Bulletin of Mathematical Biology Vol. 46, No. 4, pp. 591–621, 1984.
Printed in Great Britain

0092-8240/84$3.00 + 0.00
Pergamon Press Ltd.
Society for Mathematical Biology

RNA SECONDARY STRUCTURES AND THEIR PREDICTION[1]

■ MICHAEL ZUKER
Division of Biological Sciences,
National Research Council of Canada,
Ottawa, Canada K1A 0R6

■ DAVID SANKOFF
Centre de Recherche de Mathématiques Appliquées,
Université de Montréal,
Montréal, Canada H3C 3J7

This is a review of past and present attempts to predict the secondary structure of ribonucleic acids (RNAs) through mathematical and computer methods. Related areas covering classification, enumeration and graphical representations of structures are also covered. Various general prediction techniques are discussed, especially the use of thermodynamic criteria to construct an optimal structure. The emphasis in this approach is on the use of dynamic programming algorithms to minimize free energy. One such algorithm is introduced which comprises existing ones as special cases.

1. Introduction. A ribonucleic acid (RNA) molecule consists of a chain of ribonucleotides linked together by covalent chemical bonds. Each ribonucleotide contains one of the four bases: adenine (A), cytosine (C), guanine (G) or uracil (U), and the specific sequence of bases along the chain, the primary structure of the molecule, determines what kind of RNA it is.

In the cell, an RNA chain bends and twines about itself. Bases in close proximity form weak chemical bonds (hydrogen bonds) with one another if they are complementary: A with U and G with C. These Watson–Crick base pairs permit the molecule to assume a stable three-dimensional conformation characterized by various loops and twists. This tertiary structure determines the biochemical activity of the RNA molecule. Much effort has been invested into deductive methods for inferring tertiary structure based only on knowledge of the primary structure, since experimental techniques such as X-ray diffraction or biochemical probes are extremely costly and time consuming, if they are available at all, and generally are insufficient to determine the structure.

Biologists have simplified the study of the tertiary structure of an RNA molecule by focusing attention simply on what base pairs are involved in it. This collection of base pairs is referred to as its secondary structure. Figure 1

[1] Issued as NRCC No. 23684.

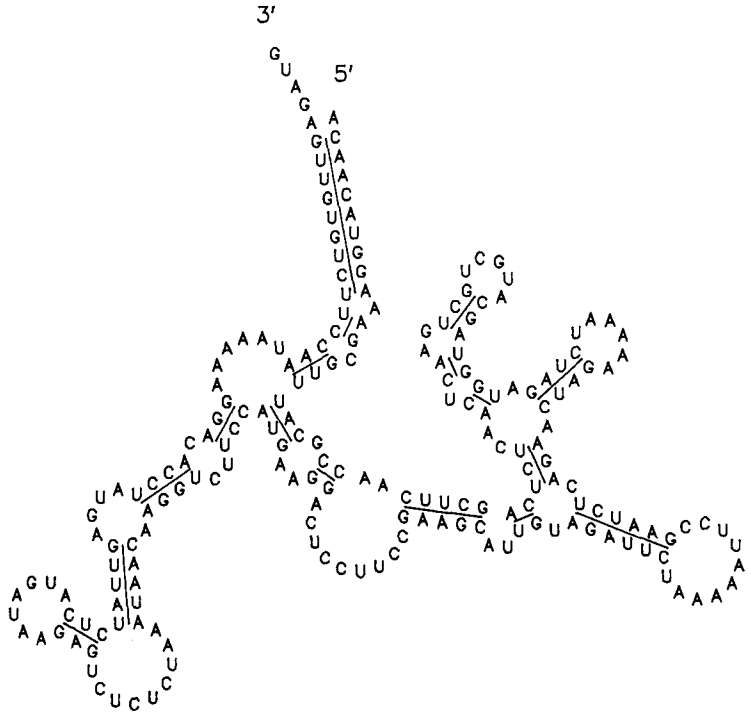

Figure 1. Secondary structure of a fragment of the Cauliflower Mosaic Virus. The linear structure begins at the 5′ terminus and continues to the 3′ terminus. The solid lines are drawn between complementary strands of hydrogen bonded nucleotides.

depicts an RNA chain folded in such a way as to illustrate the pairs constituting its secondary structure. Predicting secondary structure first and then proceeding on to tertiary structure has been a fruitful, if not infallible, approach. This review is confined to methods of secondary structure prediction (folding prediction) and closely related problems such as counting and mechanical drawing of structures.

There are three techniques which have been used to predict secondary structure. The first is to examine all possibilities, usually with the help of graphical procedures. The second is to invoke the laws of thermodynamics and to try to compute a conformation of minimum free-energy. The third approach uses phylogeny, and can be used if the sequences for functionally identical molecules have been determined for several organisms or organelles. If two or more molecules have closely related primary structures or identical biological functions, the strategy is to search for a secondary structure common to all of them.

2. Definition of Structure. We number the bases of an RNA sequence from

1 (called the 5′ terminus) to N (the 3′ terminus). A secondary structure is defined as a set S of pairs $i.j$ where $1 \leqslant i < j \leqslant N$. Each such pair of integers represents the pairing of the ith ribonucleotide in the molecule with the jth one. Not every set of pairs represents a valid secondary structure, however, with chemical and stereochemical constraints eliminating most possibilities. For example, it is reasonable in most cases to exclude sets of pairs in which some A terms pair with C terms, or G with A, or U with C. We may also exclude, as a first approximation, conflicting pairs—cases in which the same integer appears in more than one pair. As we shall see, however, most of these constraints may be formulated in terms of the thermodynamic instability of structures containing certain pairs or sets of pairs which will then be automatically excluded by any stability maximizing algorithm used to select a 'best' structure.

At the outset, then, we need impose but a single strong constraint, aside from the exclusion of conflicting pairs. This constraint ensures a planar, unknotted appearance for the structure: if $h < i < j < k$, then no secondary structure can contain both $h.j$ and $i.k$. Base pairs which violate this rule are said to form a *knot*. Note that this is not the same as the usual mathematical definition of a knot (see Figure 2). Ninio (1971) discusses this and

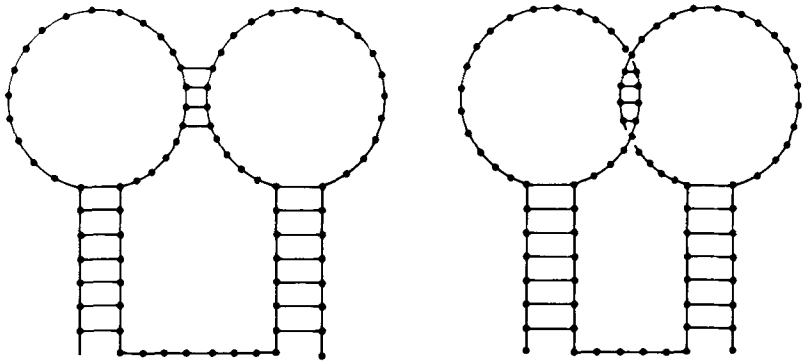

Figure 2. In the two similar configurations depicted above, only the one on the right contains a true mathematical knot, yet both depict identical, knotted secondary structures.

other topological properties of RNA secondary structure representations. Ruling out knots is basic to most secondary structure prediction algorithms, but given that knots do occur in some molecules, this constraint may seem rather arbitrary. However, if they are allowed, the prediction problem becomes much more difficult mathematically, and little realism is gained, since nothing specific is known about the thermodynamics of knotted structures. Fortunately, it is quite legitimate to regard these knots as part of

tertiary structure, and to relegate the problem of detecting them to a later stage (e.g. Studnicka *et al.*, 1978). For example, the accepted model for transfer RNA is a clover-leaf structure with no knotted base pairs (Figure 3). In reality, X-ray crystallography has shown that the three-dimensional structure is indeed knotted, but it is only unpaired regions not included in the accepted secondary structure which are responsible for the knot (Kim *et al.*, 1974). This and other examples have corroborated the working hypothesis that correct secondary structures can usually be established without reference to tertiary interactions, at least as a first step. The knot constraint is the key to most of the mathematical and computer work done on secondary structures since it ensures all structures are essentially planar and admit a simple decomposition into easily analyzable substructures.

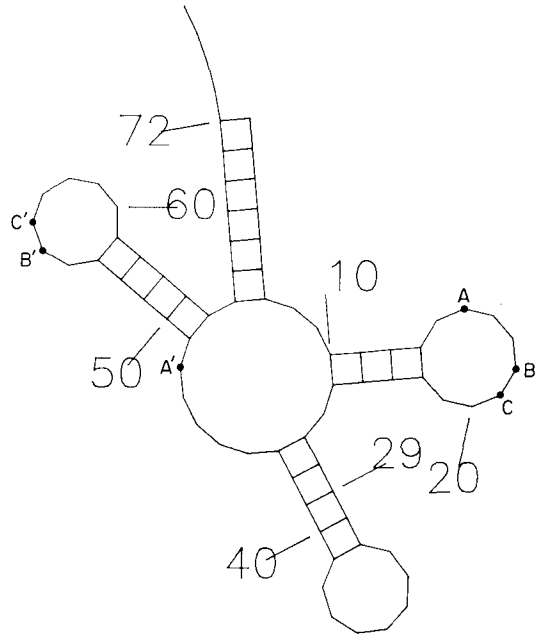

Figure 3. The clover-leaf model for yeast phenylalanine transfer RNA. Specific bases have been eliminated to emphasize the generality of the structure. In the tertiary structure (Kim *et al.*, 1974), hydrogen bonds exist between A and A′, B and B′, and C and C′. These additional bonds create a knotted structure.

3. Decomposition. We can analytically decompose any given secondary structure, S, in a unique way into a number of substructures such that each sequence term is contained in exactly one such substructure. Furthermore, the inventory of substructures we need to account for all possible S is quite small.

Suppose i and j are paired in S and $i < r < j$, but S contains no pair $x.y$ such that $i < x < r < y < j$. Then we say r is accessible from $i.j$. If $p.q \in S$ and p and q are accessible from $i.j$, we also say the pair $p.q$ is accessible. The $k - 1$ pairs and u unpaired terms accessible from $i.j$ constitute the k-cycle (also k-loop) closed by $i.j$. This is in contrast to the definition given by Sankoff *et al.* (1983) which includes the closing pair in the k-cycle. We call the accessible pairs the interior pairs of the cycle, and the closing pair the exterior pair.

If $k = 1$, the u terms between i and j form a hairpin loop. If $k = 2$ and $u = 0$, the pair $i + 1.j - 1$ constitutes a stacked pair. If $k = 2$ and $u > 0$, the 2-cycle closed by $i.j$ is either a bulge or an interior loop, depending on whether one of $i + 1$ or $j - 1$ is paired in S, or neither is. A k-cycle where $k > 2$ is called a multiple loop or multiloop. Those sequence terms contained in no k-cycle are called external.

It is not hard to show that no term belongs to more than one k-cycle, so that every term is either external (i.e. not accessible from any $i.j$) or belongs to exactly one stacked pair, hairpin, bulge, interior loop or multiple loop. Figure 4 illustrates the five types of substructures defined above.

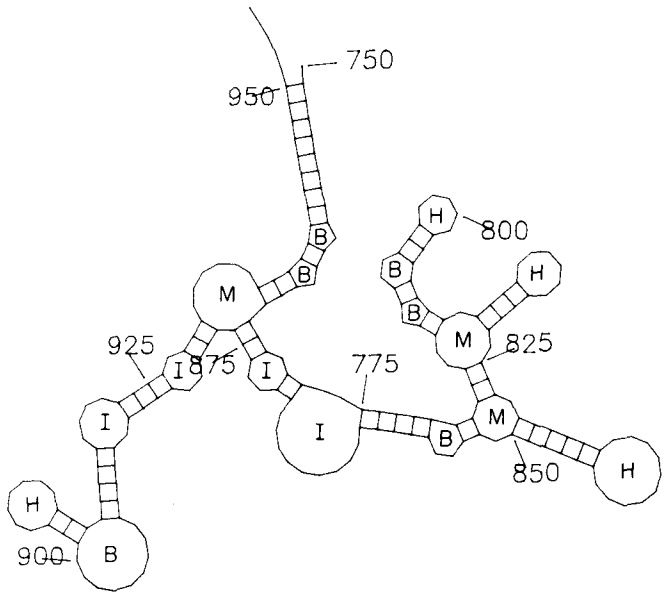

Figure 4. This is a more abstract representation of the same structure presented in Figure 1. The linear structure is depicted by the successive line segments forming the outline of the figure. Hydrogen bonds are drawn as short line segments which create ladder-like stacking regions. Other k-cycles are identified with the following code: B, bulge loop; I, interior loop; H, hairpin loop; M, multiloop.

4. Representations. There are a number of ways of representing a secondary structure which are more useful than simply listing a set of pairs. The most straightforward way is pictorially, as in Figures 1 and 4, where the RNA chain is represented by a curved line connecting a series of equidistant points disposed in such a way as to ensure that pairs of complementary bases in the secondary structure can be joined by short segments of fixed length. Such two-dimensional representations are used universally by biologists and have been used since the beginning of investigations on RNA secondary structure (e.g. Fresco *et al.*, 1960). We call this, somewhat loosely, the normal representation of secondary structure. The 'knot' constraint assures us that a planar pictorial representation is always possible without overlap, i.e. without the line representing the chain ever crossing itself, though achieving this may require irregular deformations of the looped areas of the structure. It should be noted that many knotted structures still admit a planar representation.

A more abstract type of representation was introduced by Nussinov *et al.* (1978). The bases of the RNA molecule are placed equidistant to one another along the circumference of a circle. The covalent bonds linking bases are represented by the arcs of the circle between them. Hydrogen bonds are represented by chords joining base-paired nucleotides, as in Figure 5. When viewed as a graph, the vertices are the bases, the edges are the covalent or hydrogen bonds and the faces are the collection of all k-cycles defined earlier. Although this circular representation is topologically equivalent to the normal representation in terms of mathematical graph theory, it has the special geometric property that no two chords intersect if and only if the secondary structure is unknotted.

A number of computer programs have been written to produce the normal representation. The computer programs of Studnicka *et al.* (1978) and Zuker and Stiegler (1981) both produce a line-printer output of a normal representation which are not satisfactory for large molecules whose structures are highly branched. Feldmann (unpublished) has written a program in SAIL called NUCSHO, producing a line-printer output which is very elegant and which avoids overlaps. It can handle up to 800 or so bases. Most other pictorial representations are also for video terminals or plotting devices. Osterburg and Sommer (1981) describe a program which places the closing pairs of a multiple loop equally spaced on a circle. Stacking regions interrupted by bulge or interior loops continue as 'ladders' along the same axis (see Figure 6). In general, overlaps occur with this method. A rather cumbersome feature to remover overlaps by rotating portions through specified angles has been added by Zuker (unpublished). Lapalme *et al.* (1982) produce a more pleasing output and include an interactive routine for achieving planarity. Shapiro *et al.* (1982) use the cross-hairs features of some

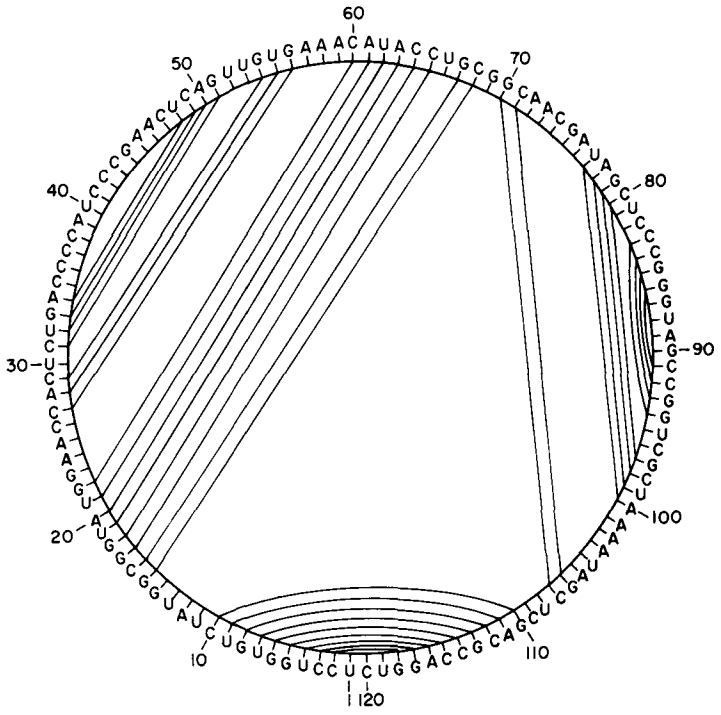

Figure 5. The secondary structure of *Anacystic nidulans* 5*S* rRNA using the circle representation of Nussinov. Some of the chords have been drawn as circular arcs for clarity.

video terminals to allow the user to point to regions that should be rotated or redrawn in a larger size. An elegant improvement of this method (Shapiro *et al.*, 1984) has an automatic untangle feature which places the parallel hydrogen bonds of stacked regions at the same angle to each other as they would be in the graphical representation of Nussinov *et al.* (1978). The secondary structure prediction method of Rindone (Auron *et al.*, 1982) actually uses a plot of the secondary structure to aid the computer in refining the structure. Changes in base-paired regions are indicated to the computer by means of a light pen pointing to the drawn structure or by typing indices.

A third type of representation is in terms of a rooted tree or forest of rooted trees, in graph theory terms, and differs from the representations mentioned above. Each pair in the secondary structure is represented by a vertex of a graph, and a directed edge leads to one vertex from another if the pairs they represent are the exterior and (one of the) interior pair(s) of the same cycle. That a tree (or forest) is formed rather than a more complex graph when a secondary structure is thus represented is a consequence of the

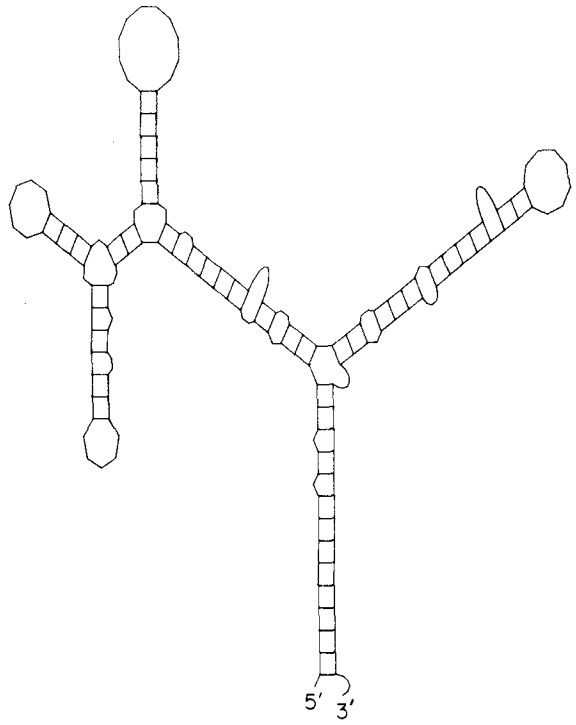

Figure 6. The secondary structure of the same Cauliflower Mosaic Virus fragment depicted in Figures 1 and 4 using the graphics program of Osterburg. The nucleotide drawing option produces an overly cluttered picture.

knot constraint. A certain amount of information is lost in this representation, such as how many unpaired bases there are in loops or external regions, and the orientation of the molecule, i.e. which part of the structure is close to the 5′ end ($i = 1, 2, \ldots$) and which part is close to the 3′ end ($i = N, N-1, N-2, \ldots$). The orientation information may be incorporated as in Figure 7(a) by imposing a left-to-right order among the edges directed away from each vertex, and on the roots of the individual trees, if the structure is a forest.

The tree representation permits a useful classification of structures according to their complexity (Waterman, 1978), as illustrated in Figure 8. A tree of order 1, the simplest kind of tree, consists of a monovalent root vertex connected through a sequence of bivalent vertices to a terminal monovalent vertex representing the closing pair of a hairpin. More complex trees are created by an iterative process of adding branches. An order n tree consists of a 'central stem'—an order 1 tree—together with two or more order $n-1$ trees attached to the stem. This attachment is effected by an edge to the root of the order $n-1$ tree from any of the bivalent vertices of the stem, or from its root.

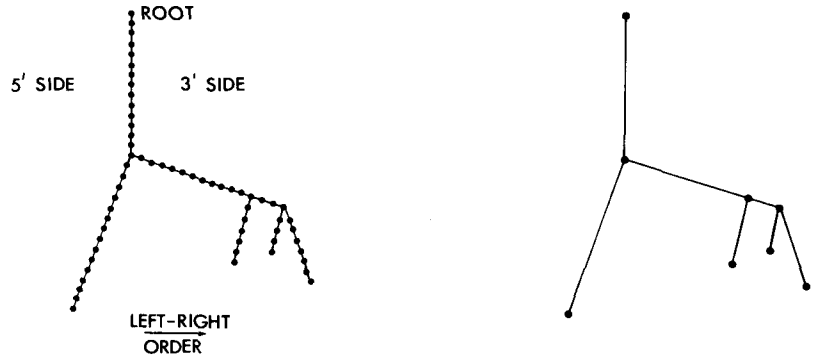

Figure 7(a). Tree representation of the Cauliflower Mosaic Virus structure shown in Figures 1, 4 and 6. Terminal vertices at the bottom of the Figure represent hairpin loops closing pairs 911–903, 845–834, 820–813 and 802–796 from left to right. (b) The shape of the same fragment.

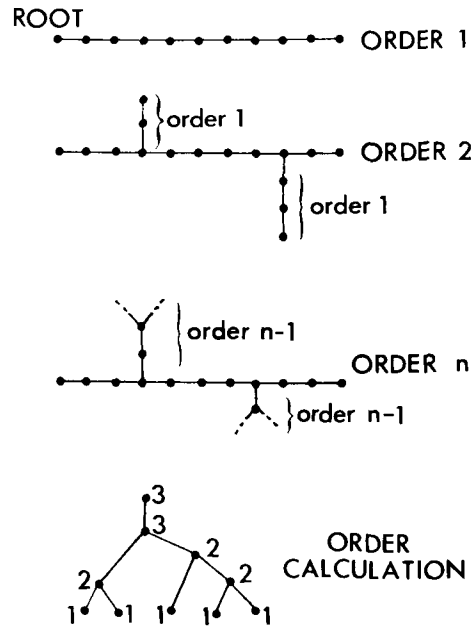

Figure 8. An illustration of the notion of the order of a tree and its calculation.

Given a tree, its order can be determined by a simple algorithm. Each terminal vertex is assigned the label '1'. For any vertex v whose outgoing edges all lead to previously labelled vertices, let r be the largest label. If this label occurs only once among these vertices, then v is also assigned label r. If, on the other hand, r occurs twice or more among these neighbouring vertices, v is assigned $r + 1$. When the algorithm terminates, the order of the tree is the largest label which has been defined.

5. Enumeration. Once the class of admissible structures has been defined, the first problem that presents itself is that of enumerating the number of structures that can be formed with N nucleotides. Let $T(N)$ be the number of secondary structures on a molecule with N labelled bases. Clearly $T(0) = T(1) = 1$, and it is stereochemically realistic to assume that two adjacent bases cannot form a hydrogen bond, so that $T(2) = 1$. For $N > 1$, the knot constraint ensures that $T(N)$ satisfies the recurrence

$$T(N+1) = T(N) + \sum_{k=0}^{N-2} T(k)T(N-k-1) \tag{1}$$

where the first summand represents the cases where the last base is not base-paired and each of the products represents the number of structures where the last base pairs with the $(k+1)$st. This formula is given by Waterman (1978), who also provides a generating function for the $T(N)$. The problem is taken much further by Stein and Waterman (1978), where the remarkable asymptotic formula

$$T(N) \sim \sqrt{\left(\frac{15+7\sqrt{5}}{8\pi}\right)} N^{-3/2} \left(\frac{3+\sqrt{5}}{2}\right)^N \tag{2}$$

is derived. This article generalizes the counting to cases where any two paired bases must have at least m intervening bases, which allows us to represent the biological situation where all hairpin loops must contain at least three unpaired bases. Explicit asymptotic results are derived for $m = 0, 1$ and 2. Two special cases are examined by Waterman (1978), where it is shown that there are precisely $2^{N-2} - 1$ secondary structures of length N with exactly one hairpin loop and that the number of structures of order 1 is asymptotically $K\lambda^N$ where λ is the largest root of $x^3 - 2x^2 - 1$ and K is a certain rational function of λ.

The above enumerations assume that base-pairing is possible between arbitrary pairs of nucleotides. The real situation is more complicated since the positions at which base pairs may occur, and hence the number of structures is dependent on the base composition of the actual sequence. If only G and A bases occur, no base pairs can form and there is only one possible structure.

A stochastic approach to this problem leads to interesting results. The bases of a molecule of length N can be regarded as independent and identically distributed random variables with probabilities $p(\mathrm{A})$, $p(\mathrm{C})$, $p(\mathrm{G})$ and $p(\mathrm{U})$ for the occurrence of A, C, G and U, respectively. The number $p = 2(p(\mathrm{A})p(\mathrm{U}) + p(\mathrm{C})p(\mathrm{G}))$ is the probability that any two bases can form a hydrogen bond. Let $\eta(i, j)$ be 1 if bases i and j can pair, and 0 otherwise. Clearly $E\{\eta(i, j)\} = p$, where E denotes mathematical expectation. Define the

random variable $R(N)$ to be the number of secondary structures on a random molecule of size N. As above, $R(0) = R(1) = R(2) = 1$ and (1) becomes

$$R(N+1) = R(N) + \sum_{k=0}^{N-2} R(k)R(N-k-1)\eta(k+1, N+1). \tag{3}$$

The three multiplicands in each sum are determined by sequence values from bases 1 to k, $k+2$ to N and bases $k+1$ and $N+1$, respectively. This makes them independent random variables. Taking mathematical expectations in (3) yields

$$E(N+1) = E(N) + \sum_{k=0}^{N-2} pE(k)E(N-k-1) \tag{4}$$

where $E(N)$ is defined to be the expected value of $R(N)$. Using the methods of Stein and Waterman (1978), it can be shown that there are constants H and α which depend on p such that

$$E(N) \sim HN^{-3/2}\alpha^N \tag{5}$$

where

$$\alpha = \left(\frac{1 + \sqrt{(1 + 4\sqrt{p})}}{2}\right)^2 \tag{6}$$

and

$$H = \frac{\alpha(1 + 4\sqrt{p})^{\frac{1}{4}}}{2\sqrt{\pi p^{3/4}}} \,. \tag{7}$$

When $p = 1$, this reduces to the Stein and Waterman result in (2), where $\alpha = \frac{1}{2}(3 + \sqrt{5}) = 2.618\ldots$. An interesting case to consider is the one when all nucleotides occur with equal probability. In this case, $p = \frac{1}{4}$ and $\alpha = 1 + \frac{1}{2}\sqrt{3} = 1.866\ldots$.

Though taking into account base complementarity reduces the overcount in Waterman's approach to enumerating secondary structures, it remains a rather high estimate of the number of biologically interesting structures. One problem is that it counts structures which contain pairs of bases which are not joined by hydrogen bonds even though they are in stereochemically favourable positions for base pairing. It would therefore be of interest to count only saturated structures; where no unpaired bases exist which could be paired without affecting the validity of the structure.

Another enumeration question concerns the number of different shapes of secondary structure. For example, though transfer RNAs may have many different lengths, and many different secondary structures,

they all have the same shape, known as the clover-leaf, as illustrated in Figure 3.

In the tree representation of a secondary structure, its shape is obtained by simply bypassing any sequence of bivalent vertices leading from a vertex A to a vertex B (neither bivalent) by a single edge from A to B, as in Figure 7(b). The problem of counting possible shapes can then be formulated in terms of counting the number of different rooted trees (or forests of rooted trees) with different left-to-right orders among the edges directed away from each vertex, all with a given number h of terminal vertices (i.e. hairpins).

The number, N, of different shapes of secondary structures with h hairpin loops, in which the 3′ and 5′ ends are paired, turns out to be $N(h) = 1, 1, 3, 11, 45, 197, \ldots$ for $h = 1, 2, 3, 4, 5, 6, \ldots$. This series is number 1163 in Sloane (1973) and counts the number of ways of parenthesizing a product of h terms. If we remove the restriction on the 3′ and 5′ pair, we can multiply the number of structures (for $h > 1$) by 2, since neither pairing nor unpairing this pair constitutes a many-to-one projection.

While distinguishing among secondary structures on the basis of every possible detail may be very costly in searching for optimal structures, it would not be useful in that context to try to evaluate only different shapes instead. Two molecules having the same shape may have very different structures when examined in detail, and very different stabilities.

6. Energy. In the ensuing sections we shall focus on methods for finding the thermodynamically most stable secondary structure for a given RNA molecule. Basic to all of this must be some way of evaluating the free energy $E(S)$ associated with any proposed structure S. The working hypothesis that makes this feasible is that if we decompose S into its disjoint substructures, with k-cycles $S_1, S_2, \ldots S_t$, then

$$E(S) = e(S_1) + e(S_2) + \ldots + e(S_t) \tag{8}$$

where $e(S_i)$ is the energetic contribution of the cycle S_i. The external bases do not contribute to the energy. The empirical knowledge needed to make use of this notion is the free-energy contribution of each of the various types of k-cycle.

A number of research programs in the early seventies contributed theoretical considerations and experimental results which enable us to assign free energy estimates with some accuracy to all k-cycles where $k = 1$ or $k = 2$. These values vary as a function of the loop type, the closing pair $i.j$ and the number of unpaired bases in the loop. Such work has been reported by Tinoco *et al.* (1971); Fink and Crothers (1972); Uhlenbeck *et al.* (1973); Gralla and Crothers (1973a and b); Tinoco *et al.* (1973) and by

Borer *et al.* (1974). This information has been summarized by Salser (1977) and is presented in Table I. More recently, these energies have been modified by Tinoco, as reported by Cech *et al.* (1983). Note that only stacked pairs contribute negative free-energy and hence provide stability to the structure. The restriction against hairpins containing less than three unpaired bases can be enforced by assigning a large destabilizing energy to this conformation. Similarly, non-Watson–Crick base pairs can be avoided by making them prohibitively expensive in terms of free energy. An exception must be made for G.U pairs, however, since these are observed to occur frequently in the interior of stacked regions.

TABLE I

Experimentally Determined Energies of 1 and 2-Cycles From a Variety of Sources as Summarized by Salser (1977)

BASE PAIRING ENERGIES IN TENTHS OF A KCAL/MOLE

STACKING ENERGIES (UG = GU)

INTERIOR CLOSING PAIR

EXTERIOR CLOSING PAIR	GU	AU	UA	CG	GC
GU	-3	-3	-3	-13	-13
AU	-3	-12	-18	-21	-21
UA	-3	-18	-12	-21	-21
CG	-13	-21	-21	-48	-43
GC	-13	-21	-21	-30	-48

BULGE LOOP DESTABILIZING ENERGIES BY SIZE OF LOOP

1	2	3	4	5	6	7	8	9	10	12	14	16	18	20	25	30
28	39	45	50	52	53	55	56	57	58	59	61	62	63	64	65	67

HAIRPIN LOOP DESTABILIZING ENERGIES BY SIZE OF LOOP

	1	2	3	4	5	6	7	8	9	10	12	14	16	18	20	25	30
CG CLOSING	999	999	84	59	41	43	45	46	48	49	50	52	53	54	55	57	59
AU CLOSING	999	999	80	75	69	64	66	68	69	70	71	73	74	75	76	77	79

INTERIOR LOOP DESTABILIZING ENERGIES BY SIZE OF LOOP

CLOSED BY	1	2	3	4	5	6	7	8	9	10	12	14	16	18	20	25	30
CG-CG	999	1	9	16	21	25	26	27	28	29	31	32	33	34	35	37	39
CG-AU	999	10	18	25	30	34	35	36	37	38	39	40	41	42	43	45	47
AU-AU	999	18	26	33	38	42	43	44	45	46	48	49	50	51	52	54	56

The use of an arbitrarily large destabilizing energy for stearically impossible hairpin loops containing 1 or 2 nucleotides is a convenient way of ensuring that they will not occur in structures predicted by energy minimization.

There are other approaches to the energy rules. Ninio (1979) and Papanicolaou *et al.* (1984) have experimented with the notion that realistic structures can be found algorithmically without excluding *a priori* all non-Watson–Crick pairs. The basic idea in the article by Ninio (1979) is to alter the energy rules so that the accepted clover-leaf structure for transfer RNAs will also be a minimum energy structure. Papanicolaou *et al.* (1984)

extend this principle to another class of RNAs (the 5*S* RNAs). Hofmann (Steger *et al.*, in preparation) has used thermodynamic calculations to extend the energy tables for folding at different temperatures. Tinoco (personal communication) has considered the notion that the destabilizing energy of loops might depend on their base composition. He has also considered the possible effect of exterior unpaired bases.

7. Historical Background. The first systematic approach to the prediction of secondary structure involved the construction of an $N \times N$ matrix where both the ith row and ith column correspond to the ith position of the sequence, and the (i, j) entry indicates whether $i.j$ is a Watson–Crick pair. Potential stems, or long stacks of base pairs, appear as diagonal patterns in the matrix. This information can then be used as the basis to a heuristic search for combinations of base-paired and unpaired regions which optimize the free energy (Tinoco *et al.*, 1971). Such methods are by no means obsolete. Quigley *et al.* (1984) use a matrix diagonal method which filters out less stable stacks of base pairs and also regions which are incompatible with data from chemical and enzymatic probes. Trifonov and Bolshoi (1983) combine a matrix method with filtering ideas borrowed from image processing to search for common base-pairing regions in related sequences. This is discussed further in Section 11.

The Pipas and McMahon (1975) algorithm represented the next logical step forward from the heuristic inspection of a matrix. A first routine in their program constructs a list of all possible stems or helical regions (sets of three or more base pairs stacked one over the other). A second routine compares all pairs of these regions for compatibility; two helical regions are compatible if they contain no base in common and produce no knot. The final part of the algorithm searches for the set of compatible regions having the lowest overall free energy. It does this by an exhaustive search, and keeping in storage the best M structures (where M may be fixed at any value) at all stages.

Though the Pipas–McMahon program contains a number of approximations and simplifications, it works well for relatively small molecules and can easily be improved to take into account more accurate energy calculations. It has a number of serious shortcomings, however, which make it infeasible for longer RNA molecules.

First, a special case of the search for a maximal set of compatible regions is well-known in computer science as the maximal clique problem. This problem is NP-complete, and no known search procedure can solve it in less than exponential time for all examples. It is not known whether compatibility matrices for helices in RNA molecules could theoretically be

this pathological, but it is clear that even typical (uncontrived) molecules will require excessive time as they become very long.

A second problem with Pipas–McMahon is that it may exclude two regions A and B as incompatible on the grounds that one base is in both. This base might be at the end of region A so that considering this region shortened by one base pair will result in a region compatible with B. Pipas and McMahon do not take account of this possibility.

Many of the shortcomings of the Pipas–McMahon method are overcome in one of the last of the non-dynamic programming methods to be developed; the APL program of Studnicka *et al.*, 1978. Unlike many of the earlier heuristic algorithms, the class of structures that are considered is carefully and explicitly spelled out. Like Pipas and McMahon, the algorithm begins by compiling a list of all possible pairing regions. Because even this list can be unacceptably long, there is a filtering option at this stage which retains only the most energetically favourable regions. The program then considers all pairs of conflicting regions. Base pairs are deleted from either or both regions until a hybrid region with minimum energy is discovered. The next stage of the program pieces together regions from the list to form structures without multiple loops. A final pass allows the creation of arbitrarily complex structures. The method as a whole is executed in a time proportional to N^5. According to the authors, a complete solution becomes impractical for sequences larger than 250–300 bases. It handles short sequences very well and, like Pipas and McMahon, has the advantage over the dynamic programming methods we will be discussing of yielding a whole range of solutions near the optimal energy.

The application of dynamic programming to the secondary structure problem seems to have been attempted independently by several groups (Waterman, 1978; Waterman and Smith, 1978; Nussinov *et al.*, 1978; Zuker and Stiegler, 1981; Mainville, 1981). This is not surprising, since folding is related to the notion of sequence alignment in the study of protein and nucleic acid homology. This problem had earlier been tackled by dynamic programming (e.g. Needleman and Wunsch, 1970). Broadly speaking, two different approaches have been taken. They differ basically in the treatment of multiple loops.

The first current can be seen in the work of Waterman (1978) and of Mainville (1981). Their algorithms are step-wise ones which first construct an optimal first-order structure. Successively higher-order structures are then computed in an iterative way using results from the previous pass. This approach is similar to the alignment algorithm developed by Sankoff (1972) and generalized by Sankoff and Sellers (1973) in which optimal alignments with 0, 1, 2, . . . gaps are computed in successive passes. Methods such as these are expensive to implement on a computer because storage

and CPU time requirements are high. As programmed, the Waterman algorithm can handle up to 200 bases.

The second approach, taken by Nussinov *et al.* (1978) and Zuker and Stiegler (1981), finds an optimal folding of arbitrary complexity in one pass. In this sense, it is similar to the Needleman–Wunsch alignment algorithm which allows an arbitrary number of gaps controlled only by the gap penalty. The original version by Nussinov *et al.* maximizes base pairing and ignores destabilizing effects of loops. Both a later version of this algorithm (Nussinov and Jacobson, 1980) and the method of Zuker and Stiegler (1981) incorporate the destabilizing effects of loops and assign weights to base-paired regions using generally accepted stacking energies (Salser, 1977) instead of merely maximizing the number of base pairs. The algorithms used in this second approach will be discussed in some detail in the next section.

8. Dynamic Programming Algorithms. For a given RNA sequence, let S be any secondary structure. Consider any pair $i.j$ in S, and let S_{ij} be the set of pairs $h.k$ in S such that $i \leqslant h < k \leqslant j$. S_{ij} is a secondary structure on $i, \ldots, j$ and the knot constraint ensures that its k-cycles are none other than all the k-cycles of S involving terms between i and j. If S is optimal then S_{ij} must also be optimal on $i, \ldots, j$, given that i and j are to be paired with one another. Otherwise, if S'_{ij} were better than S_{ij}, then $S' = (S \backslash S_{ij}) \cup S'_{ij}$ would be a secondary structure on $1, \ldots, N$ and the additivity of substructure energies would imply a contradiction

$$E(S') = E(S) - E(S_{ij}) + E(S'_{ij}) < E(S). \tag{9}$$

This is the basis for the dynamic programming approach; the optimal secondary structure for a sequence can be found by first determining the optimal structure for each part of that sequence. In this section, we present a general framework and basic algorithms adapted from Sankoff *et al.* (1983). Within this framework, we can discuss a number of variants and improvements drawn from this same and other sources. We will arrive at a synthesis representing the state of the art, with respect to generality of energy functions, optimality of solutions, and computational efficiency.

Let $C(i, j)$ be the minimal value possible for the energy $E(S_{ij})$ of a secondary structure S_{ij} on $i, \ldots, j$ given that i is paired with j. Then it follows from the additivity of the energies of substructures that

$$C(i, j) = \min_{k \geqslant 1} \left\{ \min_{\substack{s \text{ is a } k\text{-cycle} \\ \text{closed by } i.j}} \{e(s) + \sum_{\substack{p.q \\ \text{accessible} \\ \text{from } i.j}} C(p, q)\} \right. \tag{10}$$

for $i < j$, with the initial conditions $C(i, i) = \infty$. When a base pairing between

i and j is not possible, $C(i, j)$ is set to ∞. If $F(i, j)$ is defined to be the minimal energy for a structure irrespective of whether i is paired with j, then

$$F(i, j) = \min\{C(i, j), \min_{i \leqslant h < j} (F(i, h) + F(h + 1, j)\} \tag{11}$$

for $i < j$, with the initial conditions $F(i, i) = 0$. This follows from (8) since any secondary structure on $i, \ldots, j$ with i and j not paired with each other can be divided into two regions $(i, \ldots, h)$ and $(h + 1, \ldots, j)$ with no base pairing between these regions. $F(1, N)$ is then the free energy of the optimal or 'true' secondary structure.

To use the recurrence equations (10) and (11) to find this structure we must apply them to pairs (i, j) ordered in such a way as to ensure $C(i, j)$ and $F(i, j)$ are evaluated after $C(h, k)$ and $F(h, k)$ for all $i \leqslant h < k \leqslant j$. We will discuss useful orderings in the next section.

After $C(i, j)$ and $F(i, j)$ are calculated, they must be stored for use in evaluating pairs which come later in the order. It is also possible to store the k and the accessible $p.q$ which are responsible for the minimum in (10), and similarly for the h, if pertinent, in (11). It is then an easy matter to 'backtrack' through the (i, j) array to actually construct a structure with free energy $F(1, N)$. As will be discussed below, the 'pointer' array which stores the elements responsible for the minima is not usually necessary. It takes up much valuable computer space and speeds up a portion of the algorithm which is already extremely rapid when compared with filling the C and F arrays. Nussinov *et al.* (1978); Nussinov and Jacobson (1980); Comay *et al.* (1984) and Mainville (1981) make use of pointer arrays while Zuker and Stiegler (1981) do not.

Because the calculation of $C(i, j)$ in this algorithm involves evaluating all possible k-cycles, it must examine all $2k$-tuples of the form

$$i < p_1 < q_1 < \ldots < p_{k-1} < q_{k-1} < j. \tag{12}$$

Including all pairs (i, j), this takes time proportional to N^{2k}. Since the maximum k possible grows linearly with N, the algorithm requires exponential computing time, and is hence impractical.

There are two approaches to improving the speed of this algorithm. One requires a constraint on the mathematical form of $e(s)$ for all k-cycles s where $k > K$. This approach reduces computing time to be proportional to $N^{\max\{3,2K\}}$ at worst, but is limited in that it requires unrealistic assumptions if $K = 1$, and the case $K = 2$ still leads to an N^4 algorithm. The other approach is to limit the number of unpaired bases in 2-cycles, which reduces the computing time from quartic to quadratic for structures containing no multiple loops. This approach would seem to be defective in that optimal structures containing large bulge or interior loops will not be found. However,

a judicious combination of the two approaches leads to a feasible and realistic algorithm.

In the first approach, we assume $e(s) = a(i, j) + (k - 1)b + uc$ for a k-cycle s closed by $i.j$ where $k > K$, $a(i, j)$ depends on the closing pair $i.j$, and b and c are constants determining the contributions due to the $k - 1$ accessible pairs and u unpaired bases in the cycle s. Equations (10) and (11) are then replaced by:

$$C(i, j) = \min \begin{cases} \min\limits_{\substack{s \text{ is a } k\text{-cycle} \\ \text{closed by } i.j \\ (i \leqslant k \leqslant K)}} \{e(s) + \sum\limits_{\substack{p.q \\ \text{accessible} \\ \text{from } i.j}} C(p, q)\} \\ \\ \min\limits_{i<h<j-1} \{F_1(i+1, h) + F(h+1, j-1) + a(i, j)\} \end{cases} \tag{13}$$

$$F(i, j) = \min \begin{cases} \min\limits_{i \leqslant p < q \leqslant j} C(p, q) + (j - q + p - i)c + b \\ \\ \min\limits_{i \leqslant h < j} F(i, h) + F(h+1, j) \end{cases} \tag{14}$$

and

$$F_1(i, j) = \min_{\substack{i = q_0 \\ < p_1 < q_1 \\ < p_2 < q_2 \cdots \\ < p_{K-1} < q_{K-1} \\ < p_K = j}} \begin{cases} \sum\limits_{k=1}^{K-1} C(p_k, q_k) + (K-1)b \\ \\ \quad + \sum\limits_{k=0}^{K-1} (p_{k+1} - q_k - 1)c \end{cases} \tag{15}$$

with the initial conditions $F(i, i) = \infty$ and $F_1(i, i) = \infty$.

$F(i, j)$ is redefined here to be the minimum energy of a structure on $i, \ldots, j$ which contains at least one exterior pair. $F_1(i, j)$ is the minimum energy of a structure on $i, \ldots, j$ which contains exactly $K - 1$ exterior pairs. Both F and F_1 contain linear destabilizing energies according to the numbers of exterior bases and base pairs which they contain.

These recurrences ensure that a $(K + 1)$-loop or larger, closed by $i.j$, is made up of an optimal combination of a structure with exactly $K - 1$ external pairs on $i + 1, \ldots, h$, and a structure with at least one external pair on $h + 1, \ldots, j - 1$, for some h. Note that neither of these two substructures needs to be optimal itself since for K-loops and lower, the optimal $e(s)$ used in calculating C is generally not linear (as it is for $(K + 1)$-loops or larger) but is rather an arbitrary function of the composition of the loop.

The value $F(i, j)$ is no longer the minimum free energy of a structure on $i, \ldots, j$ as defined in Section 6, since it contains destabilizing energies due

to external unpaired bases and pairs. This way of imposing linearity in high order loops still leaves the energy function $e(s)$ free to take on any values when $k \leqslant K$, including those determined experimentally. Its weaknesses are that the linearity assumption is unrealistic for loops with large k (one would expect $e(s)$ to increase logarithmically from thermodynamic principles) and that it still requires computing time N^{2K}. Now, 2-cycles are very numerous in secondary structures, and 1-cycles are fairly numerous, while k-cycles with $k > 2$ are relatively few so that on biological grounds we can expect the linearity assumption not to have too drastic an effect if $K = 2$. When $K = 2$, F_1, and consequently (15), can be dropped. In this case (14) can be replaced by

$$F(i, j) = \min \begin{cases} C(i, j) + b \\ \min\limits_{i \leqslant h < j} \{F(i, h) + F(h + 1, j)\} \\ \min\{F(i + 1, j), F(i, j + 1)\} + c \end{cases} \tag{16}$$

where $F(i, i) = \infty$.

The original dynamic programming algorithm of the type discussed in this section (Nussinov, 1977; Nussinov *et al.*, 1978) was designed only to maximize the number of base pairs in a structure. This algorithm is equivalent to the modification described above where $K = 1$ and $b = c = 0$. The function $a(i, j)$ is either -1 or 0 depending on whether or not a base pairing between the ith and jth nucleotides is allowed or not. As described, this algorithm uses a single recursion equivalent to our calculation of F. The function C is not needed since

$$C(i, j) = a(i, j) + F(i + 1, j - 1) \tag{17}$$

in this special case. A generalization which optimizes a weighted base-pairing was reported by Nussinov and Jacobson (1980) in which the $a(i, j)$ could now take on arbitrary non-positive values. This article and subsequent ones (e.g. Comay *et al.*, 1984) have sacrificed the optimality of the algorithm in order to introduce experimentally determined energy rules for loop destabilizing energies. Multiloops are treated as interior loops; the destabilizing energy depends on the number of unpaired bases and the nature (G–C or A–U) of the accessible and closing base pairs. The algorithm implicitly assumes that (in our notation)

$$C(i, j) = a(i, j) + F(i + 1, j - 1) + e(s_{ij}), \tag{18}$$

where s_{ij} is the k-cycle comprising the external bases and pairs of the optimal structure on the subsequence from $i + 1$ to $j - 1$. The loss of optimality stems from the fact that the substructure formed by removing the $i.j$ base

pair is not always an optimal one. In general, there may be a suboptimal structure S on the subsequence from $i + 1$ to $j - 1$ (with energy greater than $F(i + 1, j - 1)$) which, together with a destabilizing energy $e(s)$ smaller than $e(s_{ij})$, yields a better overall energy for $C(i, j)$, where s is the k-cycle formed from the external bases and pairs of S. This problem was solved by Zuker and Stiegler (1981) who define a thoroughly optimal algorithm using published energies for 2-cycles while effectively setting the destabilizing effect of mutliloops to zero, as in Studnicka *et al.* (1978). This is exactly equivalent to the algorithm defined by equations (13) and (14) with $K = 2$ and $a(i, j) = b = c = 0$. The more complex algorithm alluded to in that article treats multiloops as interior loops with destabilizing energies from a published table. In the treatment of multiloops it risks the same type of suboptimality as that found in the algorithm of Nussinov and Jacobson (1980), but overall, it performs better because the more numerous bulge and interior loops are treated with complete rigour.

In the second approach to reducing the computational effort in equations (10) and (11) or (13) to (15) we limit the search for 2-cycles in such a way as to constrain the number of unpaired bases to be less than some fixed number. This is also a biologically reasonable constraint since 2-cycles are seldom very large while k-cycles for $k > 2$ are often extremely large. One exception to this is that very large 2-cycles can occur in folding at high temperatures. The time required to search for 2-loops at each (i, j) is now bounded, independent of $j - i$, and contributes only a quadratic term to the whole algorithm. When this constraint is added to the linearity constraint with $K = 2$, the dominant term becomes the search for multiple loops which takes cubic time. This bounded search technique is used in the program by Zuker and Stiegler (1981). If mutliloops are assigned experimentally determined energies which do not vary linearly with the size of the loop, then total rigour, combined with an N^3 algorithm, is not possible. Since no experimental data on the destabilizing effect of multiloops exist, it is difficult to say whether it is better to use a linear constraint with a rigorous algorithm or more plausible energies together with a slightly sub-optimal algorithm.

9. Dynamic Programming and Computer Implementation. Suppose the values of F and C are arranged in a square array, where the (i, j) cell is in row i and column j and contains both $F(i, j)$ and $C(i, j)$ for $i \leqslant j$. This fills the upper triangular half of the array. When the algorithm is implemented on a computer, two important decisions must be made. The first decision is how to store the $F(i, j)$ and $C(i, j)$ numbers in the computer. Computer memory is linear, and for large problems, not all of the numbers in F and C can reside in the central processing unit (CPU) simultaneously. Since secondary structure algorithms calculate array values largely in terms of values in

neighbouring rows and columns, it makes good sense to store the arrays in such a way that neighbouring array elements are close together in the linear order in so far as is possible, so that large jumps in the computer memory are minimized. The second decision is how to fill the array. The fill order can be arbitrary as long as one condition is met. When $F(i, j)$ and $C(i, j)$ are being computed, $F(i', j')$ and $C(i', j')$ must be known for all $i', j' \neq i, j$ such that $i \leqslant i' < j' \leqslant j$. At most nine different store and/or fill orders have been used. They can be described as in Table II.

TABLE II

1. Column row order	$(i, j) < (i', j') \leftrightarrow j < j'$ or $(j = j' \text{ and } i < i')$
2. Reverse column row order	$(i, j) < (i', j') \leftrightarrow j > j'$ or $(j = j' \text{ and } i < i')$
3. Column reverse row order	$(i, j) < (i', j') \leftrightarrow j < j'$ or $(j = j' \text{ and } i > i')$
4. Reverse column reverse row order	$(i, j) < (i', j') \leftrightarrow j > j'$ or $(j = j' \text{ and } i > i')$
5. Row column order	$(i, j) < (i', j') \leftrightarrow i < i'$ or $(i = i' \text{ and } j < j')$
6. Reverse row column order	$(i, j) < (i', j') \leftrightarrow i' < i$ or $(i = i' \text{ and } j < j')$
7. Row reverse column order	$(i, j) < (i', j') \leftrightarrow i < i'$ or $(i = i' \text{ and } j > j')$
8. Reverse row reverse column order	$(i, j) < (i', j') \leftrightarrow i > i'$ or $(i = i' \text{ and } j > j')$
9. Diagonal order	$(i, j) < (i', j') \leftrightarrow j - i < j' - i$ or $(j - i = j' - i'$ and $i < i')$

These nine orders are illustrated in Figure 9. Though values may be stored according to any of the orders, only 3, 6 and 9 can be used as fill orders. The most natural store order is 1 because this is how square arrays are normally stored by computer operating systems. The diagonal order 9 is the most natural fill order from a pedagogical point of view. One starts with short subsequences and builds up to larger and larger fragments. For purposes of memory access, however, it is quite disadvantageous. The definition of $F(i, j)$ requires that $F(i, h)$ and $F(h + 1, j)$ be accessed for all h between

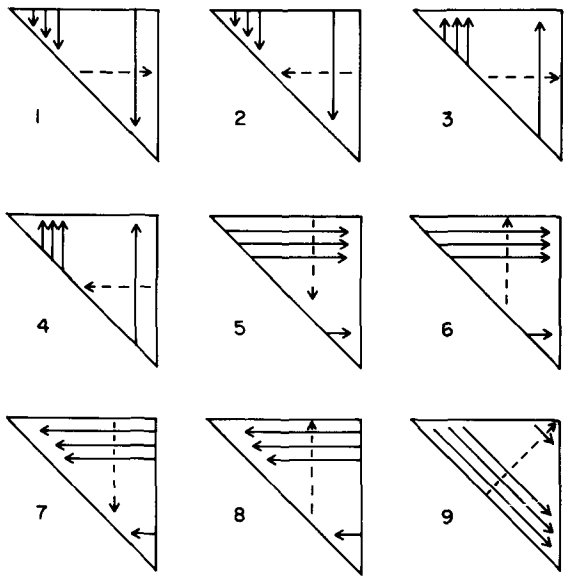

Figure 9. Nine different storage schemes for computer implementation of dynamic programming algorithms. The solid arrows indicate consecutive storage positions. The broken arrows indicate the progression from one solid arrow to the next.

i and j. With diagonal store, both the row and column search cause large jumps through the computer's memory. This leaves orders 1–8 as possibilities. It might seem desirable to use the same store and fill orders, but this is not the case. Column store (1–4) means that the current column is stored sequentially but that the row search requires large jumps. Row store (5–8) is similar. When column store is combined with row fill (or vice versa), the amount of jumping around in the memory to define $F(i, j)$ can be drastically reduced. With column store, the values $F(h + 1, j)$ are in consecutive positions in the computer's memory. The values $F(i, h)$ can be stored in a temporary array indexed only by h. This array, which is very small compared with the large triangular array, is overwritten when the algorithm proceeds to the next row. This method is described by Jacobson *et al.* (1984) where column store 1 and row fill 6 are used. This article also discusses how to store two or more numbers into one computer 'word' as well as a method for swapping large pieces of memory to and from the disk. The program described by Zuker and Stiegler (1981) uses column orders 1 and 3 for storing and filling C, respectively, and row order 5 and column order 3 for storing and filing F, respectively.

10. A Kinetic Approach. A folding algorithm which attempts to minimize

free energy need not use dynamic programming. One alternate approach is to simulate the folding as it might occur in the molecule, with one stem forming after another.

In a recent article, Martinez (1984) proposes such a kinetic algorithm. It can be summarized inductively as follows. To add the mth stem to a partially folded structure containing $m - 1$ stems, one compiles a list of all remaining stems which do not conflict with the existing structure. They are weighted according to their contribution to a decreased overall energy. The weighting actually involves temperature dependent equilibrium constants. Stem number m is chosen at random using the normalized weights as a probability measure. The Monte Carlo aspect of the algorithm can be tempered or eliminated altogether by deleting a designated percentage of stems with the highest equilibrium constants. The folding becomes completely deterministic when only the best stem may be chosen. Even when all stems are allowed to compete, lowering the temperature greatly accentuates differences between equilibrium constants, and the folding approaches a deterministic one as the temperature decreases to absolute zero.

This algorithm has been used with success on two transfer RNAs and on the *Tetrahymena thermophila* intervening sequence described by Cech *et al.* (1983). It is fast and efficient (execution time is proportional to N^2) and has the great advantage of yielding alternate solutions.

11. Phylogeny. The difficulty in inferring secondary structure is essentially that for a given molecule there are too many possible structures. Even when energy minimization is used as the selecting criterion, there are often many alternate structures close to the energy minimum.

Homologous RNAs from different organisms will tend to have roughly the same primary structure and very similar secondary structures. The fact that one sequence has an A and the other a G in a certain position will not change their abilities to take on the same secondary structure as long as the position in question is unpaired, or even if it is paired, as long as there is an appropriate change in the opposing base of each pair. This principle has been invaluable in reconstructing secondary structures. Structures which have been proposed on the basis of the sequence from a single organism have been refuted or confirmed on the basis of whether sequences determined later from other organisms are able to take on the same conformation, with the same base-paired positions.

The first step in phylogenetic analysis is usually to align a number of homologous sequences from different organisms. This usually involves positing a number of gaps in some of the sequences, with few gaps necessary if the homology is very close. The gaps represent base insertion or deletion mutations in some of the evolutionary lines. Aligning sequences to maximize

the number of identical aligned bases and to minimize the number of gaps can be done manually as in Figure 10, or with dynamic programming, using information about the phylogenetic relationship among the organisms.

Once the sequences are aligned, the next step is to identify pairs of complementary regions within each sequence which also occur in the same (aligned) positions within the other sequences. This has generally been carried out manually, which has the advantage of allowing adjustments to be made in the primary structure alignment on the basis of secondary structural evidence. Some specific examples are worth mentioning. The clover-leaf secondary structure for transfer RNA, determined through X-ray crystallography on a specific molecule (e.g. Kim *et al.*, 1974), has been shown to be applicable to all of the numerous transfer RNAs which have been sequenced to date. Another class of rather small RNAs (roughly 120 bases long), the so called 5*S* RNAs, have been extensively sequenced (e.g. Erdmann, 1982) although X-ray data are not available. Secondary structures for two classes of these RNAs were proposed as early as 1975 by Fox *et al.* on the basis of phylogeny. Secondary structures for the much larger 16*S* RNAs (roughly 1500 bases) have been proposed by several groups independently (Woese *et al.*, 1980; Glotz and Brimacombe, 1980; Stiegler *et al.*, 1981a, b) making extensive use of data from biochemical probes as well as phylogeny. A secondary structure model need not be complete. Davies *et al.* (1982) and Waring *et al.* (1983) propose a general, skeletal folding scheme for *introns* occurring in fungal mitochondria.

Studnicka *et al.* (1981) have proposed an automated method for finding common regions of base pairing in several sequences. This is applied to 17 5*S* RNA sequences. Another approach has been taken by Trifonov and Bolshoi (1983), who superimposed the matching matrices discussed above from 44 5*S* RNA sequences, and used the pattern recognition techniques of filtering to identify common base-paired regions. They were thus able to identify two alternate secondary structures for all molecules of this type.

Rather than align primary structures as a preliminary step, Sankoff *et al.* (1978) used the Pipas and McMahon (1975) program to calculate the best few secondary structures for 5*S* RNA molecules from several different species. They incorporated the comparative evidence by seeing which, if any, structures recurred among the best few for all the species.

It would be desirable to use phylogenetic and energetic considerations simultaneously in the search for the correct structure. The efforts of Studnicka *et al.* (1981) represent a first step in this direction. Sankoff (1984) has detailed a dynamic programming algorithm for simultaneously aligning and folding two or more sequences. At present, however, this approach is computationally very expensive, especially if more than two sequences are involved.

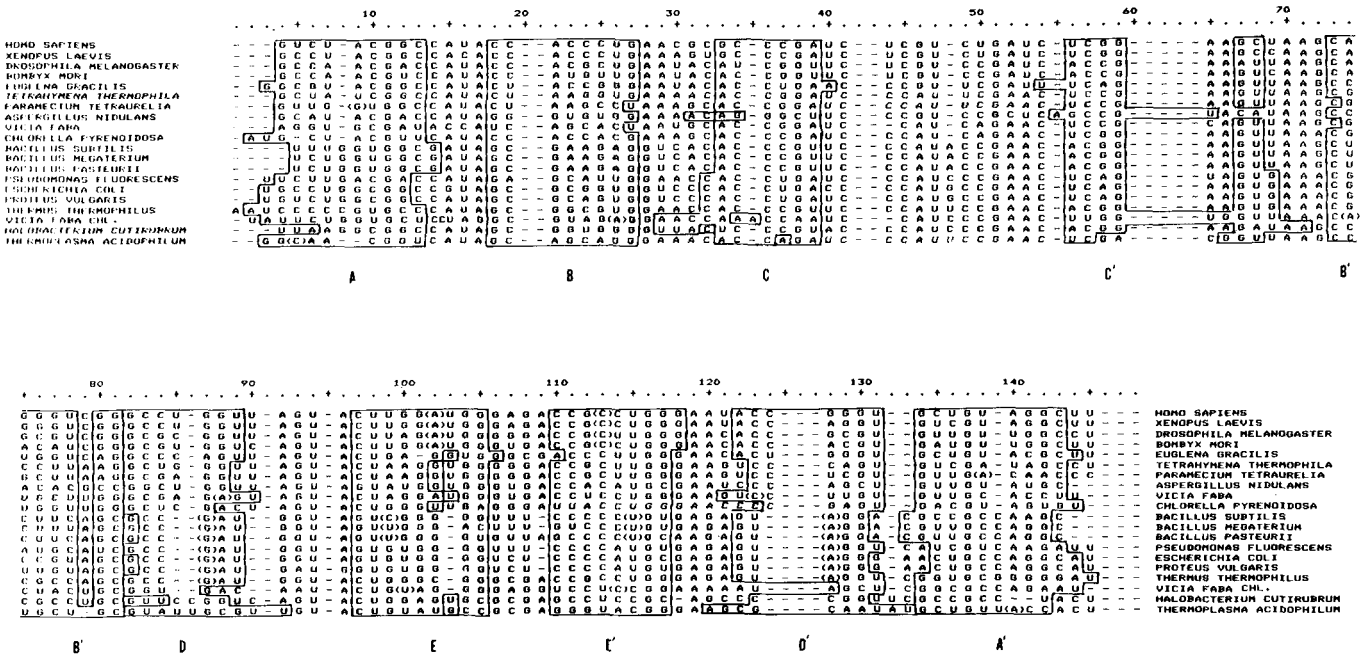

Figure 10. A manual alignment of 5*S* RNAs from twenty highly diverse organisms. Large outlined areas labelled *X*, *X'* represent complementary strands forming stacked pairs (helical regions or stems). Smaller boxes indicate bulges and interior loops within stems.

12. Circular RNA. Various *viroids*, or small viruses, consist of single stranded circular RNA. The study and prediction of circular RNA secondary structure is a straightforward extension of what has been discussed up to now. A circular RNA appearing often in the literature is the potato spindle tuber viroid (PSTV) which is commonly believed to have a rod-like secondary structure (McClements and Kaesberg, 1977; Domdey *et al.*, 1978; Gross *et al.*, 1978; Hadidi and Vournakis, 1978).

A circular RNA molecule consists of a chain of ribonucleotides linked together as in the usual definition. The difference is that the first and last nucleotides are linked together so that the chain is unbroken. Thus any consecutive numbering of the nucleotides begins at an arbitrary point, although the direction of numbering is uniquely defined. Given such a numbering, a secondary structure can be defined as earlier, except that the first and last bases cannot base pair with one another because they are now adjacent in the sequence. The main difference is that the decomposition defined earlier yields an extra substructure of a new kind if the set of base pairs is non-empty. Assuming that base pairs exist, the collection of k base pairs and u unpaired bases which are not accessible from any base pair constitute a new type of loop. This loop can be regarded as a k-cycle which includes its closing base pair(s). Thus, secondary structure prediction by energy minimization must take into account the effect of this extra loop.

Several authors suggest simply 'cutting' a circular RNA at an arbitrary point and considering the folding of the resulting linear RNA. Hofmann (Riesner *et al.*, 1983; Steger *et al.*, in preparation) has found that this method yields foldings which can be highly dependent on the cutting position, especially for folding at elevated temperatures. He makes the crucial observation that in a circular RNA, any base pair $i.j$ divides the folding into two foldings of linear RNA; a folding of the linear sequence from i to j inclusive and a folding of the linear sequence from j through the origin to i, inclusive. His dynamic programming algorithm modifies the one of Zuker and Stiegler (1981) by computing $C(j, i)$ and $F(j, i)$ for $i < j$ as minimum energies for the sequence from j through i as well as the usual $C(i, j)$ and $F(i, j)$. Then the quantity

$$\min_{1 \leqslant i < j \leqslant N} \{C(i, j) + C(j, i)\} \tag{19}$$

is the minimum folding energy of the circular RNA. This algorithm doubles computer time and storage requirements, with respect to the processing of a linear RNA of the same size.

The method of Zuker (unpublished) doubles the length of the sequence to $2N$, with nucleotides $N + 1, \ldots, 2N$ being the same as $1, \ldots, N$

respectively. The linear folding algorithm is used on the expanded sequence, with the condition that $C(i, j) = \infty$ if $j - i > N - 2$. The quantity

$$\min_{1 \leqslant i < j \leqslant N} \{C(i, j) + C(j, i + N)\} \tag{20}$$

is the minimum folding energy for the circular RNA. This algorithm quadruples computer storage requirements and roughly triples execution time, with respect to the processing of a linear RNA of the same size. However, in practice, the sequence is only extended by some fixed number of bases, say m (usually no more than 50). Then

$$\min_{\substack{1 \leqslant i \leqslant m \\ i < j \leqslant N}} \{C(i, j) + C(j, i + N)\} \tag{21}$$

will equal the quantity in (20) (and therefore the number in (19)) unless there are m consecutive unpaired bases and we are unlucky enough to have them occur starting at the first base. For large sequences, this 'pragmatic' algorithm barely adds to computation time for a linear RNA of the same size.

13. Conclusions. It must be remembered that the definition of secondary structure as simply a collection of hydrogen bonds is a crude approximation to the complicated and perhaps non-static reality of RNA folding. Any folding algorithm must therefore be regarded as an adjunct to and not a replacement of physical data. Zuker and Stiegler (1981) have advocated incorporating auxiliary information from chemical or enzymatic probes into the folding algorithm, and Sankoff *et al.* (1983) have suggested algorithmic procedures for doing so. The idea here is that if some bases are thought to pair or not to pair from the data, the folding algorithm should be adapted to predict an optimal structure subject to these constraints. This can be achieved easily with energy minimizing algorithms by altering the energy function to give large bonuses or penalties to k-cycles closed by desired or prohibited base pairs, respectively. A chemical technique known as cross-linking can sometimes detect base pairing regions between distant regions in large RNAs (e.g. Wollenzien *et al.*, 1979). Such data can be very useful, because if nucleotides $N1$ and $N2$ base pair in a molecule of size $N(1 < N1 < N2 < N)$ then the folding problem decomposes into two separate problems; the folding of the fragment from $N1$ to $N2$ and the entire sequence with the $N1$ to $N2$ fragment excised. This can sometimes substantially reduce computing time.

The general prohibition against knots is artificial. Davies *et al.* (1982), Hancock and Wagner (1982), Trifonov and Bolshoi (1983) and Weidner *et al.* (1977) advocate various models containing knots. As yet, however, there are

no stereochemical rules to guide the theorist who would like to predict structures with knots.

All folding algorithms discussed here have their merits and disadvantages. The matrix approach which recognizes patterns is both quick and inexpensive (Tinoco *et al.*, 1971; Trifonov and Bolshoi, 1983; Quigley *et al.*, 1984). Pipas and McMahon's (1975) and Studnicka's (1978) methods give multiple solutions, a valuable asset considering that secondary structure is not necessarily unique (e.g. Weidner *et al.*, 1977; Trifonov and Bolshoi, 1983). In contrast to the above methods, dynamic programming algorithms can deal with very large sequences, and in a reasonable time. The algorithm of Zuker and Stiegler (1981) has folded 2100 bases in under 42 min on a CRAY-XMP computer (Michael Ess, personal communication). By their nature, they also predict foldings for every subsequence of a folded sequence. This can be of value to those wishing to simulate the sequential folding of an RNA as it is being created (e.g. Boyle *et al.*, 1980). Although dynamic programming algorithms traditionally yield unique solutions, existing algorithms may in the future be extended to predict a variety of foldings based on the ideas of Waterman (1983) and Byers and Waterman (1984). The major practical problem here is how to choose a relatively small 'interesting' set of near optimal solutions from a potentially enormous collection. Manual phylogenetic methods lack the mathematical elegance of dynamic programming. They are also labour intensive. Nevertheless, such methods have produced dramatic results for a number of classes of small and large RNAs, and algorithms for simultaneous homological alignment and optimal folding seem a feasible direction for further advances.

The authors wish to thank B. Shapiro and J. Maizel for their plotting programs used to prepare Figures 1, 3 and 4. Similar thanks are extended to G. Osterburg (Figure 6). The data used to construct Figure 10 were supplied by R. De Wachter. The references were organized by J. M. Ridgeway.

LITERATURE

Auron, P. E., W. P. Rindone, C. P. H. Vary, J. J. Celentano and J. N. Vournakis. 1982. "Computer-Aided Prediction of RNA Secondary Structures." *Nucl. Acids Res.* **10**, 403–419.

Borer, P. N., B. Dengler, I. Tinoco, Jr. and O. C. Uhlenbeck. 1974. "Stability of Ribonucleic Acid Double-Stranded Helices." *J. molec. Biol.* **86**, 843–853.

Boyle, J., G. T. Robillard and S.-H. Kim. 1980. "Sequential Folding of Transfer RNA. A Nuclear Magnetic Resonance Study of Successively Longer tRNA Fragments with a Common 5′ End." *J. molec. Biol.* **139**, 601–625.

Byers, T. H. and M. S. Waterman. 1984. "Determining All Optimal and Near-Optimal Solutions when Solving Shortest Path Problems by Dynamic Programming." *Operat. Res.* (in press).

Cech, T. R., N. K. Tanner, I. Tinoco, Jr., B. R. Weir, M. Zuker and P. S. Perlman. 1983. "Secondary Structure of the Tetrahymena Ribosomal RNA Intervening Sequence: Structural Homology with Fungal Mitochondrial Intervening Sequences." *Proc. natn Acad. Sci. U.S.A.* **80**, 3903–3907.

Comay, E., R. Nussinov and O. Comay. 1984. "An Accelerated Algorithm for Calculating the Secondary Structure of Single-stranded RNAs." *Nucl. Acids Res.* **12**, 53–66.

Davies, R. W., R. B. Waring, J. A. Ray, T. A. Brown and C. Scazzocchio. 1982. "Making Ends Meet: A Model for RNA Splicing in Fungal Mitochondria." *Nature, Lond.* **300**, 719–724.

Domdey, H., P. Jank, H. L. Sänger and H. J. Gross. 1978. "Studies on the Primary and Secondary Structure of Potato Spindle Tuber Viroid: Products of Digestion with Ribonuclease A and Ribonuclease T1, and Modification with Bisulfite." *Nucl. Acids Res.* **5**, 1221–1236.

Erdmann, V. A. 1982. "Collection of Published 5*S* and 5.8*S* RNA Sequences and Their Precursors." *Nucl. Acids Res.* **10**, R93–R115.

Fink, T. R. and D. M. Crothers. 1972. "Free Energy of Imperfect Nucleic Acid Helices. I. The Bulge Defect." *J. molec. Biol.* **66**, 1–12.

Fox, G. E. and C. R. Woese. 1975. "5*S* RNA Secondary Structure." *Nature, Lond.* **256**, 505–507.

Fresco, J. R., B. M. Alberts and P. Doty. 1960. "Some Molecular Details of the Secondary Structure of Ribonucleic Acid." *Nature, Lond.* **188**, 98–101.

Glotz, C. and R. Brimacombe. 1980. "An Experimentally-Derived Model for the Secondary Structure of the 16*S* Ribosomal RNA from *Escherichia coli*." *Nucl. Acids Res.* **8**, 2377–2395.

Gralla, J. and D. M. Crothers. 1973(a). "Free Energy of Imperfect Nucleic Acid Helices. II. Small Hairpin Loops." *J. molec. Biol.* **73**, 497–511.

—— and ——. (1973(b). "Free Energy of Imperfect Nucleic Acid Helices. III. Small Internal Loops Resulting from Mismatches." *J. molec. Biol.* **78**, 301–319.

Gross, H. J., H. Domdey, C. Lossow, P. Jank, M. Raba and H. Alberty. 1978. "Nucleotide Sequence and Secondary Structure of Potato Spindle Tuber Viroid." *Nature, Lond.* **273**, 203–208.

Hadidi, A. and J. N. Vournakis. 1978. "Secondary Structure in Potato Spindle Tuber Viroid." *J. Supramol. Struct.* **7** (Suppl. 2), 280.

Hancock, J. and R. Wagner. 1982. "A Structural Model of 5*S* RNA from *E. Coli* based on Intramolecular Crosslinking Evidence." *Nucl. Acids Res.* **10**, 1257–1269.

Jacobson, A. B., L. Good, J. Simonetti and M. Zuker. 1984. "Some Simple Computational Methods to Improve the Folding of Large RNAs." *Nucl. Acids Res.* **12**, 45–52.

Kim, S. H., F. L. Suddath, G. J. Quigley, A. McPherson, J. L. Sussman, A. H. J. Wang, N. C. Seeman and A. Rich. 1974. "Three-Dimensional Tertiary Structure of Yeast Phenylalanine Transfer RNA." *Science* **185**, 435–440.

Lapalme, G., R. J. Cedergren and D. Sankoff. 1982. "An Algorithm for the Display of Nucleic Acid Secondary Structure." *Nucl. Acids Res.* **10**, 8351–8356.

Mainville, S. 1981. "Comparaisons et Auto-comparaisons de Chaînes Finies." Ph.D. thesis, Université de Montréal, Canada;

Martinez, H. M. 1984. "An RNA Folding Rule." *Nucl. Acids Res.* **12**, 323–334.

McClements, W. L. and P. Kaesberg. 1977. "Size and Secondary Structure of Potato Spindle Tuber Viroid." *Virology* **76**, 477–484.

Needleman, S. B. and C. D. Wunsch. 1970. "A General Method Applicable to the Search for Similarities in the Amino-Acid Sequence of Two Proteins." *J. molec. Biol.* **48**, 443–453.

Ninio, J. 1971. "Properties of Nucleic Acid Representations I. Topology." *Biochimie* **53**, 485–494.

——. 1979. "Prediction of Pairing Schemes in RNA Molecules—Loop Contributions and Energy of Wobble and Non-wobble Pairs." *Biochimie* **61**, 1133–1150.

Nussinov, R. 1977. "Secondary Structure Analysis of Nucleic Acids." Diss. Abstr. Int.

B Sci. Eng., Univ. Microfilms Int., Ann Arbor, Mich., Order No. 7805110.
—— and A. B. Jacobson. 1980. "Fast Algorithm for Predicting the Secondary Structure of Single-stranded RNA." *Proc. natn. Acad. Sci. U.S.A.* **77**, 6309–6313.
——, G. Pieczenik, J. R. Griggs and D. J. Kleitman. 1978. "Algorithms for Loop Matchings." *SIAM J. appl. Math.* **35**, 68–82.
Osterburg, G. and R. Sommer. 1981. "Computer Support of DNA Sequence Analysis." *Comput. Programs Biomed.* **13**, 101–109.
Papanicolaou, C., M. Gouy and J. Ninio. 1984. "An Energy Model that Predicts the Correct Folding of Both the tRNA and the 5*S* RNA Molecules." *Nucl. Acids Res.* **12**, 31–44.
Pipas, J. M. and J. E. McMahon. 1975. "Method for Predicting RNA Secondary Structure." *Proc. natn. Acad. Sci. U.S.A.* **72**, 2017–2021.
Quigley, G. J., L. Gehrke, D. A. Roth and P. E. Auron. 1984. "Computer-Aided Nucleic Acid Secondary Structure Modeling Incorporating Enzymatic Digestion Data." *Nucl. Acids Res.* **12**, 347–366.
Riesner, D., M. Colpan, T. C. Goodman, L. Nagel, J. Schumacher, G. Steger and H. Hofmann. 1983. "Dynamics and Interactions of Viroids." *J. Biomol. Structure and Dynamics* **1**, 669–688.
Salser, W. 1977. "Globin Messenger-RNA Sequences—Analysis of Base-Pairing and Evolutionary Implications." *Cold Spring Harbor Symp. Quant. Biol.* **42**, 985–1002.
Sankoff, D. 1972. "Matching Sequences Under Deletion-Insertion Constraints." *Proc. natn. Acad. Sci. U.S.A.* **69**, 4–6.
——. 1984. "Simultaneous Solution of the RNA Folding, Alignment and Protosequence Problems." Technical Report No. 1217, Université de Montréal, Canada.
—— and P. H. Sellers. 1973. "Shortcuts, Diversions, and Maximal Chains in Partially Ordered Sets." *Discrete Math.* **4**, 287–293.
——, A.-M. Morin and R. J. Cedergren. 1978. "The Evolution of 5*S* RNA Secondary Structures." *Can. J. Biochem.* **56**, 440–443.
——, J. B. Kruskal, S. Mainville and R. J. Cedergren. 1983. "Fast Algorithms to Determine RNA Secondary Structures Containing Multiple Loops." In *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Eds D. Sankoff and J. B. Kruskal, pp. 93–120. Reading, Massachusetts: Addison-Wesley.
Shapiro, B. A., L. E. Lipkin and J. Maizel. 1982. "An Interactive Technique for the Display of Nucleic Acid Secondary Structure." *Nucl. Acids Res.* **10**, 7041–7052.
——, J. Maizel, L. E. Lipkin, K. Currey and C. Whitney. 1984. "Generating Non-overlapping Displays of Nucleic Acid Secondary Structure." *Nucl. Acids Res.* **12**, 75–88.
Sloane, N. J. A. 1973. *A Handbook of Integer Sequences.* Academic Press.
Steger, G., H. Hofmann, B. Förtsch, H. J. Gross, J. W. Randles, H. L. Sänger and D. Riesner. "Conformational Transitions in Viroids and Virusoids: Comparison of results from energy minimization algorithm and from experimental data." *Biopolymers.* (In preparation.)
Stein, P. R. and M. S. Waterman, 1978. "On Some New Sequences Generalizing the Catalan and Motzkin Numbers." *Discrete Math.* **26**, 261–272.
Stiegler, P., P. Carbon, J.-P. Ebel and C. Ehresmann. 1981(a). "A General Secondary Structure Model for Procaryotic and Eucaryotic RNAs of the Small Ribosomal Subunits." *Eur. J. Biochem.* **120**, 487–495.
——, ——, M. Zuker, J.-P. Ebel and C. Ehresmann. 1981(b). "Structural Organization of the 16*S* Ribosomal RNA from *E. coli*. Topography and Secondary Structure." *Nucl. Acids Res.* **9**, 2153–2172.
Studnicka, G. M., F. A. Eiserling and J. A. Lake. 1981. "A Unique Secondary Folding Pattern for 5*S* RNA Corresponds to the Lowest Energy Homologous Secondary Structure in 17 Different Prokaryotes." *Nucl. Acids Res.* **9**, 1885–1904.
——, G. M. Rahn, I. W. Cummings and W. A. Salser. 1978. "Computer Method for

Predicting the Secondary Structure of Single-stranded RNA." *Nucl. Acids Res.* **5**, 3365–3387.

Tinoco, I., Jr., O. C. Uhlenbeck and M. D. Levine. 1971. "Estimation of Secondary Structure in Ribonucleic Acids." *Nature, Lond.* **230**, 362–367.

——, P. N. Borer, B. Dengler, M. D. Levine, O. C. Uhlenbeck, D. M. Crothers and J. Gralla. 1973. "Improved Estimation of Secondary Structure in Ribonucleic Acids." *Nature New Biol.* **246**, 40–41.

Trifonov, E. N. and G. Bolshoi. 1983. "Open and Closed 5*S* Ribosomal RNA, the Only Two Universal Structures Encoded in the Nucleotide Sequences." *J. molec. Biol.* **169**, 1–13.

Uhlenbeck, O. C., P. N. Borer, B. Dengler and I. Tinoco. 1973. "Stability of RNA Hairpin Loops: $A_6 - C_m - U_6$." *J. molec. Biol.* **73**, 483–496.

Waring, R. B., C. Scazzocchio, T. A. Brown and R. W. Davies. 1983. "Close Relationship Between Certain Nuclear and Mitochondrial Introns." *J. molec. Biol.* **167**, 595–605.

Waterman, M. S. 1978. "Secondary Structure of Single-stranded Nucleic Acids." In *Studies in Foundations and Combinatorics, Advances in Mathematics Suppl. Studies.* Vol. 1, pp. 167–212. Academic Press.

——. 1983. "Sequence Alignments in the Neighborhood of the Optimum with General Application to Dynamic Programming." *Proc. natn. Acad. Sci. U.S.A.* **80**, 3123–3124.

—— and T. F. Smith. 1978. "RNA Secondary Structure: A Complete Mathematical Analysis." *Math. Biosci.* **42**, 257–266.

Weidner, H., R. Yuan and D. M. Crothers. 1977. "Does 5*S* RNA Function by a Switch Between Two Secondary Structures?" *Nature, Lond.* **266**, 193–194.

Woese, C. R. L. J. Magrum, R. Gupta, R. B. Siegel, D. A. Stahl, J. Kop, N. Crawford, J. Brosius, R. Gutell, J. J. Hogan and H. F. Noller. 1980. "Secondary Structure Model for Bacterial 16*S* Ribosomal RNA: Phylogenetic, Enzymatic and Chemical Evidence." *Nucl. Acids Res.* **8**, 2275–2293.

Wollenzien, P., J. E. Hearst, P. Thammana and C. R. Cantor. 1979. "Base-pairing Between Distant Regions of the *Escherichia coli* 16*S* Ribosomal RNA in Solution." *J. molec. Biol.* **135**, 255–269.

Zuker, M. and P. Stiegler. 1981. "Optimal Computer Folding of Large RNA Sequences using Thermodynamics and Auxiliary Information." *Nucl. Acids Res.* **9**, 133–148.