ADVANCES IN APPLIED MATHEMATICS 7, 455–464

Rapid Dynamic Programming Algorithms for RNA Secondary Structure

MICHAEL S. WATERMAN*

Departments of Mathematics and of Biological Sciences, University of Southern California, Los Angeles, California 90089-1113

AND

TEMPLE F. SMITH

Department of Biostatistics, Harvard University, 44 Binney Street, Boston, Massachusetts 02115

Prediction of RNA secondary structure from the linear RNA sequence is an important mathematical problem in molecular biology. Dynamic programming methods are currently the most useful computer technique but are frequently very expensive in running time. In this paper new dynamic programming algorithms are presented which reduce the required computation. The first polynomial time algorithm is given for predicting general secondary structure. © 1986 Academic Press, Inc.

1. INTRODUCTION

In biology, structure and function are closely related. The shape of macromolecules (RNA, DNA, and proteins) and of complexes of macromolecules determines the interactions allowed and hence the processes of life. Some associated mathematical problems are easy to describe: given a linear sequence of a macromolecule, find its minimum free energy configuration. For proteins, this important problem has been much studied and little definitive progress made. See Havel and Wuthrich (1984) [2] for a nice description of the problem. For nucleic acids, the major problem receiving attention is that of predicting secondary structure of single-stranded RNA. The 3-dimensional structure of double-stranded DNA is also of interest but is not discussed here. The single-stranded RNA is viewed as a linear sequence $a_1a_2 \cdots a_n$ of ribonucleotides. Each a_i is identified with one of

*This author's work supported by System Development Foundation.

0196-8858/86 $7.50

four bases or nucleotides: A (adenine), C (cytosine), G (guanine), and U (uracil). These bases can form base pairs, conventionally A pairs with U and C pairs with G. These are called Watson–Crick pairs. In addition, the pairing of G and U is frequently allowed.

Secondary structure is a planar graph which satisfies: If a_i pairs with a_j and a_k is paired with a_l with $i < k < j$, then $i < l < j$ also. The graph theory of such structures was first discussed in Waterman [8]. Combinatorial aspects of such graphs were introduced in Waterman [8] and continued in Stein and Waterman [7]. These topics are also treated in a review by Zuker and Sankoff [11].

Dynamic programming methods to predict secondary structure were first presented by Waterman [8], Waterman and Smith [10], and by Nussinov *et al.* [5]. Zuker and Sankoff [11] provide an excellent review. Recently Sankoff [12] considers simultaneous alignment and secondary structure prediction. Dynamic programming is still the method of choice for secondary structure prediction although computation time is a limiting factor. In the present paper, some new efficiencies are presented which reduce the theoretical and practical computational complexity of the dynamic programming algorithms.

2. Basic Algorithms for Hairpins

In this section basic dynamic programming procedures for secondary structure are presented. Throughout h_{ij} is the minimum free energy (single hairpin loop) secondary structure on $a_i a_{i+1} \cdots a_j$ $(i < j)$ where a_i and a_j form a base pair and there is a single end loop. See Fig. 1. If a_i and a_j cannot base pair, $h_{ij} = +\infty$. The free energy functions are assumed to be of the form

$\alpha(a, b) =$ free energy of an a-b base pair,

$\xi(k) =$ destabilization free energy of an end loop of k bases,

$\eta =$ stacking energy of adjacent base pairs,

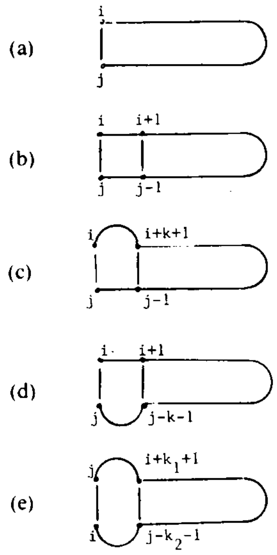
$\beta(k) =$ destabilization free energy of bulge of k bases,

and

$\gamma(k) =$ destabilization free energy of an interior loop of k bases.

It is possible to define more general energy functions. For example, ξ can depend on the adjacent base pair(s). The same complexity results derived below hold.

There are exactly five ways to build h_{ij} from the base pair and these are presented in Fig. 1, along with formulas which calculate their values. Arguments to justify the equations are well known and appear, among other places, for general energy functions in earlier papers of ours



(a) $\alpha(a_i, a_j) + \xi(j - i - 1)$

(b) $\alpha(a_i, a_j) + \eta + h_{i+1,j-1}$

(c) $\min\limits_{k \geq 1} \{\alpha(a_i, a_j) + \beta(k) + h_{i+k+1,j-1}\}$

(d) $\min\limits_{k \geq 1} \{\alpha(a_i, a_j) + \beta(k) + h_{i+1,j-k-1}\}$

(e) $\min\limits_{k_1, k_2 \geq 1} \{\alpha(a_i, a_j) + \gamma(k_1 + k_2) + h_{i+1+k_1,j-1-k_2}\}$

FIG. 1. Illustration of each of the five basic minimization situations with the corresponding formula: (a) end loop, (b) extension of helical region, (c) bulge at "i," (d) bulge at "j," (e) interior loop.

(Waterman [8], Waterman and Smith [10]) which presented an iterative dynamic programming approach. Later work such as Kruskal and Sankoff [4] and Zuker and Sankoff [11] present and survey equivalent formulas for a single pass method. For hairpins, there is essentially no difference between these methods.

To estimate the computational complexity, each step of Fig. 1 is treated. Only powers of n are, in the end, of interest for the rate of growth of computation.

For purposes of estimating computational complexity, the various formulas take time proportional to

$$\text{(a); (b)} \qquad \sum_{1 \leq i \leq j \leq n} 1 = \sum_{i=1}^{n} \sum_{j=i}^{n} 1 = O(n^2),$$

$$\text{(c); (d)} \qquad \sum_{1 \leq i \leq j \leq n} (j - i) = \sum_{i=1}^{n} \sum_{j=i}^{n} (j - i)$$

$$= \sum_{i=1}^{n} \frac{(n-i)(n-i+1)}{2} = O(n^3).$$

$$\text{(e)} \qquad \sum_{1 \leq i \leq j \leq n} \left(\sum_{i \leq i' \leq j' \leq j} 1 \right) = \sum_{1 \leq i \leq j \leq n} C \cdot (j - i)^2 = O(n^4).$$

It is clear, then, that previous algorithms for best hairpins are basically of order n^4. The purpose of the next section is to reduce these algorithms to order n^3 in general. Section 4 gives a further reduction to order n^2 for linear or concave destabilization functions.

3. Reduction of Computation Time for Hairpins

In this section some details of the computation will be discussed. In the course of this, it will become clear that a major reduction in computation can be achieved. The dynamic programming calculations are stored in a matrix and, in fact, are sometimes referred to as "matrix methods." Organizing these calculations in a way which allows visualization of the associated RNA structures is a useful device.

As above h_{ij} is the free energy of the minimum free energy hairpin on $a_i a_{i+1} \cdots a_j$ $(i < j)$ satisfying a_i and a_j forming a base pair. If a_i and a_j cannot base pair, set $h_{ij} = +\infty$. Organization of the matrix (h_{ij}), $i = 1, 2, \ldots, n$ and $j = n, n-1, \ldots$ $(i \le j)$, with the base sequence written in reverse order along the columns is illustrated as

$$\begin{array}{c|ccccc}
 & a_n & a_{n-1} & \cdots & a_2 & a_1 \\
\hline
a_1 & h_1, n & h_1, n-1 & & h_{1,2} & h_{11} \\
a_2 & h_2, n & h_2, n-1 & & h_{2,2} & \\
\vdots & & & \cdot^{\cdot^{\cdot}} & & \\
a_{n-1} & h_{n-1,n} & h_{n-1,n-1} & & & \\
a_n & h_{n,n} & & & &
\end{array}$$

Next take h_{ij} where $h_{ij} < +\infty$, i.e., where a_i and a_j can form a base pair and where $j - i - 1 \ge m$ (the minimum end loop size). Of course h_{ij} results from one of the situations discussed in Section 2, which will now be examined in greater detail.

If the base pair is at the "bottom" of an end loop, then h_{ij} equals

$$\alpha(a_i, a_j) + \xi(j - i - 1).$$

This is the only step in the minimization not indicated on the matrix in Fig. 2.

If the base pair is part of a helical region, then h_{ij} equals

$$\alpha(a_i, a_j) + h_{i+1, j-1} + \eta,$$

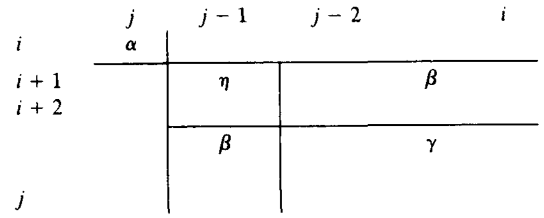


FIG. 2. Schematic representation of regions for minimization in the matrix.

which is indicated in the $(i+1, j-1)$ position of the matrix in Fig. 2 by the letter η.

If h_{ij} results from a bulge, then h_{ij} equals

$$\alpha(a_i, a_j) + \beta(k) + h_{i+k+1, j-1}$$

or

$$\alpha(a_i, a_j) + \beta(k) + h_{i+1, j-k-1},$$

where $1 \le k \le j - i - 2 - m$. In the first situation, a minimization is performed down the $(j-1)$th column. This vertical region is indicated by the symbol β as is the horizontal region in the $(i+1)$th row.

The remaining possibility is interior loops, where h_{ij} equals

$$\alpha(a_i, a_j) + \gamma(k_1 + k_2) + h_{i+1+k_1, j-1-k_2},$$

where $j - i - 3 - m \ge k_1 + k_2 \ge 2$ ($k_1 \ge 1$ and $k_2 \ge 1$). This region is indicated in Fig. 2 by γ.

As shown in Section 2, the computational complexity of helix and end loop formation is $O(n^2)$, of bulge formation is $O(n^3)$, and of interior loop formation is $O(n^4)$. The remainder of this section will show that interior loop calculations can in general be reduced to $O(n^3)$.

For calculating interior loops from an $i-j$ base pair, the possible candidate positions are (k, l) where $h_{kl} < \infty$ and (k, l) belongs to,

$$\Gamma(i, j) = \{(k, l)\colon l - k - 1 \ge m, k \ge i + 2, j - 2 \ge l\}.$$

The size of the interior loop is

$$s = (j - i - 1) - (l - k + 1) = (j - i) - (l - k) - 2.$$

This equation implies that, along lines with $l - k =$ constant, the interior loop destabilization function $\gamma(s)$ is constant. The computation is now organized to exploit this observation. Next we show that $\Gamma(i-1, j)$ is equal to $\Gamma(i, j)$ plus a horizontal line segment. (See Fig. 2.)

For $l = m, m+1, \ldots, n-2$; $h_{i,i+l+1}$ is calculated for $1 \leq i \leq n - l - 1$. This begins calculation on the line $j - i - 1 = m$ and proceeds line by line until $j - i - 1 = n - 2$ ($j = n$ and $i = 1$). For each of these lines after the first there is a position $(i-1, j)$ directly above a position (i, j) on the line below. The interior loop calculation for $(i-1, j)$ involves

$$\begin{aligned}\Gamma(i-1, j) &= \{(k, l): l - k - 1 \geq m, k \geq i + 1, j - 2 \geq l\} \\ &= \Gamma(i, j) \cup \{(i+1, l): m + i + 2 \leq l \leq j - 2\}.\end{aligned}$$

That is, $\Gamma(i-1, j)$ is the union of $\Gamma(i, j)$ and a horizontal line segment.

As noted above, all $(k, l) \in \Gamma(i, j)$ with $l - k$ constant have the same interior loop destabilization, $\gamma(j - i - l + k - 2)$. For the line $j - i = c$, define a matrix G^c where $G^c_{s,j}$ is the minimum $h_{k,l}$ with $l - k$ constant in $\Gamma(i, j)$ with s bases in the interior loop. Formally

$$G^c_{s,j} = \min\{h_{kl}: (k, l) \in \Gamma(j - c, j) \text{ and } s = c - l + k - 2\}.$$

For $j - i = c$, $2 \leq s \leq j - i - m - 3$.

To update $G^{c+1}_{2,j}$ for the line $j - i = c + 1$,

$$G^{c+1}_{s,j} = h_{i+1,j-2},$$

$$G^{c+1}_{s,j} = \min\{h_{i+1,j-s-1}; G^c_{s-1,j}\}.$$

This shows that a single matrix G can be used. To find the best interior loop configuration, compute

$$\min\{\gamma(s) + G_{s,j}: 2 \leq s \leq j - i - m - 3\},$$

which can be done in time proportional to $j - i$.

The above setup shows interior loops have an overall calculation time equivalent to that of bulges, $O(n^3)$. Moreover, this equivalence is established by showing that the interior loop problem can be given a data structure equivalent to that of bulges. The additional storage due to G is $n^2/2$ while computation time due to G is bounded by $O(n^3)$.

4. Linear and Concave Destabilization Functions

As mentioned above, significant efficiencies can be achieved for linear destabilization functions. Gotoh [1] provides a clear, complete proof that $O(n^3)$ sequence alignment algorithms can be reduced to $O(n^2)$ for linear deletion functions. For secondary structure algorithms, a similar assumption is used by Waterman [8, p. 203] to reduce computation. Also Kanehisa and Goad [3] prove that a reduction from $O(n^3)$ to $O(n^2)$ can be achieved for linear bulge functions. They state the same result for interior loop calculations but do not present any indication of proof or of an algorithm to accomplish the reduction. Here we give a proof for both bulges and interior loops and frame the proof in a manner which indicates how to perform the computations. In addition, we show as well that $O(n^2)$ computational complexity holds for concave destabilization functions, such as $\gamma(n) = a + b\log(n)$.

Define the best bulges "down" the column by

$$\mathrm{hdo}(i,j) = \min_{k\geq 1}\left\{\beta(k) + h_{i+k+1,j-1}\right\},$$

where $\beta(k) = a + b(k-1)$. Then

$$\begin{aligned}\mathrm{hdo}(i,j) &= \min\left\{a + h_{i+2,j-1}, \min_{k\geq 2}\left\{\beta(k) + h_{i+k+1,j-1}\right\}\right\}\\ &= \min\left\{a + h_{i+2,j-1}, \min_{l\geq 1}\left\{\beta(l+1) + h_{i+l+2,j-1}\right\}\right\}\\ &= \min\left\{a + h_{i+2,j-1}, \min_{l\geq 1}\left\{\beta(l) + h_{i+l+2,j-1}\right\} + b\right\}\\ &= \min\left\{a + h_{i+2,j-1}, \mathrm{hdo}(i+1,j) + b\right\}.\end{aligned}$$

Similarly, if the bulges "over" a row are considered,

$$\mathrm{hov}(i,j) = \min_{k\geq 1}\left\{\beta(k) + h_{i+1,j-k-1}\right\}$$

with $\beta(k) = a + b(k-1)$, we obtain

$$\mathrm{hov}(i,j) = \min\left\{a + h_{i+1,j-2}, \mathrm{hov}(i,j-1) + b\right\}.$$

This proof is modeled after Gotoh [1] and Waterman [9]. If the computation is carried out on lines of $j - i = c$ for $c = m, m+1, \ldots,$ it is easy to see, that a single vector of length n suffices for each of hdo(i,j) and hov(i,j).

Next, for interior loops and $\gamma(k) = c + d(k-2)$ define

$$hil(i,j) = \min_{k_1,k_2\geq 1}\left\{\gamma(k_1+k_2) + h_{i+1+k_1,j-1-k_2}\right\}.$$

Now

$$\begin{aligned}\min_{k_1>1,\, k_2\geq 1} & \left\{\gamma(k_1+k_2)+h_{i+1+k_1,\, j-1-k_2}\right\}\\ &= \min_{l\geq 1,\, k_2\geq 1}\left\{\gamma(1+l+k_2)+h_{i+2+l,\, j-1-k_2}\right\}\\ &= d+hil(i+1,\, j).\end{aligned}$$

For

$$\min_{k_1=1,\, k_2\geq 1}\left\{\gamma(1+k_2)+h_{i+2,\, j-1-k_2}\right\}$$

the problem is exactly equivalent to the bulge problem handled above.

Thus if the energy functions $\beta(k)$ and $\gamma(k)$ are linear, computation of best hairpin can be accomplished in $O(n^2)$ time and space. Other work, such as Sankoff *et al.* [6] achieves $O(n^3)$ for $\beta(k)=\gamma(k)$ linear. Of course that work includes higher order secondary structure as well.

Principles of thermodynamics suggest that destabilization functions should grow like $\log(k)$ instead of being linear. Fortunately recent work on sequence alignment algorithms by Waterman [9] provides an $O(n^2)$ algorithm for concave energy functions which includes $\log(k)$. A function w is concave if

$$w(p+q+l)-w(p+q)\leq w(q+l)-w(q)$$

for $p, q, l\geq 1$. If the inequality is equality, then w is linear. The idea is that the cost $\beta(k+1)-\beta(k)$ of an additional base in, a bulge, for example, should decrease with k.

The next consideration is that of computational efficiency for concave functions. Instead of bringing in the details of bulge and interior loop calculations, we simply consider the representative recursion

$$\mathrm{hdo}(i,\, j)=\min_{k\geq 1}\left\{\beta(k)+h_{i+k+1,\, j-1}\right\}.$$

Recall both bulge and interior loop calculations are of this nature.

Let l satisfy

$$\beta(l)+h_{i+l+1,\, j-1}\leq\beta(k)+h_{i+k+1,\, j-1},\qquad k\geq 1.$$

For $k\leq l$, $\beta(l+1)-\beta(l)\leq\beta(k+1)-\beta(k)$ and

$$\beta(l+1)+h_{(i-1)+(l+1)+1,\, j-1}\leq\beta(k+1)+h_{(i-1)+(k+1)+1,\, j-1}.$$

Therefore such k need not be considered when calculating $\mathrm{hdo}(i,\, j)$ and

$$\mathrm{hdo}(i-1,\, j)=\min\left\{\min_{k\geq l}\left\{\beta(k+1)+h_{i+k+1,\, j-1}\right\},\, \beta(1)+h_{i+1,\, j-1}\right\}.$$

Essentially the above equation states that the algorithm need only bulge to positions where length 1 bulges were optimal. We have not been able to estimate the growth of the number of such events, but the number seems to grow quite slowly.

5. Multibranch Loops

In this section we consider loops with more than one hairpin extending from them. The destabilization function $\gamma(\cdot)$ above assumes exactly one hairpin extends from the loop. Since so little is known about the energetic properties of multibranch loops, we assume that $\rho(\cdot)$ is a single destabilization function which holds for all multibranch loops.

Define $g(i, j)$ to be the minimum free energy multibranch loop structure on $a_i a_{i+1} \cdots a_j$, $g(i, j; k)$ to be the minimum free energy multibranch loop structure on $a_i a_{i+1} \cdots a_j$ and there are k unpaired bases in the multibranch loop, and $e(i, j)$ to be the minimum free energy structure on $a_i a_{i+1} \cdots a_j$ where a_i and a_j form a base pair.

The structure corresponding to $g(i, j + l; k)$ either has a_{j+1} base paired or not. If a_{j+1} is not base paired, then

$$g(i, j+1; k) = \rho(k) - \rho(k-1) + g(i, j; k-1).$$

If a_{j+1} is base paired, then

$$g(i, j+1; k) = \min_{j^*} \{ g(i, j^*; k) + e(j^* + 1, j + 1) \}.$$

Now

$$g(i, j) = \min_{k} g(i, j; k),$$

and $e(i, j)$ is obtained by minimizing over multibranch loops, end loops, interior loops, bulges and helix formation.

Storage for the multibranch loop algorithm described here is proportional to

$$\sum_{i=1}^{n} \sum_{j=i}^{n} (j - i) = O(n^3),$$

while time is proportional to

$$\sum_{i=1}^{n} \sum_{j=i}^{n} (j - i)^2 = O(n^4).$$

Increasing storage to n^3 is not desirable but previously known rigorous algorithms take time $O(n^{2L})$ where L = maximum number of arms helices from a multibranch loop. Even cloverleafs took time $O(n^6)$.

ACKNOWLEDGMENT

The authors are grateful to Michael Zuker for his useful comments about this work.

REFERENCES

1. O. GOTOH, An improved algorithm for matching biological sequences, *J. Mol. Biol.* **162** (1982), 705–708.
2. T. HAVEL AND K. WUTHRICH, A distance geometry program for determining the structures of small proteins and other macromolecules from nuclear magnetic resonance measurements of intramolecular ^{1}H-^{1}H proximities in solution, *Bull. Math. Biol.* **46** (1984), 673–698.
3. M. I. KANEHISA AND W. B. GOAD, Pattern recognition in nucleic acid sequences II. An efficient method for finding locally stable secondary structures, *Nucl. Acids Res.* **10** (1982), 265–277.
4. J. B. KRUSKAL AND D. SANKOFF, An anthology of algorithms and concepts for sequence comparison, *in* "Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison" (D. Sankoff and J. B. Kruskal, Eds.), pp. 265–310, Addison–Wesley, Reading, Mass. 1983.
5. R. NUSSINOV, G. PIECZENIK, J. R. GRIGGS, AND D. J. KLEITMAN, Algorithms for loop matchings, *SIAM J. Appl. Math.* **35** (1978), 68–82.
6. D. SANKOFF, J. B. KRUSKAL, S. MAINVILLE, AND R. J. CEDERGREN, Fast algorithms to determine RNA secondary structures containing multiple loops, *in* "Time Warps, String Edits, And Marcomolecules: The Theory and Practice of Sequence Comparison" (D. Sankoff and J. B. Kruskal, Eds.), pp. 93–120, Addison–Wesley, Reading, Mass., 1983.
7. P. R. STEIN AND M. S. WATERMAN, On some new sequences generalizing the Catalan and Motzkin numbers, *Discrete Math.* **26** (1978), 261–272.
8. M. S. WATERMAN, Secondary structure of single-stranded nucleic acids, *in* "Studies in Foundations and Combinatorics," Advan. in Math. Suppl. Studies, Vol. 1, pp. 167–212, Academic Press, New York, 1978.
9. M. S. WATERMAN, Efficient sequence alignment algorithms, *J. Theoret. Biol.* **108** (1984), 333–337.
10. M. S. WATERMAN AND T. F. SMITH, RNA secondary structure: A complete mathematical analysis, *Math. Biosci.* **42** (1978), 257–266.
11. M. ZUKER AND D. SANKOFF, RNA secondary structures and their prediction, *Bull. Math. Biol.* **46** (1984) 591–621.
12. D. SANKOFF, Simultaneous solution of the RNA folding, alignment and protosequence problems, *SIAM J. Appl. Math.* **45** (1985), 810–824.