Inferring Evolutionary Trees with PAUP*

PAUP* version 4 (Swofford, 2002) is a widely used software package for inferring evolutionary trees. PAUP* implements criterion-based methods such as parsimony (see Background Information), maximum likelihood (see Background Information), and least-squares, and provides a variety of search strategies for estimating phylogenies. PAUP* can also reconstruct trees using algorithms such as UPGMA and neighbor joining (*UNIT 6.3*). In addition to reconstructing phylogenies, other features of the program include diagnosing characters and inferring ancestral states on a phylogenetic tree and testing the robustness of phylogenetic trees using several descriptive and statistical techniques.

One of PAUP*'s strengths is the rich assortment of phylogenetic methods that can be implemented within the program. However, for people not familiar with the field of phylogenetic inference, the number of available methods and options in PAUP* may seem overwhelming. This unit is designed to provide an entry-level overview of how to implement some of the most basic phylogenetic methods available in PAUP*. The Basic Protocol uses the parsimony method to infer evolutionary trees and can be used for DNA, RNA, and protein sequences. The Alternate Protocol uses a model-based, statistical approach—maximum likelihood—to infer evolutionary trees for DNA and RNA sequences only. For obvious reasons, the authors have chosen not to describe in this unit all the possible analysis options available in PAUP*. In addition, the working details of the methods implemented in some of the examples will not be described; instead, citations to relevant and accessible references will simply be provided. After reading this unit, one should have enough experience with the program to use other available methods that are described more completely in the current program documentation.

BASIC PROTOCOL

USING PAUP* TO INFER PARSIMONY TREES FROM DNA SEQUENCES

PAUP* can be used to infer evolutionary trees and perform associated analyses in both interactive and batch mode. As the name implies, the interactive mode requires a user's input at various stages of the analysis, whereas batch mode allows the user to set up analyses in advance and run them without being in attendance. While both methods are useful, this protocol will focus mostly on how to operate PAUP* in the interactive mode. See Suggestions for Further Analysis in the Commentary for guidelines for running PAUP* in batch mode.

Necessary Resources

Hardware

PAUP* is compatible with most modern hardware configurations including Apple Macintosh (Power PC- and 68K-based processors), Intel, AMD, and a number of Unix and Linux workstations (e.g., Sun, Alpha, IBM, SGI, Power PC-based, i386-based, and others)

Software

PAUP* is distributed by Sinauer Associates at *http://www.sinauer.com*. Three versions of the program are available: Macintosh, Windows, and Portable (see Table 6.4.1). These versions provide identical analytical capabilities, but differ in the way these analyses are controlled. The Macintosh version supports a full graphical user interface, which allows the user to execute commands via menus and the command line, while the Windows and Portable versions are almost entirely command-line driven. Some menu functions are available in the

Contributed by James C. Wilgenbusch and David Swofford

Current Protocols in Bioinformatics (2003) 6.4.1-6.4.28

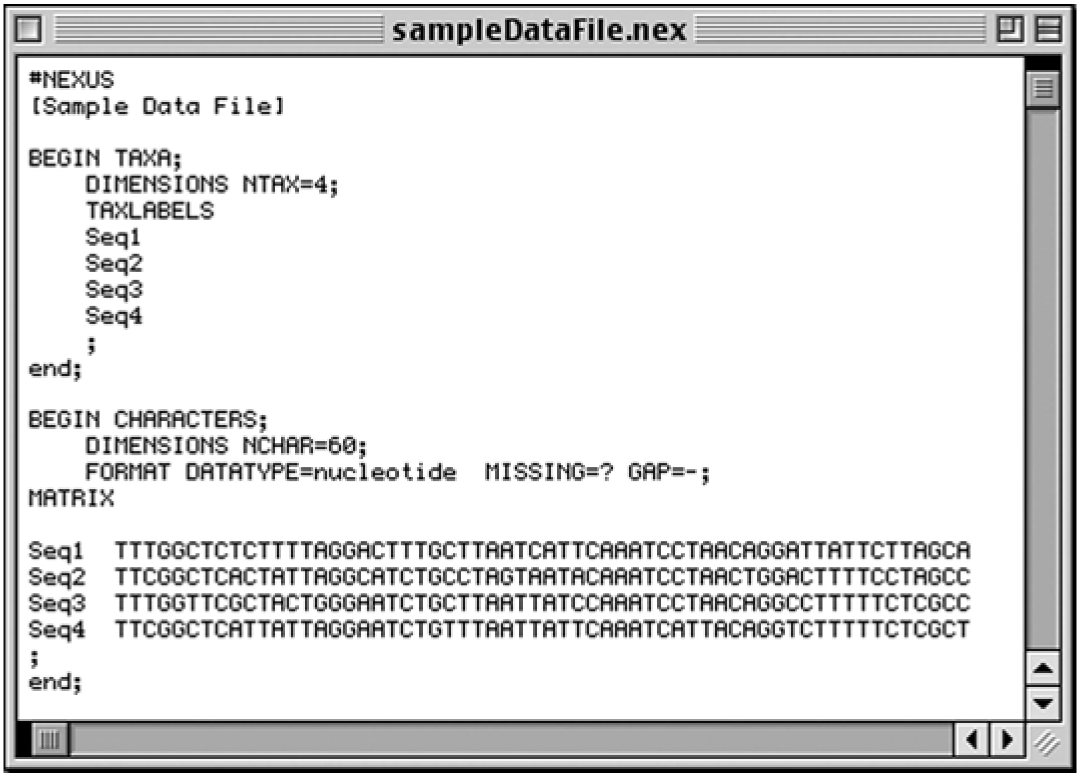
Table 6.4.1 Platforms that are Currently Supported by PAUP* Version 4.0

Interface type	Processor	Operating System
Macintosh	G3/G4	Mac OS X (classic)
	PowerPC and 68k-based	Mac 7.0 or later
Windows	i386-based	Windows 95/98/ME
		Windows NT/2000
Portable	i386-based	Linux 2.0 or later/FreeBSD
	PPC	Linux 2.0 or later
	Sparc and UltraSparc	Solaris
	Alpha	Digital Unix/True64
	MIPS II and III	Irix
	RS6000	AIX
	Possibly others	Possibly others

Windows version; however, the functions are mostly restricted to file “open” and “edit” operations. Because the command-line interface is available in all three versions, it is used for the following examples. The location of the corresponding menu item is given in the PAUP* command reference documentation (available on the PAUP* 4.0 distribution disk and on the PAUP* Web site at *http://paup.csit.fsu.edu/commandref.pdf*). In addition, several sources already offer an overview of PAUP* using the Macintosh menu interface (Hall, 2001; also see PAUP* quick-start tutorial, *http://paup.csit.fsu.edu/quickstart.pdf*). Installation instructions are included with the software.

Files

Data files for PAUP* are standard text files, which adhere to the NEXUS file specifications. The NEXUS format was designed by Maddison et al. (1997) to facilitate the interchange of input files between programs used in phylogeny and classification. The text in a NEXUS file is arranged into blocks, which are delimited by the words `begin` and `end`. The text immediately following the word `begin` defines the block type. For example, the `TAXA` and `CHARACTERS` blocks shown in Figure 6.4.1 define a simple data set composed of four taxa (sequences) and 60 nucleotide characters (sites). PAUP* supports several predefined data types: DNA, RNA, nucleotide (DNA or RNA), protein, and standard. The set of predefined character-state symbols are `ACGT` for the data type DNA, `ACGU` for RNA, the standard one-letter amino acid codes for protein, and `01` for standard. In addition, standard ambiguity codes for the molecular data types are implemented by predefined “equate” macros. Additional character states other than those represented by the predefined data types can be specified using the `SYMBOLS` subcommand (see the PAUP* command reference documentation at the above URL). PAUP* data files must start with the character string `#NEXUS`. Comments can be included in a data file by enclosing them in square brackets, as shown in Figure 6.4.1. PAUP* 4.0 is also capable of translating other file formats to the NEXUS format. Supported formats include PHYLIP (*UNIT 6.3*), Hennig86, GCG/Pileup (*UNIT 3.6*), MEGA, NBRF-PIR, FreqPars, and text (space and tab-delimited). The Support Protocol below details using PAUP* to import non-NEXUS data files. The NEXUS file used for this protocol can be found in the application subdirectory labeled `Sample-NEXUS-data` and also on the *Current Protocols in Bioinformatics* Web site at *http://www3.interscience.wiley.com/c_p/cpbi_sampledatafiles.htm*.



```
#NEXUS
[Sample Data File]

BEGIN TAXA;
    DIMENSIONS NTAX=4;
    TAXLABELS
    Seq1
    Seq2
    Seq3
    Seq4
    ;
end;

BEGIN CHARACTERS;
    DIMENSIONS NCHAR=60;
    FORMAT DATATYPE=nucleotide  MISSING=? GAP=-;
MATRIX

Seq1  TTTGGCTCTCTTTTAGGACTTTGCTTAATCATTCAAATCCTAACAGGATTATTCTTAGCA
Seq2  TTCGGCTCACTATTAGGCATCTGCCTAGTAATACAAATCCTAACTGGACTTTTCCTAGCC
Seq3  TTTGGTTCGCTACTGGGAATCTGCTTAATTATCCAAATCCTAACAGGCCTTTTTCTCGCC
Seq4  TTCGGCTCATTATTAGGAATCTGTTTAATTATTCAAATCATTACAGGTCTTTTTCTCGCT
;
end;
```

Figure 6.4.1 A sample NEXUS data set composed of 4 sequences and 60 nucleotide characters.

NOTE: Due to formatting constraints, some commands given in the following examples span multiple lines. However, when entering commands at the `paup>` prompt or into the command-line interface, all of the text preceding the semicolon should be entered on the same line.

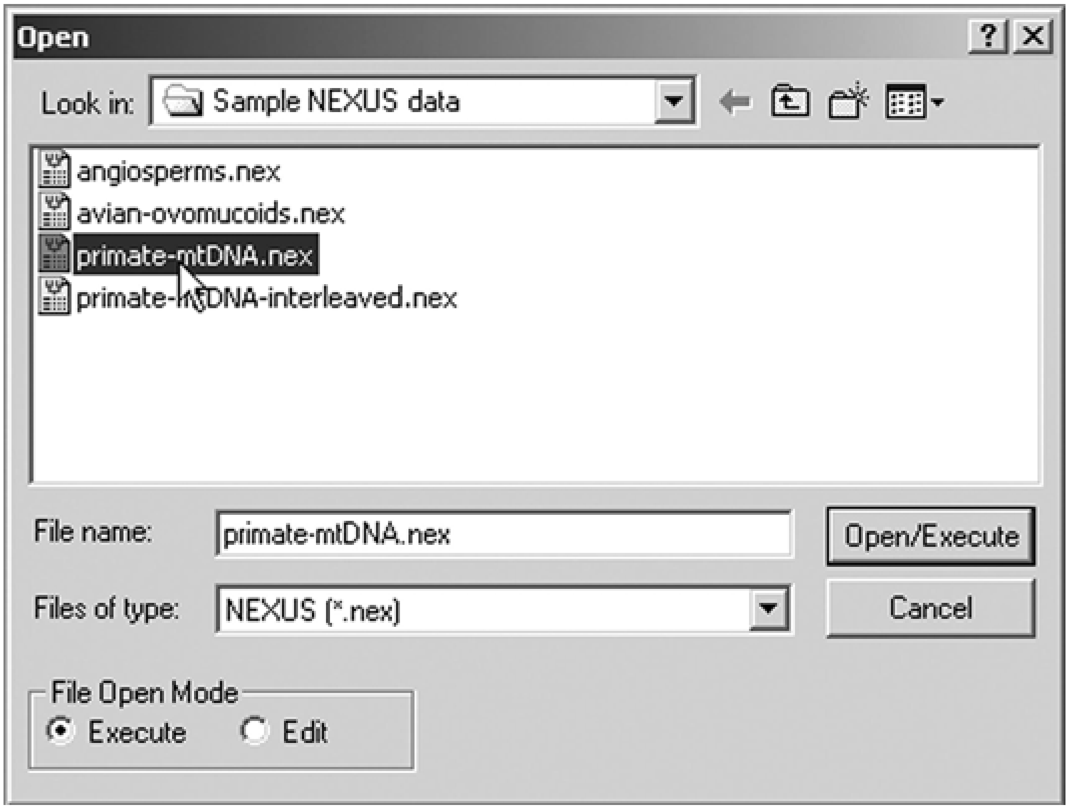
Input the sequences

1. Start PAUP*.

   The way in which PAUP is started will depend on the platform the user is using. To start PAUP* on a Windows or Macintosh machine, locate the application icon and double-click it. PAUP* will automatically launch the Open File dialog box when it is first started (Fig. 6.4.2). For the Portable interface, the command* `paup` *should be in one's path and the command should be linked to the PAUP* executable (see the README file in the program installation archive and* APPENDIX 1C*). In this case, start the program by changing directories to the subdirectory called* `Sample-NEXUS-files` *included in the PAUP* archive and then type the command* `paup` *at the shell prompt.*

2. Execute the data set.

   Macintosh and Windows users should select the file named `primate-mtDNA.nex` *from the Open File dialog box and click the Open/Execute button. Users of the Portable interface should type the command* `execute primate-mtDNA.nex;` *at the* `paup>` *prompt. After executing the sample file, PAUP* will display comments and some general information about the data (Fig. 6.4.3). For this example, the source of the data set is given, followed by comments included in the NEXUS file, a section reporting the dimensions of the data matrix, the type of data, and several other characteristics of the data set. At this point, no analyses will have been conducted; PAUP* has simply processed the data and is now waiting to be told what to do next.*



Look in: Sample NEXUS data

angiosperms.nex
avian-ovomucoids.nex
primate-mtDNA.nex
primate-mtDNA-interleaved.nex

File name: primate-mtDNA.nex

Files of type: NEXUS (*.nex)

File Open Mode
● Execute ○ Edit

Figure 6.4.2 The Open File dialog box automatically launched when the Windows and Macintosh versions of PAUP* are first started.

Manage the input data

3. Specify the sites to be included in the analysis. For example, exclude noncoding regions.

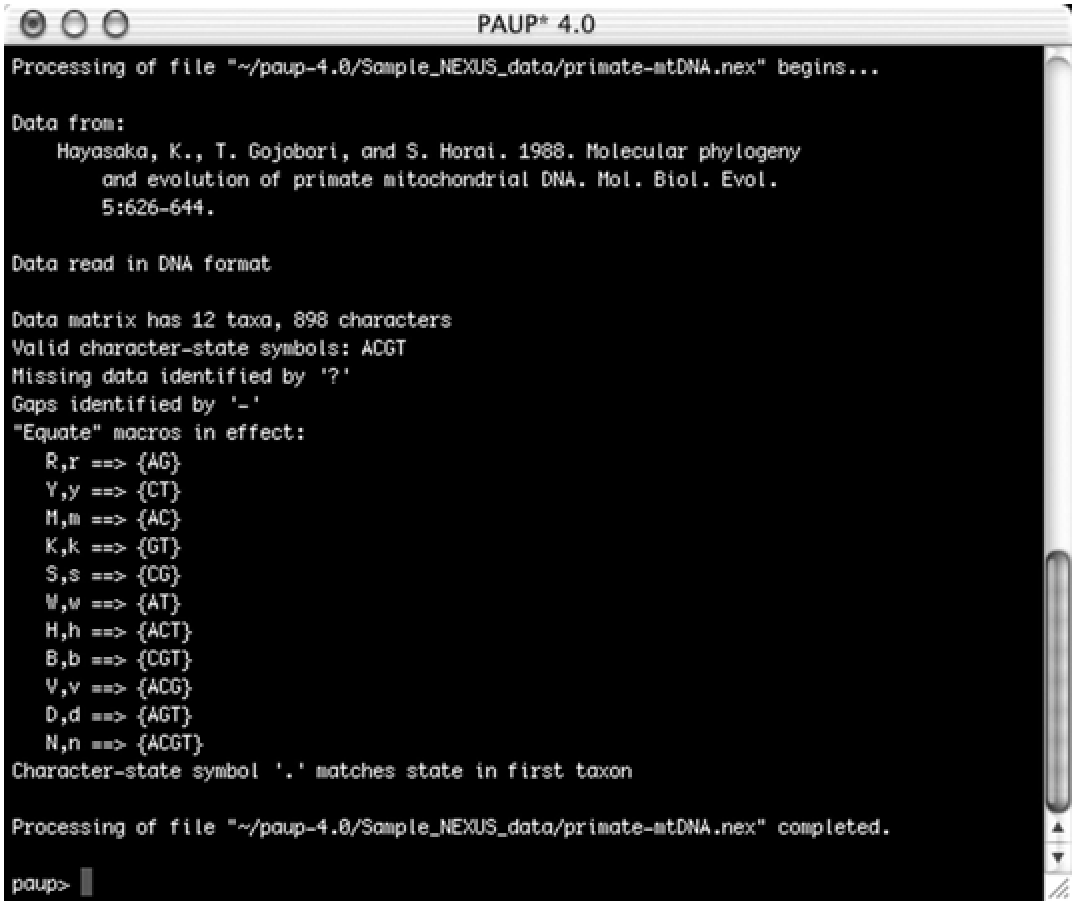
```
include coding/only;
```

Windows and Macintosh users will enter the `include` *command in the command-line interface located at the bottom of the main display window (Fig. 6.4.4). Portable-version users will enter commands at the* `paup>` *prompt.*

PAUP provides several ways to restrict an analysis to a subset of the characters included in a data matrix without permanently removing the characters from the NEXUS file. The sample data set includes protein coding and noncoding regions of primate mitochondrial DNA. Suppose one wishes to analyze only the coding regions of the data. The* `/only` *option is included when only the characters listed after the* `include` *command are to be included in the analysis. The characters belonging to these regions have already been identified in the sample file using the* `charset` *command (Fig. 6.4.5). Character sets simplify certain procedures by allowing users to refer to a group of characters with a single name (see PAUP* command reference documentation; http://paup.csit.fsu.edu/commandref.pdf). PAUP* also includes several predefined character sets that may be used with specific types of data (Table 6.4.2).*

4. Specify sequences (taxa) to be used. For example, delete all but five hominoid and three non-hominoid sequences.

```
delete M._mulatta M._fascicularis M._sylvanus
Tarsius_syrichta;
```



```
Processing of file "~/paup-4.0/Sample_NEXUS_data/primate-mtDNA.nex" begins...

Data from:
    Hayasaka, K., T. Gojobori, and S. Horai. 1988. Molecular phylogeny
        and evolution of primate mitochondrial DNA. Mol. Biol. Evol.
        5:626-644.

Data read in DNA format

Data matrix has 12 taxa, 898 characters
Valid character-state symbols: ACGT
Missing data identified by '?'
Gaps identified by '-'
"Equate" macros in effect:
   R,r ==> {AG}
   Y,y ==> {CT}
   M,m ==> {AC}
   K,k ==> {GT}
   S,s ==> {CG}
   W,w ==> {AT}
   H,h ==> {ACT}
   B,b ==> {CGT}
   V,v ==> {ACG}
   D,d ==> {AGT}
   N,n ==> {ACGT}
Character-state symbol '.' matches state in first taxon

Processing of file "~/paup-4.0/Sample_NEXUS_data/primate-mtDNA.nex" completed.

paup>
```

Figure 6.4.3 The PAUP* main display after executing the sample data set.

Equivalently:

```
undelete hominoids lemur_catta macaca_fuscata
saimiri_sciureus/only;
```

The five hominoids included in the example data set (Homo sapiens, Pan, Gorilla, Pongo, and Hylobates) have already been identified using the `taxset` *command (Fig. 6.4.5). In the same way that a* `charset` *can be used to identify a group of characters by a single name,* `taxset` *can be used to identify a group of taxa. The* `delete` *or* `undelete` *commands can be used to perform this operation. Notice that the* `/only` *option is used with the* `undelete` *command so that other taxa already in the data set are not also included in subsequent analyses. There are no predefined taxon sets; the user must define them manually as shown in Figure 6.4.5 (see PAUP* command reference documentation; http://paup.csit.fsu.edu/commandref.pdf). Spaces in taxon names are identified by using an underscore character or by enclosing the name in single quotes. PAUP* does not pay attention to the character case in taxon labels.*

Select an optimality criterion and define assumptions

5. Set the optimality criterion to parsimony.

```
set criterion=parsimony;
```

In this case, `parsimony` *will be used as the optimality criterion for selecting a tree (see Background Information). Because parsimony is the default criterion used by PAUP* when the program is first started, the* `set` *command given above is only necessary if the default*

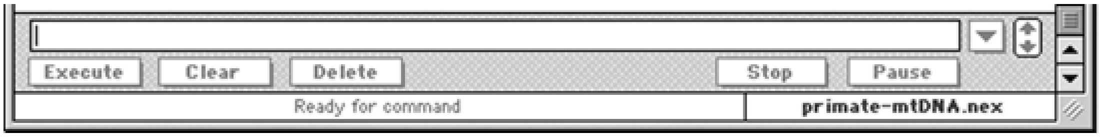
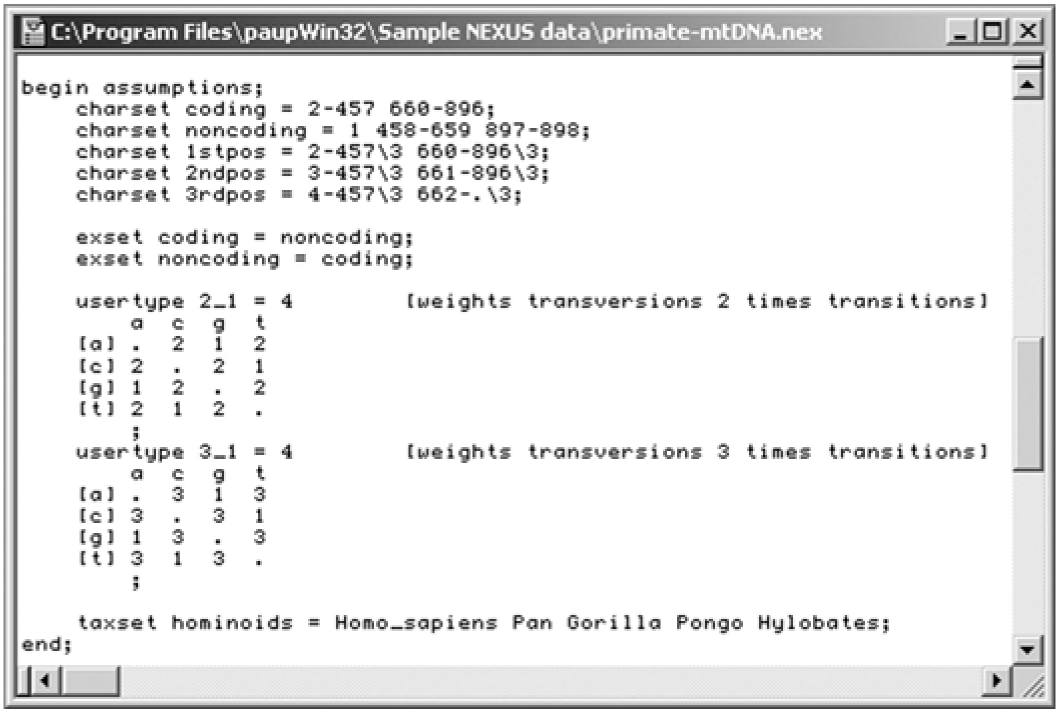


Figure 6.4.4 The command-line interface located at the bottom of the Macintosh and Windows main display window.


C:\Program Files\paupWin32\Sample NEXUS data\primate-mtDNA.nex

```
begin assumptions;
    charset coding = 2-457 660-896;
    charset noncoding = 1 458-659 897-898;
    charset 1stpos = 2-457\3 660-896\3;
    charset 2ndpos = 3-457\3 661-896\3;
    charset 3rdpos = 4-457\3 662-.\3;

    exset coding = noncoding;
    exset noncoding = coding;

    usertype 2_1 = 4        [weights transversions 2 times transitions]
        a  c  g  t
    [a] .  2  1  2
    [c] 2  .  2  1
    [g] 1  2  .  2
    [t] 2  1  2  .
        ;
    usertype 3_1 = 4        [weights transversions 3 times transitions]
        a  c  g  t
    [a] .  3  1  3
    [c] 3  .  3  1
    [g] 1  3  .  3
    [t] 3  1  3  .
        ;

    taxset hominoids = Homo_sapiens Pan Gorilla Pongo Hylobates;
end;
```

Figure 6.4.5 The `assumptions` block included at the end of the sample data set. The block contains the character and taxa sets and the user-defined character types that are used in the example analysis.

settings have been altered. If unsure, it is a good idea to go ahead and issue this command. The Alternate Protocol describes how to set up a search using the maximum-likelihood criterion.

6. Consider setting character weighting. For example:

```
weights 2:2ndpos;
```

weights all transformations at nucleotides in the `2ndpos` charset (defined in Fig. 6.4.5) by a factor of two. All other transformations have the default weight of 1.

Under the default parsimony criterion used in PAUP, transformations from one character state to another are given the same score no matter which site in the data matrix is being examined. Under some circumstances, it may be perfectly appropriate to leave the default weighting settings unchanged. For example, it may be unnecessary to weight sites differentially when a parsimony search is used only as a way to get a tree quickly for use in the parameterization of a subsequent maximum-likelihood search (see Alternate Protocol, step 2). Also, unlike maximum likelihood, the parsimony criterion lacks an objective function that may be used to select from among the many possible weighting schemes (see*

Table 6.4.2 Predefined Characters Sets and Their Corresponding Data Types

Data type	Character set name	Definition
All data types	`Constant`	Invariant characters
	`Gapped`	Characters with a gap for at least one taxon
	`Missambig`	Characters with a gap or ambiguous character for at least one taxon
	`Allmissing`	Characters with a gap for all the taxa
	`Remainder`	Characters not previously referenced in the command
	`Uninf`	Characters that are constant as well as unique to a single sequence (autapomorphic)
DNA, RNA, and Nucleotide[a]	`Pos1`	Characters defined by current `CodonPosSet` as first positions
	`Pos2`	Characters defined by current `CodonPosSet` as second positions
	`Pos3`	Characters defined by current `CodonPosSet` as third positions
	`Noncoding`	Characters defined by current `CodonPosSet` as non-protein-coding sites

[a] Only if a `Codons` block containing a `CodonPosSet` is supplied (see PAUP* command reference documentation; *http://paup.csit.fsu.edu/commandref.pdf*).

Table 6.4.3 List of Parsimony Variants Available in PAUP*

Parsimony variants implemented in PAUP*	Descriptive name	Command-line syntax
Fitch (Fitch, 1971)	Unordered	`unord`
Wagner (Kluge and Farris, 1969; Farris, 1970)	Ordered	`ord`
Camin-Sokal (Camin and Sokal, 1965)	Irreversible (normal or reversed)	`irrev.up` or `irrev.dn`
Dollo (Farris, 1977)	Dollo (normal or reversed)	`dollo.up` or `dollo.dn`
Generalized (Sankoff, 1975)	User-defined	`usertype`

Background Information). Therefore, without a priori information regarding the underlying properties of the sequences, it may be best to use equal weights for all sites.

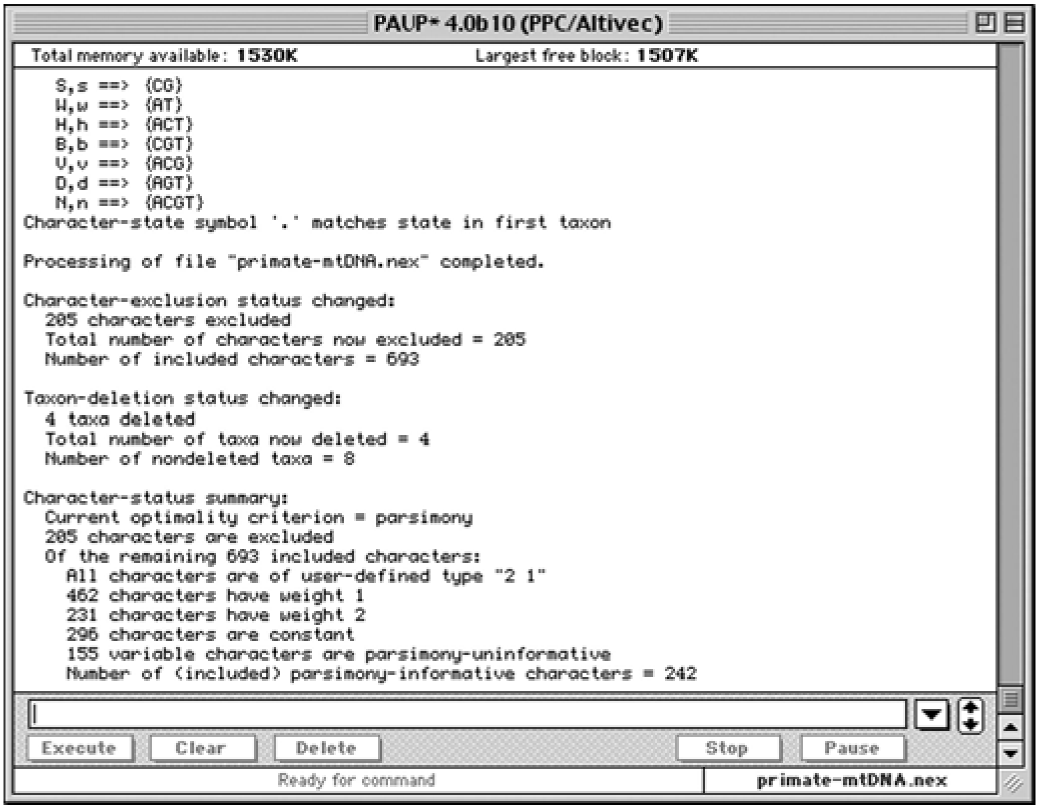
In other situations, one may prefer to give greater weight to transformations at specific sites. For example, the data to be analyzed in this example are composed of protein-coding sequences. Because transformations at second-codon positions are more highly conserved, transformations at these positions will be given greater weight relative to transformations at first- and third-codon positions. In the example here, weighting second-position codons is relatively simple since codon positions have already been identified in the sample file using the `charset` *command (Fig. 6.4.5). Another way to assign weights to a set of sites is to explicitly list the site number following each weight. For example the command:*

```
weights 3: 2 5 8 11, 2:1 4 7;
```

will weight transformations at the nucleotide positions 2, 5, 8, and 11 by a factor of three and positions 1, 4, and 7 by a factor of two.

7. Consider setting character types.

```
ctype 2_1:all;
```



PAUP* 4.0b10 (PPC/Altivec)

Total memory available: **1530K** Largest free block: **1507K**

```
   S,s ==>  {CG}
   W,w ==>  {AT}
   H,h ==>  {ACT}
   B,b ==>  {CGT}
   V,v ==>  {ACG}
   D,d ==>  {AGT}
   N,n ==>  {ACGT}
Character-state symbol '.' matches state in first taxon

Processing of file "primate-mtDNA.nex" completed.

Character-exclusion status changed:
  205 characters excluded
  Total number of characters now excluded = 205
  Number of included characters = 693

Taxon-deletion status changed:
  4 taxa deleted
  Total number of taxa now deleted = 4
  Number of nondeleted taxa = 8

Character-status summary:
  Current optimality criterion = parsimony
  205 characters are excluded
  Of the remaining 693 included characters:
    All characters are of user-defined type "2 1"
    462 characters have weight 1
    231 characters have weight 2
    296 characters are constant
    155 variable characters are parsimony-uninformative
    Number of (included) parsimony-informative characters = 242
```

Figure 6.4.6 Character-status summary after several data management operations.

PAUP allows the user to explore different parsimony variants by assigning user-defined character types (Table 6.4.3) for each character included in the data matrix. By default, PAUP* assumes that all character states are unordered and that the transformation cost from one state (nucleotide) to any other state is equal to one. For the same reasons described in step 6, the default settings may be justified, in which case the user can simply skip this step altogether. For this example, a user-defined character type will instead be declared that assigns a higher cost to transversions (changes between a purine and a pyrimidine) than to transitions (changes between two purines or changes between two pyrimidines). More specifically, the user-defined character type is set up so that the cost of transversions is twice that of transitions. A step matrix that defines this relationship in a way that PAUP* can understand is already given in the data file (Fig. 6.4.5). The command given in this step forces PAUP* to apply the transformation cost to all of the characters currently included in the analysis.*

8. Check the current character settings.

```
cstatus;
```

Before going on to the next set of steps ("Search for a tree") it is generally good practice to check the status of the characters to be included in the search. Searching for an optimal tree can be a time-consuming operation, so a little extra caution at this stage of the analysis can save time in the long run. At the bottom of the display window shown in Figure 6.4.6, a summary of the current character status is given. The output shows that the optimality criterion is set to parsimony (step 5), 205 characters included in the input matrix will not be used in the analysis (the noncoding regions excluded in step 3), all of the remaining characters are assigned the user-defined type `2 1` *(step 7), and transformations at 231 character sites (i.e., second codon positions) will be weighted twice that of the others (step 6). The remaining information summarizes several qualities of the data set after the* *assumptions have been set. A more detailed summary for each character in the data matrix can be obtained by using the* `cstatus` *command and turning the* `full` *option on. For example:*

```
cstatus full;
```

Search for a tree

9. Select a search strategy.

PAUP provides two general classes of methods for searching for optimal trees; exact and heuristic. Exact methods guarantee that the optimal tree(s) will be found, but may require prohibitive amounts of computer time for data sets composed of more than 11 to 20 sequences. On the other hand, while not guaranteeing that the optimal tree will be found, heuristic methods require far less computer time. Because many data sets compiled today are composed of more than 20 sequences, this example will focus on the options needed to conduct a heuristic search. Users interested in conducting an exact search should review the* `alltrees` *and* `bandb` *commands described in the current PAUP* command reference documentation.*

In addition to criterion-based methods for reconstructing evolutionary trees, PAUP can implement neighbor joining and UPGMA clustering algorithms. These methods have the advantage of being extremely fast even for very large data sets (e.g., 100 or more sequences). However, a drawback of these methods is that they do not explicitly attempt to optimize any particular objective function and therefore do not allow comparisons of preselected trees or searches for sets of trees that include both optimal and near-optimal trees. Since neighbor joining and UPGMA are discussed in UNIT 6.3, these methods will not be discussed here, except to say that more information regarding these methods can be found in the current PAUP* command reference documentation (see commands* `upgma` *and* `nj`*).*

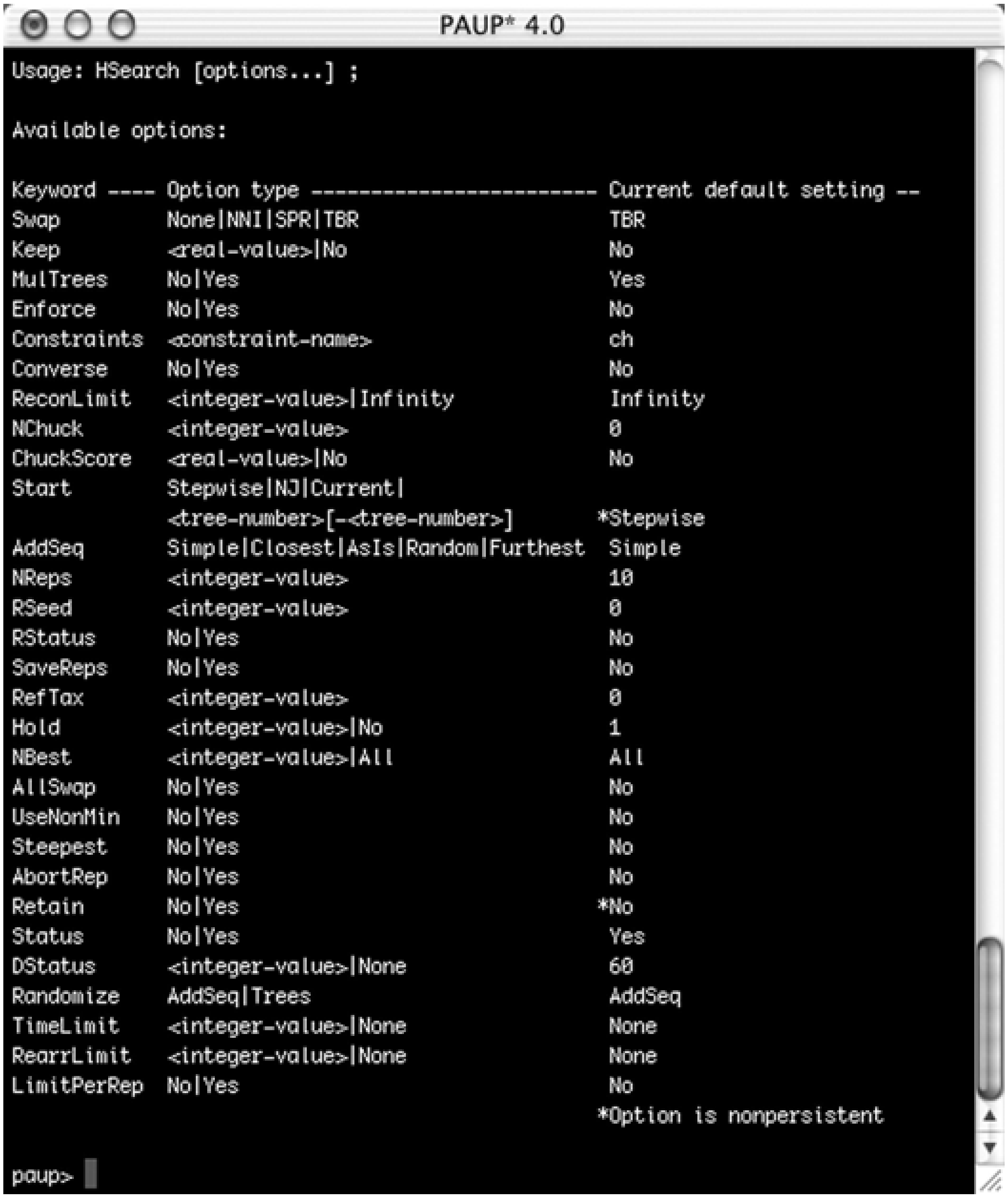
10. Select the heuristic search options.

```
hsearch ?;
```

Typing a command name followed by a question mark will generate a complete list of the options available under that command. While the list of available options under the "`hsearch`*" command is extensive (Fig. 6.4.7), depending on the data set being analyzed, it might only be necessary to alter the default settings for a small set of the available options. This example will focus entirely on three of the options listed in Figure 6.4.7—*`start`*,* `addseq`*, and* `swap`*. Information regarding* `hsearch` *options not discussed in this protocol can be found in the PAUP* command reference documentation.*

In general terms, a heuristic search in PAUP can be divided into two stages. The first stage involves getting a starting tree. The main options that deal with this stage of the search are* `start` *and* `addseq`*. By default, PAUP* uses the stepwise-addition algorithm (e.g., Farris, 1970) to get a starting tree. Other ways to get a starting tree include the neighbor-joining algorithm (*`start=nj`*; UNIT 6.3) and simply selecting one or more of the trees currently in memory (*`start=current` *or* `start=`*a range of tree numbers). For this example, the stepwise-addition option (the default) will be used to get a starting tree. The stepwise-addition option requires that the order in which sequences are joined together be specified, to construct a starting tree. PAUP* implements five addition sequence algorithms. Unfortunately, no one algorithm works best for all data sets (see Swofford et al., 1996). In this example, the random addition sequence algorithm will be used (see step 11) to construct multiple starting trees (the default is ten), each of which is then passed on to the second stage of the heuristic search, branch swapping.*

In simple terms, the second stage attempts to find better trees by evaluating rearrangements of the starting tree. PAUP implements three branch-swapping algorithms listed here in ascending order of effectiveness: nearest-neighbor interchanges (option* `nni`*), subtree pruning-regrafting (option* `spr`*), and tree bisection and reconnection (option* `tbr`*). As might be expected, effectiveness comes at the cost of increased search effort. For this example, branches will be swapped using the subtree pruning-regrafting algorithm. For a*



```
Usage: HSearch [options...] ;

Available options:

Keyword ---- Option type ------------------------ Current default setting --
Swap         None|NNI|SPR|TBR                      TBR
Keep         <real-value>|No                       No
MulTrees     No|Yes                                Yes
Enforce      No|Yes                                No
Constraints  <constraint-name>                     ch
Converse     No|Yes                                No
ReconLimit   <integer-value>|Infinity              Infinity
NChuck       <integer-value>                       0
ChuckScore   <real-value>|No                       No
Start        Stepwise|NJ|Current|
             <tree-number>[-<tree-number>]        *Stepwise
AddSeq       Simple|Closest|AsIs|Random|Furthest   Simple
NReps        <integer-value>                       10
RSeed        <integer-value>                       0
RStatus      No|Yes                                No
SaveReps     No|Yes                                No
RefTax       <integer-value>                       0
Hold         <integer-value>|No                    1
NBest        <integer-value>|All                   All
AllSwap      No|Yes                                No
UseNonMin    No|Yes                                No
Steepest     No|Yes                                No
AbortRep     No|Yes                                No
Retain       No|Yes                               *No
Status       No|Yes                                Yes
DStatus      <integer-value>|None                  60
Randomize    AddSeq|Trees                          AddSeq
TimeLimit    <integer-value>|None                  None
RearrLimit   <integer-value>|None                  None
LimitPerRep  No|Yes                                No
                                                  *Option is nonpersistent

paup>
```

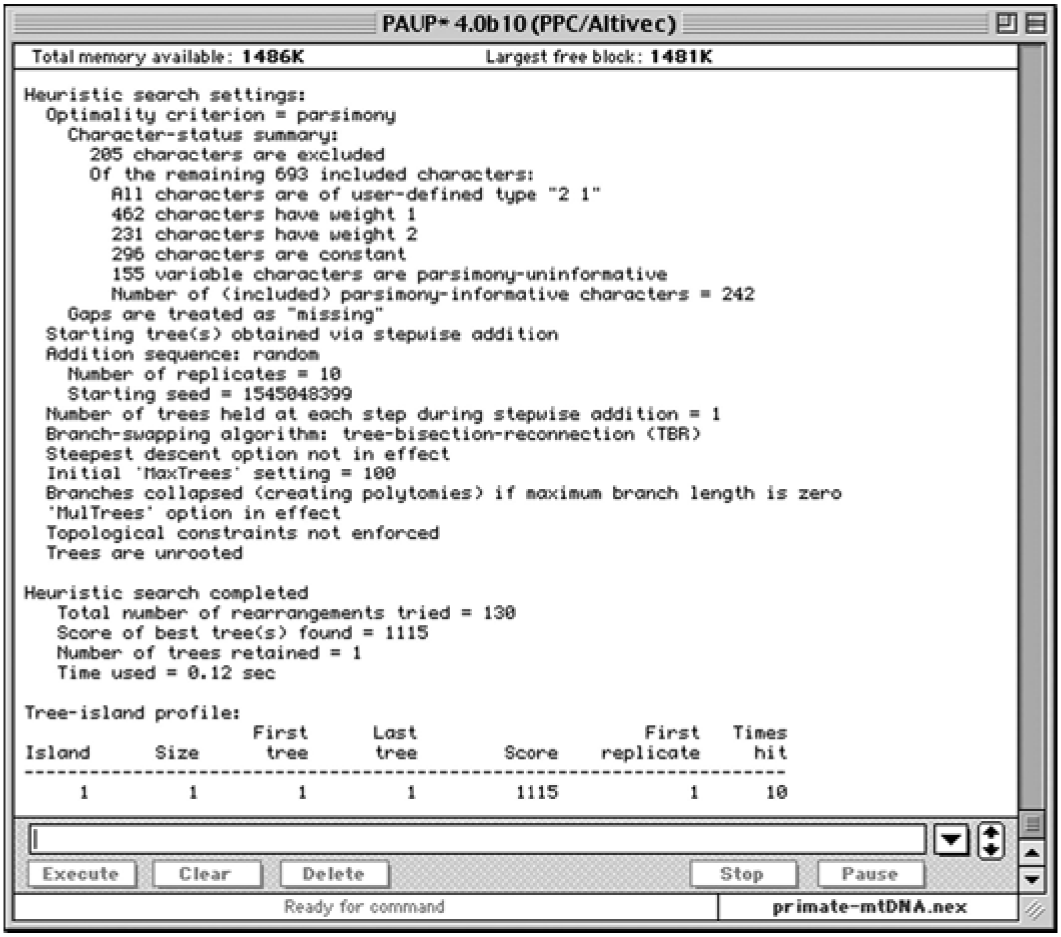
Figure 6.4.7 A summary of the available options under the heuristic search command and the current default setting of each option.

complete discussion of the branch-swapping algorithms used by PAUP, see Swofford et al. (1996).*

11. Start the heuristic search.

    ```
    hsearch start=stepwise addseq=random swap=spr;
    ```

    Once the search is started, PAUP* will again display general information about the status of the characters and the options used for the heuristic search. When the search completes, PAUP* will display general information about the results of the search (Fig. 6.4.8). If logging was started before executing the search (e.g., `log start file=mylogfile.log;`), a record of the information output to the display



```
Total memory available: 1486K                    Largest free block: 1481K

Heuristic search settings:
  Optimality criterion = parsimony
    Character-status summary:
      205 characters are excluded
      Of the remaining 693 included characters:
        All characters are of user-defined type "2 1"
        462 characters have weight 1
        231 characters have weight 2
        296 characters are constant
        155 variable characters are parsimony-uninformative
        Number of (included) parsimony-informative characters = 242
    Gaps are treated as "missing"
  Starting tree(s) obtained via stepwise addition
  Addition sequence: random
    Number of replicates = 10
    Starting seed = 1545048399
  Number of trees held at each step during stepwise addition = 1
  Branch-swapping algorithm: tree-bisection-reconnection (TBR)
  Steepest descent option not in effect
  Initial 'MaxTrees' setting = 100
  Branches collapsed (creating polytomies) if maximum branch length is zero
  'MulTrees' option in effect
  Topological constraints not enforced
  Trees are unrooted

Heuristic search completed
   Total number of rearrangements tried = 130
   Score of best tree(s) found = 1115
   Number of trees retained = 1
   Time used = 0.12 sec

Tree-island profile:
                     First      Last                        First   Times
Island      Size      tree      tree        Score    replicate     hit
-----------------------------------------------------------------------
     1         1         1         1         1115            1      10

Ready for command
```

Figure 6.4.8 The summary display of the random addition heuristic search.

window would be saved to a file. Likewise, logging can be turned off at any point in the analysis (e.g., `log stop;`).

In some cases, multiple trees of the same length may be recovered by a search. In this case, however, a single tree from a single island with a length of 1115 is found. Islands are meant to represent sets of optimal trees separated by "seas" of suboptimal trees. To get from one island to the next, one must pass through at least one suboptimal tree (see Swofford et al., 1996; Page and Holmes, 1998).

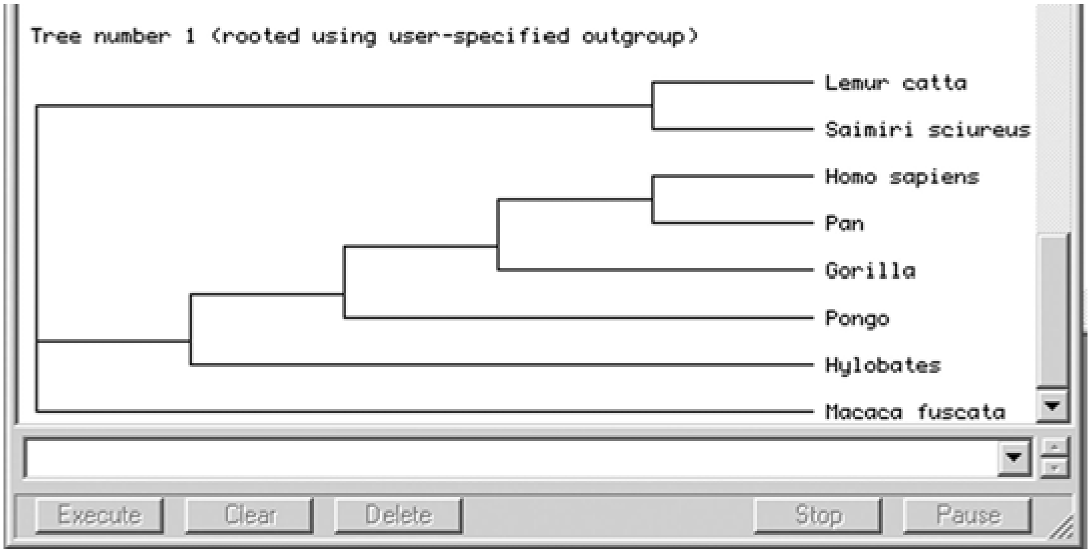
Up to this point no pictures of trees have been printed to the display. In the next set of steps, procedures to display and print the tree found by the heuristic search will be covered.

Print and save trees

12. Define the outgroup sequences.

```
outgroup lemur_catta macaca_fuscata saimiri_sciureus;
```

Ordinarily, PAUP stores the trees found by a search, such as the one just conducted, as unrooted trees. These trees are displayed as rooted trees in the output listing using the default method of outgroup rooting. If no sequences have been assigned to the outgroup, by default, PAUP* will use the first sequence in the matrix as the outgroup. In this example, all three non-hominoids will be defined to the outgroup list. In this case, the* `outgroup` *command is used followed by the three outgroup taxa labels. For a discussion of how to select an outgroup see* UNIT 6.1.



```
Tree number 1 (rooted using user-specified outgroup)
```

Figure 6.4.9 A simple diagram of the tree found by the random addition sequence heuristic search.

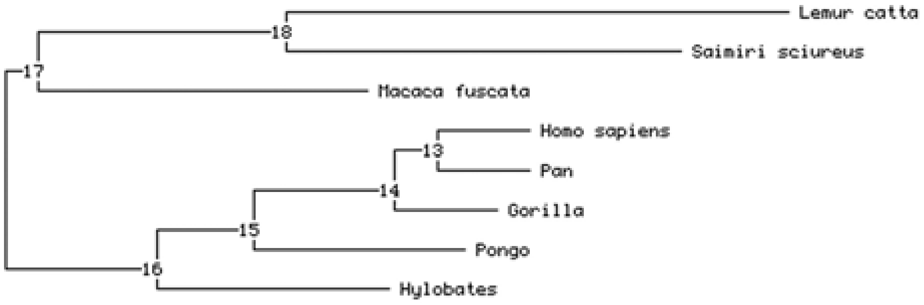


Figure 6.4.10 Phylogram of the single most parsimonious tree.

13. Display the tree found by the search.

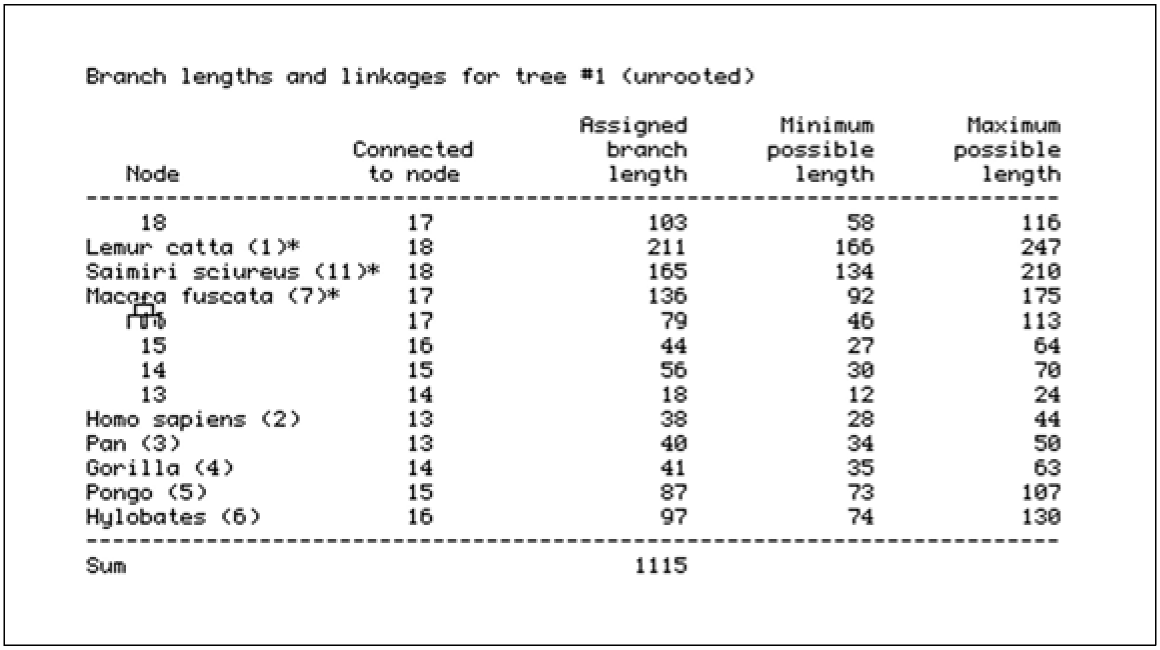
```
showtrees all;
```

The simplest way to display a picture of the tree recovered by the preceding heuristic search is to use the `showtrees` *command. This command will display a basic diagram depicting the branching order of the sequences (Fig. 6.4.9). To get more information about the tree recovered by the search (e.g., branch lengths, estimates of ancestral character states, etc.) use the* `describetrees` *command (see step 14).*

14. Display a phylogram with a list of branch lengths.

```
describetrees /plot=phylogram brlens=yes
rootmethod=outgroup outroot=monophyl;
```

The syntax given above tells PAUP to draw the maximum-parsimony tree again, but this time as a phylogram. In this case, the tree branch lengths are proportional to the amount of implied evolutionary change and the outgroup taxa are shown as a monophyletic sister group with respect to the ingroup sequences (Fig. 6.4.10). The command also asks PAUP* to output a complete list of branch lengths including the minimum and maximum possible*



```
Branch lengths and linkages for tree #1 (unrooted)

                                    Assigned       Minimum       Maximum
                    Connected         branch      possible      possible
   Node              to node          length        length        length
-------------------------------------------------------------------------
    18                  17               103            58           116
Lemur catta (1)*        18               211           166           247
Saimiri sciureus (11)*  18               165           134           210
Macaca fuscata (7)*     17               136            92           175
    16                  17                79            46           113
    15                  16                44            27            64
    14                  15                56            30            70
    13                  14                18            12            24
Homo sapiens (2)        13                38            28            44
Pan (3)                 13                40            34            50
Gorilla (4)             14                41            35            63
Pongo (5)               15                87            73           107
Hylobates (6)           16                97            74           130
-------------------------------------------------------------------------
Sum                                     1115
```

Figure 6.4.11 The table of branches and linkages output by the `describetrees` command.

lengths of each branch over all of the most parsimonious reconstructions (see Swofford and Maddison, 1987; also see Fig. 6.4.11). Many more options are available under the `describetrees` *command; rather than discuss these options here, users interested in this topic should consult the PAUP* command reference documentation.*

15. Save trees and print a high-resolution tree.

```
savetrees file=parsTree.tre brlens=yes;
```

The Macintosh version is currently the only version of PAUP that supports printing of high-resolution trees (using the Print Trees menu command). High-resolution pictures of the tree found by the PAUP* search can still be obtained for the Windows or the Portable versions of PAUP*; however, a couple of extra steps are required. In short, the tree must be saved to a NEXUS-formatted tree file (the default) and then opened using a third-party tree-printing software package such as the program TreeView by Rod Page. Also see UNIT 6.2, where the software package TreeView is discussed in detail.*

16. Exit the program.

```
quit;
```

USING PAUP* TO INFER A MAXIMUM-LIKELIHOOD TREE FROM DNA SEQUENCES

ALTERNATE PROTOCOL

In addition to the parsimony criterion described above, PAUP* can be used to infer evolutionary trees using several maximum-likelihood models of DNA substitution (see Background Information). PAUP*, however, does not provide maximum-likelihood models for amino acid substitution. *UNIT 6.1* gives a general discussion of the maximum-likelihood method and a more detailed discussion of the subject can be found in Swofford et al. (1996). In short, the maximum-likelihood method selects the hypothesis that maximizes the probability of obtaining the observed data. In the context of phylogenetic inference, the hypothesis is the tree (i.e., the branching order of the sequences as well as the branch lengths) and the observed data are the DNA sequences. The hypotheses (i.e., trees) are evaluated under a proposed model of DNA substitution. Of course, as the number of sequences in a data matrix increases, so too does the number of possible hypotheses that need to be evaluated. This problem is shared by all criterion-based methods; however, with maximum likelihood, scoring a single tree can be very time-consuming. In the past, this practical constraint discouraged some people from using maximum likelihood. Fortunately, advances made in computational processing power, coupled with algorithmic improvements, have made it much more practical to used this method. For example, it is now not uncommon to see 50 or more sequences being analyzed in maximum-likelihood searches.

This protocol describes one way to implement a maximum-likelihood tree search using PAUP* in practical terms. The process can be divided into three parts: (1) getting a tree; (2) selecting a model of DNA substitution; and (3) searching for the best (optimal) tree under the selected model. Essentially, steps 1 and 3 were covered in the Basic Protocol. As such, the focus will be mostly on describing how to select a model of DNA substitution that can be used in a subsequent search.

The authors also wish to make it clear that there are many ways to select an appropriate model of DNA substitution for a set of aligned sequences. Most of these methods attempt to select a model that optimizes a tradeoff between fit to the data and model complexity. In practice, methods used to select a model of sequence evolution range from fully automated—e.g., ModelTest (Posada and Crandall, 1998; *UNIT 6.5*)—to fully interactive. For this example, the model selection part is demonstrated using the fully interactive approach. This approach is suggested for two reasons. First, by walking step-by-step through the model-selection process it is hoped that the reader will become acquainted with some of the options available in PAUP* for implementing likelihood-based analyses. Second, by selecting a model interactively it is often possible to learn about specific aspects of sequence evolution that might otherwise go unnoticed.

Necessary Resources

Hardware

PAUP* is compatible with most modern hardware configurations including Apple Macintosh (PPC and 68k-based processors), Intel, AMD, and a number of Unix and Linux workstations (e.g., Sun, Alpha, IBM, SGI, PPC-based, i386-based, and others). Note that maximum-likelihood analyses are generally more computationally demanding than parsimony or distance analyses. As such, maximum-likelihood analyses run on older hardware equipped with slower CPUs will in some cases run prohibitively slowly. In general, however, most Pentiums, PowerPCs, and Unix workstations will be able to complete a maximum-likelihood search in a "reasonable" amount of time.

Software

PAUP* is distributed by Sinauer Associates at *http://www.sinauer.com*. Three versions of the program are available: Macintosh, Windows, and Portable (see Table 6.4.1). These versions provide identical analytical capabilities, but differ in the way these analyses are controlled. The Macintosh version supports a full graphical user interface, which allows the user to execute commands via menus and the command line, while the Windows and Portable versions are almost entirely command-line driven. Some menu functions are available in the Windows version; however, the functions are mostly restricted to file "open" and "edit" operations. Because the command-line interface is available among all three versions, it is used for the following examples. The location of the corresponding menu item is given in the PAUP* command reference documentation (available on the PAUP* 4.0 distribution disk and on the PAUP* Web site at *http://paup.csit.fsu.edu/commandref.pdf*). In addition, several sources already offer an overview of PAUP* using the Macintosh menu interface (Hall, 2001; PAUP* quick-start tutorial, *http://paup.csit.fsu.edu/quickstart.pdf*). Installation instructions are included with the software.

Files

Data files for PAUP* are standard text files, which adhere to the NEXUS file specifications. The NEXUS format was designed by Maddison et al. (1997) to facilitate the interchange of input files between programs used in phylogeny and classification. The text in a NEXUS file is arranged into blocks, which are delimited by the words `begin` and `end`. The text immediately following the word `begin` defines the block type. For example, the `TAXA` and `CHARACTERS` blocks shown in Figure 6.4.1 define a simple data set composed of four taxa (sequences) and 60 nucleotide characters (sites). PAUP* supports several predefined data types: DNA, RNA, nucleotide (DNA or RNA), protein, and standard. The set of predefined character-state symbols are `ACGT` for the data type DNA, `ACGU` for RNA, the standard one-letter amino acid codes for protein, and `01` for standard. In addition, standard ambiguity codes for the molecular data types are implemented by predefined "equate" macros. Additional character states other than those represented by the predefined data types can be specified using the `SYMBOLS` subcommand (see the PAUP* documentation). PAUP* data files must start with the character string `#NEXUS`. Comments can be included in a data file by enclosing them in square brackets, as shown in Figure 6.4.1. PAUP* 4.0 is also capable of translating other file formats to the NEXUS format. Supported formats include PHYLIP (see *UNIT 6.3*), Hennig86, GCG/Pileup (see *UNIT 3.6*), MEGA, NBRF-PIR, FreqPars, and text (space and tab-delimited). The Support Protocol below details using PAUP* to import non-NEXUS data files. The NEXUS file used for this protocol can be found in the application subdirectory labeled `Sample-NEXUS-data` and also on the *Current Protocols in Bioinformatics* Web site at *http://www3.interscience.wiley.com/c_p/cpbi_sampledatafiles.htm.*

NOTE: Due to formatting constraints, some commands given in the following examples span multiple lines. However, when entering commands at the `paup>` prompt or into the command-line interface, all of the text preceding the semicolon should be entered on the same line.

Execute the data file and get a tree quickly

1. Start PAUP* and execute the data file.

```
execute primate-mtDNA.nex;
```

See Basic Protocol, step 1, for a description of how to start the PAUP application. Users who have already worked through the procedures described in the Basic Protocol should quit PAUP* before starting this example (e.g., type* `quit`*). Restarting the program will ensure that the option settings are set to the factory default and that all of the characters and taxa included in the sample file will also be included in the subsequent example analysis. An alternative to quitting the program is to manually reset the PAUP* options to their factory default values and restore character weights, the deleted taxa, and excluded characters (sites). For example:*

```
include all;
undelete all;
wts 1:all;
factory;
```

2. Get a tree quickly.

```
nj;
```

As discussed in UNIT 6.3, the neighbor-joining method is generally a good way to get a "rough" estimate of the phylogeny very quickly. Submitting the command `nj` *tells PAUP* to construct a neighbor-joining tree using the set of pairwise "p-distances" (the default distance for nucleotide characters). A p-distance, also known as a dissimilarity distance, is simply the total number of observed differences between a pair of sequences divided by the total number of sites compared. The p-distance is one of many distance measures implemented in PAUP*. Use the* `dset` *command to switch to another distance (e.g., perhaps one that accounts for superimposed substitutions). For example, the following command changes the pairwise distances calculated by PAUP* to the log-determinant distance (Lockhart et al., 1994; Steel, 1994).*

```
dset distance = logdet;
```

In this example, the tree topologies obtained by using the uncorrected and corrected distances are the same. Another way to get a tree quickly is to read it from a NEXUS-formatted tree file. For example, the following syntax will replace all trees currently in PAUP memory with the tree or trees contained in the file.*

```
gettrees file=myfavorite.tre mode=3;
```

To add a tree or trees from a file to the tree or trees currently in memory, change the `mode` *option to* `7` *(see the PAUP* documentation for a complete description of the* `gettrees` *options).*

Select an appropriate model of DNA substitution

3. Select a starting model of DNA substitution.

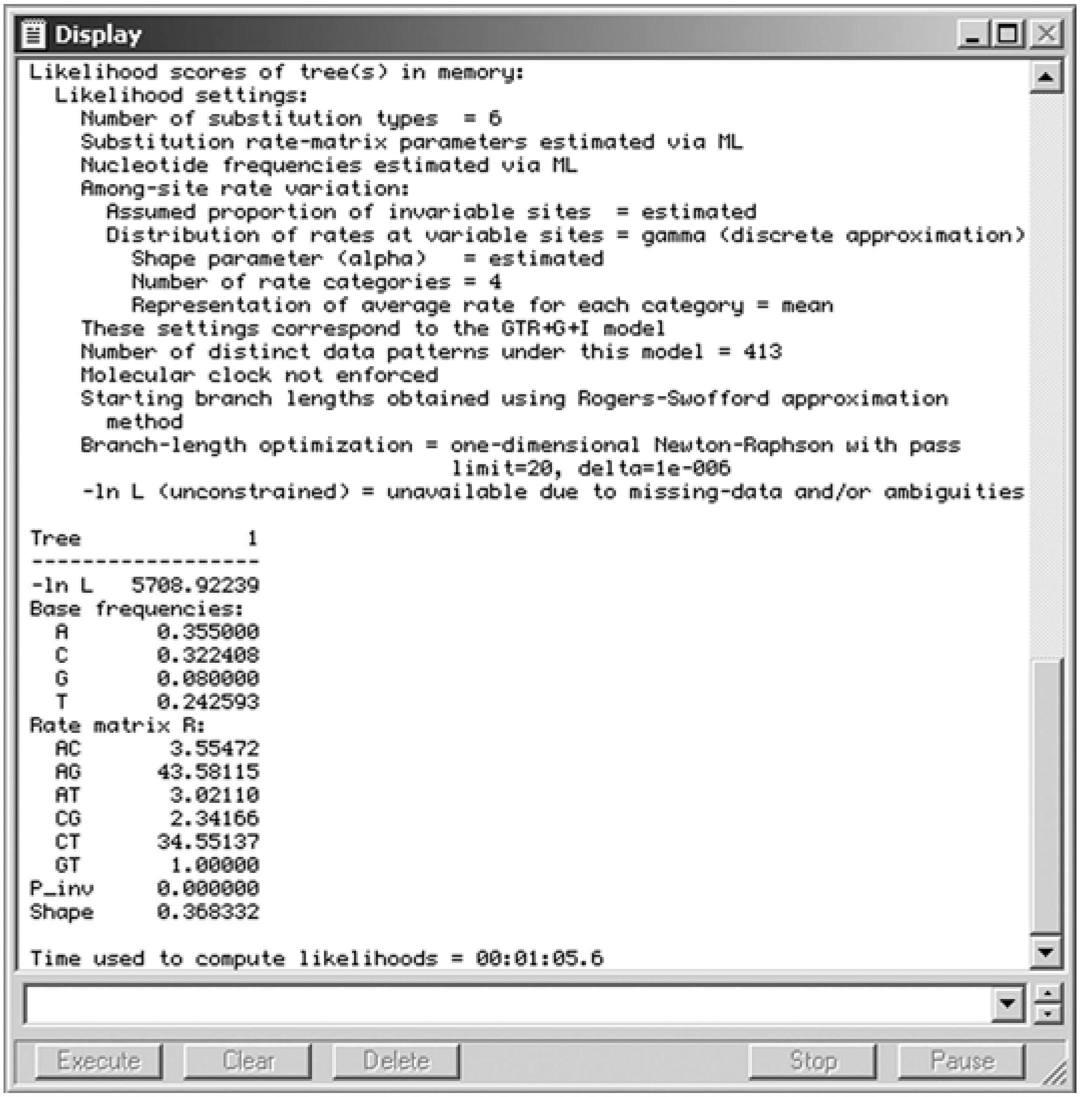
```
lscores 1/nst=6 rmatrix=estimate basefreq=estimate
rates=gamma shape=estimate pinvar=estimate;
```

Start by evaluating the likelihood of the neighbor-joining tree currently in memory using the most parameter-rich DNA substitution model available in PAUP—i.e., the General Time Reversible model (see e.g., Yang, 1994a) with six substitution types, base composition estimated from the data, and among-site rate heterogeneity estimated using the invariable sites plus gamma model (GTR + I + Γ model; Gu et al., 1995; Waddell and Penny, 1996). The* `nst=6` *and* `rmatrix=estimate` *options tell PAUP* to estimate the rate of nucleotide substitution separately for six substitution types (A ↔ C, A ↔ G, A ↔ T, C ↔ G, C ↔ T, G ↔ T). By default PAUP* does not assume that the nucleotide composition is equal; rather, base frequencies are set to the value observed in the alignment. For this example, however, the authors have chosen to estimate base composition using maximum likelihood (e.g.,* `basefreq=estimate`*). Finally, the remaining three options (i.e.,* `rates=gamma shape=estimate pinvar=estimate`*) relax the assumption that the rate of substitution among sites is equal.*

After the `lscores` *command is executed, it may take several minutes, depending on the speed of the computer, for PAUP* to complete the parameter optimizations. When the calculation is complete, look at the output in the display window (Fig. 6.4.12). One obvious parameter that can be eliminated from the estimation procedure is* `pinvar` *(i.e., proportion of invariable sites). The value of* `pinvar` *is equal to zero, suggesting that the gamma-distributed-rates model alone may accommodate the among-site rate heterogeneity. Furthermore, eliminating the* `pinvar` *parameter from the initial model will reduce the number of free parameters without changing the likelihood of the model. In the next iteration of the* `lscores` *command, an attempt will be made to simplify the DNA substitution model even further by assuming that all sites are changing at the same rate.*

4. Reduce the number of parameters that PAUP* needs to estimate (i.e., restrict the model).

```
lscores 1/nst=6 rmatrix=estimate basefreq=estimate
rates=equal pinvar=0;
```

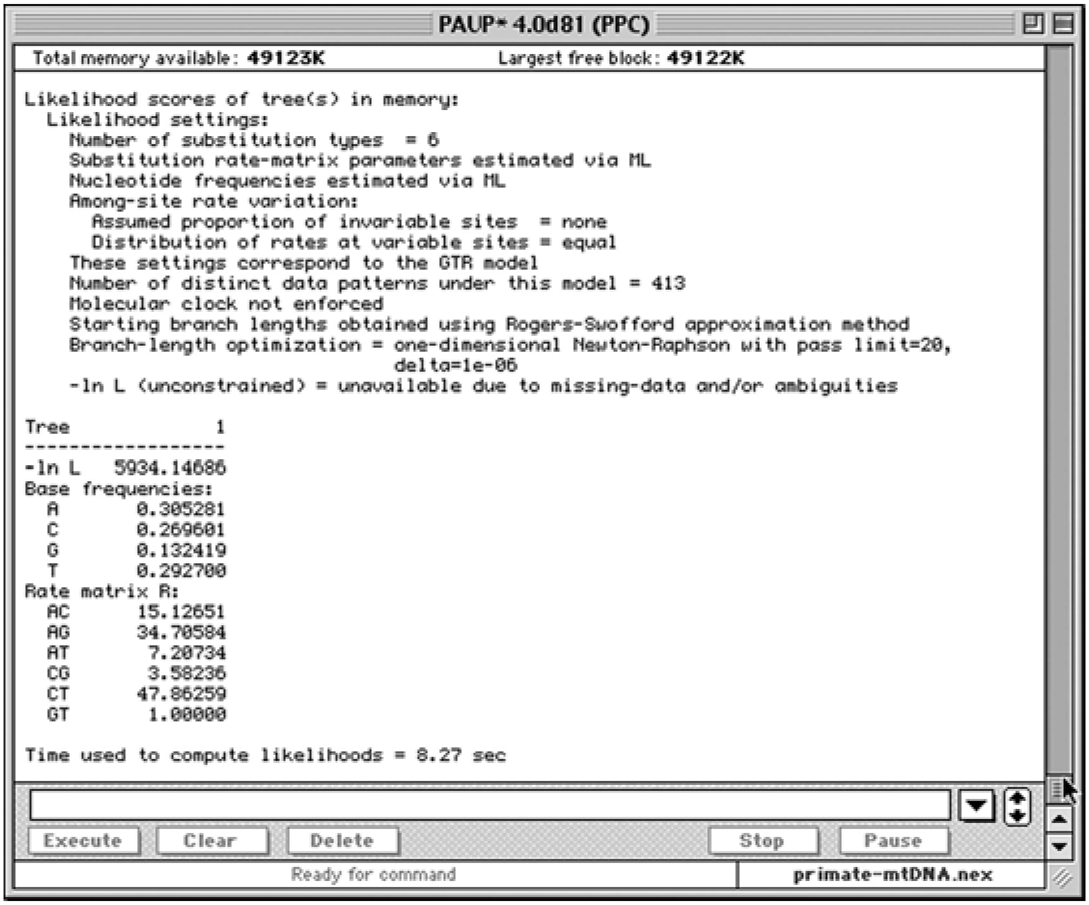


```
Likelihood scores of tree(s) in memory:
  Likelihood settings:
    Number of substitution types  = 6
    Substitution rate-matrix parameters estimated via ML
    Nucleotide frequencies estimated via ML
    Among-site rate variation:
      Assumed proportion of invariable sites  = estimated
      Distribution of rates at variable sites = gamma (discrete approximation)
        Shape parameter (alpha)   = estimated
        Number of rate categories = 4
        Representation of average rate for each category = mean
    These settings correspond to the GTR+G+I model
    Number of distinct data patterns under this model = 413
    Molecular clock not enforced
    Starting branch lengths obtained using Rogers-Swofford approximation
      method
    Branch-length optimization = one-dimensional Newton-Raphson with pass
                                 limit=20, delta=1e-006
    -ln L (unconstrained) = unavailable due to missing-data and/or ambiguities

Tree             1
------------------
-ln L   5708.92239
Base frequencies:
  A       0.355000
  C       0.322408
  G       0.080000
  T       0.242593
Rate matrix R:
  AC       3.55472
  AG      43.58115
  AT       3.02110
  CG       2.34166
  CT      34.55137
  GT       1.00000
P_inv     0.000000
Shape     0.368332

Time used to compute likelihoods = 00:01:05.6
```

Figure 6.4.12 Model parameters estimated on the neighbor-joining tree under the General Time Reversible model (see Yang, 1994a) with among site rate heterogeneity estimated using the invariable sites plus gamma model (Gu et al., 1995; Waddell and Penny, 1996).

The main display window (Fig. 6.4.13) shows the new parameter estimates and the log-likelihood score for the restricted substitution model. The restricted model score is roughly 225 log-likelihood units worse then the more general model—i.e., the model that allows rates to vary among nucleotide positions (see step 3). A likelihood-ratio test (Cox and Hinkley, 1974) could be used to determine whether the restricted model significantly decreases the model fit. However, in this case the magnitude of the difference is so large that among-site rate heterogeneity obviously improves the overall model fit. This result should not be terribly surprising considering that the data set is composed of nucleotide sites from coding and noncoding regions, which are clearly under different structural and functional constraints.

Returning to the GTR + I + Γ model output (Fig. 6.4.12) it appears that it might be possible to capture the major properties of the substitution process without having to estimate six separate rate categories. That is, in the next iteration of the `lscores` *command substitution rates will only be estimated for transitions (A ↔ G and C ↔ T) and transversion (A ↔ C, A ↔ T, C ↔ G, G ↔ T) and the parameters needed to estimate among site rate heterogeneity using the gamma-distributed rates model will be restored.*



```
Likelihood scores of tree(s) in memory:
  Likelihood settings:
    Number of substitution types  = 6
    Substitution rate-matrix parameters estimated via ML
    Nucleotide frequencies estimated via ML
    Among-site rate variation:
      Assumed proportion of invariable sites  = none
      Distribution of rates at variable sites = equal
    These settings correspond to the GTR model
    Number of distinct data patterns under this model = 413
    Molecular clock not enforced
    Starting branch lengths obtained using Rogers-Swofford approximation method
    Branch-length optimization = one-dimensional Newton-Raphson with pass limit=20,
                                 delta=1e-06
    -ln L (unconstrained) = unavailable due to missing-data and/or ambiguities

Tree            1
------------------
-ln L   5934.14686
Base frequencies:
  A       0.305281
  C       0.269601
  G       0.132419
  T       0.292700
Rate matrix R:
  AC      15.12651
  AG      34.70584
  AT       7.20734
  CG       3.58236
  CT      47.86259
  GT       1.00000

Time used to compute likelihoods = 8.27 sec
```

Figure 6.4.13 Model parameters estimated on the neighbor-joining tree under the General Time Reversible model minus among site rate heterogeneity.

5. Further refine the model.

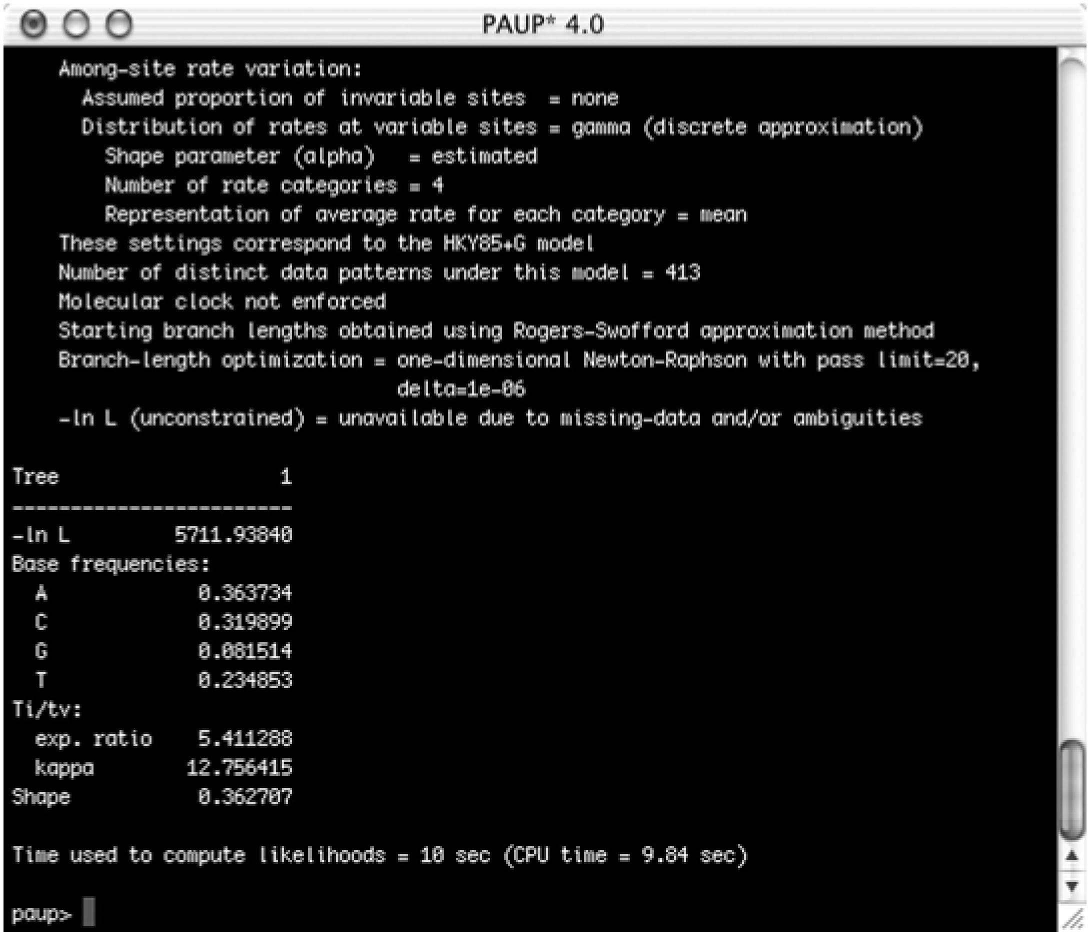
```
lscores 1/nst=2 tratio=estimate basefreq=estimate
rates=gamma shape=estimate;
```

The model specified in this iteration of the `lscores` *command is generally referred to as the HKY + Γ model (Hasegawa et al., 1985; Yang, 1994b). In this case, the restricted model (HKY + Γ) only differs from the more general model (GTR + Γ) by 3 log likelihood units (Fig. 6.4.14). The removal of four substitution classes seems to have a relatively small impact on the overall model fit, especially when compared to the difference achieved when the single among-site rate heterogeneity parameter was removed. Furthermore, according to a likelihood ratio test, the more parameter-rich GTR model does not significantly improve the model fit (P = 0.19776). The test statistic for the likelihood-ratio test is equal to two times the difference between the log-likelihood scores [e.g., 2(5711.94 – 5708.92)] and it is chi-square distributed with degrees of freedom equal to the difference between the number of free parameters used by each substitution model [e.g., 10(GTR + Γ) – 6(HKY + Γ)]. Therefore, with a critical value of 0.05 we reject the more complicated GTR + Γ model and will use the model and the model parameters estimated in this step in the forthcoming heuristic search.*

Search for an optimal tree using the selected model

6. Set the optimality criterion to Likelihood.

```
set criterion=likelihood;
```



```
    Among-site rate variation:
      Assumed proportion of invariable sites  = none
      Distribution of rates at variable sites = gamma (discrete approximation)
        Shape parameter (alpha)   = estimated
        Number of rate categories = 4
        Representation of average rate for each category = mean
    These settings correspond to the HKY85+G model
    Number of distinct data patterns under this model = 413
    Molecular clock not enforced
    Starting branch lengths obtained using Rogers-Swofford approximation method
    Branch-length optimization = one-dimensional Newton-Raphson with pass limit=20,
                                 delta=1e-06
    -ln L (unconstrained) = unavailable due to missing-data and/or ambiguities

Tree                   1
------------------------
-ln L         5711.93840
Base frequencies:
  A             0.363734
  C             0.319899
  G             0.081514
  T             0.234853
Ti/tv:
  exp. ratio    5.411288
  kappa        12.756415
Shape           0.362707

Time used to compute likelihoods = 10 sec (CPU time = 9.84 sec)

paup>
```

Figure 6.4.14 Model parameters estimated on the neighbor-joining tree under the HKY + Γ model plus among the gamma distribute rates.

By default, the optimality criterion is set to maximum parsimony. The `set` *command is used to change the criterion so that all subsequence searches will use the selected maximum-likelihood model.*

7. Set the maximum-likelihood model parameters.

```
lset nst=2 rmatrix=previous basefreq=previous
rates=gamma shape=previous;
```

This command is a shortcut that takes the parameters estimated by the `lscores` *command in step 5 and fixes them for subsequent likelihood operations. Alternatively, the likelihood parameters can be set manually by copying the parameter values into an* `lset` *command. For example:*

```
lset nst=2 tratio=5.411288 basefreq=(0.36734 0.319899
0.081514) shape=0.362707;
```

The parameter values listed above were taken from the likelihood scores output for the HKY85 + Γ model (Fig. 6.4.14).

8. Run a heuristic search using the maximum-likelihood model.

```
hsearch;
```

The same set of heuristic search options (see Fig. 6.4.7) is available for each of the three optimality criteria implemented by PAUP (i.e., parsimony, maximum likelihood, and least-squares distances). The reader should refer to steps 9 through 10 in the Basic Protocol for more information regarding this step. After the search finishes, the maximum-likelihood tree can be displayed by following steps 12 through 15 of the Basic Protocol.*

9. If the topology found by the preceding heuristic search differs from the starting tree, it is possible to further refine the maximum-likelihood parameter estimates (and associated optimal tree) by repeating steps 3 to 7 above until the topology does not change (see Swofford et al., 1996). In this example, however, the topologies did not differ so the protocol will stop here.

The optimal tree is a function of both data and parameters, so if one changes parameters one may also change the tree. Given this relationship, ideally, one would estimate model parameters for all of the trees found during a heuristic search.For most data sets, however, such a search would be prohibitively time-consuming. Instead, the successive-approximations approach (Swofford et al., 1996) described above is an effective way to implement a model-based tree search and also arrive at a stable set of model parameters.

SUPPORT PROTOCOL

USING PAUP* TO IMPORT NON-NEXUS DATA FILES

Although a number of computer packages save multiple sequence alignments to the native PAUP* file format, situations may arise where it is necessary to analyze a data set that is not in the NEXUS format. To save the user from the trouble of having to reformat the non-NEXUS file manually, PAUP* provides a command called `tonexus` that can be used to translate several popular data file formats. The file formats that can be translated to the NEXUS format are: GCG/MSF, NBRF-PIR, PHYLIP, MEGA, FreqPars, and Text (tab or space delimited). Examples of all these data formats can be found on the PAUP* Web site at *http://www.paup.csit.fsu.edu/nfiles.html*. Getting a non-NEXUS file to the stage were it is ready to be analyzed requires two general steps: (1) converting and saving the non-NEXUS file to a NEXUS file and (2) executing the newly converted NEXUS file. An example of how to do this is given below.

Necessary Resources

Hardware

PAUP* is compatible with most modern hardware configurations including Apple Macintosh (PPC and 68k-based processors), Intel, AMD, and a number of Unix and Linux workstations (e.g., Sun, Alpha, IBM, SGI, PPC-based, i386-based, and others).

Software

PAUP* is distributed by Sinauer Associates at *http://www.sinauer.com*. Three versions of the program are available: Macintosh, Windows, and Portable (see Table 6.4.1). These versions provide identical analytical capabilities, but differ in the way these analyses are controlled. The Macintosh version supports a full graphical user interface, which allows the user to execute commands via menus and the command line, while the Windows and Portable versions are almost entirely command-line driven. Some menu functions are available in the Windows version; however, the functions are mostly restricted to file “open” and “edit” operations. Because the command-line interface is available among all three versions, it is used for the following examples. The location of the corresponding menu item is given in the PAUP* command reference documentation (available on the PAUP* 4.0 distribution disk and on the PAUP* Web site at *http://paup.csit.fsu.edu/commandref.pdf*). In addition, several sources already offer an overview of PAUP* using the Macintosh menu interface (Hall, 2001; PAUP* quick-start tutorial, *http://paup.csit.fsu.edu/downl.html*). Installation instructions are included with the software.

Files

A sample PHYLIP file will be used in the following example. The file can be found in the PAUP* application subdirectory labeled `Sample import data` and also on the *Current Protocols in Bioinformatics* Web site at *http://www3.interscience.wiley.com/c_p/cpbi_sampledatafiles.htm*.

1. Start PAUP*.

   Refer to the PAUP application documentation for a description of how to launch the program. When the Macintosh or Windows version of PAUP* is first started, the program will automatically launch an Open File dialog box. Files opened via this dialog box must already be in the NEXUS format. Close the dialog box by clicking the Cancel button.*

2. Convert a PHYLIP-formatted file to NEXUS.

```
[Windows] cd '..\Sample import data';
[Portable] cd ../Sample_import_data;
tonexus format=phylip interleaved=yes
datatype=nucleotide fromfile=phylip_3x.dat
tofile=test.nex;
```

   If the non-NEXUS file is not in the active PAUP directory, then one must first use the* `cd` *command to move to the correct directory or supply the complete filename with the* `fromfile` *and* `tofile` *options. Because some directory names in this example include spaces, the full path must be enclosed by single quotes. Users of the Portable version of PAUP* can avoid some typing by using the tilde (~) symbol in place of the complete path to their home directory (e.g.,* `tofile=~/mydatafiles/paupfiles.nex`*). The options given under the* `tonexus` *command are mostly self-explanatory. The* `interleaved` *option tells PAUP* to expect the data in the PHYLIP file to be arranged into blocks. By default, PAUP* assumes that the data are nucleotide characters, so the* `datatype` *option need not be explicitly stated.*

3. Execute the converted file.

```
execute test.nex;
```

   It is now possible to use the tree-building and character-diagnosing features described in the Basic and Alternate Protocols.

GUIDELINES FOR UNDERSTANDING RESULTS

The heuristic search procedure discussed above attempts to find optimal trees according to the currently selected criterion. Additionally, it will usually be of interest to determine the strength of support for the implied groupings. A common way to assess the support for the nodes of a tree topology under a specified reconstruction method is to used a nonparametric bootstrap (Felsenstein, 1985) or jackknife analysis (Penny and Hendy, 1985; Farris et al., 1996). These methods are discussed in *UNIT 6.1* so it will only be mentioned that both of these analyses are implemented in PAUP* and can be executed using the `bootstrap` or `jackknife` commands, respectively. As with the other PAUP* commands discussed in this unit, a brief description of the options available under each of these commands can be found in the current PAUP* command reference documentation available on the PAUP* Web site.

In some cases, one may be interested in knowing how the complete optimal tree compares with other candidate trees. The relative differences among the scores of candidate trees under the various optimality criteria provide a basis for this type of comparison. However, it is natural to want to know whether the magnitude of the difference between tree scores is great enough that a specific tree can be excluded with a certain degree of confidence. Methods designed to do this—i.e., Templeton (1983), Kishino and Hasegawa (1989), and Shimodaira and Hasegawa (1999; see also Goldman et al., 2000) are also available in PAUP* and can be implemented using the options under the `pscores` and `lscores` commands. Users who are interested in these methods should consult the PAUP* documentation. A number of other tree diagnostics are available in PAUP*; however, there is not enough space to discuss them all in this unit. Again, for those interested in this topic, the authors recommend looking at the available options under commands such as `contree`, `describetrees`, `reconstruct`, and `treedist`.

Finally, the methods just mentioned attempt to assess the reliability of trees given a specific method of analysis. As discussed below (see Commentary), however, the performance of a method is not guaranteed under all circumstances. For example, the relevance of a confidence test is severely compromised if the phylogenetic reconstruction method chosen is biased in some fundamental way. For example, the problem of long-branch attraction (see Commentary) can lead to high confidence (e.g., as measured by bootstrap support) in incorrect phylogenetic groupings. One simple way to explore the possibility that results are being influenced by a particular methodological bias is to apply several different methods of analysis, each subject to its own potential biases and limitations, to the data matrix. The observation that all of the methods give similar results provides some encouragement that the results were not adversely affected by problems specific to particular methods. If, on the other hand, different methods give different results, it is important to investigate what properties of the data set might be causing some or all of the methods to be misled.

COMMENTARY

Background Information

Model-free and model-based approaches to phylogeny

In a perfect world, reconstruction of phylogenies would be simple. For example, sequences would have a level of variation appropriate to the problem being addressed and the potentially misleading effects of convergence, parallelism, and reversal could safely be ignored. Unfortunately, this ideal situation is rarely realized. To cope with these realities, developers of phylogenetic methodology have taken two basic and often conflicting strategies. The first approach attempts to reconstruct the phylogeny without reference to an explicit model of evolutionary change. The second class of methods adopts a statistical approach to the problem of phylogenetic inference, relying on the use of stochastic models of the evolutionary process to assess the degree of fit between candidate trees and the observed data. The parsimony and maximum-likelihood methods discussed above are the most well known examples of "model-free" and "model-based" methods, respectively. It is important, however, to realize that neither model-free nor model-based approaches are free from specific shortcomings and limitations. Furthermore, the distinction between them is not always clear; e.g., it is possible to define parsimony as a maximum-likelihood method under an extremely general (and probably too general) substitution model (Tuffley and Steel, 1997). Other methods do not sort cleanly into one of these two categories. Distance methods can range from "model-based" to "model-free" depending on whether stochastic models are used to transform raw sequence differences into expected amounts of change according to the chosen model. In the following section, some advantages and disadvantages to of the two approaches will be highlighted and users will be provided with relevant and accessible references for further study.

Parsimony

The parsimony method selects the tree that minimizes the number of "steps" (character changes or substitutions) required to explain the observed data. The parsimony criterion is often defended on the basis of simplicity: without evidence to the contrary, sequence identities are assumed to be due to common ancestry.

When conflicts are unavoidable, the explanation that minimizes the number of ad hoc assumptions of homoplasy (similarity for reasons other than common ancestry, including convergence, parallelism, and reversal) is preferred. A number of parsimony variants can be implemented in PAUP* according to the way in which character states are allowed to change. No matter what constraints are imposed on character changes, however, the goal of all parsimony-based methods remains the same—i.e., minimization of the amount of implied evolutionary change. Unordered-character or Fitch parsimony (Fitch, 1971) assigns an equal cost to transformations from any character state to any other character state. In ordered-character or Wagner parsimony (Kluge and Farris, 1969; Farris, 1970), character states are represented by a linearly ordered transformation series and changes are scored according to the number of intermediate states linking any pair of states. A number of other parsimony variants are reviewed in Swofford et al. (1996). Of these, a very general method described originally by Sankoff (1975) is the one most often applied to molecular sequence data. This "generalized parsimony" method requires the user to specify a matrix giving the cost of a change from any character state to any other. Cost matrices ("step matrices") are typically used to incorporate additional assumptions about the evolutionary process into the analysis. For example, transition bias can be accommodated by assigning higher costs to transversions than to transitions, as demonstrated in the Basic Protocol. One special case of generalized parsimony is "transversion" parsimony, in which transition substitutions are ignored by assigning them a cost of zero (or equivalently, by recoding the data matrix as two-state characters representing purine versus pyrimidine).

The combination of intuitive appeal, ease of implementation, and speed of calculation causes parsimony to remain one of the most highly used methods in contemporary phylogenetics. There are, however, several well defined problems associated with parsimony methods. The most commonly cited criticism of parsimony methods involves their potential for statistical inconsistency in many situations (Felsenstein, 1978; Hendy and Penny, 1989; Kim, 1996; see also Swofford et al., 1996). Particular inequalities in the rates of substitution along paths of an evolutionary tree can cause the probability that character states are identical due to convergent, parallel, or back substitutions to exceed the probability that identical states are shared purely due to common ancestry. As a result, the estimate of the tree topology made by parsimony can converge to an incorrect result with increasing certainty as more data are accumulated (Felsenstein, 1978). This phenomenon, typically referred to as "long-branch attraction," can lead to either artifactual groupings or inappropriately high confidence in correct groupings (see Swofford et al., 2001, for a recent discussion). A few real-data examples of apparent long-branch attraction have been identified in parsimony analyses, as well as in likelihood analyses in which overly simplistic models were used (e.g., Sullivan and Swofford, 1997). However, considerable debate continues as to whether long-branch attraction is a real phenomenon or merely an irrelevant and purely hypothetical concern, largely owing to the fact that the true tree is never known with certainty.

For the analysis of sequence data, the sensitivity of parsimony to long-branch-attraction artifacts can be reduced by giving less weight to sites that are more prone to accept changes (e.g., third codon positions in protein-coding sequences) or to substitutions that appear to occur more frequently (e.g., transitions; Hillis et al., 1994a). Unfortunately, an infinite number of possible weighting schemes exist, and it is difficult to choose among them because of the lack of a "common currency" for evaluating them: i.e., there is no basis for comparing tree lengths obtained under one weighting scheme with those from a different weighting scheme. A possible solution is to perform a "sensitivity analysis" in which a variety of weighting strategies are tested. If all weighting schemes lead to selection of the same tree(s), then this issue is less problematic. However, when different weighting schemes produce different tree topologies, some justification must be provided for the selection of a particular parsimony variant or conclusions should be limited to those that are more stable across weighting schemes.

Maximum likelihood

Maximum-likelihood phylogenetic inference seeks the tree that maximizes the probability of obtaining the observed data given some model of evolutionary change. While maximum-likelihood methods have long been a part of statistical interference (see Efron, 1998), likelihood-based analyses have only relatively recently gained a foothold in phylogenetic inference—e.g., for gene frequencies see Cavalli-Sforza and Edwards (1967); for nucleotide sequences see Felsenstein (1981);

and for amino acid sequences see Kishino et al. (1990). Despite their relatively recent introduction, however, these methods have caught on rapidly and now rival parsimony methods in popularity, especially for the analysis of nucleotide sequences. The increasing acceptance of likelihood stems from several important advantages of model-based methods. For example, the maximum-likelihood model used to infer a phylogeny can be designed to explicitly estimate several key elements of the nucleotide substitution process from the observed sequence data. Also, unlike the situation involving alternative parsimony variants, maximum likelihood provides an objective criterion to measure the goodness-of-fit between specific models and the observed data. In this way, models can be designed that are complex enough to capture the most important aspects of the substitution process, but simpler models are chosen over more complex ones when the additional parameters do not significantly improve the overall fit. Another important feature that distinguishes likelihood from parsimony methods is that likelihood methods incorporate branch-length information into the phylogenetic inference procedure. That is, the probability that substitutions will occur along long branches is greater than that for shorter branches. It is this property that makes likelihood methods less sensitive to long-branch attraction—i.e., the kinds of misleading character patterns that cause parsimony to be misled are expected to occur at high frequency on the true tree under the model; thus, likelihood is not confused by them. Additionally, all characters in the data matrix contribute information to the analyses. Constant (invariant) sites and sites that require the same number of steps on any possible tree are uninformative in parsimony analysis, whereas they provide information on rates of change that is critical for likelihood inference.

One of the greatest limitations of the likelihood method is that it requires significantly more computational time for tree searches than parsimony or distance methods (Sanderson and Kim, 2000). For example, when 50 or more sequences are included in an analysis, the computational demands of the likelihood method may become so great that less thorough methods for exploration of tree space become necessary. While there are some algorithmic shortcuts that can be employed to extend the use of likelihood-based methods for large-scale phylogeny inference, the computational complexity of this method still represents an important obstacle to its widespread use and acceptance with large data sets. The authors would simply urge users to keep the analysis time in perspective; if months or even years are taken to assemble a set of sequences for phylogenetic analysis, then spending a few days or weeks for the computer runs should not be considered unreasonable.

Evaluating results

Important generalizations regarding the performance of phylogenetic methods have been provided by analytical results (Felsenstein, 1978; Chang, 1996; Rogers, 1997; Tuffley and Steel, 1997), simulation studies (Huelsenbeck and Hillis, 1993; Gaut and Lewis, 1995; Huelsenbeck, 1995; Bruno and Halpern, 1999; Swofford et al., 2001), and experimental phylogenetics (Hillis et al., 1992; Hillis et al., 1994b; Cunningham et al., 1998). While these studies have certainly helped practitioners choose among the numerous available phylogenetic reconstruction methods, they do not lead to unambiguous recommendations. A common tendency is to overgeneralize the results of these studies. A method should not be rejected simply because it performs poorly for a single set of simulation parameters, nor should a method be uncritically accepted because it performs well under a restricted range of conditions (Hillis et al., 1996). Careful attention should be paid to the appropriateness of the assumptions of a method for the data being analyzed. This suggestion does not, however, imply that the assumptions of a method must be fully satisfied in order for that method to be useful, as it is often far better to use model-based methods that use imperfect models than to abandon the use of models entirely (Sullivan and Swofford, 2001).

Critical Parameters and Troubleshooting

As emphasized above, it is unrealistic to think that a user could select an appropriate method for reconstructing a phylogeny without first examining the data set. In the same vein, it is impossible to specify a set of analysis defaults in PAUP* that will be optimal for all data sets. In general, PAUP* default settings are chosen because they are compatible with all of the data types that can be read by the program and not because they represent the "best" general analysis settings. Accordingly, it is important to consider the state of key default parame-

```
#NEXUS
[Practice batch file]
[Numbers correspond to steps in Basic Protocol.]
Begin paup;
    set autoclose=yes warntree=no warnreset=no;
    log start file=practice.log replace;
 [2] execute primate-mtDNA.nex;
 [3] include coding/only;
 [4] delete M._mulatta M._fascicularis M._sylvanus Tarsius_syrichta;
 [5] set criterion=parsimony;
 [6] weights 2:2ndpos;
 [7] ctype 2_1:all;
 [8] cstatus;
 [11]hsearch start=stepwise addseq=random swap=spr;
 [12]outgroup lemur_catta macaca_fuscata saimiri_sciureus;
 [13]showtrees all;
 [14]describetrees 1/plot=phylogram brlens=yes rootmethod=outgroup
outroot=monophyl;
 [15]savetrees file=parsTree.tre brlens=yes replace;
end;

[Numbers correspond to steps in Alternative Protocol.]
Begin paup;
 [1] include all;
 [1] undelete all /cleartrees=yes;
 [1] wts 1:all;
 [1] factory;
 [2] nj;
 [3] lscores 1/nst=6 rmatrix=estimate basefreq=estimate rates=gamma
      shape=estimate pinvar=estimate;
 [4] lscores 1/nst=6 rmatrix=estimate basefreq=estimate rates=equal
      pinvar=0;
 [5] lscores 1/nst=2 tratio=estimate basefreq=estimate rates=gamma
      shape=estimate;
 [6] set criterion=likelihood;
 [7] lscores 1/nst=2 tratio=previous basefreq=previous rates=gamma
      shape=estimate;
 [8] hsearch;
     log stop;
end;
```

Figure 6.4.15 Commands needed to run Basic Protocol and Alternate Protocol in batch mode. Note that each command and its associated options must end with a semicolon. In this way, two commands can occupy a single line, provided that a semicolon separates them. Likewise, one command can span multiple lines provided the last line ends in a semicolon.

ters before starting an analysis. A complete enumeration of all the options available in PAUP* would be impractical in the context of this publication (such a list does exist in the current command reference document; *http://paup.csit.fsu.edu/commandref.pdf*). Instead, a few of the commands and options that are most generally used will be mentioned and discussion will be included on how to see their current default settings.

By default, the optimality criterion is set to `parsimony`. Changing the criterion requires the use of the `set` command. For example:

```
set criterion=likelihood;.
```

The `set` command controls many of the general options whose scope extends beyond a single command. Typing `set ?` will produce a list of the all the `set` command options and their current state. The specific settings for each of the three optimality criteria, `parsimony`, `likelihood`, and `distance`, are controlled by the `pset`, `lset`, and `dset` commands, respectively. Finally, search options can be found under the `alltrees`, `bandb`, or `hsearch` commands, depending on whether an exact or heuristic search is requested. Likewise, typing any command followed by a question mark will cause PAUP* to display a complete list of the available options and their current status.

Suggestions for Further Analysis

All of the analyses described up to this point have been run in the interactive mode. This mode is especially useful in the exploratory stage of an analysis, when an appropriate model and corresponding set of parameters have not yet been selected. After deciding on the general requirements of an analysis, however, it is usually most efficient to include the necessary commands and option settings in "paup blocks" and run the analysis in batch mode. In this way, the analysis can be run without requiring a user to be present at the various steps of the analysis to enter commands. In addition, repeating an analysis is just a matter of re-executing the file containing the paup blocks. Some users add another level of automation to noninteractive analyses by controlling PAUP* using Shell, Perl, or Python scripts. The added layer of scripting allows users to automatically generate NEXUS analysis files, process PAUP* results, and even create and execute additional paup command files based on results from a previous analysis.

In the example shown in Figure 6.4.15, all the commands required to complete the example analyses described in the Basic and Alternate Protocols are contained in two paup blocks. A `set` command was added at the beginning of the paup block to suppress the dialog box indicating that the heuristic search has completed and several other warnings (Macintosh and Windows only). In addition, the log command was included so that the results printed to the main display are also saved to a file. To run this analysis in batch mode, copy the text given in Figure 6.4.15 to a file and save the file in the same directory as the `primate-mtDNA.nex` file. Next, start PAUP* and execute the newly saved file (see Basic Protocol, step 1).

Literature Cited

Bruno, W.J. and Halpern, A.L. 1999. Topological bias and inconsistency of maximum likelihood using wrong models. *Mol. Biol. Evol.* 16:564-566.

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

Cavalli-Sforza, L.L. and Edwards, A.W.F. 1967. Phylogenetic analysis: Models and estimation procedures. *Am. J. Hum. Genet.* 19:233-257.

Chang, J.T. 1996. Full reconstruction of Markov models on evolutionary trees: Identifiability and consistency. *Math. Biosci.* 137:51-73l.

Cox, D.R. and Hinkley, D.V. 1974. Theoretical statistics. Chapman and Hall, London.

Cunningham, C.W., Zhu, H., and Hillis, D.M. 1998. Best-fit maximum-likelihood models for phylogenetic inference: Empirical tests with known phylogenies. *Evolution* 52:978-987.

Efron, B. 1998. R.A. Fisher in the 21st century. *Stat. Sci.* 13:95-122.

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

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

Farris, J.S., Albert, V.A., Kallersjö, M., Lipscomb, D., and Kluge, A.G. 1996. Parsimony jackknifing outperforms neighbor-joining. *Cladistics* 12: 99-124.

Felsenstein, J. 1978. Cases in which parsimony and compatibility methods will be positively misleading. *Syst. Zool.* 27:401-410.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol.* 17:368-376.

Felsenstein, J. 1985. Confidence limits on phylogeny: An approach using the bootstrap. *Evolution* 39:783-789.

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

Gaut, B.S. and Lewis, P.O. 1995. Success of maximum likelihood phylogeny inference in the four-taxon case. *Mol. Biol. Evol.* 12:152-162.

Goldman, N., Anderson, J.P., and Rodrigo, A.G. 2000. Likelihood-based tests of topologies in phylogenetics. *Syst. Biol.* 49:652-670.

Gu, X., Fu, Y.-X. and Li, W.-H. 1995. Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites. *Mol. Biol. Evol.* 12:546-557.

Hall, B. 2001. Phylogenetic Trees Made Easy: A How-to Manual for Molecular Biologists. Sinauer Associates, Sunderland, Mass.

Hasegawa, M., Kishino, H., and Yano, T. 1985. Dating the human-ape split by a molecular clock of mitochondrial DNA. *J. Mol. Evol.* 22:160-174.

Hendy, M.D. and Penny, D.A. 1989. A framework for the quantitative study of evolutionary trees. *Syst. Zool.* 38:297-309.

Hillis, D.M., Bull, J.J., White, M.E, Badgett, M.R., and Molineux, I.J. 1992. Experimental phylogenetics: Generation of a known phylogeny. *Science* 255:589-592.

Hillis, D.M., Huelsenbeck, J.P., and Swofford, D.L. 1994a. Hobgoblin of phylogenetics? *Nature* 369:363-364.

Hillis, D.M., Huelsenbeck, J.P., and Cunningham, C.W. 1994b. Application and accuracy of molecular phylogenies. *Science* 264:671-677.

Hillis, D.M., Mable, B.K., and Moritz, C. 1996. Applications of molecular systematics: The state of the field and a look to the future. *In* Molecular Systematics, 2nd ed. (D.M. Hillis, C. Moritz, and B.K. Mable, eds.), pp. 515-543. Sinauer Associates, Sunderland, Mass.

Huelsenbeck, J.P. and Hillis, D.M. 1993. Success of phylogenetic methods in the four-taxon case. *Syst. Biol.* 42:247-265.

Huelsenbeck, J.P. 1995. Performance of phylogenetic methods in simulation. *Syst. Biol.* 44:17-48.

Kim, J. 1996. General inconsistency conditions for maximum parsimony: Effects of branch lengths and increasing numbers of taxa. *Syst. Biol.* 45:363-374.

Kishino, H. and Hasegawa, M. 1989. Evolution of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in *Hominoidea*. *J. Mol. Evol.* 29:170-179.

Kishino, H., Miyata, T., and Hasegawa, M. 1990. Maximum likelihood inference of protein phylogeny and the origin of chloroplasts. *J. Mol. Evol.* 30:151-160.

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

Lockhart, P.J., Steel, M.A., Hendy, M.D., and Penny, P. 1994. Recovering evolutionary trees under a more realistic model of sequence evolution. *Mol. Biol. Evol.* 11:605-612.

Maddison, D.R., Swofford, D.L., and Maddison, W.P. 1997. NEXUS: An extensible file format for systematic information. *Syst. Biol.* 46:590-621.

Page, R.D. and Holmes, E.C. 1998. Molecular Evolution: A Phylogenetic Approach. Blackwell Science, Oxford, U.K..

Penny, D. and Hendy, M.D. 1985. Testing methods of evolutionary tree construction. *Cladistics* 1:266-272.

Posada, D. and Crandall, K.A. 1998. MODELTEST: Testing the model of DNA substitution. *Bioinformatics* 14:817-818.

Rogers, J.S. 1997. On the consistency of maximum likelihood estimation of phylogenetic trees from nucleotide sequences. *Syst. Biol.* 46:354-357.

Sanderson, M.J. and Kim, J. 2000. Parametric phylogenetics? *Syst. Biol.* 49:817-829.

Sankoff, D. 1975. Minimal mutation trees of sequences. *SIAM J. Appl. Math.* 28:35-42.

Shimodaira, H. and Hasegawa, M. 1999. Multiple Comparisons of Log-Likelihoods with Applications to Phylogenetic Inference. *Mol. Biol. Evol.* 16:1114-1116.

Steel, M. 1994. Recovering a tree from the Markov leaf colourations it generates under a Markov model. *Appl. Math. Lett.* 7:19-23.

Sullivan, J. and Swofford, D.L. 1997. Are guinea pigs rodents? The utility of models in molecular phylogenetics. *J. Mamm. Evol.* 4:77-86.

Sullivan, J. and Swofford, D.L. 2001. Should we use model-based methods for phylogenetic inference when we know assumptions about among-site rate variation and nucleotide substitution pattern are violated? *Syst. Biol.* 50:723-729.

Swofford, D.L. 2002. PAUP*. Phylogenetic Analysis Using Parsimony (*and Other Methods). Version 4. Sinauer Associates, Sunderland, Mass.

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

Swofford, D.L., Olsen, G.J., Waddell, P.J., and Hillis, D.M. 1996. Phylogenetic inference. *In* Molecular systematics, 2nd ed. (D.M. Hillis, C. Moritz, and B.K. Mable, eds.). pp. 407-514. Sinauer, Sunderland, Mass.

Swofford, D.L., Waddell, P.J., Huelsenbeck, J.P., Foster, P.J., Lewis, P.O., and Rogers, J.S. 2001. Bias in phylogenetic estimation and its relevance to the choice between parsimony and likelihood methods. *Syst. Biol.* 50:525-539.

Templeton, A.R. 1983. Convergent evolution and non-parametric inferences from restriction fragment and DNA sequence data. *In* Statistical Analysis of DNA Sequence Data. (B. Weir, ed.) pp. 151-179. Marcel Dekker, New York.

Tuffley, C., and Steel, M. 1997. Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bull. Math. Biol.* 59:581-607.

Waddell, P.J. and Penny, D. 1996. Evolutionary trees of apes and humans from DNA sequences. *In* Handbook of symbolic evolution (A.J. Lock and C.R. Peters, eds.) pp. 53-73. Clarendon Press, Oxford, U.K.

Yang, Z. 1994a. Estimating the pattern of nucleotide substitution. *J. Mol. Evol.* 39:105-111.

Yang, Z. 1994b. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *J. Mol. Evol.* 39:306-314.

Internet Resources

http://paup.csit.fsu.edu/

PAUP Web site.*

http://pauptech.csit.fsu.edu/~paupforum/

PAUP technical forum.*

http://mailer.csit.fsu.edu/mailman/listinfo/paupinfo/

PAUP information mailing list.*

http://www.sinauer.com/

PAUP publisher, Sinauer Associates, Inc. Web site.*

Key References

Hillis et al., 1996. See above.

A general discussion of issues and controversies pertaining to phylogenetic analyses.

Page and Holmes, 1998. See above.

An accessible introduction to phylogenetic theory, terminology, and practice.

Swofford et al., 1996. See above.

A detailed description of most methods commonly used in phylogenetic inference.

Contributed by James C. Wilgenbusch and
David Swofford
Florida State University
Tallahassee, Florida