Proc. Natl. Acad. Sci. USA
Vol. 77, No. 11, pp. 6309–6313, November 1980
Biochemistry

Fast algorithm for predicting the secondary structure of single-stranded RNA

(computer program/polynucleotide/RNA folding)

RUTH NUSSINOV* AND ANN B. JACOBSON†

*Department of Structural Chemistry, The Weizmann Institute of Science, Rehovot, Israel; and †Department of Microbiology, State University of New York, Stony Brook, New York 11794

Communicated by Richard B. Setlow, August 18, 1980

ABSTRACT **A computer method is presented for finding the most stable secondary structures in long single-stranded RNAs. It is 1–2 orders of magnitude faster than existing codes. The time required for its application increases as N^3 for a chain N nucleotides long. As many as 1000 nucleotides can be searched in a single run. The approach is systematic and builds an optimal structure in a straightforward inductive procedure based on an exact mathematical algorithm. Two simple half-matrices are constructed and the best folded form is read directly from the second matrix by a simple back-tracking procedure. The program utilizes published values for base-pairing energies to compute one structure with the lowest free energy.**

Due to the rapid increase in our knowledge of the nucleotide sequence of many long single-stranded RNAs, it is of interest to attempt to predict the secondary and tertiary structure of these molecules.

A simple method for estimating the free energy of loops found in single-stranded RNA based on their sequence was developed several years ago (1–6). By utilizing this method, the most probable loop structure for a given sequence is obtained from comparison of the relative stability of all of the possible structures that can form. Although this approach alone works easily for short nucleotide sequences, longer sequences require that many alternate structures be assessed and computer assistance becomes essential.

A number of algorithms have been developed to apply free energy rules to polynucleotide chains (2, 7–9). The basic method in all of these approaches has been similar. Perfectly matched helices in the sequence are identified. Consistent sets of these helices are then assembled, and the overall free energy of each assembled structure is calculated individually. For long chains, the combinatorial aspects of this approach are very large (10, 11) and the time required for the calculations is extremely long.

We have developed an approach to computer folding of large polynucleotide chains in which the algorithm is about 100 times faster than existing approaches. Two simple half-matrices are constructed by an inductive procedure which considers the energy contributions of individual base pairs. The loop structure with the lowest free energy is read directly from the second matrix in a simple fashion. The basic algorithm and its mathematical proof have been presented (12). It was developed initially simply to maximize base pairing along a polynucleotide chain. More recently, we realized that the rules for calculating loop stability based on free energy can be incorporated into the algorithm as well.

This presentation provides a simplified explanation of the original algorithm for maximal matching as well as a description of the procedure developed for incorporating energy rules.

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U. S. C. §1734 solely to indicate this fact.

METHODS

Basic Formulation of the Method. The algorithm is designed to evaluate the contribution of individual base pairs to the secondary structure of a polynucleotide chain. The basic principle on which it rests is best understood by considering a sequence of nucleotides B_1 to B_n which lie on the circumference of a circle (Fig. 1*A*). Nonintersecting arcs, drawn inside the circle, link individual base pairs. This type of structure corresponds to a simple planar secondary structure. Knotted or pseudoknotted structures are not allowed (9). A more conventional representation of the structural features in Fig. 1*A* is obtained by shrinking each of the connecting arcs in the circle to one fixed length (Fig. 1*B*). A structure with one arm results. The various features common to this type of structure are shown in the figure.

In order to find the best folded form of the sequence B_1 to B_n, we will consider whether any bond B_xB_y will be included in the optimal structure. Such a bond will make a direct local contribution to the free energy of the region in which it is formed. There is, however, a more important global effect. The arc connecting B_x and B_y divides the circle (Fig. 1*A*) into an upper and a lower portion. Due to simple planarity requirements, further arcs can only be drawn inside each part separately. Thus, the total free energy of the folded structure formed will be determined by the energy of the lower and upper sections and the local contribution of B_xB_y. The best folded form will be obtained when the free energy of each of these three components has been minimized. The algorithm provides a systematic inductive search procedure to realize these conditions.

Algorithm for Maximal Matching. Any algorithm for folding a long single-stranded polynucleotide chain depends in part on folding rules which specify the manner in which the individual nucleotides are allowed to pair with one another. In addition, it contains a search procedure that is used to find the best folded form in an efficient manner. The algorithm for maximal matching utilizes a simple set of folding rules. The stability of G·C pairs is considered to be equal to that of A·U pairs. Contributions due to stacking are ignored, as are the destabilizing effects of single-stranded loops. Under these conditions, the problem of folding a nucleotide sequence into a structure with minimal free energy becomes the simpler problem of finding a structure with the maximum number of base pairs.

The search procedure utilized by the algorithm proceeds in a simple inductive manner from short subsections of a given nucleotide sequence to sections of increasing length. Optimal folding is determined for each section until the optimal folding

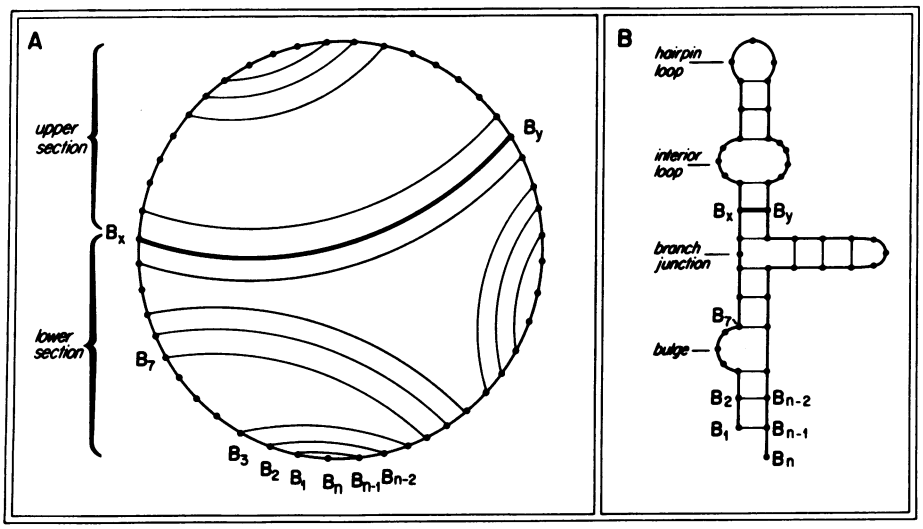

FIG. 1. Base pairing in a simple planar structure. (*A*) Extended form. Base pairs are represented by arcs which join nucleotides located along the circumference of the circle. (*B*) Condensed form. The individual bonds from the structure shown in *A* were shortened and set to a fixed length, resulting in a more conventional representation of a base-paired structure with one branch. Single-stranded loops are labeled according to the nomenclature of Tinoco *et al.* (2).

of the entire molecule is obtained. To illustrate the approach we will consider the sequence $B_1 \ldots B_n$ containing the subsection $B_i \ldots B_j$ of length p. The method used to find the maximal number of base pairs occurring in $B_i \ldots B_j$ is shown in Fig. 2. The variable k is allowed to assume each position from B_i to B_{j-1}. At each point the algorithm tests the ability of B_k to pair with B_j. It further determines the total number of base pairs in the section by checking for the number of base pairs present in the subsections $B_i \ldots B_{k-1}$ and $B_{k+1} \ldots B_{j-1}$. After all positions of k have been tested, the best value obtained is saved and stored in the matrix $M(i,j)$. If B_j cannot pair with any k in $B_i \ldots B_j$, then $M(i,j) = M(i,j-1)$. Information on base pairing for additional sections of length p is obtained by incrementing i and j successively. The maximum number of base pairs that can form in sections of increasing length is obtained by incrementing p to $p+1$ and repeating the search. Thus, for each interval $B_i \ldots B_j$ within the sequence:

$$M(i,j) = \max \begin{cases} M(i,k-1) + M(k+1,j-1) + 1 \\ M(i,j-1) \quad i \le k < j = i + p \end{cases} \quad [1]$$

Because $M(i,j)$ is filled with sections of increasing length, the values $M(i,k-1)$, $M(k+1,j-1)$, and $M(i,j-1)$ are known and can be read directly from the matrix. This results in an extremely efficient search procedure. The value 1 corresponds to the base pair B_kB_j. The last value in $M(i,j)$ is obtained when the section $B_i \ldots B_j$ corresponds to $B_1 \ldots B_n$. This value of $M(i,j)$ specifies the maximum number of base pairs that can be found for the entire sequence.

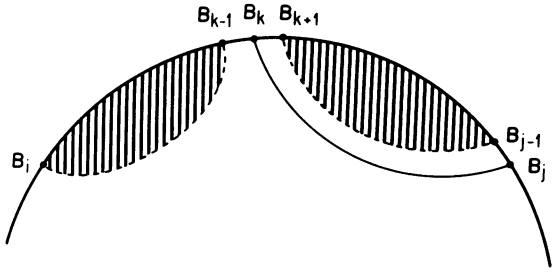

FIG. 2. Various possibilities for B_kB_j bonds as k sweeps across the sequence $B_i \ldots B_{j-1}$ and the separate simple planar structures $(i, k-1)$ and $(k+1, j-1)$, obtained this way.

An example of the actual structure of a simple matrix is shown in Table 1. It was constructed for the 25 nucleotide sequence: C-G-G-G-C-G-G-C-C-G-G-C-C-C-C-G-G-G-C-C-G-C-G-G-C. The data are stored in the lower left half of the matrix $M(i,j)$. The first values were obtained by letting $p = 1$ and scoring pairs formed between adjacent nucleotides along the sequence. These values were stored in the uppermost diagonal of the matrix. Successive diagonals were obtained by increasing p in a stepwise fashion. In each case the base pair B_kB_j (from Fig. 2) divides the segment under consideration into two subsections. The maximal pairs in each subsection were learned by reading values previously recorded in the matrix $M(i,j)$ for these smaller sections. The maximum number of base pairs found for the entire 25 nucleotide sequence is located in $M(1,25)$ and is equal to 12.

The method outlined above provides a procedure for determining the maximum number of base pairs that form within a known sequence of nucleotides. To fold this sequence, it is also necessary to identify the individual nucleotides that pair. For this, the algorithm produces a second NXN matrix $K(i,j)$ which contains the numerical position of the base B_k that allows maximal base pairing within each segment $B_i \ldots B_j$. This information is used to find the best folded form in the following manner. The algorithm begins with the entire nucleotide sequence $B_1 \ldots B_n$. The information stored in $K(1,n)$ gives the position of k which, when paired with B_n, leads to optimal folding of the entire sequence. The formation of this base pair divides the entire sequence into two subsections. Optimal folding of each subsection is found in a similar fashion by reading the value of $K(i,j)$ for the subsection. The process is illustrated in Fig. 3. The first base pair formed is $B_nB_{K(1,n)}$ which divides the sequence into two subsections $B_1 \ldots B_{K(1,n)-1}$ and $B_{K(1,n)+1} \ldots B_{n-1}$.

The section $(B_1 \ldots B_{K(1,n)-1})$ is further subdivided by the formation of a base pair between $B_{K(1,n)-1}$ and $B_{K(1,[K(1,n)-1])}$. As shown in Fig. 3, the repetition of this procedure for sections of decreasing size generates a group of nested structures. In each case, the base pair chosen is the one that leads to maximal base pair formation for the entire section. Thus, the final folded form contains the maximum number of base pairs that can be found for the entire nucleotide sequence, and the individual base pairs are precisely specified.

Table 1. Matrices $M(i,j)$ and $K(j,i)$ for the 25-nucleotide illustration shown in Fig. 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1		(1)	0	0	3	0	0	3	3	0	0	3	3	3	0	1	0	0	3	3	0	3	0	0	(3)
2	1		0	0	2	0	0	2	2	0	0	2	2	2	2	0	0	0	2	2	0	2	0	0	2
3	1	0		0	3	0	0	3	3	0	0	3	3	3	0	15	0	0	3	3	0	3	0	0	3
4	1	0	0		4	0	0	4	4	0	0	4	(4)	0	0	14	14	0	4	0	14	0	(14)	0	4
5	2	1	1	1		(5)	0	7	0	5	0	(7)	0	0	0	5	5	5	0	0	5	0	5	5	0
6	2	1	1	1	1		0	6	6	0	0	6	6	0	0	14	14	0	6	0	14	0	14	0	6
7	2	1	1	1	1	0		7	0	9	0	7	0	0	0	9	9	9	0	0	9	0	9	9	0
8	3	2	2	2	2	1	1		0	8	(8)	0	0	0	0	8	8	8	0	0	8	0	8	8	0
9	4	3	3	3	2	2	1	0		(9)	0	11	0	0	0	9	9	9	0	0	9	0	9	9	0
10	4	3	3	3	3	2	2	1	1		0	10	10	0	0	14	14	0	10	0	14	0	14	0	10
11	4	3	3	3	3	2	2	2	1	0		11	0	0	0	13	13	13	0	0	13	0	13	13	0
12	5	4	4	4	4	3	3	2	2	1	1		0	0	0	12	12	12	0	0	12	0	12	12	0
13	6	5	5	5	4	4	3	2	2	2	1	0		0	0	13	13	13	0	0	13	0	13	13	0
14	7	6	6	5	4	4	3	2	2	2	1	0	0		0	14	14	0	18	0	14	0	14	0	18
15	7	7	6	5	4	4	3	2	2	2	1	0	0	0		(15)	0	0	17	17	0	(17)	0	0	17
16	8	7	7	6	5	5	4	3	3	3	2	1	1	1	1		0	0	16	16	0	16	9	0	16
17	8	7	7	7	6	6	5	4	4	4	3	2	2	2	1	0		0	17	17	0	17	9	0	17
18	8	7	7	7	7	6	6	5	5	4	4	3	3	2	1	0	0		(18)	0	(20)	0	20	0	18
19	9	8	8	8	7	7	6	5	5	5	4	3	3	3	2	1	1	1		0	19	0	19	19	0
20	10	9	9	8	7	7	6	5	5	5	4	3	3	3	3	2	2	1	0		20	0	20	0	24
21	10	9	9	9	8	8	7	6	6	6	5	4	4	4	3	2	2	2	1	1		21	0	0	21
22	11	10	10	9	8	8	7	6	6	6	5	4	4	4	4	3	3	2	1	1	1		22	0	24
23	11	10	10	10	9	9	8	7	7	7	6	5	5	5	4	3	3	3	2	2	1	1		0	23
24	11	10	10	10	10	9	9	8	8	7	7	6	6	5	4	3	3	3	3	2	1	1	0		24
25	12	11	11	11	10	10	9	8	8	8	7	6	6	6	5	4	4	4	3	3	2	2	1	1	

$M(i,j)$ — $K(j,i)$

The base pairs in the final folded form are: 3 and 25, 1 and 2, 14 and 23, 13 and 24, 17 and 22, 7 and 12, 15 and 16, 20 and 21, 5 and 6, 8 and 11, 18 and 19, and 9 and 10.

Fig. 4 shows the best folded form obtained for the 25-nucleotide example discussed above. The manner in which it was obtained is shown in Table 1. The values for $K(i,j)$ are stored in the upper right-hand side of the table. Because $M(i,j)$ and $K(i,j)$ are constructed for $i < j$, both matrices were half empty and were combined. To save space, $M(i,j) = K(j,i)$. The values of K chosen for the final folding are indicated by circles. The base pair that occurs between nucleotides 3 and 25 was found in $K(1,25)$. It divides the entire sequence into two sections. Subsequent base pairs were obtained for subsections as described above.

Algorithm for Structures with Minimal Free Energy. In the preceding discussion, two basic strategies were utilized in searching for maximally paired loop structures. Inductive optimization was used to fill the matrices $M(i,j)$ and $K(i,j)$ with information on the number and position of the base pairs formed in nucleotide segments of increasing size. Backtracking was utilized to obtain the structure with the best folded form from the $K(i,j)$ matrix. A similar search procedure can be used to obtain structures with minimal free energy. Only one simple backtracking routine has been added to determine the location of each newly formed base pair with respect to its neighbors. In addition, a mechanism for closing single-stranded loops has been introduced.

The basic approach is similar to the one described for the algorithm for maximal matching. For the nucleotide sequence $B_1 \ldots B_n$, we wish to know whether the bond B_xB_y will be found in the structure with minimal free energy (Fig. 1A). To obtain

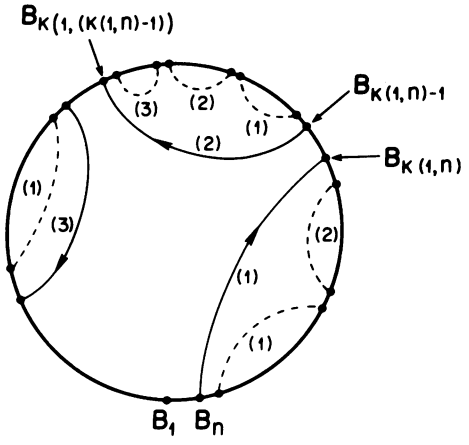

FIG. 3. The backtracking procedure. Full lines indicate first-generation backtracking steps numbered according to the order in which they are obtained. These yield the main branches of the secondary structure tree. Broken lines indicate second-generation backtracking done for each branch separately.

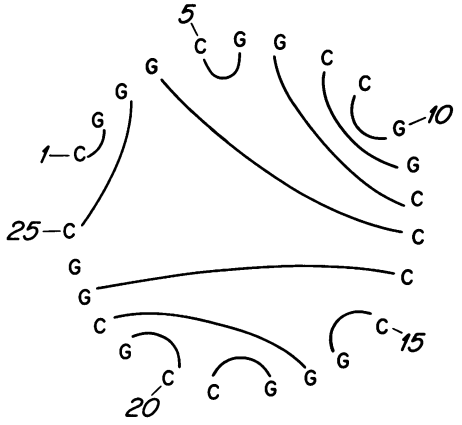

FIG. 4. Secondary structure for the 25-nucleotide chain with maximal pairings as deduced from the algorithm for maximal matching.

this information, the folding rules of the algorithm are modified to allow an estimate of the free energy of loop structures based on sequence data. The method for doing this is well established (1–6) and is determined from the sum of the contributions from helices, single-stranded loops, and bulges. The algorithm is further modified so that it can identify the location of a given base pair with respect to adjacent helical or single-stranded regions.

The search procedure is conducted in the following manner. Each nucleotide segment is examined for the structure with minimal energy by the inductive method described above. Eq. 1 becomes:

$$E(i,j) = \min. \begin{cases} E(i,k-1) + E(k+1,j-1) + E_{kj} \\ E(i,j-1) \quad i \le k < j - l \text{ min.} \end{cases} \quad [2]$$

That is, the energy (E) of the minimal energy structure formed between B_i and B_j is determined by examining each possible base pair B_kB_j within the region. In each instance the energy of the entire segment is determined by adding the energy contributions of the segments to the left and right of B_kB_j to that of B_kB_j itself. The lowest energy value found is stored in the matrix $M(i,j)$. The position of B_k that achieves this value is stored in $K(i,j)$. The minimum hairpin loop allowed is l min = 3. The final folding of the molecule is determined by the same backtracking procedure as described in the algorithm for maximal matching.

The value of E_{kj} will vary depending on whether the base pair B_kB_j is associated with an adjacent base pair, a single-stranded bulge, or an internal loop or a branched structure. A partial backtracking procedure is utilized to locate and evaluate the position of B_kB_j with respect to base pairs adjacent to it. The backtracking procedure is similar to the one used to find the best folded form from the K matrix. However, in this instance, only one generation of backtracking is required. We begin by allowing k_1 to equal $K(k+1,j-1)$; i.e., k_1 is the nucleotide between $k+1$ and $j-1$ which gives a structure with minimal energy when paired with $j-1$. Because the entire sequence is analyzed in an inductive manner, this information was stored in the K matrix during a previous run. Should $K(k+1,j-1) = 0$, the search is continued with decreasing values of j. For nonbranched loops, the value of k_1 alone is sufficient to locate the structure adjacent to B_kB_j as illustrated in Fig. 5. In the simplest case, $k_1 = k+1$ and B_{k+1} pairs with B_{j-1}; i.e., B_kB_j lies adjacent to a base pair (Fig. 5A). When $k_1 > k+1$, a single-stranded region occurs on the left side of the loop. This region can form a bulge or be part of an internal loop, depending on whether the opposite strand also contains nonpaired nucleotides. A single-stranded region is formed on the right side when B_{k_1} pairs with $B_j < j-1$. Fig. 5 B–D illustrates several of these situations.

Branch structures occur when several independent hairpin loops can form between B_k and B_j. These are identified by completing the first generation of backtracking as illustrated in Fig. 3. We let $k_2 = K(k+1,k_1-1)$, etc., and continue until the last element is found. [Because at least five nucleotides are required to form a hairpin loop, the search procedure is stopped when $k_n - (k+1) < 5$.] The number of branches is given by n. Fig. 5E illustrates the identification of a structure with two branches.

The value that E_{kj} assumes when it closes a bulge or internal loop is always higher than the energy value of the preceding helix. To allow closure of these structures, we have assigned an energy value to unpaired nucleotides in the section $B_i \ldots B_k$ equal to a bulge loop containing the same number of nucleotides. The value is used temporarily and does not enter in the final energy calculation of the correctly folded structure. This simple device has allowed us to fold a number of large RNA structures correctly (see below). However, other methods of loop closure are still under investigation.

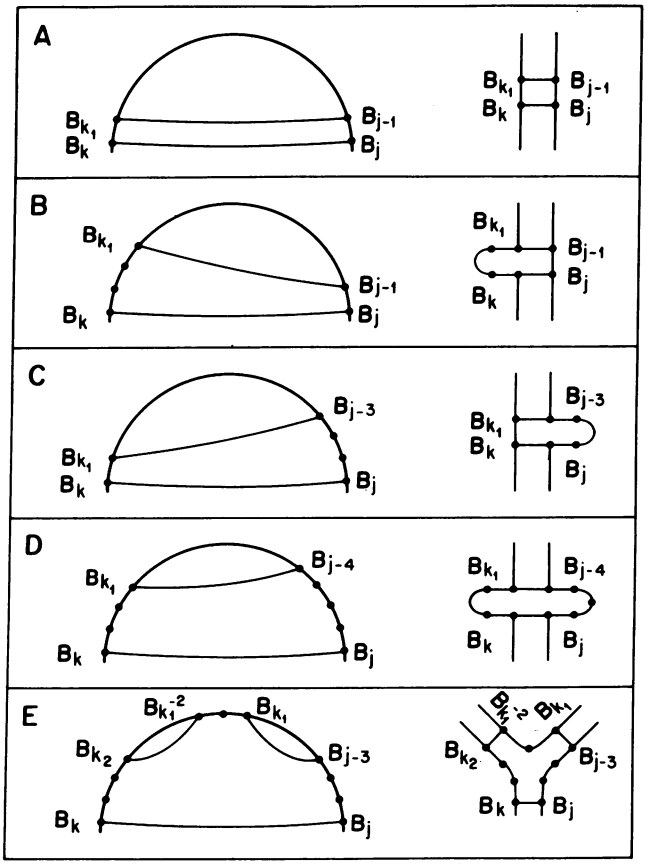

FIG. 5. Illustration of the partial backtracking procedure utilized to identify the structure located adjacent to the base pair B_kB_j. The left and right sides of each panel represent the same hydrogen-bonded structure drawn either in an extended or condensed form. (A) $k_1 = 1$. The base pair B_kB_j lies adjacent to the pair $B_{k+1}B_{j-1}$. (B) $k_1 = 3$. The base pair B_kB_j lies adjacent to a single-stranded bulge of two nucleotides on the left side which forms when B_{k+3} pairs with B_{j-1}. (C) $k_1 = 1$. The base pair B_kB_j lies adjacent to a single-stranded bulge of two nucleotides located on the right. It is formed when B_{k+1} pairs with B_{j-3}. (D) $k_1 = 3$. The base pair B_kB_j lies adjacent to an internal loop formed when B_{k+3} pairs with B_{j-4}. (E) $n = 2$. A branched structure. The two bonded pairs closest to B_kB_j are $B_{j-3}B_{k_1}$ and $B_{k_1}B_{k_2}$. As indicated in the text, the five unpaired nucleotides in the center of the junction are evaluated as if they belong to a five-membered internal loop.

The computer instructions for the above algorithm are simple and consist, in essence, of three nested "do" loops which shift the length and position of the nucleotide segment being analyzed and optimize base-pairing within each segment. The time requirement is $\approx N^3$ for a chain N nucleotides long. This is substantially faster than other existing codes (2, 7–9) and allows massive applications. The core requirement is N^2. The actual write-up contains approximately 500 instructions (it can be obtained from the first author upon request).

RESULTS

The present fast algorithm first arose in efforts to solve the stable loop structures seen by electron microscopy in the RNA bacteriophage MS2 (13). The results will be presented in detail elsewhere. The program was run on an IBM 370/168 computer for MS2 sections between 100 and 350 nucleotides long. The time (T) required was 15 sec for $N = 100$, 54 sec for $N = 200$,

3 min for $N = 300$, and 5 min for $N = 350$ nucleotides. Thus, these results fit well with the expected $T \approx N^3$ behavior.

The program has also been applied to the potato spindle tuber viroid ($N = 359$ nucleotides). More than two-thirds of the base pairs found were identical with those suggested in the original publication on the structure of this RNA (13). The differences were located in three small regions of the structure. In one case, the computer found an equally stable alternate structure; and in two cases slightly better structures were obtained. However, all of the differences observed were minor in nature, and the overall appearance of the folded molecule was virtually identical with the published structure. The whole folding of the viroid RNA was carried out in a single run and required exactly 5 min, illustrating the efficiency of the approach.

DISCUSSION

We have presented an approach to the combinatorial problem of finding a planar secondary structure with the lowest free energy that is systematic and builds an optimal structure in a straightforward inductive procedure based on an exact mathematical algorithm. Our approach is both simpler and faster than other existing procedures and allows application on a scale that could not be contemplated previously. It is subject to one serious limitation. The computer core required is N^2. This prevents folding sequences whose length exceeds 1000 nucleotides in a single run on existing computers. Very large runs could be performed on computers with virtual memory but the running time would be extremely long.

The algorithm was tested for its ability to fold a number of sequences of biological interest. The loop structures predicted by the algorithm for several sections of RNA from bacteriophage MS2 were compared with structures obtained by visual inspection and with those generated by other computer programs. The structures generated by the algorithm were always found to be identical with or better than structures obtained by other procedures. As described above, the folding obtained for the RNA from the potato spindle tuber viroid was also similar to the structure that has been published (13). The additional tests were performed with a number of different tRNA sequences. The minimal energy cloverleaf form was not obtained for any of these structures. The source of the difficulty with tRNA sequences is not yet clarified. Recent studies show that a simple modification of the algorithm which allows it to look forward when closing single-stranded loops permits us to fold many tRNAs correctly. These results will be described in detail elsewhere.

The approach we describe here is different than others in that it leads to the formation of one optimal structure rather than a series of related, alternate structures. Despite the limitations discussed above, the advantage of our approach lies in its extreme rapidity and the fact much larger sequences can be searched than with other programs that have been developed to date. It is clear, however, that some biological applications may require information on alternate, less-stable conformations in addition to an optimal structure. In such situations, a combined approach with other programs may prove useful, allowing examination of the nature of particular structures in more detail once they have been located within larger sequences.

The present algorithm was developed to facilitate our studies on loop structures seen by electron microscopy of single-stranded RNA from bacteriophage MS2. The structures that have been visualized differ completely from those predicted previously from sequence data (14). It will be of interest to determine whether the use of high-speed algorithms that can scan large sequences for optimal structures will improve our ability to predict these structures correctly. Alternatively, information from electron microscopy could be utilized to guide a computer search for the nucleotide sequences most compatible with electron microscopic data.

We thank J. L. Sussman and C. Sander for helpful discussions of this work. We also thank the staff of the Weizmann Institute Computer Center for their support. The research was supported in part by National Institutes of Health Grant AI15273 to A.B.J.

1. Delisi, C. & Crothers, D. M. (1971) *Proc. Natl. Acad. Sci. USA* **68**, 2682–2685.
2. Tinoco, I., Jr., Uhlenbeck, O. C. & Levine, M. D. (1971) *Nature (London)* **230**, 362–367.
3. Gralla, J. & Crothers, D. M. (1973) *J. Mol. Biol.* **73**, 497–511.
4. Tinoco, I. Jr., Borer, P. N., Dengler, B., Levine, M. D., Uhlenbeck, O. C., Crothers, D. M. & Gralla, J. (1973) *Nature (London) New Biol.* **246**, 40–41.
5. Borer, P. N., Dengler, B., Tinoco, I., Jr. & Uhlenbeck, O. C. (1974) *J. Mol. Biol.* **86**, 843–853.
6. Gralla, J. & Crothers, D. M. (1973) *J. Mol. Biol.* **78**, 301–319.
7. Salser, W. (1977) *Cold Spring Harbor Symp. Quant. Biol.* **62**, 985–1002.
8. Pipas, J. M. & McMahon, J. E. (1975) *Proc. Natl. Acad. Sci. USA* **72**, 2017–2021.
9. Studnicka, G. M., Rahn, G. M., Cummings, I. W. & Salser, W. A. (1978) *Nucleic Acids Res.* **5**, 3365–3387.
10. Waterman, M. S. (1978) *Studies in Foundations and Combinatorics, Advances in Mathematics Supplementary Studies*, Vol. 1, pp. 67–211.
11. Nussinov, R. (1977) Dissertation (Rutgers Univ., New Brunswick, NJ).
12. Nussinov, R., Pieczenik, G., Griggs, J. R. & Kleitman, D. J. (1978) *SIAM J. Appl. Math* **35(1)**, 68–82.
13. Gross, H. J., Domdey, H., Lossow, C., Jank, P., Raba, M., Alberty, H. & Sanger, H. L. (1978) *Nature (London)* **273**, 203–208.
14. Jacobson, A. B. & Spahr, P. F. (1977) *J. Mol. Biol.* **115**, 279–294.