Volume 14 Number 1 1986 Nucleic Acids Research

Assembly of overlapping DNA sequences by a program written in BASIC for 64K CP/M and MS-DOS IBM-compatible microcomputers

Robert E.Johnston[1*], J.M.Mackenzie, Jr.[1] and W.G.Dougherty[2]

Departments of [1]Microbiology and [2]Plant Pathology, North Carolina State University, Raleigh, NC 27695-7615, USA

Received 6 June 1985

ABSTRACT. The SEQALIGN programs[1] described in this report aid in the assembly of up to 100 individual overlapping DNA sequences generated by M-13 subcloning and sequencing methods. The program produces a printout of the aligned sequences presented in register. Use of the program will be facilitated because 1) it is written with the Microsoft BASIC interpreter, 2) sequence data may be entered and edited using WORDSTAR or similar word processing programs, and 3) hardware requirements for execution of the program on CP/M or MS-DOS (IBM-PC compatible) systems are minimal.

INTRODUCTION. Various subcloning strategies (1,2) for determination of DNA sequence have increased the ease and rapidity with which sequence information can be generated. In these procedures, a large DNA clone is digested with restriction endonucleases, fragments are subcloned into an appropriate vector, and the fragments are sequenced individually. Using nucleases with different specificities, a series of overlapping fragments of the original DNA sequence is obtained. However, the location of each fragment within the complete sequence and the orientation of the fragment usually are not readily apparent. As manual assembly of these fragments into the complete sequence is a time consuming and tedious undertaking, several computer programs have been devised which automatically or semi-automatically assemble the fragments into a proposed consensus sequence (3-7). Such programs require access to mainframe or minicomputers. Other programs are offered as sections of expensive commercial software packages. It was our goal to write a series of programs which

[1]SEQALIGN is available for a $50 contribution to the Microbiology Enrichment Fund, Department of Microbiology, North Carolina State University, Raleigh, North Carolina 27695-7615.

would address the assembly problem yet be accessible to virtually anyone using these sequencing techniques. This constraint required that the programs run on small CP/M or MS-DOS based microcomputers, that they be written in BASIC, and that they be capable of handling a sufficient number of fragments to be of significant use.

HARDWARE AND SOFTWARE REQUIREMENTS. These programs were developed on a Televideo 803, a CP/M based microcomputer with 64K RAM and two 368K double-sided, double-density disk drives. We routinely place SEQALIGN and the MBASIC interpreter on one disk and sequence data on the other. However, only one drive was required, as this was sufficient to store the interpreter, the SEQALIGN programs, the sequence fragments, and the intermediate files created during program execution. We also have compiled the programs for use in CP/M based systems and have adapted them with minimal adjustment for use on the standard IBM-PC and PC-compatible microcomputers. The printer used during development was an Epson FX-80. The SEQALIGN programs were written with Microsoft BASIC-80, rev. 5.21.

RESULTS.

Description of the SEQALIGN Programs. The central program, SEQALIGN, displays the main menu shown in figure 1, from which various functions may be selected. Selection of Option 1, "On Screen Assistance", displays a similar menu except that selections from the assistance menu first indicate the use of the given program prior to execution.

Option 2, "WORDSTAR File Conversion", allows a sequence file created or edited in WORDSTAR to be used by the SEQALIGN programs. To establish a WORDSTAR sequence file containing the fragments to be aligned, the file is opened in the non-document mode and given a filename of up to eight characters with no extension. Word processing programs other than WORDSTAR should be compatible with SEQALIGN as long as embedded control characters are not inserted into the file. A reference number for the first fragment is entered on the first line, followed by a carraige return (<CR>). (Each line entered in the WORDSTAR file should end with <CR>.) The second line is the name of the first frag-

```
                SEQUENCE ALIGNMENT PROGRAMS

ON SCREEN ASSISTANCE.............................1

WORDSTAR FILE CONVERSION.........................2
INDEPENDENT SEQUENCE FILE CREATION...............3
SEQUENCE, ALIGNMENT AND MATCH FILE RETRIEVAL.....4
SEARCH, ALIGNMENT & PRINTOUT OF SEQUENCES........5
PRINTOUT OF ALIGNED SEQUENCES....................6
COMPLEMENT GENERATION............................7
SPURIOUS MATCH SUPPRESSION.......................8

RETURN TO OPERATING SYSTEM.......................9
RETURN TO MBASIC COMMAND LEVEL..................10
```

Figure 1. SEQALIGN Main Menu. SEQALIGN functions are summoned by number from this menu. Option 1 provides a similar menu by which function descriptions may be obtained. Upon completion of a particular function, the operator is returned to the main menu.

ment (up to 10 characters in length), usually indicating the restriction enzyme used for its generation. The number of 100-base records occupied by the sequence of the fragment is entered on the third line. For instance, if a fragment contains 350 bases, it would occupy 4 records. The fourth line contains the first 100 bases of sequence, the fifth line the next 100, etc. An * is placed immediately following the last base in the sequence to mark the end of the fragment. The next line begins the information regarding the second fragment which is entered in precisely the same order as above. The WORDSTAR file conversion option stores the sequence information contained in the WORDSTAR file in random access format and simultaneously creates a housekeeping file which allows more direct access to the DNA sequence data. This arrangement allows WORDSTAR to be used for editing of existing sequences as well as addition of new sequences to the file. After any WORDSTAR manipulation, the conversion option reconstitutes the sequence and housekeeping files into the proper format. (This option does not alter an existing complement file. Complement files may be generated as a part of the search and alignment, option 5, or by using option 7, complement generation.)

Option 3, "Independent Sequence File Creation", is provided for direct entry of sequence data without WORDSTAR. The sequence and housekeeping files created by this option are identical to those created by option 2, and may be used interchangeably.

Option 4, "Sequence, Alignment and Match File Retrieval", recalls sequence and complement data from the appropriate files. Alignment and match files are created as part of the search program, option 5, and may be recalled using this option.

Option 5, "Search, Alignment & Printout of Sequences", includes programs which automatically 1) generate complements of each fragment, 2) store the sequences at the 5' and 3' ends of each fragment, 3) use these 5' and 3' sequence blocks in a search of every other fragment and its complement, 4) create a match file containing the results of the search, 5) use this information to order the fragments, and 6) present the assembled sequence both in tabular form (the alignment file) and as a printout of the sequence fragments in register. The programs are flexible. As new fragments are added to the sequence file during the course of a sequencing project, the complement generation and search may be limited to those combinations involving the new fragments. Any additional matches are added to the existing match file on disk, and the composite file is used for alignment.

Option 6, "Printout of Aligned Sequences", uses a previously generated alignment file to produce automatically a printout of the sequence fragments in register. Alternatively, the 5' and 3' positions of particular fragments in the assembled sequence may be entered manually. At the user's option, the contributing fragments at each position of the assembled sequence are compared and differences between the fragments are indicated by *'s on the printout.

Option 7, "Complement Generation", generates complements for single fragments or groups of fragments within a file. This option is especially useful when only one sequence has been edited in the WORDSTAR file.

Option 8, "Spurious Match Suppression", is useful in cases where the program mistakenly identifies a match between two sequences as discussed below. The spurious match can be identified and suppressed so as not to contribute to the alignment. This program also allows the user to suppress all matches involving a particular fragment.

Options 9 and 10 allow exit from SEQALIGN.

Description of the Search and Alignment Algorithims. Potential relationships between sequence fragments of the same orientation are indicated in figure 2. The 5' end of fragment A may be contained within fragment B, the 5' end of fragment B may be contained within fragment A, or the entire sequence of fragment A or B may be contained within the other fragment. If the sequences of both A and B are free of errors, the relationship between A and B may be defined by using a block of nucleotide sequence from the 5' end of A in a search for an identical sequence within B and in the reciprocal, using a 5' block of sequence from B to search within fragment A. The only proviso would be that the 5' search blocks be sufficiently large to avoid finding spurious matches based on random identities. Unfortunately, insertion, deletion and substitution errors occur in the generation of nucleotide sequence data. Moreover, the error frequency is highest at the ends, especially at the 3' end where the sequence is deduced from the gel position of poorly resolved, long DNA fragments. This necessitates the comparison of the two fragments along the length of the overlap to find the best alignment. This problem has proved to be an impediment to the development of DNA sequence assembly programs on smaller computers (8). In programs designed for larger computers, the problem may be solved by insertion of gaps and assignment of gap penalties to arrive at an optimum arrangement of the fragments in question. We have utilized a simpler procedure in which multiple, adjacent search blocks are selected: 3 from the 5' end of each fragment, 4 from the 3' end and 1 internal block beginning at base 101. These are used independently to search for identities within the other fragments and their complements, and the matches found are used to calculate the positions of one fragment within another. In the clones we have sequenced (9-12), a search block size of 10-12 bases effectively located all the legitimate overlaps with only an occassional spurious match. Decreasing the search block size below 10 did not increase the number of legitimate overlaps identified using several test sequence files; increasing the search block size to greater than 12 avoided all spurious matches but resulted in missing some legitimate overlaps. SEQALIGN allows the user to designate the search block size and to suppress

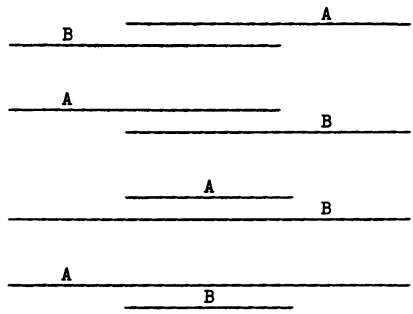

Figure 2. Potential 5' Overlaps Between Two Fragments with the Same Orientation. Possible relationships of two fragments, A and B, are illustrated. 5' ends of the fragments are to the left.

any spurious matches which may occur.

In the example in figure 2, fragments A and B could be the original sequences as entered. Alternatively, one or both fragments could be the complement of the entered sequence. SEQ-ALIGN uses search blocks from the entered sequence and searches both the entered sequence and the complementary sequence of each fragment. Of the categories of match which are found, three types are utilized in the subsequent alignment algorithim. These are 5' matches found within "+" fragments (the sequences as entered), 5' matches found within "-" fragments (the complements of the entered sequences), and 3' matches found within "-" fragments.

To find and order related fragments, the information regarding the 5' "+" matches is placed into two related, two-dimensional array variables, R(x,y) and S(x,y), for each fragment in the file. Ultimately, fragments related by 5' "+" matches and their relative 5' positions will be grouped together in R(x,y) and S(x,y) variables, respectively. In each case, x equals the number of the referenced fragment. R(x,1) is set equal to x, and the variables R(x,2...n) carry the numbers of fragments which are related to the referenced fragment by 5' "+" matches. In cases where no 5' "+" matches are found for x, R(x,1)=0. S(x,2...n) are the 5' positions of R(x,2...n) within x. S(x,1)=1 by definition except where no 5' "+" matches are found for x in which case S(x,1)=0. Therefore, for each x-related fragment contained in

R(x,2...n), the fragments contained in R(R(x,2...n),2...m) also must be related to x. For R(x,2) as an example, the program places the fragments in R(R(x,2),2...m) into R(x,n+1...n+m-1) if the additional fragments do not duplicate fragments already contained in R(x,1...n). Likewise, 5' position values in S(R(x,2),2...m) are placed in S(x,n+1...n+m-1) after displacement by the value in S(x,2) which gives the 5' position of R(x,2) in relation to x. R(R(x,2),1...m) and S(R(x,2),1...m) then are set equal to 0, and the program proceeds to evaluate the fragments in R(R(x,3),2...m). After related fragments and their relative 5' positions are grouped together in R and S variables, the order of fragments within each group (in a R variable) is derived by sorting based on the 5' positions in the relevant S variable. Fragments which are not related to any other fragment by 5' "+" matches also are assigned to R and S variables and are treated in subsequent steps as single member groups.

The preliminary groups generated from the 5' "+" match data in the above step will be of two types. One type will include fragments derived from one strand of the original cloned DNA, and the other type will have been derived from the opposite strand. The program searches the 5' and 3' "-" matches to relate the two types of preliminary groups. An alignment file (figure 3) is produced which contains the position of each fragment within the assembled sequence for a given group of related fragments.

Presentation of Aligned Sequences. The result of the alignment is presented as in figure 4, with each fragment printed in register according to its position in the assembled sequence. A consensus sequence is easily derived by inspection, as errors of insertion, deletion and substitution are readily apparent. On occassion, the search program will find a spurious match based on a short stretch of identical bases in otherwise unrelated fragments. This results in the misalignment of these fragments and is obvious when scanning the printout of the aligned sequences. The causative spurious match can be identified from the registered printout, suppressed and realigned in option 8.

Program Capacity. SEQALIGN is designed to process a maximum of

SEQUENCE FILE = A:TEST12

	SEQUENCE NUMBER	5' POSITION, ASSEMBLED SEQUENCE	3' POSITION, ASSEMBLED SEQUENCE
GROUP 1	-6	1	225
	11	24	249
	12	24	250
	13	28	251
	2	219	467
	14	235	473
	-5	301	558
GROUP 2	15	1	234
	1	6	220
GROUP 3	10	1	121
	9	32	247
	-8	123	358
GROUP 4	3	1	233
GROUP 5	4	1	229
GROUP 6	7	1	219

Figure 3. Alignment File Printout. After identifying related fragments and their relative positions, SEQALIGN presents the data in tabular form. Positive fragment numbers indicate the orientation of the fragment as entered; negative numbers indicate complements of entered sequences.

100 sequence fragments. However, the sequence information is stored on disk until used by the program, allowing individual fragments to be quite large. Because all the search blocks are stored in machine memory throughout the search routine, the size limitation for an individual fragment being searched depends on

Table 1. Time Requirements

	100 Fragments	15 Fragments
CP/M		
Interpreter	4.5 hr[a]	8.5 min
Compiled	50 min	4.0 min
IBM-PC		
Interpreter	3.5 hr	10.5 min

[a]The time required for the completion of the automatic portions of SEQALIGN were determined. This included complement generation, search, alignment and printout of the sequence fragments in register.

the number of fragments in the file. We have used the program successfully in the alignment of a 300 base fragment within a larger fragment of almost 12,000 bases.

The program execution times for the automatic portions of SEQALIGN (complement generation, search, alignment and printout) are given in table 1. With the 100 fragment file (approximately 30,000 nucleotides), the printout alone required 20 min. Using a complliled version of SEQALIGN to analyse the 15 fragment file, approximately 45 sec elapsed before the execution speed of the program became limited to the speed of the printer. In order to conserve space in machine memory, SEQALIGN makes considerable use of disk storage. The resulting disk I/O makes a substantial contribution to the time required for execution. By placing compiled SEQALIGN and the sequence data on a 500K RAM disk, computation time for the 100 fragment file was reduced to 20 min.

DISCUSSION.

The SEQALIGN programs offer two advantages. First, they allow the entry and editing of nucleotide sequence data with WORDSTAR. Second, their hardware and software requirements are minimal, thus making them highly accessible to most molecular biologists. Although designing the programs for use in a minimal environment has necessitated some loss of execution speed, we feel that the increase in accessibility more than compensates. Moreover, we believe that the loss of execution speed is insignificant relative to the total data acquistion time in a sequencing project. Because the complement generation, search, alignment and printout routines are automatic, SEQALIGN may be conveniently left to run overnight.

SEQALIGN does not automatically produce a proposed consensus sequence but rather provides a printout of the contributing fragments in register. We have found that the consensus base at each position of the assembled sequence may be easily discerned from the data presented in the registered format. Where discrepancies exist, the choice of a consensus base may be deduced after allowance for the particulars of each contributing fragment. As a practical matter, we simply read portions of the assembled sequence from individual fragments as dictated by vis-

```
SEQUENCE FILE = A:TEST12.BF


                                  1
SEQ. # 6 C' KTAQ10        TCTATGTCTG TGTTGTAGAG AAAGATCAAA AGGATGATAA AGTCGCTGAG
SEQ. # 11 KS28                                GATCAAA AGGATGATAA AGTCGCTGAG
SEQ. # 12 KF18                                GATCAAA AGGATGATAA AGTCGCTGAG
SEQ. # 13 KSAL14                                  AAA AGGATGATAA AGTCGCTGAG

                                 51
SEQ. # 6 C' KTAQ10        CAGGCTTCAA AGGATAGGGA TGTCAATGCT GAACTTCAGG AACATTCTCA
SEQ. # 11 KS28            CAGGCTTCAA AGGATAGGGA TGTCAATGCT GGAACTTAGG AACATTCTCA
SEQ. # 12 KF18            CAGGCTTCAA AGGATAGGGA TGTCAATGCT GGAACTTCAG GAACATTCTC
SEQ. # 13 KSAL14          CAGGCTTCAA AGGATAGGGA TGTCAATGCT GGAACTTCAG GAACATTCTC

                                101
SEQ. # 6 C' KTAQ10        GTTCCACGAA TAAATGCTAT GGCCACAAAA CTTCAATATC CAAGGATGAA
SEQ. # 11 KS28            GTTCCACGAA TAAATGCTAT GGCCACAAAA CTTCAATATC CAAGGATGAA
SEQ. # 12 KF18            AGTTCCACGA ATAAATGCTA TGGCCACAAA ACTTCAATAT CCAAGGATGA
SEQ. # 13 KSAL14          AGTTCCACGA ATAAATGCTA TGGCCACAAA ACTTCAATAT CCAAGGATGA

                                151
SEQ. # 6 C' KTAQ10        AGGGGAGGTA GTTGTAAACT TGAATCACCT TTTAGGATAC AAGCCACAGC
SEQ. # 11 KS28            AGGGGAGGTA GTTGTAAACT TGAATCACCT TTTAGGATAC AAGCCACAGC
SEQ. # 12 KF18            AAGGGGAGGT AGTTGTAAAC TTGAATCACC TTTTAGGATA CAAGCCACAG
SEQ. # 13 KSAL14          AAGGGGAGGT AGTTGTAAAC TTGAATCACC TTTTAGGATA CAAGCCACAG

                                201
SEQ. # 6 C' KTAQ10        AAATTGACTT GTCAAATGCT CTCGA*
SEQ. # 11 KS28            AAATTGACTT GTCAAATGCT CGAGCCACAC ATGAGCAGTT TGCCGCGTC*
SEQ. # 12 KF18            CAAATTGACT TGTCAAATGC TCGAGCCACA CATGAGCAGT TTGCCGCGTG
SEQ. # 13 KSAL14          CAAATTGACT TGTCAAATGC TCGAGCCACA CATGAGCAGT TTGCCGCGTG
SEQ. # 2 KTAQ2                          TC GAGGCCACAC ATGAGCAGTT TGCCGCGTGG
SEQ. # 14 TEST 6                                    GCAGTT TGCCGCGTGG

                                251
```

Figure 4. Assembled Sequence Presented as Printout of Ordered Fragments in Register. A portion of a registered printout is shown. Deletion, insertion and substitution errors, such as those at positions 82 and 88 of fragments 6 C' and 11, respectively, are detected by visual inspection. The concensus sequence is read as denoted by the solid line.

ual inspection of the registered printout (see figure 4). Currently, we are adding a program to SEQALIGN which will produce a WORDSTAR file containing the consensus sequence upon manual entry of the base numbers to be read from each fragment.

ACKNOWLEDGEMENTS. This report is paper no. 9922 of the Journal Series of the North Carolina Agricultural Research Service (NCARS), Raleigh, North Carolina 27695-7601. This work was supported by NCARS, NIH grant AI-19433, and NSF grant 83-09249.

*To whom correspondence should be addressed

REFERENCES.

1. Messing,J. (1982) in Genetic Engineering: Principles and Methods, Setlow,J.K. and Hollander,A. Eds., pp. 19-35, Plenum, New York.
2. Messing, J. (1983) Methods in Enzymology 101, 20-78.
3. Gingeras,T.R., Milazzo,J.P., Sciaky,D. and Roberts,R.J. (1979) Nuc. Acids Res. 7, 529-545.
4. Staden,R. (1980) Nuc. Acids Res. 8, 3673-3694.
5. Clayton,J. and Kedes,L. (1982) Nuc. Acids Res. 10, 305-321.
6. Staden,R. (1982) Nuc. Acids Res. 10, 4731-4751.
7. Peltola,H., Soderlund,H. and Ukkonen,E. (1984) Nuc. Acids Res. 12, 307-321.
8. Malthiery,B., Bellon,B., Giorgi,D. and Jacq,B. (1984) Nuc. Acids Res. 12, 569-579.
9. Parks,T.D., Dougherty,W.G., Levings III,C.S. and Timothy,D.H. (1984) Plant Physiol. 76, 1079-1082.
10. Parks,T.D., Dougherty,W.G., Levings III,C.S. and Timothy,D.H. (1985) Current Genetics, in press.
11. Allison,R.F., Sorenson,J.C., Kelly,M.E., Armstrong,F.B. and Dougherty,W.G. (1985) Proc. Natl. Acad. Sci., USA, in press.
12. Dougherty,W.G., Allison,R.F., Parks,T.D., Johnston,R.E., Field,M. and Armstrong,F.B. (1985) Virol., in press.