Improvements to a program for DNA analysis: a procedure to find homologies among many sequences

Cary Queen[1]*, Mark N.Wegman[2] and Laurence Jay Korn[3]

[1]Laboratory of Biochemistry, National Cancer Institute, NIH, Bethesda, MD 20205, [2]Thomas J. Watson Research Center, International Business Machines, Yorktown Heights, NY 10598, and [3]Department of Genetics, Stanford University School of Medicine, Stanford, CA 94305, USA

Received 12 October 1981

ABSTRACT

We have devised an algorithm for finding partial homologies among a set of nucleotide sequences. The algorithm and other improvements have been incorporated into a commonly used computer program for the analysis of sequence data.

INTRODUCTION

The discovery of rapid techniques for sequencing DNA has been paralleled by the development of computer programs for analyzing DNA sequences (for review see ref. 1). We have previously described one of the most extensive such programs (2), which has since been expanded and modified (3,4). This computer program has been used as an aid in the analysis of transcription termination sites (2,5), origins of DNA replication (6), satellite DNA (7), bacterial transformation sequences (8), splice junctions (9), small nuclear RNAs (10,11), insertion elements (12), SV40 variants (13), retroviruses (14), and genes for tRNA (15), ribosomal proteins (16), 5S ribosomal RNA (17), large ribosomal RNAs (18-20), β-globin (21), immunoglobulins (22), hormone precursors (23), ovomucoid (24), dihydrofolate reductase (25), chorion proteins (26), restriction enzymes (27), and the tryptophan operon (28-30).

Our program (2) has the ability to compare two DNA sequences for regions of partial homology. It is often important, however, to search a whole set of sequences for homologous regions appearing in most or all of them. For example, promoters for E. coli RNA polymerase are characterized by partial homologies in the -10 region (Pribnow box, ref. 31,32) and -35 region, with each region consisting of 6-7 nucleotides (reviewed in ref. 33). Similarly, one goal of current research on eucaryotic pro-

moters is to find functionally relevant sequence homologies (e.g., the "Hogness box," ref. 34).

Parital homologies among a set of sequences are not readily detected by a computer program that compares sequences two at a time, because each pair of sequences contains many homologies not shared by the others. We have therefore added to our program a procedure for detecting multi-sequence homologies, based on an algorithm that analyzes all the sequences simultaneously. We have also added to the program procedures for determining fragment lengths produced by multiple restriction enzyme digests and for performing other convenient functions. The expanded program is available from L.J.K. upon request.

MATERIALS AND METHODS

The program was written in the language PL/I and developed using the NIH extended WYLBUR editor. The completed program has been compiled using an IBM optimizing PL/I compiler and run on an IBM 3033 computer.

RESULTS AND DISCUSSION

A procedure for finding homologies. We present first an example of the new procedure's applications; a description of the homology algorithm and instructions for using the program are given below. We applied the program to 7 of the earliest well-characterized promoters for E. coli RNA polymerase: λ P_R, T7 A3, lac, gal, trp, $tRNA^{Tyr}$, and SV40 (for sequences and references, see ref. 33). The program was required to try to find, in at least 5 out of the 7 promoter sequences, 6-nucleotide long segments (6-mers) that differ from a consensus 6-mer by no more than 1 nucleotide. Moreover, the distances of the 6-mers from the respective RNA startsites were required not to differ by more than 4 nucleotides. Two such sets of homologous 6-mers are shown in Figure 1 as detected by the computer. One homology clearly corresponds to the -35 promoter region and the other to the -10 region (33). Two other homologous sets, overlapping the ones shown, were also found.

Analogous results were obtained when other selections of promoters (33) were analyzed or when somewhat different restrictions were made on the number of mismatches allowed and the number of sequences missing the homology. However, when the criteria for matching were made too stringent,

1	6	TTGACT	1		
2	5	TTGACA	2	28	TACGAT
3	5	TTTACA	3	29	TATGTT
4			4	29	TATGCT
5	5	TTGACA	5		
6	6	TTTACA	6	28	TATGAT
7			7	28	TATAAT
CONSENSUS		TTGACA	CONSENSUS		TATGAT

Figure 1. Homologies found by the program. The promoters compared were (33): 1, λP_R; 2, T7A3; 3, lac; 4, gal P2; 5, trp; 6, $tRNA^{Tyr}$; 7, SV40. The first numbers indicate the sequence and the second numbers indicate the position in the sequence (position 1 is 40 nucleotides before the RNA startsite). In each homology, the sequences for which there is no entry did not contain a 6-mer differing from the consensus 6-mer by 1 or fewer nucleotides.

no homologies were found because of the intrinsic variation among promoters. Hence, the program can probably best be applied to a new set of sequences by trying several different homology criteria and by analyzing the sequences in medium-sized groups. Applications of the program to eucaryotic promoter sequences will be presented elsewhere.

Other improvements to the program. We added a procedure to determine the fragment sizes produced by a double or multiple restriction digest of a DNA molecule, which can be specified as linear or circular. The results are displayed in an easily readable table, as shown in Figure 2.

The original program could translate a DNA sequence into amino acids in any reading frame (3). A modification allows the printing of all but a specified set of amino acids to be suppressed. In this way particular codons, such as initiation or termination codons, can be highlighted (see below).

We have also added a routine that lists the digestion products of an RNA sequence by pancreatic or T1 RNAse, an aid to analyzing RNA fingerprints. A final new procedure combines specified parts of two nucleotide sequences into a third sequence, thus simulating ligation or splicing.

The homology algorithm. The algorithm depends on 6 variables (Table 1): K, L, M, N, S, T. Given L sequences of length T, part of the algorithm finds each oligonucleotide of length N (N-mer) such that at least K of the sequences contain an N-mer not differing from it by more than M nucleotides, with the positions of the K N-mers not differing by more than S. We first outline this part of the algorithm in the case S=T, so there are no restrictions on the relative positions of the N-mers.

	# OF SITES	SITES	FRAGMENTS	FRAGMENT ENDS	
1 BAM H1 (GGATCC)					
2 PVU 2 (CAGCTG)					
	4				
		270 (2)	2007	3506	270
		1716 (2)	1446	270	1716
		2533 (1)	973	2533	3506
		3506 (2)	817	1716	2533

Figure 2. Double restriction enzyme digest of SV40 genome calculated by the program. The enzymes used are printed and numbered at the left. All sites for the enzymes in the SV40 sequence (35,36) are found by the program and listed consecutively, with the enzyme that cuts at a site noted in parentheses. In a separate list, the fragments produced by the double digest are arranged in descending order by length. The sites at the ends of each fragment are printed next to it.

For each sequence, an array is created with 4^N entries corresponding to the 4^N possible N-mers, and the values of all the entries of the array are set to 0. For each N-mer occurring in any of the sequences, the corresponding entry in the array belonging to that sequence is increased by 1. Moreover, the entry corresponding to each other possible N-mer that differs at M or fewer positions from the N-mer being considered is also increased by 1. (For example, if N=4 and M=1, there will be 12 such

TABLE I. Parameters of the Homology Procedure

Algorithm Variable	Program Parameter	Function of the Variable (Parameter)
L		Number of sequences compared
T		Length of sequences compared
N	SEARCH	Length of the homologies sought
M	MISSED	Number of mismatches allowed between homologous N-mers
K	MINSEQ	Minimum number of sequences required to have a homology
S	SHIFT	Maximum difference in relative position of homologous N-mers

N-mers: 3 differing from the given N-mer in the first position, 3 differing in the second position, etc.). All the N-mers occurring in the L sequences are systematically treated this way in the following order: first the N-mers beginning with the first nucleotides of the sequences, then the N-mers beginning with the second nucleotides, and so forth. When this process is complete, an entry in the array corresponding to a sequence will be greater than 0 if the corresponding N-mer appears in that sequence with at most M changes.

The algorithm uses an additional array A with 4^N entries assigned to N-mers: the value of each entry is equal to the number of the other arrays whose corresponding entry is greater than 0. The array A is constructed simultaneously with the others, by adding 1 to an entry whenever one of the corresponding entries in the arrays belonging to sequences becomes greater than 0. When an entry in the array A becomes equal to K, the corresponding N-mer is added to a list. At the conclusion of the process, this is the desired list containing each N-mer such that at least K of the sequences contain an N-mer not differing from it at more than M positions (Note that a single set of K N-mers in the sequences can give rise to many entries on the list, differing from each other and the K N-mers at no more than M positions).

The algorithm is easily extended to the case when S<T by arranging for the various arrays to progressively register only the N-mers contained in a section of the sequences S nucleotides long. Specifically, as soon as appropriate entries of the arrays are increased to reflect the N-mers beginning in the I-th positions of the sequences, other appropriate entries are decreased to reflect the N-mers beginning in the (I-S-1)-th positions that must no longer be counted. Hence, the S-nucleotide long region in which the N-mers can be found "moves along" the sequences. The process is completed in a time proportional to only the first power of the total sequence length.

To complete the algorithm, it is necessary to retain for each N-mer listed a corresponding set of homologous N-mers contained in at least K of the sequences. For this purpose, another array with 4^N entries is created for each sequence. These arrays are filled in simultaneously with the former arrays. Each entry progressively records the position in the corresponding sequence of the last N-mer (if any) to differ from the entry's associated N-mer at no more than M positions (i.e., the last N-mer to have contributed to the corresponding entry in the former arrays). When

an N-mer is added to the list described above, the positions in K or more sequences of the last N-mers to have differed from it at M or fewer positions are read from the new arrays and listed with it. Moreover, this set of positions is compared with all sets of positions already on the list, and if it is identical to a previous one (see above), the new N-mer is eliminated. Indeed, it is the set of K or more homologous N-mers in the sequences that is actually of interest, and that need not be duplicated. Finally, a consensus N-mer is calculated for each set of K or more N-mers whose positions are on the list, and each set of N-mers is printed with its consensus (Figure 1).

Use of the program. The basic input format for our program has been described (3). When using the new homology procedure, only the sequences to be compared and their names should be placed in the section for entering sequences. Then a line reading 'COMPARE' 0 should be put in the section for entering the procedures applied to individual sequences, in order to activate the homology routine. Only the characters A, C, G, T and U in the sequences will participate in correct matches. If the sequences are not all of the same length, the program will "pad" the shorter ones on the left in order to make them so; this will change the numbering of their nucleotides.

Several program parameters should be used to direct the homology procedure by fixing the values of the algorithm variables (Table 1). The parameter SEARCH defines the length of the N-mers sought, i.e., sets N itself, while the parameter MISSED sets the number of mismatches allowed between N-mers. As currently implemented, the program only allows two values for SEARCH, namely SEARCH=4 with MISSED=0 or 1, and SEARCH=6 with MISSED=0, 1 or 2. These values are sufficient to find homologies of all lengths, because longer homologies will consist of clusters of shorter ones. The parameter MINSEQ specifies the minimum number of sequences required to contain homologous N-mers; the parameter SHIFT specifies the maximum difference in the relative positions of the N-mers. For instance, in the example given above (Figure 1), the parameters were as follows: SEARCH=6, MISSED=1, MINSEQ=5, SHIFT=4.

To determine the fragment lengths produced by a multiple restriction digest, the enzyme recognition sites and names should be placed in the usual input section (3) immediately preceded and followed by lines reading 'DO' 'MULTIPLE'. Several sets of enzymes for multiple digests, each set surrounded by separate lines reading 'DO' 'MULTIPLE', may be included to-

gether in the input section along with single enzymes. (The same enzyme may appear more than once). Then a line reading 'NAME' 14 0 in the procedure section (3) will specify that the digest calculations be performed on the named sequence (Figure 2).

To use the modified translation feature, the parameter SIGNAL should be set equal to a list of amino acids (possibly including an "END") in single quotes. Upon invoking the translation procedure with a line 'NAME' 13 0, only those codons corresponding to the listed amino acids will be translated. For example, setting SIGNAL='END' specifies that only termination codons be translated and SIGNAL='MET END' specifies that initiation and termination codons be translated.

The procedure to list pancreatic and T1 RNAse digestion products of an RNA sequence is invoked as usual; its procedure number is 17. Finally, to combine two sequences a line containing both of their names should be put in the input section normally used to compare two sequences for homologies, preceded by a line reading 'LINK' 'LINK'. Lines specifying limits and orientations of the sequences in the usual way (3) may be inserted between these two lines in order to combine only parts of the sequences. The combined sequence generated by this procedure should be stored in the computer system so that it can be analyzed by the program.

ACKNOWLEDGMENTS

We would like to thank W. Tucker, B. Sollner-Webb, D. Wallace and R. Young for commenting on the manuscript and J. Warwick for efficient typing.

This work was supported in part by American Cancer Society Grant CD 122 to LJK.

*Current address: Center for Cancer Research, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

REFERENCES

1. Gingeras, T.K. and Roberts, R.J. (1980). Science 209, 1322-1325.
2. Korn, L.J., Queen, C.L. and Wegman, M.N. (1977). Proc. Natl. Acad. Sci. USA 74, 4401-4405.
3. Queen, C.L. and Korn, L.J. (1980). Methods Enzymol. 65, 595-609.
4. Sege, R., Söll, D., Ruddle, F.H. and Queen, C. (1981). Nucleic Acids Res. 9, 437-444.

5. Küpper, H., Sekiya, T., Rosenberg, M., Egan, J. and Landy, A. (1978). Nature 272, 423-428.
6. Martens, P.A. and Clayton, D.A. (1979). J. Mol. Biol. 135, 327-351.
7. Hsich, T. and Brutlag, D. (1979). J. Mol. Biol. 135, 465-481.
8. Danner, D.B., Deich, R.A., Sisco, K. and Smith, H.O. (1980). Gene 11, 311-317.
9. Trapnell, B.C., Tolstoschev, P. and Crystal, R.G. (1980). Nucleic Acids Res. 8, 3659-3672.
10. Epstein, P., Reddy, R. and Busch, H. (1981). Proc. Natl. Acad. Sci. USA 78, 1562-1566.
11. Wise, J.A. and Weiner, A.M. (1980). Cell 22, 109-118.
12. Nisen, P., Purucker, M. and Shapiro, L. (1979). J. Bact. 140, 588-596.
13. McCutchan, T., Singer, M. and Rosenberg, M. (1979). J. Biol. Chem. 254, 3592-3597.
14. Shimotohno, K., Muzutani, S. and Temin, H. (1980). Nature 285, 550-554.
15. Hovemann, B., Sharp, S., Yamada, H. and Söll, D. (1980). Cell 19, 889-895.
16. Post, L.E., Strycharz, G.D., Nomura, M., Lewis, H. and Dennis, P.P. (1979). Proc. Natl. Acad. Sci. USA 76, 1697-1701.
17. Korn, L.J. and Brown, D.D. (1978). Cell 15, 1145-1156.
18. Sollner-Webb, B. and Reeder, R.H. (1979). Cell 18, 485-499.
19. Young, R.A., Macklis, R. and Steitz, J.A. (1979). J. Biol. Chem. 254, 3264-3271.
20. Long, E.O., Rebbert, M.L. and Dawid, I.B. (1981). Proc. Natl. Acad. Sci. USA 78, 1513-1517.
21. Konkel, D.A., Tilghman, S.M. and Leder, P. (1978). Cell 15, 1125-1132.
22. Hieter, P.A., Max, E., Seidman, J.G., Maizel, J.V. Jr. and Leder, P. (1980). Cell 22, 197-207.
23. Nakanishi, S., Inoue, A., Kita, T., Nakamura, M., Chang, A.C.Y., Cohen, S.N. and Numa, S. (1979). Nature 278, 423-427.
24. Buell, G.N., Wickens, M.P., Carbon, J. and Schimke, R. (1979). J. Biol. Chem. 254, 9277-9283.
25. Nunberg, J.H., Kaufman, R.S., Chang, A.C.Y., Cohen, S.N. and Schimke, R.T. (1980). Cell 19, 355-364.
26. Jones, C.W. and Kafatos, F.C. (1980). Cell 22, 855-867.
27. Newman, A., Rubin, R.A., Kim, S. and Modrich, P. (1981). J. Biol. Chem. 256, 2131-2139.
28. Nichols, B.P., Miozzari, G.F., Van Cleemput, M., Bennett, G. and Yanofsky, C. (1980). J. Mol. Biol. 142, 503-517.
29. Nichols, B.P., Van Cleemput, M. and Yanofsky, C. (1981). J. Mol. Biol. 146, 45-54.
30. Wu, A.M., Chapman, A.B., Platt, T., Guarente, L.P. and Beckwith, J. (1980). Cell 19, 829-836.
31. Pribnow, D. (1975). Proc. Natl. Acad. Sci. USA 72, 784-788.
32. Schaller, H., Gray, C. and Herrmann, K. (1975). Proc. Natl. Acad. Sci. USA 72, 737-741.
33. Rosenberg, M. and Court, D. (1979). Ann. Rev. Genet. 13, 319-353.
34. Goldberg, M. Ph.D. Thesis. Stanford University (1979).
35. Fiers, W., Contreras, R., Haegeman, G., Rogiers, R., Van de Voorde, A., Van Heuverswyn, H., Van Herreweghe, J., Volkaert, G. and Ysebaert, M. (1978). Nature 273, 113-119.
36. Van Heuverswyn, H. and Fiers, W. (1979). Eur. J. Biochem. 100, 51-60.