A primer to phylogenetic analysis using the PHYLIP package

Jarno Tuimala
Fifth Edition

ISBN 952-5520-02-1

Index

Preface

The purpose of this tutorial is to demonstrate how to use PHYLIP, a collection of phylogenetic analysis software, and some of the options that are available. This tutorial is not intented to be a course in phylogenetics, although some phylogenetic concepts will be discussed briefly. There are other books available which cover the theoretical sides of the phylogenetic analysis, but the actual data analysis work is less well covered.

Here we will mostly deal with molecular sequence data analysis in the current PHYLIP version 3.66.

Commands that the user should type are written with `12 pt Courier` font. File names are written with `12 pt Courier New` font. Output from the programs is represented with `10 pt Courier` font.

I want to thank Joe Felsenstein for his extensive and constructive comments on this text. He has continued to offer feedback for several years, and his efforts are acknowledged with great appreciation.

11th July 2006
Jarno Tuimala

Introduction

What is PHYLIP?

PHYLIP is a comprehensive phylogenetic analysis package created by Joseph Felsenstein at the University of Washington. This package can do many of the phylogenetic analyses available in the literature today. Methods that are available in the package include parsimony, distance matrix, and likelihood methods, including bootstrapping and consensus trees. Data types that can be handled include molecular sequences, gene frequencies, restriction sites, distance matrices, and 0/1 (binary) discrete characters.

Installation

PHYLIP is freely available from:

`http://evolution.genetics.washington.edu/phylip.html`.

It ships with a comprehensive manual covering the usage of different programs. Manual is written in HTML-format, so you can read it using a web browser.

If you're using a *Windows machine*, installation is easy. Download the three zip-files (`phylip.exe`, `phylipwx.exe`, `phylipwy.exe`), and extract them to a preferred folder. The subfolder exe contains all the programs. Manual can be found from the subfolder doc.

For *Macintosh OS X* you may download the packaged disk image (Phylip3.66.dmg). It is compressed, so you need to expand it, and copy the resulting folder to a desired location. Alternatively, you may compile the programs from their sources as outlined in the UNIX installation below. There are source codes and ready made compilations available for older Macintosh systems, Mac OS 8 or 9, also.

Installation for *UNIX systems* is also quite straight-forward. These instruction apply for RedHat-based Linux systems. Installation on main frame can require tweaking of the Makefile. Download the source code and documentation package (phylip-3.66.tar.gz) into a suitable folder. Unzip the package with gzip utility (`gzip –d phylip-3.66.tar.gz`) and expand the tar ball (`tar xvf phylip-3.66.tar`). Move to the newly formed folder containing the source codes (`cd phylip3.6/src`). The folder contains a file called Makefile. Installation of the PHYLIP programs is done simply by typing `make install`. The default Makefile usually works fine.

The draw programs (Drawgram, and Drawtree) need an installation of the X Windows development environment (Athena Widgets), and without it you'll get some error messages during installation, *e.g.*,

```
ld32: WARNING 84 : /usr/lib32/libX11.so is not used for resolving any symbol.
ld32: WARNING 84 : /usr/lib32/libXaw.so is not used for resolving any symbol.
ld32: FATAL   12 : Expecting n32 objects: draw.o is of unknown type.
```

Thus, you might need to change the path to the Widgets on the following lines in the Makefile:

```
# for Linux with X Windows development packages installed
# or for MacOS X with X Windows installed
DFLAGS = $(CFLAGS) -DX -I/usr/X11R6/include


# if the Xlib library for the X windowing system is somewhere
# unexpected, you may have to change the path /usr/X11R6/lib in this one
#
# For gcc for Linux with X windows development packages installed
# or for MacOS X with X windows installed
DLIBS= -L/usr/X11R6/lib/ -lX11 -lXaw -lXt
```

After a successful installation, two new directories appear under the phylip3.66 folder. These contain documentation (doc) and executables (exe).

There is more advice on installing the PHYLIP package in the UNIX environment in the manual, so read it meticulously before proceeding. If everything else fails, you might want to check the site `http://www.biolinux.org/phylip.html` for ready compilations. However, a fresh compilation on your machine might be more up-to-date.

User interface

The programs are controlled through a menu, which asks the users which options they want to set, and allows them to start the computation. The data are read into the program from a text file, which the user can prepare using any word processor or text editor (but it is important that this text file *not* be in the special format of that word processor - it should instead be in flat ASCII or Text Only format). Some sequence alignment programs, like ClustalX and T-Coffee, can write data files in the PHYLIP format.

Most of the programs look for the data in a file called `infile`. If they do not find this file they then ask the user to type in the file name of the data file. Output is written onto special files with names like `outfile` and `outtree`. Trees written onto `outtree` are in the Newick format, an informal standard agreed to in 1986 by authors of a number of major phylogeny packages (Felsenstein, PHYLIP documentation).

Getting started – datafiles and programs

Always keep records

It is very important to keep record of lab-procedures you have done, but it is even more so with computer analyses. You might easily get confused with many many result files, especially, if you have not given them informative names. And, the computer can crash, or the hard-drive may become corrupt, and you can lose your work. After such incidents it is easier to recover the work you have done, if you have kept a good analysis record.

So, always keep records. It indicates you have been working.

Sequence alignment

PHYLIP programs read the aligned sequences in PHYLIP-format. Usually you can recognize the files in this format from the .phy extension associated with the files. Aligned sequences in the suitable format can be produced, e.g., with the program ClustalX, which is freely available from:

`http://www-igbmc.u-strasbg.fr/BioInfo/ClustalX/Top.html`.

Just be sure that you save the aligned sequences in PHYLIP-format.

If you need to edit the alignment (with text-editor) or to do some analyses in the other programs that do not read PHYLIP-format, save the alignment also in the .aln format (Clustal-format). The editing of the Clustal alignment format is easier than the editing of PHYLIP-format, and Clustal will readily read in the .aln format, if you later need to convert the edited sequences into some other format.

Any multiple sequence alignment can also be manually reformatted with a text editor. The format requirements for PHYLIP are rather stringent, and any deviation will result in a program that hangs, usually with the error message Unable to allocate memory.

The file must confirm to the following (Felsenstein, PHYLIP documentation):

1. The file begins with the information about the number of sequences and the number of nucleotides or amino acids in the alignment.
2. The sequence names must be exactly 10 characters long. Spaces can be added to the end of shorter names to make them this length. Do not use Tab characters for this.
3. Gaps must be indicated by - .
4. Missing data or missing information (no sequence) is indicated by ?.
5. Spaces between the alignment blocks are allowed. This normally makes the alignment more readable. Spaces are usually inserted into the alignment every 10 bases or amino acids.
6. Blanks will be ignored, and so will numerical digits. This allows GENBANK and EMBL sequence entries to be read with minimum editing.

Example of the formatted sequences:

```
     5   100
Rabbit     ?????????? ?????????C CAATCTACAC ACGGG-GTAG GGATTACATA
Human      AGCCACACCC TAGGGTTGGC CAATCTACTC CCAGGAGCAG GGAGGGCAGG
Opossum    AGCCACACCC CAACCTTAGC CAATAGACAT CCAGAAGCCC AAAAGGCAAG
Chicken    GCCCGGGGAA GAGGAGGGGC CCGGCGG-AG GCGATAAAAG TGGGGACACA
Frog       GGATGGAGAA TTAGAGCACT TGTTCTTTTT GCAGAAGCTC AGAATAAACG

           TTTGGATGGT AG---GATAT GGGCCTACCA TGGCGTTAAC GGGT-AACGY
           TTTCGACGGT AA---GGTAT TGGCTTACCG TGGCAATGAC AGGT-GACGG
           TTTCGACGGT AA---GGTAT TGGCTTACCG TGGCAATGAC AGGT-GACGY
           TTTCGACGGT AA---GGTAT TGGCTTACCG TGGCAATGAC AGGT-GACGG
           TTTCGATGGT AA---GGTAT TGGCTTACCG TGGCAATGAC AGGT-GACGG
```

Possible ambiguities (such as N, Y or R nucleotides) are also handled correctly, and do not cause trouble.

Font files

In order to be able to use the tree-drawing tools, the font files need to be in the same folder as the Drawtree or Drawgram program(s). If you are using PHYLIP on a PC from the same folder it was installed in, you should not encounter any troubles. However, this is not strictly necessary, just remember to copy the font files with tree drawing programs to the same folder. Or, better still, copy and rename your favorite fontfile as `fontfile` and keep only it with the tree drawing programs. There are six different fonts available:

font1	simple sans-serif Roman
font2	medium quality sans-serif Roman
font3	high quality sans-serif Roman
font4	medium quality sans-serif Italic
font5	high quality sans-serif Italic
font6	Russian Cyrillic

Running PHYLIP programs

The programs are used in a sequential way. The output from the first program is used as an input in the next program. The trick is to know how to use the programs in suitable combinations. See the flow charts in the end of this book for some suggestions.

In Windows, the PHYLIP programs can be invoked by double-clicking on the icon or by typing the name of the program on the command line. It is advisable to use programs from the command line, because then you will be better able to see, *e.g.*, the error messages that might appear. In NT-line Windows versions (NT, 2000 and XP) the DOS prompt, *i.e.*, command line, can be invoked from Start -> All Programs -> Accessories -> Command Prompt.

Most PHYLIP programs run in the same way. The input for a program is taken from a file called `infile` - if the program does not find this file it then asks the user to type in the file name of the data file. The results are written in a file called `outfile`. Some programs may write both `outfile` and a file called `outtree` or `plotfile`.

Because most of the programs use the default names for the input and output files, you need to be sure to rename the files you want to save before proceeding to further analysis. Otherwise you risk losing your results. For example, you get a distance matrix (`outfile`) from the program Dnadist, but you want to try different settings for the matrix calculations. Then, before doing the matrix calculation again, rename `outfile` to `Dnadist_out_F84` or something similar, so that you can tell different analysis results apart after you have ceased to work.

Essential programs

Here is a list of the programs that can be used for the molecular sequence data analysis. The programs are divided into the method categories. The choice of the correct analysis method is left for the user.

Distance methods

These programs are intended to be used sequentially. First a distance matrix is calculated by Dnadist or Protdist program from the multiple sequence alignment. The matrix is then transformed into a tree by Fitch, Kitsch or Neighbor program. Programs Dnadist and Protdist create a file `outfile`. Before running Fitch, Kitsch ot Neighbor, `outfile` should be renamed, either as `infile` ot with another file name. Fitch, Kitsch and Neighbor programs create both `outfile` and `outtree`.

Dnadist	DNA distance matrix calculation
Protdist	Protein distance matrix calculation
Fitch	Fitch-Margoliash tree drawing method without molecular clock
Kitsch	Fitch-Margoliash tree drawing method with molecular clock
Neighbor	Neighbor-Joining and UPGMA tree drawing method

Character based methods

These programs read in the sequence alignment, and produce either one or multiple trees in the output files `outfile` and `outtree`.

Dnapars	DNA parsimony
Dnapenny	DNA parsimony using branch-and-bound
Dnaml	DNA maximum likelihood without molecular clock
Dnamlk	DNA maximum likelihood with molecular clock
Protpars	Protein parsimony
Proml	Protein maximum likelihood

Resampling tool

This program reads in a sequence alignment, and generates a specified number of random samples into a file `outfile`. These random samples are usually used in subsequent analysis as a sequence alignment file with the option M ("use multiple datasets") turned on.

Seqboot	Generates random samples by bootstrapping or jack-knifing

Tree drawing

These programs draw a tree from the specifications in the Newick-format. For example, the specification can be in a file produced by the program Dnaml. The Newick file `outtree`

produced by Dnaml should be renamed to `intree` before visualizing the tree. Drawgram and Drawtree produce a file `plotfile`, whereas Retree saves the result in a file `outtree`.

Drawgram	Draws a rooted tree
Drawtree	Draws an unrooted tree
Retree	Interactive tree-rearrangement

Consensus trees

This program constructs a consensus tree from multiple trees. For example, Dnapars can produce multiple trees, which can be summarized by the program Consense. Also the results of bootstrapping are summarized by the program Consense as a majority rule tree.

Consense	Draws consensus trees from multiple trees

Tree distances

This program computes, *e.g.*, a topology-based distance between two or more trees. The distance can be used to assess or compare the results from different analyses.

Treedist	Computes distances between trees based on tree topology

Quick start

Here a DNA sequence data is used as an example. In the example using PHYLIP in Windows operating system, three programs are used to construct and plot a tree by Neighbor joining (a distance method) using the F84 evolutionary model. Details about other methods are available in the succeeding sections.

Distance methods

Align your DNA sequences and save the alignment in PHYLIP-format as `alignment.phy`. Start the program Dnadist by typing `Dnadist` to the command prompt or double clicking on the program's icon.

First Dnadist (and all the other programs also) checks whether there is a file `infile` in the folder you started the program in. If it does not find `infile` it asks you to type in the name of the sequence alignment file.

```
Dnadist: can't find input file "infile"
Please enter a new file name> alignment.phy
```

Note that the programs are easiest to use if both the programs and the datafiles are in the same folder as in the example above. If datafiles are in a different folder, you can type in the whole path to the file, e.g., if the files were in the folder `D:\data` you would type

```
Dnadist: can't find input file "infile"
Please enter a new file name> D:\data\alignment.phy
```

All PHYLIP programs are menu-driven. Below is the menu written by Dnadist. Every line in the menu starts with a capital letter or number. You can change the settings of the program by typing in the letter or the number in front of the option you would like to change. For example, typing "d" and pressing Enter, would cycle through different evolutionary models implemented in Dnadist. After you are satisfied with the settings (for this quick start, do not change any options), you should type in "y" and press Enter. This starts the run.

```
Nucleic acid sequence Distance Matrix program, version 3.66

Settings for this run:
  D  Distance (F84, Kimura, Jukes-Cantor, LogDet)?  F84
  G          Gamma distributed rates across sites?  No
  T                 Transition/transversion ratio?  2.0
  C            One category of substitution rates?  Yes
  W                        Use weights for sites?  No
  F                Use empirical base frequencies?  Yes
  L                       Form of distance matrix?  Square
  M                    Analyze multiple data sets?  No
  I                   Input sequences interleaved?  Yes
  0            Terminal type (IBM PC, ANSI, none)?  ANSI
  1             Print out the data at start of run  No
  2           Print indications of progress of run  Yes

  Y to accept these or type the letter for one to change
y
```

Dnadist prints indications of the run (below). After it has finished calculating all the pairwise distances between the sequences, it tells you so (`Done.`). These pairwise distances are saved in a file `outfile`. The file contains just plain text, and you may want rename the file as `outfile.txt` so that it opens automatically in Notepad when you double-click it.

```
Distances calculated for species
    Rabbit      ....
    Human       ...
    Opossum     ..
    Chicken     .
    Frog

Distances written to file "outfile"

Done.
```

Next rename `outfile` as `infile`, and run the program Neighbor (type in `Neighbor`). The next menu should appear. Now Neighbor has read the pairwise distances from the file `infile`, and does not ask you for a new filename. You can again modify the settings to your liking, but for this quick start just type in `y` and press Enter.

```
Neighbor-Joining/UPGMA method version 3.66

Settings for this run:
  N       Neighbor-joining or UPGMA tree?  Neighbor-joining
  O                       Outgroup root?  No, use as outgroup species  1
  L         Lower-triangular data matrix?  No
  R         Upper-triangular data matrix?  No
  S                       Subreplicates?  No
  J     Randomize input order of species?  No. Use input order
  M           Analyze multiple data sets?  No
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                       Print out tree  Yes
  4       Write out trees onto tree file?  Yes


  Y to accept these or type the letter for one to change
y
```

Like Dnadist, Neighbor also prints out indications of the run. After completing the analysis, the program tells you so (`Done.`).

```
Cycle   2: OTU   4 (   0.62698) joins OTU   5 (   0.95492)
Cycle   1: OTU   3 (   0.73871) joins node  4 (   0.17009)
last cycle:
 OTU   1  ( 0.05116) joins OTU   2  ( 0.23064) joins node  3  ( 0.12944)

Output written on file "outfile"

Tree written on file "outtree"

Done.
```

The tree is now contained in the files `outfile` and `outtree`. You can view the graphical tree in outfile by opening it in some text editor. Neighbor has now drawn the following tree from our example data set.

```
     +---------------------Opossum
  +---2
  !   !    +-----------------Chicken
  !   +----1
  !        +---------------------------Frog
  !
  3-Rabbit
  !
  +------Human
```

Note that the tree should be viewed using a font such as Courier, where all the letters take the same amount of space. Otherwise the nodes and branches might become disconnected.

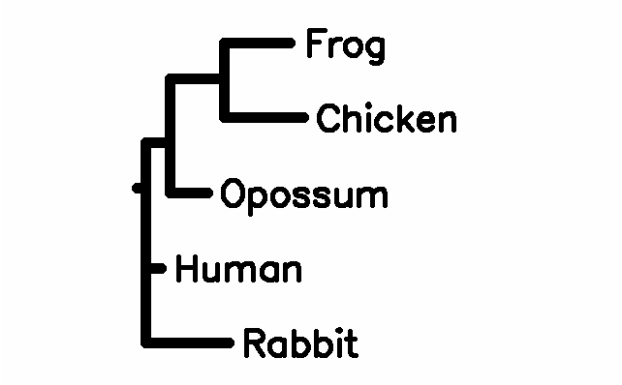
Tree drawing

Next you can draw a nicer looking graphical tree from the file `outtree` using the program Drawgram. First, rename the file `outtree` as `intree`, and start the program Drawgram. Drawgram first searches for a file called `fontfile` from the current folder, and if it is unable to find it (if you have not renamed one of the original font files as `fontfile`), it asks for the name of the font file. You should then specify which font to use by typing in its name (`font1` – `font6`). After specifying that, the menu of the program appears. Now you should change the final plotting devise as MS-Windows Bitmap using the option P. Postscript output format is ideally suitable for publication images, but it is slightly complicated to use on a basic Windows machine. The program also asks for the dimensions of the tree – you might initially try 640 x 400. The settings are accepted by typing in "y" and pressing Enter.

```
DRAWGRAM from PHYLIP version 3.66
Reading tree ...
Tree has been read.
Loading the font ...
Drawtree: can't find font file "fontfile"
Please enter a new file name> font1
Font loaded.

Rooted tree plotting program version 3.66

Here are the settings:
 0  Screen type (IBM PC, ANSI):  IBM PC
 P       Final plotting device:  Postscript printer
 V           Previewing device:  MS Windows display
 H                  Tree grows:  Horizontally
 S                  Tree style:  Phenogram
 B          Use branch lengths:  Yes
 L             Angle of labels:  90.0
 R      Scale of branch length:  Automatically rescaled
 D       Depth/Breadth of tree:  0.53
 T      Stem-length/tree-depth:  0.05
 C    Character ht / tip space:  0.3333
 A             Ancestral nodes:  Weighted
 F                        Font:  Times-Roman
 M          Horizontal margins:  1.65 cm
 M            Vertical margins:  2.16 cm
 #              Pages per tree:  one page per tree

 Y to accept these or type the letter for one to change
y
```

Drawgram opens a new window, where you can see a preview of the tree. If you're satisfied with the results, select from the File menu (in the same newly-opened window) "plot". A new file (`plotfile`) should now appear in the current folder. If you rename it as `plotfile.bmp` you would be able to open it in some graphics package for more modifications.

The final picture looks like this:



The length of the branch is the number of nucleotide or amino acid changes that are expected in that particular branch in the tree. The number of changes is estimated using the evolutionary model specified in the Dnadist program.

Amino acid sequences

The sequence of the programs is similar to the one presented here one with one important exception. When using amino acid sequences for inferring phylogenies with distance methods, the distance matrix is calculated using program Protdist, not Dnadist.

Basic analyses in more detail

Before proceeding further, please read the previous sections. They contain relevant information, if you haven't used PHYLIP earlier.

There are three different ways to analyze DNA or amino acid sequence data in PHYLIP. Parsimony and maximum likelihood are character based methods, which means that they treat every single site of the multiple sequence alignment independently. Distance methods summarize the differences between sequences by calculating a pairwise distance measure between all aligned sequences. In either case, after the data analysis and tree-drawing, the validity of data can (or should, according to some) be assessed by a bootstrap analysis.

The programs can be invoked by double-clicking on their icons. If you have done some analyses before within the same folder, the program detects that `outfile` already exists, and it ask you to:

```
Dnadist: the file "outfile" that you wanted to
     use as output file already exists.
     Do you want to Replace it, Append to it,
     write to a new File, or Quit?
     (please type R, A, F, or Q)
```

Replace means to overwrite, Append adds the new results to the end of an existing file, and File asks for a new file name. You can also Quit the program.

Distance methods

For the distance method analysis you'll need at least two programs. Dnadist or Protdist calculates a matrix of pairwise distances between every sequence in the file. The tree is inferred from those distances by Neighbor, Fitch or Kitsch program.

DNA data

The Dnadist program writes out a menu:

```
Nucleic acid sequence Distance Matrix program, version 3.66

Settings for this run:
  D  Distance (F84, Kimura, Jukes-Cantor, LogDet)?  F84
  G          Gamma distributed rates across sites?  No
  T                Transition/transversion ratio?  2.0
  C            One category of substitution rates?  Yes
  W                        Use weights for sites?  No
  F                Use empirical base frequencies?  Yes
  L                       Form of distance matrix?  Square
  M                    Analyze multiple data sets?  No
  I                   Input sequences interleaved?  Yes
  0              Terminal type (IBM PC, ANSI, none)?  ANSI
  1               Print out the data at start of run  No
  2             Print indications of progress of run  Yes

  Y to accept these or type the letter for one to change
```

The settings can be changed by typing in the letter before the option, and pressing Return. For example, typing "d" and return, would cycle through the different distance calculation methods. These distances are also called evolutionary models. Evolutionary models are mathematical formulas which try to compensate for the multiple substitution problem and transition-transversion bias.

Briefly, Jukes-Cantor distance assumes that all substitutions are equally likely to happen. Kimura distance has two different change rates (rate parameters), one for transitions and the other for transversions. These models also assume that the equilibrium frequencies of all the bases are 0.25. F84 distance has two rate parameters, one for transitions, and the other for transversions, but also allows the equilibrium frequencies of the bases to differ from each other. The LogDet distance should be used if there are (large) base frequency differences between sequences in the tree. The LogDet distance cannot cope with ambiguity codes. It must have completely defined sequences.

The transition/transversion ratio (Ts/Tv) can be modified if more detailed data is available on an Ts/Tv ratio. If not, the Ts/Tv ratio of 1-2 is often a good approximation to the situation with most mammalian nuclear genes.

Usually the phylogenetic methods assume that all the sites are evolving at the same speed. This is clearly an unrealistic assumption. For example, third codon positions evolve with a higher speed than second positions. That is because most of the substitutions in the third position are selectively neutral whereas substitutions in the second codon position always lead to amino acid changes. This rate heterogeneity can be treated by using gamma distribution. The shape of the gamma distribution is defined by a parameter called alpha (α). If you activate option "g" you'll be prompted to enter this shape parameter:

```
Coefficient of variation of substitution rate among sites (must be
positive)

In gamma distribution parameters, this is 1/(square root of alpha)
```

If you know alpha, you can calculate the prompted CV by the equations (Felsenstein, PHYLIP documentation):

$1 / \sqrt{}\,(\text{alpha}) = \text{CV}$
$\text{alpha} = (1 / \text{CV})^2$

A good approximation of the alpha for many protein coding genes is 0.5, but you can also estimate this value using programs such as TREE-PUZZLE, which is not a part of PHYLIP package. For a maximum likelihood solution using PHYLIP programs, you can apply the example in the section *Estimating transversion/transition ratio*.

Another layer of rate variation also is available. The option "c" allows user-defined rate categories. The user is prompted for the number of user-defined rates, and for the rates themselves, which cannot be negative but can be zero. These numbers are defined relative to each other, so that if rates for three categories are set to 1 : 3 : 2.5 this would have the same meaning as setting them to 2 : 6 : 5. The assignment of rates to sites is then made by reading a file whose default name is "categories". It should contain a string of digits 1 through 9. A new line or a blank can occur after any character in this string.

Thus the categories file might look like this:

```
12223111112241115511553333333444
```

“The user can assign categories of rates to each site (for example, we might want first, second, and third codon positions in a protein coding sequence to be three different categories. This is done with the categories input file and the option "c". We then specify (using the menu) the relative rates of evolution of sites in the different categories. For example, we might specify that first, second, and third positions evolve at relative rates of 1.0, 0.8, and 2.7.” (Felsenstein, PHYLIP documentation).

“The weights-option ("w") allows us to specify weights on the individual characters. The weights cause a character to be counted as if it were *n* characters, where *n* is the weight. By use of the weights we can give weight to some characters, and drop others from the analysis. In the molecular sequence programs only two values of the weights, 0 or 1 are allowed, except for Dnapars”. (Felsenstein, PHYLIP documentation) An exception to this is Dnapars which accept weights from 0-9…35 (0-9 with numbers and 10-35 with letters). For more information on weighting, see the section *Weighting* in *advanced topics* chapter.

If you want to specify the base frequencies by yourself, you can do that by invoking the option "f". The program then prompts you to type in the frequencies of different bases.

After you have modified the settings to your liking, you can calculate the distance matrix by typing "y" and pressing return.

This will produce a distance matrix with default name `outfile`:

```
    5
Rabbit      0.0000  0.2818  0.9386  0.9830  1.2617
Human       0.2818  0.0000  1.0795  1.1496  1.5312
Opossum     0.9386  1.0795  0.0000  1.5380  1.8615
Chicken     0.9830  1.1496  1.5380  0.0000  1.5819
Frog        1.2617  1.5312  1.8615  1.5819  0.0000
```

The `outfile` needs to be renamed, because the programs will go awry if they try both read and write from and to the same file. After renaming this new file can be used as an input into the Neighbor, Fitch, or Kitsch program.

Neighbor writes out a menu, where you can again change the settings:

```
Neighbor-Joining/UPGMA method version 3.66

Settings for this run:
  N       Neighbor-joining or UPGMA tree?  Neighbor-joining
  O                       Outgroup root?  No, use as outgroup species  1
  L         Lower-triangular data matrix?  No
  R         Upper-triangular data matrix?  No
  S                       Subreplicates?  No
  J     Randomize input order of species?  No. Use input order
  M           Analyze multiple data sets?  No
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                       Print out tree  Yes
  4       Write out trees onto tree file?  Yes


  Y to accept these or type the letter for one to change
```

At this point, you can specify the outgroup of the tree. A sister taxon to the ingroup (the taxa under investigation) is rather often used as an outgroup. For example, if human, chimpanzee and gorilla are under scrutiny, orangutan could be specified as an outgroup.

Also Fitch writes out a menu, where setting can be modified, but the menu looks a bit different:

```
Fitch-Margoliash method version 3.66

Settings for this run:
  D      Method (F-M, Minimum Evolution)?  Fitch-Margoliash
  U                  Search for best tree?  Yes
  P                                Power?  2.00000
  -      Negative branch lengths allowed?  No
  O                       Outgroup root?  No, use as outgroup species  1
  L         Lower-triangular data matrix?  No
  R         Upper-triangular data matrix?  No
  S                       Subreplicates?  No
  G                 Global rearrangements?  No
  J     Randomize input order of species?  No. Use input order
  M           Analyze multiple data sets?  No
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                       Print out tree  Yes
  4       Write out trees onto tree file?  Yes

  Y to accept these or type the letter for one to change
```

Fitch-Margoliash is the distance based optimization method, which mean that it searches for a tree with the smallest squared distance between the distances and their predictions from the tree.

In Fitch you can also randomize the input order of the sequences with option "j", jumble. Often the input order of the sequences affects the outcome of the analysis. This can be assessed by randomizing the input order. The program also asks you to specify the number of times you want to randomize the input order of the sequences. It is advisable to do jumbling at least 10 times, because it almost certainly improves the results.

You also have the option of using a user-defined tree with option "u" which controls whether negative branch lengths are allowed. In this case the program will, as default, read the user-defined tree from the file `intree`. This also activates a new option "n". If you choose to use the branch lengths from the user tree, Fitch calculates the Sum of Squares and the Average Percent Standard Deviation for the user-defined tree.

Confirm the setting by typing "y" and pressing Return. Neighbor creates two new files, `outfile` and `outtree`.

Outfile contains detailed information about the analysis and its results. It also contains the inferred tree drawn with symbol graphics. Also the estimated branch lengths are reported in the file.

```
   5 Populations

Neighbor-Joining/UPGMA method version 3.6

 Neighbor-joining method

 Negative branch lengths allowed


      +---------------------Opossum
  +---2
  !   !    +------------------Chicken
  !   +----1
  !        +----------------------------Frog
  !
  3-Rabbit
  !
  +------Human

remember: this is an unrooted tree!

Between        And            Length
-------        ---            ------
   3          Human             0.23064
   3             2              0.12944
   2          Opossum           0.73871
   2             1              0.17009
   1          Chicken           0.62698
   1          Frog              0.95492
   3          Rabbit            0.05116
```

File `outtree` contains the tree in a computer-readable format. It also reports the branch lenghts (the numerical values reported after : marks), which are the same as in the file `outfile`.

```
(Human:0.22996,((Frog:0.95134,Chicken:0.63056):0.16672,Opossum:0.74182):0.1
2891,Rabbit:0.05184);
```

Protein data

The protein distance method works more or less similarly as with DNA data. The program Protdist writes out a menu with modifiable settings. After you have modified them to your liking, the program produces an `outfile`, which contains pairwise distances. Those distances can be transformed into a tree as has been described in the DNA data-section.

```
Protein distance algorithm, version 3.66

Settings for this run:
  P  Use JTT, PMB, PAM, Kimura, categories model?  Jones-Taylor-Thornton
  G  Gamma distribution of rates among positions?  No
  C           One category of substitution rates?  Yes
  W                    Use weights for positions?  No
  M                   Analyze multiple data sets?  No
  I                  Input sequences interleaved?  Yes
  0                Terminal type (IBM PC, ANSI)?  ANSI
  1            Print out the data at start of run  No
  2          Print indications of progress of run  Yes

Are these settings correct? (type Y or the letter for one to change)
```

There are five options of evolutionary models to choose from: JTT (Jones-Taylor-Thornton), PAM (Dayhoff), PMB, Kimura and categories. The PAM model uses the DCMut version of Dayhoff's original PAM model for calculations. JTT and PAM are widely used amino acid substitution matrices (models), and PMB is a much resent model derived from conserved blocks in the Blocks database. The Kimura model corrects for multiple hits, and categories is a model put together by Felsenstein. See the PHYLIP documentation for more information.

```
Computing distances:
  CAS1_HUMAN
  CAS1_RABIT    .
  CAS1_MOUSE    ..
  CAS1_BOVIN    ...
  CAS1_SHEEP    ....
  CAS1_PIG      .....

Output written to output file
```

Parsimony methods

DNA data

The programs available for DNA parsimony are Dnapars and Dnapenny. The differences between these DNA parsimony programs are in the algorithm. Dnapars searches for the most parsimonious tree by a heuristic algorithm which does not guarantee that the shortest tree is found. Program Dnapenny uses the branch-and-bound algorithm, which guarantees that the shortest tree is found, but takes quite much computer time.

Program Dnapars is controlled through this menu:

```
DNA parsimony algorithm, version 3.66

Setting for this run:
  U                 Search for best tree?  Yes
  S                       Search option?  More thorough search
  V              Number of trees to save?  10000
  J   Randomize input order of sequences?  No. Use input order
  O                        Outgroup root?  No, use as outgroup species  1
  T              Use Threshold parsimony?  No, use ordinary parsimony
  N           Use Transversion parsimony?  No, count all steps
  W                         Sites weighted?  No
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                        Print out tree  Yes
  4          Print out steps in each site  No
  5  Print sequences at all nodes of tree  No
  6       Write out trees onto tree file?  Yes

  Y to accept these or type the letter for one to change
```

Invoking the option "s" will allow you to specify more or less thorough search. The more thorough search will save multiple equally parsimonious trees without collapsing those branches that do not have support (no evidence of any change happened in that branch), and it does rearrangements on all parts of those trees. Less thorough search collapses non-supported branches before rearrangements. This leads to fewer rearrangements and faster analysis. You can even decide to do rearrangements on one tree only. This means that only a random one of the multiple equally parsimonious trees is rearranged. The search is then much more restricted, which is not necessarily a good situation.

Often parsimony analysis produces many equally parsimonious trees, which can then be analyzed more closely with, *e.g.*, maximum likelihood or some other methods. You can specify the number of trees to save with option "v".

Sometimes it is a good practice to limit the number of changes one site can contribute to the tree. This can be accomplished with option "t". With "t" you can specify a threshold above which all the changes are regarded as not counting further.

Transversion parsimony (option "n") uses only transversions for the parsimony analysis. This tries to remove the bias resulting from the more rapid accumulation of transition substitutions to the DNA.

The Option "5" can be used to infer ancient states of sites. If you choose to print sequences at all nodes of the tree, the `outfile` will contain more information than normally:

```
From    To     Any Steps?  State at upper node
                           . means same as in the node below it on tree
          1                GTYCAGGGCT -GGGCATAAA AGGCAGAGCA GGGCCAGCTR
   1      2         yes    ...NR..... .?........ .......... A.RT......
   2      3         yes    ..C..CB... ..CA.VDGHD T.V...CV.. ........A.
   3   Frog         yes    C..AA.TTTG GC..TGG.TT ..A....A.. T.A..GT..G
   3   Chicken      yes    .A.GG.C... .-...CA.CG ..CT.T.C.C .CGGG....A
   2   Opossum      yes    ..TTG...GC CA..G..... .........T ..A.T..T.T
   1   Human        yes    AGC....... .......... ..T...G... .A....T..A
   1   Rabbit       yes    ..T....A.. T......... .......... ....-....G

          1                CTGCTTACAH
   1      2         yes    .W.....A.M
   2      3         yes    .....C....
   3   Frog         yes    .T.A....CA
   3   Chicken      yes    GA..C..G.C
   2   Opossum      yes    .A..A.C.TA
   1   Human        yes    T........T
   1   Rabbit      maybe   .........C
```

Program Dnapenny has a different kind of menu:

```
Penny algorithm for DNA, version 3.66
 branch-and-bound to find all most parsimonious trees

Settings for this run:
  H        How many groups of  100 trees:  1000
  F        How often to report, in trees:   100
  S           Branch and bound is simple?  Yes
  O                     Outgroup root?  No, use as outgroup species  1
  T               Use Threshold parsimony?  No, use ordinary parsimony
  W                     Sites weighted?  No
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                       Print out tree  Yes
  4         Print out steps in each site  No
  5  Print sequences at all nodes of tree  No
  6       Write out trees onto tree file?  Yes

Are these settings correct? (type Y or the letter for one to change)
```

You can specify ("h") how many hundreds of trees you want to search, and how often the report is printed on the screen ("f").

You can also cause the branch-and-bound algorithm to reconsider the order of the species with option "s". This will cause the analysis to take a longer time, but according to Felsenstein “it might prove of use on some data sets [of intermediate messiness]”.

Program Dnapars gives only a little information during the run:

```
Adding species:
   1. Rabbit
   2. Human
   3. Opossum
   4. Chicken
   5. Frog
```

```
Doing global rearrangements on all trees tied for best

  !---------!
   .........
   .........

Output written to output file

Trees also written onto tree file

Done.
```

Dnapenny gives information about the search in real time, so you can estimate how long the run is going to take:

```
How many
trees looked                                        Approximate
at so far      Length of        How many            percentage
(multiples     longest tree     trees this long     searched
of  100):       found so far     found so far        so far
----------     ------------     ------------        ------------
1               89               1                   50 %
2               89               1                   100 %

Output written to output file

Trees also written onto tree file
```

Both programs produce files `outfile` and `outtree`, but they look a bit different. Dnapars infers the branching order of the tree and estimates the branch lengths, but Dnapenny only infers the branching order. The length of the tree is on a line “requires a total of xxxx.xx”. A smaller value means a more parsimonious tree.

Thus, the `outfile` and `outtree` of Dnapars:

```
DNA parsimony algorithm, version 3.6

One most parsimonious tree found:

                     +--------------------Frog
            +--------3
  +---------2        +-----------------Chicken
  |         |
  |         +----------------Opossum
  |
  1----------Human
  |
  +------------Rabbit

requires a total of   3019.000

  between      and       length
  -------      ---       ------
     1           2       0.181782
     2           3       0.150139
     3      Frog         0.347616
     3      Chicken      0.304468
     2      Opossum      0.291273
     1      Human        0.182778
     1      Rabbit       0.219167
```

```
(((Frog:0.34762,Chicken:0.30447):0.15014,Opossum:0.29127):0.18178,Human:0.1
8278,Rabbit:0.21917);
```

look different from those produced by Dnapenny:

```
Penny algorithm for DNA, version 3.6
 branch-and-bound to find all most parsimonious trees


requires a total of           3019.000

One most parsimonious tree found:


  +-----------Rabbit
  !
  !        +--Frog
  1     +--3
  !  +--2  +--Chicken
  !  !  !
  +--4  +-----Opossum
     !
     +--------Human

  remember: this is an unrooted tree!


(Rabbit,(((Frog,Chicken),Opossum),Human));
```

Protein data

The protein parsimony program Protpars is comparable to Dnapars, and its menu looks very similar:

```
Protein parsimony algorithm, version 3.66

Setting for this run:
  U                 Search for best tree?  Yes
  J   Randomize input order of sequences?  No. Use input order
  O                       Outgroup root?  No, use as outgroup species  1
  T              Use Threshold parsimony?  No, use ordinary parsimony
  C               Use which genetic code?  Universal
  W                       Sites weighted?  No
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                        Print out tree  Yes
  4          Print out steps in each site  No
  5  Print sequences at all nodes of tree  No
  6       Write out trees onto tree file?  Yes

Are these settings correct? (type Y or the letter for one to change)
```

You can specify the genetic code used for the analysis by option "c". You have an option to choose from the universal code and four mitochondrial codes.

However, its `outfile` and `outtree` are more similar to the output of Dnapenny:

```
Protein parsimony algorithm, version 3.6

One most parsimonious tree found:

           +-----CAS1_PIG
         +--5
         !  !  +--CAS1_SHEEP
      +--3  +--4
      !  !     +--CAS1_BOVIN
   +--2  !
   !  !  +--------CAS1_MOUSE
   1  !
   !  +-----------CAS1_RABIT
   !
   +--------------CAS1_HUMAN

   remember: this is an unrooted tree!


requires a total of   1061.000


((((CAS1_PIG,(CAS1_SHEEP,CAS1_BOVIN)),CAS1_MOUSE),CAS1_RABIT),
CAS1_HUMAN);
```

Maximum likelihood methods

There are two maximum likelihood programs for DNA data, and two for protein data. Dnaml uses a maximum likelihood method without molecular clock, and Dnamlk assumes a molecular clock. Proml does not and promlk does assume molecular clock.

DNA data

Menu of Dnaml:

```
Nucleic acid sequence Maximum Likelihood method, version 3.66

Settings for this run:
  U                 Search for best tree?  Yes
  T        Transition/transversion ratio:  2.0000
  F       Use empirical base frequencies?  Yes
  C                One category of sites?  Yes
  R           Rate variation among sites?  constant rate
  W                       Sites weighted?  No
  S        Speedier but rougher analysis?  Yes
  G                Global rearrangements?  No
  J   Randomize input order of sequences?  No. Use input order
  O                        Outgroup root?  No, use as outgroup species  1
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                        Print out tree  Yes
  4       Write out trees onto tree file?  Yes
  5   Reconstruct hypothetical sequences?  No

  Y to accept these or type the letter for one to change
```

Most of the options has already been covered in the chapters above. However, there two new option, "s" and "g" with which you can modify the search parameters a bit.

“The option "s" turns on or off the search method which iterates or does not iterate the branch lengths in all topologies. Turning this option off ("No, not rough") will cause the program to run more slowly, but it will also be a bit more likely to find the tree topology of highest likelihood.” (Felsenstein, PHYLIP documentation)

“The "g" (global search) option causes, after the last species is added to the tree, each possible group to be removed and re-added. This improves the result, since the position of every species is reconsidered. It approximately triples the run-time of the program.” (Felsenstein, PHYLIP documentation) This is equivalent to the rearrangements done in the parsimony program Dnapars. Specifically, the program uses the SPR (subtree pruning and regrafting) method for rearranging (or as others say, swapping) the trees.
“If more than one category is specified, then another option, "a", becomes visible in the menu. This allows us to specify that we want to assume that sites that have the same regional rate category are expected to be clustered so that there is autocorrelation of rates. The program asks for the value of the average patch length. This is an expected length of patches that have the same rate. If it is 1, the rates of successive sites will be independent. If it is, say, 10.25, then the chance of change to a new rate will be 1 / 10.25 after every site. However the "new rate" is randomly drawn from the mix of rates, and hence could even be the same. So the actual observed length of patches with the same rate will be a bit larger than 10.25. Note below that if you choose multiple patches, there will be an estimate in the output file as to which combination of rate categories contributed most to the likelihood.” (Felsenstein, PHYLIP documentation)

The menu of the Dnamlk:

```
Nucleic acid sequence
   Maximum Likelihood method with molecular clock, version 3.66

Settings for this run:
  U                 Search for best tree?  Yes
  T        Transition/transversion ratio:  2.0
  F       Use empirical base frequencies?  Yes
  C   One category of substitution rates?  Yes
  R           Rate variation among sites?  constant rate
  G                Global rearrangements?  No
  W                      Sites weighted?  No
  J   Randomize input order of sequences?  No. Use input order
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                       Print out tree  Yes
  4       Write out trees onto tree file?  Yes
  5   Reconstruct hypothetical sequences?  No

Are these settings correct? (type Y or the letter for one to change)
```

The `outfile` of the Dnaml is quite similar to the outfile of Dnamlk. The first of these is the `outfile` from Dnaml analysis, and the latter is the same file from Dnamlk.

```
Nucleic acid sequence Maximum Likelihood method, version 3.6

Empirical Base Frequencies:

   A       0.25650
   C       0.21951
   G       0.22091
  T(U)     0.30309

Transition/transversion ratio =   2.000000

  +-----Human
  |
  |           +--------------------------Frog
  |     +-----3
  1-----2     +------------------Chicken
  |     |
  |     +--------------------Opossum
  |
  +--Rabbit

remember: this is an unrooted tree!

Ln Likelihood = -9695.14457

 Between        And            Length      Approx. Confidence Limits
 -------        ---            ------      ------- ---------- ------

    1          Rabbit            0.09743     (  0.06975,     0.12511) **
    1          Human             0.20090     (  0.16716,     0.23474) **
    1             2              0.18632     (  0.12823,     0.24451) **
    2             3              0.19152     (  0.11910,     0.26384) **
    3          Frog              0.90555     (  0.78325,     1.02776) **
    3          Chicken           0.58309     (  0.49455,     0.67166) **
    2          Opossum           0.69201     (  0.60518,     0.77894) **

     *  = significantly positive, P < 0.05
     ** = significantly positive, P < 0.01

Nucleic acid sequence
   Maximum Likelihood method with molecular clock, version 3.6


Empirical Base Frequencies:

   A       0.25650
   C       0.21951
   G       0.22091
  T(U)     0.30309

Transition/transversion ratio =   2.000000

  +--------------------------------------Frog
--4
  !      +--------------------------------Chicken
  +------3
         !       +-----------------------Opossum
         +-------2
                 !                 +-------Human
                 +-----------------1
                                   +-------Rabbit
```

```
Ln Likelihood = -9714.14764

Ancestor      Node      Node Height     Length
 --------      ----      ---- ------     ------
 root             4
   4          Frog          0.77626      0.77626
   4             3          0.13842      0.13842
   3          Chicken       0.77626      0.63785
   3             2          0.27815      0.13973
   2          Opossum       0.77626      0.49812
   2             1          0.62602      0.34787
   1          Human         0.77626      0.15025
   1          Rabbit        0.77626      0.15025
```

As you can see, Dnaml produces confidence intervals of the branch lengths of the tree, but Dnamlk doesn't. Dnaml also produces a rough estimate of the p-value that the length of the branch is significantly greater than zero. This test is done by comparing the tree with the inferred branch length to a tree where the same branch has been scaled to be zero. This result is only an approximation, so do not rely on it when interpreting the results.

The reported likelihood (Ln Likelihood) is actually a natural logarithm of the likelihood. The closer the likelihood value is to zero, the better. It can be used for statistical testing of hypothesis, as will be discussed in the advanced analysis chapter.

The `outtree` from both programs is identical in form except that the tree has a three-way fork at its base:

```
(Human:0.20090,((Frog:0.90555,Chicken:0.58309):0.19152,
Opossum:0.69201):0.18632,Rabbit:0.09743);
```

Protein data

The startup menu of the program Proml is similar to Dnaml:

```
Amino acid sequence Maximum Likelihood method, version 3.66

Settings for this run:
  U                 Search for best tree?  Yes
  P    JTT, PMB or PAM probability model?  Jones-Taylor-Thornton
  C               One category of sites?  Yes
  R           Rate variation among sites?  constant rate of change
  W                       Sites weighted?  No
  S        Speedier but rougher analysis?  Yes
  G                Global rearrangements?  No
  J   Randomize input order of sequences?  No. Use input order
  O                        Outgroup root?  No, use as outgroup species  1
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                        Print out tree  Yes
  4       Write out trees onto tree file?  Yes
  5   Reconstruct hypothetical sequences?  No

  Y to accept these or type the letter for one to change
```

The program Proml assumes no molecular clock, and produces output files which are nearly identical to the output files of Dnaml.

Resampling procedure

The idea behind resampling (bootstrapping and other methods) is to assess how reliable a tree we can produce with the dataset at hand. Initially, the sequence alignment is analyzed in the usual way. Then, resampling proceeds (see the flow chart in the end of this book, also) by first creating a number (100-10000) of random datasets from the original dataset. These random datasets are analyzed in exactly the same way the original dataset was analyzed, and the results from the random datasets are summarized by constructing a majority rule consensus tree (program Consense). Resampling with replacement (bootstrapping) can be done with the program Seqboot:

```
Bootstrapping algorithm, version 3.66

Settings for this run:
  D      Sequence, Morph, Rest., Gene Freqs?  Molecular sequences
  J  Bootstrap, Jackknife, Permute, Rewrite?  Bootstrap
  %    Regular or altered sampling fraction?  regular
  B      Block size for block-bootstrapping?  1 (regular bootstrap)
  R                    How many replicates?  100
  W               Read weights of characters?  No
  C                 Read categories of sites?  No
  S     Write out data sets or just weights?  Data sets
  I             Input sequences interleaved?  Yes
  0      Terminal type (IBM PC, ANSI, none)?  ANSI
  1       Print out the data at start of run  No
  2     Print indications of progress of run  Yes

  Y to accept these or type the letter for one to change
```

You have several options to choose from. First, there are three resampling procedures available in the Seqboot program (option "j"). Bootstrapping creates (with block size=1, option "b") new data sets, which are of the same size as the original sequence alignment. In bootstrapping resamples, every sequence site is represented in the alignment a random number of times, including possibly zero times. The jackknife deletes a random 50% of sites from the original alignment, and none of the sites can occur in the resamples more than once. Permutation of species within each site produces data matrices that have the same number and kinds of characters but no taxonomic structure. It is used for different purposes than the bootstrap, as it tests not the variation around an estimated tree but the hypothesis that there is no taxonomic structure in the data: if a statistic such as number of steps is significantly smaller in the actual data than it is in replicates that are permuted, then we can argue that there is some taxonomic structure in the data (though perhaps it might be just a pair of sibling species). The program also converts PHYLIP-formatted data into other formats (Nexus and XML) using the “rewrite” selection (option J).

With option "r" we set the number of resampled data matrixes produced.

You can also use weights with the bootstrapping procedure (option "w"). If you want to save hard disk space, you can generate, a file containing sets of weights insetad of new datasets. These weights can then be used in the same way as the new bootstrapped dataset. Instead of using multiple datasets in the analysis program, you then need to use these sets of weights.

After accepting the settings, Seqboot asks for a random number. Pick an odd random number. Just keep in mind that using the same random number produces always exactly the same result. If you need to run bootstrapping several times, change the random number between the runs.

Accepting the settings produces a file where there are 100 random samples:

```
   6   333
CAS1_MOUSE   KLCLAAAFMR RHHHSSNNNN VSSSSQQQQQ QQHHSSE--- --------II IFQQPKYYYL
CAS1_RABIT   KLCLTTALRK KHHHLL---- HLLLKLLQQE QQPPSSQEEI ILLKKERLRR RFQQTVPPPL
CAS1_PIG     KFCLVVALRK KPPPLL---- HQQQEHHQQN EEPPSRELLF FKKEERKRFF FPVVPLLLLS
CAS1_HUMAN   RLCLVVALRK KPPPLL---- YPPPERRQQN PPSSSS---- ---------- ----PIPPPL
CAS1_SHEEP   KLCLVVALRK KPPPII---- HQQQGLLPP- ------EVVL LNNEEN-RFF FVAAPFPPPE
CAS1_BOVIN   KLCLVVALRK KPPPII---- HQQQGLLQQ- ------EVVL LNNEEN-RFF FFAAPFPPPE
  6   333
CAS1_HUMAN   MRRLLLLLCL AAARPKKPLL RYPPPRRLQN PESSSE---- ---------- --PPPPLEE
CAS1_SHEEP   MKKLLLLLCL AAARPKKPII KHQQQLLSP- ------EEVN N-LLRFFVVV VVPPPPPEVV
CAS1_MOUSE   MKKLLLLLCL AAAMPRRHSS RVSSSQQTQQ QSSSSEEE-- -----IIFFK KKPPYYYLNN
CAS1_PIG     MKKLLLFFCL AAARPKKPLL RHQQQHHLQN EDSRREEELR RKFFRFFPPE EEPPLLLSQQ
CAS1_RABIT   MKKLLLLLCL AAARHKKHLL GHLLLLLTQE QESSSEQQEE ERKKLRRFFV VVTTPPPLEE
CAS1_BOVIN   MKKLLLLLCL AAARPKKPII KHQQQLLPQ- ------EEVN N-LLRFFFFV VVPPPPPEVV
[And so forth...]
```

Then we proceed with this file as if it was the original sequence alignment. In our case, lets produce distance trees with the program Protdist. We start by renaming outfile to infile, and invoking the program:

```
Protein distance algorithm, version 3.66

Settings for this run:
  P   Use JTT, PMB, PAM, Kimura, categories model?  Jones-Taylor-Thornton
matrix
  G  Gamma distribution of rates among positions?  No
  C           One category of substitution rates?  Yes
  W                    Use weights for positions?  No
  M                   Analyze multiple data sets?  No
  I                  Input sequences interleaved?  Yes
  0                Terminal type (IBM PC, ANSI)?  ANSI
  1            Print out the data at start of run  No
  2          Print indications of progress of run  Yes

Are these settings correct? (type Y or the letter for one to change)
m
How many data sets?
100
```

We have to tell the program that there are multiple datasets in the same file by changing the option “m”. Then the distance matrix will be calculated for all the 100 random samples. The outfile produced contains all these distance matrixes:

```
    6
CAS1_MOUSE     0.00000  0.80622  1.46041  1.14997  2.11928  1.98883
CAS1_RABIT     0.80622  0.00000  1.35772  0.82152  1.76401  1.49899
CAS1_PIG       1.46041  1.35772  0.00000  1.30075  0.61110  0.58505
CAS1_HUMAN     1.14997  0.82152  1.30075  0.00000  1.72499  1.56997
CAS1_SHEEP     2.11928  1.76401  0.61110  1.72499  0.00000  0.08268
CAS1_BOVIN     1.98883  1.49899  0.58505  1.56997  0.08268  0.00000
```

```
    6
CAS1_HUMAN     0.00000  2.03032  1.01938  1.27690  0.69497  1.95053
CAS1_SHEEP     2.03032  0.00000  2.15271  0.71847  1.79089  0.16285
CAS1_MOUSE     1.01938  2.15271  0.00000  1.47633  0.90248  1.87560
CAS1_PIG       1.27690  0.71847  1.47633  0.00000  1.52777  0.77397
CAS1_RABIT     0.69497  1.79089  0.90248  1.52777  0.00000  1.54938
CAS1_BOVIN     1.95053  0.16285  1.87560  0.77397  1.54938  0.00000

[And so forth]
```

These distance matrices are then used to infer the tree with the original method, *i.e.*, if the original data was analysed with F84-distance and Neighbor-joining, the bootstrapping analysis has to be performed with the same settings.

Again, in the program Neighbor the multiple datasets option (m) has to be chosen. The resulting outtree contains trees for all the 100 random datasets.

```
((CAS1_PIG:0.09668,(CAS1_SHEEP:0.11341,CAS1_BOVIN:-0.03073):0.46006):0.75811,
CAS1_HUMAN:0.41697,(CAS1_MOUSE:0.55088,CAS1_RABIT:0.25534):0.16567);
(((CAS1_SHEEP:0.14929,CAS1_BOVIN:0.01356):0.52397,CAS1_PIG:0.14082):0.82720,
CAS1_MOUSE:0.54493,(CAS1_HUMAN:0.38539,CAS1_RABIT:0.30958):0.06851);

[And so forth]
```

These trees are then combined in a consensus tree with the program Consense.

```
Consensus tree program, version 3.66

Settings for this run:
 C         Consensus type (MRe, strict, MR, Ml):  Majority rule (extended)
 O                                   Outgroup root:  No, use as outgroup
species  1
 R               Trees to be treated as Rooted:  No
 T           Terminal type (IBM PC, ANSI, none):  ANSI
 1                Print out the sets of species:  Yes
 2         Print indications of progress of run:  Yes
 3                              Print out tree:  Yes
 4               Write out trees onto tree file:  Yes

Are these settings correct? (type Y or the letter for one to change)
```

There are four consensus tree types to choose from. Strict consensus creates a tree which only includes the set of sequences, if it occurs in all the trees. The MR, MRe and M1 all produce a majority rule trees with slightly different options. The default method (MRe) will include into the new tree all the groups of sequences which are present in more than 50% of the trees, plus the most frequent others that are compatibel with these. Ml lets you to specify the percentage. Note, that the consensus tree from bootstrapping samples should always be drawn with majority rule method.

The `outfile` and `outtree` contain the information on how many times each set of species was find to be together in the random samples. If this value is under, say 70-95%, the result should be interpreted with caution. This also implies that there probably is not enough data to differentiate between topologies when the sets in them have low bootstrap values.

```
Majority-rule and strict consensus tree program, version 3.61
Species in order:
```

```
  CAS1 PIG
  CAS1 SHEEP
  CAS1 BOVIN
  CAS1 HUMAN
  CAS1 MOUSE
  CAS1 RABIT

Sets included in the consensus tree

Set (species in order)     How many times out of  100.00



...***                     100.00
.**...                     100.00
....**                     77.00



Sets NOT included in consensus tree:
Set (species in order)     How many times out of  100.00

...*.*                     16.00
...**.                      7.00

Extended majority rule consensus tree


CONSENSUS TREE:
the numbers at the forks indicate the number
of times the group consisting of the species
which are to the right of that fork occurred
among the trees, out of 100.00 trees

                +-------------CAS1 HUMAN
         +100.0-|
         |      |      +------CAS1 RABIT
         |      +-77.0-|
  +------|             +------CAS1 MOUSE
  |      |
  |      |             +------CAS1 SHEEP
  |      +-------100.0-|
  |                    +------CAS1 BOVIN
  |
  +---------------------------CAS1 PIG

  remember: this is an unrooted tree!
```

The concomitant `outtree` contains the same information. The bootstrapping values are reported in the place of branch lenghts.

```
(((CAS1_HUMAN:100.0,(CAS1_RABIT:100.0,CAS1_MOUSE:100.0):77.0):100.0,(CAS1_S
HEEP:100.0,CAS1_BOVIN:100.0):100.0):100.0,CAS1_PIG:100.0);
```

The bootstrapping procedure implemented in PHYLIP does not perform the analysis on the tree inferred from original dataset. Instead, Felsenstein has argued that one reasonable trees would be the one recovered from the random samples as described above. However, the original inferred tree and the tree produced by bootstrapping are usually pretty similar.

Drawing the tree

We can draw a picture of an unrooted tree from the information contained in the file "plottree" with the program Drawgram.

```
Drawgram: can't find input tree file "intree"
Please enter a new file name> plottree
DRAWGRAM from PHYLIP version 3.6
Reading tree ...
Tree has been read.
Loading the font ....
Drawgram: can't find font file "fontfile"
Please enter a new file name> font1
Font loaded.

Rooted tree plotting program version 3.66

Here are the settings:
 0  Screen type (IBM PC, ANSI):  ANSI
 P       Final plotting device:  Postscript printer
 V           Previewing device:  X Windows display
 H                  Tree grows:  Horizontally
 S                  Tree style:  Phenogram
 B          Use branch lengths:  Yes
 L             Angle of labels:  90.0
 R      Scale of branch length:  Automatically rescaled
 D       Depth/Breadth of tree:  0.53
 T      Stem-length/tree-depth:  0.05
 C    Character ht / tip space:  0.3333
 A             Ancestral nodes:  Weighted
 F                        Font:  Times-Roman
 M          Horizontal margins:  1.65 cm
 M            Vertical margins:  2.16 cm
 #              Pages per tree:  one page per tree

 Y to accept these or type the letter for one to change
```

First, we want to be able to view the file easily in Windows, and we change the Final plotting device by typing "p" and Return.

A new menu opens:

```
Which plotter or printer will the tree be drawn on?
(many other brands or models are compatible with these)

   type:       to choose one compatible with:

        L          Postscript printer file format
        M          PICT format (for drawing programs)
        J          HP Laserjet PCL file format
        W          MS-Windows Bitmap
        F          FIG 2.0 drawing program format
        A          Idraw drawing program format
        Z          VRML Virtual Reality Markup Language file
        P          PCX file format (for drawing programs)
        K          TeKtronix 4010 graphics terminal
        X          X Bitmap format
        V          POVRAY 3D rendering program file
        R          Rayshade 3D rendering program file
        H          Hewlett-Packard pen plotter (HPGL file format)
        D          DEC ReGIS graphics (VT240 terminal)
        E          Epson MX-80 dot-matrix printer
        C          Prowriter/Imagewriter dot-matrix printer
        T          Toshiba 24-pin dot-matrix printer
        O          Okidata dot-matrix printer
        B          Houston Instruments plotter
        U          other: one you have inserted code for
 Choose one:
```

From this we will choose MS-Windows Bitmap by typing "w" and Return. The program then asks the desired resolution of the picture (on the next page). After specifying that, you'll be dropped back to the main menu.

```
Please select the MS-Windows bitmap file resolution
X resolution?
640
Y resolution?
400
```

After accepting the settings by typing in "y" and Return, a new window opens. In this window you can preview the tree, and if it looks good, plot it into file by selecting from File menu option plot. The tree is plotted in the `plotfile`. The options of the program Drawtree are similar, but the program produces a rooted tree.

Advanced topics

Here some more advanced analysis options are considered. These options are less often used by researchers, especially when they do not know about these possibilities.

User trees
User trees have multiple purposes. They can be used for transition/transversion ratio approximation, likelihood comparisons (statistical comparisons of the trees), etc. These purposes will be covered in the succeeding sections.

User trees should be in a plain text file named `intree`. Some programs do not read the branch lengths or the number of the trees, so this needs a little experimentation. However, the basic form of the `intree` file is:

```
(Frog,(Chicken,(Opossum,(Human,Rabbit))));
```

The easiest way to produce user-trees is to perform an analysis with some of the programs, and then modify the file outtree with the program Retree or by some text editor.

Program Retree
Retree reads in a treefile, which is in Newick format. For example, a tree produced with Dnaml can be directly imported into Retree. In this example we modify the tree:

```
 (Human:0.20090,((Frog:0.90555,Chicken:0.58309):0.19152,
Opossum:0.69201):0.18632,Rabbit:0.09743);
```

when the program starts, you get a menu:

```
Tree Rearrangement, version 3.66

Settings for this run:
  U           Initial tree (arbitrary, user, specify)?  User tree from tree
file
  N   Format to write out trees (PHYLIP, Nexus, XML)?  PHYLIP
  0                     Graphics type (IBM PC, ANSI)?  ANSI
  W       Width of terminal screen, of plotting area?  80, 80
  L                         Number of lines on screen?  24

Are these settings correct? (type Y or the letter for one to change)
```

After you have confirmed these setting, a new view opens:

```
                                  ,--------------------------1:Frog
            ,---------------------8
  ,---------7                     `-----------------------------2:Chicken
  !         !
  !         !
  !         `---------------3:Opossum
  !
--6------4:Human
  !
  `-----------------------------------5:Rabbit

NEXT? (Options: R . = U W O M T F D B N H J K L C + ? X Q) (? for Help)
```

The options of the program can be printed on the screen by typing "?" and Return:

```
 . Redisplay the same tree again
 = Redisplay the same tree without/with lengths
 U Undo the most recent change in the tree
 W Write tree to a file
 + Read next tree from file (may blow up if none is there)
 R Rearrange a tree by moving a node or group
 O select an Outgroup for the tree
 M Midpoint root the tree
 T Transpose immediate branches at a node
 F Flip (rotate) subtree at a node
 D Delete or restore nodes
 B Change or specify the length of a branch
 N Change or specify the name(s) of tip(s)
 H Move viewing window to the left
 J Move viewing window downward
 K Move viewing window upward
 L Move viewing window to the right
 C show only one Clade (subtree) (might be useful if tree is too big)
 ? Help (this screen)
 Q (Quit) Exit from program
 X Exit from program

 TO CONTINUE, PRESS ON THE Return OR Enter KEY
```

Now you can reroot the tree, swap the branches, etc. In this case, we want to remove branch lengths from all the branches of the tree. This can be done by invoking the option "b":

```
NEXT? (Options: R . = U W O M T F D B N H J K L C + ? X Q)(? for Help) b
Specify length of which branch (0 = all branches)? 0

 (this operation cannot be undone)

   enter U to leave the lengths unchanged
OR enter R to remove the lengths from all branches:
r

         ,-----1:Frog
  ,-----7
  !      !  ,--2:Human
  !      `--8
  !         `--3:Opossum
  !
--6------------4:Chicken
  !
  `------------5:Rabbit
```

Next we want to save the tree and quit the program:

```
NEXT? (Options: R . U W O T F D B N H J K L C + ? X Q) (? for Help) x
Do you want to write out the tree to a file? (Y or N) y
Retree: the file "outtree" that you wanted to
     use as output tree file already exists.
     Do you want to Replace it, Append to it,
     write to a new File, or Quit?
     (please type R, A, F, or Q)
r

Enter R if the tree is to be rooted
OR enter U if the tree is to be unrooted: u

Tree written to file “outtree”
```

The tree written by program Retree in file outtree is:

```
(((Frog,Chicken),Opossum),Human,Rabbit);
```

Estimating the transition/transversion ratio

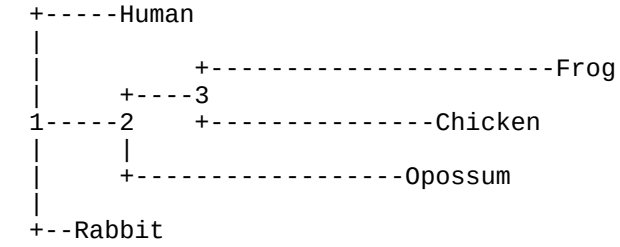
The estimated ratio depends on the specified outgroup. So, if you have information about the outgroup you are planning to use in the forthcoming analysis, you should use it in this estimation process too.

One way to estimate the transition/transversion ratio is to run maximum likelihood program Dnaml multiple times with different transversion/transition ratios, and try to find the value, which maximizes the likelihood.

The maximum likelihood method works this way (* = highest likelihood value corresponding to the maximum likelihood estimate of s/t-ratio):

s/t-ratio	likelihood	s/t-ratio	likelihood
1.25	-9644.49078	1.00	-9643.20009
1.15	-9642.29536	0.95	-9644.9407
1.10	-9641.96561*	0.90	-9647.57138
1.05	-9642.23988	0.85	-9651.21944

You should use the inferred transition/transversion ratio to reanalyze your dataset, if you have previously used the default settings. In the example, the tree is similar to one inferred using the default settings:



But the lengths of the branches are different (compare to the tree inferred using Dnaml in sections above):

```
Ln Likelihood = -9641.96561

 Between        And            Length      Approx. Confidence Limits
 -------        ---            ------      ------- ---------- ------
    1          Rabbit           0.10097     (  0.07432,     0.12755) **
    1          Human            0.18761     (  0.15605,     0.21917) **
    1             2             0.18557     (  0.13596,     0.23522) **
    2             3             0.13641     (  0.07800,     0.19475) **
    3          Frog             0.75496     (  0.66344,     0.84638) **
    3          Chicken          0.51613     (  0.44578,     0.58657) **
    2          Opossum          0.57699     (  0.50869,     0.64529) **

     *  = significantly positive, P < 0.05
     ** = significantly positive, P < 0.01
```

The procedure outlined above can be quite slow, and a good compromise for estimating the transition/transversion ratio is to acquire a good tree and then estimate the parameters using the tree. In practice, this can be done using a user tree in program Dnaml and letting the program to re-estimate the branch lengths. Even faster would be to use the user tree but not allow Dnaml to re-estimate the branch lengths. Run the analysis with user tree multiple times and supply different estimates of transition/transversion ratio. Then, as outlined above, pick the results giving the highest likelihood value.

Estimating base frequencies

Above, a maximum likelihood method for estimating transition/transversion ratio is presented. A similar method can be used for inferring the maximum likelihood estimators of base frequencies.

Empirical frequencies are:

```
   A       0.25650
   C       0.21951
   G       0.22091
  T(U)     0.30309
```

We can start the estimation by option "f", which then prompts for base frequencies. The frequency values should be separated by blanks:

```
Base frequencies for A, C, G, T/U (use blanks to separate)?
0.25 0.22 0.22 0.31
```

Note the likelihood of the old (-9695.14457) and new (-9696.03008) analysis. In this example, the likelihood got lower after modifying the base frequencies. This indicates that the empirical frequencies were better. This procedure can be repeated a number of times in order to get the best estimate, but it requires multiple runs with different base frequencies and might get tedious.

Testing molecular clock

The molecular clock assumption can be tested by the two maximum likelihood programs, Dnaml and Dnamlk. The test, which compares two likelihoods is called likelihood ratio test.

First the analysis is run by using the program Dnamlk, which produces an unrooted tree with molecular clock assumption:

```
(Frog:0.77626,(Chicken:0.63785,(Opossum:0.49812,(Human:0.15025,
Rabbit:0.15025):0.34787):0.13973):0.13842);
```

Note the log-likelihood of the tree -9714.14764.

The program Dnaml can read in this treefile, which has first to be named `intree`. Remember to run the analysis with the user tree option ("u") turned on. You should also run the analysis without using the lenghts on the user tree.

Note the log-likelihood of tree from the Dnaml analysis -9695.14457

Now calculate the difference between the log-likelihoods: 2 * (-9714.14764 + 9695.14457) = 38.006. This difference is then compared to the chi-square distribution with df (degrees of freedom) equal to (number of sequences -2). The table consists of dfs on the left side, and p-values on the upper side. The values in the crossing of the p-value and df is called the critical value of the distribution. P-value is the risk that we conclude by chance that sequences did not evolve according to a clock, if they actually did evolve with a clock. Often the p-value is set to be 0.05.

If the difference is larger than the critical value mentioned in the table below, the sequences did not evolve according to a molecular clock.

In our case, the difference is 38.006. That is larger than the critical value with df=3 and p-value=0.05, which is 7.815. Thus, we conclude that we can reject the clock hypothesis, but by chance we might do that in 1 / 20 cases if the same analysis were repeated (p-value of 0.05) when there actually is a clock.

	Probability of exceeding the critical value				
df	0.10	0.05	0.025	0.01	0.001
1	2.706	3.841	5.024	6.635	10.828
2	4.605	5.991	7.378	9.210	13.816
3	6.251	**7.815**	9.348	11.345	16.266
4	7.779	9.488	11.143	13.277	18.467
5	9.236	11.070	12.833	15.086	20.515
6	10.645	12.592	14.449	16.812	22.458
7	12.017	14.067	16.013	18.475	24.322
8	13.362	15.507	17.535	20.090	26.125
9	14.684	16.919	19.023	21.666	27.877
10	15.987	18.307	20.483	23.209	29.588

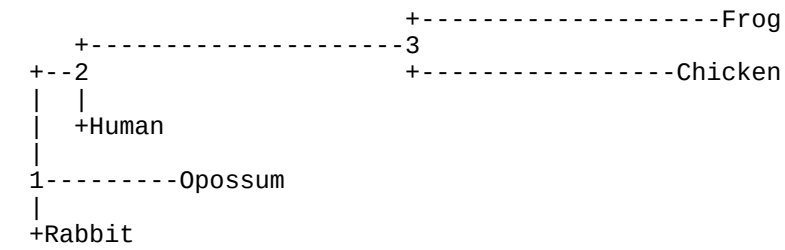
Inferring ancient states of sequence sites

Why are we interested in reconstructing ancestral character states? It can provide important information about adaptive radiations and key innovations for these radiations. One interesting approach has been to use the inferred ancient sequences to produce an inferred ancient protein. The biochemical properties of this protein could be studied and compared to the modern proteins.

The ancestral states of sequence sites can be inferred either by parsimony or by maximum likelihood. It is often thought, that the maximum likelihood method is more accurate than the parsimony method for inferring ancient sequences.

Ancestral states can be inferred in programs Dnapars, Protpars, Dnaml, Dnamlk, Proml, and Promlk by turning on the option "5", print sequences at all nodes of tree and reconstruct hypothetical sequences, respectively.

Tree is identical for both methods



[with the maximum maximum likelihood method]

```
Probable sequences at interior nodes:

  node                    Reconstructed sequence (caps if > 0.95)

    1        GTTCAGGACT TGGGCATAAA AGGCAGAGCA GGGCcAGCTG CTGCTTACAC
    2        agcCAGGgCT tGGGCATAAA AGtCAGgGCA GaGCcAtCTa tTGCTTACAt
    3        rrcrrcyrcy rscacrrgyr ygycarcrcr rrgyyakcrr yygyycarry
 Frog        CTCAACTTTG GCCATGGGTT TGACAGCACA TGATCGTCAG CTGATCAACA
 Chicken     GACGGCCGCT rrCACCAGCG TGCTATCCCC ACGGGAGCAA GAGCCCAGAC
 Human       AGCCAGGGCT rGGGCATAAA AGTCAGGGCA GAGCCATCTA TTGCTTACAT
 Opossum     GTTTGGGGGC CAGGGATAAA AGGCAGAGCT AGATTAGTTT CAGCATCATA
 Rabbit      GTTCAGGACT TGGGCATAAA AGGCAGAGCA GGGCrAGCTG CTGCTTACAC
```

Maximum likelihood method writes the sequence sites with over 95% probability with upper case, sites with 50-95% probability with lower case, and those with less than 50% probability with an ambiguity code.

[with the parsimony method]

```
From    To     Any Steps?    State at upper node
          1                  GTTCAGGGCT ?GGGCATAAA AGGCAGAGCA RGGY?AGCTD
   1      2          yes     GTCCAGGGCT -GGGCATAAA AGNCAGVGCA RGGYCAKCTR
   2      3          yes     GTCVACBGCT -?CACVDGHD TGNCAGCVCA DGGBCAKCAR
   3   Frog          yes     CTCAACTTTG GCCATGGGTT TGACAGCACA TGATCGTCAG
   3   Chicken       yes     GACGGCCGCT --CACCAGCG TGCTATCCCC ACGGGAGCAA
   2   Human         yes     AGCCAGGGCT -GGGCATAAA AGTCAGGGCA GAGCCATCTA
   1   Opossum       yes     GTTTGGGGGC CAGGGATAAA AGGCAGAGCT AGATTAGTTT
   1   Rabbit        yes     GTTCAGGACT TGGGCATAAA AGGCAGAGCA GGGC-AGCTG


          1                  CTGCTTAMAM
   1      2          no      CTGCTTAMAM
   2      3          yes     CTGCTCAVAM
   3   Frog          yes     CTGATCAACA
   3   Chicken       yes     GAGCCCAGAC
   2   Human         yes     TTGCTTACAT
   1   Opossum       yes     CAGCATCATA
   1   Rabbit       maybe    CTGCTTACAC
```

If the parsimony inferred state is "?" or an ambiguity code, there are multiple equally parsimonious states; the user has to work these out by hand. In addition, "?" means that a deletion might or might not have happened. “N” indicates a nondeleted base that is ambiquous.

Statistical tests of trees

Statistical tests can be performed for both parsimony and maximum likelihood trees. The tests are performed by putting multiple trees in the `intree` file. Actually, the tests are automatically performed if there are multiple trees in the `intree` file:

```
(((Frog,Chicken),Human),Opossum,Rabbit);
(((Frog,Human),Chicken),Opossum,Rabbit);
(((Frog,Opossum),Human),Chicken,Rabbit);
(((Frog,Chicken),Rabbit),Opossum,Human);
(((Frog,Rabbit),Human),Opossum,Chicken);
```

To perform the test, load the sequence data, and invoke option U (No, use user trees in input file) in a parsimony or maximum likelihood program.

Parsimony programs, for example Dnapars, perform Templeton's test if there are two trees to compare, but uses Shimodaira-Hasegawa's method for more than two trees. Shimodaira-Hasegawa's method uses a resampling method to correct the test results for multiple comparisons. This resampling asks for a random number when performed.

This test finds the best trees among the competing hypothesis. Consult the column "Significantly worse?". If it states "No" the tree getting the best score is not significantly different from the tree it was compared to. Thus, it is not possible to pick the best of trees 3-5 on basis of this test:

```
Tree    Steps   Diff Steps  P-value   Significantly worse?

  1    3134.0      38.0     0.001            Yes
  2    3194.0      98.0     0.009            Yes
  3    3096.0  <------ best
  4    3107.0      11.0     0.141            No
  5    3124.0      28.0     0.221            No
```

Maximum likelihood programs perform a Kishino-Hasegawa test for two trees, and Shimodaira-Hasegawa's test for more than two trees.

This test finds the best trees among the competing hypothesis. Consult the column "Significantly worse?". If it states "No" the tree getting the best score is not significantly different from the tree it was compared to. Thus, it is not possible to pick the best of trees 1, 4, and 5 on basis of this test:

```
Tree    Ln L      Diff Ln L     P-value   Significantly worse?

  1 -9711.73593    -0.49089      0.139           No
  2 -9738.53093   -27.28588      0.003           Yes
  3 -9730.79910   -19.55405      0.001           Yes
  4 -9711.24505  <------ best
  5 -9725.82664   -14.58159      0.449           No
```

LogDet-distance

LogDet-distance has been developed to account for the base frequency differences between lineages. However, the LogDet-distance does not give a reliable tree, when there are large rate differences between sites in the sequence.

LogDet-distances would be usable when the base frequencies in different lineages are not constant. In such cases, LogDet distances often outperform maximum likelihood method.

How to test for base frequency heterogeneity? Dnaml writes a table of empirical base frequencies in the current dataset. The downside is that these frequencies can not be computed for only two sequences. Here is an example of base frequencies calculated for three different sets of three taxa:

```
  A       0.28378
  C       0.19595
  G       0.31081
 T(U)     0.20946

  A       0.27027
  C       0.23649
  G       0.27703
 T(U)     0.21622

  A       0.25850
  C       0.29252
  G       0.26531
 T(U)     0.18367
```

There are some differences between lineages, but not too large (about 5%), and the usual distance method should do fine. However, this estimation approach fast becomes complicated when the number of sequences goes up. One limitation of the LogDet distance is that it may sometimes be infinite, if there are too many changes between certain pairs of nucleotides. This can be particularly noticeable with distances computed from bootstrapped sequences.

Computing topological distances between trees

Topological distances are calculated with the program treedist. The symmetric distance introduced here does not consider the branch lengths, only the tree topologies. The symmetric distance between two trees is a count of partitions present in the other but not in the another tree.

To calculate the symmetric distance, you need a standard `intree` as an input:

```
(((Frog,Chicken),Human),Opossum,Rabbit);
(((Frog,Human),Chicken),Opossum,Rabbit);
(((Frog,Chicken),Human),Opossum,Rabbit);
(((Frog,Opossum),Human),Chicken,Rabbit);
(((Frog,Chicken),Human),Opossum,Rabbit);
(((Frog,Chicken),Rabbit),Opossum,Human);
(((Frog,Chicken),Human),Opossum,Rabbit);
(((Frog,Rabbit),Human),Opossum,Chicken);
```

The treedist writes a menu:

```
Tree distance program, version 3.66

Settings for this run:
 D                          Distance Type:  Branch Score Distance
 O                          Outgroup root:  No, use as outgroup species  1
 R         Trees to be treated as Rooted:  No
 T    Terminal type (IBM PC, ANSI, none):  ANSI
 1  Print indications of progress of run:  Yes
 2                 Tree distance submenu:  Distance between adjacent pairs

Are these settings correct? (type Y or the letter for one to change)
```

Invoking the option "2" let's you to specify which way the trees are compared. Choosing Option P calculates the pairwise topological distances for all the trees in the file `intree`.

```
Tree Pairing Submenu:
 A     Distances between adjacent pairs in tree file.
 P     Distances between all possible pairs in tree file.
 C     Distances between corresponding pairs in one tree file and other.
 L     Distances between all pairs in one tree file and another.

 Choose one: (A,P,C,L)
p
```

Then the program ask what kind of a matrix to return:

```
Distances output options:
 F     Full matrix.
 V     One pair per line, verbose.
 S     One pair per line, sparse.

 Choose one: (F,V,S)
f
```

As a last step, you need to specify the distance type (Option D):

```
Tree distance program, version 3.66

Settings for this run:
 D                          Distance Type:  Symmetric Difference
 O                          Outgroup root:  No, use as outgroup species  1
 R         Trees to be treated as Rooted:  No
 T    Terminal type (IBM PC, ANSI, none):  ANSI
 1  Print indications of progress of run:  Yes
 2                 Tree distance submenu:  Distances between all possible
                                             pairs in tree file.

Are these settings correct? (type Y or the letter for one to change)
```

The distances are calculated for all pairs of trees. Results are by default saved in the file `outfile`:

```
Tree distance program, version 3.66

Symmetric differences between all pairs of trees in tree file:

          1     2     3     4     5     6     7     8
      \-----------------------------------------------
    1 |   0     2     0     4     0     2     0     4
    2 |   2     0     2     4     2     4     2     4
    3 |   0     2     0     4     0     2     0     4
    4 |   4     4     4     0     4     4     4     4
    5 |   0     2     0     4     0     2     0     4
    6 |   2     4     2     4     2     0     2     4
    7 |   0     2     0     4     0     2     0     4
    8 |   4     4     4     4     4     4     4     0
```

Program treedist is handy when there are multiple trees, for example, equally parsimonious trees, and pairwise comparisons need to be made fast. As is obvious from the example above, the pairwise distance of 2 means one swapping of two terminal branches (two sequences).

Weighting

Weights can be used to analyze different subsets of characters (by weighting the rest as zero). For example, it might be of interest to compare the trees inferred from the first, second and the third codon positions. This can be done using the weights. The weights are saved in a file named "weights". The file should contain a text string of weights, one weight given to every sequence alignment position. For example, the weight-pattern given below will use only the third codon position for inferring the tree:

001001001001001001001001001001001001001001001001001

The weights can continue on several lines, and blanks between the lines will be ignored.

Weights are also handy, if you want to analyze different parts of the sequences, for example, only conserved areas of the protein. You don't need to edit the original data files if you just create new weights.

You can also check for the uniformity of the substitution rates of different codon positions by using weights. Create three weight files, where you specify to include only the first, second or the third codon positions in the analysis of a 9-site data set there might be:

First positions:	100100100
Second positions:	010010010
Third positions:	001001001

These weight specifications must be exactly as long as your sequence is. Otherwise you will get into trouble. Then run the program Dnapars with the same material using the option "w" once with every weight-specifications. Use the topology inferred without weights as user tree (invoke Option U), and note the number of changes:

All positions: 49 changes
First positions: 20 changes
Second positions: 12 changes
Third positions: 17 changes

It now becomes visible that there are different numbers of changes in different codon positions. It seems that the second position has fewer changes than other positions. This is a reasonable result, because substitutions in the second codon position more often lead to an amino acid change than substitutions in the first position.

In the example, the first and third codon position have nearly the same number of changes. It also reasonable to expect that the third codon position will have more changes than the first, but this was not supported by the data in our example.

Why should we check for unequal rate of substitution in different positions? One reason is to check whether there are some evolutionary constraints for substitutions in certain codon positions. Another reason is to check for saturation of the substitutions in different codon positions. If there are large differences between the number of changes, especially, if the number of substitutions in the third codon positions is high, saturation of that position is a likely explanation. In the case of saturation, DNA sequences that include third position may give erroneous results, and it could be better to use protein sequences or the DNA sequence consisting of only the first and the second codon positions.

Another method for testing for the sequence site rate heterogenity is to calculate pairwise distances using the different codon sites (program Dnadist):

```
First positions:
    5
Rabbit      0.0000  0.1578  1.4088  0.5954  0.8499
Human       0.1578  0.0000  0.8327  0.4690  4.2225
Opossum     1.4088  0.8327  0.0000  0.5029  5.9886
Chicken     0.5954  0.4690  0.5029  0.0000  3.3429
Frog        0.8499  4.2225  5.9886  3.3429  0.0000

Secon positions:
    5
Rabbit      0.0000  0.0754  0.2923  0.6328  0.8892
Human       0.0754  0.0000  0.1695  0.4178  0.6464
Opossum     0.2923  0.1695  0.0000  0.2996  0.7112
Chicken     0.6328  0.4178  0.2996  0.0000  0.6150
Frog        0.8892  0.6464  0.7112  0.6150  0.0000

Third positions:
    5
Rabbit      0.0000  0.4062  0.5623  0.8167  1.2231
Human       0.4062  0.0000  0.2666  0.4173  1.0609
Opossum     0.5623  0.2666  0.0000  0.1644  0.8987
Chicken     0.8167  0.4173  0.1644  0.0000  1.3639
Frog        1.2231  1.0609  0.8987  1.3639  0.0000
```

After calculating the pairwise distances, we can check whether the number of substitutions seems to differ between codon position, which it in our example most certainly does.

Note that, if the species are not closely related (for example, human and frog) the sequences might have more amino acid changes than silent substitutions.

Dnaml, HMM, gamma distribution and rate heterogeneity

Program Dnaml (and Dnamlk, Proml and Promlk) implements two different layers of base substitution rate heterogenity:

```
Settings for this run:
  U                 Search for best tree?  Yes
  T        Transition/transversion ratio:  2.0000
  F       Use empirical base frequencies?  Yes
  C                One category of sites?  Yes
  R            Rate variation among sites?  constant rate
  W                       Sites weighted?  No
  S        Speedier but rougher analysis?  Yes
  G                Global rearrangements?  No
  J   Randomize input order of sequences?  No. Use input order
  O                       Outgroup root?  No, use as outgroup species  1
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  IBM PC
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                        Print out tree  Yes
  4       Write out trees onto tree file?  Yes
  5   Reconstruct hypothetical sequences?  No
```

The first layer models the rate heterogeneity by a hidden Markov model (HMM). This is done automatically.

“HMM allows us to specify with option "c" to the program that there will be a number of different possible evolutionary rates, what the prior probabilities of occurrence of each is, and what the average length of a patch of sites all having the same rate. The program then computes the likelihood by summing it over all possible assignments of rates to sites, weighting each by its prior probability of occurrence. There is also a possibility to set that the rates in adjacent sites are correlated with each other with option "a".”

“Another layer of rate variation is also available. The user can assign categories of rates to each site (for example, we might want first, second, and third codon positions in a protein coding sequence to be three different categories. This is done with the categories input file and the C option. We then specify (using the menu) the relative rates of evolution of sites in the different categories. For example, we might specify that first, second, and third positions evolve at relative rates of 1.0, 0.8, and 2.7.” (Felsenstein, PHYLIP documentation)

There is also a possibility to change the gamma-distribution shape parameter with option "r". At the moment the gamma-parameter can not be directly estimated, but it can be inferred by an iteration method described above in the section Estimating the transition/transversion ratio. After the iteration, the best tree can be compared to the tree without rate heterogenity by the likelihood ratio test with df=1 (see, Testing molecular clock) or by Kishino-Hasegawa test using user trees (see, Statistical tests of trees).

“If both user-assigned rate categories (with categories file) and regional rate variation (the Hidden Markov Model rates) are allowed, the program assumes that the actual rate at a site is the product of the user-assigned category rate and the Hidden Markov Model regional rate. (This may not always make perfect biological sense: it would be more natural to assume

some upper bound to the rate, as we have discussed in the Felsenstein and Churchill paper). Nevertheless you may want to use both types of rate variation." (Felsenstein, PHYLIP documentation)

Multiple outgroups

"It's not a feature but is not too hard to do in many of the programs. In parsimony programs like mix, for which the "w" (weights) and "a" (Ancestral states) options are available, and weights can be larger than 1, all you need to do is:"

(a) In mix, make up an extra character with states 0 for all the outgroups and 1 for all the ingroups. If using Dnapars the ingroup can have (say) "G" and the outgroup "A".
(b) Assign this character an enormous weight (such as Z for 35) using the "w" option, all other characters getting weight 1, or whatever weight they had before.
(c) If it is available, Use the "a" (Ancestral states) option to designate that for that new character the state found in the outgroup is the ancestral state.
(d) In mix do not use the "o" (Outgroup) option.
(e) After the tree is found, the designated ingroup should have been held together by the fake character. The tree will be rooted somewhere in the outgroup (the program may or may not have a preference for one place in the outgroup over another). Make sure that you subtract from the total number of steps on the tree all steps in the new character.

"In programs like Dnapars, you cannot use this method as weights of sites cannot be greater than 1. But you do an analogous trick, by adding a largish number of extra sites to the data, with one nucleotide state ("A") for the ingroup and another ("G") for the outgroup. You will then have to use Retree to manually reroot the tree in the desired place." (Felsenstein, PHYLIP documentation)

Multiple outgroups as described above cannot be used with maximum likelihood programs.

Error messages

Here are some of the most commonly encountered error messages, and what to do to correct them. The first three are not actually error messages at all, but an essential part of the normal function of the programs.

1. Can't find infile

```
Dnapars.exe: can't find input file "infile"
Please enter a new file name>
```

There is no file named `infile` in the folder you are running the program from, and the program asks for the name of the input file. This is easily corrected: Type in the name of the input file (sequence alignment file).

2. Outfile already exists

```
Dnapars.exe: the file "outfile" that you wanted to
     use as output file already exists.
     Do you want to Replace it, Append to it,
     write to a new File, or Quit?
     (please type R, A, F, or Q)
```

There is already an `outfile` in the same folder you are running the program from. You have to decide whether to replace (overwrite) it, append it, or quit. You can also specify a new outfile name.

3. Outtree already exists

```
Dnapars.exe: the file "outtree" that you wanted to
     use as output tree file already exists.
     Do you want to Replace it, Append to it,
     write to a new File, or Quit?
     (please type R, A, F, or Q)
```

There is already an `outtree` in the same folder you are running the program from. You have to decide whether to replace (overwrite) it, append it, or quit. You can also specify a new outtree name.

4. Wrong data type

```
ERROR: a function asked for an inappropriate amount of memory:  -4 bytes
    This can mean one of two things:
    1.  The input file is incorrect (perhaps it was not saved as Text),
    2.  There is a bug in the program.
    Please check your input file carefully.
    If it seems to be a bug, please mail joe@gs.washington.edu
    with the name of the program, your computer system type,
    a full description of the problem, and with the input data file.
    (which should be in the body of the message, not as an Attachment).
Hit Enter or Return to close program.
  You may have to hit Enter or Return twice.
```

There is probably an `infile` in the same folder you are running the program from. The problem is that this infile is in a format that the current program can't use. For example, you might have renamed a distance matrix as infile when creating Neighbor joining trees. You might then be using Dnapars program, which does not know how to read a distance matrix, because it expects to find sequences in the file. Rename or remove the infile, and the error should disappear.

5. Wrong program

```
ERROR: bad base: F at site     1 of species   1
Hit Enter or Return to close program.
  You may have to hit Enter or Return twice.
```

Or:

```
The infile is of the wrong type
Hit Enter or Return to close program.
  You may have to hit Enter or Return twice.
```

Or:

```
Unexpected End of File
Hit Enter or Return to close program.
  You may have to hit Enter or Return twice.
```

You are probably trying to use the wrong kind of data in the current program. This error message is related to the sequence type: you have used amino acid sequences as input in the DNA sequence analysis program. Use the correct input file, and the error should disappears.

Scripting

Scripting can be used for automating some analyses, when needed. It is especially attractive in UNIX / Linux system where jobs can be submitted to a queue. It is also a good idea in Windows / DOS environment, for example, if several analyses need to be run over the weekend: script can do the analysis during weekend and organize the results so that they are easily checked by human eye on Monday morning. Scripting means that a file, which contains a list directives is created, and instead of running the individual programs, the script is executed.

Scripting in UNIX, Linux and Mac OS X

Scripting in UNIX is much simpler than in DOS. Let's create an example script which runs Dnaml analysis for the dataset alveolata.phy. First, we need to find out, which are the commands and options we need for running Dnaml. We can do this by first doing a test run of the program. Let's also start the work in an empty folder, where outfiles or outtrees are not present from previous analyses, but which contains the sequence alignment file and the appropriate PHYLIP program (here Dnaml). When running, Dnaml first asks for a filename, and then the menu appears. We want to use taxon 8 as an outgroup in this example, so we invoke option O, and give it a number 8. After that, we want to run the analysis, which starts in Dnaml by typing Y. Now our script looks like following:

```
#!bin/csh/
dnaml <<EOF
alveolata.phy
o
8
y
EOF
```

Save the script in a file called batch (you can modify this). Give the batch file execute permissions (type `chmod u+x batch`), and submit it for running by typing, *e.g.*, `source batch` in the command prompt.

Scripting in DOS

Let's next do the same analysis using scripts in DOS. In DOS we need two files, the batch file, which starts the run and another file, which contains all the options to be used as input to Dnaml. In DOS batch files have an identifier `.bat`, and make sure you save the file with this extension. Otherwise DOS will not run the script at all!

So, first we create the file batch.bat in the empty folder containing only the sequence alignment and the program Dnaml. The file contains just one line, which tells the computer to run Dnaml from the same folder, and that file input.txt contains the options for the Dnaml run. Batch.bat looks like following:

```
dnaml < input.txt
```

If you would like to save the text Dnaml normally writes to the screen and have it appear in a file called screenout.txt, use the following line instead:

```
dnaml < input.txt > screenout.txt
```

Now the file input.txt contains the same sequence of options as the batch file created for UNIX analyses, and it looks as follows:

```
alveolata.phy
o
8
y
```

You can start the Dnaml -script in DOS by double clicking on its icon.

More advanced scripts

I like using scripts in DOS for doing multiple analyses, because it is easier to modify the batch-file than to run several programs separately. The following DOS script runs Dnaml for the dataset (sequence alignment) alveolata.phy, calculates topological distances between all the best trees (see Computing topological distances between trees) and finally performs the Shimodaira-Hasegawas's test (see above the section Statistical tests of trees).

The file batch.bat looks like this:

```
dnaml < input1.txt > screenout1.txt
copy outtree parsimony-tree-outtree.txt
rename outtree intree
rename outfile parsimony-tree-outfile.txt
treedist <input2.txt > screenout2.txt
rename outfile treedists-outfile.txt
dnaml < input3.txt > screenout3.txt
rename outfile SH-outfile.txt
```

The settings for individual programs are in the files input1.txt, input2.txt and input3.txt.

Input1.txt:
```
alveolata.phy
O
8
Y
```

Input2.txt:
```
2
P
F
Y
```

Input3.txt:
```
alveolata.phy
U
O
8
11
Y
```

After creating these four files, start the run by double-clicking on the batch.bat. The results appear in four separate files: parsimony-tree-outtree.txt and parsimony-tree-outfile.txt (trees from initial parsimony analysis), treedists-outfile.txt (topological distances), and SH-outfile.txt (Shimodaira-Hasegawas's test).

Recommendations

These recommendations cover some aspects of the actual phylogenetic data analysis that were not discussed in the examples above. You should adapt the data analysis recommendations here, because they highlight some of the most commonly encountered problems.

Some pragmatic warnings

1. Always use several outgroups, if possible. This way you can check that all outgroups really are outgroups.
2. Each method has different inherent weaknesses, and it might be a good idea to try several methods, because they have strengths in different areas. Try parsimony, maximum likelihood, and minimum evolution with LogDet-distances, and compare the results. If all the methods produce more or less the same tree, then your data apparently doesn't have any major pitfalls. Hint: you can compare different trees visually, but also using program treedist.

   "Maximum parsimony can be mislead if there is too much heterogeneity in substitution rates among lineages (the classic "long edges attract" problem) in the underlying true phylogeny. Minimum evolution using LogDet distances can be mislead if there is too much site-to-site rate heterogeneity, or if some of the pairwise distances are undefined (use the "showdist" command to check). Maximum likelihood under the HKY-gamma model can be mislead if parameters that are assumed to be constant

across the phylogeny (such as the tratio or base frequencies) actually vary among lineages in the true phylogeny." (David Swofford, PAUP FAQ, http://paup.csit.fsu.edu/paupfaq/faq.html)

"For example, if there is strong rate heterogeneity in your data (let's say the shape parameter is estimated to be 0.01), then the LogDet and parsimony trees fall under a certain degree of suspicion compared to the likelihood tree, which should be relatively immune to this pitfall since the model used allows for rate heterogeneity. If the parsimony tree differs from the LogDet and likelihood tree, look for evidence of long branch (edge) attraction in the parsimony tree. If the LogDet tree differs from the parsimony and likelihood trees, see if the base frequencies vary considerably between tip taxa (a useful tool for this purpose is the basefreq command)." (David Swofford, PAUP FAQ, http://paup.csit.fsu.edu/paupfaq/faq.html)

3. Always bootstrap or jackknife your dataset.
4. Try excluding one taxon from the analysis. If the result changes dramatically, then the excluded taxon causes systematic errors. In ideal case, the trees would be identical before and after exclusion.
5. If you're analyzing several genes, different genes might or might not produce the same results. If the results are discordant, then check your data for non-orthologous genes (if you're interested in species phylogenies), because they might cause trouble. Other sources of systematic errors, like lateral or horizontal gene transfer may confound the analysis, also.

PHYLIP programs

PHYLIP programs classified according to the material and method

	Maximum likelihood	Parsimony	Distance matrix	Lake's invariants	Reliability
DNA	Dnaml*** Dnamlk****	Dnapars Dnapenny Dnamove*****	Dnadist* Fitch*** Kitsch**** Neighbor**	Dnainvar	Dnacomp Seqboot
Protein	Proml Promlk	Protpars	Protdist* Fitch*** Kitsch**** Neighbor**		Seqboot
Restriction sites			Restdist* Fitch*** Kitsch**** Neighbor**		Seqboot
Continuous characters	Contml		Fitch*** Kitsch**** Neighbor**		Seqboot Contrast
Discrete characters		Pars Mix Penny Move***** Dollop Dolpenny Dolmove***** Factor			Clique
Tree drawing	Drawgram Drawtree Consense Treedist Retree				

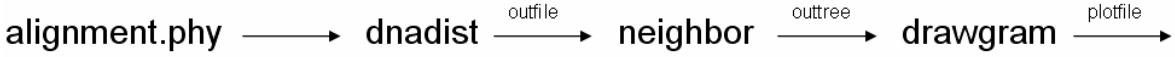
* utilized for matrix calculation
** NJ and UPGMA
*** FM without molecular clock
**** FM with molecular clock
***** interactive tree building

Flow charts

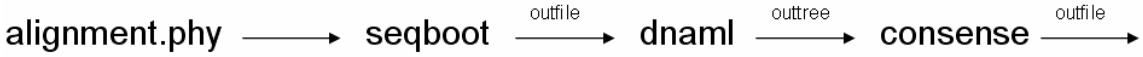
The following data flow charts describe some basic analyses. Flow chart A describes a maximum likelihood analysis for DNA sequences. Flow chart B describes an analysis using Neighbor joining method for DNA sequences. Flow chart C describes a bootstrapping analysis for DNA sequences using maximum likelihood as the analysis method. Note that you might need to rename the files between the analysis steps. For example, the outfile from Dnadist in flowchart B cannot be directly used in Neighbor. You should rename the outfile first, for instance, to infile.

A

B



C



In order to assess the reliability of the data using the bootstrapping method, you should first make the conventional analysis (flow chart A) using whatever analysis method is suitable for your purposes. After that, you should perform bootstrapping analysis (flow chart C) using exactly the same analysis method you used for conventional analysis. Here, we have used program Dnaml for maximum likelihood analyses, but it can be replaced with, *e.g.*, Dnapars for parsimony analysis or Dnadist + Neighbor for analysis using distance methods. Note that in bootstrapping analysis using Dnaml, Dnapars *etc.* it might not be advisable to use as many jumbles as in the conventional analysis, because the analysis programs will then perform a specified number of jumbles for every random sequence dataset, and that might take a very long time. A single jumble per bootstrap dataset is probably enough.

PHYLIP is freely available from http://evolution.genetics.washington.edu/phylip.html.

PHYLIP -- Phylogeny Inference Package (Version 3.2).
Felsenstein, J. 1989. *Cladistics* **5:** 164-166