METREE: a program package for inferring and testing minimum-evolution trees

Andrey Rzhetsky and Masatoshi Nei

Abstract
The METREE program package for estimating phylogenetic trees with the minimum evolution method is written in Turbo C 2.0 and is intended to be used on any IBM-compatible personal computers that have a mathematical coprocessor. The package is simple to use and is menu driven. A program for visualizing and printing out the final tree is also included.

Introduction
The METREE program package is intended to find the minimum evolution (ME) tree that has the smallest value of the sum of branch lengths ($S$) for a set of sequences, identify a set of trees whose $S$ are not significantly different from that of the ME tree, and print out the resulting trees in a publishable form. The branch lengths for each topology are estimated by the ordinary least-squares method from a matrix of evolutionary distances. The algorithms for finding and testing the minimum evolution tree have been described by Rzhetsky and Nei (1992, 1993).

Terminology


Temporary minimum evolution tree. In the search for the ME tree the program may identify a tree with the minimum $S$ value for a subset of topologies. Until a new tree with a smaller $S$ value is found, this tree is called ‘a temporary ME tree’.

$S_T$ or $S(T)$. The sum of branch lengths for the temporary ME tree computed with the ordinary least-squares method.

$D$, $sd(D)$. $D$ is the difference $S - S_T$, where $S$ is the sum of branch lengths computed for a topology under consideration, and $sd(D)$ is the standard error of $D$.

Topological distance. Topological distance between two trees as measured by the ‘partition metric’ (see Penny and Hendy, 1985; Rzhetsky and Nei, 1992).

Files on the METREE diskette
The files included in the diskette are as follows.

- TETRAPOD.SET—an example input file;
- ME_TREE.EXE, LS_2.EXE, INVERT.EXE—a set of tree-making programs;
- TREEVIEW.EXE, TREENSHOW.EXE, TREEVIEW.DOC—programs for visualizing trees and a manual for the TREEVIEW program. These two programs were written and kindly provided by Koichiro Tamura.

Input data format

Distance matrix as input data
The input file with a distance matrix should have name extension ‘DIS’ (e.g. CRAB.DIS or TETRAPOD.DIS) and contain the following information:

| LINE #1 | Title. |
| LINE #2 | The number of sequences, blank symbol, sequence length (number of nucleotides or amino acids used per sequence). |
| LINE #3 | First species name. |
| LINE #4 | Second species name. |

...  

| LINE #(N + 2) | Last species name. |
| LINE #(N + 3) | Distance between species 2 and 1, comma. |
| LINE #(N + 4) | Distances between species 3 and 1 (followed by comma), and species 3 and 2 (followed by comma), respectively. |

...  

| LINE #(2N + 2) | All distances between species $N$ and all other species (each distance followed by comma). |

See example file TETRAPOD.DIS.
Sequences as input data

The input file with sequence data should have name extension 'SET' (e.g. TETRAPOD.SET) and contain the following information:

LINE #1 Title.
LINE #2 The number of sequences, blank symbol, sequence length.
LINE #3 First sequence name.
LINE #4 Second sequence name.
...
LINE #\((N + 2)\) Last sequence name.
LINE #\((N + 3)\) Sequences start from this line in an arbitrary format (end-of-line and blank symbols are ignored). Sequences could be either in uppercase or lowercase letters. For nucleotide sequences, '-' stands for gap, 'N', '+' or '?' stand for ambiguous or missing data. For amino acid sequences the use of 'N' for ambiguous data is not allowed.

See example file TETRAPOD.SET.

Output data format

After running the programs, you may find a number of new files in your directory. The following is a brief description of these files.

'TRE'-files (e.g. TETR0000.TRE or TETR0001.TRE). A file with this name extension contains detailed information about one of the trees. A TRE file has the following format:

1. The number of sequences.
2. Sequence names.
3. Tree topology and branch length estimates. Normally, the standard errors and significance levels of branch lengths are also included. In the case of 'Quick search for the ME tree', however, these quantities are not computed. In this case zero standard errors are written in the file.
4. Distance measure used for computing evolutionary distances.
5. Sum of branch lengths, S (for temporary ME tree), or \(D, sd(D)\) and the significance level according to the \(t\)-test with an infinite number of degrees of freedom.
6. Tree topology defined by partition codes.

TRE files are used as input files for programs TREEVIEW and TREESHOW and are not intended for manual editing. You may need to look through these files only to find the \(S\) value for a particular tree or the distance measure used.

Files with name extensions '000', '002', '004', '006', etc. These files contain a list of topologies already examined. For example, file TETRAPOD.000 contains the topology of the temporary ME tree for the TETRAPOD.SET data file, TETRAPOD.006 contains the list of all topologies that have topological distance 6 from the temporary ME tree. The topologies of the trees are stored in 'partition codes' and are not intended for manual editing.

File LS.TMP contains the variance–covariance matrix in binary format.

In files with name extension 'ME' (e.g. TETRAPOD.ME) the topology of the temporary ME tree is stored.

Starting the program

To start the program, you should perform the following operations. Make directory 'ME' (or with any other name) on the hard drive. Copy all files from the diskette into this directory. Arrange the sequence data into a 'SET' file as described above. Then, type ME_TREE, strike 'Enter' and respond to queries.

The first menu to appear looks as follows.

Your choices are:
Study minimum evolution tree
Display already saved trees
View saved trees showing standard errors of branch lengths
Quit

If you select the first line of the menu, program LS_2.EXE will be started. Selection of the second and third lines invokes TREEVIEW.EXE and TREESHOW.EXE programs respectively (see the enclosed manual on TREEVIEW.EXE written by Koichiro Tamura). If the first item of the first menu is chosen, the following prompt appears.

Do you want to start with
Distance matrix
Sequence data

This prompt allows you to choose either distance matrix data or sequence data input file type (see 'Input format' section above). After you make this decision, another menu appears.

Your choices are
Compute NJ tree with standard errors of branch lengths
Quickly search for ME tree (no standard errors of branch lengths)
Search for ME tree (slower, computing s.e.'s of branch lengths)
Search for ME tree and alternative trees
Calculate standard errors of branch lengths for the final ME tree
Quit the program

Compute NJ tree with standard errors of branch lengths
This procedure allows you to compute an NJ tree and then estimate branch lengths with the ordinary least-squares method. You can repeat this procedure several times for different distance measures. The trees produced will be saved in different files with name extension ‘TRE’. Note that the topology of the NJ tree is often identical with that of the ME tree unless the number of sequences is very large (Rzhetsky and Nei 1992, 1993).

Quickly search for ME tree
To speed up the search, no covariance matrix and no standard errors of branch lengths are computed in this case. The program simply examines different tree topologies starting from the NJ-tree with the aim of finding the ME tree. Only temporary and the final ME trees are saved.

Search for ME tree (computing s.e.’s of branch lengths)
In this case the standard errors of branch lengths are computed for each temporary ME tree. This may slow down the search considerably if the number of temporary ME trees is large. With respect to other aspects, this option is identical with the previous one.

Search for ME tree and alternative trees
In this case $sd(D)$ is computed for each alternative topology tested. If the value of $D$ is not significantly different from 0 (with default significance level 95%), the alternative tree is saved into the ‘TRE’ file.

Calculate standard errors of branch lengths for the final ME tree
If you find the ME tree with the ‘Quick search’ procedure, you may use this option to compute the standard errors of branch lengths. This option also can be used to recompute branch lengths and the confidence probability values for each interior branch for the final ME tree using different distance measures.

![Block-scheme of the current algorithm used in the search of the ME tree.](image-url)
Quit the program
The meaning is obvious.

Generating alternative trees
There are two ways of finding alternative trees in this package. One is to use the topological distance as described by Rzhetsky and Nei (1992, 1993) and the other is to use a bootstrap method.

In the case of the ‘topological distance’ method all trees with \( d_T = 2 \) and \( d_T = 4 \) in the neighborhood of the temporary ME tree are examined. If one of these trees has an \( S \) value smaller than those for the temporary ME tree, this tree becomes the next temporary ME tree. The process is repeated until no trees with a smaller \( S \) values are found for the trees with \( d_T = 2 \) and \( d_T = 4 \) in the neighborhood of the temporary ME tree.

In the case of the bootstrap method, bootstrapping is used as a tool to generate tree topologies around the temporary ME tree rather than to study the reliability of a particular topology. The \( S \) value is computed for each bootstrap-generated topology by using the entire original data set, and this \( S \) is compared with that of the temporary ME tree.

Our bootstrap method is different from the standard one and uses a scaling factor, \( f \), to increase or decrease the expected topological distance between the temporary ME tree and a bootstrap-generated topology. This \( f \) may be either \( < 1 \) or \( > 1 \). Suppose that we need to analyze a set of sequences where \( M \) homologous sites are available for comparison. In our bootstrap method one should just pick up \( M \cdot f \) sites at random (where some sites could be sampled more than once) and compute the distance matrix between the sequences using only these \( M \cdot f \) sites. Then the resulting distance matrix is used to construct the NJ tree (see Saitou and Nei, 1987). Obviously, when \( f = 1 \) we have the standard bootstrap method. When \( f \) is less than 1, the distance matrix obtained tends to be more erratic than in the case of the standard bootstrap. Thus, bootstrap-generated trees tend to have a larger average topological distance from the ME tree as \( f \) becomes smaller. In contrast, when \( f \) is greater than 1, the distances between the sequences tend to be more stable than in the usual bootstrapping. Therefore, a set of topologies closer to the temporary ME tree are examined.

Depending on the data set analyzed, it might be reasonable to choose either \( f \) greater than 1 or \( f \) less than 1. For a relatively small number of sequences, the use of bootstrapping with a small \( f \) may generate virtually all plausible topologies for a given data set and find all topologies that are ‘good candidates’ for being the true tree. For a large number of short sequences (which usually provide a very unstable distance matrix) the bootstrap technique with \( f > 1 \) may facilitate the search for the ME tree (which is typically located in a close topological neighborhood of the NJ tree or coincides with the NJ tree).

System requirements and ‘licence agreement’
Programs ME_TREE, LS_2 and INVERT were written for IBM-compatible computers with Intel 80286 processors and 80287 coprocessors (or more advanced). The programming language used is Borland Turbo C (v. 2.0). The minimum system requirements for the package are dictated by size of the data set that one intends to study. For a small number of sequences (up to 15) a relatively slow AT-type computer can be used. However, our programs are able to handle large data sets (> 100 sequences). In the latter case a quicker computer (with a 386/387 or 486 CPU) with a sufficiently large hard drive (> 40 Mbytes) is recommended. Generally, we do not advise starting these programs from a floppy disk, even for a small data set, since the computation becomes extremely slow in this case and there is always some danger of running out of disk space. The minimum size of RAM required is 640 K.

You are free to distribute this program package as long as no fee is charged.

Acknowledgements
We thank Koichiro Tamura for providing programs TREEVIEW and TREESHOW. We would appreciate any comments on the performance and possible improvements of the programs from the users. This work was supported by NIH and NSF grants to M.N.

References

Received November 1, 1993; accepted February 8, 1994