A Simple Method for Aligning Many Protein Sequences

Peter Bladon[†]

Department of Pure and Applied Chemistry, University of Strathclyde, Glasgow G1 1XL Scotland, United Kingdom

Received July 9, 1999

A simple extension of the Needleman and Wunsch algorithm for aligning pairs of protein sequences allows it to be used for the efficient generation of very large multiple-sequence alignments whose members are similar. This technique could have applications in a broad range of high-volume genomics projects.

INTRODUCTION

In 1970 Needleman and Wunsch[1] published details of an efficient algorithm for optimal alignment of a pair of protein sequences. The method is based on searching for the identity (or similarity) of amino acid residues in the two sequences. Gaps are introduced into the two sequences until maximal identity (or similarity) is achieved. The extension of the process to align more than two sequences is not trivial. Murata et al.[2] provided a method for aligning three sequences, but to process more than three sequences would have needed the use of multidimensional arrays of dimensionality equal to the number of sequences being aligned. This is clearly not a workable extension. However, the Needleman and Wunsch algorithm and derivations of it do form the basis of most methods of aligning multisequences; such methods treat the sequences pairwise, introducing (and retaining) gaps as needed, and repeat the process until no further modification of sequences is needed. An example of this method is the program MULTAL.[3] Clearly such methods scale as the square of the number of sequences to be treated. In contrast, the method described in the present paper, although based on the Needleman and Wunsch algorithm, scales directly as the number of sequences to be aligned. Other methods depend on the use of prior information on the underlying biological relationship of the sources of the sequences. Thus, the so-called *progressive alignment techniques* require the prior availability of phylogenetic trees. These methods are not appropriate when the aim of alignment is the generation of such evolutionary information.

A discussion of the methods of aligning protein sequences is contained in the book by Doolittle,[4] and more extensive reviews are contained in two volumes in the series *Methods in Enzymology*. See, for example, the articles by Taylor[5,6] Feng and Doolittle,[7] and Higgins et al.[8]

METHOD

The method described here is offered as a simple rapid method of performing an initial alignment of a series of protein sequences. It uses a choice of the usual matrices (Dayhoff, PAM250, etc.) for comparing the amino acid residues, but apart from this and the user's choice of the template sequence (see below) and the maximum width of allowed gaps, only the information contained within the sequences is used. The steps in the process are as follows:

Step 1. A "template" sequence is selected. This is a sequence usually chosen from those to be aligned, either that deemed to be most representative of the set or the longest member of the set. The template sequence may, however, be chosen from outside this set. In all cases a copy of this sequence is used throughout the subsequent processes.

Step 2. Each of the sequences in the set to be aligned is taken in turn, and becomes the "trial" sequence, and is aligned with the template sequence using the normal Needleman and Wunsch algorithm. Gaps are introduced into both the trial sequence and the template sequence. Gaps introduced into the template sequence are retained when the next trial sequence is taken and aligned. The result of this process is a set of sequences that are not yet aligned with each other and in which only the last one will be optimally aligned with the template sequence. In general, if all the sequences of the starting set are of the same length, at this stage the set of sequences will increase in length from first to last, since the later sequences will have more gaps added.

Step 3. All gaps are removed from the set of trial sequences. (Alternatively, the original ungapped set of sequences can be reloaded.) Gaps are not removed from the extended template sequence.

Step 4. Each of the sequences in the trial set is aligned with the extended template sequence. A modified version of the Needleman and Wunsch algorithm is now used; in this version gaps are only introduced into the trial sequences. No further gaps are introduced into the already extended template sequence. After this stage it will be found that the set of trial sequences will be aligned with each other (and with the template sequence).

Step 5. If, in the aligned array of trial sequences, there are gaps that are common to all the sequences, these are removed. This step is usually only necessary when the template sequence has been chosen from outside the trial set.

RESULTS

Figure 1 shows the partial results of the alignment of a series of 23 sequences of mitochondrial cytochrome *b*

[†] Present address: Interprobe Chemical Services, Gallowhill House, Larch Ave., Lenzie, Kirkintilloch, Glasgow, Scotland G66 4HX. Telephone/Facsimile: +44-(0)141-578-1109. E-mail: cbas25@strath.ac.uk. URL: http://www.interprobe.co.uk.

10.1021/ci9904362 CCC: $20.00
Published on Web 02/06/2001

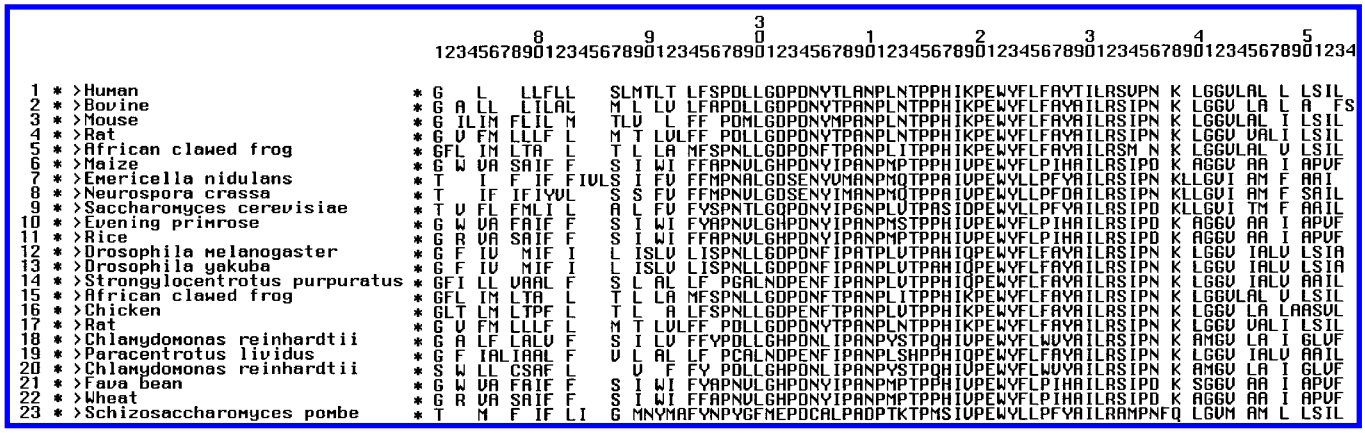

Figure 1. Alignment of the mitochondrial cytochrome *b* sequence. The region of residues 271−354 (after alignment) are shown.

proteins. These were extracted from the PIR database.[9] (To economize in space in the figure, the sequence identifying codes have been removed.) The longest sequence (before alignment) was number 22 (wheat, which had 398 residues), and this was chosen as the template sequence. The Dayhoff "log-odds" matrix[10] was used for alignment. The introduction of an unlimited number of gaps, up to a maximum size of 10 residues per gap, was allowed. The longest sequence after alignment had 470 residues plus gaps. This result is typical when highly conserved sequences are compared. Thus in the figure there is a region on the right where relatively few gaps are introduced and where the alignment is good. On the left is a typical region where the alignment is poorer, where the gaps are larger, and where the individual alignments could be criticized (for example, the alignment of the phenylalanine residues at positions 281−283).

DISCUSSION

Resources Required. To process k sequences of length n, and with maximum length of a single gap l, the cpu time required by this new method scales as $O(knl)$. Since all transformations are done in place, the storage requirements scale as $O(kn)$. The gap parameter l works as a look-ahead limiter for the matching process. In the limit where $l > n$, the time scaling becomes $O(kn^2)$ due to the design of the program. (Note that the total *number* of gaps is unrestricted.) These time relationships are have been confirmed in use of the software (see Table 1).

Comparisons with Other Methods. According to Kececioglu,[11] sequence alignment by dynamic programming methods scale as $O(2^k n^k)$ for time and $O(n^k)$ for storage. These requirements rule the methods out for any but the smallest data sets. Popular methods based on merged iterative pairwise alignment require times $O(k^2 n^2)$ for the alignments plus $O(k^2(\log k) + kn)$ for the merging. The method described in this paper compares very favorably with these requirements. It performs well on data sets which are larger than those than can be handled by other methods (see Kececioglu[11] and Reinert et al.[12]). Comparison of timings were carried out for alignment of typical sets of sequences using the method described here and using CLUSTAL-W.[8] The results are shown in Table 1.

General Remarks. The basis of the method can be criticized for the requirement that a template sequence is needed. The template needs to be representative of the series as a whole; if it is not representative, then no alignment will be achieved. In contrast to iterative alignment schemes, a rogue template sequence will either result in no alignment or misalignment of the other sequences. In practice, however, the whole art of sequence alignment is subject to the personal preferences of the user, for example, in the choice of the alignment matrix. It would be a simple matter to alter the code to make the longest sequence the automatic choice as template. An alternative extension of the program would allow each sequence in turn to function as template, with the best overall alignment being chosen finally. This would result in time scaling as $O(k^2 nl)$ [or in the large gap limit $O(k^2 n^2)$, which is the same as for iterative pairwise alignment].

In contrast to many other methods, the quality of the pairwise correlations (the scoring) is not used within the program. Scoring is provided at the end of the process and can be used (for example) in the construction of phylogenetic trees.

Table 1. Elapsed Times for Aligning Representative Sequences[a]

				time (s)	
data set[b,c]	k[d]	n[e]	l[f]	PRESTO	CLUSTAL-W
1	42	341	10	42	169
1	42	341	20	59	
1	42	341	30	76	
2	26	273	10	18	41
2	26	273	20	22	
2	26	273	30	27	
3	18	557	10	34	85
3	18	557	20	49	
3	18	557	30	66	
4	60	404	10	74	436
1*	21	341	30	20	
1*	7	341	10	10	
3*	18	500	10	33	
3*	18	200	10	13	
3*	18	100	10	9	
4*	23	398	10	29	(cf. Figure 1)

[a] The program MULTAL referred to in the paper by Reinert et al.[12] could not handle the alignment of a set of 15 prion proteins (similar to those in the set used in this paper). The program referred to in the paper by Kececioglu11 took 176.6 s to align a set of six tyrosine kinases of length 280 residues (but could not handle a set of seven such sequences because of insufficient memory). [b] Data sets were abstracted from the PIR database:[9] 1 = glyceraldehyde-3-phosphate dehydrogenases; 2 = prion proteins; 3 = tyrosine kinases; 4 = mitochondrial cytochromes *b*. [c] Sequence sets marked with an asterisk were made by editing and/or truncating the corresponding parent sets. [d] k is the number of sequences in the set. [e] n is the length of the longest sequence in the set. [f] l is the maximum allowed length of a single gap.

In common with many other methods of alignment this method runs into difficulties when trying to match the extreme ends of sequences. There are also instances of it producing alignments to which an expert might object, for example, when there are long runs of the same amino acid type.

On balance, it is believed that the present extension of the Needleman and Wunsch method is valuable in affording a simple and rapid starting point for fine-tuning of alignments, particularly in the restricted field of alignment of large sets of highly similar sequences. Potential applications include population genotyping for discovering (and subsequently predicting) drug sensitivity/toxicity, investigation of predisposition to certain diseases, or any other field of study involving large numbers of sequences. The principles of the method may have applications in separate but related fields. These include the abstraction from databases of sequences similar to a target sequence[5] and the alignment of sets of homologous sequences of low similarity.[15]

IMPLEMENTATION

This method is incorporated in the peptide and nucleotide analysis package PRESTO, where there are auxiliary facilities for manipulating and editing of sequences, and a range of alignment matrices. PRESTO is available as part of the INTERCHEM modeling software, available from Interprobe Chemical Services (for the address see the footnote on the first page of the paper). The code of PRESTO is written in Fortran 77, but the simplicity of the extensions to the basic Needleman and Wunsch algorithm described above would allow the method to be implemented in other languages, or even as an interpreted command sequence.

The "wall clock" timings of all the runs in PRESTO (and also the comparison experiments run with CLUSTAL-W) were made on an SGI O2 machine, fitted with an IP32 Mips 5000 processor and 128 MB memory. During the runs there was no other activity except that due to the operating system.

ACKNOWLEDGMENT

The author wishes to thank Professor Douglas McGregor of the Department of Computer Science, and Dr. Mark Dufton of the Department of Pure and Applied Chemistry, University of Strathclyde, for helpful advice and discussions. The author is also indebted to Dr. Charles Hodgman of Glaxo Wellcome Research and Development, Stevenage, for indicating potential uses for the technique described. He also wishes to thank a reviewer for comments on an earlier draft and suggestions of additional references.

REFERENCES AND NOTES

(1) Needleman, S. B.; Wunsch, C. D. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *J. Mol. Biol.* **1970**, *48*, 443−453.

(2) Murata, M.; Richardson, J. S.; Sussman, J. L. Simultaneous Comparison of Three Protein Structures. *Proc. Natl. Acad. Sci. U.S.A.* **1985**, *82*, 3073−3077.

(3) Taylor, W. R. A Flexible Method to Align Large Numbers of Biological Sequences. *J. Mol. Evol.* **1988**, *28*, 161.

(4) Doolittle, R. F. *Of Urfs and Orfs*; University Science Books: Mill Valley, CA, 1986.

(5) Taylor, W. R. Hierarchical Method to Align Large Numbers of Biological Sequences. *Methods Enzymol.* **1990**, *183*, 456−474.

(6) Taylor, W. R. Multiple Protein Sequence Alignment: Algorithms and Gap Insertion. *Methods Enzymol.* **1996**, *266*, 343−368.

(7) Feng, D.-F.; Doolittle, R. F. Progressive Alignment of Amino Acid Sequences and Construction of Phylogenic Trees from Them. *Methods Enzymol.* **1996**, *266*, 368−382.

(8) Higgins, G. G.; Thompson, J. D.; Gibson, T. J. Using CLUSTAL for Multiple Sequence Alignments. *Methods Enzymol.* **1996**, *266*, 383−402.

(9) PIR−International Protein Sequence Database, National Biomedical Research Foundation, 3900 Reservoir Road NW, Washington, DC 20007.

(10) Schwartz, R. M.; Dayhoff, M. O. In *Atlas of Protein Sequence and Structure*; Dayhoff, M. O., Ed.; National Biomedical Research Foundation: Washington, DC; Vol. 5, 1972.

(11) Kececioglu, J. The Maximum Weight Trace Problem in Multiple Sequence Alignment. *Lect. Notes Comput. Sci.* **1993**, *684*, 106−109

(12) Reinert, K.; Lenhof, H.-P.; Mutzel, P.; Mehlhorn, K.; Kececioglu, J. D. A Branch-and-Cut Algorithm for Multiple Sequence Alignment. *Recomb 97* **1997**, *97*, 241−250.

(13) Lawrence, C. E.; Altschul, S. F.: Boguski, M. S.; Liu, J. S.; Neuwald, A. F.; Wootton, J. C. Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment. *Science* **1993**, *262*, 208−214.

CI9904362