

Getting a Tree Fast: Neighbor Joining, FastME, and Distance-Based Methods

Distance methods, and especially Neighbor Joining (NJ; Saitou and Nei, 1987), are popular methods for reconstructing phylogenies from alignments of DNA or protein sequences (UNIT 2.3). They are fast, allowing hundreds and even thousands of taxa to be dealt with by ordinary computers. The speed of these methods greatly simplifies the use of the bootstrap procedure (Page and Holmes, 1998; Graur and Li, 2000), which assesses the confidence level of inferred clades. They provide a simple way to incorporate knowledge of the evolution of the sequences being studied, depending on how the distance matrix is estimated. Numerous simulation studies have demonstrated their topological accuracy, and, unlike parsimony methods, they are not hampered by inconsistency (or “Felsenstein zone”; Swofford et al., 1996). The popularity of NJ, among the numerous existing distance-based methods, is explained by its speed and by the fact that its topological accuracy remains relatively close to that of recent approaches—i.e., FITCH (Felsenstein, 1997), BIONJ (Gascuel, 1997a), WEIGHBOR (Bruno et al., 2000), and FastME (Desper and Gascuel, 2002, 2004). However, several simulation studies (e.g., Vinh and Von Haeseler, 2005) showed that, with a high number of taxa, NJ is outperformed by FastME, both in terms of computing time and topological accuracy. Therefore, this latter program should be considered preferable for large-scale studies.

NJ and other current distance methods do not assume a molecular clock (Page and Holmes, 1998), as opposed to the Unweighted Pair Group Method Using Arithmetic averages (UPGMA; Sokal and Michener, 1958), which is precluded for most phylogenetic studies. The basic assumption is that sequences have been evolving along a tree and independently among the lineages. This tree can differ from the species tree in cases of horizontal transfer or sequence duplication (UNIT 6.1). Other assumptions are related to the sequence evolution model used to estimate distances. Models applicable to distance methods are homogeneous (i.e., constant over time) and assume that each site in the sequence evolves independently. However, some model parameters can differ from site to site. For example, mutation rates can vary across sites to represent structural/functional constraints on the residues, or the fast rate of the third codon position.

Distance methods are thus “model based,” just like maximum-likelihood methods (see Swofford et al., 1996, for discussion of these methods and comparison between them). However, the way the computations are performed is simpler and more approximate. Consequently, distance methods are faster than maximum-likelihood methods, but do not achieve the same topological accuracy. The comparison with parsimony is more complicated, since parsimony is sometimes inconsistent, but accurate when no long (e.g., outgroup) branch tends to attract other branches and perturb the resulting tree. A good practical approach is then to avoid parsimony when long branch attraction is suspected and otherwise to run both parsimony and distance approaches and compare the results.

Application of any distance-based method usually requires the following steps (see Fig. 6.3.1).

- a. Choose a sequence evolution model and use it to estimate the distance matrix (Support Protocols 1 and 2).
- b. Run the tree-building algorithm (Basic Protocol or Alternate Protocols 1, 2, or 3) and eventually return to step (a), for example to check that the resulting tree is

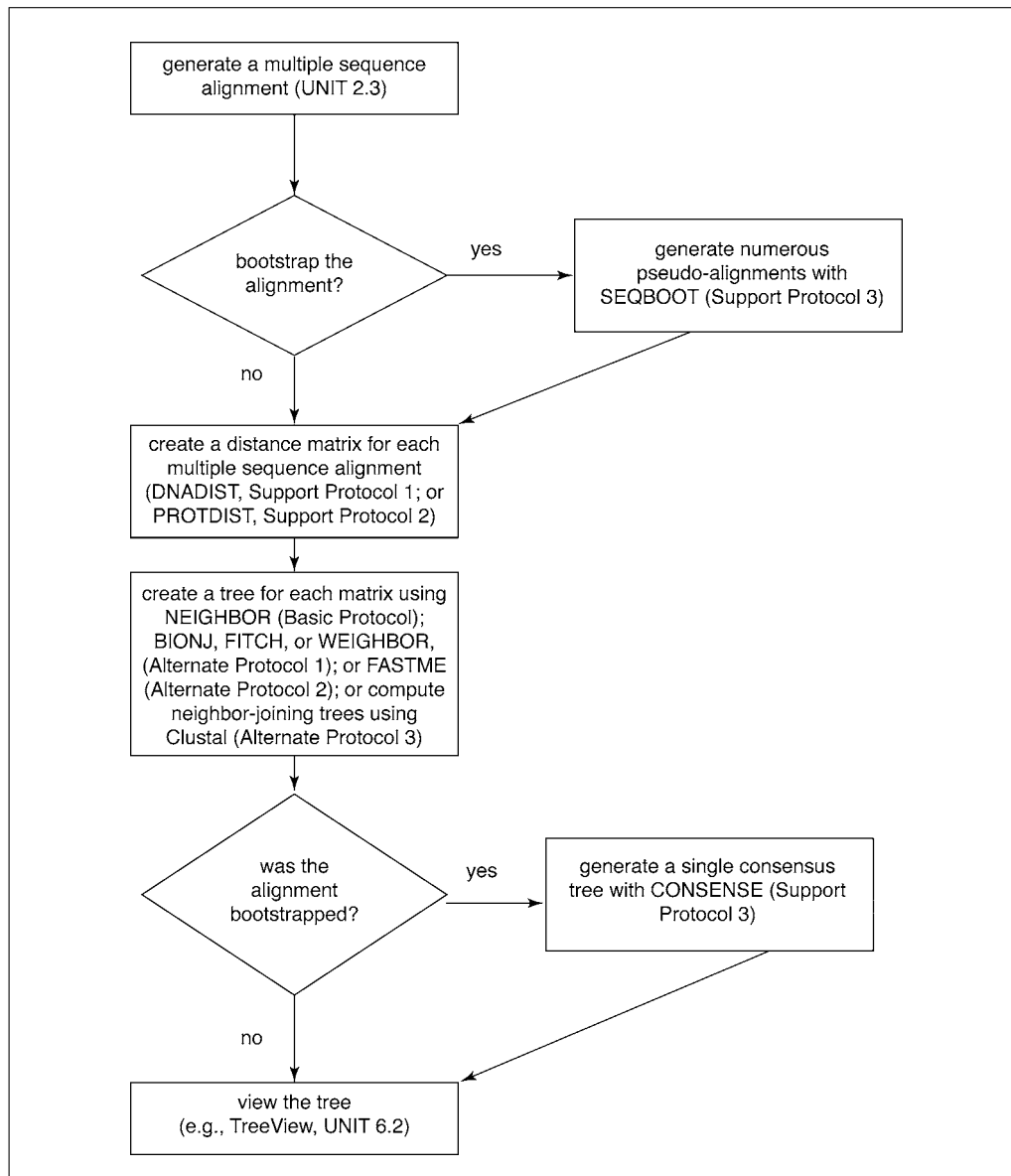


Figure 6.3.1 Flowchart illustrating the relationship between the multiple protocols presented in this unit.

not too sensitive to the model parameter values. The influence of taxon sampling, notably the presence/absence of the outgroup taxa, also has to be checked.

- c. Perform the bootstrap procedure to assess the significance level of the inferred clades (Support Protocol 3).

BASIC PROTOCOL

USING THE NEIGHBOR PROGRAM FROM THE PHYLIP PACKAGE TO CONSTRUCT A PHYLOGENETIC TREE

This protocol describes the use of NEIGHBOR (see Fig. 6.3.1), included in the PHYLIP 3.6 package (latest 3.65 version is identical; Felsenstein, 1989), which is distributed by Joe Felsenstein (University of Washington) and is one of the most widely used software packages in phylogeny studies. NEIGHBOR is the PHYLIP implementation of Neighbor Joining (Saitou and Nei, 1987). Distance estimation is performed using DNADIST or PROTDIST (Support Protocols 1 and 2). To accomplish the bootstrap procedure, first resample the sites using SEQBOOT (Support Protocol 3), then apply DNADIST or PROTDIST, run NEIGHBOR, and extract the bootstrap tree using CONSENSE (Support

**Getting a Tree
Fast: Neighbor
Joining, FastME,
and Distance-
Based Methods**

6.3.2

Protocol 3). Finally, the resulting tree can be drawn using a program such as TreeView (UNIT 6.2) or NJplot (Perrière and Gouy, 1996).

Necessary Resources

Hardware

PHYLIP executables are available for Windows, Mac OS 9 and OS X, and Linux. The PHYLIP C source code is also available for Unix, Linux, or OpenVMS systems.

Software

PHYLIP is available for free from <http://evolution.genetics.washington.edu/phylip.html>. The package contains C source codes, documentation files, and a number of different types of executables. Its Web page contains information on PHYLIP and ways to transfer the executables, source code, and documentation. The documentation is remarkably clear and complete, and provides a number of useful references.

Files

NEIGHBOR requires a distance matrix (or a set of distance matrices when the bootstrap procedure is used), which is estimated by DNADIST (Support Protocol 1) or PROTDIST (Support Protocol 2) from a multiple sequence alignment (e.g., UNIT 2.3). The file contains a number of taxa on its first line. Each taxon starts a new line with the taxon name, followed by the distance to the other taxa, and there is a new line after every nine distances. Taxon names have ten characters and must be blank-filled to be of that length. The default matrix format is square (Fig. 6.3.2) with zero distances on the diagonal. In the case of multiple matrices, as obtained with the bootstrap, matrices are given in the same format one after the other, without omitting the number of taxa at the beginning of each new matrix.

1. Download and install PHYLIP according to the program documentation (see Necessary Resources, above).
2. Generate a distance matrix for the multiple sequence alignment of interest by running either DNADIST (for DNA sequence alignments; see Support Protocol 1) or PROTDIST (for protein sequence alignments; see Support Protocol 2).
3. Begin a NEIGHBOR session in PHYLIP by double clicking on its icon.
4. At the prompt, enter the distance matrix file name and the name for the outfile, which will contain a simple representation of the output tree. The default files are `infile` and `outfile`, respectively, but the authors strongly recommend redefining these files to avoid possible confusions or deleting previously computed files.

8									
Candida_al	0.0000	0.0939	0.0224	0.1737	0.1632	0.2507	0.2757	0.3050	
Saccharomy	0.0939	0.0000	0.0966	0.1434	0.1582	0.2381	0.2064	0.2614	
Candida_tr	0.0224	0.0966	0.0000	0.1791	0.1632	0.2591	0.2855	0.3160	
Protomyces	0.1737	0.1434	0.1791	0.0000	0.0259	0.2235	0.2232	0.2820	
Taphrina_d	0.1632	0.1582	0.1632	0.0259	0.0000	0.2585	0.2581	0.3318	
Filobasidi	0.2507	0.2381	0.2591	0.2235	0.2585	0.0000	0.1386	0.1370	
Spongipell	0.2757	0.2064	0.2855	0.2232	0.2581	0.1386	0.0000	0.0791	
Athelia_bo	0.3050	0.2614	0.3160	0.2820	0.3318	0.1370	0.0791	0.0000	

Figure 6.3.2 Distance matrix in square format.

```

C:\PHYLIP\exe\neighbor.exe
Please enter a new file name> test-matrix

neighbor.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>
f
Please enter a new file name> test-outfile

Neighbor-Joining/UPGMA method version 3.6a2.1
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)? (none)
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
neighbor.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>
f
Please enter a new file name> test-outtree

```

Figure 6.3.3 The NEIGHBOR screen showing options for renaming files as well as options for settings and their defaults.

When a file called *infile* already exists in the *PHYLIP* directory, *NEIGHBOR* does not ask for the input file and reads the existing *infile*. Similarly, the option of renaming the output is only given if a file called *outfile* already exists. If no such file exists, *NEIGHBOR* automatically writes the output to a file called *outfile*.

5. After entering the file information, select among several options (see Fig. 6.3.3), which, a priori, have to be used with their default values, except **M** in the case of the bootstrap procedure. When options have been determined, type **Y** to run *NEIGHBOR*.

These options are as follows. **N** defines the method to be used; *NJ* (default option) is preferred over *UPGMA*, which assumes a molecular clock. **O** makes it possible to specify which species is to be used to root the tree; when **O** is on, the user is asked for the rank of the outgroup species in the input (matrix) file, otherwise the default outgroup species is the first; this outgroup (rooting) species is used in the tree printed in the *outfile*. **L** and **R** have to be switched on when the matrix is not square but lower-triangular and upper-triangular, respectively. **S** has to be on when the data contain subreplicates; it allows *NEIGHBOR* to read the input data, but the number of replicates is ignored. **J** enables one to choose a random order of species; the user is then asked for a "seed"; however, *NEIGHBOR* is almost insensitive to species ordering. **M** has to be used in the case of the bootstrap procedure (Support Protocol 3) to provide the number of pseudo-matrices. **0** defines the terminal type; this may affect the ability of the programs to display their menus and results, but the none option is usually sufficient. The **1** and **2** options are used to check the data and the progress of run; the authors suggest switching them off, notably for large trees and bootstrap studies. When **3** is Yes (default value), the tree or trees are printed in the *outfile*; this is useful to quickly visualize trees with moderate

```

(( (Candida_tr:0.0137,Candida_al:0.0086):0.0526,Saccharomy:0.0316):0.0351,
 (Taphrina_d:0.0160,Protomyces:0.0098):0.0665,
 (Athelia_bo:0.0600,Spongipell:0.0190):0.0480,Filobasidi:0.0612):0.0964);

(Candida_tr:0.01367,(Saccharomy:0.03307,((Protomyces:0.00957,
Taphrina_d:0.01633):0.06809,(Filobasidi:0.05464,(Spongipell:0.01908,
Athelia_bo:0.06002):0.04361):0.10745):0.03164):0.05098,Candida_al:0.00873);

```

Figure 6.3.4 Two trees in Newick format, which were obtained from the distance matrix in Figure 6.3.2 by BIONJ and NEIGHBOR, respectively. Both trees have identical topologies, but slightly different branch lengths.

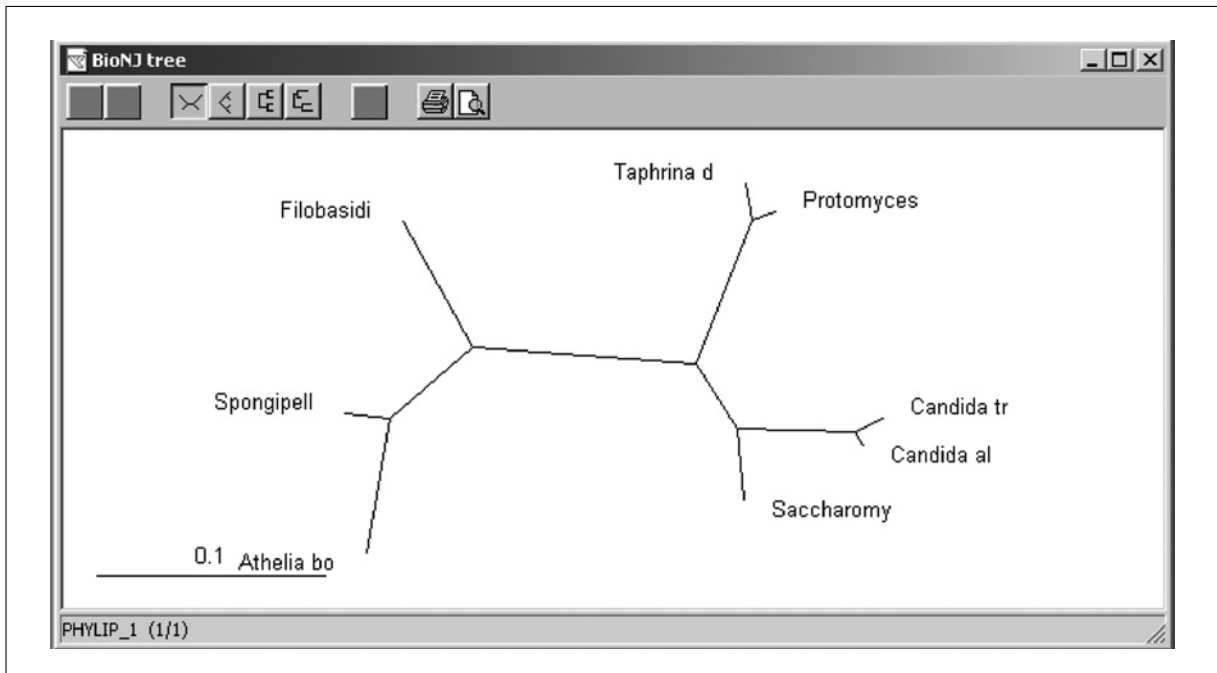


Figure 6.3.5 TreeView representation of the BIONJ tree of Figure 6.3.4.

numbers of taxa. When 4 is Yes (default value), the trees are written in Newick format in the outtree file, and can then be drawn using TreeView (UNIT 6.2) or, in case of multiple data sets, combined by CONSENSE to obtain the bootstrap tree (Support Protocol 3). To change the default values, simply type the option character. For example, typing 2 changes the progress of run status from Yes to No, and typing 2 again returns one to Yes.

- Finally, NEIGHBOR asks for the outtree file, which will contain the tree in Newick format (UNIT 6.2). The resulting tree can be visualized in the outfile, but a better view is obtained by applying TreeView (UNIT 6.2) to the outtree file.

The option of renaming the outtree file is only given if a file called outtree already exists. If no such file exists, NEIGHBOR automatically writes the output to a file called outtree, which may be a source of confusion. Inferred trees are unrooted and written in Newick format (UNIT 6.2). For example, the BIONJ tree in Figure 6.3.4 is made of three subtrees, containing (Candida_tr, Candida_al, and Saccharomy), (Taphrina_d and Protomyces) and (Athelia_bo, Spongipell, and Filobasidi), respectively, as can be shown from its TreeView representation (Fig. 6.3.5; see UNIT 6.2 for discussion of TreeView and Newick). Each subtree is made up of two subtrees or taxa; the numbers in Figure 6.3.4 indicate the branch lengths. Both trees in Figure 6.3.4 have identical topologies (even when the way they are encoded in Newick format looks quite different) but (slightly) different branch lengths.

```

+Candida_tr
!
! +-Saccharomy
! !
4--5      +Protomyces
! ! +---3
! ! ! +Taphrina_d
! +-6
! !      +---Filobasidi
!      +-----2
!          ! +Spongipell
!          +---1
!          +---Athelia_bo
!
+Candida_al

```

Figure 6.3.6 NEIGHBOR tree, as represented in the `outfile`.

Applying NEIGHBOR to the matrix of Figure 6.3.2, one obtains in the `outfile` the tree shown in Figure 6.3.6, while in the `outtree` file we have the second tree from Figure 6.3.4, in Newick format. This tree is equivalent to that of Figure 6.3.5.

7. To assess the tree quality, bootstrap the tree according to Support Protocol 3.

SUPPORT PROTOCOL 1

DISTANCE MATRIX ESTIMATION FROM DNA (OR RNA) SEQUENCES USING DNADIST

Distance estimation is the first step in reconstructing a phylogenetic tree using a distance-based method. DNADIST, from the PHYLIP package, estimates the pairwise evolutionary distances between nucleotide sequences under various models of nucleotide substitutions. These models account for hidden substitutions and incorporate knowledge about the mutation process. Distance estimation is based on the maximum-likelihood principle (Swofford et al., 1996). The model choice is sensitive and influences the distance values, and then the tree to be constructed. DNADIST reads a multiple sequence alignment and outputs a distance matrix. When the bootstrap procedure is used, the input file contains the pseudo-alignments one after the other, and the output file contains the corresponding pseudo-matrices in the same order.

Necessary Resources

Hardware

PHYLIP executables are available for Windows, Mac OS 9 and OS X, and Linux. The PHYLIP C source code is also available for Unix, Linux, or OpenVMS systems.

Software

DNADIST is part of the PHYLIP package. PHYLIP is available for free from <http://evolution.genetics.washington.edu/phylip.html>. The package contains C source codes, documentation files, and a number of different types of executables. Its Web page contains information on PHYLIP and ways to transfer the executables, source code, and documentation. The documentation is remarkably clear and complete, and provides a number of useful references.

Files

DNADIST requires DNA multiple sequence alignments in PHYLIP format, as obtained from alignment programs such as ClustalX (UNIT 2.3). The first line contains the number of taxa and sites; next come the taxon data with a new line per taxon. Taxon names have ten characters and must be blank-filled to be of that

```

8 95
Candida_al      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGTTGTTGTTCTTTTATT-GACGCAAT
Saccharomy      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGGTGGTGTTTTTTTAAT-GACCCACT
Candida_tr      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGTTGTTGTTCTTTTATT-GACGCAAT
Protomyces      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGGCGATGTTCTTTTCTT-GACTCGCC
Taphrina_d      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGGCGATGTTCTTTTCTT-GACTCGCC
Filobasidi      AGTCTTAACAGTAAACGATGCCGACTAGGGATCGGCCACGTCAATCTCT--GACTGGGT
Spongipell      AGTCTTAACAGTAAACTATGCCGACTAGGGATCGGGCGATCTCAAACCTT-ATGTGTCGCT
Athelia_bo      AGTCTTAACAGTAAACTATGCCGACTAGGGATCGGACAACCTCAATTTTGATGTGTGTT

CGGCACCTTACGAGAAATCA-AAGTCTTTGGGCC
CGGCACCTTACGAGAAATCA-AAGTCTTTGGGTTC
CGGCACCTTACGAGAAATCA-AAGTCTTTGGGG?
CGGCACCTTATGAGAAAGGA-AAGTTTTTGGGTTC
CGGCACCTTATGAGAAAAA????????????
CGGCACCTTACGAGAAATCA-AAGTCTTTGGGTTC
CGGCACCTTACGAGAAATCA-AAGTCTTTGGGTTC
CGGCACCTTACGAGAAATCA-AAGTCTTTGGGTTC

```

Figure 6.3.7 Alignment in interleaved PHYLIP format.

```

8 95
Candida_al      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGTTGTTGTTCTTTTATT-GACGCAAT
                  CGGCACCTTACGAGAAATCA-AAGTCTTTGGGCC
Saccharomy      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGGTGGTGTTTTTTTAAT-GACCCACT
                  CGGCACCTTACGAGAAATCA-AAGTCTTTGGGTTC
Candida_tr      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGTTGTTGTTCTTTTATT-GACGCAAT
                  CGGCACCTTACGAGAAATCA-AAGTCTTTGGGG?
Protomyces      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGGCGATGTTCTTTTCTT-GACTCGCC
                  CGGCACCTTATGAGAAAGGA-AAGTTTTTGGGTTC
Taphrina_d      AGTCTTAACCATAAACTATGCCGACTAGGGATCGGGCGATGTTCTTTTCTT-GACTCGCC
                  CGGCACCTTATGAGAAAAA????????????
Filobasidi      AGTCTTAACAGTAAACGATGCCGACTAGGGATCGGCCACGTCAATCTCT--GACTGGGT
                  CGGCACCTTACGAGAAATCA-AAGTCTTTGGGTTC
Spongipell      AGTCTTAACAGTAAACTATGCCGACTAGGGATCGGGCGATCTCAAACCTT-ATGTGTCGCT
                  CGGCACCTTACGAGAAATCA-AAGTCTTTGGGTTC
Athelia_bo      AGTCTTAACAGTAAACTATGCCGACTAGGGATCGGACAACCTCAATTTTGATGTGTGTT
                  CGGCACCTTACGAGAAATCA-AAGTCTTTGGGTTC

```

Figure 6.3.8 Alignment in sequential PHYLIP format.

length. The taxon names are followed by the sequences, which must either be “interleaved” or “sequential” (Figs. 6.3.7 and 6.3.8). The sequences can have internal blanks in the sequence but there must be no extra blanks at the end of the terminated line. The three symbols N, X and ? indicate an unknown nucleotide while a dash (–) indicates a deletion. In the case of multiple data sets, as provided by SEQBOOT, pseudo-alignments are given in the same format one after the other, without omitting the number of taxa and the number of sites at the beginning of each new set.

1. Download and install the PHYLIP package, and initialize a DNADIST session by double clicking on its icon.
2. At the prompt, enter the sequence alignment file name and the name for the output, which will contain the distance matrix. The default files are `infile` and `outfile`, respectively, but the authors strongly recommend redefining these files to avoid possible confusion, or deletion of previously computed files.

If a file called `infile` already exists in the PHYLIP directory, DNADIST does not ask for the input file, but reads the existing `infile`. Similarly, the option of renaming the output is only given if a file called `outfile` already exists. If no such file exists, DNADIST automatically writes the output to a file called `outfile`.

```

C:\PHYLIP\exe\dnadist.exe
dnadist.exe: can't find input file "infile"
Please enter a new file name> test-DNA

dnadist.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>
f
Please enter a new file name> test-matrix

Nucleic acid sequence Distance Matrix program, version 3.6a2.1

Settings for this run:
D Distance <F84, Kimura, Jukes-Cantor, LogDet?? F84
G Gamma distributed rates across sites? No
I 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?? <none>
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

```

Figure 6.3.9 The DNADIST screen with options for renaming files and setting parameters. The default parameters are shown.

- Then the menu of Figure 6.3.9 appears, which asks for important and sensitive choices.

The remaining steps of this protocol primarily describe options requiring in-depth explanations or where the default values often need to be changed. More details are given in the DNADIST documentation. To change the default values, simply type the option character. For example, typing I changes the sequence format from interleaved to sequential, and typing I again returns to the interleaved format.

Set the parameters

- D** defines the substitution model. All models assume that sites evolve independently. The four available models are nested, i.e., Jukes-Cantor is a special case of Kimura, which is a special case of F84, which is a special case of LogDet. Jukes-Cantor (Jukes and Cantor, 1969) assumes only one substitution rate, Kimura (Kimura, 1980) allows for a difference between transition and transversion rates, while F84 (Kishino and Hasegawa, 1989; Felsenstein and Churchill, 1996) is similar to Kimura but allows for different frequencies of the four nucleotides, and LogDet does not impose any restriction on the 16 rates (except those induced by the Markovian nature of the process). So LogDet (Steel, 1994) is the most flexible model, but is often overparametrized, unless the sequences are very long (say >3000). F84 (the default option) is a good compromise, notably when the base frequencies are not equal. When they are almost equal, Kimura is a good choice, while Jukes-Cantor is overly simple in most cases.

Note that all sites (informative or not) must be given to DNADIST for these models to be used in the correct way.

- G** asks whether or not the substitution rates vary across sites. Biologically speaking, the answer is clearly yes. It has been demonstrated that the Gamma distribution (Swofford et al., 1996), which is defined by a parameter usually denoted as α , is a

good model to account for this variability. α was estimated between 0.05 and 1.0 for numerous data sets (Yang, 1966), which indicates that rates strongly vary across sites (variability increases as α decreases). However, the default option of DNADIST is to not correct for this variability (i.e., $\alpha = \infty$), which is a common practice.

Jin and Nei (1990) recommend using $\alpha = 1.0$ or 2.0 . The authors have recently demonstrated (Guindon and Gascuel, 2002) that uncorrected distances are often better suited, especially when the molecular clock is more or less satisfied. Therefore, a pragmatic approach is to use the default option, and to check whether or not using a reasonable value (e.g., 1.0 or 2.0) for α changes the result.

However, DNADIST does not use the standard α parameter, but rather the “coefficient of variation” (CV), which is equal to $1/\alpha^{1/2}$. One obtains $CV = 2.0, 1.0,$ and 0.5 when $\alpha = 0.25, 1.0,$ and $4.0,$ respectively. Moreover, the LogDet model cannot be combined with the gamma correction.

6. **T** asks for the transition/transversion ratio. The default value is 2.0, and there is no way to estimate this value within PHYLIP.

Hopefully, the results are not very sensitive to the value of this parameter (unless it is extreme). It is possible to estimate it using simple formulas from Kimura (1980).

7. **C** allows user-defined categories, for example to specify that third-position bases have a different rate than first and second positions. This option allows the user to make up to 9 categories of sites, but, as with the LogDet model, using too many categories can make the model overparametrized. The user is asked for the relative rates within each category. The assignment of rates to sites is then made by reading a file whose default name is `categories`.

An example and more details are given in the DNADIST documentation. There is no program from PHYLIP for estimating the different rates, but just as with the above ratio, these parameters are not very sensitive (unless extreme).

8. **W** allows the user to select subsets of sites. Basically it has to remain “No” (the default value), unless the user wants to check the influence of various categories of sites.

See DNADIST documentation for more details.

9. **F** must remain as Yes in any practical situation.

10. **L** defines the matrix format, square (default value) or lower-triangular.

11. **M** has to be used in the bootstrap procedure (see Support Protocol 3). The user is then asked for the number of pseudo-alignments in the input file. Otherwise the default value (No) is required.

12. **I** defines the multiple sequence alignment format, which is interleaved or sequential (Fig. 6.3.7 and 6.3.8, respectively).

13. Once all options have been determined, type Y to compute the distance matrix.

With the working example of Figure 6.3.7 and all default values, DNADIST returns the matrix of Figure 6.3.2.

DISTANCE MATRIX ESTIMATION FROM PROTEINS USING PROTDIST

PROTDIST is analogous to DNADIST (Support Protocol 1). It is first necessary to provide the file names, which the program initially assumes to be `infile` and `outfile` (see Fig. 6.3.10). The `infile` contains the protein multiple sequence alignment (UNIT 2.3). The format is analogous to that used with nucleotide sequences (Support Protocol 1), except that, with proteins, the three symbols X, -, and ? indicate an unknown amino acid,

SUPPORT PROTOCOL 2

Inferring Evolutionary Relationships

6.3.9

```

C:\PHYLIP\exe\protdist.exe
protdist.exe: can't find input file "infile"
Please enter a new file name> test-PROT

protdist.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>
f
Please enter a new file name> test-matrix

Protein distance algorithm, version 3.6a2.1
Settings for this run:
P   Use JTT, PAM, Kimura or 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)?             <none>
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>

```

Figure 6.3.10 The PROTDIST screen showing options for renaming files and setting parameters. The default parameter settings are shown.

a deletion, and an unknown including deletion, respectively (see PHYLIP documentation `sequence.html` for more details). The final distance matrix is written to `outfile`, unless the user selects a different name. After providing the file names, the user then deals with the options (see Fig. 6.3.10). The main option is **P**, which selects among five substitution models differing depending on the matrix of substitution rates. The substitution models are as follows:

Dayhoff PAM 001 matrix. This matrix (Dayhoff et al., 1979) is an empirical one that scales probabilities of change from one amino acid to another, assuming that the total change between the two amino acid sequences is 1% (UNIT 3.5). It allows the evolutionary distance to be computed in terms of expected fraction of amino acids changed.

PMB (Probability Matrix from Blocks). This model is derived (Veerassamy et al., 2003) using the Blocks database of conserved protein motifs and is a continuation of BLOSUM scoring matrices, which are widely used for protein sequence alignments (UNIT 3.5). Note that this model is only available in the latest PHYLIP version (3.65).

Jones-Taylor-Thornton model. This model (Jones et al., 1992) is analogous to PAM, but the estimation of the probabilities of change is based on a much larger set of proteins. Thus it is to be preferred over the original PAM.

Kimura's distance. This model (Kimura, 1983) assumes only one substitution rate, and does not take into account which amino acids differ.

The Categories distance. This model, devised by Joe Felsenstein, is conceptually close to Kimura's two-parameter model for DNA sequences (Kimura, 1980). The amino acids are grouped into a series of categories, and a distinction is made between transitions (change within a category) and transversions (change from one category to another). When this option is selected, the user is asked for a number of other options (e.g., the amino acid categorization), but the authors suggest using default values that approximate the PAM model (UNIT 3.5).

As already stated, the Jones-Taylor-Thornton model is preferred over PAM in any situation, while PMB seems to be an interesting (but new) option. These three models, however, require heavy computation, and the same holds for the Categories model. The Kimura model is therefore a good option for large data sets or atypical (e.g., membrane) proteins. For the other options, see comments on DNADIST in Support Protocol 1.

BOOTSTRAPPING USING SEQBOOT AND CONSENSE

A tree such as that shown in Figure 6.3.5 does not indicate the reliability of the inferred clades. The bootstrap procedure is a sound and accurate way to obtain this information, and its use is greatly facilitated by the speed of distance methods. Within PHYLIP, the bootstrap procedure is achieved as shown in the flowchart of Figure 6.3.1. It successively uses: (1) SEQBOOT, (2) DNADIST or PROTDIST, (3) NEIGHBOR (or any other distance method, see Alternate Protocols 1 and 2), and (4) CONSENSE.

Necessary Resources

Hardware

PHYLIP executables are available for pre-386 DOS, 386/486/Pentium DOS, Windows 3.1, Windows 95/98/NT, 68k Macintosh, or PowerMac. The PHYLIP C source code is also available for Unix, Linux, or VMS systems.

Software

SEQBOOT and CONSENSE are part of the PHYLIP Package. PHYLIP is available for free from <http://evolution.genetics.washington.edu/phylip.html>. The package contains C source codes, documentation files, and a number of different types of executables. Its Web page contains information on PHYLIP and ways to transfer the executables, source code, and documentation. The documentation is remarkably clear and complete, and provides a number of useful references.

Files

SEQBOOT requires a multiple sequence alignment in the PHYLIP format, as obtained from alignment programs, such as ClustalX (UNIT 2.3); it computes pseudo-alignments by sampling at random with replacement the sites in the original (input) alignment, and outputs these pseudo-alignments in the PHYLIP format. Pseudo-alignments are processed by DNADIST or PROTDIST (Support Protocols 1 and 2) and transformed into pseudo-matrices, which are written in the PHYLIP format. The pseudo-matrix file is then used by NEIGHBOR to build pseudo-trees, written in Newick format (Basic Protocol and UNIT 6.2). Finally, the pseudo-tree file is used in CONSENSE to obtain the bootstrap tree, also written in Newick format.

1. After downloading and installing PHYLIP, start a SEQBOOT session by doing double clicking on its icon.
2. Create pseudo-alignments from the aligned sequences using SEQBOOT.

*The SEQBOOT screen is illustrated along with its options in Figure 6.3.11. To obtain more reliable results, the **R** option, which corresponds to the number of replicates, has to be changed from 100 (the default value) to 1000 (or more in large studies). SEQBOOT allows for site categories and weights (options **W** and **C**, see Support Protocol 1). **F** can be used for large studies to save space on one's system (see SEQBOOT documentation file). **I**, **0**, **1** and **2** have the same meaning as in other PHYLIP programs (see Support Protocol 1); the authors suggest switching **2** to avoid displaying the (extensive and useless) progress of run on the terminal. The default values have to be conserved for the other options, which correspond to non-sequence data (**D**) or other resampling procedures (**J** and **B**).*

```

C:\PHYLIP\exe\seqboot.exe
seqboot.exe: can't find input file "infile"
Please enter a new file name> test-BOOT

Bootstrapping algorithm, version 3.6a2.1

Settings for this run:
D Sequence, Morph, Rest., Gene Freqs? Molecular sequences
J Bootstrap, Jackknife, Permute, Rewrite? Bootstrap
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
F Write out data sets or just weights? Data sets
I Input sequences interleaved? Yes
0 Terminal type (IBM PC, ANSI, none)? (none)
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
Random number seed (must be odd)?
19

seqboot.exe: the file "outfile" that you wanted to
use as output data 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)
f
Please enter a new file name> Pseudo-alignments

```

Figure 6.3.11 The SEQBOOT screen showing options for renaming files and setting parameters. The default parameters are shown.

3. Apply DNADIST (Support Protocol 1) to the pseudo-alignment file to obtain the pseudo-matrices.

*DNADIST is used as described in Support Protocol 1, except that the number of data sets (replicates) must be given using the **M** option. Switching the **2** option is also relevant.*

4. Apply NEIGHBOR (Basic Protocol) or any other distance method (Alternate Protocols 1 and 2) to the pseudo-matrix file, indicating the number of matrices with the **M** option, and switching the **2** option.
5. Obtain the bootstrap tree by double-clicking CONSENSE and applying it to the pseudo-tree file.

*The CONSENSE screen is illustrated along with its options in Figure 6.3.12. The default input file name is intree, while, as for NEIGHBOR (Basic Protocol), the outfile will contain a simple representation of the bootstrap tree. The **C** option defines the type of consensus method; MR or Mre should be selected. The former will provide only clades occurring in more than 50% of the pseudo-trees, while the latter will complete these well supported clades by clades below 50%; only bootstrap supports above 50% have a clear mathematical meaning (Berry and Gascuel, 1996), but lower supports can be informative in some cases. The threshold of clade selection can also be user-defined by selecting M1. **O** has the same meaning as for NEIGHBOR (Basic Protocol) and can be used to define the outgroup species. **R** has to remain No when using NEIGHBOR and related methods that infer unrooted trees. **T** defines the terminal type, just like **0** in other PHYLIP programs (Basic Protocol). When **1** is on, CONSENSE outputs in outfile the species list and all clades that belong to at least one of the pseudo-trees. When option **3** is turned off, the outfile is not created and this cancels (among other things) the previous option. When option **4** is on, the bootstrap tree in Newick format is written in the outtree file. Finally, switching on the **2** (progress of run) option is relevant.*

```

C:\PHYLP\exe\consense.exe
consense.exe: can't find input tree file "intree"
Please enter a new file name> pseudo-trees

consense.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)
f
Please enter a new file name> bootstrap-file

Consensus tree program, version 3.6a2.1
Settings for this run:
C          Consensus type (MRe, strict, MR, M1):  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):  (none)
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>
y

consense.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)
f
Please enter a new file name> bootstrap-tree

```

Figure 6.3.12 The CONSENSE screen showing options for renaming files and setting parameters. The default parameters are shown.

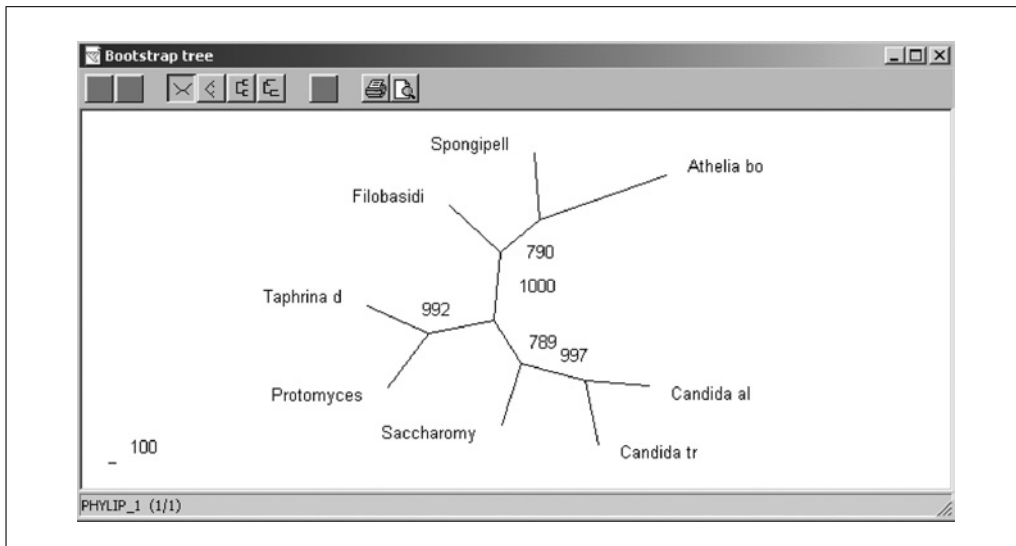


Figure 6.3.13 TreeView representation of the bootstrap tree that is obtained with NEIGHBOR with 1000 replicates. Bootstrap supports are associated with internal branches (or clades). For example, Spongipell, Athelia_bo is supported by 790 pseudo-trees out of 1000.

- Finally, CONSENSE requests a new name for the outtree file. Although it is possible to view the resulting tree in the outfile, a better view is obtained by applying TreeView (UNIT 6.2) to the outtree file.

When applying these steps (with 1000 replicates) to the original alignment of Figure 6.3.7, the bootstrap tree of Figure 6.3.13 is obtained. The branch lengths correspond to the bootstrap supports, which are explicitly shown in the case of internal branches. Note that due to the random nature of the process, bootstrap supports can differ slightly from one run to another.

USING BIONJ, WEIGHBOR, OR FITCH TO CONSTRUCT A TREE

This protocol provides descriptions of BIONJ and WEIGHBOR, which are PHYLIP-compatible, and FITCH, which is available in PHYLIP. These three programs have a higher topological accuracy than NEIGHBOR. The resulting trees, however, are often similar or identical to NEIGHBOR trees, at least with a low number of taxa (e.g., <20). When this number increases, the various methods tend to return different trees, and their advantage over NEIGHBOR increases. BIONJ is about the same speed as NEIGHBOR; WEIGHBOR is about 500 times slower than NEIGHBOR; and FITCH is even slower than WEIGHBOR (see below for more details). The matrix distance computation (Support Protocols 1 and 2) and the bootstrap procedure (Support Protocol 3), are not described here, since they are performed exactly as with NEIGHBOR (Basic Protocol).

Using BIONJ

BIONJ is available free from <http://www.lirmm.fr/~w3ifa/MAAS/BIONJ/BIONJ.html>. This Web page contains documentation and articles, test sets, and executables for Windows PC and PowerMac, as well as the C source code. Once downloaded (and compiled on Unix and related systems), BIONJ must be placed in the PHYLIP directory.

BIONJ asks for the distance matrix input file and the name of the tree output file. The distance matrix must be square and written in PHYLIP format. The file can contain one or several matrices, as obtained when using SEQBOOT plus DNADIST or PROTDIST, but the user is not asked for the number of matrices. BIONJ then returns as many trees as there are matrices. These trees are written in Newick format (*UNIT 6.2*). In case of a single matrix, the resulting tree can be viewed using TreeView (*UNIT 6.2*). With multiple matrices and trees, CONSENSE (Support Protocol 3) must be used, just as with NEIGHBOR (Basic Protocol).

Applying BIONJ to the matrix of Figure 6.3.2, the tree of Figure 6.3.4 is obtained, with TreeView representation as shown in Figure 6.3.5. This tree differs from the NEIGHBOR tree (Basic Protocol) by the branch lengths but not by the topology, which is not surprising in view of the low number of taxa.

Using WEIGHBOR

WEIGHBOR is available in C source code free from <http://www.t10.lanl.gov/billb/weighbor/>. This Web page contains documentation, the seminal article, the C source code, and selected executables for the Windows, Linux, and Solaris operating systems. Then, WEIGHBOR must be placed into the PHYLIP directory.

Just like BIONJ, WEIGHBOR asks for the input and output files, and the input file can contain one or several matrices. WEIGHBOR then asks for the sequence length and the number of symbols, i.e., 4 for DNA or RNA sequences and 20 for proteins. When the input file contains several matrices, WEIGHBOR returns the same number of trees, which must be dealt with by CONSENSE (Support Protocol 3), just as with NEIGHBOR (Basic Protocol).

Using FITCH

FITCH is available in PHYLIP and runs on numerous systems (see Basic Protocol, Necessary Resources).

FITCH is able to deal with multiple data sets, just as NEIGHBOR, BIONJ, or WEIGHBOR, and its menu is analogous to that of NEIGHBOR (see Basic Protocol). All options must, a priori, conserve their default values, except **G** and **J** which can be used to search the tree space more extensively (at the expense of longer run times). **G** can be switched to **Yes** to search for global rearrangements that improve the least-squares fit of the tree.

J takes advantage of the fact that FITCH does not systematically find the same tree, depending on the taxon ordering. When **J** is switched to Yes, FITCH asks for a seed to initiate the random ordering procedure, and then for the number of times the randomization procedure has to be used. The resulting tree is the best tree that is obtained from all random orderings. The higher their number, the better the solution, but the longer the computing time. A value of 10 seems to be a reasonable compromise, but is too high for large data sets, for which the **J** option has to be switched off.

USING FastME TO CONSTRUCT A TREE

This protocol provides a description of the phylogeny program FastME, which is PHYLIP-compatible. FastME builds trees with high accuracy, at least as accurately as NEIGHBOR and FITCH in simulations, but it builds the trees much more quickly, even more quickly than NEIGHBOR and BIONJ. This protocol describes FastME version 1.0, which has been fully studied and published (Desper and Gascuel, 2002, 2004), but a new version will be made available soon that handles all of the tasks of distance methods, including calculating distance matrices from sequence data and bootstrapping. FastME 1.0 only builds trees, just like NEIGHBOR. The matrix distance computation (Support Protocols 1 and 2) and the bootstrap procedure (Support Protocol 3) are not described here, since they are performed exactly as with NEIGHBOR (Basic Protocol).

FastME complies with the minimum evolution principle (ME), which involves minimizing the tree length (sum of branch lengths). The general approach is to do a topology search, calculating the tree length for each topology and searching for the topology with shortest tree length. FastME can do the search efficiently by using small topological moves, leading to changes in tree length that can be calculated quickly, without recalculating each individual branch length. It then starts from an initial tree and refines this tree until no more moves improve the current tree.

Necessary Resources

Hardware

FastME is written in C, and can be used on any platform that supports a C compiler. Executables are available for Windows, Linux, and Macintosh operating systems. These are command-line executables, but a PHYLIP-type interface is expected to be released soon. Software is available at the FastME home page (<http://atgc.lirmm.fr/fastme/>)

Software

The FastME program (<http://www.lirmm.fr/fastme>)

Files

See *UNIT 2.3* for the input formats. The input file can contain multiple data sets, if the user chooses, but each matrix should be square (as opposed to upper- or lower-diagonal).

1. FastME is called from the command prompt on all machines. To use FastME, open a command-line shell window (e.g., on a Windows machine by choosing Command Prompt under Accessories in the Start menu, or by choosing Run from the Start menu and typing `cmd` in the text box). Switch to the directory containing the data file(s) and call FastME by typing `fastme` at the command prompt, with the required arguments described below.

ALTERNATE PROTOCOL 2

All the options described in the following steps are displayed as part of the online help, which is obtained by the command `fastme -h`; see Figure 6.3.14. The bottom line in this figure runs FastME with default options, an input file called `distfile` that contains 1000 distance matrices, and an output file called `treefile`.

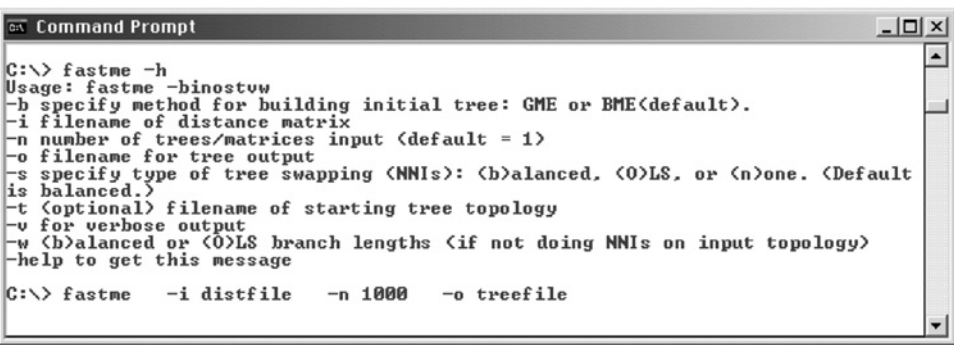
2. Use `-i` [name of input file] to specify the name of the input file, replacing the brackets and text within them with the name of the input file.
3. Use `-b` BME, `-b` GME (or `-b` NS) to specify the method for building the initial tree. BME and GME are the greedy insertion algorithms introduced in (Desper and Gascuel, 2002); BME (`-b` BME) greedily uses balanced least-squares minimum evolution, while GME (`-b` GME) uses ordinary least-squares minimum evolution. NS (`-b` NS) uses the neighbor-joining algorithm described in the Basic Protocol.

With n taxa, the fastest option is to use GME, which requires computational time proportional only to n^2 . BME requires time proportional to $n^2 \text{ diam}(T)$, where $\text{diam}(T)$ is the diameter of the tree, i.e., the length (number of branches) of the longest path in T . BME is the default option, being the most accurate, but GME is slightly faster and gives topological accuracy similar to that of BME after post-processing. GME and BME are “greedy,” i.e., they iteratively build on partial solutions by optimally inserting the new taxon, keeping the partial tree fixed. NS requires time proportional to n^3 and performs the same optimization as BME, over a slightly larger search space.

4. Alternatively, use `-t` [filename of starting tree topology] to start the topology search from a user-provided starting tree topology, instead of the BME or GME tree.

For example, it can be relevant to start with NJ (Basic Protocol) or BIONJ (Alternate Protocol 1) trees, which tend to be more accurate than GME and BME while being relatively fast (but, again, the difference tends to disappear after post-processing).

5. Use `-n` [number of trees/matrices input] to input the number of data sets to be considered. The default value for this parameter is 1.
6. Use `-o` [filename for tree output] to specify the name of the output file for the trees.
7. Use `-s` to specify type of nearest-neighbor interchanges (NNIs), or tree swapping, to be used as a post-processing step. FastME can do NNIs to search the tree topology space, while seeking to optimize either the balanced least-squares (`-s` BME) or the ordinary least-squares (`-s` OLS) minimum evolution criterion. It is also possible to forego this step using (`-s` none).



```
C:\> fastme -h
Usage: fastme -binostw
-b specify method for building initial tree: GME or BME(default).
-i filename of distance matrix
-n number of trees/matrices input (default = 1)
-o filename for tree output
-s specify type of tree swapping (NNIs): (b)alanced, (O)LS, or (n)one. (Default
is balanced.)
-t (optional) filename of starting tree topology
-v for verbose output
-w (b)alanced or (O)LS branch lengths (if not doing NNIs on input topology)
-help to get this message

C:\> fastme -i distfile -n 1000 -o treefile
```

Figure 6.3.14 FastME screen (in command-prompt window) showing options for renaming files and setting parameters.

The default is to use balanced least-squares (-s BME), and it is recommended to use this setting. The post-processing step is where FastME gets its power. Each NNI requires only time proportional to n, if OLS minimum evolution is used, or proportional to n diam(T) if balanced least-squares minimum is used. The latter option is recommended, as the BLS method has been shown to be considerably more accurate than OLS minimum evolution when biological data sets are considered.

8. Use -w b (for balanced) or -w O (for OLS) to select branch lengths to assign to a topology.

This option is only needed if -t is selected with an input topology, and -s none is selected, implying no topology searching is done. In this case FastME just estimates the branch lengths of this topology, according to one of the two approaches. Again, BME (-w b on the command line) should be preferred with biological data sets.

COMPUTING NJ TREES USING CLUSTAL

This protocol describes the use of Clustal (see *UNIT 2.3*) to build neighbor-joining (NJ) trees. Although Clustal is not intended primarily as a tree-building program, it is a useful tool for quickly getting a tree for a set of sequences. On the other hand, it does not provide the user with all of the possibilities of PHYLIP, notably concerning distance estimation. The program is available in two versions: ClustalX (Thompson et al., 1997), which has a graphical interface, and ClustalW, which has a text-based interface. ClustalW can be used interactively through a simple menu system or from the command line, which makes it a useful tool for batch processing alignments or generating phylogenies as part of a CGI script. This protocol will provide instructions for both the graphical interface of ClustalX, and the ClustalW command line.

Clustal can output trees in a variety of formats. The default is the Newick format used by many phylogenetic programs (see *UNIT 6.2* and Basic Protocol). Clustal can also write trees in its own format, and can save the pairwise distances in PHYLIP format.

Necessary Resources

Hardware

Clustal can be run on Macintosh, Windows, and Unix systems. For full details see *UNIT 2.3*.

Software

ClustalX or ClustalW

Files

See *UNIT 2.3* for the input formats.

Building an NJ tree

Before building a tree there are various options the user can set that control how the pairwise distances between sequences are computed, and the output format for the tree. In ClustalX these options are set using the commands on the Trees menu (Fig. 6.3.15); in ClustalW they are set on the command line.

1. Install ClustalX or ClustalW and create a multiple sequence alignment (*UNIT 2.3*).
2. Selecting the Exclude positions with gaps option (command-line equivalent in ClustalW, /TOSSGAPS) forces Clustal to ignore any site where a gap occurs in any of the sequences when computing pairwise distances (Fig. 6.3.15).

A priori, this command should be chosen, because distance estimation from sequences with gaps does not have sound mathematical foundations. However, removing all sites

ALTERNATE PROTOCOL 3

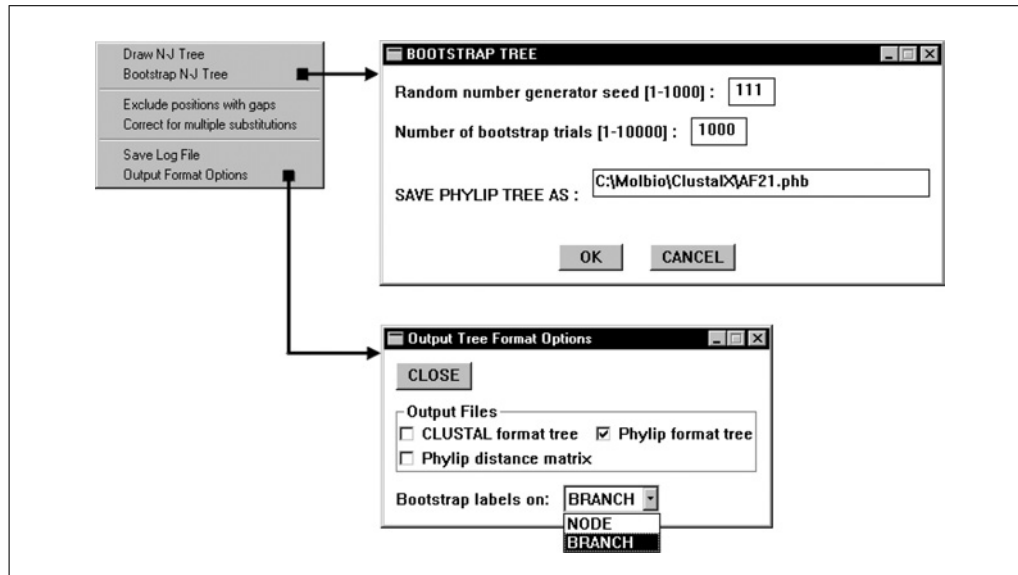


Figure 6.3.15 The Trees menu in the program ClustalX showing the menu commands and dialog boxes used to control how the program constructs neighbor-joining trees. Note that Exclude Positions with Gaps and Correct for Multiple Substitutions are not selected. If they were selected, a check mark would appear next to each option.

with a gap sometimes makes the phylogenetic signal so low that the resulting tree is no longer supported in the bootstrap procedure. So both approaches should be tested.

3. If the user selects the Correct for multiple substitutions option (indicated by a tick beside the menu command; command-line equivalent in ClustalW, /KIMURA), then ClustalX will use either the Kimura 2-parameter model (Kimura, 1980) or Kimura's (1983) correction for nucleotides and proteins, respectively, to compute pairwise distances between sequences.

This should be the default option. If this option is not chosen, then there is no correction for multiple substitutions.

4. Select the output format.

The menu of options for ClustalX is shown in Figure 6.3.15. The default output tree format is the Newick (or "PHYLIP") format. The command-line equivalent (ClustalW) for this format is: /OUTPUTTREE=phylip. ClustalX can also write trees in its own format by checking the CLUSTAL format tree box (/OUTPUTTREE=nj), and can save the pairwise distances in PHYLIP format by checking the PHYLIP distance matrix box (/OUTPUTTREE=dist).

5. Having set the options for the analysis and output (or having simply taken the defaults), the command Draw N-J Tree will construct the tree.

In ClustalX the user is presented with a dialog box asking for confirmation of the output tree file name. Typically, the tree file is given the name of the user's sequence file plus the extension .phb. Click on OK to construct the tree. However, the Draw N-J Tree command is somewhat oddly named, as it does not "draw" the tree. To see the tree, you will need to use a tree drawing program such as TreeView (UNIT 6.2). The command-line equivalent for an NJ tree using the default settings is:

```
clustalw/INFILE=your-aligned-sequence-file/TREE
```

To use the Kimura correction, and ignore all sites with gaps, the command-line equivalent is:

```
clustalw/INFILE=your-aligned-sequence-file/TREE/  
KIMURA/TOSSGAPS.
```

The bootstrap procedure

In addition to the options that affect tree construction above, there are additional options relevant to bootstrapping.

6. Clustal stores the bootstrap values in the tree description inside square brackets, either as branch labels or as node labels.

The alternative placements are controlled by the Output Tree Format Options menu (Fig. 6.3.15). As discussed in UNIT 6.2, there is little consensus on how to store bootstrap values in tree descriptions. Widely used programs such as TreeView (UNIT 6.2) do not recognize bootstrap values stored as branch labels, and so in order to display these values in TreeView the bootstrap values must be placed on the nodes (command-line equivalent, /BOOTLABELS=node).

7. Having set the options for the bootstrap analysis and output, the command Bootstrap N-J Tree will perform the bootstrapping.

In ClustalX the user is presented with a dialog box asking for a “seed” for the random number generator used to create the bootstrap pseudoreplicates, the number of pseudoreplicates (“trials”) to generate (the default is 1000), and the name of the file to which the bootstrap tree will be written (typically the tree file is given the name of your sequence file plus the extension .phb; see Fig. 6.3.15). Click on the OK button to perform the bootstrapping. The command-line equivalent is: clustalw/INFILE=your-aligned-sequence-file/BOOTSTRAP.

GUIDELINES FOR UNDERSTANDING RESULTS

Phylogenetic trees reconstructed by distance methods do not fundamentally differ from trees reconstructed by any other approach (see UNIT 6.1). The main specificity is related to branch lengths. NJ and BIONJ can provide negative branch-length estimates, which have to be seen as null. Such negative values do not indicate any sort of “reverse evolution.” Null (or close to zero) branches indicate an irresolution of the tree, which may correspond to a multifurcation, but more likely reflects the weakness of the phylogenetic signal. WEIGHBOR consistently set the negative branches to zero, while FITCH and FastME never provide negative branches as a result of the principles upon which they are based.

The strength of the inferred branches is measured by the bootstrap procedure. Short branches are generally poorly supported, but with distance-based approaches it may happen that long branches also have a low support. The bootstrap procedure must therefore be used, which is done at low computation cost due to the speed of these approaches. The interpretation of bootstrap supports is a difficult question, but any branch with a support lower than 50% should be considered an irresolution (Berry and Gascuel, 1996).

However, in some cases wrong inferences can have high bootstrap support. For example, when very long sequences are used (as is the case when several genes are combined within the same study), bootstrapping the data does not change the resulting tree, which may be partly erroneous. The stability of the tree then has to be tested by other approaches. Notably, the tree must be robust with respect to the presence/absence of the outgroup, which possibly attracts some ingroup taxa, to model parameter variations, and to gene sampling when several genes are combined.

COMMENTARY

Background Information

The rationale of distance-based approaches

Let S be the set of sequences being studied and T the true evolutionary tree of these sequences. Assume that the sequences have been correctly aligned, so that the sites correspond to homologous positions (see *UNITS 2.1 & 2.4*). Now consider the true number of substitutions that is attached to every branch of T , i.e., the number of substitutions that occurred in the past from the sequence situated at one branch extremity to the sequence at the other extremity. These substitution numbers are unknown but well defined. They induce the evolutionary distance between any pair of taxa, as the sum of the substitution numbers attached to the path separating both taxa in T . In other words, the evolutionary distance between any pair of taxa is equal to the number of substitutions from one sequence to the other. For mathematical reasons first discovered by Zaretiskii (1965), there is an equivalence between the above-defined distance and T . Knowing T and the substitution numbers per branch allows the computation of the pairwise distances between taxa. More importantly, the true tree T and the substitution numbers per branch can be reconstructed from the matrix D of pairwise evolutionary distances.

Obviously, and unfortunately, the true number of substitutions that separates any pair of taxa is unknown. Due to hidden (parallel or convergent) mutation events, the true number of substitutions is always greater than or equal to the number of observed differences between both sequences. When the number of differences is small, both quantities are close. However, the gap increases when the evolutionary distance increases. The distance-based approach therefore involves estimating the evolutionary distance from the observed differences, assuming a stochastic model of sequence evolution. The simplest model, that of Jukes and Cantor (1969), supposes that all sites evolve independently and identically according to a Markovian process that is defined by a unique parameter representing the instantaneous probability of change from one nucleotide to another. This model establishes a mathematical relationship between the evolutionary distance (now defined as the ratio between the true number of substitutions and the sequence length) and the proportion of observed differences. More realistic models have been proposed, such as those described above

(Support Protocols 1 and 2), but the basic principle remains identical. An estimate \hat{D} of D is first computed, and then an estimate \hat{T} of T is computed using \hat{D} . The accuracy of \hat{T} increases with the reliability of \hat{D} .

The estimated evolutionary distance matrix \hat{D} no longer exactly fits a tree, but is usually very close to a tree. For example, the working data set of Figure 6.3.7 has been extracted from TreeBASE (<http://www.treebase.org/treebase/index.html>) and corresponds to 67 fungal sequences (accession no. M520). DNADIST and NEIGHBOR with default options construct a tree that explains more than 98% of the variance in the distance matrix (this value was computed by a simple program devised by one of the authors, which is not available from PHYLIP). The resulting tree and the distance matrix are thus extremely close, so the mere principle of the distance approach appears to be well founded in this case (and in most cases).

Even though the estimated distance matrix is usually very close to a tree, tree reconstruction from such an approximate matrix is much less obvious than in the ideal case where the matrix perfectly fits a tree. Various methods have been proposed, which differ according to the criterion they optimize and according to their tree-building strategy. For all known criteria, the optimization task is NP-hard (i.e., can require exponential computing time) so all practical methods are heuristic and do not guarantee that the best tree will be found. However, due to the closeness between the distance matrix and a tree, all (reasonable) methods usually find similar trees that are fairly accurate estimates of the true tree.

Neighbor-Joining algorithm

Neighbor Joining (NJ) is derived from AD-TREE (Sattath and Tversky, 1977). It was proposed by Saitou and Nei (1987) and studied in depth by several authors (Studier and Keppler, 1988; Rzhetsky and Nei, 1993; Atteson, 1997; Gascuel, 1997b; Desper and Gascuel, 2005).

NJ is an agglomerative algorithm. At each step, it uses the distance matrix $\hat{D} = \delta_{ij}$ where i and j are either taxa or clusters of taxa agglomerated during previous steps. Based on these distances, two taxa are selected to be merged. Denoting r as the number of "taxa" in \hat{D} , and Q_{ij} as the criterion value for the agglomeration

of i and j , the pair agglomerated is the one minimizing:

$$Q_{ij} = (r-2)\delta_{ij} - \Delta_i - \Delta_j, \text{ where } \Delta_x = \sum_{y=1}^r \delta_{xy}$$

Equation 6.3.1

Once the pair i, j to agglomerate is selected, NJ creates a new node u which represents the root of the new cluster. NJ then estimates the branch lengths δ_{iu} and δ_{ju} and reduces the distance matrix by replacing the distances relative to taxa i and j by those between the new node u and any other node x using:

$$\delta_{ux} = \frac{1}{2}(\delta_{ix} - \delta_{iu}) + \frac{1}{2}(\delta_{jx} - \delta_{ju})$$

Equation 6.3.2

The process stops when $r = 2$, with the last branch length being equal to the last value in the distance matrix. The successive mergings achieved by NEIGHBOR are available in its outfile.

The Q criterion enables numerous interpretations, the most popular being that it corresponds to the least-squares length estimate of the tree under construction. However, this result was fully proven only recently by the authors of this unit in Desper and Gascuel (2005), where they showed that NJ actually minimizes the balanced least-squares tree length estimate, first proposed by Pauplin (2000); see below for further discussion. Accordingly, NJ tends to produce a tree with minimal length. More importantly, when applied to any tree distance D that perfectly fits a tree T , Q designates with certainty a pair of neighbors of T . This induces the statistical consistency of NJ, which is an essential property of phylogeny reconstruction methods, i.e., NJ recovers the true tree T with certainty, as soon as \hat{D} is sufficiently close to the true evolutionary distance matrix D .

The BIONJ algorithm

The BIONJ algorithm (Gascuel, 1997a) is a variant of NJ. It is based on the fact that NJ remains consistent when its reduction formula (see Equation 6.3.2) is replaced by:

$$\delta_{ux} = \lambda_{ij}(\delta_{ix} - \delta_{iu}) + (1 - \lambda_{ij})(\delta_{jx} - \delta_{ju})$$

Equation 6.3.3

where λ_{ij} is any number in $[0,1]$ that varies depending on the merged pair i, j but not on x . So once the pair i, j has been selected, BIONJ computes the value λ_{ij}^* that minimizes the sum

of the variances of the δ_{ux} estimates. In this way, more reliable estimates will be available to select the pairs of taxa to be agglomerated during the next steps. Moreover, since the process is repeated at each step, these estimates will become better and better in comparison with NJ estimates as the algorithm proceeds.

To achieve this, BIONJ uses a simple first-order model of variances and covariances of evolutionary distance estimates obtained from sequences. This model indicates that the variance of any distance estimate δ_{xy} is approximately proportional to δ_{xy} , while the covariance of δ_{xy} and δ_{zt} is roughly proportional to the length of the intersection of paths (x,y) and (z,t) in the true tree T (Nei and Jin, 1989; Bulmer, 1991). This yields the formula:

$$\lambda_{ij}^* = \frac{1}{2} + \varphi$$

Equation 6.3.4

where φ is a correction term that depends on δ_{iu} and δ_{ju} (at least when i and j are original taxa). When δ_{iu} and δ_{ju} are equal, then $\varphi = 0$, $\lambda_{ij}^* = 1/2$, and BIONJ is equivalent to NJ. When both differ, i.e., when the substitution rates vary among lineages, φ becomes not null and places more confidence on the shorter and hence more reliable distance. So BIONJ has a clear advantage over NJ when the molecular clock is markedly violated, whereas both methods are close in the opposite case.

WEIGHBOR

WEIGHBOR follows the same agglomerative scheme as NJ. It modifies the reduction step, in a way analogous to BIONJ, but also modifies the selection step to take into account the high variance of long-distance estimates. Instead of using NJ's selection criterion (see Equation 6.3.1), WEIGHBOR combines two criteria. When i and j are neighbors in T and when \hat{D} perfectly fits T , then one has the two following properties:

Additivity:

$$\delta_{ik} - \delta_{jk} \text{ is independent of } k (\neq i, j)$$

Positivity:

$$\delta_{ik} + \delta_{jl} - \delta_{ij} - \delta_{kl} \geq 0 \text{ for any } k, l (\neq i, j)$$

Equation 6.3.5

Since \hat{D} is imperfect, these properties are only approximately satisfied, and one has to find the pair i and j that fits them best. To achieve this, WEIGHBOR assumes that distance estimates are mutually independent

and have Gaussian distribution with variance as induced by the Jukes and Cantor (1969) model. Within this model, the variance of the distance estimate is proportional to the distance around 0 (as in the BIONJ model), but increases exponentially when the distance becomes larger. This model allows one to compute the likelihood that i and j are neighbors. Considering the above defined additivity, one has the following criterion (to be minimized):

$$\text{Additivity } (i, j) = \sum_{k \neq i, j} \frac{(\delta_{ik} - \delta_{jk} - \overline{(\delta_{ik} - \delta_{jk})})^2}{\text{VAR}(\delta_{ik}) + \text{VAR}(\delta_{jk})}$$

Equation 6.3.6

where the bar denotes the average over k ($\neq i, j$). A similar criterion corresponds to the positivity property. *Additivity* is used to indicate the best pairs, which are finally selected using *Positivity*. This approach, which fully takes into account the high variance of long evolutionary distances, makes WEIGHBOR more resistant than NJ and BIONJ to the influence (attraction or distraction) of long branches.

FITCH

FITCH is the implementation (Felsenstein, 1997) of the basic principles described in the seminal paper of Fitch and Margoliash (1967). Its algorithmic strategy is not agglomerative but additive. FITCH constructs a tree by iteratively adding taxa to a growing tree. At each step, it performs tree swapping to improve the goodness-of-fit, using nearest-neighbor interchange (i.e., exchange of subtrees separated by 3 branches). Finally, once a first tree has been constructed, it optionally (see discussion of FITCH in Alternate Protocol 1) performs a more extensive search in the tree space by considering global rearrangements—every subtree is removed from the tree and put back on in all possible ways so as to have a better chance of finding a better tree. The resulting tree may be sensitive to the initial taxon ordering, even when the swapping procedures tend to lower its influence. So the jumbling procedure (Alternate Protocol 1) must be used, unless there are computational time constraints.

FITCH optimizes the weighted least-squares criterion. Let (δ_{ij}) be the matrix of distance estimates and (\hat{t}_{ij}) the distance matrix induced by the inferred tree \hat{T} and its branch lengths. The weighted least-squares fitting of

\hat{T} is defined by:

$$\text{WLS}(\hat{T}) = \sum_{i \neq j} \frac{1}{\text{VAR}[\delta_{ij}]} (\hat{t}_{ij} - \delta_{ij})^2$$

Equation 6.3.7

where $\text{VAR}[\delta_{ij}]$ is the variance of the δ_{ij} estimate. This criterion has to be minimized, and has value 0 when \hat{T} perfectly represents (δ_{ij}) . Various solutions are possible for the variance of δ_{ij} , which may be written as $\text{VAR}[\delta_{ij}] = \delta_{ij}^p$. When the power p is null, all variances are equal to 1.0, and the higher variance of long distances is not taken into account. When $p = 1$, the variance of δ_{ij} is equal to δ_{ij} , and the model is equivalent to that of BIONJ without the covariance terms. The best results, however, are obtained with $p = 2$, which corresponds to the solution of Fitch and Margoliash (1967) and is quite close to the WEIGHBOR model. This is the default option of FITCH.

The criterion in the above sum of squares equation not only concerns the topology of \hat{T} , but also its branch lengths. Minimizing this criterion induces branch length estimates which have to be positive for the approach to be consistent. This is one other default option (to be conserved) of FITCH.

FastME

FastME builds trees using the following principle. For each tree T , least-squares length estimates are assigned to each branch. Next, the sum of the branch lengths is calculated and set to the value $l(T)$. The tree minimizing $l(T)$ is chosen, which is in spirit analogous to parsimony, but this choice requires (relatively) complex mathematical explanations to be fully understood (Desper and Gascuel, 2005). FastME allows the user to search topologies when branch lengths are estimated either by ordinary least-squares (OLS) or balanced least-squares (BLS). Ordinary least-squares branch lengths are assigned according to Equation 6.3.7, with $p = 0$ and variances constant. Balanced least-squares (Pauplin, 2000; Desper and Gascuel, 2004) represents a weighted scheme per Equation 6.3.7, with:

$$\text{VAR}[\delta_{ij}] = 2^{p_{ij}}$$

Equation 6.3.8

where p_{ij} is the path length (number of branches) from taxon i to taxon j in T . In other words, variances in the balanced least-squares scheme increase exponentially as a function of the evolutionary distance, in a way similar to WEIGHBOR's.

FastME builds an initial tree additively, as does FITCH. Each taxon i is inserted optimally into the tree built on the first $(i - 1)$ taxa. Alternatively, the user can provide the initial topology, e.g., using NJ or BIONJ.

From the initial topology, FastME searches through the space of tree topologies by testing each possible nearest neighbor interchange (NNI; Fig. 6.3.16). This search can be performed quickly, as each possible NNI can be tested in constant time. While either OLS or BLS length estimates can be used, this discussion will focus on the latter, which have been demonstrated to have better statistical properties for biological data sets (Desper and Gascuel, 2004). The value of the topology change can be expressed as a linear sum of “balanced” average distances.

Balanced average distances (Pauplin, 2000) are defined with respect to a tree T . If a and b are leaves of T , $\delta_{[a]|(b)}^T = \delta_{ab}^T$, the distance from a to b in T , is defined. More generally, if A and B are the leaf sets of two disjoint subtrees of T , $\delta_{A|B}^T$ is defined recursively. Presuming that it is not the case that both A and B are singleton sets, without loss of generality there are two subtrees B_1 and B_2 that meet at an internal node to form B . Then the equation:

$$\delta_{A|B}^T = \frac{1}{2}(\delta_{A|B_1}^T + \delta_{A|B_2}^T)$$

Equation 6.3.9

is defined. Suppose $T \rightarrow T'$ is the topology transformation resulting from the NNI in Figure 6.3.16. Then:

$$l(T) - l(T') = \frac{1}{4}(\delta_{A|C}^T + \delta_{B|D}^T - \delta_{A|B}^T - \delta_{C|D}^T)$$

Equation 6.3.10

FastME first calculates the value of $\delta_{U|V}^T$ for each pair of disjoint subtrees U, V in T . Using the structure of T , this can be done in time proportional to n^2 , where n is the number of taxa. After this is done, each possible value of an NNI can be calculated using Eq. 6.3.10 in constant time. Once the optimal NNI is found, the topology is changed and the matrix of average distances is updated in time proportional to $n \text{ diam}(T)$, where $\text{diam}(T)$ is the length (number of branches) of the longest path between any taxon pair.

Desper and Gascuel (2002) and Vinh and Von Haeseler (2005) showed via simulations that BLS FastME post-processing improves the quality of the output tree when the input tree is produced by any of the major distance algorithms. Furthermore, the authors of this unit have demonstrated (Desper and Gascuel 2005) that the NJ algorithm is another type of greedy algorithm optimizing the BLS criterion, albeit with a restricted search space; Equation 6.3.1 represents the difference in BLS tree length that is obtained by agglomerating the pair i, j , and NJ greedily agglomerates taxon pairs until a fully resolved trees is obtained.

Method comparison

Numerous computer simulations have been performed to compare the topological accuracy of phylogeny reconstruction methods. The principle is: (a) consider a “true tree,” (b) evolve an initial random sequence along this tree to obtain “contemporary sequences,” (c) reconstruct a tree from these sequences, (d) finally, compare the inferred tree to the true tree. Drawing definitive conclusions from such a study is difficult because the results depend on the true tree, on the evolutionary conditions, and on numerous parameters. Moreover, numerous available studies have considered a

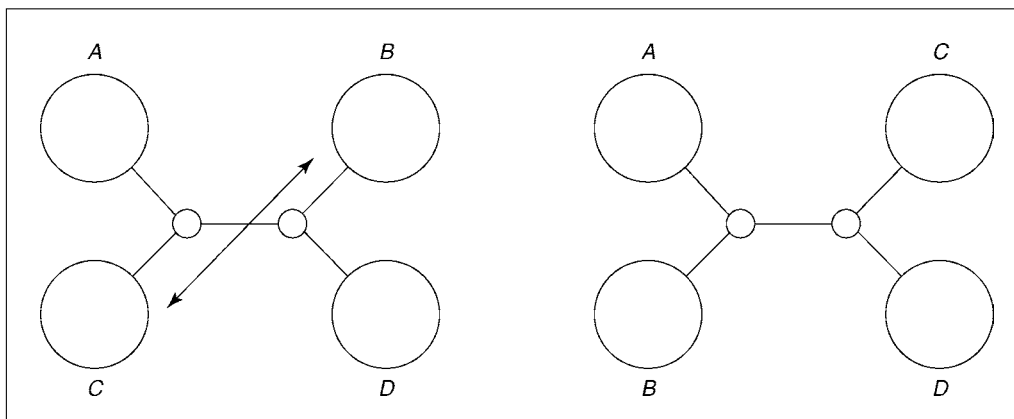


Figure 6.3.16 Nearest Neighbor Interchange (NNI) swapping subtrees B and C .

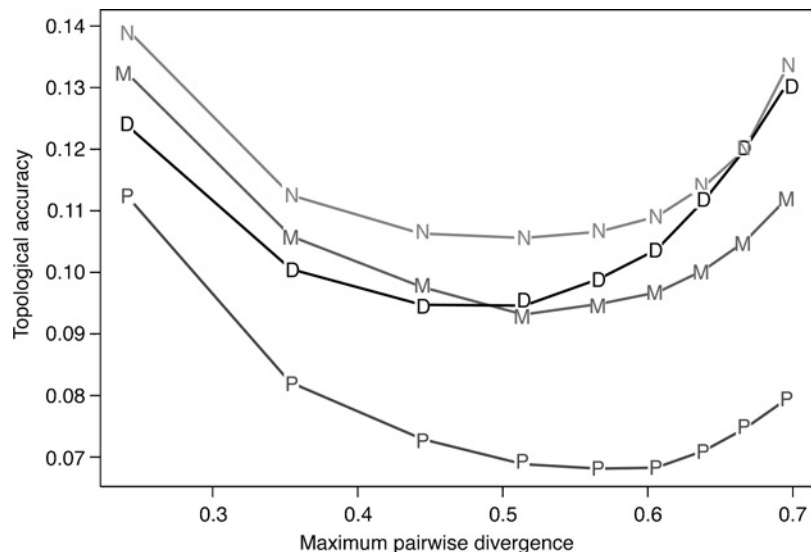


Figure 6.3.17 Topological accuracy of NEIGHBOR (N), FastME (M), DNAPARS (D) and PHYML (P) with 5000 randomly generated 40-taxon Trees. Maximum pairwise divergence is given in number of substitutions per site. Topological accuracy is measured by the number of clades in the inferred tree that are not in the correct tree, divided by the total number of clades; 0.0 corresponds to identical trees, while 1.0 means that both trees have no clade in common.

low number of true trees and few taxa (usually 12 or less).

The authors of this unit have recently tried to overcome these limits by randomly generating thousands of trees, with realistic numbers of taxa (from 24 to 100), under a broad variety of evolutionary conditions. Figure 6.3.17 displays the topological accuracy of four programs: NJ, FastME, DNAPARS (parsimony-based algorithm used with default options, from the PHYLIP package), and PHYML (Guindon and Gascuel, 2003), which is a fast maximum-likelihood algorithm. The results were obtained with 40-taxon trees generated using the standard Yule-Harding speciation process, maximum pairwise divergence uniformly drawn from [0.1,1.1], and the molecular clock varying from full satisfaction to strong violation. Figure 6.3.17 clearly shows the advantage of using FastME rather than NJ, as FastME is more accurate than NJ whatever the maximum pairwise divergence among taxa. With this data set, WEIGHBOR and FITCH are very close to FastME, while BIONJ accuracy is midway between NJ's and FastME's (results not shown). Moreover, Figure 6.3.17 gives a clear view of the relative topological accuracy of the three main approaches: distance, parsimony, and maximum-likelihood. The latter is the best method in all conditions, and the gap with the other approaches is rather impressive. DNAPARS is

globally equivalent to FastME, but its relative accuracy depends on the maximum-pairwise divergence; it performs well with low evolutionary rates, but its performance is no better than NJ's under the opposite condition.

More taxa are required to distinguish among the best distance-based methods. In Desper and Gascuel (2004), the authors of this unit generated 100-taxon trees using the Aldous distribution on trees, which generalizes the simple Yule-Harding distribution. 600-bp DNA sequences were evolved through each tree topology, with rates of evolution varying from site to site according to a covarion model (Galtier, 2001). From the resulting DNA sequences, distance matrices were calculated using the Nei and Jin (1989) estimate for gamma-distributed rates. Algorithms tested included FastME, NJ, WEIGHBOR, and the weighted least squares heuristic search of PAUP*, which is similar to FITCH but much faster. Results indicated that FastME has best topological accuracy, whatever the value of various simulation parameters related to tree shape, molecular clock violation, maximum pairwise divergence, and covarion model of sequence evolution.

Finally, very large-scale simulations with thousands of taxa were published recently by Vinh and von Haeseler (2005), to compare fast distance methods, i.e., NJ, BIONJ, BME (FastME's greedy insertion algorithm

based on balanced minimum evolution), and STC (Vinh and von Haeseler's own method based on the use of short triplets). They found that STC produced the best initial topologies for post-processing, followed by BME. More importantly, they also found that FastME post-processing (with balanced minimum evolution) improved the topological accuracy, without regard to which algorithm was used to select an initial topology. Furthermore, the accuracy was fairly uniform after post-processing without regard to the selection of the initial topology. Thus, it appears that FastME post-processing is the key step here, and that it should be used whenever large data sets are analyzed. Moreover, as the initial tree-building algorithm has low influence, using the BME algorithm for this purpose (FastME default option) is a simple, fast and accurate solution.

It can be seen that contrast between methods in Figure 6.3.17 is not very high, even when significant. The contrast between run times in Table 6.3.1 is much more impressive. NJ, BIONJ, and FastME are one order of magnitude faster than any of the other methods. They require about one-half sec-

ond to deal with 250 taxa, and Vinh and Von Haeseler (2005) showed that these three methods need only a few minutes to build a tree with 5000 taxa on a standard computer (see Howe et al., 2002, for the analysis of even larger data sets). In fact, they require much less time than DNADIST requires to compute the matrix of pairwise distances. But this latter program is rather slow, due to the use of optimization-based distance estimators, and faster analytical solutions do exist to estimate evolutionary distances. WEIGHBOR is about 500 times slower than NJ and is only applicable to data sets with up to a few hundred sequences, while FITCH is very slow, as it requires about 12 hr to analyze 250 taxa. DNAPARS is also quite slow, as it requires ~8 hr to deal with 250 taxa, but TNT (see Internet Resources) is much faster and finds better trees (24 parsimony steps, with 250 taxa) than DNAPARS. DNAML is also slow, even when using the speedier option (as performed by the authors of this unit), and requires about 100 min to deal with 250 taxa; moreover, its results in terms of likelihood are rather poor as compared to PHYML (DNAML tree is about 500 log likelihood points below PHYML's, with 250 taxa). Finally, PHYML is

Table 6.3.1 Run Times for Various Tree-Building Methods

Method ^{a,b,c}	Run time (sec)		
	Number of taxa		
	40	100	250
DNADIST	0.09	0.65	25
FastME	0.008	0.055	0.34
NJ	0.0045	0.035	0.25
BIONJ	0.0052	0.055	0.60
WEIGHBOR	1.1	18	255
FITCH	6	335	43,200
DNAPARS	6	230	30,000
TNT	5	13	330
DNAML	26	186	6,000
PHYML	7.5	20	390

^aDistance estimation: DNADIST; distance-based tree building methods, FastME, NJ, BIONJ, WEIGHBOR, and FITCH; parsimony methods, DNAPARS and TNT; maximum-likelihood methods, DNAML and PHYML.

^bTNT was run with 500 (40 and 100 taxa) or 50 (250 taxa) starting points and TBR option. All other programs were run with default options and K2P model of sequence evolution. 40-taxon and 100-taxon (simulated) data sets were taken from (Guindon and Gascuel, 2003); results were averaged over 5000 and 30 data sets, respectively. The 250-taxon (biological) data set was used in (Stamatakis et al., 2004) to compare maximum-likelihood programs.

^cRun times are in seconds with a Windows PC 1.8 MHz and 1.0 Gb RAM.

rather efficient as it requires about 6 min with 250 taxa; it can be used up to 500 taxa and has very good topological accuracy (Figure 6.3.17; Guindon and Gascuel, 2003).

These results show that, with very large data sets, e.g., >1000 taxa, the distance approach is the only one to be applicable, as soon as fast algorithms are used, i.e., NJ, BIONJ, or FastME. FastME should be preferred, as it clearly has the best topological accuracy among the distance methods, while NJ is worse and BIONJ is in between (Vinh and Von Haeseler, 2005). With large data sets involving a few hundred taxa, fast parsimony (e.g., TNT) and maximum-likelihood (e.g., PHYML) programs become applicable, and the latter should be preferred over any other approach, as it has the best topological accuracy. However, distance methods are still of interest for fast exploratory study, and to perform bootstrap analysis, which is very demanding in terms of computing time. Finally, with moderate data sets, e.g., <100 taxa, all methods are applicable; the use of distance methods should then be limited to fast analysis, and maximum-likelihood should be employed in other cases.

Critical Parameters and Troubleshooting

Distance-based approaches are sensitive to the way evolutionary distances are estimated. When the sequences exhibit few differences, all sequence evolution models become equivalent, and the model choice is not crucial. For example, when two sequences have 0.1 sites that differ with 0.07 transitions and 0.03 transversions, the Jukes and Cantor distance estimate is equal to 0.1073, the Kimura two-parameter estimate is 0.1086, and the Jin and Nei estimate (with $\alpha = 1.0$) is 0.1183. Distance estimation, however, becomes very sensitive to the model choice when the maximum pairwise divergence among the sequences increases. For example, consider two sequences with half of the sites being different, with 0.35 transitions and 0.15 transversions, the Jukes and Cantor estimate is equal to 0.824, the Kimura two-parameter estimate is 1.037, and the Jin and Nei estimate ($\alpha = 1.0$) is 2.940. Therefore, data sets where the sequence divergence is too high (say >1.0) must be considered suspicious and should be discarded. Note that the presence of such high divergence makes the alignment itself very difficult and prone to errors. With more reasonable maximum divergence, the stability of the results despite model variations is a positive point. Moreover, the presence of distant outgroup

taxa is a perturbation factor in all reconstruction steps (alignment, distance estimation, and tree building) and should be avoided, at least in a first analysis.

Suggestions for Further Analysis

Distance methods are available in numerous phylogeny software packages. Notably, PAUP* (release 4.0b10; UNIT 6.4) provides very fast versions of NJ, FITCH, and BIONJ, as well as a larger variety of evolutionary distance estimates than that provided by DNADIST and PROTDIST. DAMBE (Xia et al., 2001) also contains an implementation of FastME and a complete environment to perform bootstrap, drawing the trees and computing the distances.

Parsimony approaches do not outperform distance methods (see Fig. 6.3.17), but their principle is so different that finding the same tree using both is generally considered to be a strong support for that tree. PAUP* (UNIT 6.4) and TNT (see Internet Resources) provide fast parsimony implementations.

PHYML (Guindon and Gascuel, 2003) is a fast and accurate maximum likelihood software (see Fig. 6.3.17 and Table 6.3.1), which should be preferred over other approaches with data sets that are not too large (<500 taxa). PHYML is freely downloadable at <http://atgc.lirmm.fr/phyml/>, where a Web server is also available.

Literature Cited

- Atteson, K. 1997. The performance of the NJ method of phylogeny reconstruction. *In* Mathematical Hierarchies and Biology (B. Mirkin, F.R. McMorris, F.S. Roberts, and A. Rzhetsky, eds.) pp.133-148. American Mathematical Society, Providence, R.I.
- Berry, V. and Gascuel, O. 1996. Interpretation of bootstrap trees: Threshold of clade selection and induced gain. *Mol. Biol. Evol.* 13:999-1011.
- Bruno, W.J., Socci, N.D., and Halpern, A.L. 2000. Weighted neighbor joining: A likelihood-based approach to distance-based phylogeny reconstruction. *Mol. Biol. Evol.* 17:189-197.
- Bulmer, M. 1991. Use of the method of generalized least squares in reconstructing phylogenies from sequence data. *Mol. Biol. Evol.* 8:868-883.
- Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. 1979. A model for evolutionary change in proteins. *In* Atlas of Protein Sequence and Structure (M.O. Dayhoff, ed.), vol. 5, pp. 345-352.
- Desper, R. and Gascuel, O. 2002. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *J. Comp. Biol.* 9:687-705.
- Desper, R. and Gascuel, O. 2004. Theoretical foundation of the balanced minimum evolution method of phylogenetic inference and its relationship to weighted least-squares tree fitting. *Mol. Biol. Evol.* 21:587-598.

- Desper, R. and Gascuel, O. 2005. The minimum evolution distance-based approach to phylogenetic inference. *In* Mathematics of Evolution and Phylogeny (O. Gascuel, ed.) pp. 1-32. Oxford University Press, Oxford.
- Felsenstein, J. 1989. PHYLIP—Phylogeny inference package (version 3.2). *Cladistics* 5:164-166.
- Felsenstein, J. 1997. An alternating least-squares approach to inferring phylogenies from pairwise distances. *Syst. Biol.* 46:101-111.
- Felsenstein, J. and Churchill, G.A. 1996. A hidden Markov model approach to variation among sites in rate of evolution *Mol. Biol. Evol.* 13: 93-104
- Fitch, W.M. and Margoliash, E. 1967. Construction of phylogenetic trees. *Science* 155:279-284.
- Galtier, N. 2001. Maximum-likelihood phylogenetic analysis under a covarion-like model. *Mol. Biol. Evol.* 18:866-873.
- Gascuel, O. 1997a. BIONJ: An improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.* 14:685-695.
- Gascuel, O. 1997b. Concerning the NJ algorithm and its unweighted version, UNJ. *Mathematical Hierarchies and Biology* (B. Mirkin, F.R. McMorris, F.S. Roberts, and A. Rzhetsky, eds.) pp. 149-170. American Mathematical Society, Providence, R.I.
- Graur, D. and Li, W.-H. 2000. *Fundamentals of Molecular Evolution*. Sinauer Associates, Sunderland, Mass.
- Guindon, S. and Gascuel, O. 2002. Efficient biased estimation of evolutionary distances when substitution rates vary across sites. *Mol. Biol. Evol.* 19:534-543.
- Guindon, S. and Gascuel, O. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.* 52:696-704.
- Howe, K., Bateman, A., and Durbin, R. 2002. QuickTree: Building huge Neighbour-Joining trees of protein sequences. *Bioinformatics* 18:1546-1547.
- Jin, L. and Nei, M. 1990. Limitations of the evolutionary parsimony method of phylogenetic analysis. *Mol. Biol. Evol.* 7:82-102.
- Jones, D.T., Taylor, W.R., and Thornton, J.M. 1992. The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.* 8:275-82.
- Jukes, T.H. and Cantor, C.R. 1969. Evolution of protein molecules. *Mammalian Protein Metabolism* (H. N. Munro, ed.) pp.21-132. Academic Press, New York.
- Kimura, M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.* 16:111-120.
- Kimura, M. 1983. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge, U.K.
- Kishino, H. and Hasegawa, M. 1989. Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in *Homoidea*. *J. Mol. Evol.* 29:170-179.
- Nei, M. and Jin, L. 1989. Variances of the average numbers of nucleotide substitutions within and between populations. *Mol. Biol. Evol.* 6:290-300.
- Page, R.D.M. and Holmes, E.C. 1998. *Molecular Evolution: A Phylogenetic Approach*. Blackwell Scientific, Oxford, U.K.
- Pauplin, Y. 2000. Direct calculation of a tree length using a distance matrix. *J. Mol. Evol.* 51:41-47.
- Perrière, G. and Gouy, M. 1996. WWW-Query: An on-line retrieval system for biological sequence banks. *Biochimie* 78:364-369.
- Rzhetsky, A. and Nei, M. 1993. Theoretical foundation of the minimum-evolution method of phylogenetic inference. *Mol. Biol. Evol.* 10:1073-1095.
- Saitou, N. and Nei, M. 1987. The neighbor-joining method: A new method for reconstruction of phylogenetic trees. *Mol. Biol. Evol.* 4:406-425.
- Sattath, S. and Tversky, A. 1977. Additive similarity trees. *Psychometrika* 42:319-345.
- Sokal, R.R. and Michener, C.D. 1958. A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull.* 38:1409-1438.
- Stamatakis, A., Ludwig, T., and Meier, H. 2004. RAxML-III: A fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics* 21:456-463.
- Steel, M.A. 1994. Recovering a tree from the Markov leaf colourations it generates under a Markov model. *Appl. Math. Lett.* 7:19-23.
- Studier, J.A. and Keppler, K.J. 1988. A note on the neighbor-joining algorithm of Saitou and Nei. *Mol. Biol. Evol.* 5:729-31.
- Swofford, D.L., Olsen, G.L., Waddell, P.J., and Hillis, D.M. 1996. Phylogenetic inference. *In* *Molecular Systematics* (D.M. Hillis, C. Moritz, and B.K. Mable, eds.) pp. 407-514. Sinauer Associates, Sunderland, Mass.
- Thompson, J.D., Gibson, T.J., Plewniak, F., Jeanmougin, F., and Higgins, D.G. 1997. The ClustalX windows interface: Flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Res.* 25:4876-4882.
- Veerassamy, S., Smith, A., and Tillier, E.R. 2003. A transition probability model for amino acid substitutions from blocks. *J. Comput Biol.* 10:997-1010.
- Vinh, L.S. and von Haeseler, A. 2005. Shortest triplet clustering: Reconstructing large phylogenies using representative sets. *BMC Bioinformatics* 6:92.
- Xia, X. and Xie, Z. 2001. DAMBE: Data analysis in molecular biology and evolution. *J. Hered.* 92:371-373.

Yang, Z. 1996. Among-site rate variation and its impact on phylogenetic analyses. *TREE* 11:367-372.

Zaretzkii, K. 1965. Reconstructing a tree from the distances between its leaves. *Uspehi Matematicheskikh Nauk* 20:90-92 (in Russian).

Internet Resources

<http://atgc.lirmm.fr/fastme>

This web page from the authors provides FastME C source code and binaries for Windows, MAC OS and LINUX, as well as several papers to understand in depth the minimum evolution principle, its algorithms and its properties.

<http://atgc.lirmm.fr/phyml>

This web page provides PHYML binaries for Windows, MAC OS and LINUX, and a web server to run PHYML online.

<http://www.cladistics.com/webtnt.html>

Goloboff, P., Farris, S., and Nixon, K. 2000. TNT: Tree analysis using new technology. Beta version, published by the authors, Tucumán, Argentina.

<http://evolution.genetics.washington.edu/phylip/software.html>

Joe Felsenstein's Web page, containing an extensive list of phylogeny software programs, including numerous distance-based methods.

Contributed by Richard Desper
Department of Biology
University College
London, United Kingdom

Olivier Gascuel
Equipe "Méthodes et Algorithmes pour la
Bioinformatique"
LRMM-CNRS
Montpellier, France