Analysis of large nucleic acid dot matrices on small computers

Stephen E.Zweig

Department of Pharmacology, Baylor College of Medicine, Houston, TX 77030, USA

Received 5 August 1983

ABSTRACT

A UCSD Pascal program was developed which can analyze nucleic acid dot matrices of up to 9500 x 9500 in size on the Apple II computer. Although matrices of such size consume large amounts of computer memory, this program minimizes these problems by analyzing only small strips of the matrix at a time, and then transferring the results to a floppy disk or printer. Compression and memory efficient code further enhance the size of the matrix that can be analyzed. By generating an image of the dot matrix using software, and sending this image directly to an Epson dot matrix printer, a very detailed print out may be produced. The program has a number of user selectable options which allow a great deal of control over the analysis. The program contains no computer dependent code, and thus should work on all systems that can run UCSD Pascal.

INTRODUCTION

The dot matrix method for analyzing nucleic acid sequences is a powerful tool for quickly spotting unsuspected sequence patterns (1,2). In this method, two DNA sequences are placed along the two sides of a matrix, and a dot is placed in the interior of the matrix whenever the two sequences match up according to the dictates of a particular filtering algorithm (3).

This technique is particularly useful for analyzing very long nucleic acid sequences, but here a problem emerges. The matrix formed by the comparison of two long sequences can become huge, and consume massive amounts of computer memory. This can quickly overwhelm the limited memory space available to small computers. Previous implementations of the dot matrix method either implemented it on large computers, such as the Digital VAX system (3), or else avoided the problem by analyzing smaller dot matrices using the high resolution graphics memory of the computer to hold the matrix (4). This latter approach is limited by the hardware of the computer in question. Using the Apple II high resolution graphics, for example, the largest matrix that can be displayed at any one time is 280 by 180.

Modern dot matrix printers have an ability to generate very detailed high resolution output. This ability far exceeds the high resolution graphics capabilities of many current computers. By synthesizing images with software, bypassing the high resolution graphics mode, it is possible for computers without graphics capability to generate very detailed images.

This paper describes a nucleic acid dot matrix program which produces very high resolution dot matrices using such an approach. Using this method, the width of the printed matrix is limited only by the capabilities of the dot matrix printer (960 dots for the Epson MX-80, or 1632 dots for the Epson MX-100). The length of the matrix is effectively unlimited. By computing the matrix as a series of long thin strips, and then transferring the strips to the printer (or to a floppy disk for storage and latter printing), only between 1 to 2 Kilobytes of computer memory need to be allocated for matrix storage. On the 64 Kilobyte Apple II, this frees memory to use for the storage of the two nucleic acid sequences used to generate the matrix. Eàch of these may be up to 9500 bases long. By the additional use of compression techniques matrixes of up to 9500 by 9500 can thus be analyzed in high detail with a common and inexpensive personal computer.

PROGRAM USE

The UCSD Pascal operating system is required to use this program. Sequence data is entered via the UCSD Pascal text editor, and must adhere to the following format:

```
Line 1:  Sequence title - up to 80 characters.
Line 2:  Unused.  Usually left blank.
Line 3:  Sequence data - entered upper case without gaps.
   .
   .
Line N:  Last line of sequence data.  No extra spaces.
```

Each line may be of any length up to 80 characters, but no empty lines after line 2 are allowed. Sequence data entered according to other formats, such as the common (10 bases) space (10 bases)...format can be easily converted over by using the Pascal editor "Replace" command: "/R/ ///". By this option, data created for other UCSD Pascal nucleic acid analysis programs, such as those of Larson and Messing (4), can be quickly converted for use with this program.

Once the appropriate sequence has been entered and stored using the UCSD Pascal Filer, the matrix program can be started from the Pascal Command mode by typing "X" (for execute) "Matrix". The dot matrix program will start, and give the following prompt:

```
MATRIX: E(nter, D(ump, C(orr, K(runch, R(eplay, G(rid, Q(uit?
```

To enter sequence data, the user types "E". The program will then respond with:

```
DATA ENTRY:
Name of the "A" (horizontal axis) sequence?
```

Here the user responds with the disk drive and file name of the horizontal "A" axis sequence to analyze. In the Pascal operating system, the first floppy disk is designated #4:, and the second is designated #5:. If the sequence data was stored in the first disk drive as "DNA.ONE.TEXT", the user would then respond to this question with "#4:DNA.ONE.TEXT" (all quotation marks listed are for readibility, and should not be typed in). The program will then ask for the name of the second sequence to analyze, and the user should again respond as appropriate. The computer will attempt to read these files and convert them to its own internal format. If it is unable to find the files, the user will be requested to try again.

To verify that the appropriate sequences were read, the user may view the sequences on the screen by typing "D" to invoke the "D(ump" option.

Due to the limitations of the MX-80 printer, the width of the printed matrix must be less than 961 dots (1633 for the MX-100 printer). Larger matrixes may be computed and viewed with lower resolution, however, by using the "K(runch" option. This turns on the compression algorithm. Usually set to one, it can be reset to any number from 1 to 30. A compression of N, for example, will produce a 1/N sized replica of the full matrix. With compression, every N dots on the horizontal axis are compressed into one dot by a logical "or" operation. To speed up execution time, a trade off is made in the vertical axis. Only every Nth vertical matrix location is used for the search. As a result, there is a slight asymmetry in the placement of individual dots in compressed matrixes, but the larger features are preserved. This trade off increases run speed by a factor of N, and makes it feasible to run large matrixes in a reasonable span of time.

Normally, the dot matrix is output with a grid superimposed to facilitate analysis. The grid places reference lines at multiples of 50 times the compression factor. This grid may be turned on and off using the "G(rid" option.

The "R(eplay" option is used to retrieve and print matrixes that have previously been computed and stored on a floppy disk. This procedure will ask for the disk drive and file name of the analyzed matrix, and the user should respond as appropriate.

To analyze a matrix, the user uses the "C(orr" (correlate) option. The computer will then respond with:

```
CORRELATE: W(hole matrix, P(artial, C(lose up, or Q(uit?
```

To do the entire matrix, the user should type "W". To look more closely at a selected area of the matrix (which can be of any size and location) the user should type "P". To determine what sequences are actually involved in the area of interest, the user should type "C". The "C(lose up" command causes the computer to display a magnified 20 x 20 portion of the matrix on the computer's screen with the corresponding sequences indicated along the matrix sides.

Choosing either "P" or "C" will cause a series of questions to be asked. The computer will display:

```
COORDINATE ENTRY:
Sequence start number of the "A" sequence?
```

This allows the user to tell the computer the numbering system that has been used to describe the sequence data. If the beginning of the sequence has previously been designated by some number other than one, the user should enter the appropriate number at this point. If, for example, the sequence had started at -150, the user should enter "-150". Otherwise the user should respond with "1". Following this, the computer will ask for the starting number of the "B" sequence.

Next, using the above numbering system, the computer will ask for the coordinates where the upper left hand portion of the partial matrix should start. It requests this information by asking:

Enter the "A" starting location to compute:

If the user wished to produce a partial matrix of a larger matrix with the partial matrix's upper left hand corner starting at location A=1000, B=2000; the user should enter "1000" to this question, and "2000" to the following "B" sequence question.

If the "P(artial" option was chosen, the computer will ask for the width and length of the partial matrix to compute. It will ask:

Enter WIDTH of partial matrix ("A" sequence):

Here the user would enter the desired width and length. For the "C(lose up" option, these questions are not asked, all matrixes being automatically set to 20 x 20.

At this point, questions as to the type of filtering algorithm to use are asked. The computer will display:

FILTER OPTIONS:

Look for H(omology or S(econdary structure?

Choosing "H" will cause match ups to be made if two sequence bases are the same, while choosing "S" will cause match ups to be made only if there as A-T, G-C, or A-U type basepairing.

The next processing question asked will be:

Correlate how many base pairs (1..30)?

This controls the filtering of the data. Selecting 5 here, for example, would instruct the computer to scan 5 bases both forward and backward from it's current matrix location to check for a series of possible sequence match ups. For homology searches, both forward and inverse sequence alignment is tested for. For secondary structure, only inverse sequence alignment is tested. The computer will also need to know how accurate a match up criterion is being set. It requests this information by asking:

Accept how many correct match ups?

If perfect correlation is desired, the user should enter the same number as the previous question. If, however, a certain a-

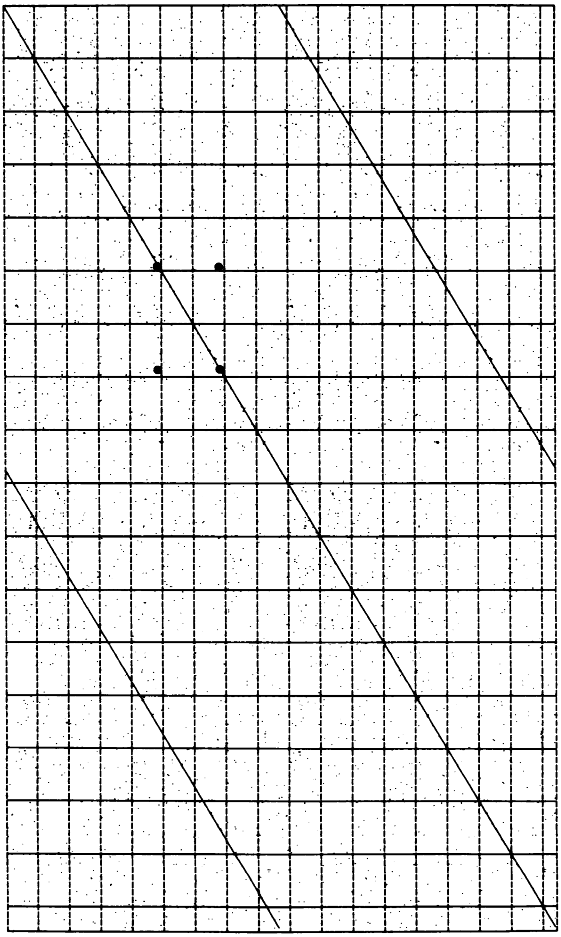

Figure 1. Analysis of a 8724 x 8724 matrix composed of two repeated 4362 base pBR322 sequences. The repeat is seen as the two diagonal lines parallel to the main diagonal. The area bounded by the four dots is shown in Fig. 2.

mount of mismatch is acceptable, a smaller number may be entered. For example, entering "4" here would tell the computer that 4 out of 5 bases is an acceptable criteria for outputing a dot. This latter option is useful for secondary structure and distant homology searches.

The last question is:

Matrix to P(rinter or D(isk?

This directs the resulting matrix output to these respective

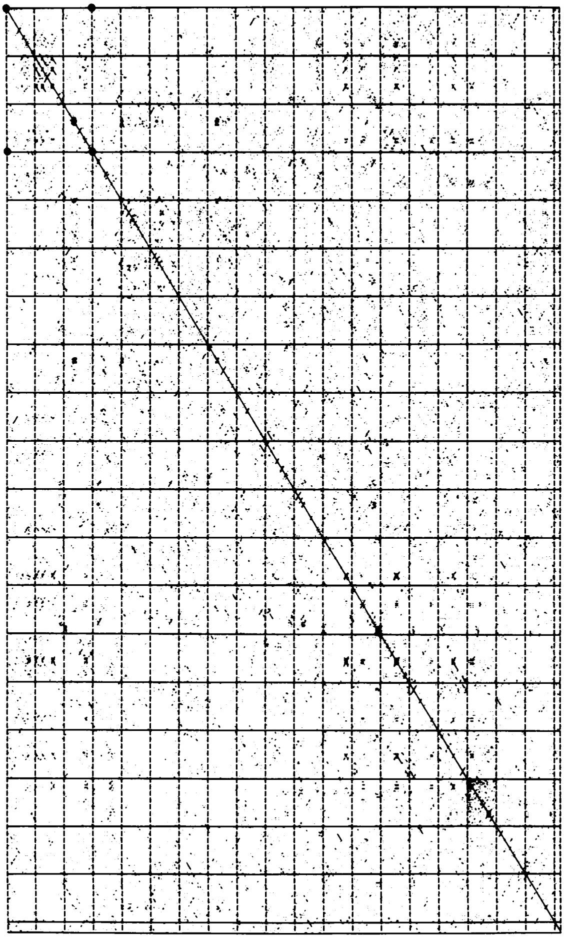

Figure 2. A 960 x 960 matrix blow up of the region indicated in Fig. 1. More detail may now be seen. The pBR322 origin of replication is bounded by four dots. This region is shown in Fig. 3.

locations. To use the disk option, the user should have the appropriate number of empty UCSD Pascal formatted disks on hand. With no compression, a 960 by 990 matrix will fill up one entire (Apple) disk. A longer matrix will require additional disks, with the user being prompted to change disks as they fill up.

EXAMPLE OF USE

To analyze very large matrixes, it is useful to first get an

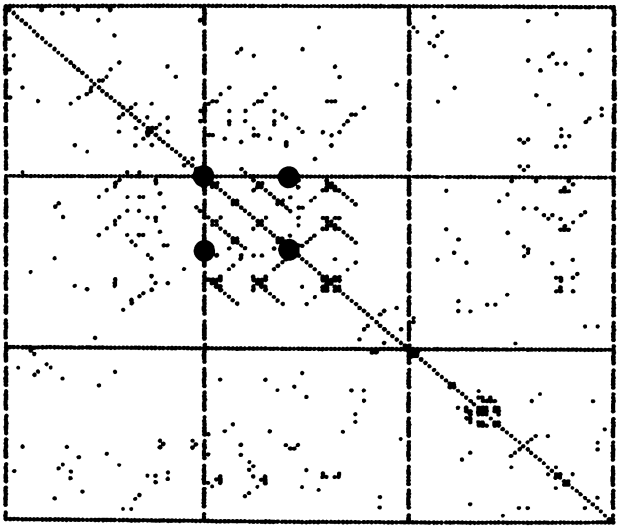

Figure 3. A 150 x 150 matrix of the pBR322 origin of replication shown in Fig. 2. A number of small repeats may be seen. Two of the repeats are bounded by four dots, and this area is shown in Fig. 4.

```
           2484 -->
           A A A A G G C C A G C A A A A G G C C A

           - - - - - - - - - - - - - - - - - - - -
2484  A  ¦ * . . . . . . . . . . * . . . . . . . .
  ¦   A  ¦ . * * . . . . . . . . . * * . . . . . .
  ¦   A  ¦ . * * . . . . . . . . . * * . . . . . .
  V   A  ¦ . . . * . . . . . . . * . . * . . . . .
      G  ¦ . . . . * . . . . . . . . . . * . . . .
      G  ¦ . . . . . * . . . . . . . . . . * . . .
      C  ¦ . . . . . . * * . . . . . . . . . * * .
      C  ¦ . . . . . . * * . . . . . . . . . * * .
      A  ¦ . . . . . . . . * . . . . . . . . . . *
      G  ¦ . . . . . . . . . * . . . . . . . . . .
      C  ¦ . . . . . . . . . . * . . . . . . . . .
      A  ¦ * . . * . . . . . . . * . . * . . . . .
      A  ¦ . * * . . . . . . . . . * * . . . . . .
      A  ¦ . * * . . . . . . . . . * * . . . . . .
      A  ¦ . . . * . . . . . . . * . . * . . . . .
      G  ¦ . . . . * . . . . . . . . . . * . . . .
      G  ¦ . . . . . * . . . . . . . . . . * . . .
      C  ¦ . . . . . . * * . . . . . . . . . * * .
      C  ¦ . . . . . . * * . . . . . . . . . * * .
      A  ¦ . . . . . . . . * . . . . . . . . . . *
```

Figure 4. A 20 x 20 close up view of repeats "3" and "4" described by Sutcliffe (5). The sequence can be seen along the sides of the matrix.

overview of the entire matrix at low resolution, and then zoom in to selected areas of interest at higher resolution.

Fig. 1 shows an analysis of a very large, 8724 x 8724, test matrix. Each side of this matrix is composed of two 4362 base pBR322 sequences joined front to back. The sequence used to construct this matrix was converted from a file used by the Pascal programs of Larson and Messing (4). A high degree (10 matches out of 10 bases) of filtering was used to generate a noise free image. To output this matrix on an MX-80 printer, a compression of 10 was used. The pBR322 repeat can clearly be seen as the two diagonal lines parallel to the main diagonal. Note that due to the nature of this printer, the dots that make up the horizontal axis of very large matrixes are printed closer together than the dots that make up the vertical axis. This leads to a rectangular rather than to a square matrix. Each grid in Fig. 1. corresponds to a 500 x 500 area. The four larger dots represent the boundary of the 960 x 960 matrix shown in higher detail in Fig. 2.

Fig. 2 shows a "P(artial matrix" view of a selected 960 x 960 area indicated by four dots in Fig. 1. This area covers positions 2434 to 3394 in the pBR322 sequence. This region contains the origin of replication for pBR322, and this portion is indicated by the four larger dots in the upper left hand corner. This is shown in greater detail in Fig. 3. Less filtering (5 matches out of 5 bases) was used here. Compression is 1. Each grid now corresponds to a 50 x 50 area. Note that a number of small repeats and inverted repeats are visible.

Fig. 3 shows a "P(artial matrix" view of the 150 x 150 area surrounding the pBR322 origin of replication. The matrix starts at location 2434. A number of sequence repeats, previously described by Sutcliffe, are evident in the center square. The four larger dots represent an area surrounding repeats "3" and "4" described in his paper (5). This is shown in Fig. 4. The printing of matrixes with widths of less than 481 (817 for the MX-100 printer) allows the MX-80 printer to operate in a mode which gives horizontal dots the same spacing as vertical dots. This produces an easier to see image. The filtering and compression for this matrix are the same as in Fig. 2.

Fig. 4 shows a "C(lose up" view of the area indicated by the four dots in Fig. 3. This option allows the user to see exactly which bases are involved in the structures of interest. Here, the sequence of Sutcliff's repeats "3" and "4" may be directly read from the sides of the matrix. This feature allows the user to preview the area on the screen, and then print the results if so desired.

HARDWARE REQUIREMENTS

The program was designed to run on a minimal system. Matrixes of up to 9500 by 9500 can be analyzed on a 64 K apple II, II+ or IIe with only a single disk drive and an Epson or comparable dot addressable dot matrix printer. To maintain ease of modification and generality, the program contains no system dependent code. It thus should run with little or no modification on any computer that supports UCSD Pascal. Run times are long. The 960 x 960 matrix shown in Fig. 2 took 18 hours to generate, and the 8724 x 8724 matrix shown in Fig. 1 took 11 days. The cost of apple computer time is essentially nothing, however, and for most problems, overnight runs are sufficient.

To receive a copy of the program, including documented source code, send a blank floppy disk with a self addressed stamped mailer to the author. A much faster IBM Pascal version for the IBM Personal Computer is also available.

ACKNOWLEDGEMENTS

The author would like to thank Drs. Harris Busch, Lawrence Rothblum, and Chirala Subrahmanyam for their advice and encouragement.

REFERENCES

1. Heiter, P.A., Max, E.E., Seidman, J.G., Maizel, J.V., and Leder, P. (1980) Cell 22, 197-207.
2. Steinmetz, M., Frelinger, J.G., Fisher, D., Hunkapillar, T., Pereina, D., Weissman, S.M., Uehara, H., Nathenson, S., and Hood, L. (1981) Cell 24, 125-134.
3. Novotny, J. (1982) Nucleic Acid Res. 10, 127-131.
4. Larson, R., and Messing, J. (1982) Nucleic Acid Res. 10, 39-49.
5. Sutcliffe, J.G. (1979) Cold Spring Harbor Symp. Quant. Biol. 43,77-90.