Volume 10 Number 1 1982 Nucleic Acids Research

Two-dimensional graphic analysis of DNA sequence homologies

Robert Harr [1,2], Per Hagblom[1] and Petter Gustafsson[1]

[1]Department of Microbiology, University of Umeå, S-901 87 Umeå, and [2]Chalmers University of Technology, Sven Hultins väg, S-412 96, Göteborg, Sweden

Received 23 September 1981

ABSTRACT

We describe a computer program designed to facilitate the pattern matching analysis of homologies between DNA sequences. It takes advantage of a two-dimensional plot in order to simplify the evaluation of significant structures inherited in the sequences. The program can be divided into three parts, i) algorithm for search of homologies, ii) two-dimensional graphic display of the result, iii) further graphic treatment to enhance significant structures.

The power of the graphic display is presented by the following application of the program. We have conducted a search for direct repeats in the mouse immunoglobulin κ-chain genes. Both the five J DNA sequences and other shorter repeats were found. We also found a longer stretch of homology that could indicate the presence of duplicated DNA in the J4, J5 region.

INTRODUCTION

The progress in DNA sequencing techniques during the last years has made a rapidly growing number of sequences available for analysis of their intrinsic information. It has also become clear that in order to analyze the information carried by DNA sequences there has been a necessity to use computers and specially designed computer programs. For example in the case of shotgun M13-sequencing of large DNA-sequences the use of computers should not be underestimated.

A number of papers describing computer analysis of DNA sequences (1-4) and RNA sequences (5, 6) have been published. The user-interactive programs designed to facilitate the analysis of DNA sequences (1-4) differ both in program structure and algorithm used to find regions of homology between two DNA-sequences. Staden (4, 5) uses an algorithm which does not permit "looping out" of nucleotides. The algorithm described by Korn (2) allows "looping out" of nucleotides but cannot find so called branched homologies.

We have found, using published computer programs, that the search for homologous stretches between two DNA sequences are hampered by the poor display of the result. We have found a need for novel types of displays to solve pattern matching problems encountered in DNA sequence homology studies. In this paper we present an extended Korn-algorithm (2) that allows the finding of "branched" homologies. We also present a type of graphic display that enhances the evaluation of the results from homology studies by the cooperative power of a two-dimensional (2-D) graphic plot and the human eye.

Description of the program.

The program can be divided into three parts:

i) A search algorithm used to find homologies.

ii) A 2-D graphic display used to present found homologies.

iii) Further graphic analysis of the presented 2-D plot.

The program is written in PASCAL 6000 (7) and consists of 1,700 statements organized in 65 subroutines. The drawing of the picture in the 2-D plot uses a graphic support software, General Purpose Graphic System - Fortran (GPGS-F) (8), implemented on our computer. The data analysis was carried out on a CDC Cyber 170/230 Dual computer and a Tektronix 4051 desk top computer was used as a graphic terminal connected to a Tektronix 4662 plotter.

i) The algorithm to find homologies: When the job is initiated the program asks for sequences, regions to be searched, minimal (min) string length of homology (L) and min number of hits within string (H) (Fig. 1). At any time during the run the parameters defined in Fig. 1 can be changed using the code words defined in the legend of Fig. 1. Using the defined parameters the search for homologies starts. The algorithm used to find homologies is based on the algorithm presented by Korn et al. (2). In short it is composed of the following steps.

The search for homologies starts by comparing the last but one nucleotides in one sequence with the first nucleotide in the other sequence. When the search for homologies has been completed with the two sequences in this position, the first sequence is "slided" one step and the search is rerun. Within each search the computer looks for two succeeding nucleotide pairs that match. When such a match is found the comparison is continued if any of the following statements holds true (the algorithm now continues in a recursive way):

a) Next pair of nucleotides match.

```
TYPE SOS FOR EXPLANATION
TYPE T TO TERMINATE

                        RAND3                          RAND4

 L  H  P D      NA   IA    CA   FLA          NB   IB    CB   FLB
 =========      ====================        ====================
 6  4    D                        1,54 ALL                    1,37 ALL

INSERT CHANGES (SEPARATE WITH COMMA)
(RETURN WILL RUN PROGRAME AT PRESENT STATUS,)
?
```

Figure 1. List of defineable parameters.
The parameters listed in the figure are used to define a search run. All parameters are entered from the keyboard. The following parameters may be set: Sequences, first and last position in the sequences, min string length, min number of hits and mode of listing. During the run all following parameters mentioned above may be changed. Abbreviations: L = Min string length; H = Min number of hits; P = Printer; D = Graphic display; NA, NB = New sequence A and B, respectively; IA, IB = Inverted sequence A and B. respectively; CA, CB = Reversed complement of sequence A or B, respectively; FLA, FLB = First and last position in sequence A and B, respectively.

b) Next pair does not match but two out of the following three do.
c) By "looping out" 1, 2 or 3 in the first and then in the second sequence and finding at least 3 consequtive bases that match in the other sequence.

As soon as one of those conditions holds true the algorithm calls itself in a recursive manner and tries to extend the found homology. When none of those conditions holds true the program checks if the found match fulfills the conditions defined prior to start of the search. The program then falls back through the recursive routine and for every backstep tries the possibilities in the algorithm, which have not yet been tested. This method makes it possible to find so called branched homologies, see Fig. 6. This method also avoids the possibility of being side-tracked by a sequence section that fulfills only one of the steps in the algorithm. When the search has been completed all homologies found to hold the

conditions defined in the algorithm above and fulfill min string length and min number of hits, are stored in a memory (file storage). The data stored in this file is later used in the graphic presentation of the result.

ii) 2-D graphic display (Harr-plot) and iii) Further graphic analysis.

The part of the computer program that builds up the 2-D picture consists of about 500 lines organized into 25 subroutines. For the actual drawing of the picture we take advantage of the graphic support software, GPGS-F, see above. This program package is written in FORTRAN and is called from different subroutines in the PASCAL-program. The GPGS-F program is used to initiate and define the terminal to be used. Other rou+ines are used to define the X-Y coordinate system, to draw lines and to write numbers and letters. The enlargement of defined parts of the picture is done by rerunning the graphic segment of the program with a new "window" and new parameters.

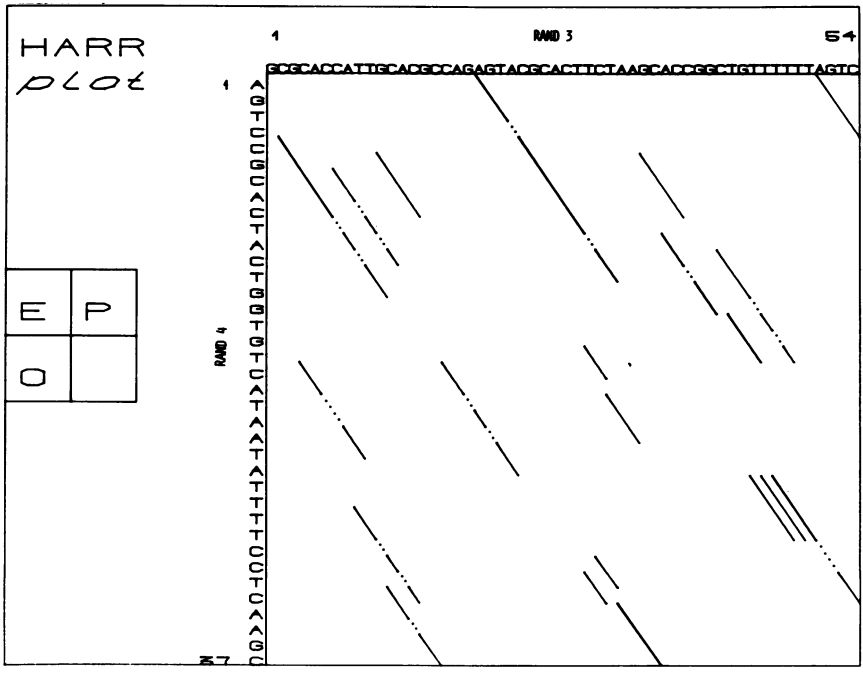

Figure 2. 2-D graphic plot of sequence homologies between two random sequences. Rand 3 and rand 4 are two random DNA sequences generated by a deck of cards. The parameters used were: Min string length = 6 and min number of hits = 4.

The graphic program uses the results found during the search of homologies between the two DNA-sequences (see above). The picture is composed of several elements (Fig. 2 and 3). An X-Y coordinate system is drawn to represent the two DNA-sequences. At the start and end of each axis the numbers of the first and last nucleotides are written. If the sequence is less than 100 nucleotides long the bases are written on the screen. On the left side of the picture a four-boxed square is located (see Fig. 2). This box is used to control the different options of the graphic display. The letters E, P and O in this square stand for exit, printed list and original picture, respectively. The choise between the different options is made by placing the graphic cursor in respective box.

Homologies found are drawn as diagonal lines within the coordinate system. Complete matches are presented as solid lines while mismatches, insertions and deletions are shown as dotted lines.

To increase the power of the analysis we have included a subroutine

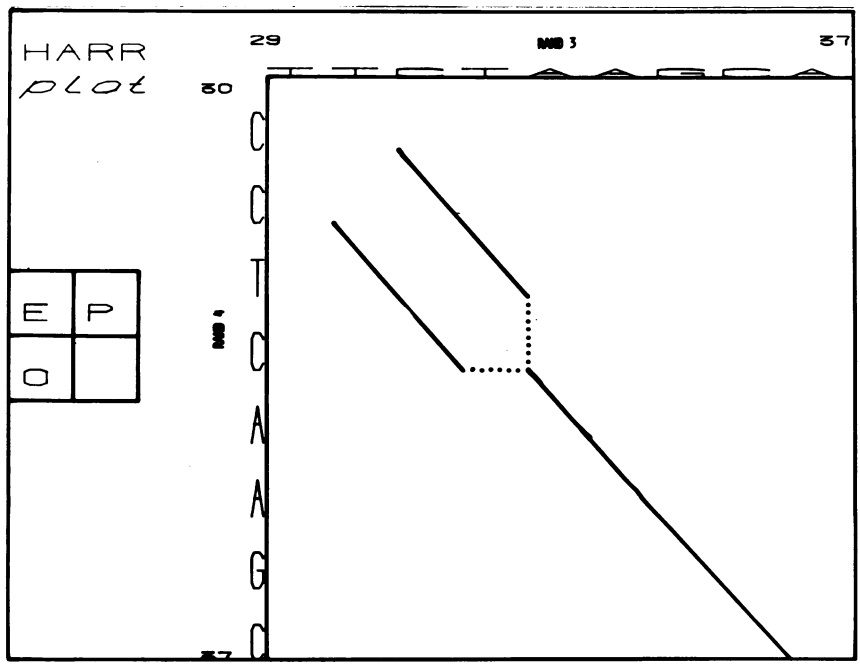

Figure 3. 2-D graphic plot of an enlarged part of Figure 2. For parameters see Fig. 2.

which makes it possible to enlarge defined portions of the plot. The operator defines the new window by placing the cursor first in the upper left and then in the lower right corner of the area to be enlarged. The program now redraws the picture with the redefined coordinates (see Fig 3).

RESULTS AND DISCUSSION

We have constructed a computer program that simplifies the evaluation of the result from DNA sequence computer analyses. The algorithm used to find homologous regions are based on the algorithm published by Korn et al. (2) but has been extended to handle "branched" homologies (Fig. 6). We find this algorithm better than the one used by Staden (3) due to the fact that Korn's algorithm also can find homologous regions where small deletions and insertions have taken place.

The program has been set up with the intention to be easy to use and as flexible as possible. When a search run is initiated the parameters

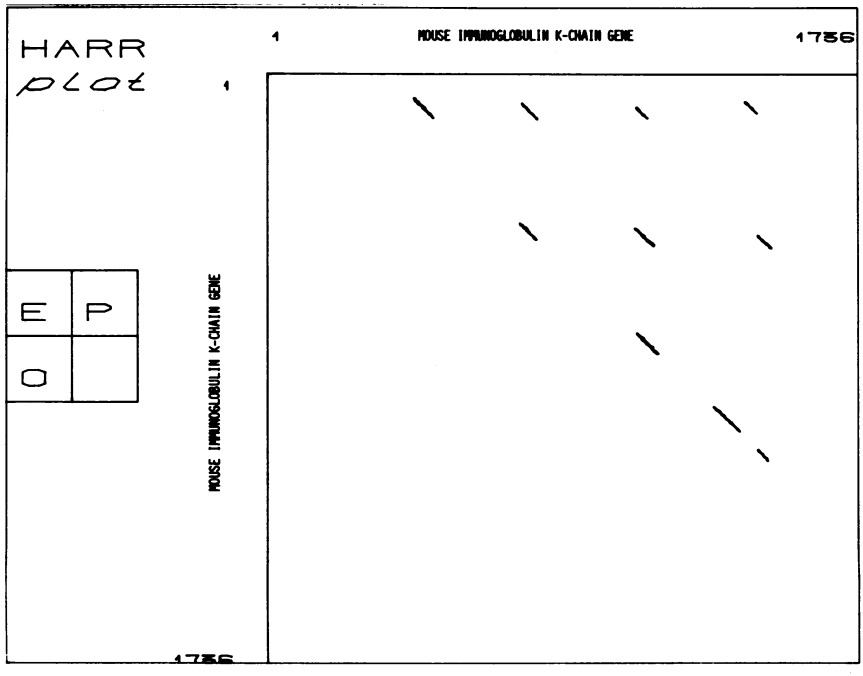

Figure 4. 2-D graphic plot showing sequence homologies within the sequence coding for immunoglobulin light chain J (10). Parameters used were: Min string length = 35 and min number of hits = 25.

are entered from the key board while all parameters used in the graphic display are entered from the screen by use of the cursor. To speed up the movement of the cursor we have found it advantageous to use a Joystick control.

In all work performed in a computer the time consumed by the program is important. As far as we can see our program is as fast as other programs published. For comparison between two DNA sequences, both 2,000 bases long, the program spends 5 to 8 minutes in the CPU of our computer. The time to draw a picture is short but depends on data density of the picture and the transmission rate between the computer and the terminal. With a baud rate of 2,400 the drawing of a complete picture like the one shown in Fig. 2 is performed in less than 15 seconds. When a new window is chosen to enlarge an interesting part of the original graph, the new graph is drawn from the same memory as the original one, and the time to draw the new graph, like the one shown in Fig. 3, is less than 5 seconds.

The power of the presentation can be examplified by the result in Fig. 2. If the result presented in this graph was to be listed in the classical way as a printed list, the result should be composed of 22 individual homologies. The result shown in Fig. 4 would make up at least 200 individual homologies if presented as a printed list. The power of the 2-D plot is mainly due to the sum of the presentation of the homologies as diagonal lines and the direct presentation of the position of each line in the coordinate system. The solving of pattern matching problem presented by DNA sequence homologies is greatly facilitated by this type of plot.

We have tested the program by conducting a search for direct repeats within an immunoglobulin gene. The DNA sequence of immunoglobulin genes are known to consist of a number of different direct repeats of different lengths (9, 10). As an example, we have chosen the mouse immunoglobulin κ-chain gene. The sequence of this gene was published by Sakano *et al.* (10). In this paper they present the interesting result that this particular DNA-fragment contains five joining (J) DNA segments for the κ-chain gene. We tested this sequence using our program and the result is shown in Fig. 4. Four of the five J-segments show up as a regular triangular pattern and was easily identified. The first J-segment does not show up since it is only compared with itself. However, the presence of the first J-segment is easily seen as the upper four diagonals. Those define the homology between the first J-segment and the four other. The graph further stresses the unmatched power of the human eye to discover regular patterns in a picture.

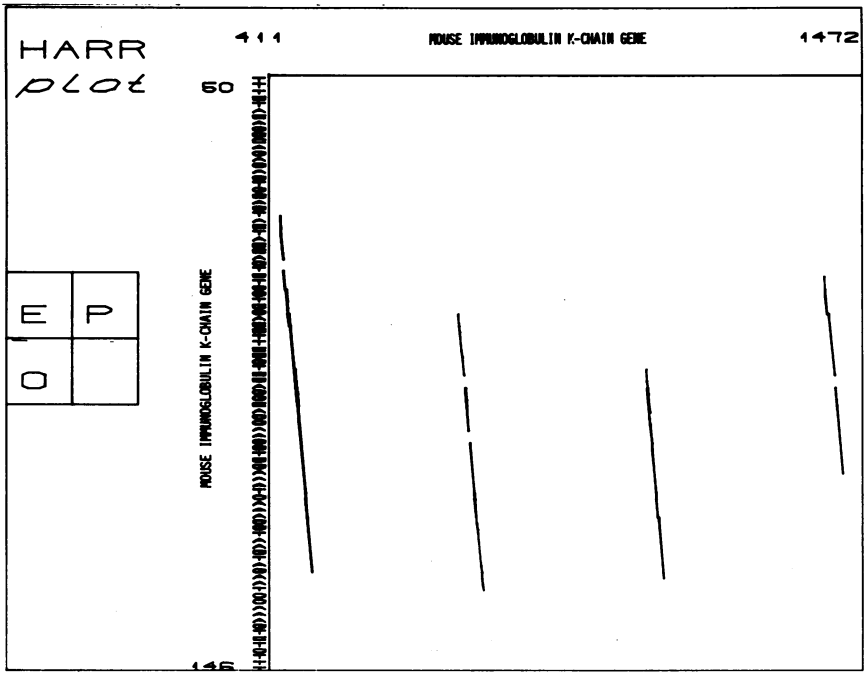

Figure 5. 2-D graphic plot of enlarged part of Fig. 4. The plot shows the homology between J1 and J2 to J5 sequences. For parameters used, see Fig. 4.

An enlargement of one piece of the picture to show the upper four repeated J-segment is shown in Fig. 5. The sequence presented to the left of the graph is the sequence of the first J-segment (J1) and the 4 vertical bars within the graph are J2 to J5. The parameter setting in this run was a string length of 35 and a min number of hits within string of 25. Also with a redefinition of the parameters to 10 and 9, respectively the pattern is easily seen (data not shown). With this parameter setting a number of other shorter direct repeats show up. Those can be studied due to the possibility of rapid enlargement of parts of the picture.

Another interesting result can be seen in the lower right part of Fig. 4. The longer line present in this corner of the plot represents an extended homology in the region around J4 and J5. This reveals an extension of the homology at the 5' end of J4 and J5. This implies a common origin of not only the sequence of J4 and J5 but also the flanking sequences.

Our program presents a new way to solve the pattern matching problems

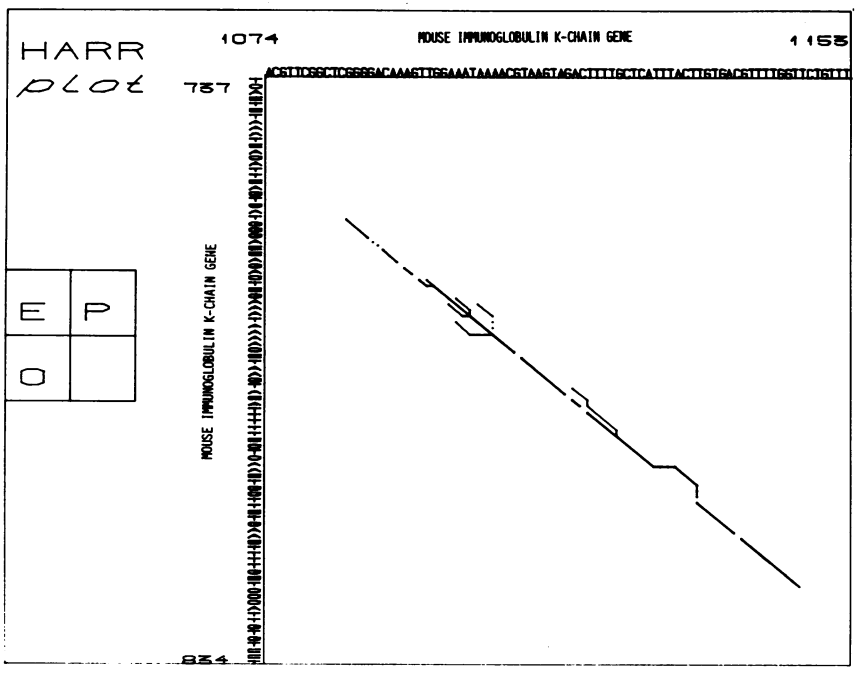

Figure 6. 2-D graphic plot of an enlarged part of Figure 4. The plot shows the homology between J3 and J4. For parameters used see Fig. 4.

that people working with DNA sequences are confronted with. This type of presentation is a useful tool in extracting significant features out of DNA sequences and is superior to the printed lists that so far have been the common method. The capabilities of the program can be further enhanced by adding new routines that for example could give an estimation of the statistical significance of the result.

ACKNOWLEDGEMENTS

We wish to thank Lennart Edblom and Lars-Erik Janlert at the Department of Computer Science, University of Umeå for helpful advice. We also thank the staff of the Umeå Datacentral (UMDAC), Umeå, Sweden.

All correspondence concerning this manuscript should be addressed to: Department of Microbiology, University of Umeå.

REFERENCES

1. Gingeras, T. R., Milazzo, J. P., Sclaky, D., and Roberts, R. J. (1979) Nucleic Acids Res. 7, 529-545.
2. Korn, L. J., Queen, C. L., and Wegman, M. W. (1977) Proc. Natl. Acad. Sci. USA 74, 4401-4405.
3. Staden, R. (1977) Nucleic Acids Res. 4, 4037-4051.
4. Staden, R. (1978) Nucleic Acids Res. 5, 1013-1015.
5. Staden, R. (1980) Nucleic Acids Res. 8, 817-825.
6. Zuker, M., and Stiegler, P. (1981) Nucleic Acids Res. 9, 133-148.
7. Staunstrup, J., and Skov Jensen, E. (1980) Recau Pascal Manual. The Regional EDP-Center at the University of Aarhus.
8. Zachrisen, M. (1978) GPGS-F User's Guide, 3rd ed. Tapir Forlag, Trondheim-NTH.
9. Davis, M. M., Kim, S. K., and Hood, L. (1980) Cell 22, 1-2.
10. Sakano, H., Hüppi, K., Heinrich, G., and Tonegawa, S. (1979) Nature 280, 288-294.