Volume 12 Number 1 1984 Nucleic Acids Research

Simplified computer programs for search of homology within nucleotide sequences

Manfred Kröger* and Anneliese Kröger-Block

Institut für Biologie III, Universität Freiburg, Schänzlestrasse 1, D-7800 Freiburg im Breisgau, FRG

Received 25 August 1983

ABSTRACT

Four new computer programs for search of homology within nucleotide sequences are presented. The main scope of the program design is flexibility, independence of sequence length and the capability to be used by any molecular biologist without any prior computer experience. The programs offer a linear search, a search for maximal identity, an alignment along a given sequence and a search based on homology within the amino acid coding capacity of nucleotide sequences. The language is Fortran V. Copies are available on request.

INTRODUCTION

A number of fairly sophisticated computer programs for search of homology within nucleic acids are described in the literature[1-9]. They deal with two very important questions in molecular biology, the evolutionary or phylogenetic aspect and the search for identical sequences during the course of DNA sequencing. Some programs are especially designed for a quick screening of large data collections, some introduce certain rules regarding proposed mechanism of evolution. Some of these programs either restrict the search sequences to 1000 or 4000 nucleotides or need long times for calculation or use rules which are still in discussion. In most cases the answers given by the computer are seemingly perfect and the user often can not check on the calculations, since they are based on qualified mathematics.

When we started to work out our concept for homology search programs, we did not want to write just another highly sophisticated program, but we wanted to provide a simple, almost self-defining program. This is of especial importance, since this should be a program which any molecular biologist can use di-

rectly without any specific knowledge of computer programs. In addition our program should still allow a personal interpretation of the data. Along this line we have developed already a program for sequence interpretation[10].

We started with a search program for identical or almost identical sequences within a given sequence. Later a program looking for maximal homology was designed, which was then enlarged to give an alignment similar to the well known graphical alignment programs[3,5]. Finally, we designed a program search for homology based on amino acid coding capabilities of nucleic acid sequences.

GENERAL SCOPE AND PROGRAM ARCHITECTURE

All programs use the same kind of data, which are different nucleotide sequences with unlimited extent, in order to handle easily sequences like the lambda sequence of 48,502 nucleotides.

They allow direct selection of certain areas within an unlimited sequence both for active and passive search.

They automatically use the active search sequence - defined as the smaller sequence - in both orientations.

They allow a search using both any sequence within the passive search sequence and any foreign sequence.

The active search sequence may be already stored within the computer or may be directly typed in.

They allow any type of data within the active and passive search sequence. Only those letters representing the one letter base group code according to Stockwell[11] are accepted for search (see Fig. 1). All other letters, signs or numbers are treated as undefined positions. Undefined positions may generally be represented as 'N' or as '-'. All blanks are neglected, thus sequences are always treated as continuous sequences.

All calculations regarding the extent of homology are according to the value matrix for sequence homology given in Fig. 1.

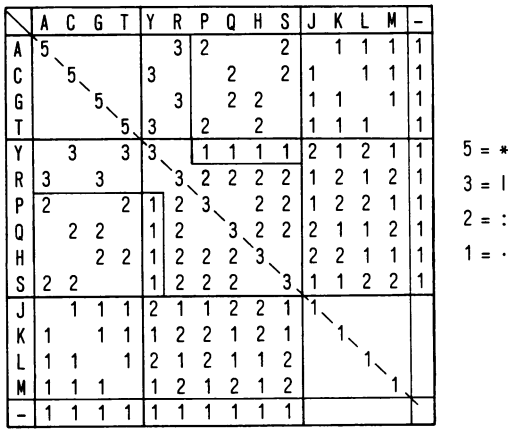
All programs run on simple computer hardware and do not need any specific graphical display.

All calculations result in exact numbers for the region of homology.

All programs are written in Fortran V as dialogue programs.

VALUE MATRIX FOR SEQUENCE HOMOLOGY

The central part of our homology program is a matrix for score values. This matrix is shown in Fig. 1 and uses the one letter group code introduced by Stockwell[11]. This code covers all theoretically possible combinations of certain or uncertain positions within any DNA sequence. The practical background of such a code is the DNA sequencing technique. Often only imperfect information can be read from the gels, such as C or T, which equals code Y; or the information 'no G', which equals code L. The same information can be drawn from a given protein



One letter Base Group Codes (acc. Stockwell)

Code	R	Y	P	Q	H	S	J	K	L	M	-
Bases indicated	A	T	A	G	G	A	C	A	A	A	A
	G	C	T	C	T	C	G	G	C	C	C
							T	T	T	G	G
											T
but not							A	C	G	T	

Figure 1. Value matrix for sequence homology. All calculations regarding the extent of homology within the described programs use this matrix, which is described in detail in the text. The symbols for the point scores are shown at the right side. The one letter base group code is according to Stockwell[11].

regarding the degeration of the genetic code.

The score values consider the extent of homology of each nucleotide pair. We chose a point score of 0, 1, 2, 3 or 5 points. Any undefined position versus any undefined position is 0. Any undefined position versus any defined position is 1. One point is also given for any partial correspondence like J = no A, if e.g. C, G or T is in the search sequence. 2 points are given for an extended partial correspondence like J = C, T or G if e.g. Y = C or T is in the search sequence. This is because they only differ in the fact that J additionally allows G. 3 points are given for any identity in the R, Y, P, Q, H, S group or for the fact that e.g. C is 50% true for Y = C or T. The next higher point score may have been 4 points. But we chose 5 points in order to reward the perfect match with an extra bonus.

LINEAR HOMOLOGY SEARCH

This program compares a given active sequence across its en-

tire length with any part of the passive search sequence of equal length. According to the value matrix given in Fig. 1 the program calculates an average score value for each individual comparison. In order to give an extra bonus for stretches of unbroken identity, every step prolonging a perfect match adds another point to the score. The score is finally divided by the number of nucleotides in the active search sequence. This average score is maximal 5 points for total identity.

Within the starting routine for this linear homology search program the user defines the area for passive search first, and then the area for active search together with the minimal score value. If for instance the total active search sequence should be compared with a minimum score of 3.5, the command is 1,0,3.5. The program protocolls every match with a score of 3.5 or more. If no match with this score could be found, the maximal score is protocolled instead. For an example see Fig. 2.

In order to provide an optical distinction between the different score values the program prints passive and active search sequences underneath each other separated by a * sign for perfect match - 5 points -, a I for 3 points, a : for 2 points and a . for 1 point. No homology will lead to an empty space between the two lines. All sequences are defined through a protocoll of their exact location. If the active search sequence matches in the reverse orientation a R is printed behind the average score.

MAXIMAL HOMOLOGY SEARCH

This program searches for stretches of uninterrupted homology. It allows the comparison of an unlimited passive search sequence and an active search sequence interval of up to 1600 nucleotides.

Scores of 5, 3, or 2 points according to the value matrix in Fig. 1 are treated identically and prolongate the stretch of un-

```
}1,0,1.1
     1              40
     AGTGTTGACTHT-AKTAKGGTCGTACACTTGATCAAGCTQ
     *** ******|*.*.**.********** ****** ***:
     AGTCTTGACTHTCAKTAAGGTCGTACACATGATCATGCTG
     1              40              Average score = 4.3
     40              1
     QAGCTTGATCAAGTGTACGACCLTALT-ASAGTCAACACT
       *****       .** *  * **. .*   ***
     AGTCTTGACTHTCAKTAAGGTCGTACACATGATCATGCTG
     1              40              Average score = 1.1 R
```

Figure 2. Example of a printout provided by the Linear Homology Search Program. The same sequences are compared in direct and inverse orientation, since the average score of 1.1 is fairly low. Any higher score would have only provided the upper result.

interrupted homology. A score of 1 or 2 will, however, break off the search.

The program searches either for a maximum or for a certain number of nucleotides with uninterrupted homology. For maximum search a code number 0 is required. If any certain number is typed in instead, all stretches with at least this number of uninterrupted homology is provided. If no homology with the given number can be found, the program protocolls the highest number found. All runs can be repeated with any lower number. In this case the already provided information will not be given again. All printouts provide the homologous sequence and the appropriate passive search sequence together with their exact location, a line of either *, I or : symbols, an indication of the number of uninterrupted homology, a calculation of the average score value and an indication whether the reverse orientation is found.

An example of a printout provided by the maximal homology search program is given in Fig. 3.

ALIGNMENT PROGRAM FOR MAXIMAL HOMOLOGY

This program is an enlargement of the maximal homology search program described above. The passive search sequence is again unlimited and the active search sequence interval may now contain up to 5000 nucleotides. In a first step, areas of highest uninterrupted homology are searched and sorted by their length. The program allows to assign up to 30 sorted sequences. The sequences within the table are then sorted in ascending order. The distance between each end and start of any two maximal homology areas is calculated both for the passive and active search sequence. If the numbers derived from this calculation are equal or differ only by one, the program protocolls these sequences as belonging together. In the final printout the area between any of these two areas of maximum homology stretches are filled with either of the four signs for homology. The missing nucleotide as

```
0 = maximum, any number = minimal number of homologies
} 12

     1          27          27 uninterrupted homologies
     TTCTCATGTTTGACAGCTTATCATCGA
     ***************************

     TTCTCATGTTTGACAGCTTATCATCGA
     1          27    Average score = 5.0

    44          67          24 uninterrupted homologies
     AACGCAGTCAGGCACCGTGTATGR
     ************************|

     AACGCAGTCAGGCACCGTGTATGA
    66          89    Average score = 4.9
```

Figure 3. Example of a printout provided by the Maximal Homology Search Program. All sequences with 12 or more nucleotides uninterrupted homology are provided in this example. With the code 0 only the upper sequence with 27 identical nucleotides would have been provided.

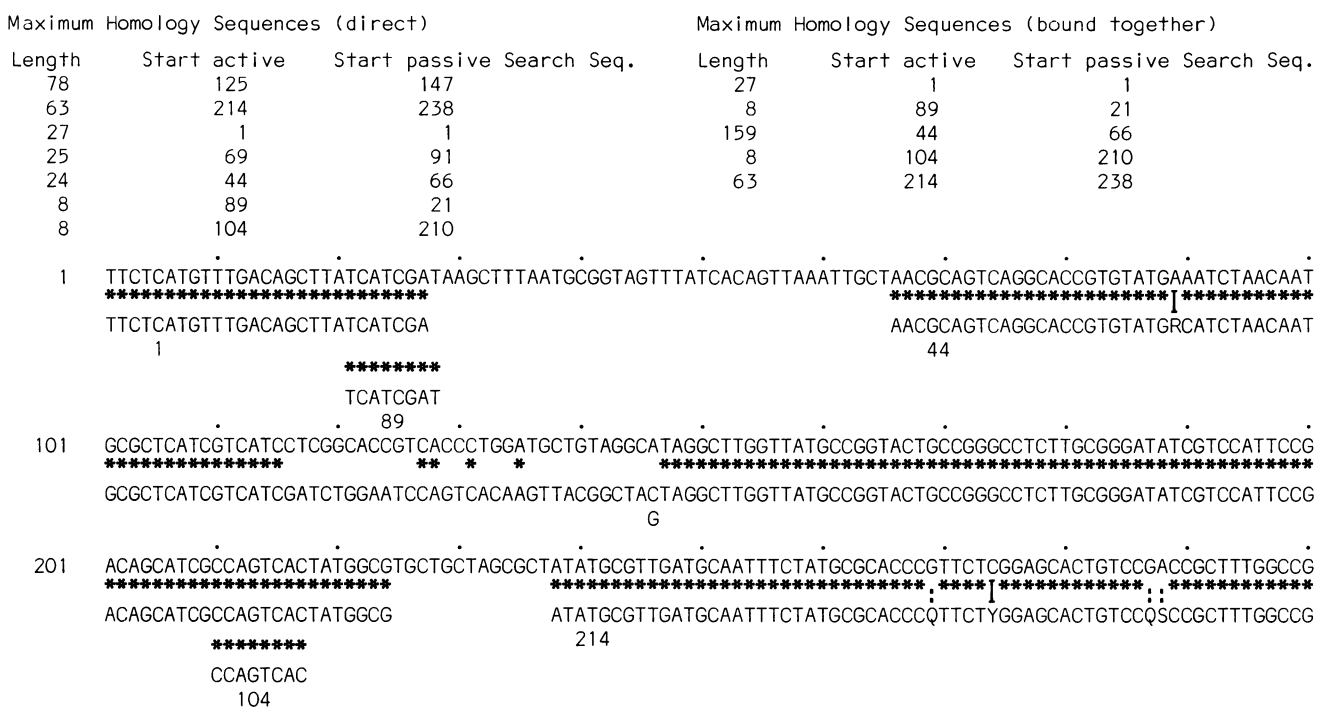


Figure 4. Example of a printout provided by the Alignment Program for Maximal Homology. Two tabels are provided giving the sequences found either according their length or according the position within the passive search sequence. In the second case they are already combined for maximum length. At position 146 an additional nucleotide was found, which is protocolled at the end. The two sequences with 8 identical nucleotides may be disregarded in the final interpretation.

well as the additional one are always located at the position immediately preceding the second stretch of maximal homology. The additional nucleotide is printed with a line offset and moreover especially protocolled at the end of the passive search sequence.

The printout provided by this alignment program contains the entire passive search sequence. All active search sequences are printed in the exact location underneath the passive search sequence. If two such sequences are found, the second one is again printed with a line offset but in the correct position.

An example for a printout provided by this alignment program is given in Fig. 4.

AMINO ACID HOMOLOGY

This program uses nucleotide and amino acid sequence data as well for active search. The passive search sequence is any nucleotide sequence of unlimited length. The basis of homology is, however, the amino acid sequence coded for within the appropriate search sequence. The program compares all three reading frames of the passive search sequence with all six possible reading frames of the appropriate active search sequence. The search is for maximum or for a given number of perfect matches only.

Basis of comparison is the standard one letter code for amino acids. Thus, if the active search sequence should be typed in directly, it must be either as nucleotide sequence or in the one letter code for amino acid sequences. Consequently, the provided printout contains only this one letter code, together with the calculated location within active and passive search sequence. Reverse orientation is again protocolled if pertinent. An example for this program is given in Fig. 5.

```
0 = maximum, any number = minimal number of homologies
} 8
    128        157        10 uninterrupted homologies
      AR*THHRQGV
      **********

      AR*THHRQGV
     17         46

    550        573         8 uninterrupted homologies
      CYARQAER
      ********

      CYARQAER
    118        141
```

Figure 5. Example of a printout provided by the Amino Acid Homology Program. All sequences with 8 or more identical amino acids are protocolled. Note that also a termination codon (*) is accepted as identical information. The numbers provided correspond to the nucleic acid sequences compared. If an amino acid sequence is used for comparison, the single letter code must be used and the lower line will represent numbers within the amino acid sequence directly.

DISCUSSION

We present four different computer programs for search of homology within nucleotide sequences. Though they fulfill different scopes, they generally show the same architecture. They are written as dialogue programs and can be used without prior computer experience. All programs can be handled directly and the calculation only very rarely takes more time than one voluntarily wants to wait in front of the computer. However, this depends on the capacity of the computer.

In routine lab work we normally use the linear homology search for DNA sequencing. The printouts provided by the computer are easy to interpret. Once the user is familiar with the average score system, he may even use it for more sophisticated homology search. However, we use the program primarily for search of stretches of uninterrupted homology to fit a certain sequence into a given sequence.

Since most of the research in our lab is done on the comparison of related bacteriophages, we use the maximum homology search for a first screening. If the maximal homology reaches only 8 nucleotides, we consider these sequences to be unrelated. If we find higher homology we use the alignment program to get an idea how much homology is between the areas of big homology. In many cases the differences are fairly small - regarding mostly single base changes - or are deletions or insertions. In all these cases we prefer our alignment program over any graphical approach, since with our program a direct correlation with exact numbers is possible. In addition, if no homology was found immediately, because the program is not able to assign insertions or deletions correctly, one can easily draw such homology with a pencil directly onto the printout. This allows some freedom within the interpretation of the data.

The amino acid homology program provides another alternative for comparison of functionally related areas. We believe that this program is of special value, since it uses the nucleotide sequences directly for this type of search. This simplifies fairly and avoids additional errors.

In order to provide maximal users comfort all programs described here and elsewhere[10] are grouped together in a menu

type of supervising program, called KROEGERMENUE. Every individual program is called via a code number. All questions in the different interactive parts require standard types of answers. All provided printouts are fairly similarly organized. This allows everyone to become easily familiar with the program collection.

Our programs will be extended in the future to allow not only the coding capability as basis for the homology, but also secondary structure elements within the search sequence. If, for instance, every protein binding certain areas of nucleic acids proves to contain helical structures at the amino terminus, this should be used as a basis for homology search.

ACKNOWLEDGEMENTS

All programming was done on the Univac 1108 computer of the Universitäts-Rechenzentrum of the Albert-Ludwigs-Universität Freiburg. We would like to thank Mrs. M. Scheufens for expert help in programming the alignment program and Dr. G. Hobom for constant interest and suggestions.

*To whom correspondence should be addressed

REFERENCES

1. Wilbur, W.J. and Lipman, D.J. (1983) Proc. Natl. Acad. Sci. USA 80, 726-730.
2. Dumas, J.-P. and Ninio, J. (1982) Nucl. Acids Res. 10, 197-206.
3. Novotny, J. (1982) Nucl. Acids Res. 10, 127-131.
4. Felsenstein, J., Sawyer, S., and Kochin, R. (1982) Nucl. Acids Res. 10, 133-139.
5. Goad, W.B. and Kanehisa, M.I. (1982) Nucl. Acids Res. 10, 247-263.
6. Harr, R., Hagblom, P., and Gustafson, P. (1982) Nucl. Acids Res. 10, 365-374.
7. Queen, C., Wegman, M.N., and Korn, L.J. (1982) Nucl. Acids Res. 10, 449-456.
8. Staden, R. (1977) Nucl. Acids Res. 4, 4037-4051.
9. Sankoff, D., Cedergren, R.J., and McKay, W. (1982) Nucl. Acids Res. 10, 421-431.
10. Kröger, M. and Kröger-Block, A. (1982) Nucl. Acids Res. 10, 229-236.
11. Stockwell, P.A. (1982) Nucl. Acids Res. 10, 115-125.