalent to the marginal reconstruction of Swofford's PAUP\*, or the Fossil Likelihood reconstruction of Pagel's Discrete.

You can use likelihood for the reconstruction by selecting "Likelihood Ancestral States" when first requesting Trace Character History, or from the Method submenu of the Trace menu after Trace Character History is already active. When using Likelihood Ancestral States in Trace Character History, it is recommended that you use a Tree Form for the drawing that uses spots at the nodes (for example, (Tree Window)Drawing>Tree Form>Balls & Sticks). These spots

at the nodes will indicate relative likelihoods with pie diagrams as in Schluter et al. 1997.

At present **only categorical** characters are supported by the likelihood calculations. Models for DNA and protein evolution are not yet available for use by likelihood. Two models of evolution are currently supported, the Mk1 model and the AsymmMk model.

- **Mk1** model ("Markov k-state 1 parameter model") is a k-state generalization of the Jukes-Cantor model, and corresponds to Lewis's (2001) Mk model. The single parameter is the rate of change. Any particular change (from state 0 to 1 or state 3 to 2, for example) is equally probable. Mesquite's rate of change parameter is equivalent to the q values of Pagel's Multistate program when the q's are constrained to be equal. Thus for a character with three states 0, 1 and 2, the Mk1 model would have an instantaneous rate matrix of the following form:

To	0	1	2
From 0	-	q	q
1	q	-	q
2	q	q	-

- **AsymmMk** model ("Asymmetrical Markov k-state 2 parameter model") has two parameters: one for the rate of change from state from 0 to 1 (the "forward" rate) and one for the rate of change from 1 to 0 (the "backward" rate). Thus, this is a simple model that allows a bias in gains versus loses. As of version 1.1 of Mesquite, this model supports only binary (0,1) characters. Mesquite supports two alternative ways to describe the model. The two parameters can be forward rate and backward rate, or overall rate and bias of gains versus losses. Thus, if forward and backward rates are both 0.5, then this can alternatively be described as a rate of 0.5 and a bias of 1.0 (i.e., unbiased). The conversion between the two representations is done by the following formulas: forward rate = overall rate \* square root(bias); backward rate = overall rate / square root(bias). This conversion means that forward rate / backward rate = bias. Thus an AsymmMk model would have an instantaneous rate matrix of the following form:

To	0	1
From 0	-	f
1	b	-

As of version 1.1 of Mesquite, the AsymmMk model has two options for the handling of the root. (1) "**Root State Frequencies Equal**": With this option, the root is permitted (or required, depending on your point of view) to have expected state frequencies different from those implied by the model. In estimating the likelihood of the model, and in calculating marginal likelihoods for states at internal nodes, probabilities can be summed over all possible ancestral state reconstructions. This effectively treats the expected state frequencies of the root as equal (0.5/0.5). This is the approach of Schluter et al. (1997), Pagel (1999) and Mesquite versions 1.0 through 1.06. When rates

of gains and losses are different, this can yield perplexingly ambiguous states at the root of the tree (Schluter et al, 1997), which can be viewed positively as conservative, or negatively as a consequence of a contradiction between an implicit assumption of equal state frequencies at the root and biased equilibrium state frequencies implied by the model. (2) "**Root State Frequencies Same as Equilibrium**": With this option, the expected frequencies at the root are assumed to be consistent with the model's rates. A difference between rates of gains and losses in the model implies biased equilibrium frequencies. These implicit equilibrium frequencies are used as priors for calculating the likelihood of the model and for calculating likelihoods of ancestral states. This approach is now the default in Mesquite. It can be viewed positively as applying the model of evolution consistently throughout the tree, or negatively as imposing the assumption of a prior (albeit one derived from the data) at the root. These two options can give remarkably different reconstructions when one state is rare and the forward and backward rates are estimated from the data. The options can be chosen in the model's [editor](#).

Many programs bundle the rate of evolution into the branch lengths of the tree itself. Thus, to change the rate of evolution, the tree needs to be stretched or shrunk; there is no separate rate parameter that belongs to the stochastic model of evolution. This works well as long as the branch lengths are understood in the same way by the model and the tree, i.e., the tree's time units (calibration of time scale) are the same as that of the model. However, in Mesquite different calculations might make different assumptions about the time scale: coalescence calculations might need the tree's branches measured in generations, while a Jukes Cantor model might assume they are in expected nucleotide substitutions. Thus, many stochastic models in Mesquite have an extra parameter compared to other programs: the scaling of the model to the tree. For this reason Mk1 has a rate parameter to scale the rate against the tree.

If parameters of a model are unspecified, Mesquite currently [estimates](#) them based on the data. **Note:** Mesquite currently estimates parameters on each character separately, not on the entire data matrix. In addition Mesquite's likelihood calculations do NOT estimate branch lengths. They use pre-existing branch lengths (if a branch length is unassigned, it is treated as 1.0).

Mesquite cannot do likelihood calculations in trees with soft polytomies, or if some taxa have polymorphisms or ambiguous in the character. Missing data and gaps (inapplicable) are permitted; the calculations are then done as if taxa so coded are absent from the tree. The calculations also require that the states of a character are contiguous from zero; i.e., the character cannot have only states 0,1 and 3.

Other programs that reconstruct ancestral states using likelihood are Pagel's Discrete and Swofford's PAUP\*.

**Making, editing and applying probability models:** To use the likelihood calculations,

stochastic (probabilistic) models of evolution must be defined. Two models are predefined: a general Mk1 model and a general AsymmMk model. Both of these have their parameters unspecified.

You can also create your own models and specify their parameters by selecting Characters>New Character Model>Markov k-state 1-parameter model (to make an **Mk1** model) or Characters>New Character Model>Asymmetrical 2-param. Markov-k model (to make an **AsymmMk** model). In either case a window will appear in which you can specify the parameters. The Mk1 model allows you to change the rate. Also, you can change the maximum state allowed using a menu item in the Mk1\_model menu (e.g., to restrict it to binary characters, choose 1 as the maximum state). The AsymmMk model allows you to change the forward and backward rates, and the assumption about root state frequencies. You can also choose to express the two parameters in the AsymmMk model as a rate (which controls both forward and backward rates) and a bias (which controls the ratio of forward to backward rates). A bias of greater than 1 means forward changes are more probable; a bias of less than 1 means that backward changes are more probable.

After creating a model, you can edit it by selecting it under Characters>Edit Character Model. You can rename or delete a model by going to the List of Character Models window available under Characters.

Once models are defined they can be applied and used. When setting up a likelihood calculation, if you indicate to use "Stored Probability Model", the calculation will use the selected model for all characters. Alternatively, if the characters used are stored in a matrix (instead of generated temporarily such as by simulations), then each character can be assigned a model in advance of the calculation. This can be done by going to the List of Characters window, selecting the row(s) corresponding to the desired character(s), and then touching on the column heading "Probability Model". A drop-down menu contains a submenu that allows you to select the models to apply. These models will remain assigned to the characters if you save and reopen the file. You can also change the current probability model applied to a character by selecting a module in the Probability Model submenu of the Trace menu. Once models are assigned to the characters, then these are treated as the "Current" models applied to the characters. To indicate that the likelihood calculations use these assigned models, indicate "Current Probability Model" when asked for the source of models.

**Optimization Settings:** Models used for likelihood calculations may have adjustable settings for the optimization routines used to estimate parameter values. These settings can be changed under Characters>Model Settings; once changed the settings are universal, applying to all calculations with that category of model. They are stored in the preferences directory and used again the next time you start Mesquite. The **Mk1** Model has one setting: The coarseness of the intervals surveyed in optimizing the rate parameter. Wider intervals may result in finding the optimum more quickly when rates are high, but may make less accurate when rates are low. Mesquite by default tries the optimization twice, first with width 1.0 then again with width 10.0, and then chooses the best results. You can request that

Mesquite try a single fixed width; suggested values are 1.0 to 20.0. To request Mesquite that use its default strategy, enter a width of 0. The **AsymmMk** model has one setting, concerning the values of parameters used as starting points in the search for optimal parameter values. This setting is used when both parameters of the model are unspecified and need to be estimated. There are three options: (1) The rate is first estimated using the **Mk1** model, and then the optimization routine is given that rate plus a bias of 1.0 as starting values. (2) The forward and backward rates of 1.0 and 1.0 are used as starting values, and then 1.0 and 0.1, and then 0.1 and 1.0. Of these three attempts, the parameter values from the attempt yielding the highest likelihood is chosen. (3) Options 1 and 2 are combined, resulting in four attempts to estimate parameters, first using the **Mk1** rate then the three alternative forward and backward rates as starting values. Of the four attempts, the parameter values from the attempt yielding the highest likelihood is chosen. The default option is (1). Because the **AsymmMk** sometimes uses the **Mk1** model, changing the setting of the **Mk1** model may affect results from the **AsymmMk** model.

**Reporting of results:** Likelihood ancestral state reconstructions can be reported in various ways. A first issue is whether only the best estimates are shown at a node or instead the support for each state is shown at a node, regardless of how strong or weak. This is controlled in Trace Character History by the Display Proportional to Weights menu item. If it is selected, then support for each state is shown; otherwise, only the states judged best are shown. The judgment of what are the best states is made according to a decision threshold  $T$ , such that if the log likelihoods of two states differ by  $T$  or more, the one with lower likelihood (higher negative log likelihood) is rejected. This is set using the Likelihood Decision Threshold menu item.

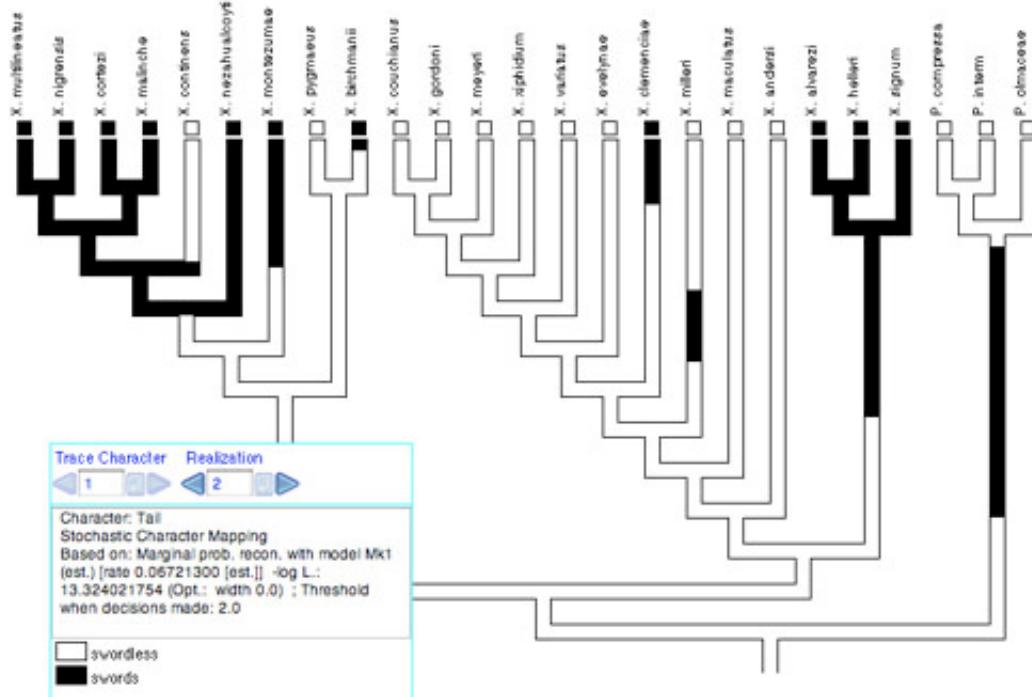
What states are judged best can be viewed using Trace Character History in several ways: (1) When the cursor is held over a branch and the list of states appears at the bottom of the Trace Legend, the states judged best according to the threshold are marked with an asterisk; (2) In the Text view of the window, the list of reconstructions shows an asterisk by each state judged best at the mode; and (3) When Display Proportional to Weights is turned off, only the best states are shaded on the branches. The threshold and best states are also used in Trace Character Over Trees.

Another issue is whether the likelihoods of alternative states are reported as is (Raw Likelihoods) or not. The other two options are to report proportional likelihoods (likelihoods of states are scaled so that they add up to 1, and thus for each state is shown its proportion of total likelihood) or negative log likelihoods. The former are convenient for interpretation and visualization; the latter may be more easily used in statistical tests.

## Stochastic character mapping

Stochastic character mapping (Neilsen, 2002; Huelsenbeck, et al. 2003) simulates realizations of precise histories of character evolution in a way consistent with likelihoods of

the model and ancestral states. By "precise" we mean that the realizations depict not just the states at nodes of the tree, but the states at all points along a branch between the nodes. Thus, with stochastic character mapping you often see changes within branches that are in principle unobservable but at the same time predicted, indeed demanded, by the rates of character change. For instance, there is stochastic character mapping used with Trace Character History:



Stochastic character mapping is chosen as a Reconstruction Method when you start Trace Character History, or from the Trace menu. It can be done currently only with non-molecular categorical data. Stochastic character mapping in Mesquite is primarily a visualization tool at present. While there are many interesting calculations that can be derived from these realizations, as done by the program SIMMAP (Bollback, 2006), none of them is yet done by Mesquite.

## Comparison and Interaction with other Programs

**NEXUS-based programs:** Mesquite currently saves the models of evolution for likelihood in the private MESQUITECHARMODELS block in NEXUS files. A private block is used because there is as yet no standard for designating such models. Thus, PAUP\*, MrBayes and other programs doing likelihood calculations will not be able to access these character models.

**Comparison with MacClade:** Users familiar with MacClade ([macclade.org](http://macclade.org)) will notice some of its features missing from Mesquite, and vice versa. MacClade is restricted to parsimony reconstructions, but has the following features that Mesquite currently lacks. MacClade's Trace Character facility has the ability to fix states at a node (the paintbrush tool) and to show individual MPR's (MPRs mode, formerly Equivocal Cycling). MacClade's Trace All Changes mode and Changes & Stasis chart summarize reconstructed changes in all characters.

Parsimony models include Dollo, irreversible and character state trees. Mesquite, on the other hand, includes likelihood reconstructions, reconstructions for continuous characters better integrated with Trace Character, branch-length sensitive calculations and other features such as Trace Over Trees.

**Pagel's Discrete and Multistate programs:** Pagel's Discrete and Multistate programs also do likelihood reconstructions of ancestral states. Discrete's Fossil Likelihoods with the Global option corresponds to Mesquite's Likelihood Ancestral States. To obtain reconstructions in Multistate equivalent to Mesquite's, define the restricted model equivalent to Mk1 or AsymmMk, using a series of "restrict" commands to set the q parameters equal as appropriate. Use the "test" command to estimate the q rates. Then, restrict the rates to those estimated. Next, use the "fossil" command to fix the node of interest to each of the states in turn, after each using the "test" command to ask Multistate to evaluate the likelihood. These likelihoods are the global likelihoods for the states at the node.

Discrete and Multistate have several features not available currently in Mesquite, including the Local option for parameter estimation, more complete reporting of statistics for the reconstructions, and calculations to test correlation among characters using likelihood ratio tests.

Mesquite can import and export data files for Discrete and Multistate (ppy files). To import, select the file with Mesquite and choose Pagel format in the import dialog box. To export, select File>Export....

## Acknowledgments

David Swofford assisted by providing code in C, translated by us to Java, for the optimization routines used in the likelihood reconstruction. Peter Midford wrote much of the module that modified the likelihood calculations to perform stochastic character mapping. Paul Lewis, David Swofford and Mark Holder helped us with useful discussions on likelihood reconstructions.

## References

Bollback, J. P. 2006 SIMMAP: Stochastic character mapping of discrete traits on phylogenies. BMC Bioinformatics. 7:88.

Huelsenbeck, J.P., R. Nielsen, and J.P. Bollback. 2003. Stochastic mapping of morphological characters. Systematic Biology 52:131-158.

Lewis, P.O. 2001. A likelihood approach to estimating phylogeny from discrete morphological character data. Systematic Biology 50:913-925.

Lutzoni F, M. Pagel & V. Reeb. 2002. Major fungal lineages are derived from lichen symbiotic ancestors. Nature 411: 937-940.

- Maddison, D.R. and W.P. Maddison. 2000. MacClade version 4: Analysis of phylogeny and character evolution. Sinauer Associates, Sunderland Massachusetts.
- Nielsen, R. 2002. Mapping mutations on phylogenies. *Systematic Biology*. 51:729-739.
- Pagel, M. 1999. The maximum likelihood approach to reconstructing ancestral character states of discrete characters on phylogenies. *Systematic Biology*. 48: 612-622.
- Pagel, M. 2000. Discrete, version 4.0. A computer program distributed by the author.
- Pagel, M. 2002. Multistate, version 0.6. A computer program distributed by the author.
- Schluter D, T. Price, A.O. Mooers, D. Ludwig. 1997. Likelihood of ancestor states in adaptive radiation. *Evolution*. 51: 1699-1711.
- Swofford, D.L. 2002. PAUP\*. Phylogenetic Analysis Using Parsimony (\*and Other Methods), Version 4.0. Sinauer Associates, Sunderland, Massachusetts.

# Processes of Character Evolution

Some characters evolve more quickly than others; some characters depend on others in their evolution. Discovering the nature of these evolutionary processes for a character from an analytical point of view involves determining a model and its parameters. For analyses that involved combined models of character change, speciation and extinction, see the page on [Diversification](#).

## Contents

- Single characters
  - [Estimating Parameters](#)
- Correlated characters
  - [Visualizations](#)
  - [Pairwise Comparisons](#)
  - [Felsenstein's contrasts](#)
  - [Pagel's Correlation Method](#)
  - [Use with Pagel's Discrete and Multistate](#)

## Estimating parameters

Maximum Likelihood estimates of rates and biases can be obtained for categorical characters for two simple models, the [Mk1](#) model and the [AsymmMk](#) model. For more information on these models, see the page on [ancestral state reconstruction](#). Mesquite cannot yet estimate parameters for models of DNA sequence evolution.

Three modules provide calculations to estimate parameters for the Mk1 and AsymmMk models:

- **Mk1 Estimated Rate**— Estimates the rate of a character's evolution under the simple Mk1 model.
- **Forward/Backward Rates** — Uses maximum likelihood to estimate the rates of forward and backward changes (0 to 1 and 1 to 0 changes respectively), or alternatively the overall rate and the bias in gains versus losses, using the AsymmMk model on a tree for a given character.
- **Asymmetry Likelihood Ratio Test** — Calculates the test statistic for the likelihood ratio test comparing the asymmetrical and one parameter models  $[2\ln(L(\text{Asymm.})/L(\text{Mk1}))]$ , on a tree for a given character.

These calculations consider a categorical character and a tree. As such, they can be considered to be values describing a character (and thus are available when analyzing characters, as for instance in a Characters Bar Chart or Scattergram or a List of Characters Window) or values describing a tree (and thus are available when analyzing trees, as for

instance in a Trees Bar Chart or Scattergram or a List of Trees Window). To access them as values for characters, select "Character value with current tree" or "Character value with tree". To access them as values for trees, select "Tree value using character". Two example files illustrate parameter estimation, Mesquite\_Folder/examples/Ancestral\_States/15a-estimatingParameters.nex and Ancestral\_States/15b-estimatingParameters.nex

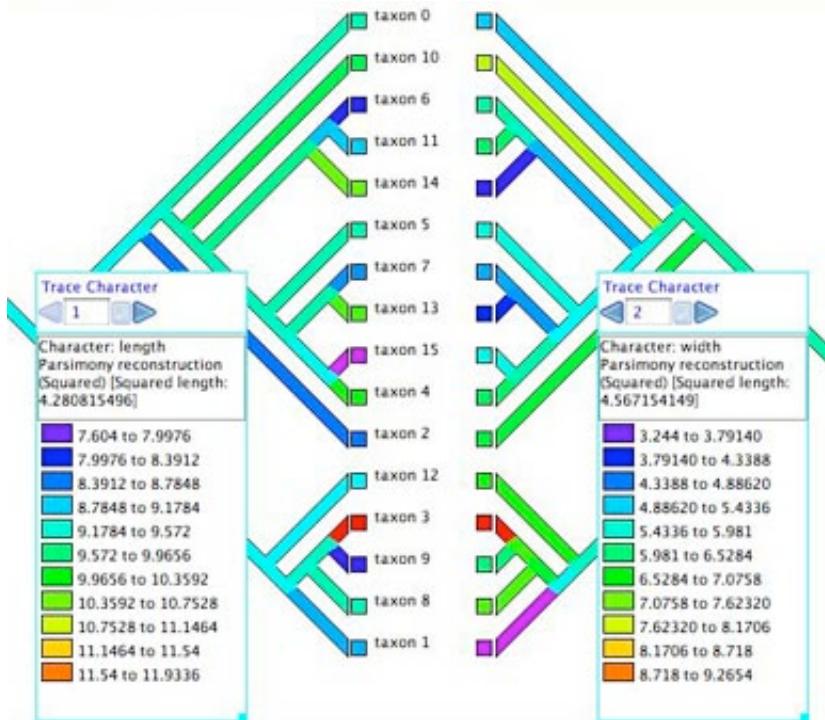
You can explore the **likelihood surface** of the two parameters AsymmMk model using the (Tree Window)Analysis>Other Analyses>Likelihood Surface AsymMk menu item.

## Correlations: Visualizations

To study correlations or associations among characters, there are both correlation calculators (see [Pairwise comparisons](#), [Felsenstein's contrasts](#), and [Pagel's correlation method](#), below) and heuristic visualizations. The latter include:

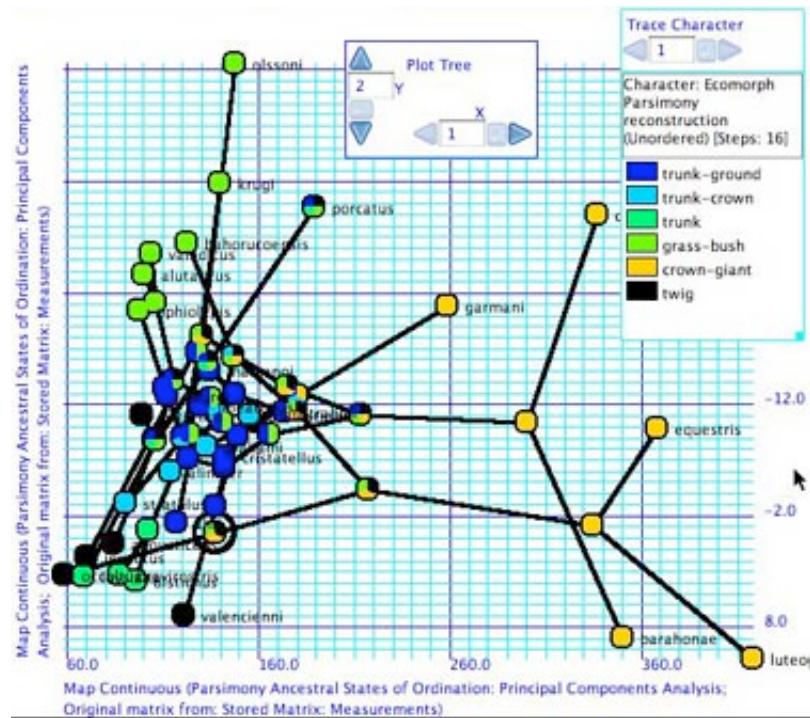
- **Mirror Tree Window** – When a Tree Window is open, you can request and alternative view of the same tree by selecting [Tree>Mirror Tree Window](#). This shows the same tree as in the tree window, shown in duplicate tips-to-tips. The purpose of this is to allow you to display two different visualizations (one at left, one at right) and compare them. Character correlations can be explored by tracing evolution of two characters, as shown here.

Example files: Basic\_Examples/tree\_viewing/08-mirrorTree.nex; Ancestral\_States/15-Mk1AsymmCompare.nex; Pairwise\_Comparison/01-pairwise.nex.

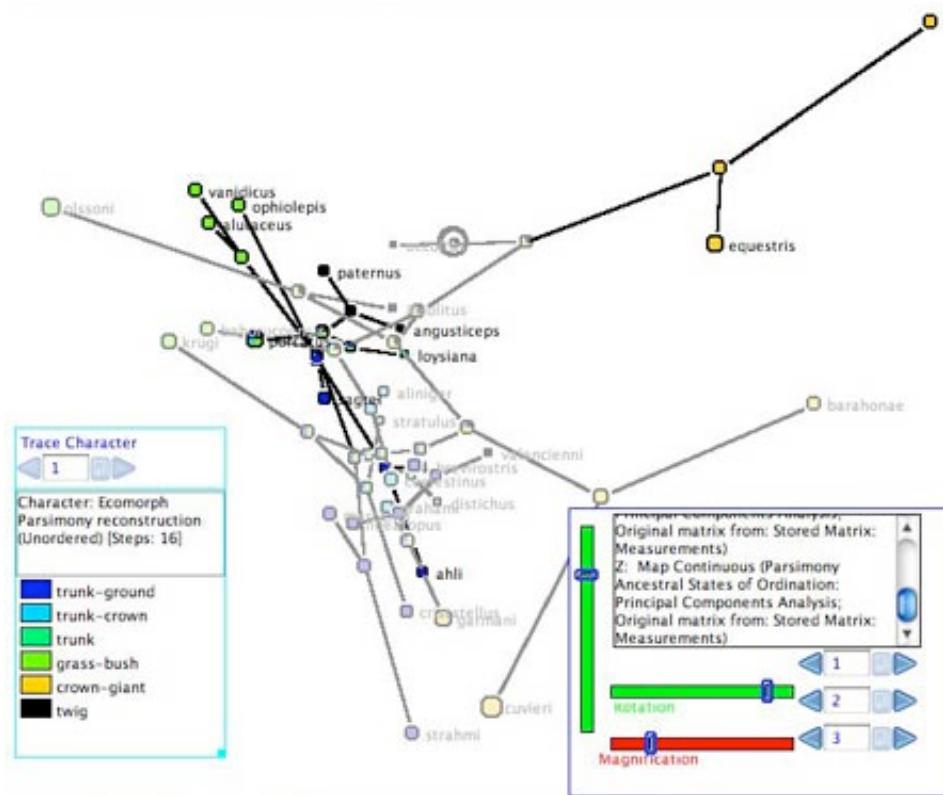


- **Plot Tree 2D** – Plots the tree in a 2-dimensional space, available as a tree drawing form in the [Drawing>Tree Form](#) submenu. If the axes represent the state of the taxa in two continuous characters, then this allows one to map the tree into the character

space, which may suggest patterns or correlations. The internal nodes of the tree can be placed at the reconstructed ancestral states. An example is shown here.  
Example file: Multivariate\_Continuous/07-anoles.nex



- **Plot Tree 3D** – Plots the tree in a 3-dimensional space, available as a tree drawing form in the Drawing>Tree Form submenu. This is part of the Rhetenor package. As with Plot Tree 2d, this allows one to map the tree into the character space. The tree can be rotated in space using the Rotation sliders in the legend. An example is shown here.  
Example file: Multivariate\_Continuous/08-anoles.nex



- **Taxa Scattergram** – Select Analysis>New Scattergram For>Taxa to obtain a bivariate plot for taxa. You will be asked whether to use the same or different calculations for the two axes. By "Different" is meant two entirely different calculations, such as the percentage of missing data in the taxon on one axis, and the state of a continuous variable on the other. Choose "Same" and then, if asked, indicate you want "Continuous state of taxon". You will therefore be plotting the taxa according to their states in one character versus another. If PDAP is installed, you will be able to do linear regression by selecting Scatogram>Analysis>Other Choices..., and choosing one of the Scattergram Diagnostics. Note: any correlation seen is aphylogenetic. Phylogenetic correlations can be studied by using the Felsenstein's contrasts calculations in PDAP.  
Example files: Multivariate\_Continuous/01-wingsPlot.nex and subsequent

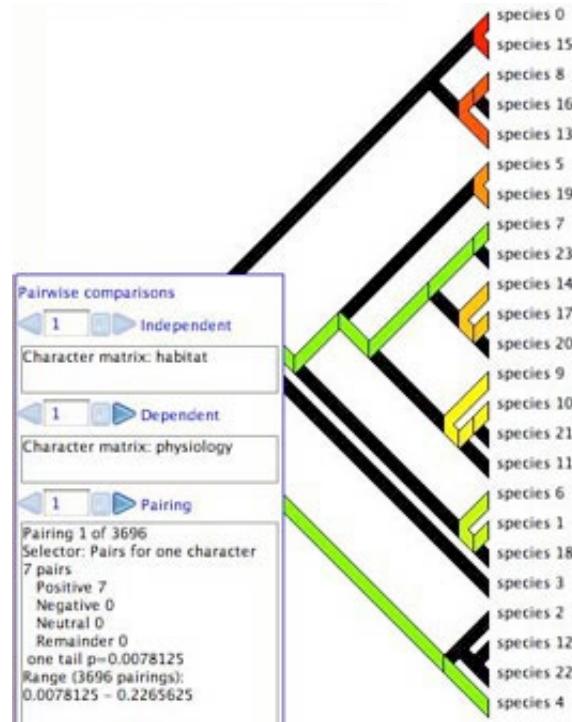
## Correlations: Pairwise comparisons

Character correlations can be tested using pairwise comparisons as described by Read & Nee (1995) and W. Maddison (2000). This is available under the Analysis menu of Tree Windows. The module chooses pairs of taxa, and indicates how the pairs compare in two characters: does the member of the pair with the higher value (say, state 1) in one character have higher or lower value in a second character? A summary over all pairs is given in the legend, as shown below. There are three options for choosing pairs:

- Most pairs – choose pairs to maximize number of pairs, regardless of the states in the characters
- Pairs for one character – choose pairs of taxa that differ in the state of the first

character (independent variable)

- Pairs for two characters – choose pairs of taxa that differ in the state of both characters

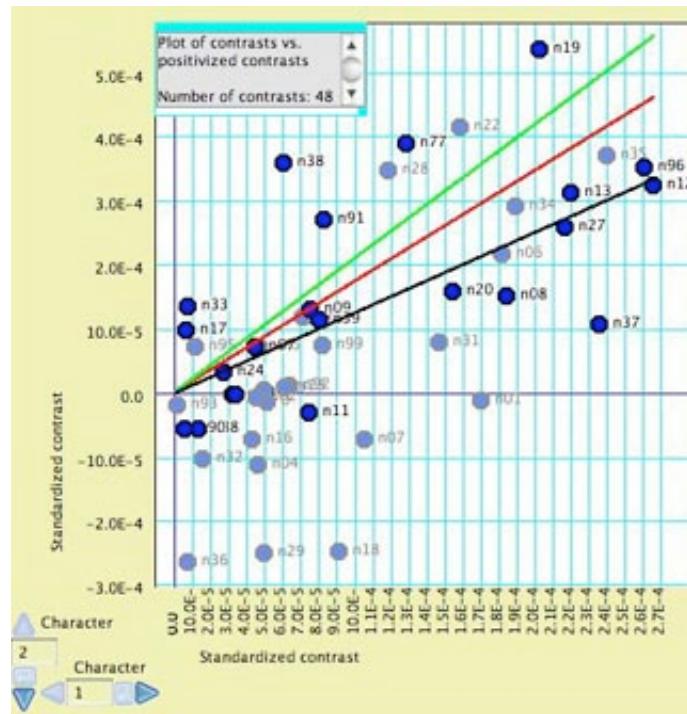


The graphical display shows the current pairing chosen; you can scroll through all pairings using the legend.

Example files: Pairwise\_Comparison/01-pairwise.nex and subsequent

## Felsenstein's Independent Contrasts

Correlations among continuous valued characters can be studied using the separately-available [PDAP](#) package (Midford et al., 2003), which (among other things) calculates Felsenstein's (1985) independent contrasts and displays them in a scatterplot:

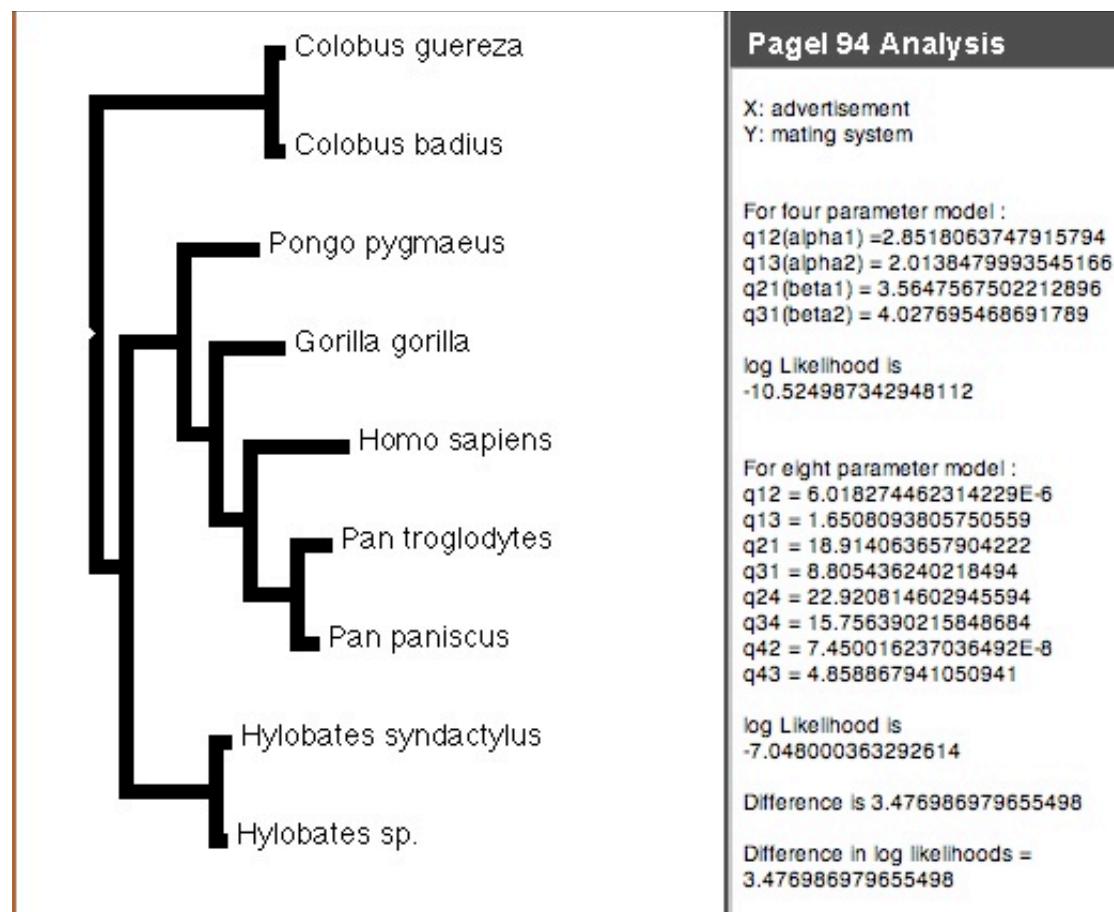


The points in the plot are nodes in the tree, with the X and Y axes representing the independent contrast across the node in each of the two characters. Regression lines, confidence intervals and other statistics can be calculated by PDAP. When only some nodes in the tree are selected, they are highlighted in the plot as shown above.

The PDAP documentation or example files should be consulted for more details.

## Pagel's Correlation Method

Pagel's (1994) Correlation test is available via the `correl` package (Midford & Maddison) distributed with Mesquite. It uses likelihood to test whether the evolution of two binary (0,1) characters is independent. To begin an analysis, choose Correlation Analysis from the Analysis menu of a tree window. If there is more than one option for method, choose Pagel 94 (if there is only one option, you won't be asked). You will be asked how intense to make the likelihood search (number of extra iterations), whether to present a p-value, and how many simulations to use to estimate the p-value. Options are explained in more detail on the [page](#) focusing on the analysis. A panel showing the analysis will appear in the tree window:



If the difference in likelihoods between the four (independent) and eight (correlated) models is high enough, then we can reject the null hypothesis of independence. A separate page of the manual describes the [Pagel94 modules](#) in more detail.

## Use with Pagel's Discrete and Multistate

Mesquite can import and export files for use by the Discrete (Pagel, 2000) and Multistate (Pagel, 2002) programs. For import, attempt to read the Pagel format file in Mesquite, and choose the file format from the import dialog box. For export, select the Export... menu item from the File menu.

## References

- Felsenstein, J. 1985. Phylogenies and the comparative method. *American Naturalist*, 125:1-15.
- Maddison, W.P. 2000. Testing character correlation using pairwise comparisons on a phylogeny. *J. Theoretical Biology*. 202: 195-204.
- Midford, P. E., T. Garland Jr. & W. Maddison. 2002. PDAP:PDTREE package for Mesquite, version 1.00.

- Pagel, M. 1994. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. Proc. R. Soc. London B 255: 37-45.
- Pagel, M. 2000. Discrete, version 4.0. A computer program distributed by the author.
- Pagel, M. 2002. Multistate, version 0.6. A computer program distributed by the author.
- Read, A. F. and S. Nee. 1995. Inference from binary comparative data. J. Theoretical Biology 173:99-108

# Pagel's 1994 Correlation method

Peter Midford & Wayne Maddison

The Pagel94 module implements a test of independent evolution of two binary characters. This test was first described in Pagel (1994). Although this test has been available in Pagel's own program Discrete, this module is not a simple translation of Discrete. In particular, although the module calculates the same likelihood value for a given combination of tree and character states, the maximum likelihood search and character simulation portions of the module utilize existing Mesquite code. The remainder of this documentation consists of an overview of the theory of the test, followed by detailed discussion of the methods used in the three phases of the test.

## Contents

- [Theory](#)
- [The evaluator](#)
- [The maximum likelihood estimation process](#)
- [Estimating statistical significance](#)
- [Options for Users](#)
- [References](#)

## Theory

Pagel (1994) describes a test of independent evolution of two binary characters. The test compares the ratio of likelihoods of two models: one model where the rates of change in each character are independent of the state of the other and a second model where rates of change depend on the state of the other character. The significance of the ratio is tested by comparison with simulated data, and Pagel warns against using a standard chi-square-based test in most cases.

The models are expressed as matrices of instantaneous transition rates, with each row and column specifying a pair of character states. The model of dependent change specifies 8 parameters and can be specified by the following matrix:

From/To	0,0	0,1	1,0	1,1
0,0	$-(q_{12}+q_{13})$	$q_{12}$	$q_{13}$	0
0,1	$q_{21}$	$-(q_{21}+q_{24})$	0	$q_{24}$
1,0	$q_{31}$	0	$-(q_{31}+q_{34})$	$q_{34}$
1,1	0	$q_{42}$	$q_{43}$	$-(q_{42}+q_{43})$

The zero values on the antidiagonal prevent simultaneous changes of both characters and the entries on the main diagonal normalize the matrix.

The model of independent change describes the case where the transition rates of each variable are independent of the state of the other. Pagel 1994 describes the independent model as a pair of 2x2 matrices, one for each variable. The forward ( $0 \rightarrow 1$ ) and backward ( $1 \rightarrow 0$ ) change rates of each variable are referred to as alpha and beta respectively. Another approach, taken in this module is to treat the independent case as a degenerate form of the dependent model. The model of independent change specifies 4 parameters, which can be obtained from the dependent model by setting the pairs of parameters across the antidiagonal ( $q_{13}$  and  $q_{24}$ ,  $q_{12}$  and  $q_{34}$ ,  $q_{21}$  and  $q_{43}$ ,  $q_{31}$  and  $q_{42}$ ) equal.

From/To	0,0	0,1	1,0	1,1
0,0	$-(q_{12}+q_{13})$	$q_{12}(\text{alpha1})$	$q_{13}(\text{alpha2})$	0
0,1	$q_{21}(\text{beta1})$	$-(q_{21}+q_{13})$	0	$q_{13}(\text{alpha2})$
1,0	$q_{31}(\text{beta2})$	0	$-(q_{31}+q_{12})$	$q_{12}(\text{alpha1})$
1,1	0	$q_{31}(\text{beta2})$	$q_{21}(\text{beta1})$	$-(q_{31}+q_{21})$

Our treatment of the independent (four-parameter) model is different from Pagel's; we treat it as a version of the eight-parameter model constrained by setting four pairs of parameters equal ( $q_{13}=q_{24}$ ,  $q_{12}=q_{34}$ ,  $q_{21}=q_{43}$ , and  $q_{31}=q_{42}$ ). Our four-parameter model specifies that the antidiagonal elements, indicating rates of simultaneous change, are also zero, in accord with the eight-parameter model used by us and Pagel. In contrast, Pagel (1994) treats his four-parameter model differently, as a combination of two separate asymmetric models, each represented by a 2x2 matrix. Thus, Pagel argues (p. 41) that his four-parameter model is not a constrained version of the eight-parameter model.

We feel that our formulation of the four-parameter model is a valid alternative for three reasons. First, it uses the same assumption (zero rates of simultaneous change) as used in the eight-parameter model. Second, in our explorations it appeared to arrive at the same likelihoods as obtained by Pagel's two-asymmetric-model approach. We have tested the assumption of equivalence numerous times with both this module and Discrete. Even at high transition rates, we could find no differences between the two approaches that were not more appropriately explained as stochastic behavior in the likelihood searches of the respective programs. This stochastic behavior can be observed by simply re-running an analysis multiple times and looking at both the parameter estimates and (especially for the 8-parameter model) the likelihood values.

Third, setting the anti-diagonals to zero in the four parameter model appears justified on first principles. As Pagel points out (p. 40), setting simultaneous rates to zero does not preclude the possibility of both characters changing state over a longer period of time. However, because these are instantaneous rates, the diagonal indicates the rate of events where both characters change over the same infinitesimal time  $dt$ . No matter how large the rate of character change is, the amount of time spent in transitions is still zero, thus the chance of two independent characters changing at the same time is also zero.

To calculate the likelihood of a specific set of rates giving rise to the observed data at the tips of the particular tree, the rates must be converted to transition probabilities for each

branch of the tree. Because the model is based on the assumption of a Markov process, the transition probabilities for a particular branch are dependent only on the rate matrix and the length of the branch. The conversion entails calculating  $\exp(lA)$  where A is the rate matrix and l is the branch length.

The calculation itself starts from the tips of the tree and works down each branch toward the root. At each branch the calculated likelihoods at the upper end are (effectively) multiplied by a transpose of the transition matrix to obtain the set of likelihoods at the lower end. The likelihoods at the tips are just the observed data.

There are several possible ways to combine likelihood calculations at the root. Currently the module simply adds the likelihoods for each of the four possible root states to get the overall likelihood for the model. Other combination options, including weighting root states empirically from their frequencies at the tips or by specifying an explicit prior distribution may be made available in future releases.

## The Evaluator

The evaluator calculates the likelihood of a specific set of rates giving rise to the observed data at the tips of the particular tree. This release uses two methods to calculate the exponentiation of the rate matrix. The first method, which is faster, uses spectral decomposition (e.g., eigenvalues and eigenvectors). However, this method does not work correctly when the matrix has complex eigenvalues. In this case, or any other where the spectral decomposition method produces a transition probability matrix with negative entries, the evaluator redoing the calculation using a method first described by Putzer (1966). This method is slower, but appears to handle matrices with complex eigenvalues correctly.

The current release does not allow the user to directly specify a set of parameter values and get a likelihood value. This facility may be added to future releases.

## The Maximum likelihood estimation process

To estimate maximum likelihood values for the model parameters, the Pagel94 module requires an optimizer that works with multiple variables and does not require derivatives of the function being optimized. To do this, the module calls on the widely used Brent(1973) multivariate optimizer. This optimizer has the important property of not requiring derivatives of the function being maximized. To improve the chances of finding a global minimum, the module adds two methods to the optimization process. First the tree is scaled to a height of one, which reduces the chance of overflow. Second the module chooses the best result from multiple optimizer runs starting from different points. For optimizing the independent (four parameter) model, the module runs the optimizer from three start points (10, 10, 10, 10), (0.01, 10, 0.01, 10), and (10, 0.01, 10, 0.01).

Because dependent (eight parameter) models generally have more rugged likelihood

surfaces, we found that additional searching sometimes produced substantially better answers. One example of this is the great ape data set discussed in Pagel94 and (in expanded form) in the Discrete manual. To choose multiple starting points, the model starts by estimating parameters for two six-parameter models, one where the rates of the first variable is dependent on the second but rates of the second are independent of the first (four rates for the first variable + two rates for the second), and the other with the dependencies reversed. Each six parameter model is optimized starting from the (saved) optimum of the four parameter model, with the additional parameters started from initial values that correspond to the four parameter model. The search for the eight-parameter model uses **multiple start points** derived from the six-parameter model results. The first two searches start from the values from each of the six-parameter models. Additional searches, controlled by the Likelihood iterations command, start from randomized linear combinations of the values from the six-parameter models. Specifically, the first four parameters for the eight-parameter search start from points on the line lying between the corresponding parameters for the six-parameter models. Because the random element is Gaussian distributed, the actual start point may lay outside the portion bracketed by the two values. The remaining four values are set from a corresponding parameter drawn from one of the six-parameter models with Gaussian noise added.

Note that the value of additional iteration depends both on the particular tree and data and on the goal of the analysis. If finding the best values for the parameters is important, more iteration will do a more thorough search of parameter space. If the goal is simply to perform a test of independence, less iteration and more simulation is appropriate.

Because Discrete uses a different optimizer than this module, the likelihoods and parameters found will not be identical in many cases. For most applications the differences will not be important. In other cases, the additional search this module provides should provide more optimal results.

## Estimating Statistical Significance

Pagel (1994) strongly recommends testing the observed likelihood ratios by running Monte Carlo tests using simulated data. Specifically, for the test of correlated evolution described by Pagel (1994), the four-parameter model is not exactly a special case of the eight-parameter model because dual transitions in the rate matrix (the antidiagonal elements) have been set to zero. Therefore, Pagel (1994) does not consider a likelihood ratio test based on a chi-square distribution with four degrees of freedom to be appropriate. Although a chi-square test might be better justified in Mesquite's implementation because we use a different four parameter model, it is unclear how well the chi-square will approximate the correct distribution at our sample sizes. Thus, we recommend a simulation as does Pagel.

As discussed above, for estimating statistical significance, it is better to trade off optimizer iterations for simulation sets. Note that if the optimizer appears to hang (e.g., neither progress indicator appears to be updating), it is possible to abort the simulation. In this case, the module will calculate Monte Carlo statistics based on however many simulations it

did complete, which might be usable in some cases. We recommend for published results that significance be estimated using 1000 or more simulation replicates.

## Options for Users

There are three options for users to choose. When first requested, a dialog box will ask the user for these; once the analysis has been done, the options can be changed via the Pagel94 menu.

- **Extra iterations:** as noted [above](#), the estimation of likelihood of the 8 parameter model can be requested to use multiple start points. The more start points specified, the better the estimation of the likelihood should be.
- Whether to present a **P value** or not. The P values are calculated by simulation.
- The **number of replicates** done in the simulation to estimate statistical significance. To estimate whether  $P < 0.05$ , well over 100 simulations are recommended.

## References

Brent, R. P. 1973. Algorithms for Optimization without Derivatives. Englewood Cliffs, New Jersey. Prentice Hall.

Pagel, M. 1994. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. Proc. R. Soc. London B 255: 37-45.

Putzer, E. J. 1966. Avoiding the Jordan canonical form in the discussion of linear systems with constant coefficients. American Mathematical Monthly 73:2-7.

---

# Diversification: Speciation and extinction

The diversification processes that determine the branching pattern of the phylogenetic tree, speciation and extinction, can be simulated and analyzed. There is a series of [example files](#) available showing some of these features.

- [Simulating speciation and extinction](#) to generate trees
- [Lineages Through Time](#) plots
- [Analysis of Speciation/Extinction rates](#)
- [Analysis of Character-associated Speciation/Extinction rates](#)
  - [Likelihood Calculations](#)
  - [Hypothesis Testing](#)
  - [Multiple trees](#)
- [References](#)

## Simulations

Trees can be created through simulations of speciation and extinction. Here are options:

- Tree simulators available to make tree blocks ([Taxa&Trees>Make New Trees Block from>Simulated Trees](#)) or to generate trees on demand (e.g., for charts or in the tree window).
  - **Uniform speciation (Yule)** (trees package) - Generates tree by a pure birth process (a Yule process). The chance of speciation is equal for all tips. Options: total time depth of tree.
  - **Uniform speciation with sampling** (TreeFarm package) – Generates a tree by Yule process, as above, but with a total number of species greater than in the taxa block. Extra species are then randomly sampled out, to leave the tree with the appropriate number of species. Barraclough & Nee (2001) discuss how this sampling alters the branch length distribution of the tree.
  - **Birth/Death process trees** (Diverse package) – Generates trees with a speciation rate and extinction rate.
- **Trees & Diversification Characters** (Diverse package) – Simulates speciation and extinction, simultaneous with the evolution of characters that influence speciation and extinction rates. Select the Trees & Diversification Characters item in the Taxa&Trees menu to generate the tree block and a character matrix; the i'th tree was evolved in connection with the i'th character in the matrix.
  - **BiSSE Trees & Characters** – by this model, a categorical character evolves with two states, which can cause their lineages to differ in their speciation rates, and in their extinction rates. You will be given the following options for setting up the simulation:

Rate 0 → 1:	0.0050
Rate 1 → 0:	0.0050
Rate of Speciation with 0:	0.1
Rate of Speciation with 1 :	0.1
Rate of Extinction with 0:	0.05
Rate of Extinction with 1:	0.05
Desired Tree Size:	40
Continue to Tree Size:	48
<b>OK</b>	

The rates of character change, speciation and extinction represent the 6 parameters of the BiSSE model. The **Desired Tree Size** is the number of terminal taxa in the trees you want to produce. By default this is the current number of taxa in the data file, but you could choose a number either higher or lower. The "Continue to Tree Size" option is explained as follows. The algorithm starts speciation and extinction and eventually the desired tree size is reached. However, it doesn't stop there, because if the simulation were continued, that desired tree size might be achieved several times (fluctuating as speciation and extinction push the number of species up and down). The simulation continues until the size indicated in "Continue to Tree Size", which should be larger than the desired size. The simulation goes to that tree size and then looks back at all of the intermediate steps in which a tree of the desired number of taxa was achieved. It then selects one of these trees randomly. (NOTE: the simulation used in this module differs from that used by Maddison, Midford and Otto, 2007, who used a prerelease version of the module. Their simulation proceeded in small time slices, and stopped as soon as the desired tree size was reached. The released version samples exponential wait times to jump to the next event, and proceeds beyond the desired tree size as described above.)

- **Evolving Speciation Rate (continuous character)** – This is a speciation-only model. A continuous character evolves by Brownian motion; this character is the speciation rate. There is no extinction.

## Lineages through Time

You can display a lineages through time plot of the current tree in the tree window by selecting the (Tree Window)Analysis>Lineages through Time menu item.

## Analysis: Speciation/Extinction rates (Independent of character)

This analysis assumes that speciation and extinction rates are constant (they do not depend on any character). The calculation is available in the Analysis menu of a tree window, under **Diversification (Char. Indep.)**. The calculation is also available anywhere that a value for a tree can be calculated, for instance in the Tree Legend of the tree window (under Analysis), in the List of Trees (Columns>Number For Tree>Other Choices), and in a Trees chart.

- **Speciation/Extinction Likelihood** - This calculates the likelihood of a simple birth/death model with a speciation rate and an extinction rate, using a reduced form of the BiSSE calculations of Maddison, Midford & Otto 2007. If the rates are not indicated (i.e. are indicated as ?), then they are estimated.

## Analysis: A binary character's effect on diversification

These analyses explore whether there is an association between the states of a characters, and the rates of speciation and extinction. These analyseis are available in the Analysis menu of a tree window, under **Character-Associated Diversification**. They are also anywhere that a value for a tree can be calculated, for instance in the tree legend of the tree window (Analysis>Tree Legend>Tree Value Using Character, Show secondary choices), in the List of Trees (Columns>Number for Tree>Tree Value Using Character, Show secondary choices), and in charts of trees. They are also available in the List of Characters window (Columns>Number For Character>Character Value with Current Tree, Show secondary choices) and in charts of characters.

- **BiSSE Speciation/Extinction Likelihood** (Maddison, Midford & Otto, 2007) – This calculates likelihoods and estimates parameters of a 6 parameter model of speciation, extinction and character change. The parameters are speciation rate under state 0 ( $\lambda_0$ ), speceation rate under state 1 ( $\lambda_1$ ), extinction rate under state 0 ( $\mu_0$ ), extinction rate under state 1 ( $\mu_1$ ), rate of 0 to 1 character change ( $q_{01}$ ), and rate of 1 to 0 character change ( $q_{10}$ ). In the parameters dialog box you can set the parameter values (or leave them "?", to be estimated), and you can constrain some to be equal to others.
- **BiSSE Net Diversification Likelihood** – This is basically the same calculation as the BiSSE speciation/extinction likelihood, but is reparameterized as net diversification rate under state 0 ( $r_0$ , which is  $\lambda_0 - \mu_0$ ), under state 1 ( $r_1$ ), speciation/extinction ratio under state 0 ( $a_0$ , which is  $\lambda_0/\mu_0$ ), under state 1 ( $a_1$ ), and the two character state change rates. It is reparameterized to permit you to set constraints of equality that would otherwise be unavailable.
- **Sister Diversification** – This uses the simple sign test on independent sister clade pairs in the style of Mitter, Farrell and Wiegmann 1988. The analysis automatically finds sister clade pairs that contrast in the character state of the binary character. Each clade must be completely uniform in state. The number of pairs favoring state 1 (i.e. the clade with state 1 is larger) and favoring state 0 are counted and used as the basis for the sign test.

## The BiSSE likelihood calculations

The Diverse package's BiSSE calculations are based on the equations of Maddison, Midford & Otto (2007). The maximum likelihood calculations are performed using the optimizer of Brent (1973). To improve the performance of the optimizer, the BiSSE module invokes the optimizer from 10 randomly selected starting points. During these runs, the numerical integrator (see below) is set to run with a relatively large starting step size equal to 1/100 the length of an average branch. This length is value is allowed to vary somewhat (a range of 8:1) with the length of the particular branch, so that extremely short branches and long branches are not undercounted or overcounted respectively. The final likelihood value is then calculated using the starting point with the best likelihood, but a smaller step size (1/1000) the average branch length.

The BiSSE method requires numerical solution of a system of four differential equations along each branch of the tree. To do this, Diverse uses a RKF45 (Runge-Kutta-Fehlberg fourth order method with fifth order step-size control) algorithm to integrate along the branches. The step-size control allows the step size to vary from the starting value based on the correction calculation. Because BiSSE equations are relatively well behaved, using a higher order correction mostly serves to speed the calculation by allowing the method to grow the step size. Note that this method is a refinement of the RK4 algorithm used in Maddison, Midford & Otto (2007).

## Hypothesis Testing

To test hypotheses about the association between a character and diversification rates, you can use either the Sister Diversification calculation, which directly gives a P value, or the BiSSE Ln Likelihood Difference calculation, which can be used to get a p value. The former is an old but fairly non-parametric calculation; the latter is parametric but uses more of the information in the tree.

The BiSSE Ln Likelihood Difference calculation is used for likelihood ratio tests. It performing the calculations in constrained and unconstrained ways and calculating the difference in likelihood. You can access these calculations as a choice when you do the Character-Associated Diversification analysis in the Analysis menu of the Tree window, or in other places (such as charts) where you are calculating a value for a tree and character. You will be asked first to specify the parameters of the unconstrained model, then the parameters of the constrained model.

The likelihood difference can be used to answer whether, for instance, two parameter values are the same. Thus, the constrained model would set them equal (using the "Constrain =" choices in the dialog), while the unconstrained model would not include this constraint. When there is a difference of a single degree of freedom in the constrained and unconstrained models (i.e. comparing a five parameter and a 6 parameter model) then two times the likelihood difference appears to be approximately distributed as a chi square

(Maddison, Midford & Otto 2007), so that you can use that distribution for significance values.

It is more accurate, but more time consuming, to simulate under the null model to derive your own distribution for statistical testing. You can do this as follows.

1. With your data, estimate the values of the BiSSE parameters in the **constrained** case (e.g.,  $\lambda_0 = \lambda_1$ ).
2. Simulate trees and characters using Trees & Diversification Characters in the Taxa&Trees menu, choosing the BiSSE Trees & Characters as the simulator. Enter the parameters as you had estimated them from your data. Choose as many taxa as in your data. You might want to simulate a small number of trees, e.g. 5, at first, just to practice with these calculations. (Later increase the sample size to a large number, e.g. 1000, to derive a good distribution.)
3. Save the file (just in case).
4. Choose New Bar & Line Chart for... Characters. You will be asked a series of question. Here are the responses:
  - Source of Characters: Stored characters. If given a choice, make sure you choose the matrix you just simulated with BiSSE Trees & Characters.
  - Value (under secondary choices): Character value with respective tree
  - Source of trees: Stored trees. If given a choice, make sure you choose the trees you just simulated with BiSSE Tree & Characters.
  - Value: BiSSE Ln likelihood difference
  - BiSSE Parameters: Leave the parameters unspecified, but set the constraints to be as in your **UNCONSTRAINED** model.
  - BiSSE Parameters: Leave the parameters unspecified, but set the constraints to be as in your **CONSTRAINED** model.
5. This calculation may take a long time. If asked, indicate you want to save a table of results. Also, print the chart and save it as text once it's done. You might not want to redo it (as it takes a while).

## Multiple Trees

You can examine likelihoods, or other single parameters from the BiSSE modules, over multiple trees by asking for a tree chart and indicating you want a Tree Value using Character. You will be asked if you want to save the results as a table to a text file during the calculations; this text file will contain all of the parameter values as well, as a tab-delimited table, so that you can import it into a spreadsheet program.

You can also examine multiple trees by setting up the tree window and selecting Character-Associated Diversification in the Analysis menu to begin a BiSSE calculation. Then, you can paste the following script into the Send Script dialog box (available in the Tree window's Window menu, Scripting submenu). This script will go from tree to tree, saving results as it goes.

```
String.resultsFile 'results.txt';
saveMessageToFile *String.resultsFile 'RESULTS with different trees';
appendReturnToFile *String.resultsFile;
getWindow;
tell It;
    getNumTrees;
    Integer.numReps *It;
    ifNotCombinable *Integer.numReps;
        Integer.numReps 10; [in case indefinite number of trees]
    endIf;
endTell;
debug;
Integer.count 0;
for *Integer.numReps;
    increment.count;
    getWindow;
    tell It;
        setTreeNumber *Integer.count;
    endTell;
    getEmployee #mesquite.diverse.CharAssocDivViewer.CharAssocDivViewer;
    tell It;
        doCounts; [because counts dont auto-update when scripting]
    endTell;
    getEmployee #mesquite.diverse.BISSELikelihood.BISSELikelihood;
    tell It;
        getLastResult;
        String.result *It;
        appendMessageToFile *String.resultsFile *String.result;
        appendReturnToFile *String.resultsFile;
    endTell;
endFor;
```

## References

Brent, R. P. 1973. Algorithms for Optimization without Derivatives. Englewood Cliffs, New Jersey. Prentice Hall.

Maddison, W.P., P.E. Midford & S.P. Otto. 2007. Estimating a binary character's effect on speciation and extinction. Systematic Biology 56:701-710.

Mitter, C. , B. Farrell, and B. Wiegmann. 1988. The phylogenetic study of adaptive zones: has phytophagy promoted insect diversification? American Naturalist 132:107-128.

---

## Statistical inference using simulations or randomizations

Mesquite has various tools to simulate and randomize characters and trees, including simulations of DNA sequence evolution, tree reshuffling, coalescent gene tree simulations, and others. With these tools, you can build your own statistical inference methods tailored to your question.

For instance, you can test an hypothesis about phylogenetic structure using parametric bootstrapping, simulating many character matrices on a proposed tree, for each attempting to reconstruct the tree and assessing the reconstruction. Or, you can simulate gene trees within a population to test a population genetics or phylogeographic hypothesis. Or, you can study the effect of branch length uncertainty on calculation of Felsenstein's contrasts by adding random noise to branch lengths on a given tree, repeating many times to see the variation in calculated correlation coefficient of contrasts in two characters. Many other analyses are possible. We give an overview of the features and some examples in the following.

- Simulating and randomizing characters
    - [Overview](#)
    - [Simulating DNA sequence evolution, with examples of hypothesis testing](#)
    - [Simulating coalescence and sequence evolution within populations](#)
  - [Simulating and randomizing trees](#), including species trees and gene trees
-

# Character simulations and randomizations

Mesquite can simulate and randomize characters to build statistical tests. On this page we give an overview of these features. A more in-depth account of simulation of DNA sequence evolution is given [separately](#).

## Contents

- [Using results of simulations & randomizations](#)
- [Simulations of character evolution](#)
- [Viewing results of simulations](#)
- [Randomizing characters](#)

## Using results of simulations & randomizations

The simulated or randomized characters can be used or stored in several ways:

- The characters can be stored into matrices in the current file by choosing options in the **Make New Matrix** from submenu of the Characters menu. For instance, if you choose Simulated Matrices on Current Tree, the matrix simulated will be stored in the file.
- The characters may be used directly, at that moment, in calculations. For example, if you make a Bar & Line Chart for Characters, and choose Simulated Characters as your source of characters, the characters will be simulated and used in the chart without being stored in the file.
- A series of many data files can be saved, each one with a different replicate of the simulated or randomized data matrix. This is available through the **Save Multiple Matrices** submenu of the Character menu
- A series of many data files can be saved in combination with scripting files to instruct programs such as Swofford's PAUP to run the files. This can be done using the **Batch Architect**, a description of which is in the page on [DNA simulations](#) and some of the [Studies](#).

To replicate the results of a simulation or randomization, you can use the **Set Seed** menu item to set the random number seed used. If you are using the same conditions, including the same seed, the simulations and randomizations should be reproducible.

## Simulations of character evolution

Stochastic models can be used to simulate character evolution along the branches of a phylogenetic tree by selecting Simulated Characters (to generate characters one at a time) or Simulated Matrices on Current Tree (to generate whole matrices, on a current tree in a Tree Window), or Simulated Matrices on Trees (to generate whole matrices, each one on a different tree from a source of trees). These options are available whenever characters or matrices might be called for, for instance when making a chart of characters or matrices.

The following are the character types and models that can be simulated:

- **Evolve Categorical characters.** The following models are also discussed in the section on [likelihood reconstructions](#).
  - **Mk1 model** – Single parameter model analogous to Jukes-Cantor. Rates of change equal for all types of state-to-state changes.
  - **AsymmMk model** – Two parameter asymmetrical model with differing rates of forward and backward changes. Forward changes are those in which state number increases (e.g., state 0 to state 1); backward changes are those in which state decreases (e.g., state 1 to state 0). One can specify the forward and backward rates directly, or alternatively, one can specify an overall rate of change in combination with a bias of forward versus backward. This model will generally be appropriate only for binary characters. You can also specify whether the states are the root are assumed to be at the equilibrium frequencies implied by the model. If so, then Mesquite chooses an ancestral state in the simulations according to the equilibrium frequencies implied by the bias in gains versus losses. Otherwise, Mesquite chooses a state with equal probabilities.
- **Evolve DNA characters**
  - See page on [DNA simulations](#) for details.
- **Evolve Continous characters**
  - **Brownian motion model** – Model with a single parameter, the rate of change.

To use these simulations, the appropriate character model must be defined in advance, with all of its parameters specified. Two models come built-in: a Jukes-Cantor model for DNA simulations, and a Brownian motion model for continuous variable simulations. If you want any other models, created them using **New Character Model** in the Characters menu.

## Viewing results of simulations

Simulated characters can be used in many calculations, but if you want to visualize directly the results of a simulations you can use the **Trace Character History** feature available in the Analysis menu of the Tree Window. By default Trace Character History shows a reconstruction of ancestral states. Thus, if the character is simulated, the states at nodes shown would not be the "true" ancestral states that occurred during the simulation, but rather states inferred from the states given to the terminal taxa by the simulation. However, once Trace Character History is active, its Trace menu has a **Character History Source** menu item. Choose **Simulate Ancestral States** and specify the simulation. The states indicated at the nodes will then be the true ancestral states in the simulation. You can set the Seed to make the simulation equivalent to simulations done in other contexts.

## Randomizing characters

Existing characters can be randomized as follows:

- **Reshuffle Character** – Supplies replicate reshufflings of a single chosen character. In each reshuffling, the character states are randomly scrambled among taxa, keeping the frequencies of different character states fixed.
- **Reshuffle States within Characters** – Supplies matrices, each of which is a reshuffling of an existing matrix. The first character of the matrix is a reshuffling of the first character of the original matrix; the second character is a reshuffling of the second original character; and so on. You can think of this as reshuffling within each column

of the matrix.

- **Reshuffle Within Characters (Taxa Partitioned)** – As for Reshuffle States within Characters, but respecting taxa partitions. Each character of the matrix is a reshuffling of the respective character of the original matrix, but done only within groups of the current taxa partition. Thus, if taxa have a current [partition](#) that divides the taxa into Group A and Group B, then the reshuffling within a character first shuffles all states of the character among Group A taxa. Then, it shuffles states of the character within Group B taxa.
- **Reshuffle States Within Taxa** – Supplies matrices, each of which is a reshuffling of an existing matrix. Instead of reshuffling within each column, this reshuffles within each row. Thus, the states of each taxon are reshuffled among characters. This might be used, for instance, to generate DNA data with no phylogenetic signal that preserves the base composition of each taxon.
- **Reshuffle Within Taxa (Char. Partitioned)** – As for Reshuffle States within Taxa, but respecting character partitions. Thus, if the characters have a current partition that divides them into (for example) groups 28S and COI, then the reshuffling within a taxon first shuffles all states of a taxon among 28S sites, then shuffles all states of the taxon among COI sites.
- **Bootstrap resample** – Supplies matrices, each of which is a bootstrap resampled version of an existing matrix. Characters are sampled randomly from the original matrix and moved into the resampled matrix until it contains as many characters as were in the original matrix. Some of the original characters may by chance be sampled more than once; some may be not sampled at all.
- **Rarefy characters** – Supplies matrices, each of which is derived from an existing matrix by randomly deleting entire characters.
- **Sprinkle missing** – Supplies matrices, each of which is derived from an existing matrix by randomly assigning "missing" (?) or unassigned) to cells of the matrix with a particular probability.
- **Add noise** (for continuous matrices only) – Available in the Character Matrix Editor under Alter/Transform, this adds noise to the states of all or selected cells of the matrix.
- **Random Fill** – Available in the Alter/Transform of the Matrix menu of the Character Matrix Editor, it can be used to fill all or selected cells of the matrix with randomly chosen states.

# Simulating DNA sequence evolution, with examples of hypothesis testing

This section provides a brief introduction to some of the analyses you can do by simulating DNA sequence evolution, including:

- [Simulating DNA sequence evolution on a tree](#)
- [Building more complex models](#)
- [Batch Architect: Simulating, exporting, and analyzing multiple matrices](#)
- [Examples of tests of phylogenetic hypotheses](#)

Example files for this chapter are in the "Character\_Simulations" folder within the "examples" folder within Mesquite\_Folder.

## Simulating DNA sequence evolution on a tree

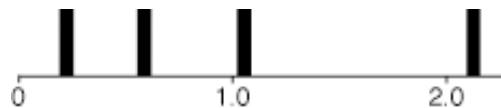
First, let's try evolving DNA sequences up the branches of a model tree, thereby creating an entire simulated matrix. Open the example file "01-modelTree.nex" and you will be presented with a tree. For the moment it doesn't matter where this tree came from; the important thing to note is that it has branch lengths inferred from a matrix of 18S rDNA; this matrix is also included in the file.

In order to simulate DNA sequences, you need:

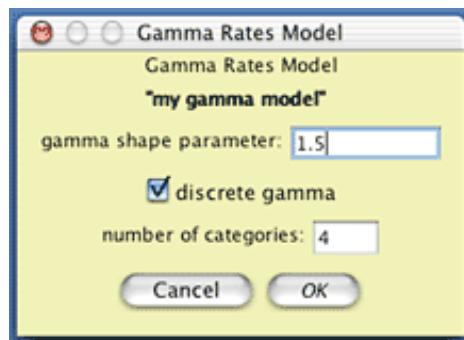
- A tree, with branch lengths specified. A tree with branch lengths is included in this example file.
- A specification of how the DNA sequences evolve. This model could have many elements. One common form of model specifies the relative frequencies of A, C, G, and T; the relative rates of change of the different characters; and the relative rates of change between A and C, A and G, and so on.

## Creating a model of character evolution

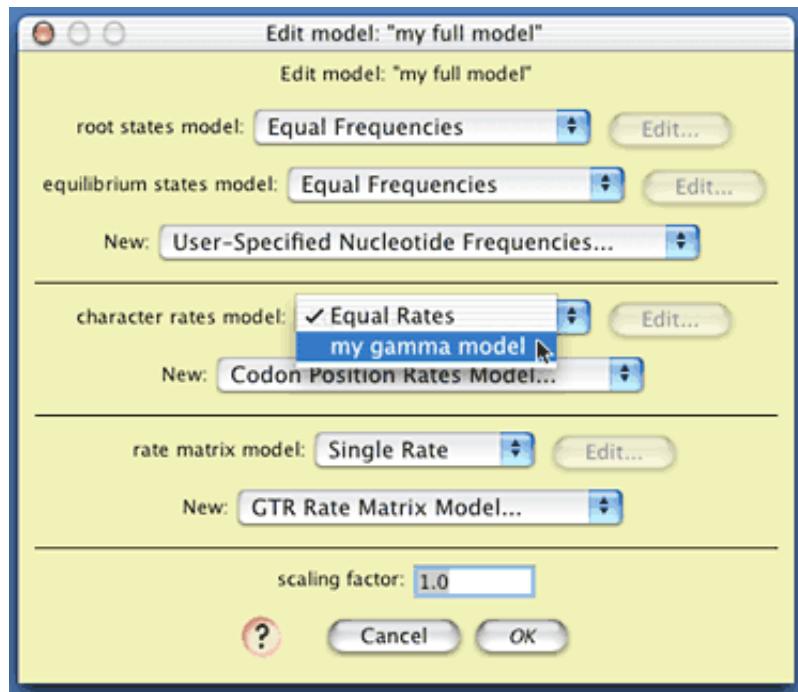
For sake of simplicity, let's presume that A, C, G, and T occur at equal frequencies, and that all pairs of nucleotides have equal rates of change between them. To make it interesting, though, let's presume that some characters evolve more quickly than others, with the distribution of rates of change following a discrete gamma distribution with shape parameter 1.5. If one presumes (as we will) that there were four categories of rates, then this model presumes that one quarter of the characters have rate 0.225, one quarter 0.589, one quarter 1.05, and the rest 2.136, as shown in the following figure:



To create this model of character rate variation in Mesquite, choose [\(Tree\) Characters>New Character Submodel>Gamma Rate Variation Model...](#) You will be presented with a dialog box in which you can choose the name of the model you will create; you might call it "my gamma model". You will then be presented with a dialog box allowing you to specify the nature of the model; set the shape parameter to 1.5:



Now we need to create a composite model of character evolution that specifies both this gamma rate variation model, the state frequency model, and the state-to-state rate model. To do this, choose [\(Tree\) Characters>New Character Model>Composite DNA Simulation Model...](#), name the model (perhaps call it "my full model"), and you will be presented with a dialog box in which you can choose the various submodels:



The only element of this model we will change is the character rates model; choose "my gamma model" from the drop-down menu to select the gamma distribution model you

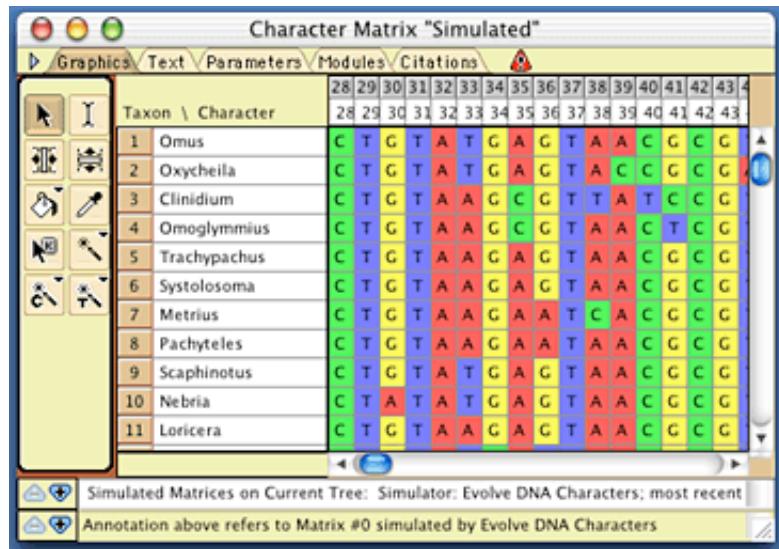
created as the model of site-to-site rate variation. The model "my full model" is now ready to use.

Note that the dialog box in which you edited "my full model" has button with a ? on it (  ).

When a button like this is present in a Mesquite window, touching on it will display a small note with information about how to use the window's feature.

## Simulating evolution

Now choose (Tree) Characters>Make New Matrix from>Simulated Matrices on Current Tree. This will ask Mesquite to simulate evolution of characters up the current tree in the tree window to produce a new matrix. You will be asked what sort of character simulator to use. As we wish to evolve a DNA sequence up the tree's branches, choose "Evolve DNA characters" from the list. (There might be other options presented, such as Evolve Continuous Characters, but we don't wish to try that now.) You will then need to chose the model of DNA sequence evolution; choose the composite one you created earlier called "full model". When it queries for the number of character, choose a large number, such as 2000. When it asks for the name of the matrix to be created, call it "simulated" (although you could choose another name, if you wish). Mesquite will then simulate evolution, and a matrix will be produced, such as this one:



The screenshot shows the Mesquite software interface with the title bar "Character Matrix 'Simulated'". The menu bar includes "Graphics", "Text", "Parameters", "Modules", and "Citations". The main window displays a character matrix with 11 taxa (rows) and 44 characters (columns). The taxa are labeled 1 through 11, corresponding to Omus, Oxycheila, Clinidium, Omoglymmius, Trachypachus, Systolosoma, Metrius, Pachyteles, Scaphinotus, Nebria, and Loricera. The columns are numbered 28 through 43. The matrix itself is a grid of colored cells representing nucleotide states (A, T, C, G). A toolbar on the left contains icons for selection, zoom, and other functions. At the bottom, there are status bars: "Simulated Matrices on Current Tree: Simulator: Evolve DNA Characters; most recent" and "Annotation above refers to Matrix #0 simulated by Evolve DNA Characters".

## Examining the results of the simulation

Let's see whether the rate variation between characters suggested by this simulated matrix matches a gamma distribution with shape parameter 1.5, as used in the model. First, save the matrix to a file so that you can read it into PAUP\*. Choose (Character Matrix) Characters>Save Copy of Matrix>simulated to save a copy of your newly created matrix; perhaps name the file "Simulated Matrix".

In PAUP\* 4, execute the file Simulated Matrix, and then execute the following commands:

```
nj;  
lset nst=1 basefreq=equal rates=gamma shape=estimate;  
lscore 1;
```

(If you don't know how to give PAUP\* commands, please read your PAUP\* documentation. There are equivalent ways to give these commands in the MacOS versions using dialog boxes, but it is much easier for us to describe what to do with these text commands.)

The first command ("nj;") will find a neighbor-joining tree. While this might not be the best way to get a tree, but it will be fast, and give a tree that is good enough for our purposes. The second command ("lset...") sets the model used for likelihood calculations to be the model used by Mesquite in the simulations, except that the shape parameter used is not specified, but is instead estimated from the data. The third command ("lscore 1;") tells PAUP\* to calculate the likelihood of the tree, in the course of which it will also estimate the shape parameter of the gamma distribution using likelihood. You might end up with PAUP\* reporting something like this:

```
Tree 1  
-----  
-ln L 19742.05025  
Shape 1.583552
```

The shape parameter should be around 1.5.

## Building more complex models

More complex composite models of DNA evolution can be built by choosing among the following:

### *Models of root state frequencies:*

- Equal Frequencies: equal frequencies of states
- Empirical Frequencies: frequencies of states the same as that found in an existing character matrix
- User-specified Nucleotide Frequencies: frequencies of nucleotides specified by the user. You can create a model of this sort by choosing (Character Matrix) Characters>New Character Submodel>User-specified Nucleotide Frequencies... and then entering the appropriate values in the dialog box.

### *Models of equilibrium state frequencies on other branches:*

- Equal Frequencies: equal frequencies of states
- Empirical Frequencies: frequencies of states the same as that found in an existing character matrix

- User-specified Nucleotide Frequencies: frequencies of nucleotides specified by the user. You can create a model of this sort by choosing (Character Matrix) Characters>New Character Submodel>User-specified Nucleotide Frequencies... and then entering the appropriate values in the dialog box.

The above two submodels corresponds in PAUP\* to the specifying the base frequencies in the likelihood settings; note that PAUP\* does not allow separate specification of the model of frequencies and the root and at other branches.

#### *Models of rate variation among characters:*

- Equal Rates: all characters evolve at the same rate.
- Codon Position Rates Model: a model specifying the relative rates of the different codon positions. If you wish to have the codon positions match those in an existing matrix, then codon positions need to be specified for a DNA matrix. To do this, choose (Character Matrix) Characters>List of Characters to see the current codon positions. Select the characters in this list window, and then touch on the title of the column "Codon Position". A drop-down menu will appear in which you can choose to set the codon positions to a sequence like 123123123... as appropriate.
- Gamma Rates Model: a model specifying that rates of characters evolve according to a gamma distribution.
- Gamma Invar Rates Model: a model specifying that a proportion of the characters are invariant, and the remainder follow a gamma distribution.
- Proportion Invariant Model: a model specifying that a proportion of the characters are invariant, and the remainder evolve at one rate.

The first model ("Equal Rates") is built in; you can create a model of the other kinds by choosing the appropriate items from the (Character Matrix) Characters>New Character Submodel menu and then entering the values in the dialog box.

#### *Model of rate matrices:*

- Single Rate: all changes between nucleotides occur at the same rate.
- Ti/Tv Rate Matrix Model: the two-parameter rate matrix model in which transversions occur at a different rate than transversions.
- GTR Rate Matrix Model: the General Time Reversible, six-parameter rate matrix model in which you can specify rates of each type of nucleotide change.

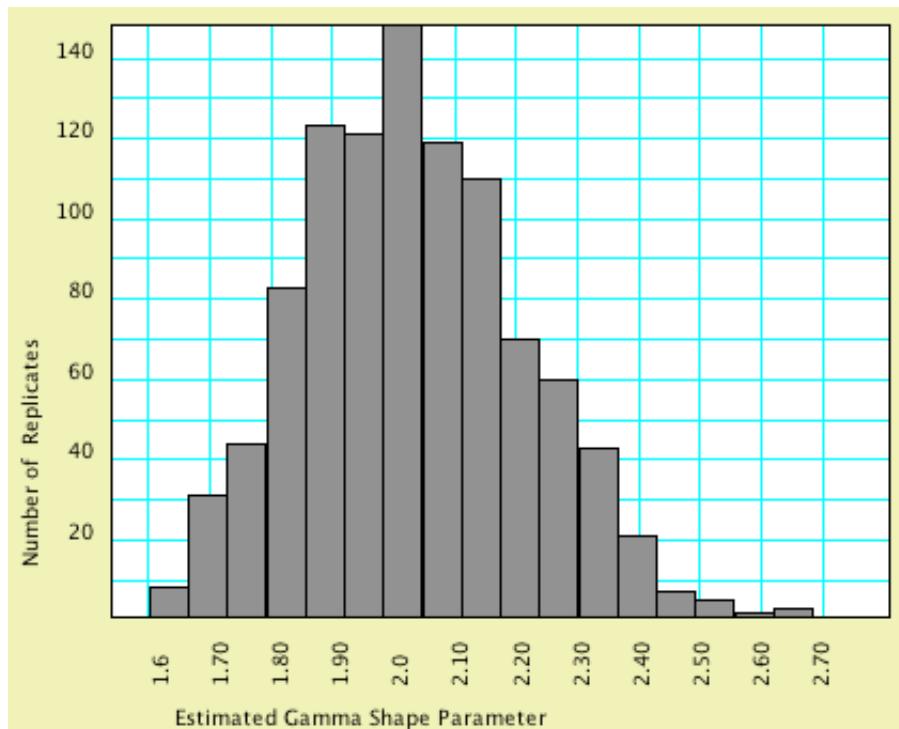
The first model ("Single Rate") is built in; you can create a model of the other kinds by choosing the appropriate items from the (Character Matrix) Characters>New Character Submodel menu and then entering the values in the dialog box.

## **Batch Architect: Simulating, exporting, and analyzing multiple matrices**

Mesquite's **Batch Architect** package can be used to create multiple simulated matrices, export them to files, and create one or more files that provide instructions to programs to

analyze the files produced. These instruction files are called "batch files".

One could, for example, examine the quality of Mesquite's simulation algorithms by simulating 100 matrices using a gamma model of character rate variation. In addition to creating 100 files with matrices, Mesquite would create a batch file containing instructions for PAUP\* to estimate the gamma shape parameter for each of the matrices and save the resulting values to a score file. Mesquite would also create a file that would provide instructions to Mesquite to allow it to read PAUP\*'s score file and display the resulting distribution of estimated gamma shape parameters in a chart. As a concrete example, simulation of 1000 matrices of 2000 characters each under a model with a gamma shape parameter of 2.0, estimating of the shape parameter by PAUP\* for each of these matrices, and processing of the results by Mesquite yielded the following histogram:



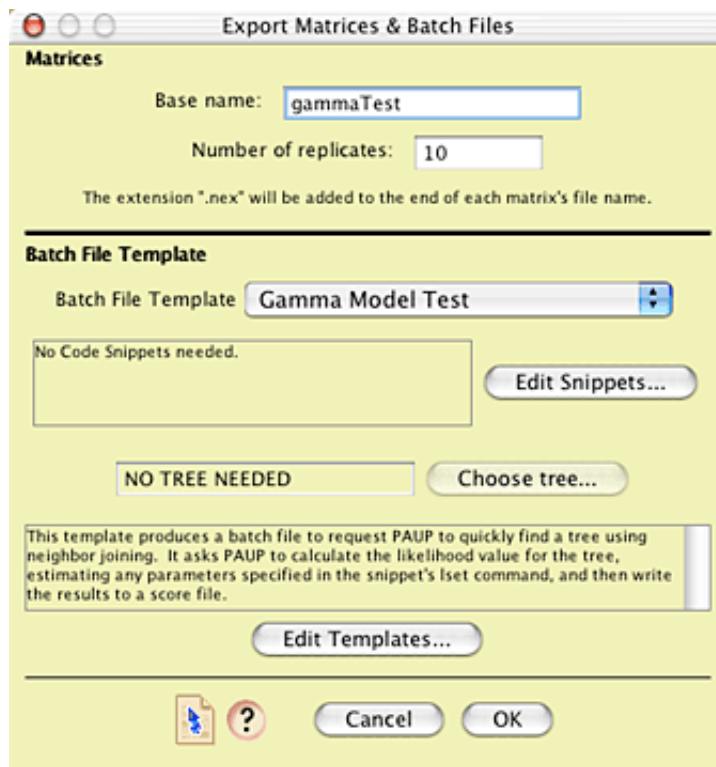
In this particular analysis, most matrices were estimated to have a gamma shape parameter close to that used in the model that generated the data; the average estimated gamma shape parameter over the 1000 replicates was 2.017.

### Conducting multiple simulations

To conduct an analysis similar to this one, open up the example file "**02-gammaTest.nex**". A complete model of DNA sequence evolution has already been added to that file, under the name "gamma2model". It specifies, among other components, character rate variation following a gamma distribution with shape parameter 2.0. First, you will need to ask Mesquite to evolve several matrices up the branches of the phylogeny. Choose ([Tree Window](#)) [Analysis > Matrices & Batch Files > Export Matrices & Batch Files...](#). You will be presented with a choice of the source of matrices; choose "Simulated Matrices on Current Tree", and press OK; for

the character simulator, choose "Evolve DNA characters". When asked to choose the model, choose "gamma2model".

You will now be presented with the main dialog box that you will need to master to use Mesquite's simulation tools. It looks approximately like this:



The top part of the dialog box allows one to choose the base part of the name of the simulated matrices that will be produced, as well as to specify how many matrices will be produced (the "number of replicates"). Let's call have the file names begin with "gammaTest", so enter that for the base name. In the interests of time, you can leave the number of replicates be only 10 (a more thorough analysis would involve many more replicates).

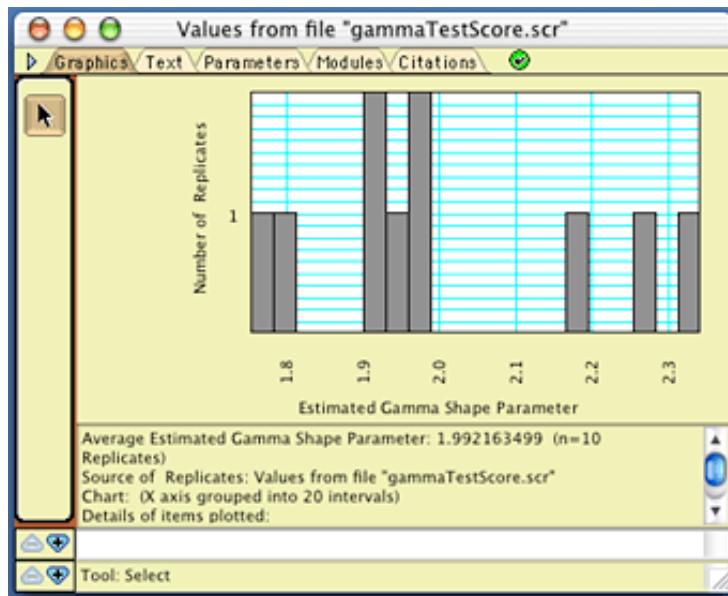
In addition to creating the matrices, Mesquite will also produce one or two files accompanying those matrices that specify commands to be used by various software (e.g. PAUP, NONA, even Mesquite) to analyze those matrices. The manner in which these accessory files or "batch files" are written is controlled by the lower portion of the dialog box. The content of the batch files is specified by a "batch file template". There are some templates supplied by Mesquite; you can also build your own. The template we need to use for this example is not built into Mesquite; you will need to load it in from a file. To do this, go into the Templates Manager, by pressing the Edit Templates button. In the dialog that is presented, you can see the build-in templates:



To load in the template we need to use, press the Load button, and open the "GammaTest.template" file that is in the "templates" folder within the Simulations examples folder. The template Gamma Model Test will now be added to the templates list. (You can see the nature of the template by selecting it in this list and pressing "Edit", but you needn't do this now.)

Once the template is loaded, press Done to return to the Export Matrices & Batch Files dialog box. Make sure "Gamma Model Test" is selected, and press "OK". You will now be asked to choose a folder to store the created files. As multiple files will be created, you may wish to create a new, empty folder to house the files. Once the location is chosen, you will be asked (after perhaps dismissing a notice or two) for the number of characters to be simulated within each matrix. Choose a fairly high number, such as 2000. Mesquite will now proceed to create through simulation all 10 files, plus two batch files.

One of these batch files will be a file containing PAUP commands that will instruct PAUP\* to read in each matrix file, quickly build a preliminary tree (using neighbor-joining, which should be good enough for our purposes), and estimate the gamma shape parameter using likelihood. The resulting value will be saved to a PAUP "scorefile". Open up PAUP\* 4 (make sure you have the latest version of PAUP\*), and ask it to execute the file "gammaCommands.nex". After PAUP has finished its analysis, go back to Mesquite, and choose (Tree Window) Analysis > Batch Architect > Show Results via Instruction File.... You will be asked to first choose an "instruction file", which is a text file telling Mesquite how to interpret another file. Choose the file "MesquiteInstructions", which is the second batch file created by the Gamma Model Test template. After Mesquite processes the instruction file, it will ask you to find the results file, which in this case is the score file created by PAUP\*. It should be called "gammaTestScore.scr" (if the base name of the matrices was "gammaTest"); open it when asked for the results file. You will then be presented with a histogram of the estimated gamma shape parameter values, that will look approximately like this:



Of course, your exact values should differ from these, but should cluster around 2.0. If you repeated this, but with many more replicates, then you should see a bell-shaped curve, as shown above.

### Testing simulations of more complex models

A test of accuracy of Mesquite's more complex models can also be done. For example, there are seven parameters in model that includes a General Time Reversible model of DNA evolution and assuming a proportion of the characters are invariant with the remainder following a gamma distribution. In 1000 replicates of 2000 characters evolved up the branches of a 49-taxon tree, the average value of these parameters as estimated by PAUP\* is close (within 1.3 %) of the value used in model used in Mesquite, as shown by the results of one particular analysis:

Parameter	Value in model	Estimated value	Difference
Rate(A<->C)	1.8700052	1.87771	0.41%
Rate(A<->G)	4.2537253	4.26319	0.22%
Rate(A<->T)	2.5286454	2.53006	0.06%
Rate(C<->G)	0.626305	0.63152	0.83%
Rate(C<->T)	8.735118	8.77468	0.45%
proportion invariant	0.506903	0.50903	0.42%
gamma shape	0.438522	0.44422	1.30%

## Examples of tests of phylogenetic hypotheses

You can use Mesquite's Batch Architect and Genesis packages to build your own statistical tests of various phylogenetic hypotheses. Some example tests are presented in the following pages:

- [Testing monophyly of a group of beetles](#)
- [Are strepsipterans related to flies? Exploring long branch attraction](#)

## Other analyses that Mesquite can do

The examples given about used the Genesis package in Mesquite to simulate the evolution of characters under a specified model of DNA evolution, and up the branches of the current tree on the screen. Many more sorts of analyses can be done, using different sources of characters other than simulated DNA data, different sources of trees other than the current tree on the screen, and different calculations by other programs.

*Some of the different sources of character matrices include:*

- Simulators that evolve data other than DNA data, including continuously valued data.
- Random modifications of existing matrices, including by non-parametric resampling (as used in classic bootstrap methods), random reshuffling of data, and jackknifing.

*Some of the different sources of trees include:*

- Simulations using markovian models of speciation
- Coalescence simulations of gene trees
- Modifications of an existing tree by random modification of branch lengths
- Modifications of an existing tree by randomly pruning a fraction of the taxa or by randomly adding taxa
- Randomly sampled trees among all possible trees
- Trees generated by randomly reshuffling taxa at their tips

## References

Shull, V., A.P. Vogler, M.D. Baker, D.R. Maddison, and P.M. Hammond. 2001. Sequence alignment of 18S ribosomal RNA and the basal relationships of adephagan beetles: Evidence for monophyly of aquatic families and the placement of Trachypachidae. *Systematic Biology*, 50:945-969.

# Gene Trees within Populations and Species Trees

Mesquite's features for handling gene trees within populations and species trees are relevant for population genetics, phylogeography, and study of gene families. They can also be used by analogy for host-parasite or other studies of associated taxa. Some of the relevant features and calculations are:

- simulations of gene trees by coalescence, either within a single population or in a diverging population or species tree
- simulations of sequence evolution, which can be used to evolve haplotypes on a gene tree
- calculations of fit of a gene tree to a population or species tree or population subdivision ([Slatkin & Maddison's s](#), [deep coalescences](#), [gene duplication and extinction](#))
- randomizations of trees to obtain null distributions for these measures of fit
- searching for population trees that optimize fit of gene trees
- cluster analysis of populations using similarities of contained gene sequences
- charts, scripting and production of batch analyses to yield statistical tests

In these analyses individual samples or haplotypes will generally be represented by taxa. Thus, each gene sequence will be a taxon, and the sequence itself will be a row in a DNA matrix.

Many of the features and calculations described below are illustrated in the example data files in the Mesquite\_Folder/examples/Coalescence/ folder. The majority of these features are provided by the Coalescence package (mesquite.coalesce) and the taxa association package (mesquite.assoc); some are provided by the distance analysis package (mesquite.distance).

## Contents

- [Importing and Exporting Data](#)
- [Single populations: Coalescence simulations](#)
  - [Example: Chart of coalescence depths](#)
- [Multiple populations or species](#)
  - [Establishing the association between genes and species/populations](#)
  - [Editing already-created taxa associations](#)
  - [Simulating coalescence within a species/population tree](#)
  - [Reconstructing gene history within a species/population tree](#)
  - [Visualizing gene history in a species/population tree](#)
  - [Measuring fit between genes and species/population trees](#)
    - [S](#)
    - [Deep coalescences](#)

- [Gene duplications/extinctions](#)
- [Example: Effect of population divergence time on s](#)
- [Inferring the population or species tree](#)
  - [Tree search](#)
  - [Cluster analysis](#)
- [Simulating sampled gene sequences](#)
  - [Generating a single matrix of sequences](#)
  - [Generating a series of matrices](#)
  - [Example: Multiple simulations of sequence samples](#)
- [References](#)

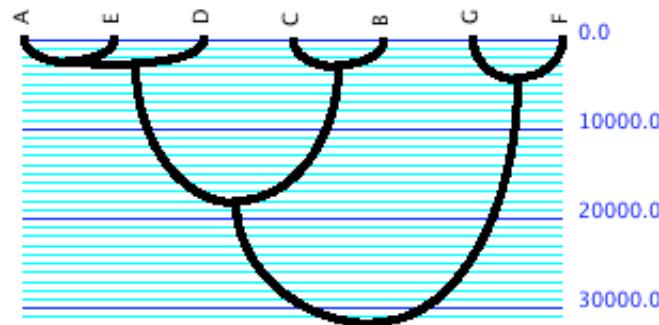
## Importing and Exporting data

Mesquite can import and export gene sequences in text files in the following formats: Simple text table, NBRF/PIF, PHYLIP, .ss (NONA, Hennig86, WinClada) as well as NEXUS files.

## Single population: Coalescence simulations

Gene trees within a single population can be simulated under the assumption of neutrality, panmixia and constant population size. A sample of such simulated gene trees can help you generate null expectations in tests, for instance of population subdivision. Mesquite simulates these gene trees by a coalescent process, beginning with the set of defined genes (taxa) and coalescing back in time until a single common ancestor is reached. Simulated coalescent trees can be viewed or used in other contexts where a source of trees is used – in the Tree Window or in Trees charts, for instance.

To view simulated gene trees, first prepare a data file with taxa representing the sampled gene copies. Select New Tree Window from the Taxa&Trees menu, and indicate you want Simulated Trees as your tree source. Choose Coalescent Trees as your tree simulator. You will be asked to indicate an effective population size. The simulated tree that appears will probably not be shown with branch lengths indicated; for optimal viewing we suggest indicating you want "Branches Proportional to Lengths" in the Drawing menu, and use the Tree Form of Curvogram. You may also want to choose a narrower line width. The tree might look something like this:



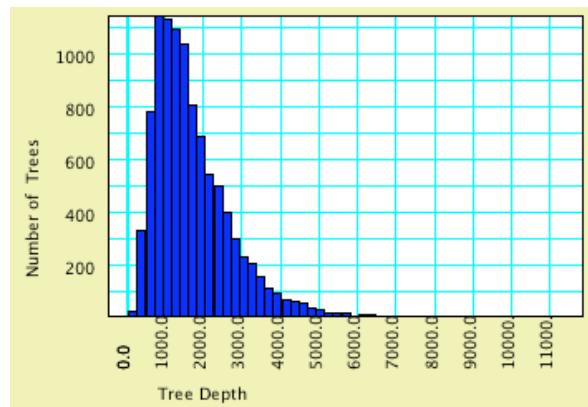
The blue numbers represent time in generations. To examine other simulated gene trees, scroll from one to another using the blue arrows in the upper left corner of the Tree Window. You will notice that depth of the trees vary, depending on when the last coalescence happened during the simulation. In order to fit the tree into the window nicely, the scale of generations changes. You may find it interesting to prevent this rescaling, so that all trees are shown to the same scale. You can do that using the "Fixed Scaling" menu item in the Drawing menu.

The parameters of the simulation may be changed using the Set Seed (Tree simulation) menu item, and the items in the Coalescence Simulations submenu. To change effective population size, select "Set Ne". These simulations treat the organisms as **haploid**. For reasonably large population sizes, an exponential approximation can be used in the simulations to avoid having to model all genes in the population explicitly. This exponential approximation is the default; you can turn it off using the "Exponential approximation" menu item.

Simulated gene trees can be generated and saved in a trees block in your data file. To do this, select Taxa&Trees>Make New Trees Block From>Simulated Trees>Coalescent Trees. Simulated gene trees can also be used in charts, as in the following example.

### Example: Chart of coalescence depths

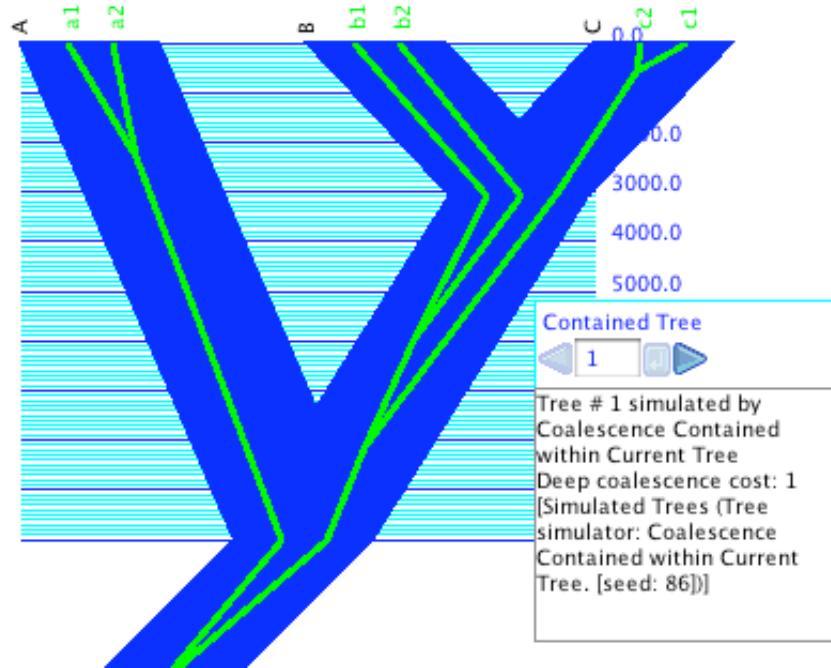
Simulated gene trees can be examined in charts by using Simulated Trees as your tree source. For instance, the example file 03-coalescenceDepth.nex shows a chart of time to final coalescence in a sample of 100 gene trees. This was made by asking for the Bar & Line Chart for trees, using Tree Depth as the value to calculate (this is a secondary choice), and Simulated Trees, Coalescent Trees as the source of trees. Here is the same chart modified to ask for a sample of 10,000 gene trees.



The effective population size is 1,000. Although the average time to final coalescence is about 1760 generations, the curve (as expected) has a long tail. By touching on the Text tab of the window you can see that there was one simulated tree with depth >11,100 generations.

# Multiple populations or species

Gene sequences from multiple populations can be analyzed and modeled in Mesquite. For instance the following shows a simple example of 6 genes within 3 extant species or populations, in which a coalescence simulation within a diverging species tree (blue) generates a hypothetical gene tree (green).



To treat a set of gene sequences as being distributed across multiple species, you incorporate them into a single matrix, then indicate to which species each sequence belongs. Three components must be established in your data file to do this:

1. A block of taxa representing the **gene sequences**. The sequences themselves do not need to be represented by a DNA matrix, although the matrix could of course be important for some analyses. For purposes of gene tree simulations, however, it is enough that each gene sequence be represented by a taxon in a taxa block.
2. A block of taxa representing the **species** (or populations). Each taxon in this block represents a different species or population.
3. A **taxa association block**, which is a special block of information that indicates how the taxa representing genes are associated with the taxa representing species. It is this that indicates for each gene what species it belongs in.

Once these three components are established, you can ask to do calculations (e.g., gene tree simulations) using the genes in a way that pays attention to the species in which each gene belongs. We will next describe how to set up these three components, then what calculations can be done.

## Establishing the association between genes and species/populations

In this section we describe how to set up a Taxa Association, that is, to indicate how the taxa in one block are associated with the taxa in another block. We will explain this in the context of gene trees and species trees to establish what genes belong to what species. However, setting up a Taxa Association can also be useful for studies of parasite and host cospeciation, or for [fusing sequences](#) of different genes into a single matrix.

Let's suppose you want to set up a file with six gene copies (sequences) distributed among three species (genes a1 and a2 in species A, b1 and b2 in species B, and c1 and c2 in species C).

1. Make a block of taxa representing the genes. This can be done by importing a data file with gene sequences, by creating a new file in Mesquite, or by selecting New Block of Taxa from the Taxa&Trees menu. Give this block of taxa a recognizable name, like "Genes". Give the genes (taxa) names in the List of Taxa window or the Character Matrix Editor. In our example, the names are a1, a2, b1, b2, c1, and c2.
2. Make a block of taxa representing the populations or species containing the genes, and give it a recognizable name, such as "Species". Give the species (taxa) names in the List of Taxa window that will appear, e.g. A, B and C.
3. Choose New Association... from the Taxa&Trees menu. You will be asked which will be the first block of taxa in the association. Although you can choose either genes or species to be the first block, it will typically be easier for you to edit the taxa association if you choose the containing taxa (species) as the first block. Then, name the association. You will be asked whether you would like to edit the association from the perspective of the first or second block of taxa; typically you would choose to edit it from the perspective of the containing (species) taxa. You will be presented with the List of Taxa window with a special Association Editor panel to the right, and a column in the table showing the genes currently associated with each species.

The screenshot shows the Mesquite software interface with the title bar 'Taxa "Species"'. Below the title bar is a menu bar with 'Graphics', 'Text', 'Parameters', 'Modules', and 'Citations'. On the left, there is a vertical toolbar with icons for selection, text, parameters, modules, and citations. The main area contains a table with the following data:

Taxon	Default	Group	Genes
1 A	1	?	a1, a2
2 B	2	?	-
3 C	3	?	-

To the right of the table is an 'Association Editor' panel titled 'Genes association'. It contains a list of genes: a1, a2, b1, b2, c1, c2. The 'b1' and 'b2' entries are highlighted in orange, indicating they are assigned to species B.

To indicate what genes are associated with a species, first select a species in the table at left. In the example above, species B is selected. Then, go to the Association Editor panel, and select the genes to be assigned to that species. Touch on the arrow button to assign the genes to the species. This replaces any current assignment to the species. If you want to add to the genes already assigned to the species, use the add button. To delete genes from an assignment, select the genes and use the subtract button.

## Editing already-created taxa associations

The instructions above indicate how to create two taxa blocks (genes, species) and link them using a Taxa Association. After you have done this, you can modify the Taxa Association by choosing Edit Association from the Taxa&Trees menu.

## Simulating coalescence within a species/population tree

Simulating gene trees evolving within a species or population tree is similar to simulating gene trees in a single population as described under Single Population. To be able to do it, however, you need to have already established an association between the gene taxa and species/populations, as described above. You also need to have a Tree Window open and showing a species tree, because the calculations need to be able to find a current species tree in which to perform the simulation.

If your data file is ready to go, you can generate and use gene trees simulated within the population tree in many contexts, such as a Tree Window or chart. You merely need to specify the source of trees to be Simulated Trees, and choose the simulator to be either "**Coalescence Contained within Current Tree**" (if you are assuming no migration) or "**Coalescence in Current Tree with Migration**" (if you are assuming migration occurs). The latter is a secondary choice, and hence will be available under "Other Choices" in a menu or by selecting "Show Secondary Choices" in a dialog. Once you've selected the coalescence simulator, you will be asked to choose Effective Population Size and, if you've chosen the option with migration, the migration rate.

The simulation starts at each extant population. Within each, the ancestry of the gene copies contained (as specified by the Taxa Association) is simulated by coalescence, going backward in time until the simulation arrives at the previous population (= species) divergence. These within-branch simulations use the same calculations and assumptions as the Single Population simulations (neutrality, panmixia). Migration, if assumed, occurs either with a constant probability per individual per generation (and it is equally likely to go to any of the contemporaneous populations), or it occurs as a burst in a single generation. The length of time allowed within this branch is the length of the branch, which is treated as the number of generations. (Thus, branch lengths of the population tree will typically be large, e.g., 1,000 to 1,000,000 or more.) The population size is determined by the chosen  $N_e$ , and is constant throughout the simulation unless modified by branch widths. Branch widths, which can be controlled by the Adjust Lineage Widths tool (the horizontal ruler) in the Tree Window, are treated as multipliers of the basic  $N_e$ . Thus, if the lineage width is unspecified or is 1.0, then the indicated  $N_e$  is used directly. If the lineage width of a branch is 0.5, then the population size along that branch is 0.5\*(indicated  $N_e$ ). Population fluctuations such as bottlenecks can be introduced explicitly along a single branch by inserting extra nodes within the branch using the Insert Nodes tool, and then varying the widths of the different segments independently. This is shown in the example file 08-fluctuating.nex.

By the time the simulation reaches a branch point, i.e. a species/population divergence,

coalescence may have resulted in a single remaining ancestor of the sampled gene copies, or there may remain more than one ancestor. Whatever gene ancestors remain, they are united with the gene ancestors remaining in the sister population into the ancestral (pre-divergence) population. Coalescence then proceeds from there, moving backwards along the ancestral branch, and so on, until the root of the species/population tree is reached. Then the simulation continues in the root until only a single gene ancestor remains. The branch lengths in the resulting gene tree reflect the generations in which each coalescence occurred.

## Reconstructing gene history within species/population tree

If we are given a gene tree and a species/population tree, how can we interpret how the gene tree fits within the species tree? In what ancestral species did each gene divergence occur? Mesquite is currently able to make this reconstruction under only one assumption: that the only process occurring is lineage sorting (there is no migration among populations), or, equivalently, gene duplication/extinction. Thus, the reconstruction reconciles the gene tree into the population tree (Page and Charleston, 1997) so as to minimize the depths of gene tree divergences (i.e., minimizing the implied incompleteness of lineage sorting), which also minimizes gene duplication/extinction events.

This reconstruction is performed by Mesquite when visualizing gene trees within species/population trees using the Contained Associates tree drawing mode and when counting deep coalescences, both of which are described below. Three parameters determine how the reconstruction is done:

- **Treat Contained As Unrooted** – If enabled, then all possible rootings of the gene tree are tried to find that which minimizes incompleteness of lineage sorting. This is typically an appropriate setting for empirical gene trees (which are typically unrooted) but inappropriate for fitting simulated gene trees into the population history on which they were simulated, because the roots of these gene trees are known. (Default: disabled)
- **Contained Polytomies auto-resolve** – If enabled, then polytomies in the gene tree will be automatically resolved into dichotomies so as to minimize incompleteness of lineage sorting. In the visualization of Contained Associates, such resolved areas are colored magenta. (Default: enabled)
- **Use Branch lengths of Contained tree** – If enabled, then the branch lengths of the contained (gene) tree will be respected in fitting into the species/population tree. Thus, if the gene tree's branches are long and the population tree's branches short, the gene tree will be interpreted as extending deep in time past the root of the species tree, even if this implies lineage sorting is more incomplete than it might otherwise need to be. If disabled, then the branch lengths of the gene tree are ignored in minimizing incompleteness of lineages sorting. For fitting simulated gene trees into the species tree on which they were simulated, it is usually best to enable this option, for then the fit will reflect the actual history. (Default: enabled)

## Visualizing gene history in a species/population tree

The visualization of green gene trees embedded with blue species trees shown elsewhere on this page is done by the Contained Associates tree drawer. This tree drawer can be used in various contexts where trees are drawn (e.g., the Tree Window), but it requires that the data file is already prepared with two blocks of taxa and their association. To ask for this visualization, have open a Tree Window showing trees of Populations (Contained Associates draws species trees with embedded gene trees). Select Drawing>Tree Form>Other Choices and then choose Contained Associates from the dialog box. You will be asked what gene trees to draw within the population tree. The gene trees could be from any of the usual sources: stored in the file, or simulated. You can choose to show gene trees simulated within the population tree itself at that moment.

Once Contained Associates is showing the tree, many of its controls will be in the Contained menu, not in the Drawing menu as usual. If you want the contained gene tree to appear in a separate window in addition to embedded in the species tree, select Contained>Display Contained Tree.

## Measuring fit between genes and species or population trees

If the genes of a population do not form a clade in the gene tree (i.e., a monophyletic group) on the gene tree, then there is discordance between the gene tree and the population subdivision. To measure this discordance, Mesquite has these measures:

- **s** (Slatkin & Maddison, 1989). This measures discord between the gene tree and the subdivision into populations; it does not pay attention to a population tree. It treats the populations as a categorical character and counts the number of parsimony steps in this character on the gene tree; the more scattered on the gene tree are the genes from a population, the higher the **s** value. If the populations have been long separate and the only process causing this scatter is migration, then **s** can be interpreted as the minimum number of migration events between the populations.  
The **s** statistic can be calculated for gene trees whenever a Taxa Association is available indicating how the genes are associated with populations.
- **Deep coalescences** (W. Maddison, 1997). This measures the discordance between a gene tree and a species or population tree. It assumes that all discord between the population and gene trees is due to incomplete lineage sorting, and counts the number of extra gene lineages (beyond the minimum of 1) on each branch of the population tree summed over all population branches. The number of extra lineages is counted on the best fit of the gene tree into the species tree using the reconstruction methods described in the previous section. Deep coalescence can be calculated in various ways:
  - Deep Coalescences (gene tree) – This calculates deep coalescence from the gene tree's point of view. That is, it can be applied to a gene tree. It seeks a current species tree (for instance, on in a Tree Window) and measures the fit of the gene tree within that current population tree.
  - Deep Coalescences (species tree) – This calculates deep coalescence from the

species (or population) tree's point of view. That is, it can be applied to a species tree. It seeks a gene tree from available tree sources (e.g., gene trees stored in the data file) and measures how well it fits within the species tree

- Deep Coalescence Multiple Loci – This also calculates deep coalescence from the species (or population) tree's point of view, but for multiple gene trees simultaneously. It seeks a set of gene trees from available tree sources (e.g., gene trees stored in the data file) and sums deep coalescences for all of them. For instance, if you choose Stored Trees as the source of gene trees, all of the gene trees in a single stored tree block will be used. By summing deep coalescences for these gene trees, each is treated as if its descent was independent from the others, that is, an independent locus. You might expect each locus to be represented by a different block of taxa, but the Coalescence package of Mesquite cannot yet sum deep coalescences across taxa blocks. What if your include loci with different sample sizes in the different populations (or species), for instance locus P has 7 sequences from species A, 3 from B, 4 from C, while locus Q has 5 sequences from A, 3 from B and 6 from C? You can create a taxa block with sufficient taxa to accommodate both (7 genes contained in A, 3 in B, 6 in C) and then for each the gene trees representing the different loci exclude the extra genes as needed.

- **Gene duplications and extinctions.** This measures the discord between a species tree and a gene tree, assuming all discord is due to duplication of gene loci and subsequent extinction of (or failure to sample) some loci in some lineages (Goodman et al, 1979; Page, 1993, Page and Charleston, 1997). Duplications/extinctions can be counted from the perspective of either the gene tree or the species tree, as follows:

- Duplications and Extinctions (gene tree) – This counts duplications and extinctions from the gene tree's point of view. That is, it can be applied to a gene tree. It seeks a current species tree (for instance, on in a Tree Window) and measures the fit of the gene tree within that current species tree.
- Duplications and Extinctions (species tree) – This counts duplications and extinctions from the species tree's point of view. That is, it can be applied to a species tree. It seeks a gene tree from available tree sources (e.g., gene trees stored in the data file) and measures how well it fits within the species tree

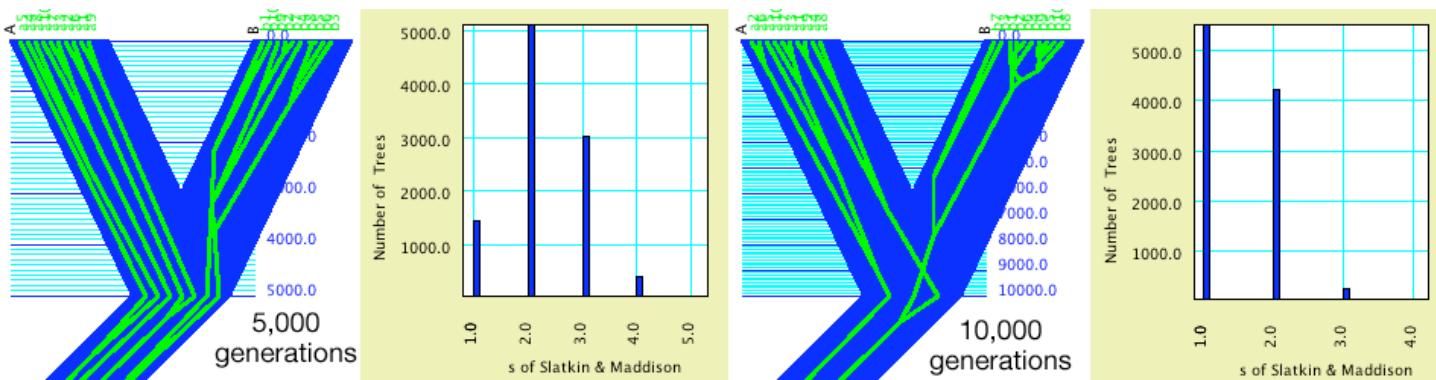
With either of these, you can choose whether to count duplications and extinctions together, or just duplications, or just extinctions. This choice can be made with the Counting Mode (Dup./Ext.) submenu.

These measures assign a value to a tree, and thus are available as "Numbers for Trees". They can be calculated and displayed in various contexts, such as a histogram of values for trees, or in the Tree Legend, or as a column in the List of Trees window. They can also be used in tree searches.

### Example: Effect of population divergence time on s

If you have reconstructed a gene tree from sampled sequences from two populations, you may want to use the degree of scrambling of the genes from the two populations as a

measure of completeness of lineage sorting, and thus time since divergence. Although the  $s$  statistic was designed to measure gene flow, it might be used instead to measure time since divergence (assuming there is no ongoing gene flow). In the example below simulations are used to derive the expected  $s$  values under different divergence times. Ten genes are in each of two populations. A population tree with branch lengths of 5,000 and 10,000 are compared. These branch lengths are used as number of generations for the coalescence simulations, which here are using an effective population size of 10,000. The green-in-blue images of gene trees in population trees show the Contained Associates tree drawing mode of the Tree Window, with the gene trees simulated by Coalescence Contained within Current Tree. The charts are a separate calculation based on the same population trees; they are Bar Charts for Trees, in which the block of taxa for the trees are the genes, the value for the trees is the  $s$  statistic, and the gene trees are derived by simulations using Coalescence Contained within Current Tree. Note that with divergence 5,000 generations ago,  $s$  values of 3 are fairly common, whereas they are quite uncommon with divergence at 10,000 generations. By adjusting branch lengths, hypotheses can be tested and confidence limits derived.



## Inferring the population or species tree

Mesquite can infer relationships of species/populations using contained gene trees or gene sequences, but currently its algorithms are relatively crude. Tree search and Cluster analysis are two options.

### Tree search

Population trees may be inferred via a tree search that finds those population trees in which observed gene trees fit best (Maddison, 1997; Page and Charleston, 1997). Mesquite's tree search facility can be used to seek population trees that minimize deep coalescences (Maddison, 1997) or duplications and extinctions. Select Taxa&Trees>Make New Trees Block From>Other Choices and indicate Tree Search in the dialog box. Select populations as the taxa for the new trees block (the search will produce population trees). When asked for the criterion for the tree search, check the "Show Secondary Choices". You will see three choices for deep coalescences. Deep Coalescences (gene tree) or Gene Duplications and Extinctions (gene tree) are inappropriate because they assess fit from the point of view of the gene

tree, but your goal is to assess and choose species (population) trees. Thus, choose either Deep Coalescence (species tree), Deep Coalescence Multiple Loci, or Gene Duplications and Extinctions (species tree). Deep Coalescence (species tree) and Gene Duplications and Extinctions (species tree) will ask you what gene tree to use as the basis for measuring fit within the candidate species trees. Deep Coalescence Multiple Loci will ask you what block of gene trees to use. More details on these criteria are given under [measuring fit](#).

Mesquite's tree search does not yet infer branch lengths, and thus the fit between gene trees and species trees is measured so as to ignore branch lengths.

## Cluster analysis

Cluster analysis can be used to infer population trees by similarity of contained genes. Select [Taxa&Trees>Make New Trees Block From>Other Choices](#) and indicate Cluster Analysis in the dialog box. Indicate you want the taxa of the new trees block to be populations (as we are building a populations tree). For the measure of distance, indicate Distance of Contained Taxa (this a secondary choice), and then for the distance among contained taxa choose Uncorrected Distance or Patristic Distance (a secondary choice). "Uncorrected distance" counts the simple number of difference in gene sequences, and thus choosing this will yield a population tree that depends on a data matrix of the contained genes, but not on a gene tree. "Patristic Distance" measures distance along the branches of the gene tree, and thus requires a gene tree but no gene sequence matrix. If for the method to count distances among contained taxa you choose "Closest", and then "Single Linkage" as the cluster method, you will have an inference method similar to that implied by Takahata (1989): the similarity between two populations is judged by their most similar pair of gene sequences (not their average pairwise sequence divergence).

## Simulating sampled gene sequences

Simulations can be used to generate gene sequences evolved under genetic drift and various models of mutation, either within a single population or within a history of diverging populations. To do this, the gene trees simulated as described above (in a [single population](#) or [multiple populations](#)) are used as the basis, and mutations layered over top of the gene tree to yield a series of simulated sampled sequences. If done on a single gene tree, a gene sequence matrix results (each taxon a sampled gene copy; each character a site in the sequence). However, one can replicate this process automatically to produce many matrices, and thus obtain statistical distributions to test hypotheses. We introduce below some of the possible ways to generate sequences. For more details see the [Character Simulations](#) page, which explains how the Genesis package of Mesquite can be used to simulate nucleotide evolution.

To simulate sequence evolution you should first define a model of evolution as described [here](#). A key issue in simulating is using the scaling factor of the model to compensate for the units by which branch lengths are measured. Gene trees simulated by coalescence have

branch lengths measured in generations, which may be in the thousands or millions, whereas most standard stochastic models expect trees whose branch lengths are much less than 10 for typical sequence divergences. For gene trees with lengths measured in generations, small scaling factors (e.g., less than 0.0001) should be used. We do not yet have recommendations as to exactly what scaling factor to use. We suggest you simulate a few matrices to find the scaling factor that gives you sequence divergences in the range desired.

## Generating a single matrix of sequences

### Gene tree already available in Tree Window

If a gene tree (simulated or otherwise) is shown in a Tree Window, you can simulate sequences simply by selecting [Characters>Make New Matrix From>Simulated Matrices on Current Tree](#). (By "shown in a Tree Window" we don't mean shown as a thin green tree within the blue population tree in the Contained Associates drawing mode. We mean in a Tree Window dedicated to showing trees for the block of taxa corresponding to genes.) If your file has multiple taxa blocks you'll be asked for which you want a new matrix; indicate the taxa corresponding to genes. Indicate that you want Evolve DNA Characters. You will be asked to choose a model of evolution and a number of characters (i.e., sequence length). After it's done, the simulated matrix will be shown to you in a Character Matrix Editor window. If the sequences appear highly saturated (many changes) it may mean the scaling factor was improperly set.

### Gene tree not in Tree Window

Select [Characters>Make New Matrix From>Other Choices...](#), then indicate Simulated Matrices on Trees as the choice. Using this, Mesquite will get the gene tree on which to simulate sequences not from a Tree Window, but from some other available source of trees, such as gene trees stored in a trees block, or simulated at that moment by coalescence. After being asked for details about the model of evolution and number of characters, you will be asked to specify "Source of trees on which to simulate character evolution for matrices". Here you choose the source of gene trees. One possibility is to choose Simulated Trees then Coalescent Trees or (under secondary choices) Coalescence Contained within Current Tree, in which case the matrix will be simulated on a gene tree simulated by coalescence at that moment.

## Generating a series of matrices

Multiple replicate sequence matrices can be generated and stored to files. This can allow you to perform a statistical test, for instance generating 100 sequence matrices under some hypothetical scenario, then examining them to see if their properties match those of an observed matrix.

### On a single gene tree

To generate multiple sequence matrices evolved on a single gene tree, display the gene tree in a Tree Window. Then choose Characters>Save Multiple Matrices>Simulated Matrices on Current Tree. You will be asked to set up the simulation as for a single matrix (see above), except that you will also be asked to supply a base name for the file, the number of matrices, and the file format. If you choose "test" as the base name, 4 matrices, and the NEXUS file format, then four matrices will be simulated and written to the files test0.nex, test1.nex, test2.nex and test3.nex. You will be asked where to save the files.

### Each matrix on a different gene tree

*Matrices alone* –To simulate a series of sequence matrices, each one evolved on a separate gene tree, select Characters>Save Multiple Matrices>Other Choices..., then select Simulated Matrices on Trees. Your choices will be similar to the preceding single tree case, except that instead of automatically choosing a gene tree in a tree window, it will use gene trees from the selected tree source. The first matrix will be simulated on the first gene tree from the tree source, the second from the second, and so on. Thus, if you choose Simulated Trees, Coalescent Trees as your tree source, then you will be generating a series of matrices, each simulated on a different gene tree simulated by coalescence within a population. These matrices represent a series of replicates of samples of gene sequences from a population evolving under drift with the specified model of mutation. You can similarly use Coalescence Contained within Current Tree to simulate genes evolving in a divergent population history.

*Matrices plus batch files* –The many sequence files generated by the preceding option can be analyzed by hand or by some other program. However, if you want Mesquite to help you analyze them automatically, you can ask it to generate batch files that can script Mesquite or some other program to analyze the resulting files one after another, automatically. A batch file is simply a text file with instructions to a program; what to do, step by step. Whether you can do your desired analysis this way depends on whether the program you want to use to analyze the files can be scripted using batch files, and whether someone has designed a batch file template for the analysis.

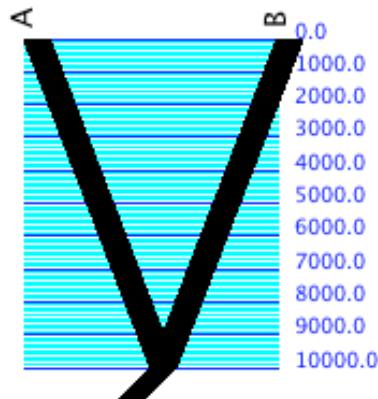
To generate a series of matrices and corresponding batch files to analyze them, select Analysis>Batch Architect> Export Matrices & Batch Files. You will be asked to specify how the matrices are to be generated, and then you will be presented with a dialog titled "Export Matrices & Batch Files". In this dialog you indicate the number of matrices to save, the base name for the files, and what batch file template to use. An example is given below, and details on the use of Export Matrices & Batch Files are given on the page on [Character Simulations](#).

### Example: Multiple simulations of sequence samples

Suppose you had 20 gene copies sampled from two populations A and B and you reconstruct a gene tree. You notice that the copies from A and B do not form respective monophyletic groups, but rather are somewhat intermingled on the tree. You calculate the degree of intermingling using Slatkin & Maddison's  $s$ , and determine it to be 4. You want to know:

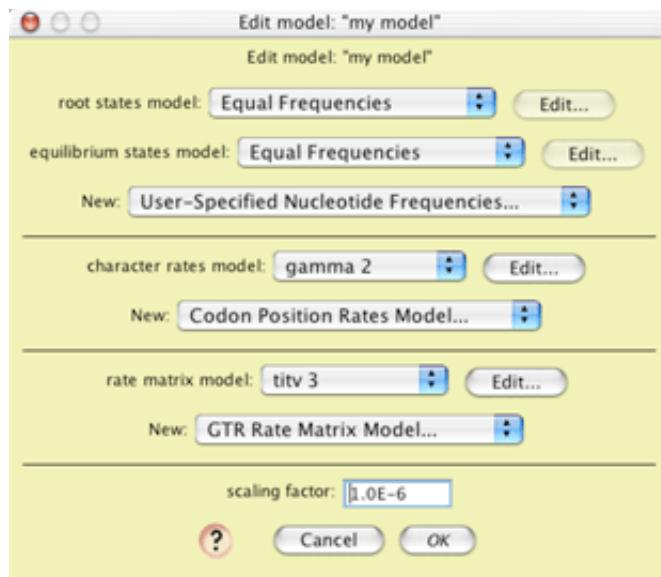
what would be the probability of observing an  $s$  of 4 if the two populations had effective population sizes about the same as the number of generations since divergence, say 10,000. (Assume the populations have been completely isolated since divergence.) How can this probability be calculated?

First set up a data file with 20 genes associated with 2 populations, and display a tree window for populations showing a tree with divergence at 10,000 generations:



One way to get a quick answer is to select [Analysis>New Bar & Line Chart for>Trees](#) and indicate you want trees representing genes, and that you want Simulated Trees, with the tree simulator Coalescence Contained within Current Tree. Indicate as the value to calculate (under secondary choices)  $s$  of Slatkin and Maddison. This would simulate a series of gene trees within your proposed population history and plot their distribution of  $s$  values. However, these are the true gene trees simulated, and you don't know that your empirical gene tree is in fact correct. It would be better if you simulated not just gene tree evolution, but gene tree reconstruction also, so that you would be able to compare simulated reconstructed gene trees with your empirical reconstructed gene tree. Thus, we will simulate the gene trees, simulate sequence evolution on them, and take those sequences and attempt to reconstruct gene trees from them.

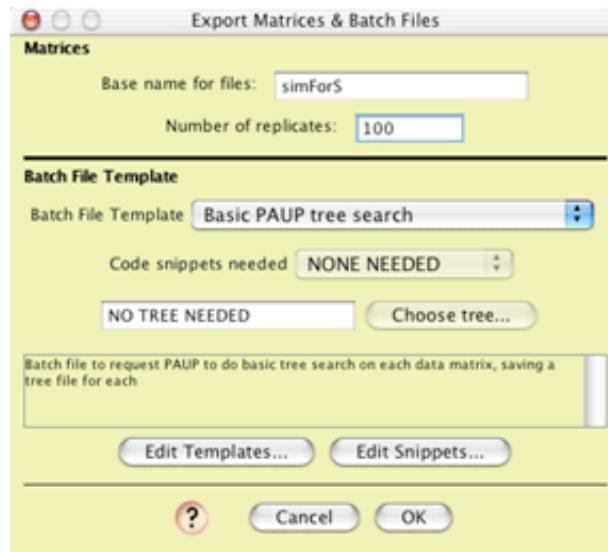
The procedure is as follows. First, set up the file to have the two taxa blocks (genes, populations) and the association between them. Display a tree window for the populations and set it to match your hypothetical scenario of population history. Our population history has two populations diverging 10,000 generations ago, with effective population size of 10,000. Build a model of sequence evolution. For instance, we built one like this:



The scaling factor (here, 1.0E-6 which is 0.000001) was chosen to yield fairly low sequence divergences under the expectation that the branch lengths on the gene tree could be as long as 10,000 to 50,000 (generations). We tested it by simulating a few matrices to see that the divergences were as desired.

Then choose Analysis>Batch Architect>Export Matrices & Batch Files, indicating to save matrices for the genes. The matrices to be exported come from Simulated Matrices on Trees (a secondary choice). Choose Evolve DNA Characters, your model, and 1000 characters to indicate how each matrix is to be simulated. For the trees on which to simulate, indicate Simulated Trees, then Coalescence Contained within Current Tree (a secondary choice). Indicate an Ne of 10,000.

You will be shown the Matrices & Batch Files dialog:



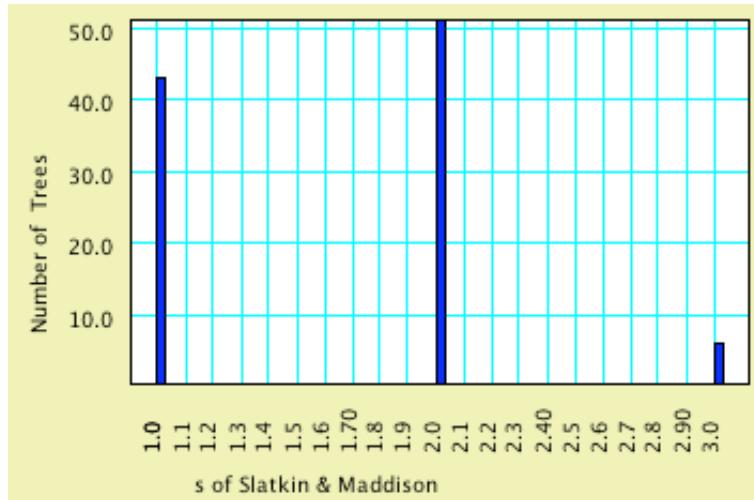
Choose a template that will reconstruct the trees for you from the matrices. We will assume for the rest of this example that the template "Basic PAUP Tree Search" was chosen. Indicate

how many matrices to make, and the base name to give to the files (here, "simForS"). Generating the matrices may take a while. Produced will be a series of files (here, "simForS0.nex", "simForS1.nex", etc.) with the matrices, and the files TreeFileList, [base name]BasisTrees.nex, and commands.nex. TreeFileList is a simple text file listing tree file names that will be produced by PAUP\*, in this case simForS0.trees, etc. [base name]BasisTrees.nex (in this example, its name will be simForSBasisTrees.nex) stores all of the gene trees simulated and used to generate the matrices. These are saved to document what was done in the simulation. The file commands.nex is the batch file that tells PAUP\* to execute each of the data files and for each infer trees. Start PAUP\* and ask it to execute commands.nex.

PAUP\* should produce the following files, assuming the base name you choose is "basename":

- basename0.trees, basename1.trees, etc. – trees reconstructed by PAUP from each of the matrices
- CBbasename – the consensus trees from each of the tree searches, stored as a series of distinct tree blocks in a single file
- consensus.trees – the consensus trees fused into a single tree block
- ConsCons.trees – the consensus of the consensus trees from all the matrices. (In this example, this file is not very useful and will likely contain fully unresolved tree.)

The key file for us is consensus.trees, because it contains the 100 trees reconstructed, one for each matrix. Back in Mesquite, ask to Link or Include this file. Then select Analysis>New Bar & Line Chart for>Trees and indicate you want trees representing genes, Stored Trees, and the value to calculate (under secondary choices)  $s$  of Slatkin and Maddison. These Stored Trees will be those reconstructed from the simulated matrices. The chart may look like this:



As you can see, it appears unlikely that you would reconstruct a gene tree giving an  $s$  value of 4 under the scenario of population history.

## References

Goodman, M., J. Czelusniak, G.W. Moore, A.E. Romeroherrera, and G. Matsuda. 1979. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Zoology* 28: 132-163.

Maddison, W.P. 1997. Gene trees in species trees. *Systematic Biology* 46:523-536.

Slatkin, M. and W. P. Maddison. 1989. A cladistic measure of gene flow inferred from the phylogeny of alleles. *Genetics* 123: 603-613.

Takahata, N. 1989. Gene genealogy in three related populations: Consistency probability between gene and population trees. *Genetics* 122:957-966.

Page, R.D.M. 1993. Genes, organisms, and areas - the problem of multiple lineages. *Systematic Biology* 42: 77-84.

Page, R.D.M and M.A. Charleston. 1997. From gene to organismal phylogeny: Reconciled trees and the gene tree species tree problem. *Molecular phylogenetics and evolution*. 7:231-240.

---

# Analyzing population structure using Slatkin & Maddison's s

## Background

This is a parametric bootstrapping approach, where an explicit model is used to generate a distribution of expected values of a test statistic. If the observed value of this test statistic falls outside of the 95% confidence interval of the simulated distribution, the model is rejected. In this case, the test statistic,  $s$ , is counting the number of changes (in parsimony steps) of a character corresponding to population membership (Slatkin & Maddison 1989). In the instructions below, we deal with two types of taxa: genes and populations. Genes are the gene sequences used to generate a gene tree; each gene sequence usually corresponds to a single sampled individual. Populations are those groupings, determined by the user, for hierarchically organizing individual gene sequences into groupings based on geography, morphology, or other information.

## Instructions & Example

These instructions are provided with a worked example. Because each model being tested is likely specific to the question being asked, many parameter values and models of population structure will be unique to each study. The models and parameter values used in this example are for educational purposes only, and we encourage investigators to allow reasonable consideration of model parameters.

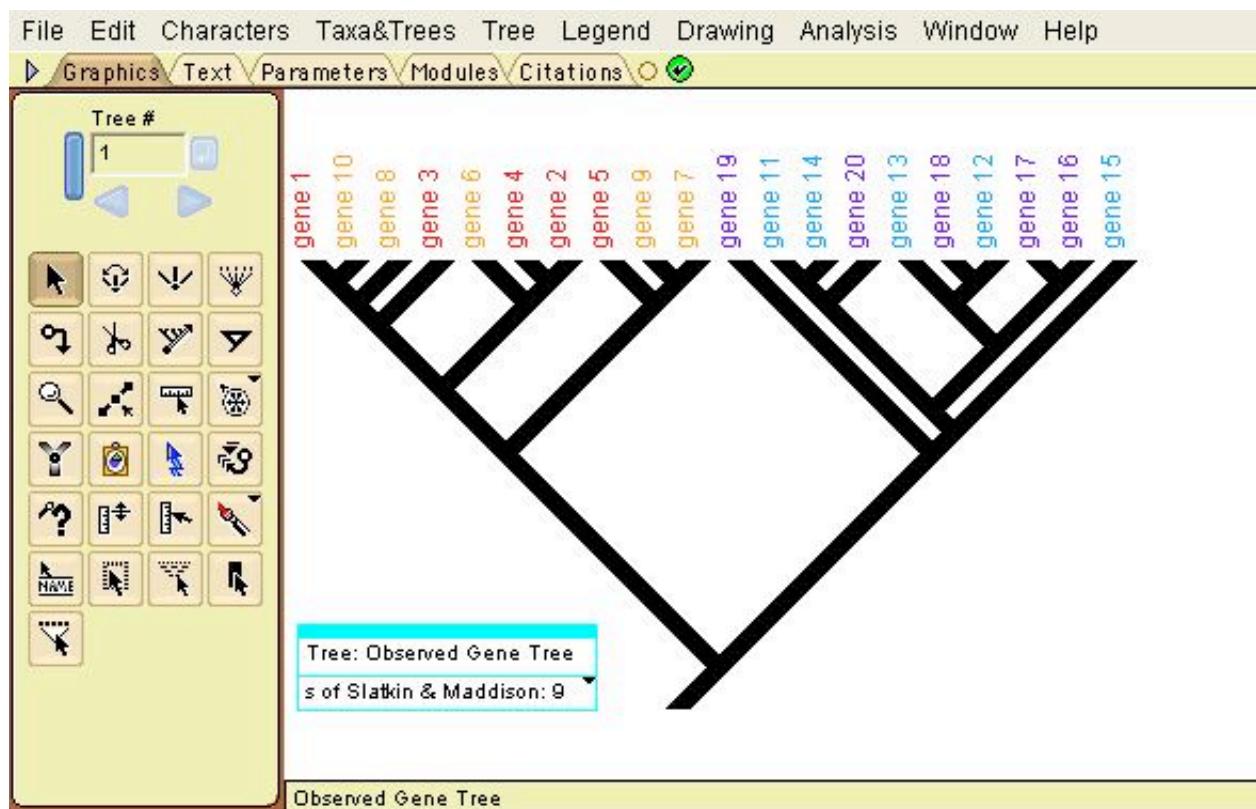
**Getting Started** – You will need a Mesquite file with the following parts:

- Taxa Block corresponding to genes sampled
- Reconstructed gene tree
- One or more Taxa Blocks corresponding to populations – You may have multiple population taxa blocks corresponding to different models of population structure. For the ease of this example, we will only deal with a single population taxa block.
- [Association](#) between the gene taxa and population taxa
- Population Tree Block corresponding to model(s) being tested – Instructions for creating these model trees are provided below.

**Calculating observed value of  $s$**  The easiest way to do this is to add a Tree Legend to the gene tree. If you do not have a tree window open showing the reconstructed gene tree, open one using [Taxa&Trees > New Tree Window](#). Add the legend from the Analysis menu ([Analysis > Tree Legend...](#)), and select "s of Slatkin & Maddison". If you have more than one association included in the file, you will be asked to choose which association to use for the calculations. Record the number that appears in the tree legend – this is the observed value of the test statistic.

In our example, we have 20 gene sequences, sampled from 4 populations. The color of the gene sequences corresponds to the population from which they were sampled (note: taxon name colors are only used for aesthetics and are not required for this type of analysis. If you would like to color taxon names, you will need to set up "[Groups](#)" of taxa, which are not the same as associations.). In this example,  $s$  is counting the number of changes (in

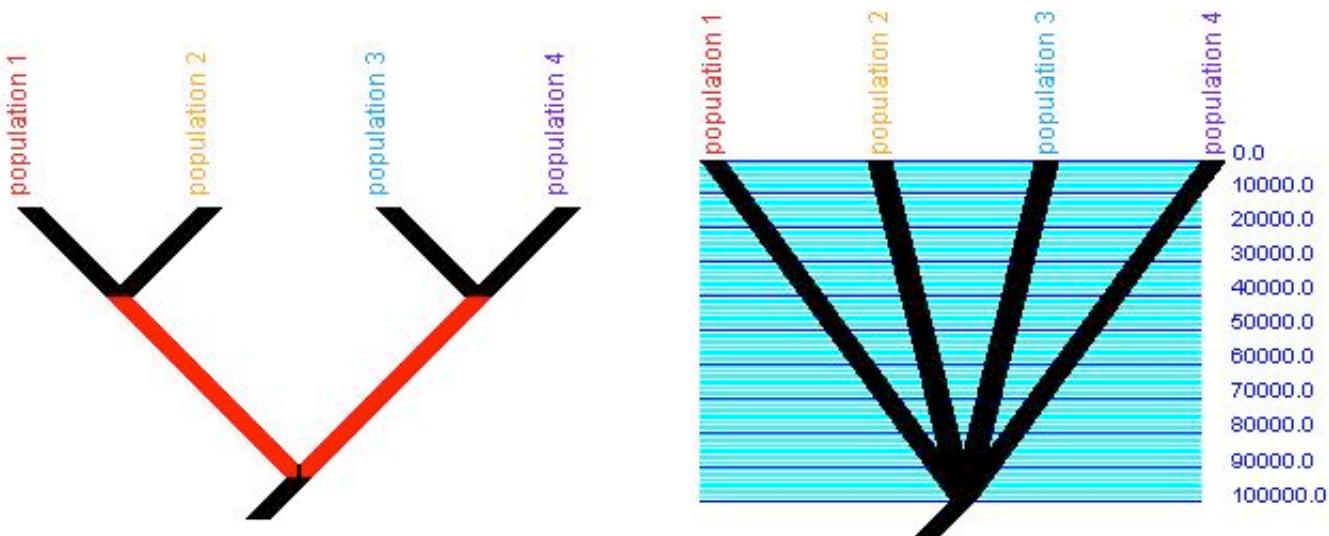
parsimony steps) of a character corresponding to population membership (as indicated by color):



Counting the most parsimonious number of changes leads to  $s = 9$

**Setting up models to be tested** Begin by opening a new tree window, corresponding to the population block of taxa. Select Default Trees as the source of trees. Manipulate this tree to reflect the model you wish to test. In order to preserve any changes you make to the tree, you must store the tree in a Tree Block (Tree > Store Tree As...).

In our example, we would like to test two models; the first is a model of population fragmentation, in which all four populations became isolated from one another, some time in the past, say 100,000 generations ago. Starting with a default symmetrical tree, we use the collapse branch tool (▼) to collapse the branches indicated in red on the tree to the left, to create the tree on the right.



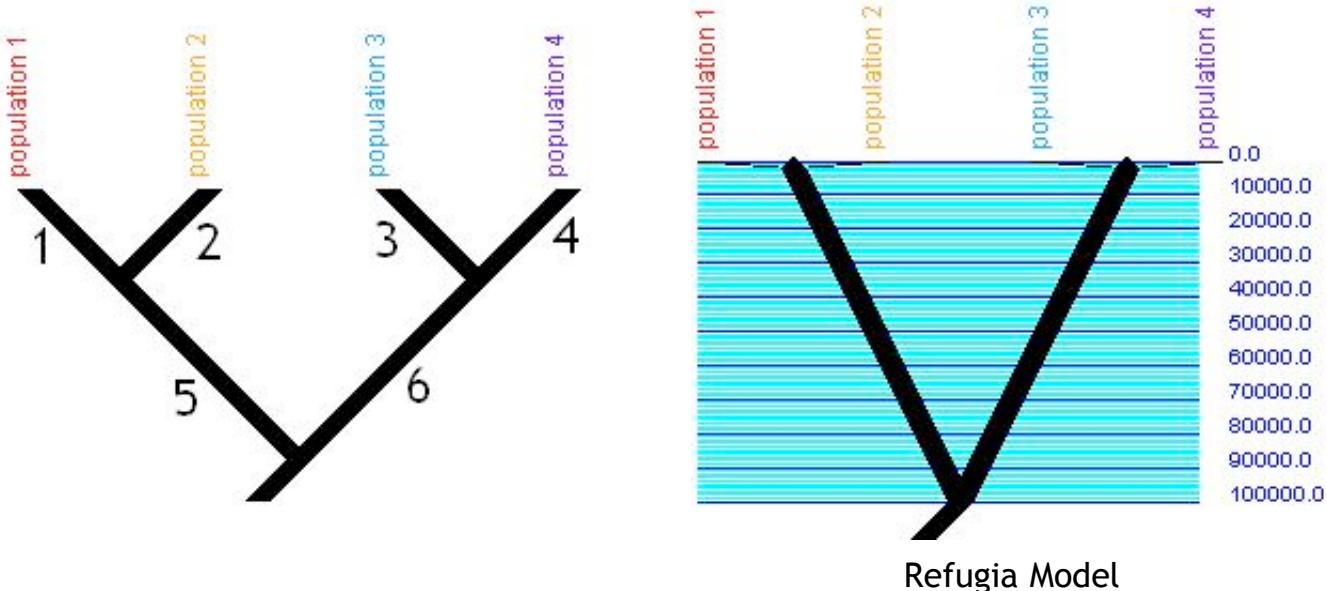
Fragmentation Model

We can set all the branch lengths to equal 100000 generations from the Tree Menu: Tree > Alter/Transform Branch Lengths... > Assign All Branch Lengths... and entering 100000 in the dialog box that appears. Remember to use Tree > Store Tree in order to retain these changes. Unless you have selected Drawing > Branches Proportional to Lengths the tree drawing does not change after you've assigned the branch lengths. You can toggle the branch lengths and scale from the Drawing Menu (Branches Proportional to Lengths and Show Scale, respectively).

For our alternative model, we will test the hypothesis that Populations 1 & 2 and Populations 3 & 4 were isolated in two refugia 100000 generations ago, and only recently split (1000 generations ago) from two into four populations. For this model tree, we manipulate the fragmentation model tree to reflect the alternative model, and store a copy of this tree under another name (using Store Tree As... instead of Store Tree to avoid overwriting the fragmentation model tree; storing this model tree in the same tree block as the fragmentation model tree will make things easier later on). Using the arrow tool (

in the population tree window, we join the branches of populations 1 & 2 and populations 3 & 4 (below, left). To assign branch lengths to individual branches, use the

Adjust Branch Length tool ( on each branch, assigning branches 1-4 a value of 1000 generations (recent split) and branches 5 & 6 a value of 99000 (so the age of the split is 100000 generations ago). Because the branch lengths of branches 1-4 are relatively short, it may be difficult to visualize them when branches are drawn proportional to lengths (below, right), so turning off Branches Proportional to Length may ease manipulation.



We will use the two model trees, fragmentation and refugia, to generate a simulated distribution of the test statistic.

**Simulating distributions of test statistic** To simulate a distribution of Slatkin & Maddison's  $s$ , bring the tree window of the population tree corresponding to the model you wish to test to the front. Select Analysis > New Bar & Line Chart for > Trees. In the "Select Taxa" dialog that opens, select the block of taxa corresponding to the gene sequences. Choose "Simulated Trees" as the source of trees, and "Coalescence Contained within Current Tree" as the Tree Simulator (in older versions of Mesquite, you may have to check the "Show Secondary Choices" box for this tree simulator to be an option). Set the effective population size when prompted, and choose "s of Slatkin & Maddison" as the value to calculate for trees ( $s$  of Slatkin & Maddison may be a Secondary Choice, so you'll have to check the appropriate box for it to appear as an option). You will be prompted to enter the number of trees for the chart, and asked if the current window is appropriate for simulations (you should check to be sure the tree window corresponding to the model you wish to test is the one Mesquite is querying you about). If you have multiple associations in your file, you will be prompted to choose the association for generating gene trees first, then you will be prompted *again* to choose the association for calculating  $s$ . These will often be the same, but certain situations may arise when the association on which the model is based is not the same as the association for which the statistic is calculated.

Mesquite uses a stepwise process to generate the distribution of the test statistic. It repeats the following steps  $N$  times, where  $N$  is the number of trees you asked Mesquite to use for calculations:

- Simulate a gene tree within population gene tree
- Calculate the value of  $s$  for that gene tree and given association
- Record the value of  $s$
- Discard gene tree

After Mesquite has recorded the simulated values, a Chart Window will show a histogram of the simulated distribution. By clicking on the "Text" tab, you can see the numbers used to generate the frequency distribution for the test statistic. These numbers can be copied & pasted into a spreadsheet program to generate graphs outside of Mesquite. Additionally, in

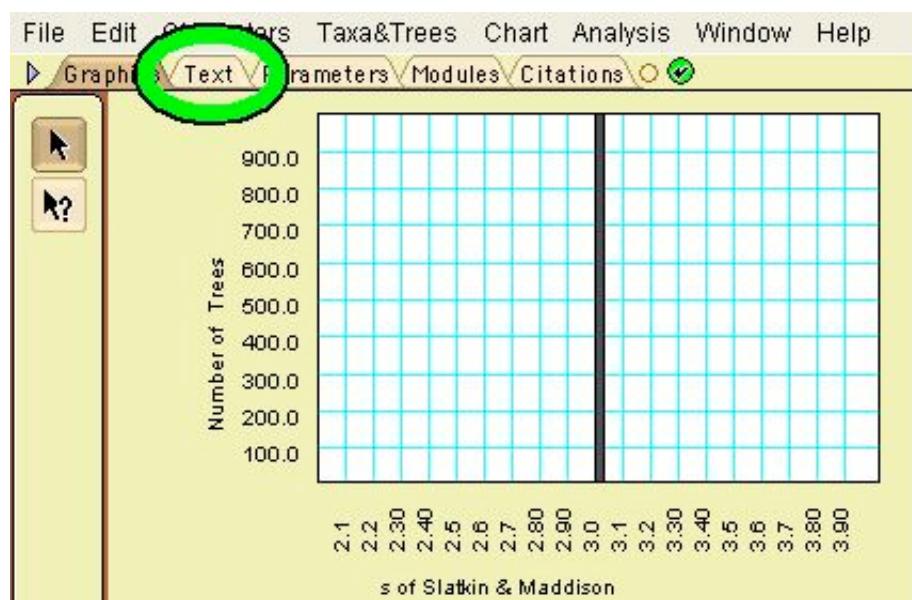
the "Graphics" view, you can ask Mesquite to draw lines on the distribution corresponding to 95% confidence interval. In the Chart Window, select Chart > Analysis > Percentiles..... From here, you can choose upper or lower bounds, as well as the size of the percentiles (e.g. for the 90% confidence interval, enter 0.1 for the Percentile Boundary).

The chart window displaying the distribution of  $s$  is dependent on the Tree Window showing the population tree on which the gene trees were simulated. By default, the Chart Window is set to re-calculate  $s$  if any changes are made to the Tree Window. Thus, if you make changes to the Tree Window (re-arrange branches, change branch lengths, or scroll between trees), the chart will re-simulate gene trees and perform necessary calculations to reflect the distribution of  $s$  corresponding to the tree shown in the Tree Window. The auto-recalculate option can be toggled on/off from the Chart menu, if desired. However, if you are testing more than one model of population structure, it is easiest to:

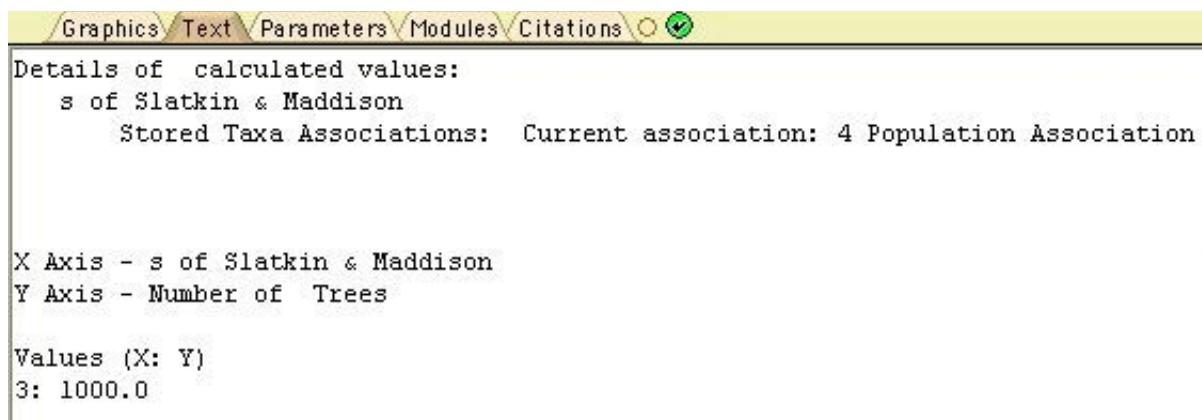
- Generate a distribution for the first model to be tested
- Record the distribution (copy & paste from the "Text" tab into a spreadsheet program)
- In the Tree Window, scroll to the next model to be tested; the distribution will be automatically recalculated, based on the new model
- Repeat for each model being tested, recording the distribution of  $s$  for each model

If the observed value of  $s$  falls outside of the 95% confidence interval for a particular model, that model, and the associated parameter values (branch lengths, population size, etc.) are not supported by the data.

We begin by simulating a distribution of  $s$  for the fragmentation model. From the Tree Window showing the fragmentation model population tree, we select Analysis > New Bar & Line Chart for > Trees, choosing, as prompted, Genes Taxa Block (taxa to calculate value), Simulated Trees (source of trees), Coalescence Contained within Current Tree (tree simulator), 10000 (effective population size),  $s$  of Slatkin & Maddison (Value to calculate), 1000 (trees for the chart), and check to be certain that Mesquite is using the fragmentation model tree. The chart window opens, showing the distribution of simulated values of  $s$ .

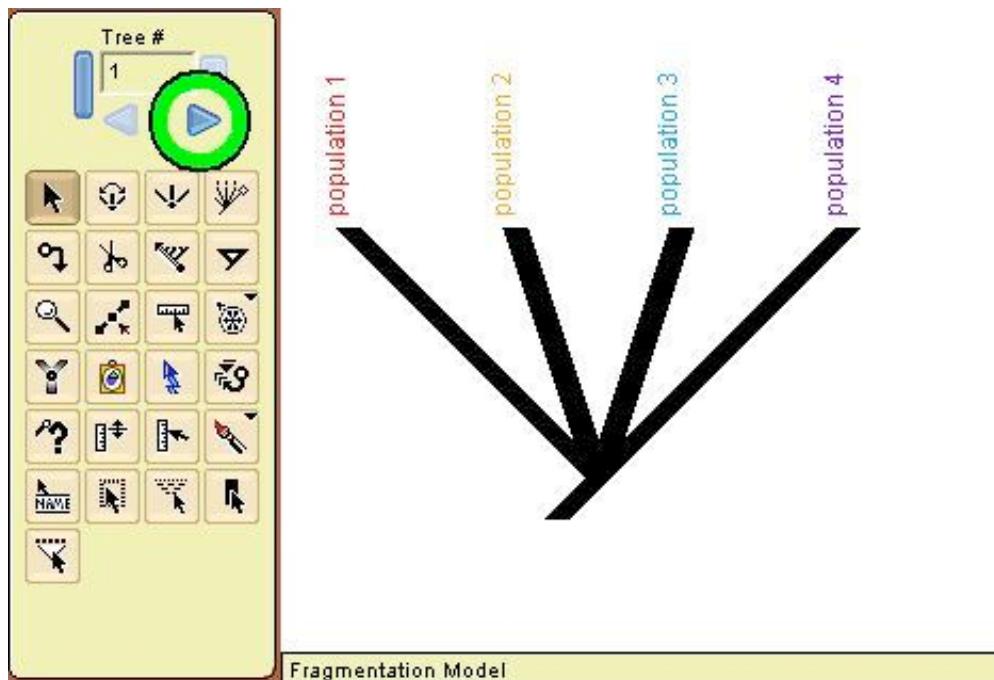


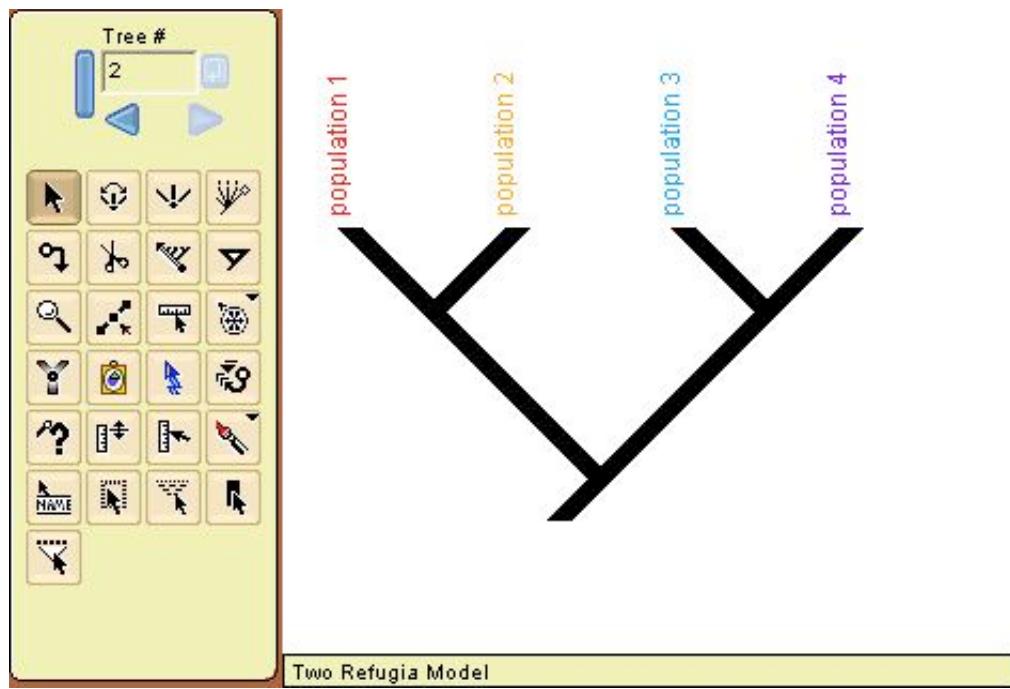
By clicking on the "Text" tab, we can see the details of the simulations:



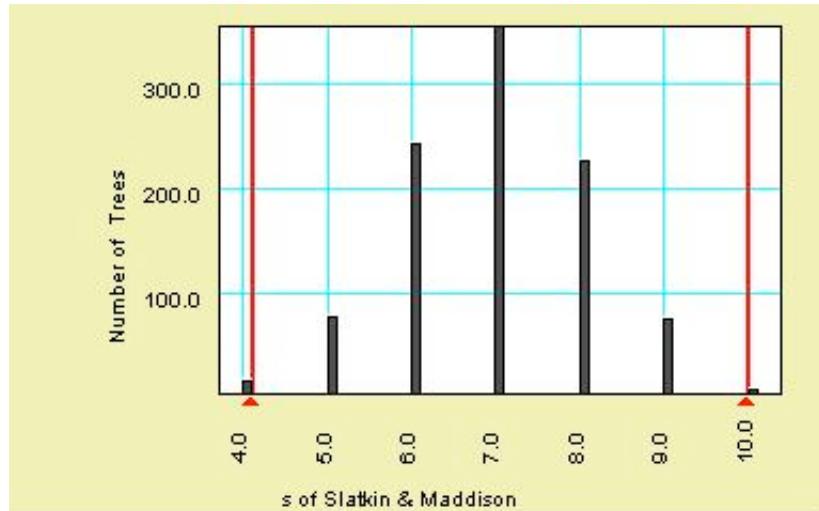
Because our observed value of  $s = 9$  was not within the 95% confidence interval of the simulated distribution, we reject the fragmentation model.

To test the refugia model, we can take advantage of the Chart Window's dependency on the Tree Window. Leaving the Chart Window open, we can scroll to the refugia model tree in the Tree Window.





The Chart Window will be updated with new simulations, based on the refugia model tree (if it does not, you may need to turn on "Auto-recalculate" from the Chart menu in the Chart Window). By selecting Chart > Analysis > Percentiles..., setting the Percentile boundary to 0.05, and checking the boxes for both upper and lower tails, we can see the 95% confidence interval for this distribution:



Because our observed value  $s = 9$  falls within the 95% confidence interval, we do not reject the two refugia model. By clicking on the "Text" tab, the full details of the distribution can be seen:

Graphics Text Parameters Modules Citations

Details of calculated values:  
s of Slatkin & Maddison  
Stored Taxa Associations: Current association: 4 Population Association

X Axis - s of Slatkin & Maddison  
Y Axis - Number of Trees

Values (X: Y)

4:	16.0
5:	77.0
6:	243.0
7:	355.0
8:	227.0
9:	74.0
10:	8.0

From these results, we reject the fragmentation model in favor of the two refugia model. Further tests of deeper and shallower divergence times can provide confidence interval for this model, for given parameter values.

In closing, it is important to consider parameter values ( $N_e$ , divergence times, etc.) and the explanatory power of the tests. In the example above, if we had used an effective population size 100 times larger ( $N_e = 1,000,000$ ), the fragmentation model is *not* rejected (the observed value of  $s = 9$  falls within the 95% confidence interval of the simulated distribution of  $s$ ). We encourage careful consideration of reasonable parameter values in these approaches. Rather than relying on one or a few point estimates of parameter values, it may be most informative to explore parameter space, and define that parameter space that would best explain the given pattern of genetic variation.

# Tree simulations and randomizations

Various packages in Mesquite can simulate or randomize trees, allowing you to build statistical tests based on null distributions or trees generated under some hypothesis.

## Contents

- [Using results of simulations & randomizations](#)
- [Evolutionary Simulations of Trees](#)
  - [Species Trees](#)
  - [Gene Trees](#)
- [Random Trees](#)
- [Randomized Trees](#)

## Using results of simulations & randomizations

The simulated or randomized trees can be used or stored in several ways:

- The trees can be stored into tree blocks in the current file by choosing options in the **Make New Trees Block** from submenu of the Taxa&Trees menu.
- The trees may be used directly, at that moment, in calculations. For example, if you make a Bar & Line Chart for trees, and choose Simulated Trees as your source of trees, the trees will be simulated and used in the chart without being stored in the file.
- A series of many files can be saved, each one with a different replicate of a block of simulated or randomized trees. This is available through the **Save Copies of Tree Blocks** submenu of the Taxa&Trees menu menu

To replicate the results of a simulation or randomization, you can use the **Set Seed** menu item to set the random number seed used. If you are using the same conditions, including the same seed, the simulations and randomizations should be reproducible.

## Evolutionary Simulations of Trees

### Species Trees

We recommend you see the page on [diversification](#) for more features related to species tree evolution.

- **Uniform speciation (Yule)** (trees package) – Generates tree by simple uniform probability speciation (a Yule process). The chance of speciation is equal for all tips. Options: total time depth of tree.
- **Uniform speciation with sampling** (TreeFarm package) – Generates a tree by Yule process, as above, but with to a total number of species greater than in the taxa

block. Extra species are then randomly sampled out, to leave the tree with the appropriate number of species. Barraclough & Nee (2001) discuss how this sampling alters the branch length distribution of the tree.

- **Birth/Death trees** (Diverse package) – Simulates a constant rate of speciation and extinction to generate a tree. For more details see the page on [diversification](#).
- **Trees depending on the evolution of a character** – in the [Diverse](#) package are also some simulators that evolve a tree and character simultaneously; the speciation and extinction rates may depend on the state of the character.

## Gene Trees

- **Coalescent Trees** (coalesce package) - Generates tree by coalescence within a single panmictic population. Options: Effective population size.
- **Coalescence Contained within Current Tree** (coalesce package) - Generates tree by a simple coalescence model of a neutral gene with constant population size, within a current species tree from a Tree window or other tree context. Branch lengths are assigned according to generation of coalescence. The species tree used is a current tree found in a Tree Window or other tree context.

## Random Trees

- **Equiprobable Trees** (trees package)- Generates trees randomly so that each possible labelled tree topology is equally likely.
- **Randomly Resolve Polytomies** (TreeFarm package) - Randomly resolves polytomies in tree. All possible resolutions are equiprobable. Thus, if the tree is a polytomyous bush, the resulting resolved trees will be distributed equivalently to that from the Equiprobable Trees module.

## Randomized Trees

- **Reshuffle Terminal Taxa** (TreeFarm package) - Shuffles (permutes) the taxa among the terminal nodes.
- **Random Branch Moves** (TreeFarm packages) – Performs a specified number of randomly -chosen branch moves.
- **Add Noise to Branch Lengths** (TreeFarm package) - Adds noise to branch lengths of tree. Noise is Normally distributed with variance specified by user. Optionally, this variance is proportional to current branch length. Negative branch lengths are not allowed, and are changed to zero. Options: variance of noise.
- **Rarefy Tree** (TreeFarm package) - Rarefies tree by randomly deleting taxa. Options: how many excluded.
- **Augment Tree Randomly** (TreeFarm package) - Augments tree by random placement of excluded taxa. Options: whether branch lengths are to be considered; whether addition is only to original branches

## References

Barraclough, T.G. & S. Nee 2001. Phylogenetics and speciation. *TRENDS in Ecology and Evolution.* 16:391-399.

---

# Molecular data

Molecular data (DNA or protein sequences) can be edited, manipulated, simulated and analyzed in various ways in Mesquite. Most of the features discussed elsewhere concerning editing and analysis of general categorical data also apply to molecular data; here we focus on features specifically designed for sequence data.

## Contents

- [Editing molecular data](#)
  - [Alter/Transform Tools](#)
  - [Alignment](#)
  - [Finding sequences](#)
  - [Display of sequences](#)
  - [Consensus sequences](#)
- [Fusing molecular matrices](#)
  - [Adding new taxa/sequences](#)
    - [Include Taxa from File](#)
    - [Drag and Drop](#)
  - [Adding genes](#)
    - [Concatenate Matrices](#)
    - [Fused Matrix Export](#)
- [Other tools for managing molecular matrices](#)
  - [Alternative Taxon Names](#)
  - [Translate to protein](#)
- [Interactions with GenBank](#)
- [Simulating DNA sequence evolution](#)
- [Statistics for DNA sequences](#)
- [DNA Distances](#)
- [Statistics for Protein sequences](#)
- [Visualizing tertiary structure](#)
- [Sequence Data within populations](#)
- [Reconstructing ancestral states](#)

		Character Matrix "Mitochondrial DNA"																						
		24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
Taxon \ Character		T	C	A	T	A	A	T	C	G	C	C	A	C	G	G	G	C	T	T	A	C	A	
1	Homo sapiens	T	C	A	T	A	A	T	C	G	C	C	A	C	G	G	G	C	T	T	A	C	A	
2	Pan	T	C	A	T	A	A	T	C	G	C	C	A	C	G	G	A	C	T	T	A	C	A	
3	Corilla	T	T	A	T	A	T	T	G	C	C	C	A	C	G	G	A	C	T	T	A	C	A	
4	Pongo	T	C	A	T	G	A	T	T	G	C	C	C	A	T	G	G	A	C	T	C	A	C	
5	Hylobates	T	C	A	T	A	T	C	G	C	C	C	A	C	G	G	A	C	T	A	A	C	C	
6	Macaca fuscata	T	T	A	T	G	A	T	C	G	C	T	C	A	C	G	G	A	C	T	C	A	C	
7	M. mulatta	T	C	A	T	G	A	T	T	G	C	T	C	A	C	G	G	A	C	T	C	A	C	
8	M. fascicularis	T	T	A	T	A	T	C	G	C	C	C	A	C	G	G	G	C	T	C	A	C	C	
9	M. sylvanus	T	T	A	T	A	G	T	T	G	C	C	A	T	G	G	G	A	T	C	A	T	C	

## Editing molecular data

Molecular data can be imported from files of NBRF, FASTA, PHYLIP, CLUSTAL, and simple table format. It can also be exported to some of these formats.

The [Character Matrix Editor](#) can be used to edit a molecular sequence matrix. Standard ambiguity codes are allowed.

## Alter/Transform Tools

The following can be applied to all or the selected portions of a molecular sequence matrix in the Character Matrix Editor. These are available under the Alter/Transform submenu of the Matrix menu (some of these are available under "Other Choices"):

- **Reverse Complement** (DNA matrix only) – reverses the order of contiguously selected blocks of sequence and complements the sequence
- **Collapse Gaps** – collapses gaps in the selected block by pushing all sites to the left, to yield unaligned sequences
- **Remove Gaps-Only Characters** – Removes from the matrix all characters that consist of nothing but gaps.
- **Shift Other to Match** – Shifts other sequences to match a region in a selected sequences; described in the manual for the [Align package](#).
- **Convert Uncertainties to Polymorphisms** – Converts all uncertainties in the selection to polymorphisms. For example, for a nucleotide sequence, a Y (C or T) will be converted into "C and T".
- **Convert Polymorphisms to Uncertainties** – Converts all polymorphisms in the selection to uncertainty. For example, for a nucleotide sequence, a cell that has both C and T will be converted into "C or T", i.e., Y.
- **Arbitrarily Resolve State Ambiguity** – Any cell that has partial uncertainty in the state (e.g., "Y", which is C or T) will be resolved into one of its states, chosen randomly. Thus, a Y will be converted to a C with 50% probability, and to a T with 50%.
- **Nucleotide complement** (DNA matrix only) – enters the complementary sequence into the selected cells
- **Trim Terminal Gap Characters** – deletes characters at edges of matrix that are gaps-only.

Other options may appear; see the page on [characters](#) for standard choices in this submenu. You can also apply the other editing tools described for [character matrices](#).

The view of the matrix can be adjusted in various ways. Cells can be colored according to the state at the site (Color Cells submenu, **Character State**) or according to a value like the GC bias (Color Cells submenu, **Cell Value**; can request this coloring to use a moving window). Examples of this are shown below. The Display submenu of the Matrix menu contains other options such as a **Bird's eye view** which makes the cells narrow to show more of the sequences.

**Copy Sequence** (at bottom of Edit menu) copies the selected cells of the matrix into the computer's clipboard as a sequence. That is, whereas the standard Copy would place into the clipboard selected pieces of the matrix in tab-delimited text format (e.g., if the sequence AATCA is selected, "A-tab-A-tab-T-tab-C-tab-A" would be copied), this modified Copy Sequence command does not include tabs (thus, "AATCA" would be copied). This style of copying is useful when interacting with programs like [Sequencher \(TM\)](#). For instance, if you want to find a piece of sequence in a matrix in Mesquite within a chromatogram viewer of Sequencher, do the following: select the sequence in Mesquite, choose Copy

Sequence, then go to Sequencher, select Find Bases, and paste the sequence as the search string.

## Alignment

The [Align](#) package contains utilities for sequence alignment. These include manual alignment tools (for shifting blocks of sequence, for example), and automated tools (e.g., sending a region of the matrix to ClustalW to align, or a pairwise alignment tool in the editor that will align one sequence to another). See the [Align](#) manual for more details.

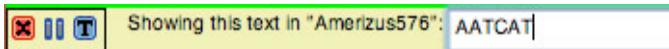
## Finding Sequences

You have several options to find sequences in a matrix.

First, you can use the Search Data  entry at the top of the Character Data Editor to enter a sequence and hit return. (If the entry indicates Search Features instead, touch on the brown & or / until it shows a violet  and is labeled Search Data.) The results of the search will be displayed in the Search results window.

Second, pieces of sequences can be found using the Find Sequence and Find All Sequences submenus of the Edit menu. The current options are:

- **Matching Sequence:** This finds sequences matching a target sequence the user enters. The search allows a certain number of mismatches. Optionally, it can search for the reverse, complement and reverse complement of the target sequence.
- **Maintain Target Match:** This highlights and maintains highlighted the first occurrence of a given sequence in a given taxon. First, you are asked which taxon to search in. Then, it displays a panel like this:



underneath the matrix. The first button (red X) is to close the panel; the second pauses the search; the third allows you to select another taxon as your focus. If you type a sequence into the text area, the matching sequence (if any) will be highlighted in the matrix. Mesquite is constantly monitoring this text, and so you don't need to give any command to find again if you change the text. This is useful if working with a program like Sequencher. If you see a stretch of sequence while viewing chromatograms that you'd like to find in the matrix in Mesquite, type in the sequence into the text box and you will quickly be taken to it in the taxon.

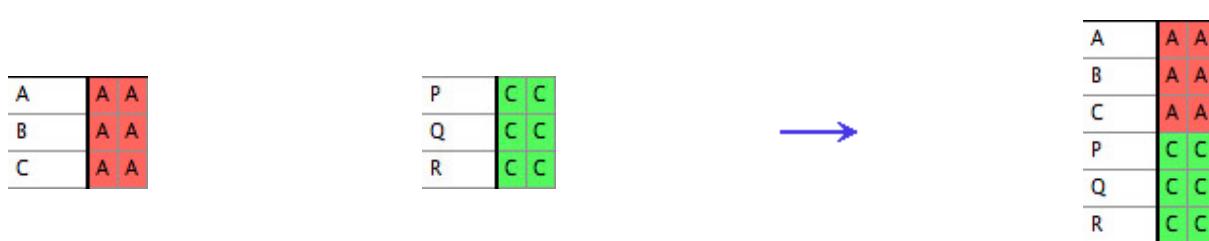
- **Maintain Clipboard Match:** This is similar to Maintain Target Match, except that it obtains the search string not from the text area but from the clipboard. If the clipboard changes, the function will automatically find the sequence again in the matrix. This is useful if working with a program like Sequencher. If you turn on Maintain Clipboard Match, then you can copy stretches of a sequence within Sequencher, and Mesquite will automatically highlight it, without your having to return to Mesquite or give any other command to it. (Mesquite is constantly monitoring the clipboard to see if it changes).

## Fusing Molecular matrices

Often you have sequences in different matrices, and you need to fuse them into a single matrix for analyses. You may be adding new sequences from an existing gene, or you may be adding new genes for existing taxa.

## Adding new taxa/sequences

If you want to take two matrices and concatenate them vertically, i.e. add new taxa to an existing set of gene sequences, then you can do it either from a menu or using Drag and Drop.



- **Include Taxa from File:** In the Taxa&Trees menu, this function reads in an external file and attempts to concatenate taxa from the file onto taxa blocks of the current file, and also to add their data to existing matrices. This is useful to add new gene sequences to an existing file. File formats that can be incorporated are NEXUS, NBRF, and FASTA. This incorporation will NOT preserve the associated information of the external NEXUS file such as footnotes for the taxa and characters.
- **Drag and Drop:** If you drag FASTA or NBRF files onto a Character Matrix Editor showing sequence data, the sequences in these files will be added to the matrix as new taxa.

## Adding genes

If you want to take two matrices and concatenate them horizontally, i.e. add new genes to an existing set of taxa, then how you do this depends on whether your taxa have the exact same names in the two matrices, or not.



- **Concatenate Matrices:** If the corresponding taxa have the exact same names in the two matrices (as shown in the example above – taxon B has the exact same name in both matrices, as does taxon C), then you can use the [\(Character Matrix Editor\)Matrix>Utilities>Concatenate Matrices](#) command to fuse the matrices. If both matrices belong to the same taxon block, then simply use the command to concatenate. On the other hand, if the matrices have overlapping but not identical taxon blocks, then you must bring both matrices along with their respective taxon blocks into memory. Let's assume you have two separate files, one for Gene 1 and one for Gene 2. Copy the first file (so as to save an original copy of gene 1 on its own), and give the copy a name like "Fused.nex". Then, open it in Mesquite. Next, use Link File in the edit menu to read in the second file. Now, go to the Character Matrix Editor for the Gene 1 Matrix, and select Concatenate Matrices from the Utilities submenu. Later you can choose Close File from the File menu to close the linked file for gene 2, and save your fused file.
- **Fused Matrix Export (NEXUS):** If the corresponding taxa have different names in the two matrices, then this is the easiest method to fuse, although it can also be useful when taxon names are the same. It is a special file exporter that can fuse matrices of different genes as it exports. It is available by choosing Export in the File menu. If you have a **single taxa block** with multiple matrices attached to it (e.g., one for each gene), then this exporter will write a single combined matrix with the sequences of different genes concatenated for each taxon. It will also write

CHARPARTITION and CHARSETS to record which section of the fused matrix corresponds to which source matrix. It will also write a MrBayes block with that program's partitioning commands. If the different matrices are of mixed type (DNA, protein, non-molecular categorical) then you have a choice between reducing them to simple categorical data or maintaining a mixed matrix (suitable for MrBayes). Often however, you will have slightly different names for the sequences of different genes, and these may therefore exist in the file represented by **different taxa blocks**. For instance, you may have a taxa block with taxa "A1", "B1", and "C1" with corresponding matrix for COI sequences (gene 1) from species A, B and C, and another taxa block with taxa "B2", "C2", and "D2" for 28S sequences (gene 2) from the same species. You want to fuse them as follows:

A1	A A
B1	A A
C1	A A

B2	C C
C2	C C
D2	C C

A	A A	-	-
B	A A	C C	
C	A A	C C	
D	- -	C C	

The Fused Matrix exporter permits you to export these into a single matrix as long as you have indicated how the sequences correspond to one another. To do this, we suggest you create a new taxa block representing the species or specimens. In this example, create a taxa block "Species" with taxa A, B and C. This will be the "master block of taxa" that will organize the export.

(Alternatively, you could choose one of the genes' taxa blocks as the master block.) [Set up a Taxa Association](#) between the master block of taxa and each of the other blocks of taxa. With the first Taxa Association between species and Gene 1 indicate that "A1" belongs with species A, "B1" belongs with species B, and "C1" belongs with species C. Set up the species-Gene 2 association similarly. The two taxa association will look like this in the List of Taxa window for the master taxa block:

Taxon	Default	Group	Associates	Associates
1 A	1	?	A1	-
2 B	2	?	B1	B2
3 C	3	?	C1	C2
4 D	4	?	-	D2

Associates  
corresp. Gene 1  
  
[ ] [+] [-]

- A1
- B1
- C1

Associates  
corresp. Gene 2  
  
[ ] [+] [-]

- B2
- C2
- D2

Then when you choose Fused Matrix Export, choose Species as your master taxa. The exporter will then find all of the data corresponding to each species, either under the species taxon itself or under one of the linked taxa indicated by the Taxa Association, and compose a fused matrix. If a single master taxon has more than one corresponding taxa in one of the other matrices, the data are merged using the same rules as for [Merge Taxa](#).

## Other Tools for Managing Molecular matrices

Managing sequences in different matrices, especially if from different genes, can be difficult. Several functions assist in this. These features are not restricted to molecular data, but we anticipate most of their use will be with sequences.

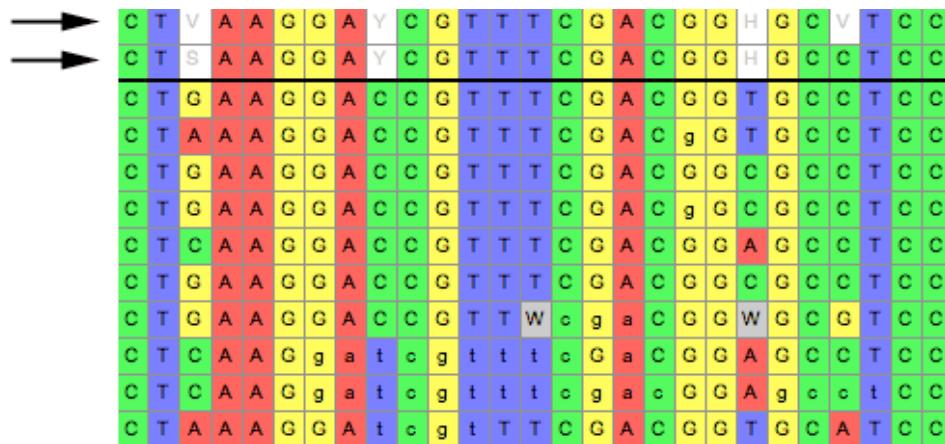
- **Alternative Taxon Names:** This feature allows you to have several alternative naming schemes for taxa, e.g. one set of names that includes your specimen codes, another set that uses formal taxonomic names (for publication), another set whose names are condensed for exporting the file to programs like CLUSTAL that can handle only short names, and so on. For details see [here](#).
- **Translation to protein:** a new matrix of amino acids, derived by translation from existing protein-coding DNA matrix, can be formed by choosing [Characters>Make New Matrix from>Translate DNA to Protein](#).

## Display of Sequences

Protein-coding sequences can be colored by the amino acid into which a triplet would be translated (under the genetic code for that triplet) by choosing Matrix>Color Cells>Color Nucleotide by Amino Acid.

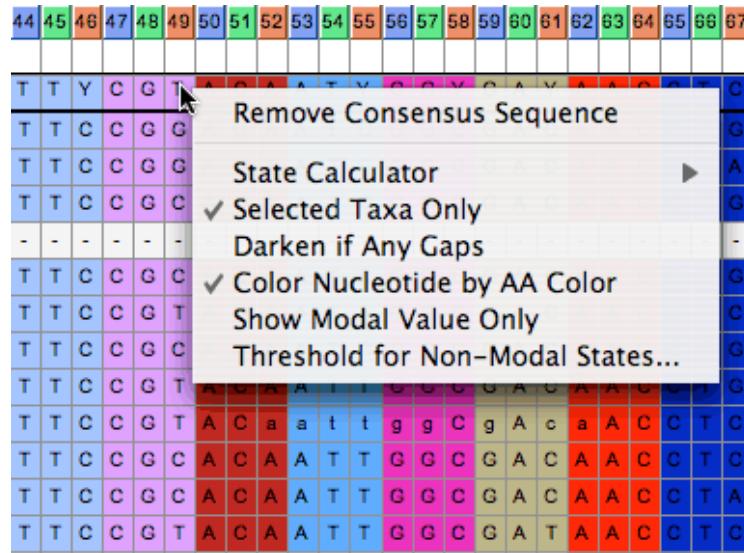
## Consensus Sequences

Consensus Sequences can be displayed above the character matrix by choosing Matrix>Add Info Strip>Consensus Sequence Strip, as indicated below by the arrows:



In the above examples, two consensus sequences strips are displayed, with slightly different options.

Options are available by touching on the consensus sequence:



## Genetic Codes

The genetic code for sequence data can be specified in Mesquite's List of Characters window (by choosing List of Characters, and then Columns>Current Genetic Codes, or with a data matrix frontmost,

Matrix>Genetic Codes...). Genetic codes are assigned to individual characters (thus allowing one to have a mixed matrix of mitochondrial and nuclear data, for example). To assign a genetic code, choose select the characters, and use the popup menu of the title of the "Genetic Code" column in the List of Characters window.

The genetic code affects, among other things, the Translate DNA to Protein command, as well as the coloring of nucleotide sequences if Color Nucleotide by Amino Acid is chosen.

## Interactions with GenBank

The following features allow one to interact with GenBank. They require that you have an Internet connection.

- **BLAST in Web Browser:** Select a sequence or portion thereof in the data matrix. Choose Matrix>Search>BLAST in Web Browser, and Mesquite will send a BLAST request to GenBank to search for matching sequences. Your default web browser should open and take you to the BLAST page.
- **Top BLAST Matches:** Select a sequence or portion thereof in the data matrix. Choose Matrix>Search>Top BLAST Matches, and you will be presented with a dialog box. In this you can choose:
  - maximum number of matches to be sought by Mesquite
  - whether the report of results should be saved to a text file
  - whether details of the taxonomic lineage of the sequences should be reported
  - whether to import the top matches into the current matrix
  - the maximum time in seconds to wait for the BLAST search to complete.
- **Fetch & Add GenBank Sequences:** This option, in Matrix>Utilities> Fetch & Add GenBank Sequences, allows one to enter a comma-delimited list of GenBank accession numbers, and Mesquite will acquire these from GenBank and import them into the current matrix.

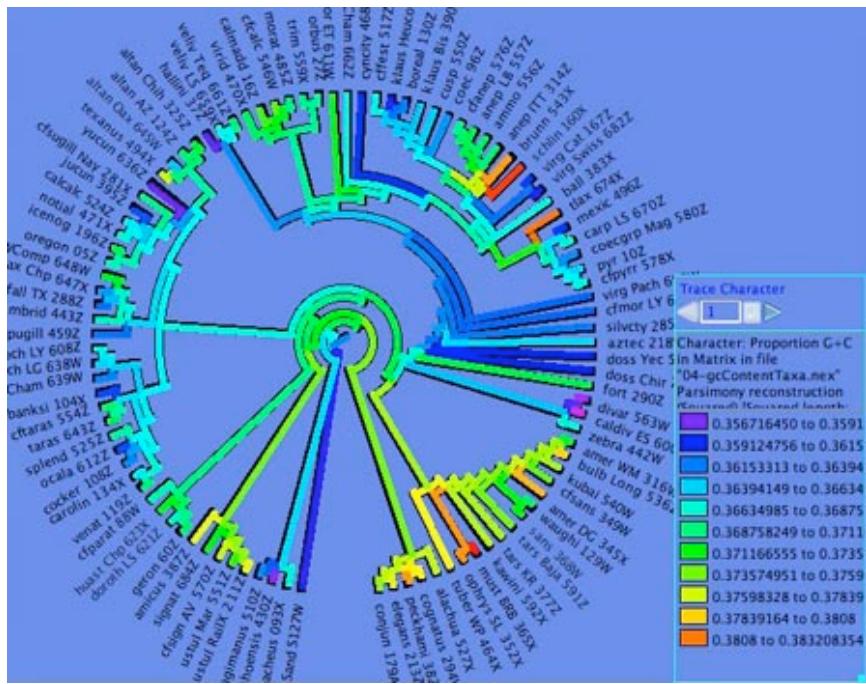
## Simulating DNA sequence evolution

DNA sequence evolution can be simulated to build statistical tests, for instance via parametric bootstrapping. See the page on [simulating DNA sequences](#).

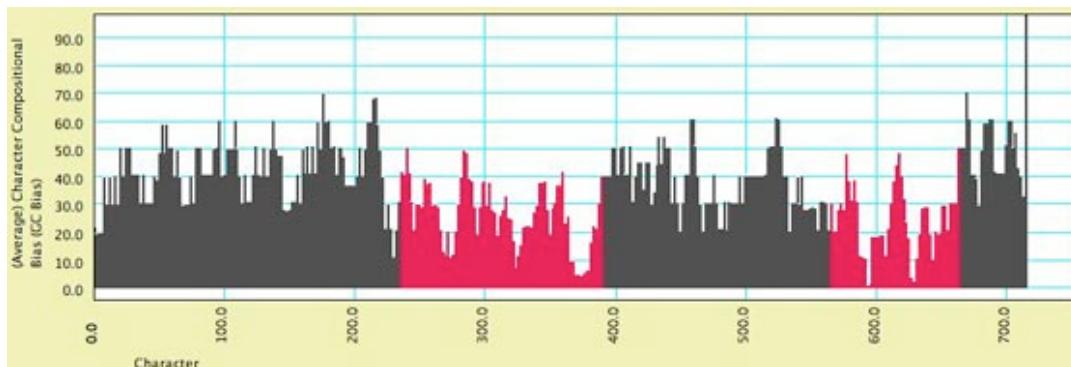
## Statistics for DNA sequences

Calculations for categorical characters in general can be applied to DNA sequences. For example, [Parsimony calculations](#) can be made for DNA sequences, as can basic descriptive statistics such as the percent of a sequence or character that is missing data or gaps. In addition, there are several modules specifically designed for DNA data, illustrated by examples in Mesquite\_Folder/examples/Molecular. These calculate compositional bias:

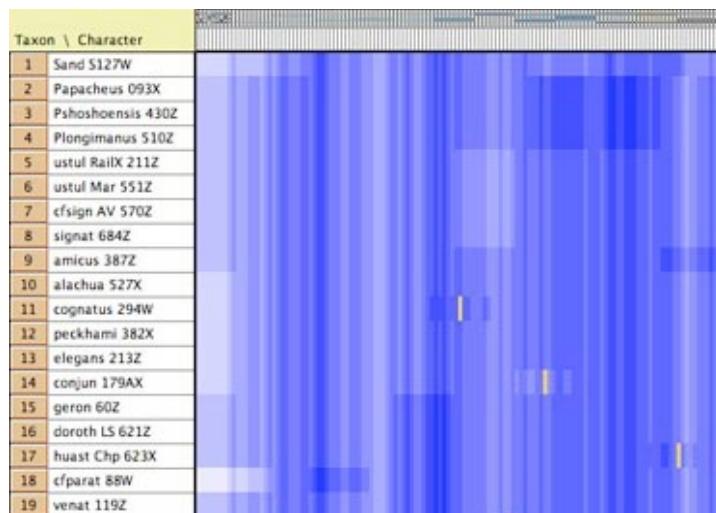
- **ACGT Compositional Bias** – This module supplies the compositional bias of taxa, measured over the taxon's sequence. The bias is treated as a continuous character, and thus can be used wherever characters are used, as for instance in the reconstruction of the evolution of compositional bias as shown in the image below. It can return either the proportion G+C, or separately A, C, G, and T proportions.



- **Character Compositional Bias** – This module supplies the compositional bias for characters. It calculates the percent of taxa with particular nucleotides (GC bias, or individual frequency of A, C, G or T) for a character. The image below shows a moving window analysis of compositional bias along a sequence; the instructions for generating the chart are given [here](#).



- **GC bias coloring of matrices** – The cells of the Character Matrix Editor may be colored according to a moving window of GC bias along the sequence, as shown below, by selecting Matrix>Color Cells>Color By Cell Value, then once shown the colors can be smoothed by a moving window analysis by selecting Matrix>Moving Window (for colors).



## DNA Distances

Mesquite supports several distances for DNA data:

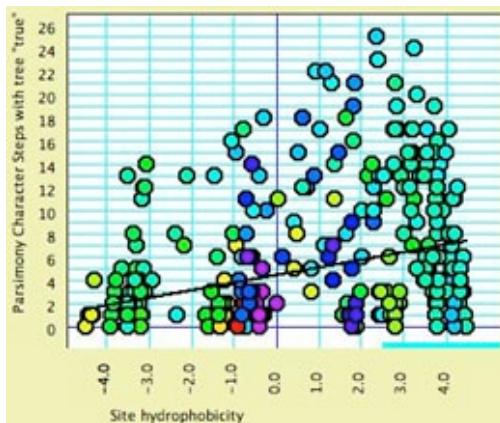
- Uncorrected (P) distances
- Jukes-Cantor
- Kimura 2-parameter
- F81 (Felsenstein, 1981)
- F84 (Felsenstein, 1984)

There are several options available (in the Distance Parameters submenu) for dealing with ambiguous bases and gaps:

- **Count Sites with Gap in Pair:** If this is chosen, then when the distance between two sequences is calculated, then sites that contain a gap in one of the sequences but not the other will be by default included in the calculation. If this option is unchecked, then any site at which either sequence is a gap will be excluded.
- **Estimate Ambiguity Differences:** If this is chosen, then sites in which one member of the pair has an ambiguity have their contribution to the distance estimated based upon the base patterns at non-ambiguous sites.
- **Base Frequency on Entire Matrix:** If this is chosen, then those distances that use as parameters the frequencies of bases (A, C, G, and T) will calculate those frequencies based on all sequences; if it is not chosen, then the frequencies are calculated based upon just the two sequences being compared at any one time.

## Statistics for Protein Data

- **Site hydrophobicity** – This module supplies the average amino acid hydrophobicity, averaged across taxa, for each site. It can be used in charts, for instance to see the relationship between a phylogenetic statistic for the site (character) and its average hydrophobicity. This [chart](#), for example, shows parsimony character steps as a function of hydrophobicity:

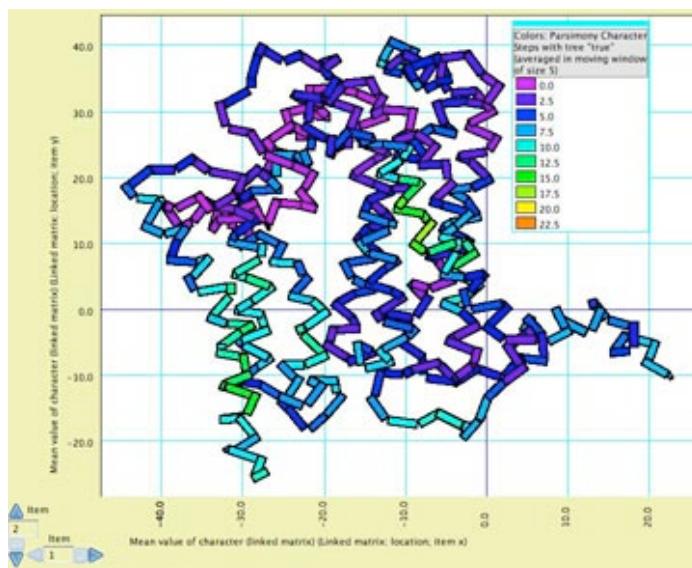


- **Amino Acid hydrophobicity** – The cells of the Character Matrix Editor may be colored according to a moving window of hydrophobicity along the sequence, as shown below, by selecting Matrix>Color Cells>Color By Cell Value, then once shown the colors can be smoothed by a moving window analysis by selecting Matrix>Moving Window (for colors).

Taxon \ Character	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
1 Lamprey	-SHQ P S I I R K T H P L L S L G N S M L V D L P S P A N I S A W W N F G S L L S L C L I I L Q I I T G L I L A M H Y T A N T E L A F S S V M H I C R
2 Alligator	-M T H Q L R K S H P I I K L I N R S L I D L P T P S N I S A W W N F G S L L G L C L I I Q I I T G F F L M M H F S S D T L A F S S V S Y T S R
3 crow	-M G L N L R K N H P L L K I I N N S L I D L P T P S N I S A W W N F G S L L G L C L I I M Q I I T G L L L A M H Y T A D T S L A F A S V A H M C R
4 chicken	-M A P N I R K S H P L L K M I N N S L I D L P A P S N I S A W W N F G S L L A V C L M T Q I I T G L L L A M H Y T A D T S L A F S S V A H T C R
5 ostrich	-M A P N I R K S H P L L K I I N N S L I D L P S P S N I S A W W N F G S L L G I C L I I T Q I I T G L L L A M H Y T A D T T L A F S S V A H T C R
6 Chrysem	-M T M N H R K T H P L T K I I N N S F I D L P S P S N I S A W W N F G S L L G T C L I I L Q I I T G I F L A M H Y S P D I S L A F S S V A H I T R
7 Pelomed	-M G T L H L K Q N P L L K I I T N K S L I N L P S P S N I S A W W N F G S L L G M C L I I L Q I I T G I F L A M H Y T P N I T T A F S S V A H I T R
8 Blue Whale	-M T N I R K T H P L M K I I N D A F I D L P T P S N I S S W W W N F G S L L G L C L I I V Q I I T G L F L A M H Y T P D T M T A F S S V T H I C R
9 Fin whale	-M T N I R K T H P L M K I V N D A F V D L P T P S N I S S W W W N F G S L L G L C L I I M Q I I T G L F L A M H Y T P D T T T A F S S V T H I C R
10 Hippo	-M T N I R K S H P L M K I I N D A F I D L P A P S N I S S W W W N F G S L L G V C L I I L Q I I T G L F L A M H Y T P D T L T A F S S V T H I C R
11 Pig	-M T N I R K S H P L M K I I N N A F I D L P A P S N I S S W W W N F G S L L G I C L I I L Q I I T G L F L A M H Y T S D T T T A F S S V T H I C R
12 sheep	-M I N I R K T H P L M K I V N N A F I D L P A P S N I S S W W W N F G S L L G I C L I I L Q I I T G L F L A M H Y T P D T T T A F S S V T H I C R
13 cow	-M T N I R K S H P L M K I V N N A F I D L P A P S N I S S W W W N F G S L L G I C L I I L Q I I T G L F L A M H Y T S D T T T A F S S V T H I C R
14 white rhino	-M T N I R K S H P L I K I I N H S F I D L P T P S N I S A W W N F G S L L G I C L I I L Q I I T G L F L A M H Y T P D T M T A F S S V A H I C R
15 Black Rhino	-M T N I R K S H P L V K I I N H S F I D L P T P S N I S S W W W N F G S L L G I C L I I L Q I I T G L F L A M H Y T P D T T T A F S S V T H I C R

## Visualizing tertiary structure

Although there are not yet dedicated windows for visualizing phylogenetic statistics in the context of molecular structure, features have been added to the Scattergram chart to allow it to be adapted for this purpose. For instance, in this image cytochrome B is shown, with the amino acids colored according to a simple phylogenetic statistic: the number of parsimony steps on a phylogeny. The colors are smoothed by a moving window, and show that several coils of the molecule, a few at the left and one deep at the right, evolve more rapidly than others. This example is illustrated in the data file at Mesquite\_Folder/examples/Molecular/06-cytochromeB.nex



To build such a chart, begin with a file with a matrix of protein sequences. The procedure is also described in the example files 08-cytochromeBlinked.nex and 09-cytochromeBscatter.nex.

- Select **New Linked Matrix** from the Characters menu. When a matrix is made to be linked to a second matrix, the two matrices are constrained to have the same number of characters.
- Indicate that you want the linked matrix to be a **Continuous** matrix, and link it to your protein matrix. Then, turn it into a three dimensional matrix (Taxa X Characters X Coordinates [x, y and z]) by using **Add Item** and **Rename Item** in the Utilities submenu of the Matrix menu of the Character Matrix Editor. The x,y,z coordinates could be added for all taxa if known, but otherwise only one taxon needs to be filled out (because we will use the average x,y,z coordinates for the amino acids).
- Once the linked matrix of xyz amino acid positions is entered, select Analysis>New Scattergram for> Characters. Indicate you want the scattergram to be for **Stored Characters**, and indicate **Same** value for the two axes. In the dialog box "Values for axes", choose **Mean Value of Character (Linked Matrix)**. In response to "Use characters from which matrix? (for Character Source)" choose the protein sequence matrix as the matrix to be used. This will plot the sites (amino acids, characters) in their correct places, but as a series of round spots.
- To change the appearance of the plot, select **Join the Dots** in the Special Effects submenu of the Scattergram menu. Then select **Thick Joints**, deselect **Show Dots**, deselect **Join First to Last**, and set the marker size larger (e.g., 8). This will result in a plot as shown above, but without the colors.
- Next, choose **Color by Third Value** from the Colors menu and choose the value by which to color the amino acids. For parsimony steps, for instance, choose **Character Value with current tree**.
- Finally, to use a moving window to smooth the colors, select **Moving Window for Colors** from the Colors menu and indicate the window size (e.g., 5).

## Sequence data within populations

See the page on [population genetics](#).

## Reconstructing ancestral states

Ancestral states of molecular characters can be reconstructed as described in the page on [reconstructing ancestral states](#). Likelihood methods are not yet available for molecular characters.

---

# Continuous Characters

Continuous characters (e.g., with values 1.21, 5.68, and so on) can be edited, manipulated, simulated and analyzed in various ways in Mesquite. Below is a brief outline of these features, some of which come from the standard packages of Mesquite, others of which come from the built-in Rhetenor package (by Dyreson and Maddison) and the separately-available [PDAP](#) package (by Midford, Garland & Maddison). Many of these calculations are more thoroughly illustrated in the example files under [Mesquite Folder/examples/Basic Examples/continuous/](#), and under [Mesquite Folder/examples/Multivariate Continuous/](#)

## Contents

- [Editing continuous data](#)
- [Reconstructing ancestral states](#)
- [Plotting trees](#)
- [Simulating character evolution](#)
- [Ordinations](#)
- [Tree reconstruction](#)
- [Felsenstein's independent contrasts](#)
- [Geometric morphometrics](#)

## Editing continuous data

Continuous data can be imported from tab-delimited tables in ASCII text files, or entered into the spreadsheet editor (Character Matrix Editor). Values entered can be negative or positive, and include exponential notation (e.g., "1.3e-6").

A continuous data matrix can have an extra dimension, in that the entry for each cell of the matrix (character state in a taxon) can have more than one number. These separate numbers are called "items", and thus a character matrix can be described as having three dimensions, characters X taxa X items. The first item could be the mean; the second the variance. Or, there could be 3 items, x, y and z, representing coordinates of a landmark in space. To manage items, use the Utilities submenu of the Matrix menu.

The following can be applied to all or the selected portions of a continuous matrix in the Character Matrix Editor. These are available under the Alter/Transform submenu of the Matrix menu:

- Fill – fills the cells with the current "paint" state
- Standardize – transforms the characters to have mean 0 and variance 1.
- Random fill – fills the cells with randomly generated states, with given mean and

variance.

- Add random noise – adds random noise to the entries.
- Add constant – adds a specified constant to all entries.
- Multiply constant – multiplies all entries by a specified constant.

Other options may appear. You can also apply the other editing tools described for [character matrices](#).

## Reconstructing ancestral states

Ancestral states of continuous characters can be reconstructed as described in the page on [reconstructing ancestral states](#).

## Plotting trees

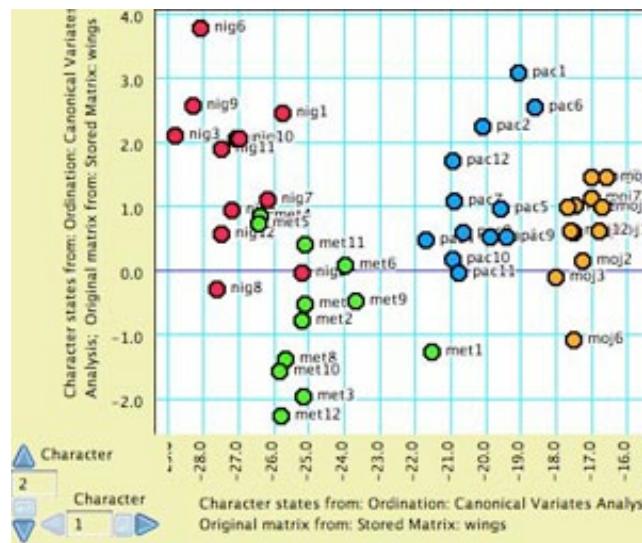
Trees can be mapped or plotted into a character space as described in the page on [processes of character evolution](#).

## Simulating character evolution

Evolution of continuous characters can be simulated by selecting Simulated Characters or Simulated Matrices, and choosing Evolve Continuous Characters. You will get to choose a model, which will be used to simulate evolution on the tree. There is one default model, a Brownian motion model with rate parameter of 1.0. You can create alternative models (e.g. other Brownian motion models) by selecting New Character Model in the Characters menu.

## Ordinations

Where matrices of continuous characters are used, for instance in plotting trees or in Taxa Scattergrams, it is possible to instead use characters representing the modified axes obtained by ordinations such as Principal Components Analysis using modules in the built-in Rhetenor package. For instance, the following Taxa Scattergram shows the results of a Canonical Variates Analysis:



How to set up this plot is explained on the page on [Charts](#).

To use ordinations, simply select Characters from Ordinations or Matrices from Ordinations wherever you might otherwise select Stored Characters or Stored Matrices. There are several options for ordinations:

- Principal Components Analysis
- Canonical Variates analysis – This requires a Taxa Partition to exist to indicate groups of taxa.
- Among-group PCA
- Within-group PCA
- Evolutionary PCA (similar to PCA but tree-based)

The "Multivariate Continuous" example files illustrate the use of these methods.

## Tree reconstruction

The tree search facility (available under [Trees&Taxa>Make New Trees Block from>Tree Search](#)) allows one to search for trees minimizing treelength as calculated by linear or squared change parsimony for continuous characters. It should be noted, however, that the current Tree Search facilities in Mesquite do not adjust branch lengths. The squared change parsimony algorithms by default weight by branch length. Thus, the search is done effectively under the constraint that all trees have branch lengths of 1.0.

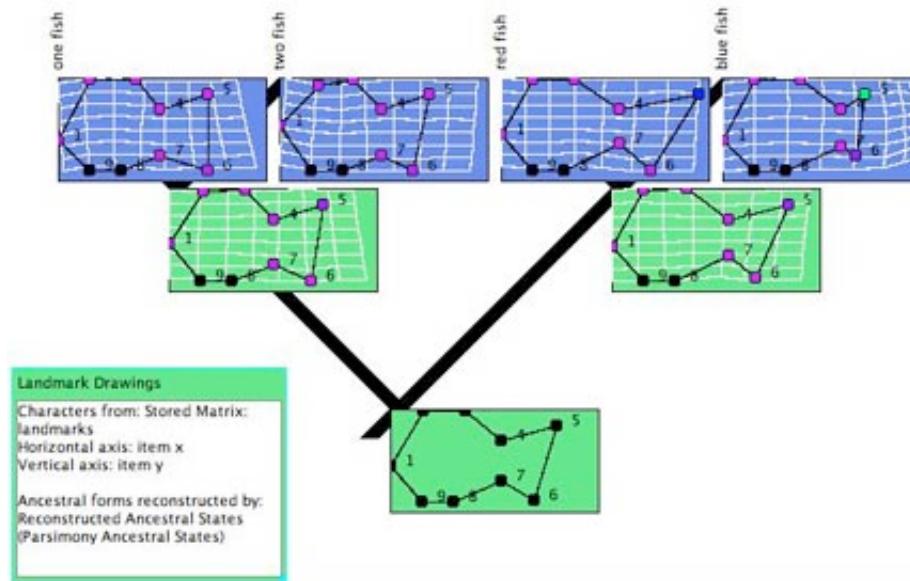
## Felsenstein's independent contrasts

Analyses of character correlations can be done by the separately-available [PDAP](#) package. This is described briefly [here](#).

## Geometric morphometrics

Landmark data can be entered in Mesquite as a continuous matrix with multiple items. Each character is a landmark, and each item is a dimension of the landmarks' coordinates. Thus, for two dimensional landmarks, the matrix could have the items "x" and "y".

Mesquite cannot yet perform Procrustes analyses to bring landmarks into a common scaling and alignment across taxa, but given that the data is already so prepared, the Landmark Drawings module of the Rhetenor package can reconstruct ancestral forms as shown below:



The algorithm used for the reconstruction is squared change parsimony.

# Testing monophyly of a group of beetles

David R. Maddison

## The question

There is an enigmatic group of terrestrial beetles called the Trachypachidae (left, below). These had traditionally been considered to be related to other terrestrial beetles in the suborder Adephaga, but later analyses of morphological data suggested (Bell, 1966; Hammond, 1979; Roughley, 1981; Ward, 1979) that they were instead closely related to some water beetles, the Dytiscoidea (right, below).



A terrestrial trachypachid (*Trachypachus holmbergi*, left) and an aquatic dytiscoid (*Hydrocanthus* sp., right).

Shull et al. (2001) report sequence data of the 18S rRNA gene suggesting that trachypachids are not related to dytiscoids, but are instead related to some other terrestrial beetles. For example, the most parsimonious trees in one analysis have trachypachids with other terrestrial adephagans, and not with dytiscoids. Forcing trachypachids with dytiscoids increases the treelength by 9 steps.

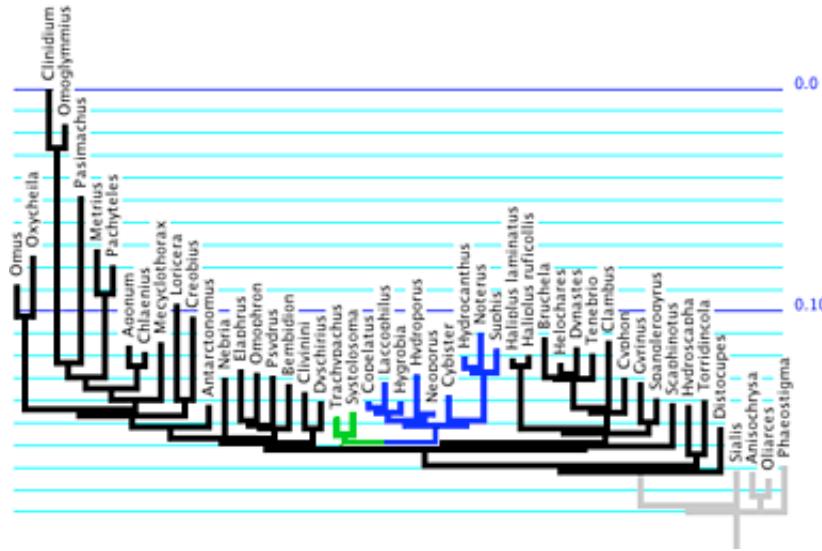
However, given the vagaries of the evolutionary process and phylogenetic inference, it is possible that even if trachypachids and dytiscoids are truly related, one might incorrectly infer that they are not, just by chance or because of difficulties such as long branch attraction (Felsenstein, 1978; Hendy & Penny, 1989; Huelsenbeck, 1997). A statistical test could help us determine whether or not a difference of 9 steps between trees with trachypachids placed with dytiscoids and unconstrained trees is expected under the

hypothesis that trachypachids and dytiscoids are related.

## A statistical test

Is 9 steps a significant difference in treelength in this context? Can we can reject the monophyly of trachypachids plus dytiscoids?

To test the hypotheses that trachypachids are related to dytiscoids, we first need to flesh out its details of the hypothesis. It needs to be detailed enough to allow us to predict the observations we would expect to see if this hypothesis were true. First of all, we will need a detailed phylogenetic hypothesis. We could find the tree of highest likelihood for 18S rDNA under the constraint that trachypachids are with dytiscoids; for one matrix, this tree is as shown below, with trachypachids marked in green and dytiscoids in blue.



Branch lengths of this tree and parameter values for a GTR + Gamma + Proportion Invariant (GTR+G+I) model of evolution are estimated by maximum likelihood using the 18S rDNA data. We thus have built a model that contains our best guess of the nature of evolution presuming that trachypachids and dytiscoids form a clade; the details of the model were established using the 18S rDNA data.

We can now ask: if evolution occurred under a GTR+G+I model with the parameter values as inferred, up the tree shown above (with trachypachids and dytiscoids forming a clade), then what would we expect the difference in treelength to be between the most parsimonious trees constrained to have trachypachids with dytiscoids and the most parsimonious unconstrained trees? Would that difference in general be similar to the observed value, 9? Or is 9 an unexpected value? Specifically, is 9 a value that we would expect to observe less than 0.05 of the time? If so, then we could reject the hypothesis that trachypachids are related to dytiscoids.

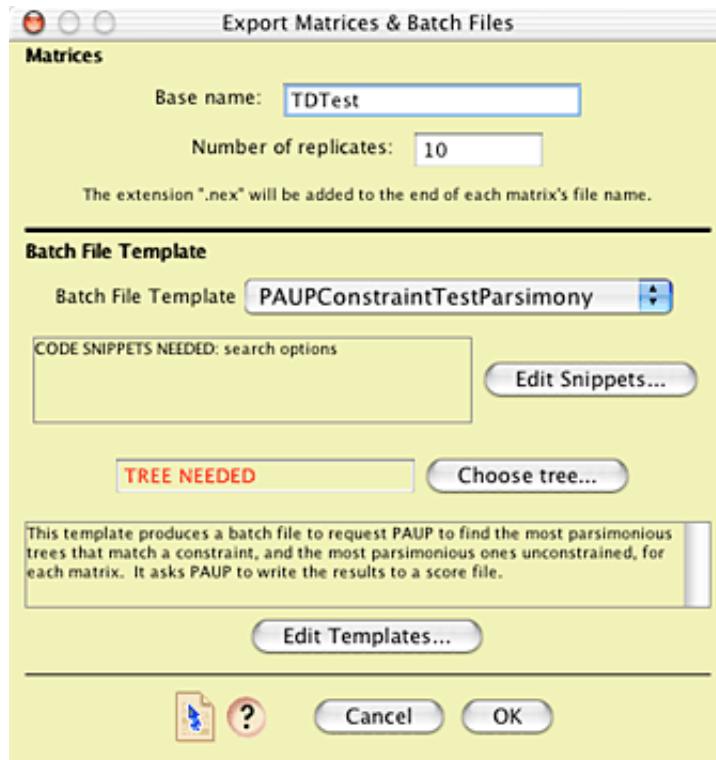
This test is an example of the use of parametric bootstrapping (Huelsenbeck et al., 1995; Swofford et al., 1996; Goldman et al., 2000). It is almost identical to the monophyly test

proposed by Huelsenbeck et al. (1996), except that they use a difference in likelihood rather than treelength as their test statistic. Treelength is used here for pedagogical reasons as it allows the reader to conduct the test quickly; the methods described below could easily be modified to use likelihood values instead.

## Building the statistical test

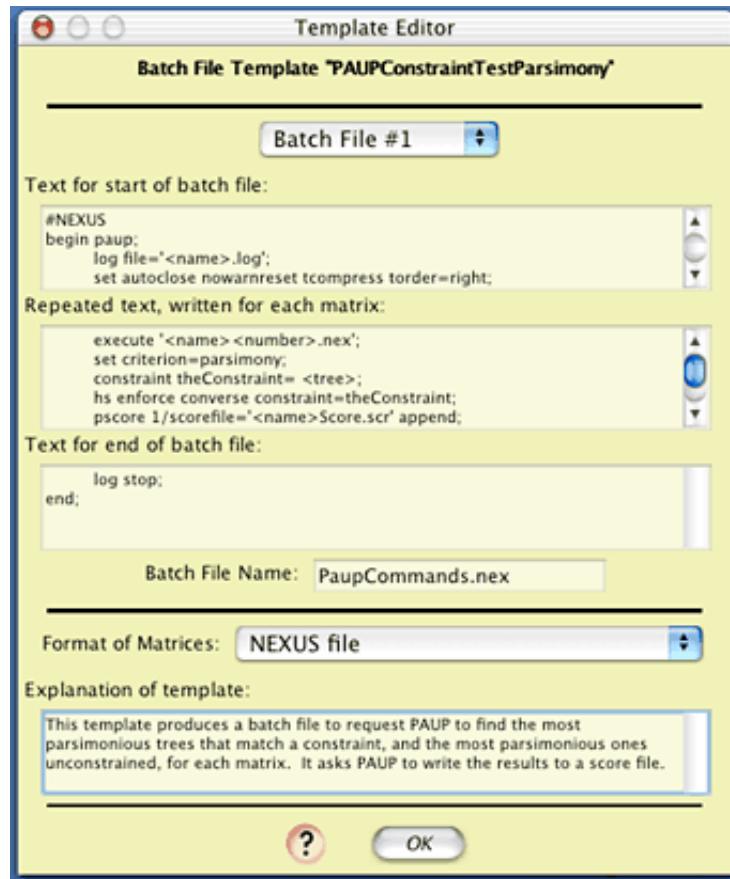
To conduct this test, in Mesquite open the example file "study001.nex", in the folder Mesquite\_Folder/docs/mesquite/studies/study001/. This file contains a matrix of 18S rDNA, and the tree of highest likelihood found in which trachypachids are with dytiscids, with branch lengths inferred from the data. It also contains a GTR+G+I model with parameters inferred by PAUP\* using maximum likelihood and 18S rDNA. We will ask Mesquite to simulate 100 matrices, and then (using the batch file Mesquite produces) ask PAUP\* to find the most parsimonious trees with trachypachids constrained to be with dytiscoids and the most parsimonious unconstrained trees, and write their treelengths to a scorefile. Mesquite will then read in the results and calculate the distribution of treelength differences, allowing us to determine if the observed value of 9 is unusual.

The simulations can be done by choosing ([Tree Window](#)) Analysis > Batch Architect > Export Matrices & Batch Files.... After selecting "Simulated Matrices on Current Tree" and "Evolve DNA Characters", choose the GTR+G+I model. You will then be presented with the Export Matrices & Batch Files dialog box. Give a base name for the matrices of "TDTest", and do 100 replicates. We will use the template "PAUPConstraintTestParsimony":



If you wish to look at the contents of this template, touch on "Edit Templates", and in the

Template Manager, select "PAUPConstraintTestParsimony", and press "View". (As this template is built-in, you can't edit it, only view it.) You will see some elements of the batch files that will be produced.



The complexity of this template needn't be of concern. If you are interested, details of how it works are presented in "[Design of Batch Templates](#)". Just press "OK" to get back to the Template Manager, and then press "Done".

You are almost ready to have Mesquite simulate the matrices. However, the template we are using requires that we specify a tree to use as a constraint tree. In this case, the constraint tree is one with trachypachids with dytiscoids, but with no other structure. A tree like this is stored in the file under the name "TD constraint". To choose it, touch the "Choose Tree" button, and select "TD constraint".

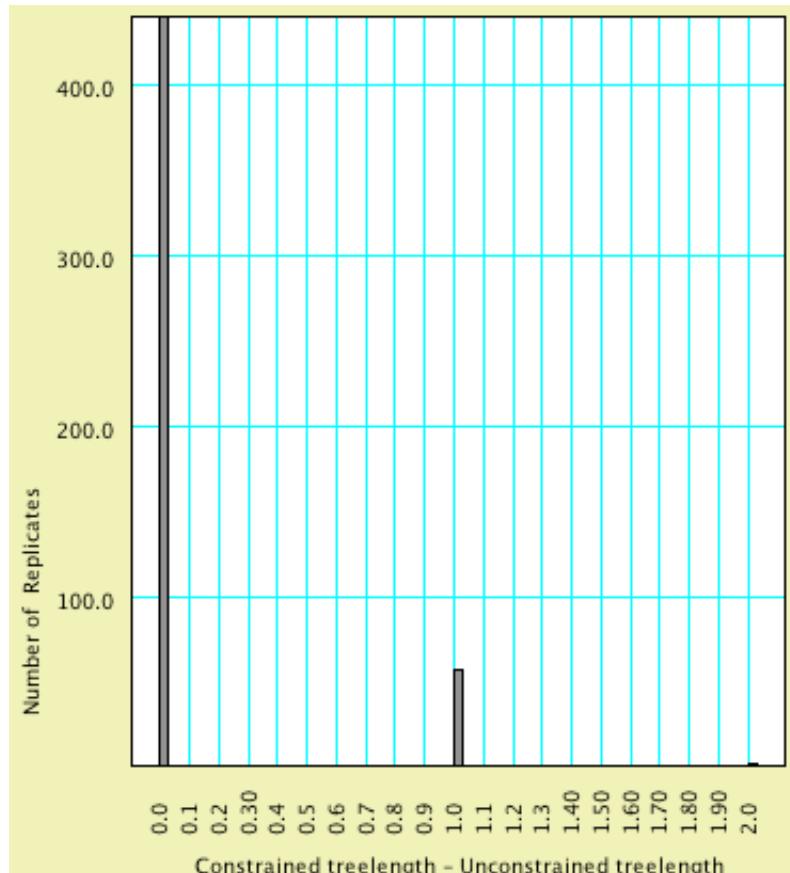
All the options have now been chosen. Press the "OK" button in the Export Matrices and Batch Files dialog box to start the simulation. You will be asked for a location to save the files, and be given some messages. When you are asked for the number of characters, Mesquite recognizes that you are calculating some elements of the model using an existing matrix (in particular, the frequencies of A, C, G, and T), and for this reason it gives as the default number of characters the number in the original matrix. That's the number we want to use in this case, as we want the simulation to be as realistic as possible.

Once the simulations are all done, then go into PAUP\*, and execute the file

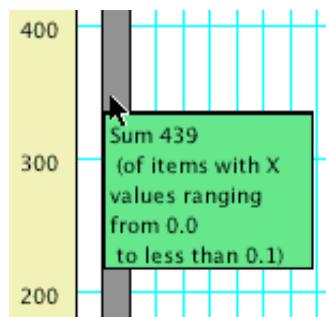
"PaupCommands.nex". When PAUP\* completes its analyses, go back to Mesquite, and choose (Tree Window) Analysis > Batch Architect > Show Results via Instruction File.... Choose the file "MesquiteInstructions", in the same folder as the simulated matrices, and then choose the results file (which should be called "TDTestScore.scr",). Mesquite should then show you a histogram of the treelength differences.

## Interpreting the results

An analysis of this exact sort but with 500 replicates yielded the following histogram:



Of the 500 replicates, 439 had a value 0, 58 had a value 1, and 3 had a value 2. You can determine these numbers by touching on each bar with the arrow:

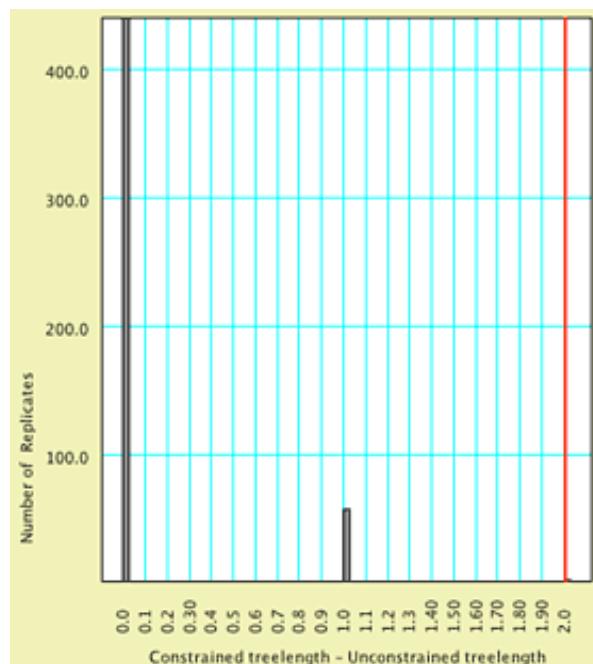


You can also see these values by going to the Text view of the chart (by touching the Text

tab at the upper part of the chart window):

▶ Graphics ▶ Text ▶ Parameters ▶ Modules ▶ Citations  
 Mesquite version 0.99      Date: Tue Aug 20 19:52:49 MST 2002  
 ======  
 Constrained treelength - Unconstrained treelength: Number of Replicates  
 0: 439.0  
 1: 58.0  
 2: 3.0

To see what values are in a specified percentile of the left or right tail of the distribution, you can choose (Bar & Line Chart) Chart> Analysis > Percentiles.... In the dialog box presented, you can choose the value of the percentile, the color of the bar to be shown, and whether the left or right tails (or both) are calculated. By default, the percentile value is 0.05; on this example, it would be displayed as:



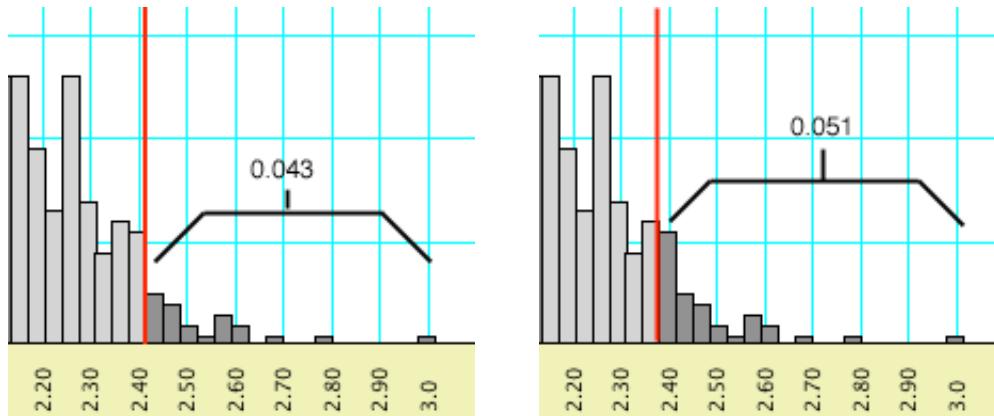
All values to the right of this red bar are thus in the extreme of the distribution, and our simulation would suggest that any value greater than or equal to 2 would occur with a frequency of less than 0.05. Our observation of 9 is thus an unlikely outcome if our hypothesis were true. Thus, we can reject our hypothesis at  $p < 0.05$ , and conclude that trachypachids and dytiscoids do not form a clade.

As it may be hard to interpret the value of the boundaries by the colored vertical lines, you might want to know the exact values of the percentile boundaries. This information can be gathered by looking at the text view for the histogram. Toward the bottom of the text view in the above example is the following:

0.0060 percentile boundary (right tail)  $\geq 2.0$   
 0.0060 is the closest percentile to 0.05 that was found,  
 and corresponds to 3 replicates out of 500.

The percentile bar shown will be that that is closest to that requested, but not over it.

The behavior of Mesquite can best be illustrated with another example. If you requested a percentile of 0.05, and if a percentile bar might be placed at 0.043 (left, below), but moving it one increment higher would include enough values to increase the percentile to 0.051 (right, below), then the percentile bar shown will be 0.043.



## Summary

The hypothesis tested herein was the monophyly of a clade. The steps in the test are:

- Determine the observed value of the test statistic for the observed DNA sequences. In this case, the treelength of unconstrained most-parsimonious trees was determined using PAUP\*, and was then subtracted from constrained (trachypachids with dytiscoids) treelength.
- The best tree matching the hypothesis to be tested (trachypachids with dytiscoids) is inferred using the available sequence data, with branch lengths inferred using maximum likelihood. This is the model tree.
- Values of parameters of a model of sequence evolution (gamma shape parameter, rate matrices, etc.) are inferred using maximum likelihood on the model tree for the DNA sequences.
- The data matrix is opened in Mesquite, along with the model tree. Submodels and a model of character evolution are created within Mesquite to match those inferred.
- Mesquite's Batch Architect is used to automate the process of simulating the evolution of multiple (100 or more) data matrices under this model, using Mesquite's Genesis package. Batch Architect also builds a command file for PAUP\* and an instruction file for Mesquite so that it can interpret the results of the PAUP\* analyses.
- The command file is executed in PAUP\*, telling PAUP\* to search for the shortest

constrained and unconstrained trees for each of the simulated matrices, accumulating the results into a score file.

- The Mesquite Instructions file is then read into Mesquite, and the score file is read, and Mesquite presents a histogram of the distribution of the test statistic (constrained treelength - unconstrained treelength).
- The observed value of the test statistic is compared to the distribution of the statistic expected under the model as determined by the simulations. If the observed value is more extreme than that expected (say, greater than 95% of the expected values), the hypothesis is rejected.

## References

- Bell, R. T. 1966. *Trachypachus* and the origin of Hydradephaga. Coleopts Bull. 20:107-112.
- Felsenstein, J. 1978. Cases in which parsimony and compatibility methods will be positively misleading. Systematic Zoology, 27, 401-410.
- Goldman, N., J. P. Anderson, and A. G. Rodrigo. 2000. Likelihood-based tests of topologies in phylogenetics. Syst. Biol. 49:652-670.
- Hammond, P. M. 1979. Wing-folding mechanisms of beetles with special reference to investigations of Adephagan phylogeny. Pp. 113-180 in Carabid beetles; their evolution, natural history, and classification (T. L. Erwin, G. E. Ball, D. R. Whitehead, and A. Halpern, eds.) . W. Junk, The Hague.
- Hendy, M.D. & Penny, D. 1989. A framework for the quantitative study of evolutionary trees. Systematic Zoology, 38, 297-309.
- Huelsenbeck, J.P., Hillis, D.M. & Jones, R. 1995. Parametric bootstrapping in molecular phylogenetics: Applications and performance. Molecular zoology: Advances, Strategies, and Protocols; Symposium held during Annual Meeting of the American Society of Zoologists, St. Louis, Missouri, USA, January 5-8, 1995 (ed. by J. D. Ferraris & S. R. Palumbi), pp. 19-45. Wiley-Liss, Inc., New York.
- Huelsenbeck, J.P. 1997. Is the Felsenstein zone a fly trap? Systematic Biology, 46, 69-74.
- Huelsenbeck, J. P., D. M. Hillis, and R. Nielsen. 1996. A likelihood-ratio test of monophyly. Syst. Biol. 45:546-558.
- Roughley, R. E. 1981. Trachypachidae and Hydradephaga (Coleoptera), a monophyletic unit? Pan-Pac. Entomol. 57:273-285.
- Shull, V., A.P. Vogler, M.D. Baker, D.R. Maddison, and P.M. Hammond. 2001. Sequence alignment of 18S ribosomal RNA and the basal relationships of adephagan beetles: Evidence for monophyly of aquatic families and the placement of Trachypachidae. Systematic Biology, 50:945-969.

Swofford, D. L., G. J. Olson, P. J. Waddell, and D. M. Hillis. 1996. Phylogenetic inference. Pp. 407-514 in Molecular Systematics (D. M. Hillis, C. Moritz, and B. K. Mable, eds.) . Sinauer, Sunderland, MA.

Ward, R. D. 1979. Metathoracic wing structures as phylogenetic indicators in the Adephaga (Coleoptera). Pp. 181-191 in Carabid beetles; their evolution, natural history, and classification (T. L. Erwin, G. E. Ball, D. R. Whitehead, and A. Halpern, eds.) . W. Junk, The Hague.

---

### Citation for this page

Maddison, D.R. 2004. Testing monophyly of a group of beetles. Study 1 *in* Mesquite: a modular system for evolutionary analysis, version 1.04, <http://mesquiteproject.org>.

---

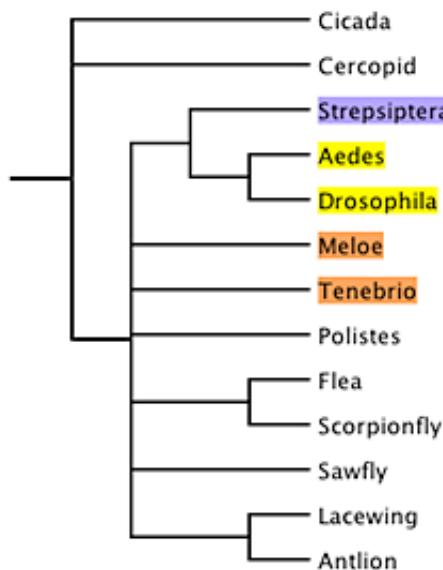
# Are strepsipterans related to flies? Exploring long branch attraction

David R. Maddison

## The question

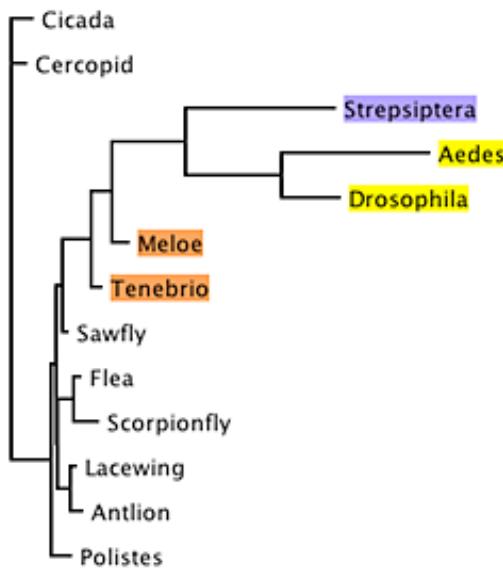
[Strepsiptera](#), sometimes known as twisted-wing parasites, is an enigmatic group of insects. They are parasites of other insects. The males have only one pair of wings and odd raspberry-like eyes and lobed antennae. Females are wingless, and in most species never leave the host. These insects have traditionally been considered related to [beetles](#), although that placement is supported by little evidence. Recently, molecular data have suggested that they may instead be related to true, two-winged flies ([Diptera](#)).

For example, parsimony analysis of a small data matrix of 18S ribosomal DNA yields (Carmean and Crespi, 1995) a phylogeny in which Strepsiptera is placed as sister group of Diptera (in yellow) rather than Coleoptera (in orange):



This is the strict consensus tree of 27 most-parsimonious trees.

However, examining the relative branch lengths for any one of the trees, it becomes evident that the branches for Diptera and Strepsiptera are unusually long:



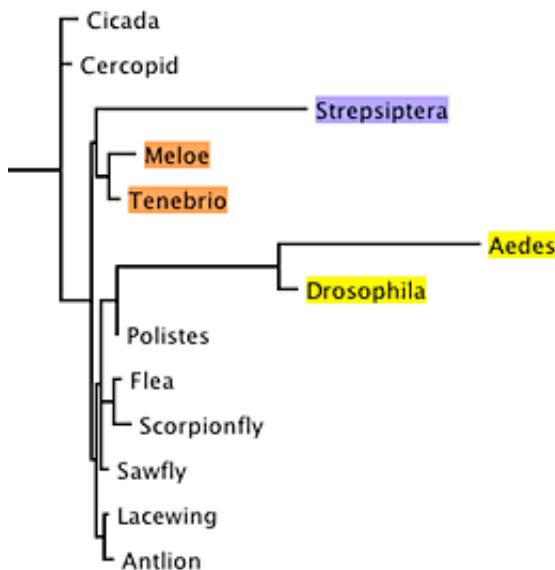
This raised the possibility that the association between Strepsiptera and Diptera in these trees was an artifact, caused by long-branch attraction (Felsenstein, 1978; Hendy and Penny, 1989).

Huelsenbeck (1997) set to determine if the relationship seen in this analysis of Strepsiptera with Diptera could be accounted for by long-branch attraction. To do this, he conducted a simulation study. We won't reproduce his whole study here, just one part of his Figure 2.

We will ask only one simple question: If strepsipterans are indeed related to beetles, would our observation that parsimony analysis yields a phylogeny with strepsipterans related to Diptera be unexpected? If we can show that it is unexpected, we can reject the notion that strepsipterans are related to beetles. If, however, inference of strepsipterans as related to Diptera is expected even if they are actually related to beetles, then we cannot so readily reject the traditional view.

## A simulation study

To conduct a simulation study, we first need a model tree. We can use a tree inferred using maximum likelihood, which places strepsipterans with beetles:



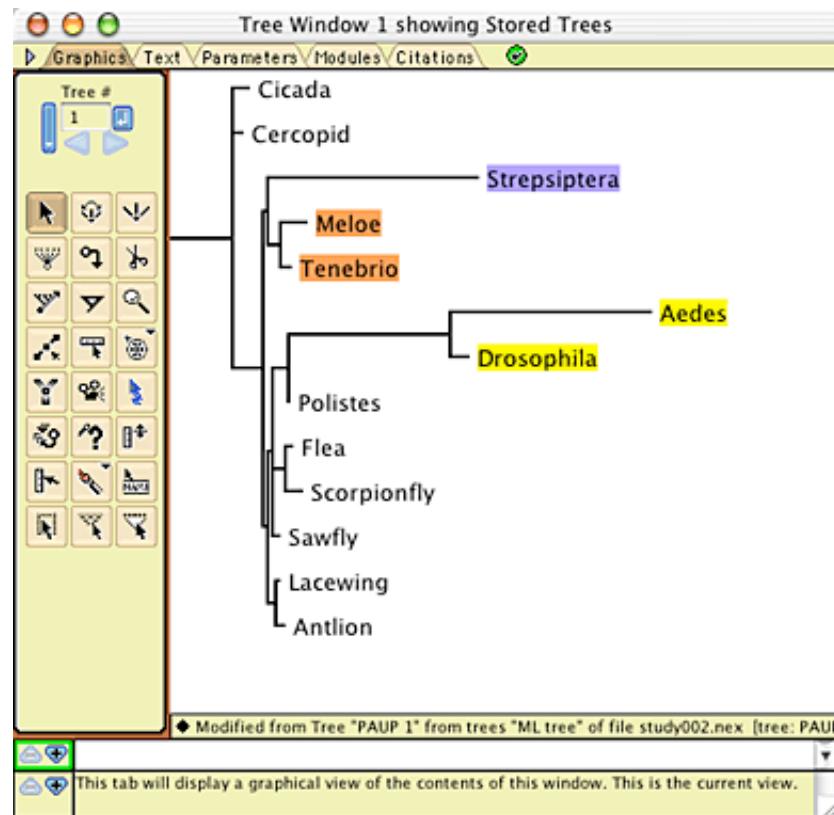
### Model Tree

The branch lengths of this tree were inferred using maximum likelihood. We also need a full model of DNA sequence evolution (which can also be inferred using maximum likelihood).

With this model in hand, we can simulate the evolution of 18S rDNA up the branches of this phylogeny, to yield a simulated matrix. We can then infer the phylogeny for this simulated matrix using parsimony, and see where Strepsiptera falls. Repeating this multiple times will give us an idea about the expected placement of Strepsiptera using parsimony inference presuming the model tree shown above.

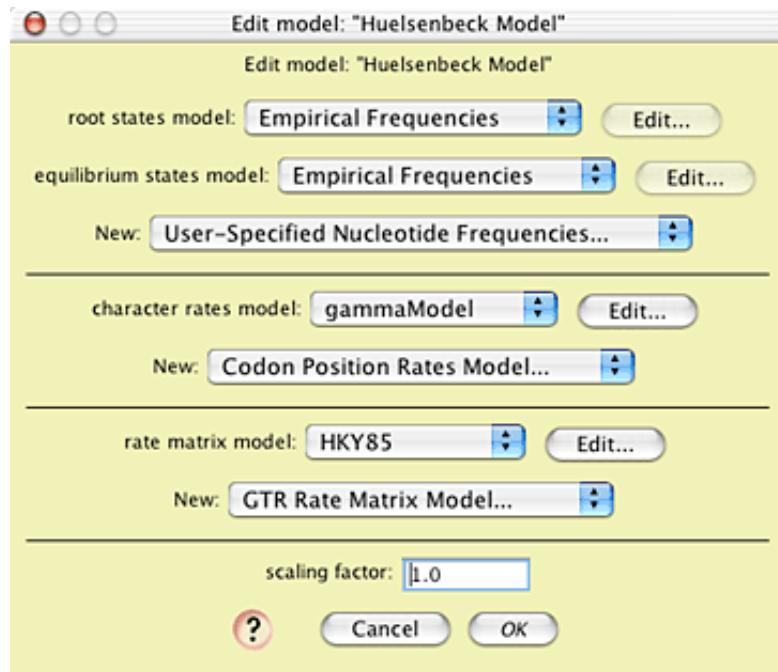
### Conducting the simulations

To conduct the simulations, in Mesquite open the example file "study002.nex", in the folder Mesquite\_Folder/docs/mesquite/studies/study002/. This file contains the 13-taxon data matrix from Carmean and Crespi (1995) as modified by Huelsenbeck (1977), as well as the model tree shown above. The model tree will appear in a window when you open the file:



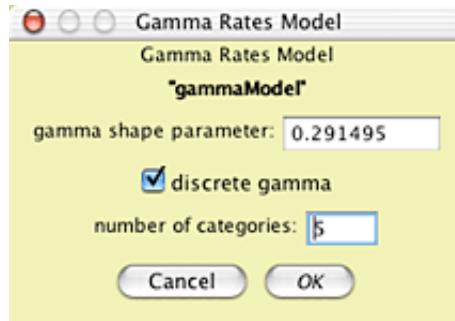
In addition, a model of character evolution has been entered into this file, with parameters obtained from maximum likelihood inference using PAUP\*4 (Swofford, 2003). These were inferred on the model tree using the observed 18S rDNA data.

You can see the model by choosing Characters>Edit Character Model>Huelsenbeck Model:

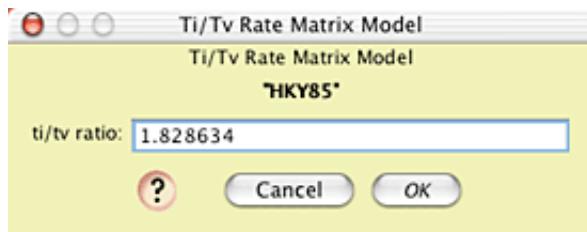


This model uses the empirical state frequencies as present in the original 18S rDNA matrix, a

model of character rate variation called "gammaModel", and a rate matrix model called "HKY85". You can see the nature of gammaModel by touching on the Edit button near its listing:

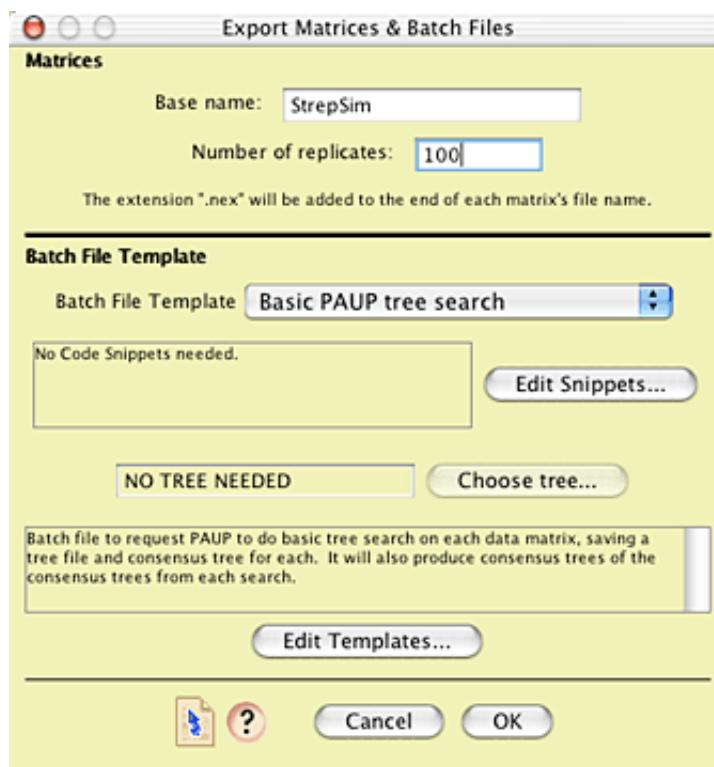


and the nature of the HKY85 model by touching on its Edit button:



With the model fully established, we can now conduct the simulation study. To do this, we want to ask Mesquite to create many simulated matrices, each evolved according to the model. We also want to create a script ("batch file") that will tell a tree-inference program (we will use PAUP\*, but another program such as NONA could be used) to find the most parsimonious trees for each of the matrices, and summarize the results. Mesquite's Batch Architect package contains the tools to automate this process.

Choose Analysis>Batch Architect > Export Matrices & Batch Files.... In the first dialog box that appears choose Simulated Matrices on Current Tree, then Evolve DNA Characters, then Huelsenbeck Model. You will be presented with the Export Matrices & Batch Files dialog box, in which you can enter the base name for the matrix files to be created and the number of matrices (100 is a good start).



The batch file template to be used is the one called "Basic PAUP tree search". This template builds a PAUP\* command file that will tell PAUP to execute and analyze each matrix in turn, and then, at the end, harvest the results and calculate a majority-rule consensus tree.

After pressing OK, you will be asked for a location to save the 100 matrices. It is recommended that you have an empty folder available into which they can be saved to avoid cluttering up another folder with many files. You will be asked one last question: the number of characters to be evolved in each matrix. As we want the simulation model to be as similar as possible to reality, we will choose to evolve the same number of characters as is present in the observed matrix, which is 770.

Mesquite will now simulate the matrices, and produce a batch file called "paupCommands.nex". This file consists of commands for PAUP\*. The start of the file looks something like this:

```
#NEXUS
begin paup;
    set autoclose nowarnreset nowarntree nowarnntsav;
execute 'StrepSim0.nex';
hs;
savetrees file = 'StrepSim0.trees';
contree / strict= yes majrule=no treefile = 'CBStrepSim' append = yes;
execute 'StrepSim1.nex';
hs;
savetrees file = 'StrepSim1.trees';
contree / strict= yes majrule=no treefile = 'CBStrepSim' append = yes;
execute 'StrepSim2.nex';
hs;
savetrees file = 'StrepSim2.trees';
contree / strict= yes treefile = 'CBStrepSim' append = yes;
```

After some initial setup, PAUP\* is to execute the first simulated matrix file ("StrepSim0.nex"), do a heuristic search (which, by default, should be for most-parsimonious trees), and then save the results to a tree file. It then saves the strict consensus tree of the most-parsimonious trees to a different tree file. It repeats this process for the next matrix, StrepSim1.nex, and the next one, StrepSim2.nex, and so on, until it gets to StrepSim99.nex, as requested near the end of the file:

```
execute 'StrepSim99.nex';
hs;
savetrees file = 'StrepSim99.trees';
contree / strict= yes majrule=no treefile = 'CBStrepSim' append = yes;

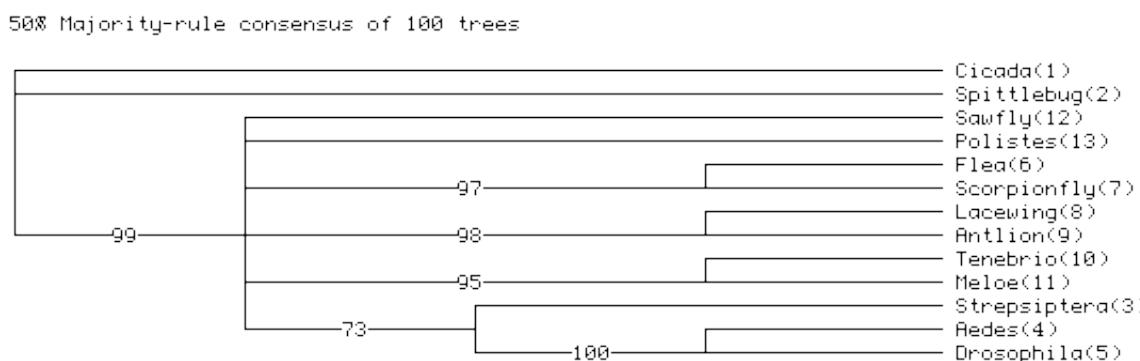
gettrees file = 'CBStrepSim' warntree = no allblocks;
savetrees file = 'consensus.trees';
contree / strict= yes treefile = 'StrictConsCons.trees';
contree / strict= no majrule=yes treefile = 'MajRuleConsCons.trees';
end;
```

After processing StrepSim99.nex, it reads in the tree file containing the accumulated consensus trees, and calculates both a strict consensus tree of the results from the 100 replicates, as well as the majority rule consensus tree of the replicate's results.

To ask PAUP\* to do this analysis, open PAUP\* and ask it to execute the file paupCommands.nex.

## Interpreting the results

After PAUP\* finished executing paupCommands.nex, near the bottom of PAUP\*'s main window will be the majority rule consensus tree of the strict consensus trees from each of the 100 replicates. It will look something like this:



The actual numbers may differ, but they should be similar to those shown.

The "73" on the Strepsiptera + Aedes + Drosophila branch indicates that in 73 of the simulated matrices, the most parsimonious trees had Strepsiptera with Diptera, even though the model tree on which the data were simulated had Strepsiptera with beetles. This indicates that the inferred placement of strepsipterans with dipterans with the observed data might be the result of an inference artifact, as inferring strepsipterans with Diptera is

expected even if they truly are with beetles.

## Summary

Simulations were done to see what trees we would expect from a phylogeny inference under a particular model tree. The steps in the study are:

- The branch lengths of the model tree are inferred using maximum likelihood in PAUP\*.
- Values of parameters of a model of sequence evolution (gamma shape parameter, transition/transversion rate) are inferred using maximum likelihood in PAUP\* on the model tree using the observed DNA sequences.
- The data matrix is opened in Mesquite, along with the model tree. Submodels and a model of character evolution are created within Mesquite to match those inferred.
- Mesquite's Batch Architect is used to automate the process of simulating the evolution of multiple (100 or more) data matrices under this model, using Mesquite's Genesis package. Batch Architect also builds a command file for PAUP\* and an instruction file for Mesquite so that it can interpret the results of the PAUP\* analyses.
- The command file is executed in PAUP\*, telling PAUP\* to search for the shortest trees for each of the simulated matrices, accumulating the consensus trees for each matrix in a tree file. PAUP\* is also instructed to calculate a majority-rule consensus tree of the results of each analysis.
- The majority-rule consensus tree is examined in PAUP\* to see what trees are expected to be inferred under these conditions.

## References

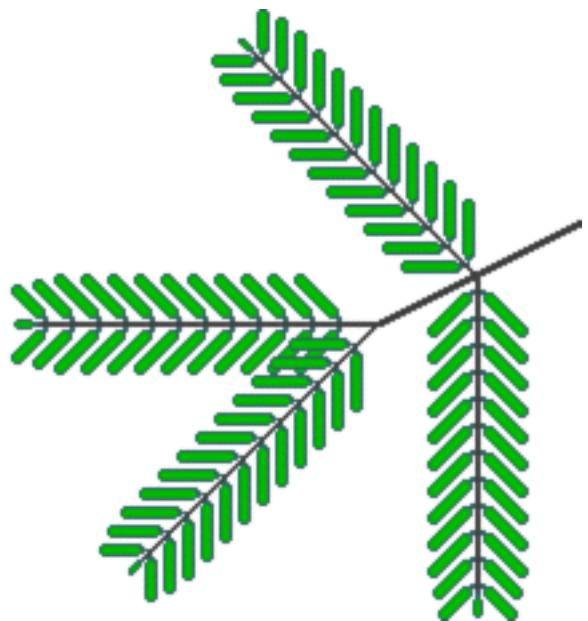
- Carmean, D., and B. Crespi. 1995. Do long branches attract flies? *Nature*, 373:666.
- Felsenstein, J. 1978. Cases in which parsimony and compatibility methods will be positively misleading. *Systematic Zoology*, 27, 401-410.
- Hendy, M.D., and Penny, D. 1989. A framework for the quantitative study of evolutionary trees. *Systematic Zoology*, 38, 297-309.
- Huelsenbeck, J.P. 1997. Is the Felsenstein zone a fly trap? *Systematic Biology*, 46, 69-74.
- Swofford, D. L. 2003. PAUP\*. Phylogenetic Analysis Using Parsimony (\*and Other Methods). Version 4 beta 10. Sinauer Associates, Sunderland, Massachusetts.

---

## Citation for this page

Maddison, D.R. 2004. Are strepsipterans related to flies? Exploring long branch attraction. Study 2 in Mesquite: a modular system for evolutionary analysis, version 1.04, <http://mesquiteproject.org>.

# MESQUITE 2



*Miscellaneous*

# Interaction with Other Programs

Mesquite needs to be able to interact with various other programs because typically it depends on them for inference of phylogenetic trees. The programs can interact through sharing files, or in a few cases directly.

## Importing/exporting files

Mesquite by default currently uses the NEXUS format for files. Other programs such as MacClade, PAUP, MrBayes and NDE all deal with NEXUS files, but there may be some commands not handled by all programs. Mesquite has in addition importers and exporters for various other file formats. You can import a file by requesting to open it; Mesquite will ask you to identify its file format. You can export a file using the Export item in the File menu.

**MacClade** – MacClade may not be able to read all of the information in a NEXUS file written by Mesquite, especially if it has multiple character or taxa blocks. You may need to simplify the file (e.g. delete some taxa blocks) for it to be read. Also, you could export the file as a Simplified NEXUS file for reading by MacClade.

**PAUP\*** – You might want to change PAUP's setting to "Issue Warning and continue" under "Warnings & errors" (set errorstop = no) so that PAUP does not complain about IDs commands within taxa blocks. Another alternative is to export the file as a Simplified NEXUS file.

**NDE** – NDE currently cannot handle some commands in the NEXUS files that Mesquite may write, including the TITLE command. If NDE warns you about this, turn off Mesquite's TITLE commands using the menu item under Defaults in the Mesquite Log window.

**CLUSTAL** – You may find it useful to use [Archive Taxon Names](#) and then [Condense Taxon Names](#) before exporting files for use by CLUSTAL, for it has difficulty with punctuation and long names.

## MrBayes interactions

There are two special file exporters useful for work with MrBayes: first, there is an exporter **Export NEXUS for MrBayes** that uses a simplified NEXUS format. It also composes a MrBayes block that includes character partitioning. A second exporter, [Fused Matrix Export \(NEXUS\)](#), can fuse different matrices in the course of exporting, create mixed matrices and include MrBayes blocks.

There is a special Tree Source (**MrBayes Trees...**) that can read a MrBayes tree file as MrBayes is producing it. If the option is so set, Mesquite can respond to new trees being added to the file. Thus, if you have a chart showing a value (say, Felsenstein's contrast correlation) over trees coming in from a MrBayes analysis, the chart will get updated when a

new tree is added. In the Tree window you can also ask to retrieve only the last tree in the file, and this will update as new trees are added. If MrBayes' .p files are in the same directory as its .t files, then MrBayes Trees will attach to the trees the tree score from MrBayes. This will be available as a NumberForTree "MrBayes Score".

A variant on MrBayes Trees is **MrBayes Max. A-Post. Tree**, which uses the .p file to find the tree with maximum posterior probability.

## CLUSTAL

Mesquite can call on [CLUSTAL to perform an alignment](#) of a selected region of the matrix.

---

# Publishing results from Mesquite

Are Mesquite's calculations well-enough tested to be reliable for published analyses? Because Mesquite is modular, the answer to the question of publication-readiness may not be a simple "yes" or "no". Mesquite modules are marked as either being prerelease versions (not ready for published results), or release versions. In addition, the modules are marked as substantive (possibly involved in producing results), or not (simple graphical or administrative modules not likely to affect results). When all of the substantive modules involved in a calculation are release versions, then we consider it as safe to publish the results as it is with any such biological software. Note however our [disclaimer](#).

How do you know if any substantive modules involved in a calculation are prerelease versions? Mesquite windows currently show either (green check) or (red !) in the information bar. The indicates that at least one module involved in producing the results of the window is marked as both substantive and pre-release; a indicates that all substantive modules are release versions. Also, if you select the Modules tab of the window to see the modules involved in the window, those modules that are both substantive and pre-release are marked by the

To help ensure that any remaining bugs are found and fixed, we urge you to pay attention to any strange, unexpected or apparently incorrect behavior of Mesquite, and to send bug reports to us at [info@mesquiteproject.org](mailto:info@mesquiteproject.org).

## How to cite Mesquite?

### Citing the system in general

The citation for this version of Mesquite is:

Maddison, W. P. and D.R. Maddison. 2007. Mesquite: a modular system for evolutionary analysis. Version 2.0 <http://mesquiteproject.org>

(The version number listed above might not be up to date. Check the Mesquite Startup window or the Project and Files window when Mesquite is running to find the version you have.)

### Citing Mesquite for analyses done

Mesquite's unusual modular nature may give great flexibility in calculations, but it can make it difficult to compose a citation for the calculation of published analyses. Here is a hypothetical example. If the analysis were mostly done by a module written by J. Doe and another by T. Za, one possible citation would be as follows: "The Snidely Index was calculated using the module SNIDIND (Doe, 2007) within the Mesquite system for

phylogenetic computing (Maddison and Maddison, 2007); its null distribution was determined by calculating it over 1000 trees simulated by the module Uniform of the SimSpeciation package (Za, 2007) with parameters  $s = 0.3$  and  $e = 0.1$ ." with the literature cited indicating:

Doe, J. 2007. SNIDIND: a Mesquite module for calculating the Snidely Index, version 2.1.

Maddison, W. P. and D.R. Maddison. 2007. Mesquite: a modular system for evolutionary analysis. Version 2.0 <http://mesquiteproject.org>.

Za, T. 2007. SimSpeciation: a package of modules to simulate evolutionary trees. Version 2.13.

## How to figure out what modules to cite

A single Mesquite analysis may be the result of the cooperation of many modules, some of which are worth citing (like a module that calculates a key value), some of which are not (like a module that draws the shape of the tree). While we could expect the user to keep track of the calculations requested and what modules to cite, Mesquite has some built-in features to help, via tabs in the information bar of each window. The two tabs that most directly help with citations are:

- **Citations tab:** when touched it shows the citations for modules involved in the analysis. This is the most direct way to find citable modules for an analysis.
- **Parameters tab:** this shows the parameters of the modules. These may include settings such as rates, weights, population sizes, the tree being used, and so on. They can be very important to help you keep track of the assumptions and input behind your results.

Another relevant tab is:

- **Modules tab:** when touched it shows in the window the employee tree of modules involved in producing the window. This includes modules involved in calculations shown. It is useful to help you understand what modules are in use, but it includes all of the modules involved, not just the ones worth citing.

## Which version is being used?

The current version of the Mesquite system being used is shown the in Mesquite window (the window that appears on startup) and in the Projects window (which appears to the left of the screen following startup). The current versions of the modules are reported in the citations view of each window.

## Disclaimer

THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. WAYNE MADDISON AND DAVID MADDISON DO NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS

REGARDING THE USE OR THE RESULTS OF THE SOFTWARE OR DOCUMENTATION IN TERMS OF THEIR CORRECTNESS, RELIABILITY, CURRENTNESS, OR OTHERWISE. IN NO CASE WILL THESE PARTIES BE LIABLE FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES THAT MAY RESULT FROM USE OF THIS SOFTWARE

---

# Support & Advice

If you think you've found a bug in Mesquite, our would like to suggest a feature, please read this page. For help in using Mesquite, see the [Help](#) page.

- [Suggesting new features](#)
- [Why is this bug here?](#)
- [Help system](#)
- [Reporting bugs](#)
- [Mesquite Listserv](#)
- [Java bugs that affect Mesquite](#)

## Why doesn't Mesquite do <insert your favorite calculation here>?

Mesquite is extensible. What it can do now is a small part of what we envisage. We're aware of many things that could be added. The biggest needs we feel at the moment are for likelihood calculations and more tools for examining character correlations.

We would welcome your suggestions at [info@mesquiteproject.org](mailto:info@mesquiteproject.org).

Of course, if you really want a new feature sooner than we can add it, you are welcome to write your own module to implement the feature. Contact us at [develop@mesquiteproject.org](mailto:develop@mesquiteproject.org) or see the Mesquite website for information on [development](#).

## Why does <insert feature> not seem to work properly?

If you think you have found a **bug** in Mesquite, please [report](#) it to us (see [below](#)). If we don't know about a bug, it is unlikely that it will be fixed. It might be natural to presume that we know about all the bugs in Mesquite, but it is not possible for us to have tried out every combination of options on every computer environment with every data matrix, model, etc. If you are not sure if it is a bug (a large number of users blame themselves when they encounter a bug, presuming it is something they have done wrong), please report it anyway. Even if your apparent "bug" is a mistake you made, your mistake might indicate a poorly designed interface that should be corrected.

That there will be bugs in Mesquite is inevitable. Mesquite is a large project, about 1500 pages of source code when printed single spaced at 7 point font (about 120,000 or more lines of source; over 5MB of ASCII text). We don't have a big team of people working on it; it was initially written by a single person, and now most of it is



written by only two people. We apologize for the bugs that remain, and look forward to your bug reports and efforts to aid us make Mesquite useful.

Mesquite is intended to become a community effort. Source code is available at the [Mesquite web site](#); we hope other programmers will help us improve the system.

## Check out the Help system first

Mesquite doesn't have conventional documentation for all of its features, but it does have various features to help you learn how to use it. Check out the [help page](#) for instructions. Also, browse through the example files, as they contain many explanations and hints.

We recognize that Mesquite is a complex system that offers many choices to the user. We plan to build macros and other tools to help build paradigmatic analyses for users who don't want to sort their way through options every time. However, they're only beginning to be built.

## Reporting bugs and requesting advice

Please report bugs to [info@mesquiteproject.org](mailto:info@mesquiteproject.org). Some bugs you may encounter will be bugs not in Mesquite but in the operating system or Java virtual machine (for example, see this [list](#)).

Partly to foster a community of users, and partly for efficiency, we are encouraging users to sign up to a [Mesquite mailing list](#) via which they can post queries about using Mesquite. Once you've signed up, you can send a message to the list.

If you're reporting a bug, it's important that you are able to specify, as precisely as possible, exactly what you did that generated the bug. Try to find a precisely repeatable series of actions that generate the bug. In any bug report, please specify the version of Mesquite and any important modules and your operating system. Be prepared to send us your data file so we can test the problem. It is also very helpful to send us the **Mesquite\_Log** file that is in **Mesquite\_Support\_Files** (which is in your home directory in Mac OS X and in "Documents and Settings\yourUserName\" in Windows). We need a copy of the log file as it was written immediately after your problem occurs.

(Of course, we recognize that some messages may be more appropriately directed directly to us. You may contact us at [info@mesquiteproject.org](mailto:info@mesquiteproject.org).

## Java bugs that affect Mesquite

Below are problems in Java on various operating systems that affect Mesquite. If these problems annoy you, please contact those responsible for your Java virtual machine.

---

## MacOS X

The recommended configuration for Mesquite on Mac OS X is OS X 10.4 with the latest Java (currently, 1.5). There are various screen redraw and crashing bugs in earlier versions of OS X and Java.

If at any point Mesquite quits spontaneously and OS X puts up a dialog that says that Mesquite has unexpectedly quit, this is what is called a "JVM crash". Technically it is not supposed to be possible for Mesquite to do this, and thus it represents a bug in Java and not a bug in Mesquite. Nonetheless, we would like to know about it, because we may be able to find a way to avoid the JVM bug.

### MacOS X running Java 1.3.1

Mesquite 2 does not run under Java 1.3 or earlier.

### MacOS X 10.0 through 10.2

We have not tested Mesquite 2 under these earlier versions of MacOS X.

### Mac OS 9

Mesquite 2 does not function on Mac OS 9, because it requires Java 1.4.

---

## Windows

- Windows bounce up and down as menus change (a rapid shrink & expansion). This is due to a design flaw either in Windows or the Java VM; there is nothing we can do about it.
  - After Mesquite was used for a while on earlier versions of Windows or Java on Windows, there was a chance that about 5% of menu labels would become randomly scrambled with labels of other menu items. For instance, the File menu might be labeled as the "Gray" menu, or the "Charts" menu, or have some other label drawn "randomly" from among the menu items within the menus. This was a bug in Sun's virtual machine. Because of this problem and the confusion it might cause, we have added a Reset Menus item in the File menu. This will force Mesquite to rebuild the menus, which usually corrects the mislabeling. However, we have not seen this behaviour for a few years.
-

## Linux/Unix etc

- Window sizes and placements sometimes are inappropriate under some window managers. Some window managers refuse to let Mesquite have control over window size and placement, and choose what appear to be random sizes and placements of windows. This is not in our control. Try changing your window manager. We have found that the default installations for KDE and Gnome of Red Hat 7 work well with Mesquite.
  - Messages about Fonts not found on startup may be given. The problem is that you need to install a supposedly optional package of fonts in order that the default installation of Sun's Java VM work properly (see <http://java.sun.com/products/jdk/1.2/changes.html#sunw>). We would have thought that if the default installation of a product claims to support a program but doesn't, then that would be considered a bug. Sun considered this to be not a bug, although they seem to have fixed it...
-

# Mesquite FAQ

[How do I find the most parsimonious tree in Mesquite?](#)

[How do I show the length of a tree?](#)

[How do I trace a character on a tree?](#)

[How do I analyze population structure using Slatkin & Maddison's s statistic?](#)

[How do I create a publication-quality tree?](#)

## How do I find the most parsimonious tree in Mesquite?

Mesquite is not designed for rigorous tree searches, but it can perform NNI and SPR tree searches. To search for the most parsimonious tree(s) for a character matrix, select Taxa&Trees > Make New Trees Block from > Tree Search > Heuristic (Add & rearrange). If you have more than one block of taxa in the file, you will be asked to choose the block of taxa you would like to perform a tree search for. Select "Treelength" as the criterion for the tree search. Select "Current Parsimony Models" as the source of current parsimony models. You have two options for the rearranger to use: NNI or SPR. NNI (Nearest-Neighbor Interchange) generally produces a tree faster than SPR (Subtree Pruning and Refractoring), but NNI has a greater chance of failing to find the most parsimonious tree(s) than SPR. You will be prompted to set the maximum number of equally good trees to store during search (MAXTREES). Increasing MAXTREES will result in an increase in search times, accompanied by a greater probability of finding the most parsimonious trees (and lower settings of MAXTREES will result in quicker, but less thorough tree searches). If multiple character matrices (for the Taxa block you are performing a tree search for) are included in the file, you will be prompted to choose the character matrix to use for the tree search. Once the tree search is complete, you will be asked if you would like to open a window to view the trees.

## How do I show the length of a tree?

Begin by opening a New Tree Window for the tree you are interested in. From this tree window, select Analysis > Tree Legend.... Choose "Treelength" as the information to be displayed in tree legend. Select "Current Parsimony Models" as the source of current parsimony models. If multiple character matrices (for the Taxa block you are measuring Tree Length for) are included in the file, you will be prompted to choose the character matrix to use for the tree length. The tree length will be shown in the legend that opens in the tree window.

## How do I trace a character on a tree?

Tracing the evolution of a character is not entirely trivial, and depends on a variety of choices you make, including which characters to trace, the method of character reconstruction (e.g., [parsimony](#), [likelihood](#), [stochastic character mapping](#)), etc. Here we provide brief instructions on reconstructing a single character on a tree, but we encourage you to read more about the details of [Tracing Character History](#).

Begin by opening a New Tree Window for the tree you are interested in. From this tree window, select [Analysis > Trace Character History](#) and "Stored Characters" as the source of characters to reconstruct. You will then be prompted to choose the reconstruction method. Parsimony models can reconstruct ancestral states for continuous and categorical data, including molecular data, while Likelihood and Stochastic Character Mapping methods can currently only be used for (non-molecular) categorical data. If you are using Likelihood or Stochastic Character Mapping as the reconstruction method, select "Current Probability Models" as the source of current probability models. If multiple character matrices are included in the file, you will be prompted to choose the character matrix to use as the source of characters to trace (Note: Mesquite will provide a list of *all* the character matrices for the corresponding Taxa block, including those for which models of character evolution are not currently available in Mesquite; e.g. even if you have selected "Likelihood" as the Character Mapping method, molecular and continuous Character Matrices, if they are present in the file, will be listed.). Choose the desired character matrix, and the first character will be traced on the tree. To scroll between characters, use the arrow on the Trace Character legend. For molecular and Categorical characters, you can change the colors of character states by double-clicking on the colored box in the Trace Character legend. Default colors for character states can be restored by selecting [Trace > Revert to Default Colors](#)

Additional information on the tools found in the [Trace](#) menu can be found in the Mesquite manual under the [Trace Character History](#) section.

## How do I analyze population structure using Slatkin & Maddison's s statistic?

We strongly urge you to read about [population genetic tools](#) in Mesquite, especially the section containing an [example](#) using Slatkin & Maddison's s statistic to test hypotheses concerning divergence times. There is also a [guide](#) which provides basic step-by-step instructions on performing this type of analysis. Additionally, the paper describing the statistic (Slatkin & Maddison, 1989, Genetics 123:603-613) is highly recommended reading.

## How do I create a publication-quality tree?

Mesquite is not designed for rigorous tree-estimation procedures, such a Maximum Parsimony, Maximum Likelihood, or Bayesian tree estimation. However, trees generated from programs such as PAUP, PHYLIP, and MrBayes can be read into and manipulated in

Mesquite. This [guide to producing quality trees](#) will help you perform these manipulations to create a tree suitable for scientific publication.

---

# New Features and History of Changes

Here are some of the more notable new features and bug fixes since version 1.0. Bugs that would have yielded errors in substantive results (e.g. a bug in a phylogenetic calculation) are marked with .

Beginning with version 1.1, a list of new modules in the latest version can be seen by choosing "List new modules" from the Help menu.

## Version 2.0

### New Features

- **Search system** – A "Search" text entry box is now available at the top of most windows (in the Log window, at the bottom). Enter text there and hit Return, and the text will be sought. There are several modes, marked by different symbols:
  - & – Search Features, AND: searches through information about Mesquite's features and returns a description of those features matching ALL of the search terms
  - / – Search Features, OR: searches through information about Mesquite's features and returns a description of those features matching ANY of the search terms
  - d – Search Data: searches the current data file or project and returns objects (e.g. taxa, characters, sequences) that match the search string.

You can choose the mode by touching on the symbol until the desired mode appears. In the Search Features modes, the results also give indication as to how the feature might be accessed. (There may even be a **Try It!** link that offers to invoke the feature.)

- **New windowing system**
  - By default, all of the windows of a single project now appear as tabbed panels within a single window. This change was made to limit window proliferation. If you want to "pop out" a tab to be a separate window, hit the curved arrow button within the tab (you can "pop in" the window again using the curved arrow button at the upper left side of the separated window). You can revert to the old system of many windows using the Open Windows as Tabs item in the Defaults menu of the log window.
- **Wizard-style dialogs** – In many contexts Mesquite needs to ask a series of questions when a feature is requested. In previous versions these would appear as a series of dialog boxes; in version 2 these now appear as a single dialog that moves from one question to another, in the style of a wizard. This change was made to make an area in the dialog to give a more thorough explanation of the request, and to minimize the visual distraction of changing dialog sizes. You can revert to the old style of separate dialog boxes (which may work better on some operating systems) using the Use Wizard-style Dialogs item of the Defaults menu of the Log window.
- **Diverse package** – There is a new package of modules that estimates and simulates

models of speciation and extinction. Some of its features are:

- Estimating speciation/extinction rates
- Detecting state-dependent differential speciation/extinction (BiSSE)
- Birth/Death tree simulation
- Simulations with evolving characters that affect speciation/extinction
- Lineages through time plots
- **Coalescence simulation with migration** – Coalescence within a population/species tree can now be simulated with a selected rate of migration per generation among populations.
- **Trace Character Over Trees** – This feature has been improved in several ways. The fraction of trees without a node is now shown, as is the fraction of trees with the node but not counted as they have ambiguous state assignment (by whatever criteria are chosen). The states at the terminal taxa are now shown. There is a new calculation mode, Average Frequencies Across Trees, available if the ancestral state reconstruction module yields frequencies or probabilities; this calculates, for each state, the average frequency across all trees (including trees in which the frequency for that state is very low or zero).
- **Sample Trees Directly from File** – One can choose as a source of trees a random subsample of the trees contained within a treefile.
- **Undo** – You can now undo some changes you might make to the character matrix in the character matrix window. In particular, edits of an individual cell, or changes made by the items in the Matrix>Alter/Transform menu, are now undoable. Other changes (e.g., deleting a character) are not yet undoable.
- **Consensus sequences** – One or more consensus sequences can be displayed in the character matrix editor.
- **Adding sequences to an existing matrix** – FASTA or NBRF files dropped onto a matrix of DNA or protein sequences can be read in and appended to those matrices.
- **Fusing genes to single matrix** – In concatenating matrices, you can now concatenate matrices with different sets of taxa, facilitating fusion of different genes into a single matrix
- **Alternative naming schemes for taxa** – This feature was introduced in version 1.1 as "Archived names". It has been expanded to permit multiple [archived naming schemes](#).
- **Matrix coloring to aid manual sequence alignment** – There are two new Color Cells options that aid with manual alignment. One ("Aligning Colors") uses colors that emphasize purine-pyrimidine differences; the other highlights sections that appear to be slightly misaligned.
- Various other smaller improvements have been made, for instance you can now import a PHYLIP tree file directly (without having to establish a taxa block first), and you can choose darker colors when assigning colors to objects.

## Bug Fixes

- (⚠) There was a bug in writing decimal numbers as strings. Frequently, the 7th or 8th significant decimal place of a number would be written incorrectly. This did not affect

how data were stored in the file; it affected only the reporting of results and screen display. This is unlikely to have a serious effect on published results for two reasons. First, it usually affected only the 7th or 8th significant decimal place. Second, the situations in which it affected a more important decimal place were obscure and perhaps rarely used by users. These are: Density output in Scattergram; Parameters Explorer for newer likelihood calculations; Location of changes in Stochastic Character Mapping. In these cases, the 3rd or 4th decimal place could be affected. This bug has been fixed.

- (⚠) On Windows operating systems, pasting into the data matrix often resulted in the loss of the first character of some rows. This may have been benign, but occasionally would result in data being pasted incorrectly (e.g. 1.234 received as 0.234). This bug has been fixed.
- (⚠) Among the coalescence calculations, the method to resolve a polytomous contained gene tree failed to resolve all polytomies needed, leaving some unresolved. This resulted in inflated Deep Coalescence counts. This bug has been fixed.
- (⚠) Fixed bug in which parsimony step matrix would be incorrectly written and could not be re-read properly if it referred to characters with more than 10 states.
- Fixed bug in which Recode dialog box would disappear unexpectedly.

## Version 1.12

### Bug Fixes

- (⚠) In versions 1.1 and 1.11 of Mesquite several of the simulations and randomizations had a bug in choosing the random number seed for the first item simulated or the first randomization. The designed behavior is for the simulations to begin with a random number seed based on the current clock, and then from that calculate base seeds for each of the replicates using the procedure described below under version 1.1 bug fixes. Version 1.12 has been corrected to use this designed behavior. However, in 1.1 and 1.11 for several of the simulations the first replicate always used 1 as its random number seed, although all subsequent replicates used the correct method as designed. This bug did not affect independence of replicates within a simulation of multiple replicates (except when multiple tree blocks were simulated using "Simulated Tree Blocks"), but it did affect independence among separate simulations/randomizations. Since the bug affected only the first replicate, it would not be expected to affect significance values unless very few replicates were used. The affected simulations/randomizations are: Simulated Trees (first tree used seed 1 always); Simulated Tree Blocks (first tree of each block used seed 1 always); Randomly Modify Trees (first tree used seed 1 always); Randomly Modify Matrices (first matrix used seed 1 always).
- (⚠) When taxa are selected by the user and Rarefy Tree is used to generate trees with a requested number of taxa randomly deleted, taxa are chosen for deletion randomly

only from among those currently selected. In previous versions of Mesquite this did not work correctly, in that frequently fewer than the requested number of taxa were deleted. This may have been evident to the user because the modified tree would have contained more taxa than expected. This bug is fixed in 1.12.

- (⚠) The likelihood calculations for categorical characters were designed only for characters with a contiguous series of states starting at 0, and to prohibit other characters. That is, a character with states 0, 1, 2 would be permitted, but a character with states 1 and 2, or a character with states 0 and 2, would not be permitted. This requirement was mentioned in the manual, but the requirement was not enforced properly within the program, and thus Mesquite permitted the calculations with some such characters. The reconstructions are incorrect. In version 1.12 the prohibition is correctly in place; Mesquite will refuse to calculate likelihoods with such characters.
- (⚠) A single character matrix can be modified from various windows, such as the List of Characters window (e.g., by moving, deleting characters) and the Character Matrix Editor. When a character matrix editor is open and a cell is being edited, the editing field could sometimes be misapplied to the wrong character or taxon if elsewhere characters or taxa are deleted, added or changed in order. The effect of this is that the state of one character or taxon can get copied inadvertently to another character or taxon. This bug has been fixed.
- A bug in the shading of character states for Trace Character History for continuous characters has been fixed. In previous versions there were situations in which a branch was given a color slightly offset from its correct color according to the legend and its state value. (The correct state could always be seen, however, by holding the cursor over the branch.) This has been fixed.
- Fixed bug in Find Sequence (Matching sequence).

## Version 1.11

### Bug Fixes

- (⚠) In previous versions of Mesquite, a Taxa Association between taxa blocks, for example one which records what haplotypes belong to what species, could come to have incorrect linkings if taxa were reordered in the file. This is not directly a substantive bug, but could yield calculations with errors if the user does not notice that taxa have become mis-associated. This bug has been fixed
- Clustal Align now works under Windows
- Various other small bugs were fixed.

## Version 1.1

## New Features

- Character evolution
  - [Pagel's \(1994\) correlation test](#) for 2 binary characters
  - [Stochastic character mapping](#) for simple categorical (non-molecular) data
  - Likelihood reconstruction of ancestral states now permits root to have [equilibrium frequencies](#) implied by model
  - Missing data permitted with categorical character likelihood and continuous parsimony
- Gene tree/species trees
  - Fit of gene tree-species tree can now be assessed by counting [duplication and extinction events](#)
  - new graphical editor for specifying [associations](#) between containing and contained taxa
- Alignment (see the a href=".../mesquite/align/aAlignIntro/index.html" Align Package manual)
  - **Manual alignment tools** in editor (Block Mover, Sequence Pusher)
  - **Automated pairwise alignment tool** in editor
  - Submission of a selected region in the matrix to be aligned by ClustalW and then reincorporation of the newly aligned region into the matrix
- [Genetic Codes](#)
  - Assignment of **genetic codes** to individual characters
  - Translation of nucleotide data to protein data
- Molecular Data
  - various tools for [altering molecular data](#), including reverse complement
- [Distances](#)
  - Jukes-Cantor, Felsenstein 81, Kimura 2-Parameter, Felsenstein 84 distances
  - Great Circle distances in the separately-released [Cartographer package](#)
- Charts
  - Values for Taxon Pairs now available in Bar & Line Charts and Scattergrams, allowing one to plot, for example, uncorrected distance versus corrected distance for all taxon pairs within a matrix.
  - Line charts
  - Bar & Line charts can now show **cumulative** results in various fashions (e.g., cumulative, cumulative average, reverse cumulative average)
- [Interactions with GenBank](#)
  - Import into matrix the top BLAST matches to sequences selected in matrix
- Interaction with [MrBayes](#)
  - **live updating** of trees produced by ongoing MrBayes analysis
  - exporter that [merges matrices](#) and includes a MrBayes block summarizing the partitions in the resulting matrix. Merged matrix can be mixed (DNA, protein, and morphological) using MrBayes's modified NEXUS format.
- Managing matrices and files
  - [fuse matrices](#) into a single matrix, even if matrices refer to different taxon names (e.g., sequences of different genes), using a [table of correspondences](#)

- among taxa blocks
  - [alternative taxon names](#) (permitting you to have alternative naming schemes for taxa in a file)
  - [include taxa](#) from NEXUS, NBRF or FASTA files into current file, e.g. to add newly acquired sequences to an existing matrix
  - import/export Phylip tree files
- Randomization (reshuffling) of matrices can now respect taxa partitions; also can reshuffle within taxa
- PDF file saving of graphics of trees and other windows
- Colors of character tracings can now be changed
- Simplified installation process for Windows computers
- A [complete list](#) of standard modules has been added to the manual.

## Bug Fixes

- (⚠) In previous versions of Mesquite, simulations and randomizations of trees and characters used a shortcut to go to the n'th replicate, cycling the (pseudo-)random number generator n times before entering the replicate. This caused adjacent replicates to be not entirely independent. In most cases we expect this bug would not have had a biasing effect if sample sizes were large. This bug has been fixed by adding 1 to the random number seed after cycling n times to generate independent random number sequences for each replicate.
- Various bugs and inefficiencies affect memory use have been solved in file reading, tree display and printing
- There were some copy/paste bugs, especially under Windows, that have been fixed
- Swapping in heuristic tree search now retains only unique trees
- NEXUS files with high-ASCII characters (e.g., letters with accents) are better handled
- Various other minor bugs and inefficiencies have been fixed
- build h61 (released 23 May 2006) has a few extra bugs fixed: (1) Categorical likelihood calculations no longer complain if root has zero length; (2) Bug in Java 1.5 on Mac OS X caused list dialogs not to respond properly to clicking "Show Secondary Choices"; workaround implemented; (3) Incompatibility with PDAP file exporter solved; (4) directory choice dialog under Mac OS X improved.

---

## Version 1.06

### New Features

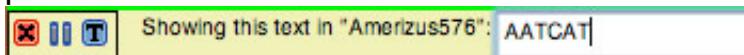
- The window showing annotations for taxa, characters and cells of a matrix has been integrated into the Character Matrix Editor as a panel within that window. This makes it easier to annotate without having to manipulate two windows. This panel can be requested by the **Show Annotations Panel** item in the Matrix menu, or by touching on the small button ( ) near the lower left of the Matrix Editor (beneath the taxon

names). The pencil tool (-pencil) has disappeared. Another change is that images in annotations can now be remote (i.e. URL's). Also, the annotations panel is also available in the List of Taxa window (for annotating taxa) and the List of Characters window (for annotating characters). In the list windows, the panel can be requested by touching on the small button (grid) at lower left of the table.

- Annotations can now be attached to the states of a categorical character, within the State Names Editor (available from the Matrix menu of the Character Matrix Editor). The annotations can be viewed by choosing **Show State Annotations** in the State\_Names menu or by touching on the small button at lower left of the table (grid).
- A legend is now available for the colors in a Character Matrix Editor. The legend can be shown by selecting **Show Color Legend** in the Matrix menu, or by touching the small button(grid) at lower left of the Matrix Editor (beneath the taxon names). If you double click on a color in the matrix, the editor will move to a cell with that color.
- There are buttons at the lower left of the Character Matrix Editor to open the List of Characters window (grid) and the List of Taxa window (grid). Reciprocally, the List of Characters window has a button (grid) to show the Character Matrix Editor.
- Character matrices can now be viewed in a "Birds-eye view" with very narrow columns.
- There is a **State Names Strip** available for categorical matrices under the Matrix menu. This shows a strip along the bottom of the matrix in which the names of characters and states are shown. Similarly when Bird's Eye View is used for a matrix, a **Bird's Eye Closeup Strip** appears below the matrix so that you can more clearly see the states of the characters of focus. Both of these strips can be made higher by going to their upper edges, clicking and dragging upward.
- You can now view a character matrix in more than one window, by requesting an **Extra Matrix Editor** from the Characters menu. This is useful, for instance, if you want to have the cells of the two matrices colored differently.
- The order of characters can now be managed through stored character orders. A **Default Order** of characters is automatically stored as characters are created. This default order is visible in the List of Characters window as the first column. If the order of characters is accidentally or intentionally changed (for instance using the **Sort Tool**) then the default order can be recovered easily by applying the Sort Tool to the Default Order column of the List of Characters window. In addition, you can store and reload other character orderings by showing the **Stored Character Order** in the List of Characters window (Columns menu). You can store the current ordering and recover previously stored orderings using the drop down menu from the Stored Order column heading. When you do this, you will see the numbers in the column change, but the characters will not immediately be reordered. The reason for this is that what is stored and displayed is simply a number for each character marking its position. It is up to you to use that ordering to resort the characters. Thus, to recover a former sequence of characters, load the stored ordering, then touch on its column with the sort tool.
- There are new facilities for **searching for sequences** in molecular data files. These are found in the Edit menu for editor windows of molecular sequence data. These are:
  - **Copy Sequence** (at bottom of Edit menu): This copies the selected cells of the

matrix into the computer's clipboard as a sequence. That is, whereas the standard Copy would place into the clipboard selected pieces of the matrix in tab-delimited text format (e.g., if the sequence AATCA is selected, "A-tab-A-tab-T-tab-C-tab-A" would be copied), this modified Copy Sequence command does not include tabs (thus, "AATCA" would be copied). This style of copying is useful when interacting with programs like [Sequencher \(TM\)](#). For instance, if you want to find a piece of sequence in a matrix in Mesquite within a chromatogram viewer of Sequencher, do the following: select the sequence in Mesquite, choose Copy Sequence, then go to Sequencher, select Find Bases, and paste the sequence as the search string.

- **Maintain Target Match** (in the Find Sequence submenu of the Edit menu): This highlights and maintains highlighted the first occurrence of a given sequence in a given taxon. First, you are asked which taxon to search in. Then, it displays a panel like this:



underneath the matrix. The first button (red X) is to close the panel; the second pauses the search; the third allows you to select another taxon as your focus. If you type a sequence into the text area, the matching sequence (if any) will be highlighted in the matrix. Mesquite is constantly monitoring this text, and so you don't need to give any command to find again if you change the text. This is useful if working with a program like Sequencher. If you see a stretch of sequence while viewing chromatograms that you'd like to find in the matrix in Mesquite, type in the sequence into the text box and you will quickly be taken to it in the taxon.

- **Maintain Clipboard Match** (in the Find Sequence submenu of the Edit menu): This is similar to Maintain Target Match, except that it obtains the search string not from the text area but from the clipboard. If the clipboard changes, the function will automatically find the sequence again in the matrix. This is useful if working with a program like Sequencher. If you turn on Maintain Clipboard Match, then you can copy stretches of a sequence within Sequencher, and Mesquite will automatically highlight it, without your having to return to Mesquite or give any other command to it. (Mesquite is constantly monitoring the clipboard to see if it changes).
- There is an optional system for assigning and maintaining ID strings to uniquely identify taxa and characters. This can be turned on via the menu items **Invent Unique IDs for New Taxa** and **Invent Unique IDs for New Characters** in the Defaults menu of the Log window. When enabled, newly created characters and taxa are automatically assigned IDs which are maintained even if the taxa or characters are rearranged through reordering, additions or deletions. These unique IDs are little-used in Mesquite at present, but eventually will permit users to reassociate taxa and characters from different files even after names have been changed (for instance, to re-read a tree file saved with different taxon names). The IDs will also be used in database connectivity and collaborative features. To assign IDs to already-created taxa or characters, go to the List of Taxa or List of Characters windows and choose **Unique IDs** from the

Columns menu. IDs can be generated using the drop down menu from the column's header. The one disadvantage of assigning IDs is that the IDs are stored in the Taxa block and Characters block of the NEXUS file, which causes some programs to refuse to read the file. (PAUP\* by default will refuse to read these files, but you can request that PAUP\* ignore these extra commands by the command "set errorstop = no;" which is available in its Options Menu, Warnings and Errors.)

- There has been a shift in NEXUS file format for **annotations** – the AN command is now used. This, unfortunately, produces files that are not readable by version 1.05. The reason for this change is to minimize text in the file when dealing with a large matrix with thousands of annotations.
- **Pairwise comparisons** were previously forced to find all possible pairs. Users can now impose a limit so that only the first *n* pairs are found and considered. This is important if there are too many pairs to efficiently consider.
- Support for the new [Cartographer](#) package.
- Groups of characters and taxa can now be assigned grayscale values instead of just colors.
- Rarefy Tree (a random tree modifier) now uses taxon selection. If some of the taxa are selected, than Rarefy Tree chooses which taxa to exclude randomly only from among the selected taxa.
- There is a new option to export files, **Old-Fashioned NEXUS**, for use by programs that understand only a small subset of the NEXUS format such as MrBayes.
- Trees can be drawn in "**Eurogram**" style by using Square Tree and selecting the menu item **Cut Corners**.
- The [Developer's documentation](#) has been updated

## Bug fixes

- (⚠) Fixed bug that caused labels for internal nodes (including Bayesian support values) to be scrambled if the branch interchange or reroot tool was used.
- (⚠) The PAUPConstraintTestParsimony template included with Batch Architecture for doing the test of monophyly has one of the searches incorrectly searching for the optimal trees with the *converse* of the constraint rather than the constraint itself; this has been fixed.
- (⚠) Fixed bug that caused tree statistics to be displayed incorrectly in columns in the List of Trees window if trees were deleted.
- Fixed bug in dialog boxes with lists from which to choose that made them unusable under Mac OS X 10.4.
- Fixed bug that prevented use of high ASCII characters in NEXUS files (e.g. "é", "ü", and so on).
- Fixed bug that greatly slowed file reading under Java 1.5
- Fixed various issues to improve performance under Mac OS X Java 1.4. (Some windows are still misdrawn, but that is due to bugs in Mac OS X.)
- Fixed other minor bugs (including, but not only: table copy/paste under Windows; use of arrow keys in matrix under Windows; retention of full character information in

concatenated matrices; stepmatrix reading when character matrix ambiguous; fill tool with gaps)

---

## Version 1.05

### New Features

- added a new Find feature (in the Edit menu) that allows you to search for sequences within DNA data matrices. The first example of a module doing this is "**Matching Sequence**" which finds sequences matching a target sequence the user enters. The search allows a certain number of mismatches. Optionally, it can search for the reverse, complement and reverse complement of the target sequence.
- added a new item in the Select menu of Character Matrix Editor, "**Select Lowercase Ends**". This selects the ends of a DNA sequence up to the first block of N sites in a row with uppercase symbols. The user chooses N. Assuming that lowercase letters are used for less certain base calls, this can be used to select and then trim (by painting with gaps) poorly-sequenced terminal regions.
- the **Concatenate Matrices** command will now allow molecular sequence data to be concatenated to the end of a matrix of standard categorical characters.
- in the Mac OS X download, added an executable to run Mesquite under Java 1.4. This may be less stable than the standard version running under Java 1.3.1. The Java 1.4 version should be used only if you have Java 1.4.2 update 2 or later on Mac OS X. (The problems with early versions of Java 1.4 appear to be restricted to Mac OS X.)

### Bug Fixes

- (⚠) fixed bug in the Brownian motion simulations of continuous character evolution. Brownian motion puts more change on long branches, less change on short branches. The bug in the simulations caused this contrast between long and short branches to be heightened – long branches had even more change relative to short branches than they should have. (Inadvertantly the effect of branch length had been squared in the calculation.)
- 

## Version 1.04

### New Features

- added Utilities submenu to Tree menu of Tree Window. Currently there is only one utility, **E-mail Tree**, which prepares a text image of the tree and puts it in the body of an e-mail message requested using a web browser's mailto: command.

### Bug fixes

- (⚠) fixed bug in two tree simulators, **Uniform speciation (Yule)** and **Uniform Speciation with Sampling**. In versions 1.03 and before, these failed to randomize tip labels, resulting in trees whose shapes matches the model's expectations, but whose particular relationships did not. Thus, in previous versions the taxa were added one at a time to a tree growing through time. For example, if there are 6 taxa, then the divergence between the 6th taxon and its sister species was always the most recent divergence in the tree. While this would not affect questions about general tree symmetry (e.g. "What is the distribution of tree asymmetries under uniform speciation?") or branch length distributions, it would affect questions about similarity between an observed tree and those expected under a null model of speciation.
  - fixed bug that caused a crash when characters were deleted from a matrix which had annotations, images or colors attached to the cells but which didn't have footnotes.
  - fixed a few minor bugs.
- 

## Version 1.03

### New Features

- added **automatic file backup**, which retains copies of previously saved versions of files. This feature was added to allow the user to return to a previous version in case of errors or other problems. To enable this, in the Log window or Projects & Files window select Automatic NEXUS backup in the Defaults menu. Indicate the number of backups to be saved. If you enter, for instance, 3, then when you save a file named "myfile", the previous version of the file (if any) will be retained and named myfileBKP1. It will be in the same directory as myfile. The version before that will be called myfileBKP2, the version before that myfileBKP3. When you save the files again, the file ending in BKP3 will be deleted, BKP2 will be named BKP3, BKP1 renamed BKP2, and the newest backup will be called myfileBKP1.
- enabled **grayscale shading** in Trace Character History for continuous-valued characters. This can be turned on in the Trace menu by selecting the "Use Grayscale If Continuous" menu item.
- Matrices can be compared by **selecting cells that differ**. To request this, choose "Select by Matrix Comparison" from the Select menu of the character matrix editor. This is similar to the Utility "Compare matrices", but instead of giving a textual summary of the differences between two matrices, "Select by Matrix Comparison" highlights those cells of the matrix that differ.

### Bug fixes

- fixed a bug that caused NOTES blocks to be written in duplicate form into all linked files (opened via Link File...) if all files were saved. This bug was generally innocent except if a linked file was subsequently unlinked from the main data file. This would

leave the linked file with a relictual copy of the NOTES block. If the main data file was then edited (e.g., footnotes changed, characters deleted or moved), and then the secondary file was later linked again to the main data file, its relictual (and now out of date) NOTES block could override the current NOTES block in the main file. This could cause footnotes in the main data file to revert to former ones, or to be applied to inappropriate characters or taxa. In addition to fixing this bug, we have added a warning that is given if Mesquite detects that a NOTES block is being overridden. If you suspect that your footnotes may have been corrupted by this bug, please contact us for assistance.

- fixed a bug in Trees Directly from File that gave spurious warnings.
  - fixed various other small bugs
- 

## Version 1.02

### New Features

- **multiple notes with images** can now be associated with taxa, characters and cells of character matrices. Prior to 1.02 a single image could be associated with each taxon, or a single footnote could be associated with each taxon, character or cell of the matrix. Now, multiple notes can be attached to each taxon, character, or cell of a matrix. Each note can have a comment, a reference, an image, and an author. Labels can be attached to the image. *To use this*, select the notes tool ( in the Character Matrix Editor. If you touch on a cell, a [Annotations window](#) will appear with the notes for that cell, if any. Buttons in this window can be used to delete and add notes and images, and to make additional notes windows. Behavior of the window can be modified using its Annotations menu. NOTE: the single footnote system still exists and is independent of this multiple notes system.
- **MRP matrices** can be generated from a set of trees, for use by Matrix Representation with Parsimony analyses to construct supertrees. To use this, select MRP Matrices from trees as a source of matrices (e.g., under Characters menu, Make New Matrix submenu, Other Choices).
- a **search facility** has been added to the Character Matrix editor and to the List windows. The cell of the matrix found by the search is highlighted and made visible. *To use this*, select Find String or Find Footnote from the Edit menu. Find String searches through the row names first (e.g., taxon names for the Character Matrix Editor), then the column names (e.g., character names), then the internal cells of the table (e.g., the character state codings). Find Footnote searches the footnotes of the Character Matrix and highlights cells containing the requested text in their footnotes. You can also search for stretches of sequences or state distributions matching the currently selected stretch using Select Same Sequence and Select Same Distribution of the Select menu of the Character Matrix Editor.
- More choices were added for **coloring character matrices**. Both text and background of a cell can now be colored according to properties of the cell. A paintbrush tool ()

was added to allow arbitrary coloring of cells of a character matrix.

- **taxa and characters can be moved** by a menu item "Move Selected [taxa or characters]To..." in the List of Taxa and List of Characters windows. Prior to 1.02 taxa and characters could be moved by clicking and dragging, but using this it was difficult to move them long distances through a matrix. *To use this*, select the taxa or characters to be moved in their respective List window, and select Move Selected from the List menu.
- Taxon selecting in the tree window has been modified, with a new taxon select tool (  ).
- Utilities to **modify taxon names** have been enhanced and added to the Character Matrix Editor in the Taxon Names submenu of the Matrix menu.
- Added **import/export of FASTA files**

## Bug Fixes

- (⚠) fixed calculation of Shared Partitions, which counted too many shared partitions between trees in some circumstances if the trees had different terminal taxa included
- resolved conflict in interpretation of tree descriptions that would result in Mesquite's interpreting trees from MrBayes and other programs as having reticulations whenever labels for internal nodes were duplicated. Mesquite now interprets these labels as cosmetic only, and hence not as indicating reticulations
- fixed bug involving the State Names Editor, by which a character might mistakenly receive a name duplicated from another character
- fixed bug in moving taxa or characters by dragging past the end of the matrix
- fixed bug in recoding characters by which state names and footnotes would not be adjusted
- fixed various other small bugs

---

## Version 1.01

### New Features

- added on-the-fly **filtering and transforming of trees**. Thus, trees being supplied from a file or simulation can be filtered according to various criteria or can be transformed with the available tree or branch length alterers. *To use this*, whenever a source of trees is needed, choose first "Filter Trees from Other Source" (if you want to select trees satisfying a criterion; more details [here](#)) or "Transform Trees from Other Source" (if you want to transform trees, e.g. reroot, scale branch lengths, etc.; more details [here](#)), then indicate what source of trees is to be filtered or transformed, and what filter or transformation is to be used.
- added module to **read trees directly from a file** one at a time; allows processing of much larger tree files (>50,000 trees). *To use this*, request "Trees Directly from File" as your source of trees for charts and other calculations. You can also build a tree block

from a sample of trees from a large file by selecting "Include partial contents" from the "Get File With Trees" submenu of the Taxa&Trees menu.

- available tree and branch length alterers can be applied to all trees in a tree block. *To use this*, go to the Utilities submenu of the List menu of the List of Trees window.
- added a new tree alterer: **outgroup rerooting** of trees. To rerooot trees using a selected set of outgroups, you first must select the outgroup taxa, then, choose the menu item "Root tree with selected taxa as outgroup", for instance in the Alter/Transform Tree submenu of the Tree menu of the tree window. More details are given on the page on [trees](#).
- added **integrated BLAST search**. *To use this*, go to a character matrix editor window showing a molecular matrix. Select one or more stretches of sequence and choose BLAST Search from the Search submenu of the Matrix menu. Mesquite will then send a request to your web browser to perform a BLAST search at NCBI's BLAST server, one request for each of the selected stretches of sequence. The results will be available in your web browser.
- added -w (**windowless**) and -b (**backgroundable**) flags for running long Mesquite jobs in the background in Unix from the command line
- improved user interface of tables (List Windows, Character Matrix Editor), including navigation with arrow keys and more informative explanations and footnotes
- Added **autotab** to the Character Matrix Editor. With this, the edited cell shifts automatically to the right or downward when a state is entered. *To use this*, select the Select and Type tool (with the letter "K") and use its drop down menu to indicate autotab right or autotab down. Autotab functions only when the Select and Type tool is the active tool.
- selection of taxa, characters and trees is now remembered when files are saved
- calculations of charts and Trace Character Over Trees can now be stopped part way to show partial results

## Bug fixes

- fixed tree printing bug by which node numbers and branch lengths always appeared
- fixed bug by which pressing the Delete key while editing in a table would be misinterpreted as a requesting to delete a character, taxon or other object
- fixed bugs in NEXUS file reading and writing (concerning SYMBOLS list, blank character names, tokenization with tab or square brackets, near-duplicate taxa blocks in same file)
- fixed bugs in State Names Editor and Character Matrix Editor concerning footnote display and editing
- fixed bug with Mac OS X 10.3 by which dialog boxes with lists always chose first item (this was not a bug in Mesquite, but rather in Mac OS X, but we have found a workaround)
- fixed bug with Windows by which pie diagrams would be misdrawn as entirely black for traced characters in the Balls and Sticks tree drawing mode (this was not a bug in Mesquite, but rather in Windows/Java, but we have found a workaround)

- several other smaller, less notable bugs were also fixed
-

# Credits

## History

Mesquite was begun in 1997, but its roots go deeper, to the initial development in 1985 of [MacClade](#), which even from the start allowed interactive manipulation of trees and interpretation of character evolution. As features were added to MacClade through the years, most from Wayne Maddison in early versions (1 and 2) and from David Maddison in later versions (2, 3 and 4), MacClade developed an exploratory approach to phylogenetic calculations with a distinctive user interface. After MacClade version 3 was released in 1992, the coauthors worked to ready version 4. Almost all of the many new features in version 4, including the new facilities for molecular sequence editing, were the result of David's efforts. Wayne's efforts on MacClade 4 involved an attempt to graft a modular architecture onto MacClade to allow plug-ins so that its capabilities could be extended. This, we hoped, would allow us and other programmers to add many new tree-based analyses to MacClade. After about a year of work on this, it became clear that grafting this new architecture on to an existing program was not going to work. MacClade was then returned to its original, non-modular state, and it was within this more traditional framework that David completed MacClade 4.

In order to build the desired modular architecture, Wayne had to start from scratch, and so a new project was born in July of 1997. The very first prototype, after one day of work, can be seen [here](#) (for the first few days it was called BeanTree, before it became known as Mesquite). Mesquite's vision, exploratory nature, and its user interface borrow extensively from ideas developed in MacClade, but the underlying architecture is quite different. Thus, Mesquite contains a mix of features borrowed directly from MacClade, features we had wanted to put into MacClade but couldn't (e.g., coordinated selection of objects, Trace Character over Trees, likelihood reconstructions), and newly conceived features.



The subsequent chronology of the Mesquite project is:

- August 1998 - first public demonstration (Cambridge University)
- July 1999 - limited seeding to a few developers (prototype version 0.9.2)
- August 1999 - project web page on-line (currently at <http://mesquiteproject.org/>)
- 29 September 1999 - broader release to developers (prototype version 0.9.5)
- early 2000 - passed 100,000 lines of total code, and 200 total modules.
- 26 June 2000 - Mesquite introduced at Evolution meetings, Bloomington, Indiana

(prototype version 0.9.28)

- 14 March 2001 - first public beta version (version 0.95.80)
- 2 April 2001 - public beta version (version 0.96)
- 24 July 2001 - version 0.98 with source code released
- 21 August 2002 - version 0.99 released
- 14 September 2002 - version 0.991 released
- 27 September 2002 - version 0.992 released (internal version (build) d24)
- 10 January 2003 - version 0.993 released (build d42)
- 7 February 2003 - version 0.994 released (build d51)
- 21 May 2003 - version 0.995 released (build e23)
- 21 June 2003 - version 0.996 released (build e30)
- 22 September 2003 - **version 1.0 released** (build e58)
- 14 January 2004 - **version 1.01 released** (build e80; build e81 released 17 January to fix cosmetic bug)
- 6 May 2004 - **version 1.02 released** (build g6; build g7 released 12 May to fix non-substantive bug)
- 1 July 2004 - **version 1.03 released** (build g19)
- 1 September 2004 - **version 1.04 released** (build g21)
- 24 September 2004 - **version 1.05 released** (build g24)
- 30 August 2005 - **version 1.06 released** (build g97)
- 18 May 2006 - **version 1.1 released** (build h60; build h61 released 23 May to fix non-substantive bugs)
- 21 June 2006 - **version 1.11 released** (build h64) [20th anniversary of first MacClade release!]
- 23 September 2006 - **version 1.12 released** (build h66)
- 21 September 2007 - **version 2.0 released** (build i68; build i69 released 24 Sept to fix non-substantive bug)

From July 1997 through October 2000, the architectural design, programming and documentation for the basic Mesquite libraries and modules was done by Wayne Maddison, with occasional input from David Maddison. David entered the project in earnest in November 2000. Other packages of modules for the Mesquite system are due to other authors: for instance, the Rhetenor package of morphometrics modules is by Eric Dyreson and Wayne Maddison. In 2006 Peter Midford joined the development team full-time thanks to the support of the CIPRES project. He has written code for new likelihood methods for understanding character evolution and diversification (e.g., Pagel's 1994 correlation test and Maddison, Midford and Otto's BiSSE method) and has helped with the core architecture. Danny Mandel also joined the team, and was instrumental in arranging Mesquite's transition to a new development system (Eclipse) and in improving the logistics of the release of new versions. In 2007 Jeff Oliver started his contributions to the tree manipulation and display code, the manual, and other aspects.

## Acknowledgments

Mesquite has been developed with the assistance of a Fellowship to WPM from the David and

Lucile Packard Foundation, an NSERC Discovery grant to WPM, NSF support via the [CIPRES project](#), and the patience of our families.

The following are gratefully acknowledged for direct assistance with Mesquite's programming or source code. David Swofford helped with the implementation of the likelihood calculations, and supplied code for the optimization routines. John Huelsenbeck gave us portions of the code used in calculation of change probabilities for the Ti/Tv rate matrix model. Lars Rosengreen helped us prepare the code for compilation by means other than Codewarrior (see his [website](#)).

We offer our thanks to the many users who have given suggestions and bug reports. We would like to single out Michel Laurin and Jonathan Coddington for extensive feedback. Maureen O'Leary gave advice on the use of Mesquite for large morphological data matrices.

Mesquite has benefitted from support to Wayne Maddison and David Maddison by the University of Arizona and the University of British Columbia, and the Wissenschaftskolleg zu Berlin.

