Volume 10 Number 24 1982 Nucleic Acids Research

An algorithm for the display of nucleic acid secondary structure

G. Lapalme[1], R.J. Cedergren[2] and D. Sankoff[3]

[1]Département d'Informatique et de Recherche Opérationnelle, [2]Département de Biochimie, and [3]Centre de Recherches Mathématiques Appliquées, Université de Montréal, Montréal, Québec H3C 3J7, Canada

Received 17 June 1982; Accepted 27 October 1982

ABSTRACT

A simple algorithm is presented for the graphic display of nucleic acid secondary structure. Examples of secondary structure displays are given for tRNA, 5S RNA and part of the 16S RNA. Due to its speed, this algorithm could easily be used in conjunction with secondary structure programs which calculate various alternate structures.

INTRODUCTION

Ever since the first RNA was sequenced, considerable effort has been invested in inferring the intramolecular base pairing (i.e. the secondary structure) of these molecules. Recently the comparison of secondary structures of ribosomal RNAs has been very useful in evaluating evolutionary relationships among diverse organisms, chloroplasts and even mitochondria [1-5]. It is clear in these works, that much time has been devoted not only in the determination of sequences but also in the graphic representation of these data. We present here a rapid, easily used algorithm for the automatic display of secondary structures of nucleic acids.

THE DATA AND THE ALGORITHM

The data for the algorithm consists of a string of letters (i.e. the primary structure) and a list of paired positions in this string (e.g. AACACCAUU / 1•9,2•8 is the input for a molecule which forms a "hairpin loop" with the first two A's paired to the last two U's). The input must obey a "no-knots" constraint:

if "i•j" and "k•l" are two sets of base pairs then

i) k and l are both between i and j

ii) neither k nor l is between i and j

0305-1048/82/1024-8351$2.00/0

The strategy followed by the algorithm is to first place (i.e. find the two dimensional coordinates) a base pair "i•j" where i and j are as far away as possible in the string (assuming i is less than j). This automaticaly determines the placement of (i+1)•(j-1), (i+2)•(j-2), up to (i+n)•(j-n) if i•j is the first in a series of exactly n+1 stacked base pairs in the secondary structure. Usually n is at least 1. When the procedure encounters a pair i•j for which (i+1) and (j-1) are not paired to each other, it must construct a loop. It does this by counting the number of bases in the loop as follows. The ith position count as one. If the i+1st base is not paired the count increases by one and the algorithm moves to the i+2nd position. If this is not paired it adds one, and so on. When a base is encountered which is at a paired position, e.g. position k, paired to position l, it adds two and moves to the l+1st position. This continues until we eventually arrive at position j. The n bases counted up to this point, including i and j are placed at the corners of a n-sided polygon for which the coordinates can be calculated based on the previous placement of the ith and jth bases. If all the bases in the molecule have now been placed, the algorithm stops as would be the case for the example which is depicted in Figure 1.

If some base pairs (e.g. k•l) have been encountered in counting n for the last loop however, there will remain some as yet unplaced bases (the (k+1)st to the (l-1)st bases). The algorithm then proceeds recursively, starting at each such base pair k•l in the loop, and so on, until all bases have been placed. The algorithm may be summarized as follows:

while there remain unplaced bases

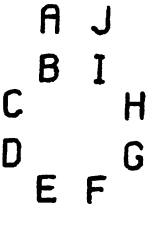

Figure 1

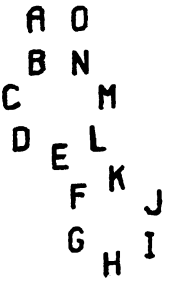

Figure 2

-find pair i•j with j-i as large as possible

loop(i,j)=

-n:={number of bases in the loop}*

-place the n-2 other loop elements based on the placement of i and j

-for each pair k•l encoutered when counting n

-call loop(k,l)

* if the next bases are paired ((i+1)•(j-1)) then n=4, a square for which the coordinates are easy to find. The implementation can be speeded up by making use of this frequent special case.

Figure 2 shows the display on inputing: (ABCDEFGHIJKLMNO / •15, 2•14, 5•12, 6•11).

The coordinates of each symbol have thus been determined given the coordinates of the first pairing. We now use a plotting package for the drawing of each symbol at each coordinate. A PASCAL program implementing the algorithm is available from the authors; it is possible to transform this algorithm to FORTRAN but the program would have to stack values explicitly to handle the recursive calls to "loop".

AVOIDING OVERLAP

The standard cloverleaf model of tRNA and the model of eukariotic 5S RNA as displayed by the above algorithm are shown in Figures 3 and 4 respectively. Attempting to use the algorithm on longer (~500 nucleotides) sequences presents new problems since occasionally parts of loop structures can overlap (see Fig. 5). We have thus modified the program so that when overlap

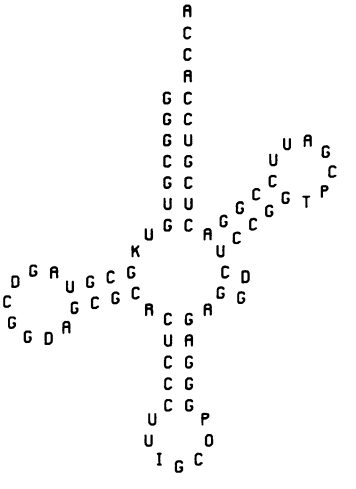

Figure 3. Display of yeast $tRNA^{Ala}$ (6). Symbols for modified nucleotides as in ref. 7.

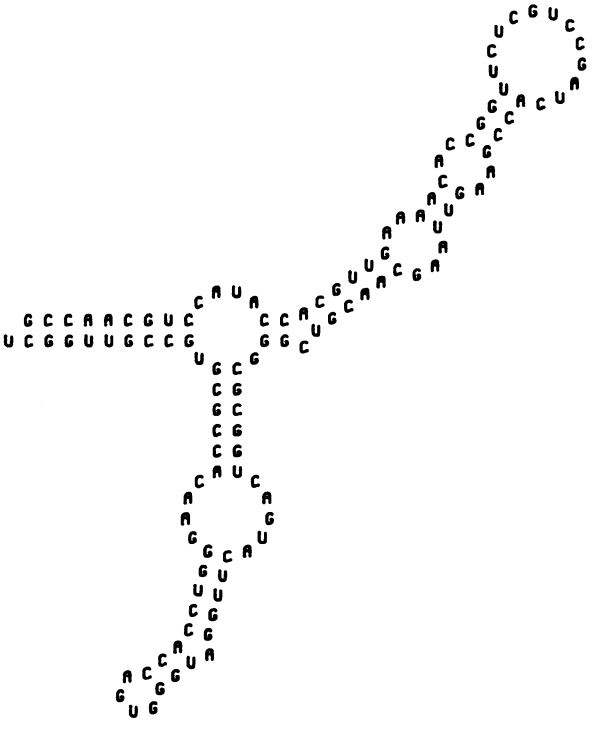

Figure 4. Display of <u>Philosamia</u> 5S RNA (8).

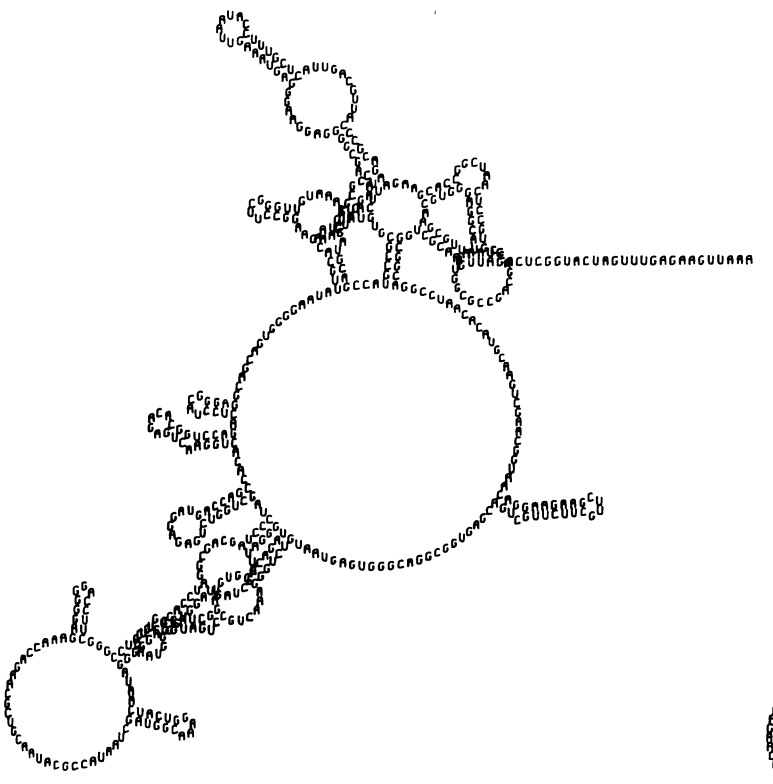

Figure 5. Preliminary display of E. coli 16S RNA using base pairing patterns as in ref. 5.

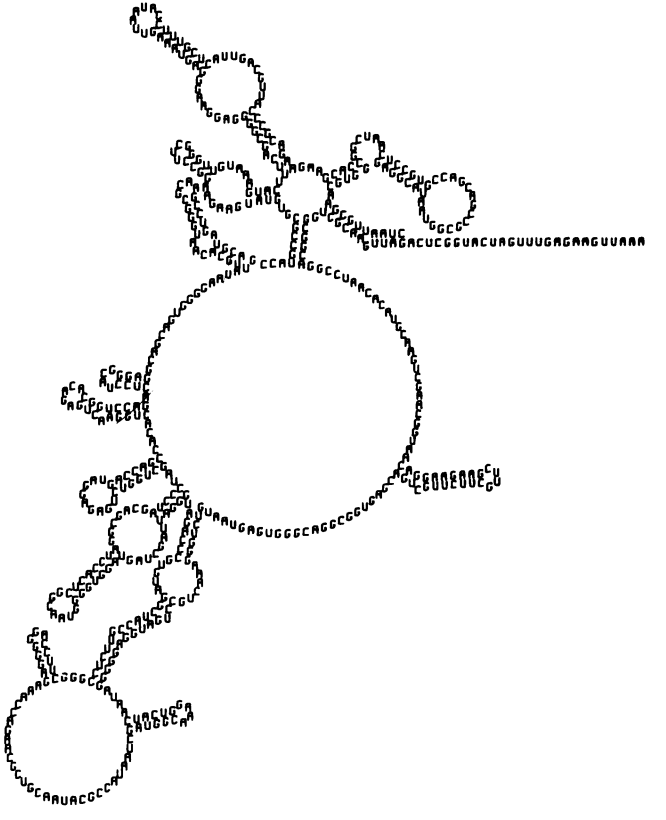

Figure 6. Modified display of E. coli 16S RNA using base pairing patterns as in ref. 5.

is noted in a preliminary run, we may modify the angle that a stack of base pairs makes with the central single stranded region. This modification also slightly increases the distance between adjacent unpaired nucleotides of the central region so that the position where the loop attaches is unambiguous (see Fig. 6). This algorithm is quite rapid (2 seconds on a CYBER 173) permitting its use in real time application as in video displays or interactive schemes. This program may also be linked to a number of available programs determining secondary structure allowing the rapid display of alternate structures (see for example ref. 9 and the Prophet system described therein).

REFERENCES

1. Glotz, C., Zwieb, C., Brimacombe, R., Edwards, K. and Kossel, H., Nucleic Acid Res. 9, 3287-3306 (1981)
2. Branlant, C., Krol, A., Machatt, M. A., Pouyet, J., Ebel, J-P., Edwards, K. and Kossel, H., Nucleic Acid Res. 9, 4303-4325 (1981)
3. Noller, H. F., Kop, J., Wheaton, V., Brosius, J., Gutell, R. R. Kopylov, A. M., Dohme, F., Herr, W., Stahl, D. A., Gupta, R. and Woese, C. R., Nucleic Acid Res. 9, 6167-6189 (1981)
4. Brimacombe, R., Nature 294, 209-210 (1981)
5. Stiegler, P., Carbon, P., Ebel, J-P. and Ehresmann, C., Eur. J. Biochem., in press (1981)
6. Holley, R. W., Apgar, J., Everett, G. A., Madison, J. T., Marquisee, M., Merril, S. H., Penswick, J. R., and Zamir, A., Science 147, 1462-1465 (1965)
7. Cedergren, R. J., LaRue, B., Sankoff, D. and Grosjean, H., CRC Crit. Rev. Biochem. 11, 35-104 (1981)
8. Gu, X-R., Nicoghosian, K. and Cedergren, R. J., Nucleic Acid Res., in press (1982)
9. Auron, P. E., Rindone, W. P., Vary, C. P. H., Celentano, J. J. and Vournakis, J. N., Nucleic Acid Res. 10, 403-419 (1982)