[21] Progressive Alignment of Amino Acid Sequences and Construction of Phylogenetic Trees from Them

By DA-FEI FENG and RUSSELL F. DOOLITTLE

Introduction

In 1970, Needleman and Wunsch published an elegant algorithm for optimally aligning two protein sequences.[1] Moreover, the scheme could be used with weighting scales that assigned scores to each set of paired residues and assessed penalties for unpaired ones (gaps). Arguably, the most popular of these amino acid substitution matrices was the PAM scale devised by Dayhoff and co-workers, which was based on observed mutations for sets of closely related protein sequences.[2,3] During the 1970s and early 1980s, numerous investigators used the Needleman–Wunsch algorithm in conjunction with various versions of the Dayhoff PAM matrix to generate binary (pairwise) alignments. On the other hand, multisequence alignments were usually made manually, the binary alignments serving as a guide.

Although in principle the Needleman and Wunsch approach could be extended to multiple sequences, in practice the computational time for such a task proved prohibitively long, the memory required for storing the necessary arrays being enormous even for sets of very short sequences.[4,5] Although global methods were devised that yielded reasonable alignments on longer sequences, they were still restricted to five or fewer sequences.[6] An even more disappointing aspect was that the pattern of gaps in multiple alignments was often inconsistent with what was observed in binary alignments; thus, the optimal alignment between the two closest sequences as indicated by a binary alignment was often altered in the presence of a third or fourth sequence.[6]

[1] S. B. Needleman and C. D. Wunsch, *J. Mol. Biol.* **48,** 443 (1970).

[2] M. O. Dayhoff, R. V. Eck, and C. M. Park, *in* "Atlas of Protein Sequence and Structure" (M. O. Dayhoff, ed.), Vol. 5, p 89. National Biomedical Research Foundation, Washington, D.C., 1972.

[3] R. M. Schwartz and M. O. Dayhoff, *in* "Atlas of Protein Sequence and Structure" (M. O. Dayhoff, ed.), Vol. 5, Suppl. 3, p. 353. National Biomedical Research Foundation, Washington, D.C., 1978.

[4] R. A. Jue, N. W. Woodbury, and R. F. Doolittle, *J. Mol. Evol.* **15,** 129 (1979).

[5] M. Murata, J. S. Richardson, and J. L. Sussman, *Proc. Natl. Acad. Sci. U.S.A.* **82,** 3073 (1985).

[6] M. S. Johnson and R. F. Doolittle, *J. Mol. Evol.* **23,** 267 (1986).

In 1987, we reported a simple progressive alignment procedure that circumvented the need for aligning sequences exhaustively.[7] Like other methods introduced at about the same time,[8,9] it actually used a binary alignment algorithm[1] iteratively, first to align the two most closely related sequences and then to align the next most similar one to those, etc. The strategy was further guided by the simple rule "once a gap, always a gap," the rationale being that the positions and lengths of gaps introduced between the more similar pairs of sequences should not be affected by distantly related ones.[7] As such, sequences were compared in an approximately descending order of similarity, each new overall alignment score being determined by comparisons of residues of the last added sequence against the averaged scores of all previously aligned residues. The program was written in such a way that the order could be refined automatically by checking the next two sequences at each round, priority being determined on the basis of the higher similarity score. This operation was continued until all the sequences were aligned. We referred to this procedure as progressive alignment. Because progressive alignment is not exhaustive, the required computational time is short compared with global alignment procedures.

Progressive Alignment and Phylogenetic Trees

Although many investigators are only interested in alignments per se, our interest has been driven by a need for the automatic construction of sequence-based phylogenetic trees. Historically, the first step in constructing a quantitative phylogenetic tree has always been to find an objective procedure for obtaining a multiple alignment. In our 1987 paper[7] we presented a suite of programs that allowed users to go directly from the aligned sequences to a phylogenetic tree, a number of different programs being used sequentially. Some steps had to be performed by hand, however, and we later described a version in which many of the steps were tied together in a more user-friendly fashion.[10] Even then, the system still had some annoying limitations, however. In this chapter we describe a much improved, easier to use, set of programs, which we refer to as ProPack: a packet of programs centering around progressive alignment (Table I). The changes pertain mainly to speeding up the calculations and reducing the number of manual operations; the basic idea of progressive alignment remains the same.

[7] D. F. Feng and R. F. Doolittle, *J. Mol. Evol.* **25,** 351 (1987).
[8] W. R. Taylor, *CABIOS* **3,** 81 (1987).
[9] G. J. Barton and M. J. E. Sternberg, *J. Mol. Biol.* **198,** 327 (1987).
[10] D. F. Feng and R. F. Doolittle, this series, Vol. 183, p. 375.

TABLE I
PROPACK SUITE OF PROGRAMS FOR PROGRESSIVE ALIGNMENT AND MAKING TREES

Program	Objective
FORMAT	Convert sequences to Old Atlas format
INSPECT	Find matching boundaries
CROP	Cut sequences to order
ARRANGE	Determine an approximate branching order
PREALIGN	Prealign cluster of sequences
ALIGN	Determine multiple alignment only
MULPUB	Change aligned sequences to compact format
TREE	Make a multiple alignment and construct a phylogenetic tree from it
BLEN	Determine branch lengths based on distance matrix
NONEG	Find best branching order with no negative branch lengths
SETREE	Convert information to a format recognized by tree drawing program
BTREE	Sample multiple alignments and generate new trees for bootstrapping procedure
CLUS	Analyze clusters (nodes) from BTREE for agreement with initial tree

Improvements

The principal new features are (a) introduction of an option for choosing an alternative amino acid substitution matrix, (b) automatic elimination of negative branch lengths in the calculation of phylogenetic trees, (c) addition of a bootstrap analysis option, and (d) inclusion of a simple program for converting output to a form that can be used to draw trees on a microcomputer with standard software. Additionally, experience has shown that the prealignment of subclusters is seldom necessary, and as a result the program PREALIGN is downplayed in the new ensemble. Instead, sequences can be input in any order, with the program ARRANGE automatically sorting them into an approximate order for processing by the main program TREE. A summary of these programs is presented in Table I, and a flowchart of the new scheme is depicted in Fig. 1.

Programs

The computer programs described here are for protein sequence comparisons (not DNA or RNA). They are written in the C language[11] and run under the UNIX operating environment, although they have also been used successfully by others with a VMS system after minor modifications.

[11] B. W. Kernighan and D. M. Ritchie, "The C Programming Language." Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

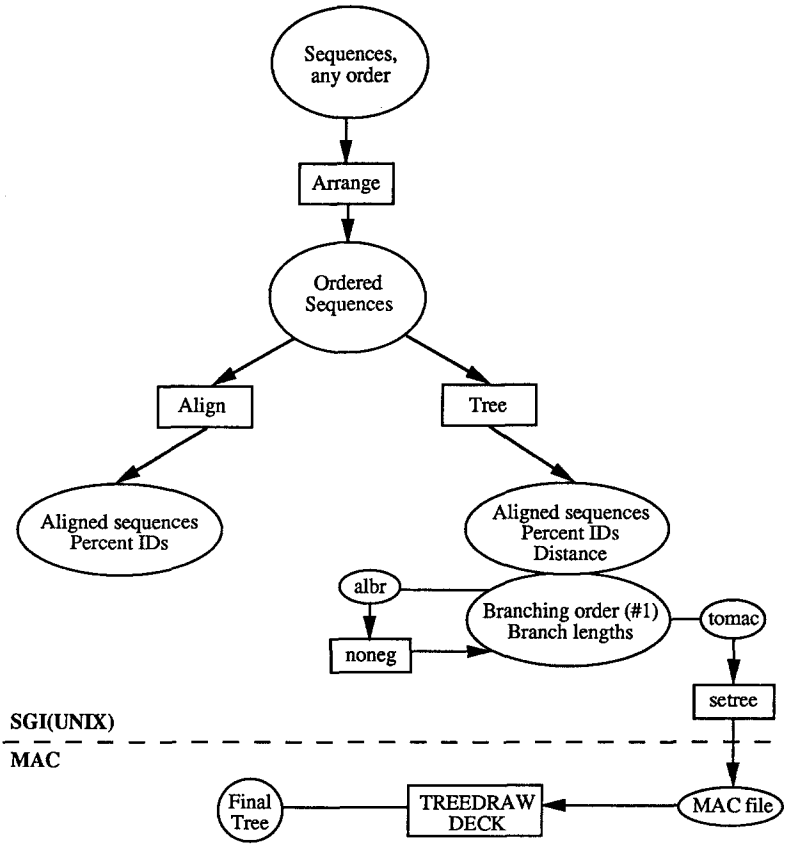

FIG. 1. Flow diagram of progressive alignment and phylogenetic tree construction. The names in ovals denote files; square boxes represent programs. All programs shown above the dashed line run under the UNIX operating system via a command line. A description of the command line format for each of the programs can be obtained simply by typing the name of the program after these programs have been compiled on the home computer.

In their present form they are exactly adapted for running on Silicon Graphics machines (IRIX 4.0.5 IOP).

The primary program TREE, which produces the multiple alignment, the branching order, and the branch lengths, can handle as many as 52 sequences (a number that is incidental to the number of lowercase and uppercase letters that are used as designation for entries). The maximum lengths of the sequences are strictly a function of the available computer memory. At the present time, sequences of 1000 residues can be aligned easily. The sequences must be provided in the "Old Atlas" format (Fig.

```
TDEHU L-lactate dehydrogenase (EC 1.1.1.27) chain M - human
PDEHU   1 K I T V V G V G A V G M A C A I S I L M K D L A D E L A L V
PDEHU  31 D V I E D K L K G E M M D L Q H G S L F L R T P K I V S G K
PDEHU  61 D Y N V T A N S K L V I I T A G A R Q Q E G E S R L N L V Q
PDEHU  91 R N V N I F K F I I P N V V K Y S P N C K L L I V S N P V D
PDEHU 121 I L T Y V A W K I S G F P K N R V I G S G C N L D S A R F R
PDEHU 151 Y L M G E R L G V H P L S C H G W V L G E H G D S S V P V W
PDEHU 181 S G M N V A G V S L K T L H P D L G T D K D K E Q W K E V H
PDEHU 211 K Q V V E S A Y E V I K L K G Y T S W A I G L S V A D L A E
PDEHU 241 S I M K N L R R V H P V S T M I K G L Y G I K D D V F L S V
PDEHU 271 P C I L G Q N G I S D L V K V T L T S E E E A R L K K S A D
PDEHU 301 T L W G I Q *
```

FIG. 2. Example of Old Atlas format.

2). The program FORMAT changes sequences into the proper form so that they are recognized by all the programs.

Although program names are capitalized throughout this chapter in order to emphasize what is actually typed, on a day-to-day basis we ordinarily use lowercase. It must be emphasized that UNIX is rigorously case sensitive.

Converting Similarity to Distance

Similarity scores for pairs of aligned sequences are converted to difference scores by the following version of the Poisson equation:[12]

$$D = -\ln S \tag{1}$$

where

$$S = [S_{\text{real}}(ij) - S_{\text{rand}}(ij)]/[S_{\text{iden}}(ij) - S_{\text{rand}}(ij)] \times 100 \tag{2}$$

$S_{\text{real}}(ij)$ is the observed similarity score for the two sequences being aligned, and $S_{\text{iden}}(ij)$ is the average of the two scores for the two sequences compared with themselves. $S_{\text{rand}}(ij)$ provides a measure of the background noise between sequences i and j. Previously $S_{\text{rand}}(ij)$ was determined by a computer-intensive shuffling operation; now it is calculated by a formula, as discussed below.

S_{rand}

The random score between two optimally aligned sequences, $S_{\text{rand}}(ij)$, is a function of the composition of the sequences, the number of internal gaps in the multiple alignment, the gap penalty, and the scoring matrix used. To put this on a mathematical basis, the following conditions obtained:

[12] D. F. Feng, M. S. Johnson, and R. F. Doolittle, *J. Mol. Evol.* **21,** 112 (1985).

a_i, b_j represent the residue type in sequence i and j, $N_a(i)$, $N_b(j)$ are the number of times a_i, b_j appear in sequences i and j, N_g is the number of internal gaps unique to either sequence i or j, irrespective of their lengths, *pen* is the gap penalty used in the calculation, and L is the overall length of the sequences after they are aligned. Now, if we avail ourselves of a suitable amino acid substitution matrix M, then

$$S_{\text{rand}} = (1/L) \Sigma \Sigma M(a_i, a_j) N_a(i) N_b(j) - N_g * pen \quad (3)$$

Equation (3) allows S_{rand} to be calculated independently of a random number generator; not only is the process speeded up, but it is more consistent.

Outline of Operations

In essence, there are 10 steps to making a sequence-based tree with our programs: (a) gather the sequences, (b) format and otherwise edit the sequences, (c) catenate the sequences into a single file, (d) arrange the sequences into an appropriate order, (e) align the sequences, (f) calculate the similarity scores and pairwise distances, (g) make a distance (difference) matrix, (h) determine the branching order and branch lengths, (i) if necessary, find an alternate solution without negative branch lengths, and (j) draw a tree. One also has the option of performing a bootstrap analysis, and we comment further on this subsequently.

Input

Gathering the sequences from appropriate databases is a straightforward matter. To put them into a suitable format for our programs, one must first strip out all nonsequence alphabetic characters, including the title (numbers will be ignored).

FORMAT temp >seq1

The program will then ask for an appropriate title and a four-character identifier. Once a sequence is in the "old Atlas" format, it can also be readily edited with a program called CROP so that only specified segments are called on.

CROP seq1 23 487 >seq1X

In this case an edited sequence composed of residues 23 to 487 is put into a separate file. CROP is a very useful program because good alignments of distantly related sequences cannot be obtained if the lengths of the sequences are too different. In this regard, the program INSPECT[13] is

[13] R. F. Doolittle, "URFs and ORFs: A Primer on How to Analyze Derived Amino Acid Sequences," p. 21. University Science Books, Mill Valley, California, 1986.

helpful for identifying appropriate cut points in sequences of different lengths or mosaic composition.

Once the sequences have been gathered, formatted, and cropped to specification, they need to be catenated into a single file. As suggested above, the order is important. Often it is sufficient to input the sequences in an approximate biological order. A more rigorous way is to find an approximate branching order based on pairwise alignments only. The program called ARRANGE makes an alignment of every pair and uses the scores to find an approximate branching order by the method of Fitch–Margoliash.[14] This approximate branching order can then be used as a guide to catenating the sequences into a starting file for the multiple alignment.

Alignment Strategy

Once the sequences are in a single file in an approximate order, they can be aligned with a program called ALIGN. Consider what goes on in this program. Let A, B, C, D, E, ..., be a set of sequences in descending similarity order as determined by ARRANGE. The ALIGN program first aligns A and B. If (AB) represents an optimal alignment between these two sequences, then C(AB) and (AB)C are tried next, the set with the higher similarity score determining whether C(AB) or (AB)C will be the three-way alignment. Assuming that (AB)C has a higher score, the next step involves trying the two alignments, ((AB)C)D and ((AB)D)C. Again, the higher similarity score determines whether ((AB)C)D or ((AB)D)C will be the four-way alignment. This procedure is repeated until all the sequences are aligned. Because of this iterative process, the order of the sequences in the resultfile alignment may be different from that generated by ARRANGE as presented in seqfile.

ALIGN seqfile seqfile1 *M* > resultfile

where *M* is the amino acid mutation matrix which will be discussed in more detail below. Seqfile1 is a file written during execution that contains the sequences with X's inserted as neutral elements at all gap positions. These elements are neutral in the sense that the matching of an X with a residue in the other sequence results in a value of zero. The number of X's tends to increase as more sequences are brought into the alignment, the direct result of the rule "once a gap, always a gap." In addition to its use in scoring within the ALIGN program itself, the file is also used as the input

[14] W. M. Fitch and E. Margoliash, *Science* **155,** 279 (1967).

file for MULPUB, a program that converts the arrangement to a close-pack form of any specified size (in column lengths). Thus,

MULPUB seqfile1 80 > seqfile*M*

rearranges the aligned sequences in a close-packed array 80 columns wide (Fig. 3).

Choice of Matrices

In addition to the Dayhoff (PAM250) matrix,[3] which has been used commonly in the past, a number of other scoring matrices have been published, including one by Gonnet *et al.*[15] and another based on block alignments by Henikoff and Henikoff.[16] Either of these matrices can be used with the ARRANGE, ALIGN, or TREE programs. In an extensive study involving enzyme sequences, we found that the differences between the PAM250 and the Gonnet matrices are small, and both produce reliable results for sequences that are closely related to each other. The BLOSUM62 matrix, on the other hand, appears to give better alignments for more distantly related sequences. For the PAM250 and Gonnet matrices the gap penalty should be set at 8; in the case of the BLOSUM62 matrix, we have found a gap penalty of 6 to be optimum. These values were initially determined by a consideration of the distribution of the scores used in these matrices and then verified empirically by inspection of the alignments produced by them. The Dayhoff PAM250, Gonnet, or BLOSUM matrices can be called simply by adding a D, G, or B to the command line, respectively.

ALIGN seqfile seqfile1 D > resultfile (use Dayhoff PAM250, gap penalty = 8)

or

ALIGN seqfile seqfile1 G > resultfile (use Gonnet, gap penalty = 8)

or

ALIGN seqfile seqfile1 B > resultfile (use BLOSUM62, gap penalty = 6)

Making a Tree

Alternatively, the sequences can be aligned and a tree constructed directly by the program called TREE. The early operations of the ALIGN

[15] G. H. Gonnet, M. A. Cohen, and S. A. Benner, *Science* **256,** 1443 (1992).
[16] S. Henikoff and J. G. Henikoff, *Proteins: Struct. Funct. Genet.* **89,** 10915 (1992).

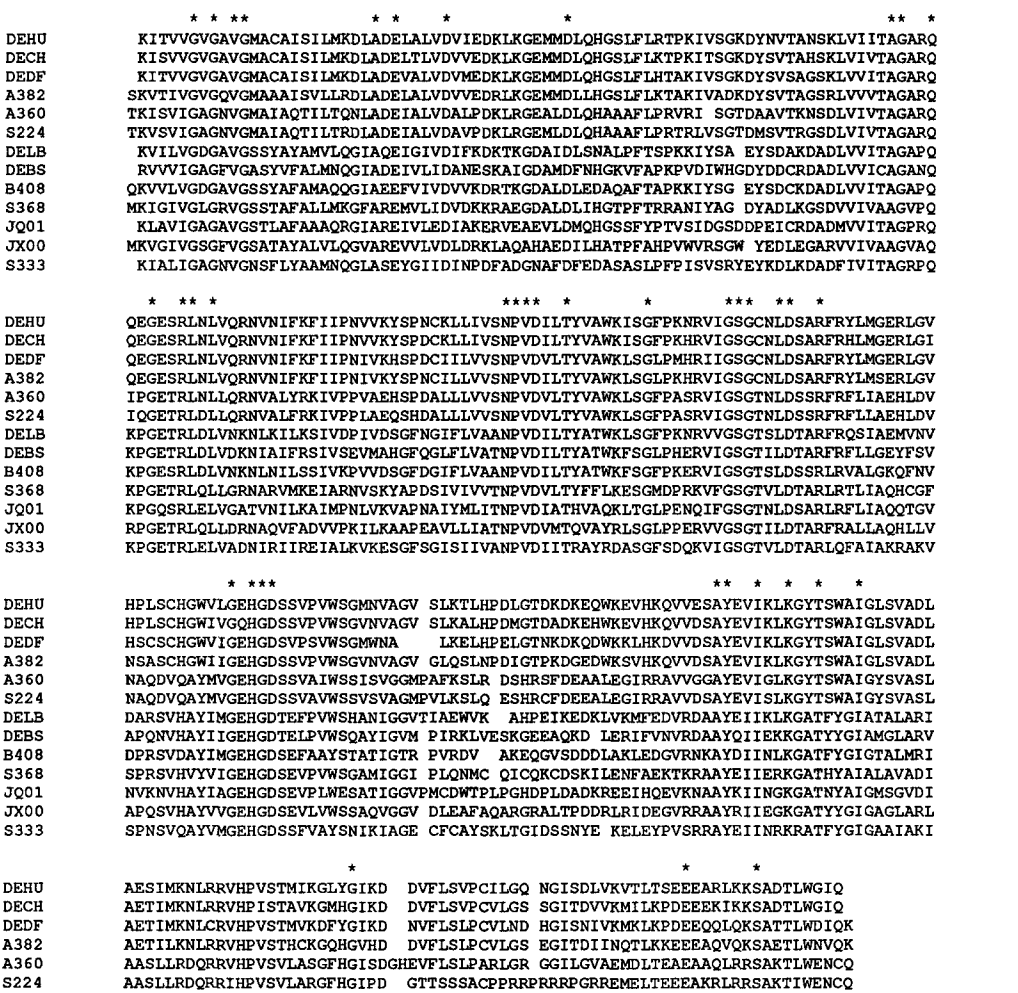

FIG. 3. Example of close-pack format that emerges from MULPUB set at 80 columns across. Thirteen L-lactate dehydrogenase sequences were aligned. The four-letter designations are as follows: DEHU, human; DECH, chicken; DEDF, spiny dogfish; A382, sea lamprey; A360, barley; S224, maize; DELB, *Lactobacillus casei;* DEBS, *Bacillus stearothermophilus;* B408, *Lactobacillus plantarum;* S368, *Thermotoga maritima;* JQ01, *Bifidobacterium;* JX00, *Thermus aquaticus;* S333, *Mycoplasma.*

and TREE programs are identical; the only reason to use ALIGN is the time saving if a tree is not the goal. The TREE program goes on to calculate the branching order and branch lengths.

TREE seqfile seqfile1 *M* > seqfile2

Moreover, the ARRANGE and TREE programs can be called in a single step by use of the shell file OVERALL, which consists of the following two lines,

ARRANGE $1 $4 > $2
TREE $1 $3 $4 > $5

The user must specify the starting file with the collected sequences and designate three new files for the storage of outputs:

OVERALL seqfile seqfile1 seqfile2 *M* > seqfile3

In this case, the final alignment, percent identities, distance, branching order, and branch lengths appear in seqfile3.

Removing Negative Branch Lengths

There are a number of different ways of reducing a difference matrix composed of *m* intergroup distances to a phylogenetic tree with *n* branch lengths. One can solve all the simultaneous equations, for example, or one can use a least squares approach, as suggested by Klotz and Blanken[17] and Li.[18] Although we favor the least squares approach, it does have the flaw that often the best mathematical solution contains one or more negative values, always associated with the inner branch segments of the tree. Most often, the problem can be remedied by simply switching the two taxa (or groups of taxa) that arise at either end of the offending segment (Fig. 4). Many times, however, finding the best arrangement free of negative values is challenging to the point of frustration.

Accordingly, we devised a program that looks for the best matrix-derived tree with no negative segments. It is called NONEG. It works by switching taxa in an iterative manner and testing the result both for the presence of negative segments and the quality of the tree. The latter is measured by a comparison of the initially provided intergroup distances with the final intergroup distances obtained from summing the appropriate branchlengths and is given as a percent standard deviation. The input data

[17] L. C. Klotz and R. L. Blanken, *J. Theor. Biol.* **91,** 261 (1981).
[18] W.-H. Li, *Proc. Natl. Acad. Sci. U.S.A.* **78,** 1085 (1981).

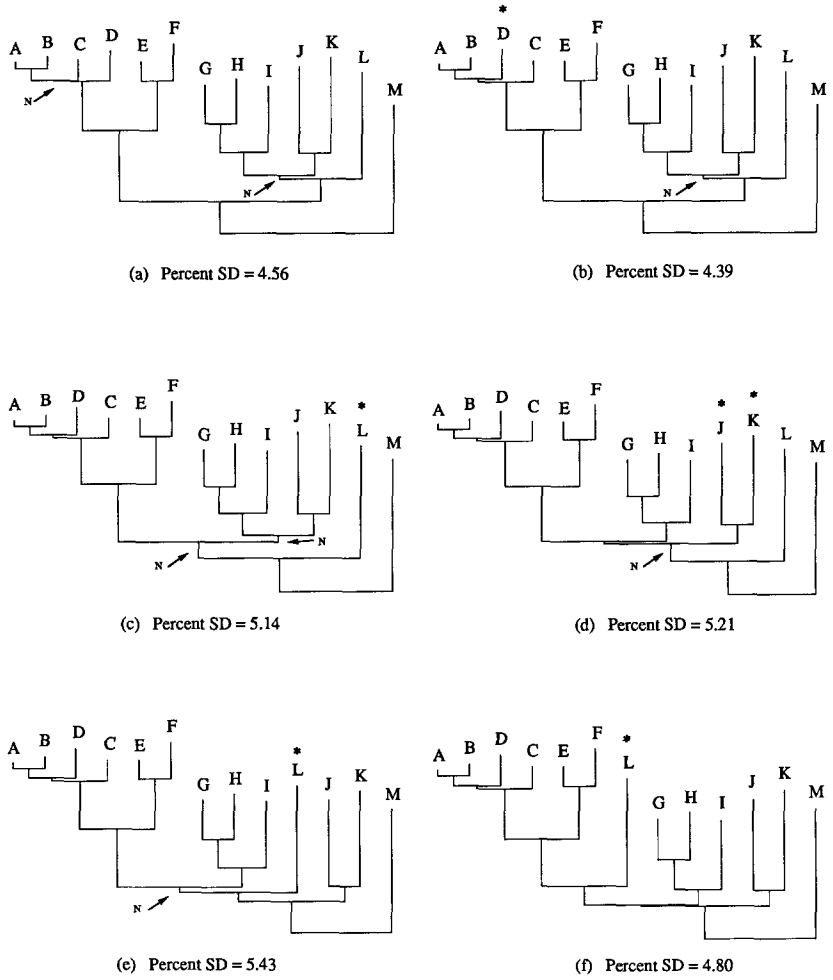

FIG. 4. Illustration of branch swapping that occurs during the elimination of negative branch lengths. In the original tree (a) the arrows indicate two negative branch lengths (for drawing purposes, they are shown as having positive lengths). Asterisks (*) denote taxa involved in switching at various steps. For example, switching the taxa C and D eliminates the first of the negative branch lengths (b) and at the same time improves the overall tree as evidenced by the percent standard deviation dropping from 4.56 to 4.39. Subsequent switchings (c, d, or e) actually make the tree worse without eliminating the final negative value, until finally (f) a solution is reached.

for NONEG appear automatically in a file called ALBR (for alternate branching) every time the program TREE is run. The protocol is simply

NONEG albr > newfile

The newfile contains the initial, intermediate, and final branching orders, along with their respective percent standard deviations.

Bootstrap Analysis

The purpose of making a phylogenetic tree is to provide an estimate of the underlying biological relationships. Because the tree is derived from a multiple sequence alignment, which in turn depends on fluctuations in the real biological world, and to a lesser extent on systematic variables such as the gap penalty and the scoring matrix, the information is expected to have inherent variability. Felsenstein applied the bootstrapping method to phylogenetic trees in an effort to estimate these statistical fluctuations.[19] The method has become a popular way of expressing confidence about the branching order on a node by node basis.

In our version, bootstrapping is invoked by substituting the program BTREE for TREE. The bootstrap itself is performed as follows: the alignment and branching order are found as described above. Then, columns of residues are sampled randomly from the multiple alignment, with replacement. Columns in which gaps (represented as neutral element X) occur in more than half of the positions, and erratic overhangs, if present, are ignored. This sample selection process continues until the lengths of the sequences are the same as the originally aligned sequences. Equations (1), (2), and (3) are used to determine the distance matrix. The best branching order with all positive branch lengths is then saved. The procedure is repeated at least 100 times. Finally, all the branching orders are compared with the original branching order on a node by node basis. The number of times a cluster (around a node) is the same as in the original tree is tabulated; the results are expressed in percentages and placed at the nodes.

Tree Draw Interface

The program TREE produces two hidden files, ALBR and TOMAC. Both contain a branching order and a set of branch lengths, but their formats are different. As noted above, ALBR (alternate branch lengths) is used as an input file for NONEG whenever one or more negative branch

[19] J. Felsenstein, *Evolution* **39,** 783 (1985).

```
(((((AB)C)D)(EF))((((GH)I)(JK))L))M
     % s.d. (obs. vs calc.) =    4.56

(((((AB)D)C)(EF))((((GH)I)(JK))L))M
     % s.d. (obs. vs calc.) =    4.39

((((((AB)D)C)(EF))(((GH)I)(JK)))L)M
     % s.d. (obs. vs calc.) =    5.14

(((((((AB)D)C)(EF))((GH)I))(JK))L)M
     % s.d. (obs. vs calc.) =    5.21

(((((((AB)D)C)(EF))((GH)I))L)(JK))M
     % s.d. (obs. vs calc.) =    5.43

((((((AB)D)C)(EF))L)((GH)I))(JK))M

Branch lengths are (r1,r2,r3,r4,r5):
                   (r6,r7,........):

   1.96     3.74     3.05     9.65     0.99
   6.91    15.76     6.24    11.94    16.38
  14.84    33.56     5.91    12.79    14.85
   9.28    21.67     4.94     0.44    28.33
  31.22     5.01    50.53

     % s.d. (obs. vs calc.) =    4.80
```

FIG. 5. Output of NONEG using the L-lactate dehydrogenase tree example.

lengths appear. It depicts the branching order with a set of consecutive lines that describe clusters:

```
ABCDEFG
DE
*
```

The asterisk denotes the end of branching order information and signifies the beginning of the array of branch lengths.

TOMAC, on the other hand, is the input file to use on the way to tree drawing; it is automatically updated by NONEG. The file begins with the convenient one-line notation of the branching order as described by Fitch[20] and is followed directly by the branch lengths, data that can be read by the program SETREE. SETREE is an interactive program in which the user has a choice of using the default four-letter code or of specifying a new name for each taxon (limited to 20 characters; no spaces or parentheses allowed). The output file, whatever it is named, will contain the phylogenetic tree information in a format that can be read by a hypercard tree drawing program called TREEDRAW DECK that has been adapted from the

[20] W. M. Fitch, *Am. Nat.* **111,** 223 (1977).

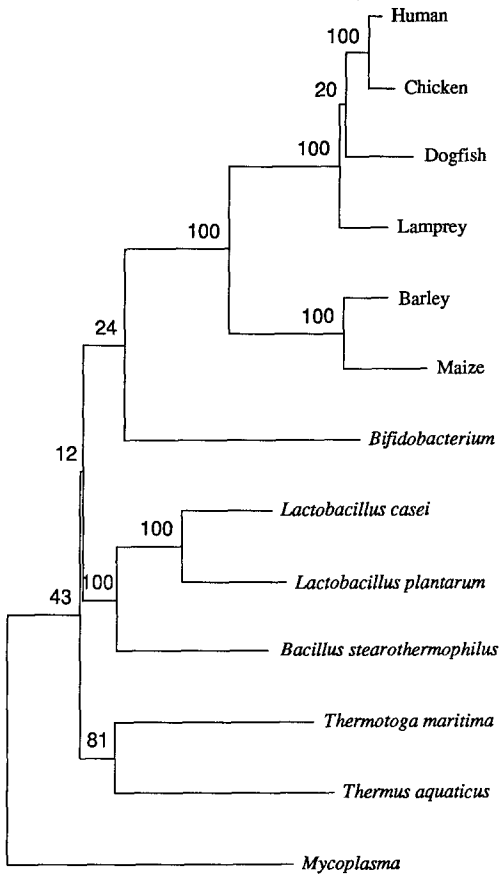

FIG. 6. Bootstrap analysis (100 trials) of 13 L-lactate dehydrogenase sequences. The analysis was restricted to trees with all positive branch lengths.

Felsenstein PHYLIP programs[21] by D. G. Gilbert of Indiana University. This program can be downloaded from the Internet (dgilbert@iubio.bio.indiana.edu) and installed on a suitable microcomputer (our version happens to interface with a MacIntosh). All of the above steps are summarized in Fig. 1.

Example

An illustration of these programs can be afforded by a consideration of a set of L-lactate dehydrogenase (LDH) sequences. From the PIR data-

[21] J. Felsenstein's PHYLIP programs can be obtained by anonymous ftp from 128.95.12.41.

base, 13 LDH sequences were taken, run through the program FORMAT, and then catenated into a file called TEST1, without regard to order. The program ARRANGE was used to calculate first all the binary alignments and put the sequences in an approximate order, and then to proceed directly to making the multiple alignment and finding a branching order.

OVERALL TEST1 TEST2 TEST3 D > TEST4

The output file TEST4 contains the aligned sequences, tables of similarity and distance, a branching order, and a set of branch lengths determined by the method of least squares. As it happens, in this instance, the table of branch lengths contains two negative values (Fig. 4a). Accordingly, the program NONEG was used to find a valid branching order.

NONEG ALBR > TEST5

The output (in TEST5) shows the beginning, intermediate, and final branching orders and their respective percent standard deviations (Fig. 5).

The SETREE program is then used to generate a file for tree draw.

SETREE tomac > tomac1

Finally, the bootstrap version of the program was conducted separately:

BTREE seqfile seqfile1 D > resultfile
CLUS tomac1 rboot > rboot1

It must be underscored that the bootstrapping is restricted to solutions with no negative branch lengths, a necessary condition if the likelihoods are to have any meaning. This condition is often ignored by other bootstrapping procedures involving amino acid alignments made by matrix methods. The results are stored in a hidden file named rboot. A full calculation generally consists of 100 bootstrap trials. When rboot is analyzed by the program CLUS, the number of times (for 100, it is the percentage) a node is in agreement with that in the starting tree is shown on the phylogenetic tree (Fig. 6).

Program Availability

All the programs described in this chapter are available by anonymous ftp from juno.ucsd.edu. After logging in with any identifying name as a password, claimants should go to the directory progs/.