Volume 10 Number 9 1982 Nucleic Acids Research

An interactive graphics program for comparing and aligning nucleic acid and amino acid sequences

Rodger Staden

MRC Laboratory of Molecular Biology, The MRC Centre, Hills Road, Cambridge CB2 2QH, UK

Received 12 March 1982; Revised and Accepted 19 April 1982

ABSTRACT

This paper describes a computer program designed to look for similarities between pairs of nucleic or amino acid sequences. The program looks both for segments of perfect identity or for regions where, using a scoring matrix, a minimum value is exceeded. The results of comparisons are presented as a matrix which is displayed on a simple graphics terminal. Use of a graphics terminal allows the user to display the whole of the two sequences in one screenful or to home-in on regions of interest to examine them in more detail. The program is interactive and so the user can easily see the effect of changes to variables and can use inbuilt editing functions to make insertions to produce alignments of the two sequences. These aligned sequences can then be saved on disk files for further processing.

INTRODUCTION

There are now many computer methods for comparing pairs of nucleic or amino acid sequences to look for similarities and a number of different approaches have been used. Needleman and Wunsch [1] developed an algorithm to find the maximum match between two sequences using a score matrix and gap penalties. Sellers [2] produced a metric measure of the distance between two sequences which could be minimised to find the best alignments. Similar methods or refinements of these techniques have been developed by Fitch [3], Sankoff [4] and Waterman, Smith and Beyer [5]. Dayhoff's method [6] compares all short segments of one sequence against all those of the other, and Korn, Queen and Wegman [7] look for blocks of perfect identity allowing mismatches and loop-outs. DIAGON, the program described here, combines some of the techniques that we consider to be most useful with some new features to produce a simple yet powerful tool for sequence comparison and alignment. The program (written in FORTRAN 77 plus some device specific graphics commands) is used on a simple interactive graphics terminal [8] connected to a VAX 11/780 computer. The algorithms are fast enough for the program to be interactive and so allow the user to try the effect of changing any of the

0305-1048/82/1009-2951$2.00/0

parameters used by the routines. The ability to perform the analysis interactively is important because the choice of some of the parameters is often critical and can only be decided by trial and error.

The basic principle of the method employed by DIAGON was first described by Gibbs and McIntyre [9] and involves producing a diagram that contains a representation of all the matches between a pair of sequences. This diagram is then scanned by eye and the human ability to recognise patterns used to detect any similarities that might be present. The diagram consists of a two dimensional plot in which the x axis represents one sequence (A) and the y axis the other (B). Every point (i,j) on the plane x,y is assigned a score which corresponds to the level of similarity between sequence characters A(i) and B(j). In the simplest use of the method a score of 1 could be assigned to every point (i,j) where A(i) = B(j), and a score of 0 to every other point. If a plot of the points in the plane was made in which all scores of 1 were marked with a dot and all those of 0 left blank, then regions of identity would appear as diagonal lines. With the comparison displayed in this form the human eye is very good at detecting regions of homology even if they are imperfect. The effects of mismatches, insertions or deletions can be seen: matches interrupted by insertions or deletions will appear as parallel diagonals, and matches interrupted by the odd mismatching pair of characters will appear as broken collinear diagonal lines. This diagram is a very useful representation but simply placing a dot for every identity is of limited value for the following reasons.

For nucleic acid sequences around 25% of the plot will contain points and it will often be very difficult to distinguish significant homologies from chance matches. For proteins many significant alignments of sequences contain almost no identities but are formed from chemically and structurally similar amino acids so that simply looking for identity would be insufficient. What is required is to first find those points that correspond to fairly strong local similarities and then to use the diagram of these points so that the human eye can be used to look for larger scale homologies. DIAGON uses two different algorithms to calculate the score for each point and the user defines a minimum score so that only those points in the diagram for which the score is at least this value will be marked with a dot. The first scoring method is to find the longest uninterrupted sections of perfect identity, i.e. those that contain no mismatches, insertions or deletions. The second method looks for sections where a proportion of the characters in the sequence are similar, again allowing no insertions or deletions. These

two scoring methods, respectively referred to as the perfect and proportional algorithms, are described below.

SCORING METHODS

Perfect matching

When looking for perfect matches the program stores the accumulated score for each diagonal at the current position, i.e. the score at any point is the number of identities found looking back along the diagonal. At each point that a mismatch occurs the score is set back to zero so that, for example, a match of length six will have on its diagonal scores of 1,2,3,4, 5,6 and the next score will be zero. An advantage of this method of calculating the scores is that the user can see the heights of the peaks. For example, if the user plots all peaks of five and above he knows that a diagonal of two points represents a match of six and a diagonal of eight a match of 12.

Proportional matching

This method, generally the most useful, was first described by McLachlan [10] and involves calculating a score for each position in the matrix by summing points found when looking forwards and backwards along a diagonal line of a given length. This length, called the span, should be an odd number so that the score for any point is correctly positioned at the centre of the span. The algorithm does not simply look for identity but uses a score matrix that contains scores for every possible pair of characters. For comparing amino acid sequences we use the score matrix shown in Table 1, which was calcualted by adding 10 (to make every term >0) to each term of the relatedness odds matrix MDM78 of Dayhoff [6]. This matrix MDM78 was calculated by looking at accepted point mutations in 71 families of closely related proteins and, of those tested by Dayhoff, was found to be the most powerful score matrix for finding distant relationships between amino acid sequences.

For DNA the score matrix is simply the matrix in Table 2(a) although it could be altered to give a positive score for, say, T-C and A-G as in Table 2(b).

THE ALGORITHMS

For the program to be fast enough to be run interactively the scoring algorithms must be efficient. For two sequences of lengths L and M the number of calculations for the two algorithms is as follows. For perfect matching the number of calculations is of the order to 2LM, i.e. independent

Table 1. Amino acid score matrix

	C	S	T	P	A	G	N	D	E	Q	B	Z	H	R	K	M	I	L	V	F	Y	W	-	X	?	
C	22	10	8	7	8	7	6	5	5	5	5	5	7	6	5	5	8	4	8	6	10	2	10	10	10	10
S	10	12	11	11	11	11	11	10	10	9	10	10	9	10	10	8	9	7	9	7	7	8	10	10	10	10
T	8	11	13	10	11	10	10	10	10	9	10	10	9	9	10	9	10	8	10	7	7	5	10	10	10	10
P	7	11	10	16	11	9	9	9	9	10	9	10	10	10	9	8	8	7	9	5	5	4	10	10	10	10
A	8	11	11	11	12	11	10	10	10	10	10	10	9	8	9	9	9	8	10	6	7	4	10	10	10	10
G	7	11	10	9	11	15	10	11	10	9	10	10	8	7	8	7	7	6	9	5	5	3	10	10	10	10
N	6	11	10	9	10	10	12	12	11	11	12	11	12	10	11	8	8	7	8	6	8	6	10	10	10	10
D	5	10	10	9	10	11	12	14	13	12	13	12	11	9	10	7	8	6	8	4	6	3	10	10	10	10
E	5	10	10	9	10	10	11	13	14	12	12	13	11	9	10	8	8	7	8	5	6	3	10	10	10	10
Q	5	9	9	10	10	9	11	12	12	14	11	13	13	11	11	9	8	8	8	5	6	5	10	10	10	10
B	5	10	10	9	10	10	12	13	12	11	13	11	11	10	10	8	8	6	8	5	7	4	10	10	10	10
Z	5	10	10	10	10	10	11	12	13	13	11	14	12	10	10	8	8	8	8	5	6	4	10	10	10	10
H	7	9	9	10	9	8	12	11	11	13	11	12	16	12	10	8	8	8	8	8	10	7	10	10	10	10
R	6	10	9	10	8	7	10	9	9	11	10	10	12	16	13	10	8	7	8	6	6	12	10	10	10	10
K	5	10	10	9	9	8	11	10	10	11	10	10	10	13	15	10	8	7	8	5	6	7	10	10	10	10
M	5	8	9	8	9	7	8	7	8	9	8	8	8	10	10	16	12	14	12	10	8	6	10	10	10	10
I	8	9	10	8	9	7	8	8	8	8	8	8	8	8	8	12	15	12	14	11	9	5	10	10	10	10
L	4	7	8	7	8	6	7	6	7	8	6	8	8	7	7	14	12	16	12	12	9	8	10	10	10	10
V	8	9	10	9	10	9	8	8	8	8	8	8	8	8	8	12	14	12	14	9	8	4	10	10	10	10
F	6	7	7	5	6	5	6	4	5	5	5	5	8	6	5	10	11	12	9	19	17	10	10	10	10	10
Y	10	7	7	5	7	5	8	6	6	6	7	6	10	6	6	8	9	9	8	17	20	10	10	10	10	10
W	2	8	5	4	4	3	6	3	3	5	4	4	7	12	7	6	5	8	4	10	10	27	10	10	10	10
-	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
X	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
?	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

of the minimum match value. For the proportional matching the number of calculations is again 2LM plus a small addition of K(L+M) where K is the span length. Again, the calculation time is largely independent of the span length. This saving in time is achieved by first calculating the values for the top and left edges of the diagram [giving the overhead of K(L+M)]. We then use a rotating score for each point in the diagram so that for each point we simply add a value to the leading edge of the diagonal and remove one from the trailing edge, hence giving a calculation time proportional to 2LM.

Table 2. DNA score matrices

	A	C	G	T	X
A	1	0	0	0	0
C	0	1	0	0	0
G	0	0	1	0	0
T	0	0	0	1	0
X	0	0	0	0	0

2(a)

	A	C	G	T	X
A	5	0	1	0	0
C	0	5	0	1	0
G	1	0	5	0	0
T	0	1	0	5	0
X	0	0	0	0	0

2(b)

The memory requirement of the scoring algorithms, including the storage of the sequences, is of order 2(L+M) for the perfect and 2L+3M for the proportional, hence making them suitable for quite small computers.

USING THE PROGRAM

The program is used on a simple graphics terminal, i.e. a keyboard with a screen on which points and lines can be drawn. The user works at the terminal and produces plots for various combinations of values for the span length and minimum scores. The plots appear as in Figures 1-4 with the top left corner of the screen corresponding to the left ends of the two sections of sequence being compared. How ever large or small a region the user elects to compare, the program expands or contracts the diagram so that the plot always fills the screen. This allows the user to gain an overall impression or to 'home-in' on particular regions and examine them in more detail. Having found a region that looks interesting the user can determine its coordinates in terms of sequence positions by use of a crosshair facility. When this option is selected a large cross appears on the screen. The user can move the cross around the screen by use of special directional keys and can have the position of the cross displayed in terms of sequence positions.

The writing of characters and plotting of points are entirely independent so that the prompts and characters typed by the user scroll upwards but the plot stays on the screen until the user elects to have it cleared. Giving the user control of screen clearing gives him the ability to overlay different plots.

The program has two statistical options to help the user choose score levels for plotting and to assess the significance of any similarity found. It can produce a cumulative histogram of observed scores for the current span length and region and it can calculate the "double matching probability" of McLachlan [10]. The double matching probability is the probability of finding particular scores given two infinitely long sequences of the composition of those being compared, with the current span length and score matrix. By using these options the user can choose to plot all the matches for which the score exceeds a given significance level (such as 1%), using either empirical or theoretical probability values. Generally it is best to begin at a low level to avoid an overcrowded diagram.

If the user finds that the two sequences do contain stretches of homology he will often want to align the sequences by inserting padding characters at deletion points. The program has a selection of options for

this purpose: it can display on the screen the two sequences, one above the other, with asterisks marking identities, it has inbuilt editing functions and can save the aligned sequences on disk files. These options, combined with the diagonal display and crosshair facility make the task of producing aligned sequences relatively easy. (A possible improvement of the alignment procedure would be to incorporate an option to apply the Needleman and Wünsch [1] or Sellers [2] algorithms to deal with any problematical sections.)

EXAMPLES

One of the projects currently being undertaken in this laboratory is the sequencing of the unc-54 myosin heavy chain gene from the soil nematode C. elegans [11]. This sequence is now nearing completion and the program described in this paper is being used to analyse both the DNA and its translation into protein sequence. Four plots showing a search for internal repeats within the rod region of the sequence are shown in the figures. The first three use the MDM78 matrix and are of the amino acid sequence and the fourth is of the corresponding DNA sequence. This section of the sequence is about 1113 amino acids in length and all diagrams cover the whole region. The N-terminal ends of the sequence are at the top left of the diagrams with the C-terminal ends at the bottom right. (The grid over the diagram is a user selectable option and is used to aid in interpreting the plots.) In all these diagrams there is, of course, a central diagonal showing self matching.

Figures 1 and 2 demonstrate the effect of a change of span length: both these plots have a score threshold at the same level of expectation (1% as calculated by the McLachlan [10] "double matching probability") and differ only in span length. The span length for Figure 1 is seven amino acids and for Figure 2 a span of 99 amino acids was used. Figure 1 is very difficult to interpret but shows that there is possibly some level of repetitiveness in the sequence but Figure 2 contains a great deal of information. The parallel diagonal lines are characteristic of a highly repetitive sequence with the separation of the diagonals showing the length of the repeating units. It can be seen that there are several repeat lengths, the shorter of these being most evident at the N-terminal end of the sequence and the longer persisting throughout the whole length of the protein. The length of these repeats can easily be measured by using the crosshair facility particularly if the user elects to expand the resolution of the plot by concentrating on a smaller area.

Figure 3 is over the same region but is for a span length of 99 amino

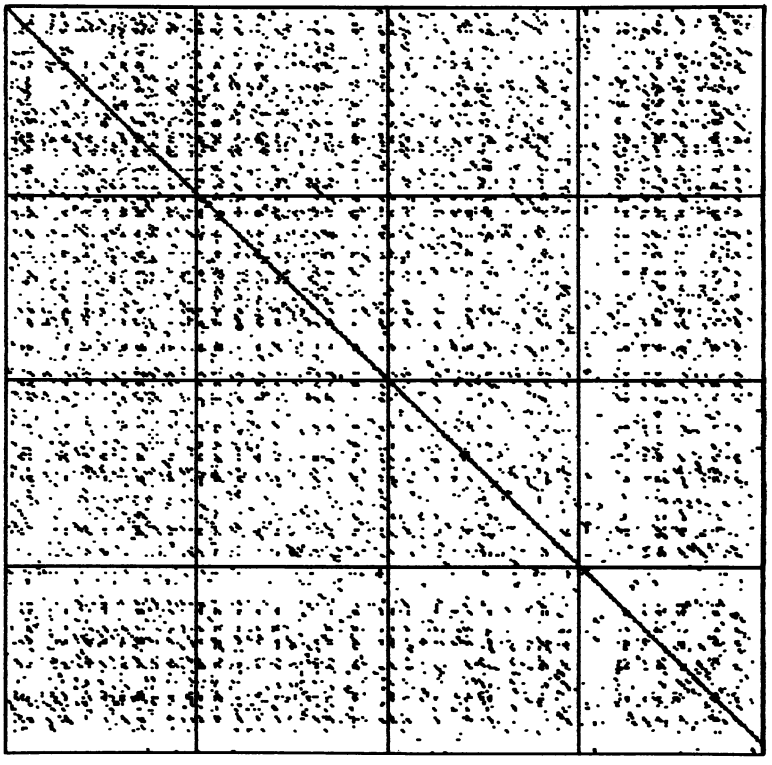

Fig. 1. A diagram (see text) for the amino acid sequence of the rod region of nematode myosin using a span of 7 and a score at the 1% level of expectation.

acids and a score threshold of 0.01%. This shows that the longer repeat unit is more persistent than the shorter.

Figure 4 is of the DNA encoding this portion of the protein and consequently is 3339 bases long. In order to make comparisons between the levels of similarity in the DNA and the protein the span length chosen is 3 x 99 = 297 bases. This plot is at the 0.01% level of expectation and it can be seen that the repeats also appear in the DNA sequence. The user can compare observed and expected score levels for both the DNA and protein using the statistical options. A paper by J. Karn and A.D. McLachlan describing the analysis of these sequences is in preparation.

DISCUSSION

The program described is used to find similarities between sequences.

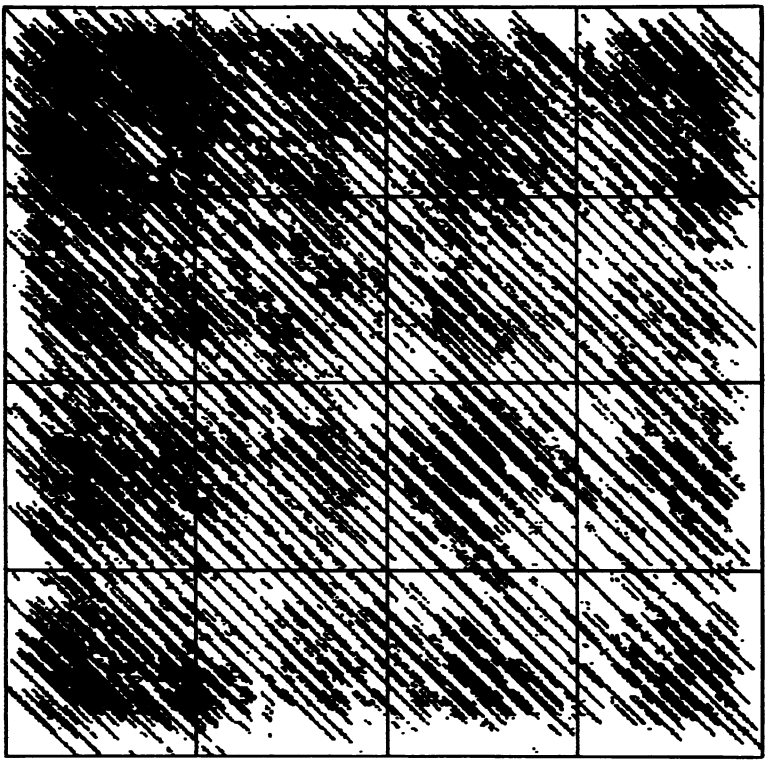

Fig. 2. A diagram (see text) for the amino acid sequence of the rod region of nematode myosin using a span of 99 and a score at the 1% level of expectation.

The visual representation and interactive nature of the program serve to keep the user close to his data and allow him to experiment with different spans and scores. For many users the diagrammatic presentation is greatly preferred to purely algorithmic methods and it can be used to give both an overall impression as well as allowing more detailed examination of local matches. The importance of experimentation is shown in the examples which demonstrate the marked effect of a change of span length and how the choice of score threshold can help in interpretation of diagrams.

Interactive work has been made possible by the recent availability of low cost graphics terminals and the use of fast new algorithms. The speed of the algorithms is largely independent of span length and score with the slowest having an execution time proportional to $2LM+K(L+M)$. The algorithms are also designed to keep program size down having a storage requirement

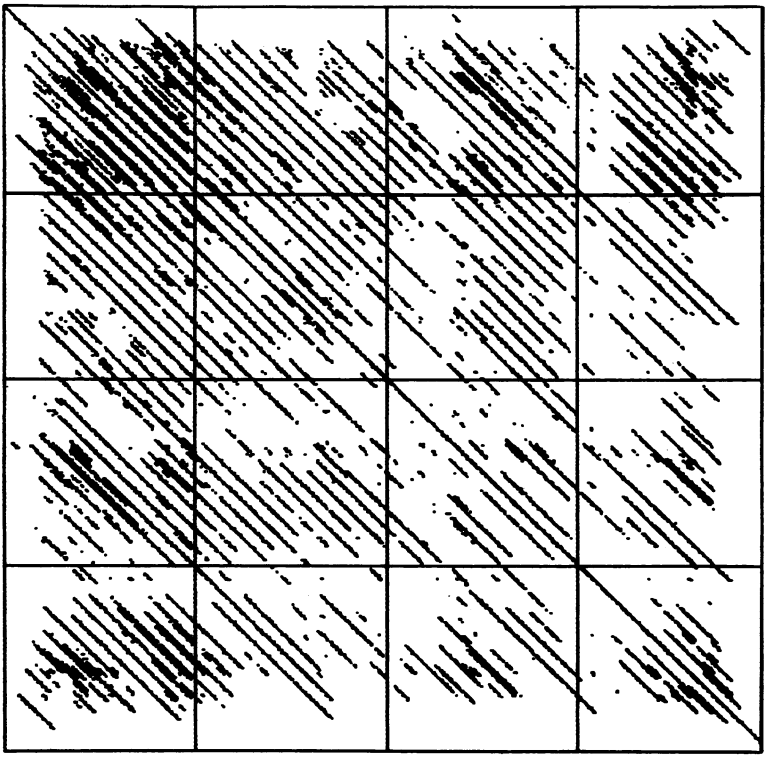

Fig. 3. A diagram (see text) for the amino acid sequence of the rod region of nematode myosin using a span of 99 and a score at the 0.01% level of expectation.

including storage of the sequences of 2L+3M. The speed and storage requirements should be compared to some Gibbs-McIntyre and all Needleman-Wunsch programs which need storage of at least LM.

Most of the programs currently available to look for similarities between sequences use Monte Carlo methods to assess the significance of observed matches. This is a time consuming process and therefore unsuited for interactive work whereas the "double matching probability" of McLachlan [10] used by this program is both a close approximation to observed distributions for random sequences and can be calculated very quickly. The user can compare the expected score distributions with those observed while running the program and copies of the two distributions can be taken for later, more careful, examination.

The screen we currently use is monochromatic but it would be very easy

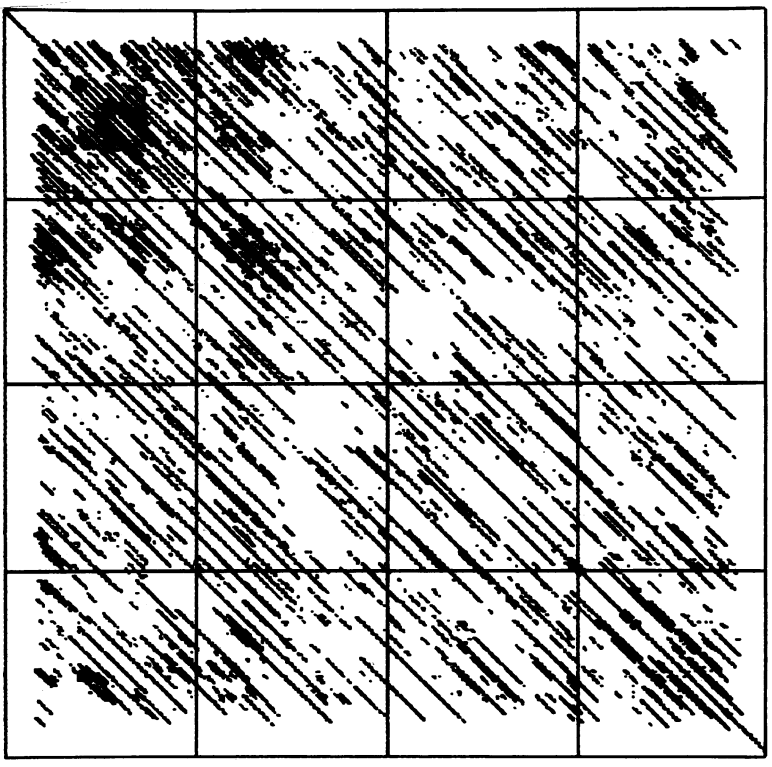

Fig. 4. A diagram (see text) for the DNA sequence that encodes the rod region of nematode myosin using a span of 297 and a score at the 0.01% level of expectation.

to modify the program to use colour and hence increase its power. Such a facility would allow, for example, identification of charged amino acids or regions of hydrophobicity, or could be used to indicate the heights of the peaks in the plots.

While this manuscript was in preparation a number of other papers describing programs addressing the same problem have appeared in the literature. Maizel [12] uses an equivalent of the "perfect matching" scoring method, Novotny [13] uses a combination of this method with the ability to group similar amino acids, Jagadeeswaran [14] uses an equivalent of the "proportional matching" algorithm looking only for identity and Harr [15] has produced a program that gives diagrammatic output from an extended Korn [7] algorithm. Goad [16] has produced a generalisation of the Needleman-Wunsch [1]/Sellers [2] algorithm that forms part of a comprehensive package of

sequence analysis programs described by Kanehisa [17].

A magnetic tape containing DIAGON, the program described in this paper, plus VAX versions of other programs [18-23] developed in this laboratory for handling and analysing sequences is available on request.

Acknowledgements

I would like to thank A.D. McLachlan for advice during this work and for giving me a copy of his subroutine for calculating the expected frequency distribution. I also thank T.S. Horsnell for use of his graphics subroutines and J. Karn and A.D. McLachlan for allowing me to use their unpublished results for Figures 1-4.

REFERENCES

1. Needleman, S.B. and Wunsch, C.D. (1970) J. Mol. Biol. 48, 443-453.
2. Sellers, P.H. (1974) J. Appl. Math. (Siam) 26, 787-793.
3. Fitch, W.M. (1969) Biochem. Genet. 3, 99-108.
4. Sankoff, D. (1972) Proc. Nat. Acad. Sci. USA 61, 4-6.
5. Waterman, M.S., Smith, T.F. and Beyer, W.A. (1976) Advan. Math. 20, 367-387.
6. Dayhoff, M.O. (1969) Atlas of Protein Sequence and Structure, National Biomedical Research Foundation, Silver, Springs, Maryland.
7. Korn, L.J., Queen, C.L. and Wegman, M.N. (1977) Proc. Nat. Acad. Sci. USA 74, 4401-4405.
8. The terminal we have used so far is the Retro-graphics VT640, a graphics enhancement to the VT100 computer terminal manufactured by Digital Equipment Corporation. This enhancement is by Digital Engineering, Inc., Sacramento, California, USA. The program requires commands to draw lines, points, a crosshair facility and for plotting and text handling to be independent. A useful addition to the hardware is a hardcopy device which can give paper copies of the screen contents.
9. Gibbs, A.J. and McIntyre, G.A. (1970) Eur. J. Biochem. 16, 1-11.
10. McLachlan, A.D. (1971) J. Mol. Biol. 61, 409-424.
11. Macleod, A.R., Karn, J. and Brenner, S. (1981) Nature 291, 386-390.
12. Maizel, J.V., Jr., and Lenk, R.P. (1981) Proc. Nat. Acad. Sci. USA 78, 7665-7669.
13. Novotny, J. (1982) Nucl. Acids Res. 10, 127-131.
14. Jagadeeswarar, P. and McGuire, P.M., Jr. (1982) Nucl. Acids Res. 10, 433-447.
15. Harr, R., Hagblom, P. and Gustafsson, P. (1982) Nucl. Acids Res. 10, 365-374.
16. Goad, W.B. and Kanehisa, M.I. (1982) Nucl. Acids Res. 10, 247-263.
17. Kanehisa, M.I. (1982) Nucl. Acids Res. 10, 183-196.
18. Staden, R. (1977) Nucl. Acids Res. 4, 4037-4051.
19. Staden, R. (1978) Nucl. Acids Res. 5, 1013-1015.
20. Staden, R. (1979) Nucl. Acids Res. 6, 2610-2610.
21. Staden, R. (1980) Nucl. Acids Res. 8, 817-825.
22. Staden, R. (1980) Nucl. Acids Res. 8, 3673-3694.
23. Staden, R. and McLachlan, A.D. (1982) Nucl. Acids Res. 10, 141-156.