BIOINFORMATICS

Vol. 15 no. 6 1999
Pages 440–445

Fast evaluation of internal loops in RNA secondary structure prediction

Rune B. Lyngsø[1,2], Michael Zuker[2] and Christian N. S. Pedersen[3,4]

[1]Department of Computer Science, University of Aarhus, DK 8000 Århus, Denmark, [2]Institute for Biomedical Computing, Washington University, St Louis, MO 63108, USA and [3]Basic Research in Computer Science, University of Aarhus, DK 8000 Århus, Denmark

Received on September 11, 1998; accepted on February 24, 1999

Abstract

Motivation: *Though not as abundant in known biological processes as proteins, RNA molecules serve as more than mere intermediaries between DNA and proteins. Research in the last 15 years demonstrates that RNA molecules serve in many roles, including catalysis. Furthermore, RNA secondary structure prediction based on free energy rules for stacking and loop formation remains one of the few major breakthroughs in the field of structure prediction, as minimum free energy structures and related quantities can be computed with full mathematical rigor. However, with the current energy parameters, the algorithms used hitherto suffer the disadvantage of either employing heuristics that risk (though highly unlikely) missing the optimal structure or becoming prohibitively time consuming for moderate to large sequences.*

Results: *We present a new method to evaluate internal loops utilizing currently used energy rules. This method reduces the time complexity of this part of the structure prediction from $O(n^4)$ to $O(n^3)$, thus reducing the overall complexity to $O(n^3)$. Even when the size of evaluated internal loops is bounded by k (a commonly used heuristic), the method presented has a competitive edge by reducing the time complexity of internal loop evaluation from $O(k^2n^2)$ to $O(kn^2)$. The method also applies to the calculation of the equilibrium partition function.*

Availability: *Source code for an RNA secondary structure prediction program implementing this method is available at ftp://www.ibc.wustl.edu/pub/zuker/zuker.tar.Z*

Contact: *rlyngsoe@daimi.au.dk*

Introduction

Structure prediction remains one of the most compelling, yet elusive, areas of computational biology. Not yielding to overwhelming numbers and resources, this area still poses a lot of interesting questions for future research. For RNA, if one restricts attention to the prediction of unknotted secondary structures, elegant dynamic programming algorithms combined with nearest-neighbor free energy parameters combine to give rigorous solutions to the problems of computing minimum free energy structures, close to optimal foldings, and partition functions that yield exact base pair probabilities. Secondary structure in RNA is the list of base pairs that occur in a three-dimensional RNA structure.

Information on the secondary structure of an RNA molecule can be used as a stepping stone to modeling the full structure of the molecule, which in turn relates to the biological function. As recent experiments have shown that RNA molecules can undertake a wide range of different functions (see Hopkin, 1998), the prediction of RNA secondary structure should continue to be important for biomolecule engineering.

A model was proposed in Tinoco *et al.* (1971, 1973) to calculate the stability (in terms of free energy) of a folded RNA molecule by adding independent contributions from base pair stacking and loop-destabilizing terms from the secondary structure. This model has proven to be a good approximation of the forces governing RNA structure formation, thus allowing fair predictions of real structures by determining the most stable structures in the model of a given sequence.

Based on this model, algorithms for computing the most stable structures have been proposed (e.g. Nussinov and Jacobson, 1980; Zuker and Stiegler, 1981). Zuker (1989) proposed a method to determine all base pairs that could participate in structures with a free energy within a specified range from the optimal. McCaskill (1990) demonstrated how a related dynamic programming algorithm could be used to calculate equilibrium partition functions, which lead to exact calculations of base pair probabilities in the model.

[4]Work done while visiting University of California, Davis, and DIMACS.

© Oxford University Press 1999

A major problem for these algorithms is the time required to evaluate possible internal loops. In general, this requires time $O(n^4)$, which is often circumvented by assuming that only 'small' loops need to be considered (e.g. McCaskill, 1990). This risks missing some optimal large internal loops, especially when folding at high temperatures, but the time required for evaluating internal loops is reduced to $O(n^2)$, thus reducing the overall complexity to $O(n^3)$. If the stability of an internal loop can be assumed only to depend on the size of the internal loop, Waterman and Smith (1986) describe how to reduce the time requirement to $O(n^3)$. [This method is also referred to by McGaskill (1990) where a combination of the above methods is proposed: a free energy function only dependent on loop size is used for large loops, while small loops are treated specially.] This is further improved to $O(n^2\log^2 n)$ for convex free energy functions in Eppstein *et al.* (1988). Affine free energy functions (i.e. of the form $a + bN$, where N is the size of the loop) allow for $O(n^2)$ computation time by borrowing a simple method used in sequence alignment by Gotoh (1982).

Unfortunately, the currently used free energy functions for internal loops are not convex, let alone affine. Furthermore, the technique described in Eppstein *et al.* (1988) hinges on the objective being to find a structure of maximum stability, and thus does not translate to the calculation of the partition function of McCaskill (1990) where a Boltzmann weighted sum of contributions to the partition function is calculated.

In this paper, we will describe a method based on a property of current free energy functions for internal loops that allows all internal loops to be evaluated in time $O(n^3)$. This method is applicable both to determining the most stable structure and to calculating the partition function. The rest of this paper is structured as follows. In Methods, we briefly discuss the problem of evaluating internal loops in RNA secondary structure prediction and introduce the notation we will be using. In Algorithms, we present a method yielding cubic time algorithms for evaluating internal loops for certain free energy functions. We argue that this method can be used with currently used free energy functions and describe how the same technique can be used to calculate the contributions to the partition function from structures with internal loops. In Implementation, we compare our method to the previously used method and in Discussion we discuss some future directions for improvements.

Methods

A secondary structure of a sequence s is a set S of base pairs $i \cdot j$ with $1 \le i < j \le |s|$ such that $\forall i \cdot j, i' \cdot j' \in S : i = i' \Leftrightarrow j = j'$. Thus, any base can take part in at most one base pair. We will only be concerned with structures that do not contain pseudoknots, i.e. for $i \cdot j, i' \cdot j' \in S : i < i' \Leftrightarrow j > j'$. We will say that $i \cdot j$ closes the substructure $S_{i,j}$, defined on $s_i \ldots s_j$ (i.e. the substructure $\{i' \cdot j' \in S \mid i \le i' < j' \le j\}$).

In this paper, we will focus on the evaluation of internal loops and bulges. Internal loops are substructures defined by two non-stacking base pairs (two base pairs, $i \cdot j$ and $i' \cdot j'$, are stacking if they are adjacent, i.e. if $i' = i + 1$ and $j' = j - 1$)1.3 $i \cdot j, i' \cdot j' \in S$ where all the bases between i and i' and j and j' are unpaired (or to put it formally, $i \cdot j$ and $i' \cdot j'$, $i < i' < j' < j$ and $j - j' + i' - i > 2$, defines an internal loop iff $\forall i < i'' < i', j' < j'' < j : \neg\exists k : i'' \cdot k \in S \vee k \cdot j'' \in S$). Bulges are a special kind of internal loop with no bases between either bases i and i' or bases j and j', i.e. either $i' = i + 1$ or $j' = j - 1$. If $i < i' < j' < j$, then $i' \cdot j'$ is called the interior base pair of the loop and $i \cdot j$ is called the exterior base pair, and is said to close the loop. We will use the same notation as in, for example, Turner *et al.* (1988) (and we refer to this article for a presentation of the full dynamic programming algorithm for RNA secondary structure prediction). In the rest of this section, we briefly discuss the matrices and energy functions needed in this paper.

Of the recursive equations defining the dynamic programming algorithm for RNA secondary structure [see Turner *et al.* (1988), pp. 180–181] the equation concerning the evaluation of internal loops is:

$$VBI(i,j) = \min_{\substack{i < i' < j' < j \\ i' - i + j - j' > 2}} \{eL(i,j,i',j') + V(i',j')\} \qquad (1)$$

Here $VBI(i, j)$ is the minimum energy of a substructure on $s_i \ldots s_j$ where $i \cdot j$ closes a bulge or internal loop and $V(i', j')$ is the minimum energy of a substructure on $s_{i'} \ldots s_{j'}$ closed by $i' \cdot j'$ (but with no restriction on what kind of substructure it closes—it might thus stack with another base pair or close a multi-branched loop as well as close an internal loop). The energy function $eL(i, j, i', j')$ specifies the destabilizing energy of the internal loop with interior base pair $i' \cdot j'$ and exterior base pair $i \cdot j$.

Equation (1) thus specifies that to find the optimal substructure on $s_i \ldots s_j$ where $i \cdot j$ closes an exterior loop, we will have to consider all possible interior base pairs and add the destabilizing energy of the loop to the energy of an optimal substructure on $s_{i'} \ldots s_{j'}$ closed by $i' \cdot j'$. With no further knowledge of eL, the evaluation of all possible internal loops would thus require time $O(n^4)$. In the next section, we will however describe how to use some properties of the energy functions currently used to reduce the complexity to $O(n^3)$.

Algorithms

A common simplifying assumption of internal loops is that the destabilizing energy can be split into contributions from:

- the size of the loop;

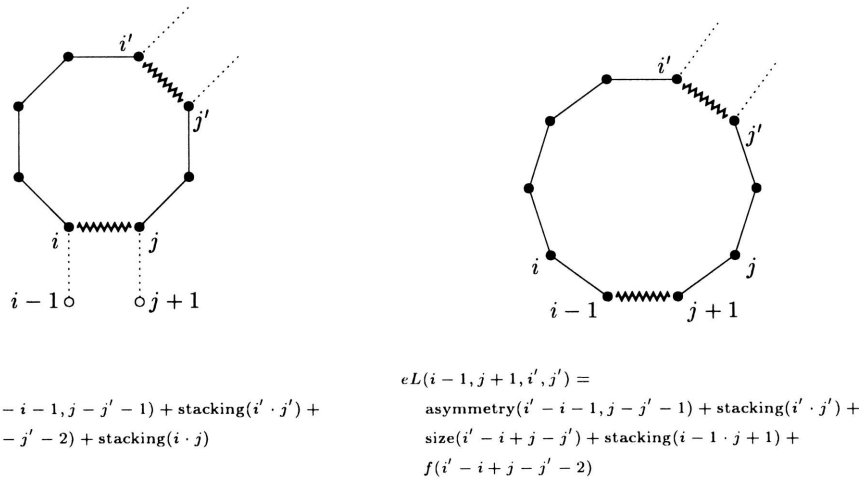

Fig. 1. The difference in destabilizing energy when extending a loop from being closed by $i \cdot j$ to being closed by $i-1 \cdot j+1$ is determined solely by the size of the loop and the change in stacking stability of the closing base pair. The interior base pair $i' \cdot j'$ is thus irrelevant for determining the stability change.

- the asymmetry of the loop;
- the stacking energies of the interior and exterior base pairs with the nearest unpaired bases;

thus allowing us to write

$$eL(i,j,i',j') = \text{size}(i' - i + j - j' - 2) + \text{asymmetry}(i' - i - 1, j - j' - 1) + \text{stacking}(i \cdot j) + \text{stacking}(i' \cdot j') \quad (2)$$

In the following, we will further assume that the lopsidedness and the size dependence of the asymmetry function can be separated out, or more specifically that:

$$\text{asymmetry}(k+1, l+1) = \text{asymmetry}(k,l) + f(k+1) \quad (3)$$

holds. Thus, the change in the asymmetry function when varying the size while maintaining lopsidedness only depends on the size of the loop.

Finding optimal internal loops

If the assumption of equation (3) holds, we propose the following algorithm as an alternative to compute the *VBI*(i, j) entries in the dynamic programming algorithm for predicting RNA secondary structure. [Actually, for an entry (i, j), we assume that most of the candidate internal loops—loops larger than size 2—have already been evaluated. The algorithm describes how to evaluate the small loops that have yet to be evaluated and how to use this information to evaluate some larger loops that are candidates for other entries of *VBI*, thus justifying the assumption.] The rationale behind the algorithm is to extend loops while retaining lopsidedness, as depicted in Figure 1.

Proposition 1. *Algorithm 1 computes VBI correctly under the assumption of equation* (*3*)*. Furthermore, the time required to compute the entire table is* $O(n^3)$.

The time complexity of $O(n^3)$ is easy to see, since each of the $O(n^2)$ entries of *VBI* uses time $O(n)$. To prove the correctness of the algorithm, we will start by sketching a simpler algorithm using space $O(n^3)$. Then, we will argue that algorithm 1 is similar to this algorithm except for the order in which the computations are carried out.

If the stability of an internal loop is assumed only to depend on the size of the loop, it has long been known that the entries of *VBI* can be computed in total time $O(n^3)$ (Waterman and Smith, 1986). We will describe a similar method that also handles stacking effects and asymmetry functions satisfying equation (3).

We define a new array *VBI′* such that *VBI′*(i, j, l) is the minimal energy of an internal loop of size l with exterior base pair $i \cdot j$. The following lemma establishes a useful relationship between the entries of *VBI′*.

Algorithm 1. Computation of *VBI*(i, j).

/ VBI(i,j) has already been calculated for all loops of size greater than 2; examine loops smaller than 2 */*

for a = 1 to 2 **do**

Let E be the energy of optimal loop of size a closed by $i \cdot j$

if $E <$ *VBI*(i, j) **then**

VBI(i, j) = E

for b = 1 to min($i-1$, $n-j$) **do**

/* *Examine loops of size* $a + 2b$ *with closing pair* $i - b \cdot j + b$.
Invariant: $E = VBI'(i - b + 1, j + b - 1, a + 2(b - 1))$ */
$E = E - \text{size}(a + 2b - 2) + f(a + 2b - 2)$
$+ \text{size}(a + 2b) + \text{stacking}(i - 1 \cdot j + 1)$
$- \text{stacking}(i \cdot j)$
if a bulge closed by $i - b \cdot j + b$ of size $a + 2b$ has energy lower than E **then**
Let E be the energy of this bulge
if $E < VBI(i - b, j + b)$ **then**
$VBI(i - b, j + b) = E$
end for
end for

Lemma 1. *If equation* (3) *holds, then for* $l > 2$

$$VBI'(i, j, l) = \min \begin{cases} VBI'(i + 1, j - 1, l - 2) + \\ \quad \text{size}(l) + f(l - 2) + \\ \quad \text{stacking}(i \cdot j) - \text{stacking}(i + 1 \cdot j - 1) \\ V(i + 1, j - l - 1) + eL(i, j, i + 1, j - l - 1) \\ V(i + l + 1, j - 1) + eL(i, j, i + l + 1, j - l) \end{cases} \quad (4)$$

Proof. By definition

$$VBI(i, j, l) = \min_{\substack{i < i' < j' < j \\ i' - i + j - j' - 2 = l}} \{eL(i, j, i', j') + V(i', j')\} \quad (5)$$

The last two entries of equation (4) handle the cases where this minimum is obtained by a bulge, i.e. at $i' = i + 1$ or $j' = j - 1$. Otherwise, the minimum is the minimum over

$$\begin{aligned} &eL(i, j, i', j') + V(i', j') \\ &\quad = \text{size}(l) + \text{asymmetry}(i' - i - 1, j - j' - 1) \\ &\qquad + \text{stacking}(i - j) + \text{stacking}(i' \cdot j') + V(i', j') \\ &\quad = \text{size}(l) + \text{asymmetry}(i' - i - 2, j - j' - 2) + f(l - 2) \\ &\qquad + \text{stacking}(i \cdot j) + \text{stacking}(i' \cdot j') + \text{V}(i', j') \\ &\quad = \text{size}(l - 2) + \text{asymmetry}(i' - i - 2, j - j' - 2) \\ &\qquad + \text{stacking}(i + 1 \cdot j - 1) + \text{stacking}(i' \cdot j') + V(i', j') \\ &\qquad + \text{size}(l) - \text{size}(l - 2) + f(l - 2) \\ &\qquad + \text{stacking}(i \cdot j) - \text{stacking}(i + 1 \cdot j - 1) \end{aligned}$$

for all $i' < j'$ with $i' > i + 1, j' < j - 1$ and $i' - (i + 1) + (j - 1) - j' - 2 = l - 2$. The last two lines of the last equation are independent of i' and j', and can thus be moved out of the minimum. The minimum of the first two lines over i' and j' satisfying the above constraints is exactly $VBI'(i + 1, j - 1, l - 2)$, thus proving the lemma. □

Lemma 1 yields the basic recursion needed to compute each entry of VBI' in constant time. (This is, of course, assuming that entries of V are ready at hand when we need them. The cost of computing the entries of V can, however, be charged to V and thus we do not have to consider it here.)

It is easily observed that VBI' contains $O(n^3)$ entries and that VBI can be calculated from VBI' as:

$$VBI(i, j) = \min_l \{VBI'(i, j, l)\} \quad (6)$$

each of the $O(n^2)$ entries being computable in time $O(n)$. Thus, VBI can be computed in time $O(n^3)$ including the time used to compute VBI'. Unfortunately, the table VBI' requires space $O(n^3)$, thus rendering this method somewhat impractical. However, it can be observed that we only need $VBI'(i, j, l)$ at most twice, namely when:

- determining whether it is a candidate for $VBI(i, j)$;
- calculating the value of $VBI'(i - 1, j + 1, l + 2)$.

This is used in algorithm 1 as whenever we have the value of $VBI'(i, j, l)$ ready at hand, we check to see whether it is a candidate for $VBI(i, j)$, i.e. whether it is the minimal value seen so far. Then, we immediately proceed to calculate the value of $VBI'(i - 1, j + 1, l + 2)$. When we get to computing the final value of $VBI(i, j)$, we know that the values of $VBI'(i, j, k)$ for all $k > 2$ have already been examined. We thus compute the minimum of equation (6) in a lazy fashion, comparing values as they become available.

Plausibility of the asymmetry function restriction

The assumption of equation (3) might seem somewhat unrealistic as, for one thing, we treat bulges just as if they were normal internal loops. If equation (3) holds for $\min(k, l) \geq c - 1$, we can, however, modify the algorithm, resulting in an increase in time complexity by a factor of c for a total time complexity of $O(cn^3)$. This is done simply by examining all the $O(cn^3)$ loops with a stem of unpaired bases shorter than c separately and then applying the technique of extending loops while retaining lopsidedness to the rest of the loops. Thus, bulges can be treated specially while only doubling the time complexity.

Papanicolaou *et al.* (1984) proposed an asymmetry penalty function on the form:

$$\text{asymmetry}(k, l) = \min\{K, N_{k,l} f(M_{k,l})\} \quad (7)$$

with $N_{k,l} = |k - l|$ and $M_{k,l} = \min\{k, l, c\}$. We observe that $N_{k+1,l+1} = N_{k,l}$ and that $M_{k+1,l+1} = M_{k,l}$ if $\min\{k, l\} \geq c$. It follows that asymmetry$(k + 1, l + 1)$ = asymmetry(k, l) if $\min\{k, l\} \geq c$ and thus asymmetry functions on this form adhere to the above relaxed assumption, allowing us to solve the RNA secondary structure prediction problem in $O(cn^3)$. In Papanicolaou *et al.* (1984), an asymmetry function with $c = 5$ was proposed. A modification of the parameters based on thermodynamic studies was proposed in Peritz *et al.* (1991). With these parameters $c = 1$, thus allowing us to only treat bulges specially. (Sequence-dependent destabilizing energies are available for internal loops of size three. These—and similar specific energy functions for small

loops—can be handled as a special case without affecting the general method for calculating internal loop stability though.)

Computing the partition function

In McCaskill (1990), it is described how to compute the full equilibrium partition function and thus the probabilities of all base pairs. The method used closely mimics the free energy calculation described above and thus it should be of no surprise that the method presented here also applies to the calculation of the partition function. In this section, we will briefly sketch how to compute the internal loops' contribution to the partition function.

In McCaskill (1990), $Q_{i,j}$ denotes the partition function on the segment from base i through base j, while $Q^b_{i,j}$ denotes the restricted partition function for the same sequence segment with the added condition that bases i and j form a base pair. Thus $Q^b_{i,j}$ corresponds to $V(i, j)$ in energy calculations. We will specify how to calculate the contributions from structures with an internal loop closed by $i \cdot j$.

From McCaskill [1990, equations (4) and (7)], it is seen that the contributions from these structures—if we consider a stacked pair an internal loop of size 0—are:

$$\sum_{i<h<l<j} e^{-eL(i,j,h,l)/kT} Q^b_{h,l} \tag{8}$$

where McCaskill [1990, equation (7)] uses $F_2(i, j, h, l)$ to gather the energies of all structures with an internal loop with base pairs $i \cdot j$ and $h \cdot l$, thus reducing the terms of the sum to $e^{-F_2(i,j,h,l)/kT}$.

Similar to the above approach, we define $Q^{il}_{i,j,l}$ to be the partition function for all structures with an internal loop of size l closed by $i \cdot j$, thus corresponding to $VBI'(i, j, l)$ in the above energy calculations. Now it can be proven that:

$$\begin{aligned} Q^{il}_{i,j,l} = {} & Q^{il}_{i+1,j-1,l-2} e^{(\text{size}(l-2) - \text{size}(l) - f(l-2))/kT} \\ & \cdot e^{(\text{stacking}(i+1\cdot j-1) - \text{stacking}(i\cdot j))/kT} \\ & + Q^b_{i+1,j-l-1} e^{-eL(i,j,i+l,j-l-1)/kT} \\ & + Q^b_{i+l+1,j-1} e^{-eL(i,j,i+l+1,j-1)/kT} \end{aligned} \tag{9}$$

by similar arguments as in the proof of lemma 1. Note that the only reason we have split the first exponential is to make the equation fit in a column. We can rewrite equation (8) as:

$$\sum_{l=0}^{j-i-2} Q^{il}_{i,j,l} \tag{10}$$

and based on equations (9) and (10) we can now proceed to present algorithm 2 to handle internal loop contributions to the partition function; the observant reader will notice the close similarity between algorithms 1 and 2. Again, it is an easy observation that the time complexity is $O(n^3)$ and the correctness of algorithm 2 can be proven by arguments similar to the above.

Algorithm 2. Computation of the contribution of internal loops to the partition functions.

/ Contributions from loops larger than 2 has already been added to $Q^b_{i,j}$; add contributions from loops smaller than 2*/*

for $k = 1$ **to** 2 **do**

Let Q be the partition function for all internal loops of size k

$Q^b_{i,j} = Q^b_{i,j} + Q$

for l = 1 **to** min(i – 1, n – j) **do**

$$\begin{aligned} Q = Q e^{(\text{size}(k+2l-2) - \text{size}(k+2l) - f(k+2l-2))/kT} & \\ \cdot e^{(\text{stacking}(i+1\cdot j-1) - \text{stacking}(i\cdot j))/kT} & \\ + Q^b_{i+1,j-l-1} e^{-eL(i,j,i+1,j-l-1)/kT} & \\ + Q^b_{i+l+1,j-1} e^{-eL(i,j,i+l+1,j-1)/kT} & \end{aligned}$$

$Q^b_{i-l,j+l} = Q^b_{i-l,j+l} + Q$

end for

end for

Implementation

The method described here has been implemented in *ZUKER* (ZUKER—Unlimited Ken Energy-based RNA-folding, the name reflecting that no limit is imposed on how far to look for the closing base pair of an internal loop), a C program to find the optimal structure of an RNA sequence based on energy rules. To be able to compare the performance of this method to previously used methods, compiler directives determine whether the compiled code will use complete enumeration of all internal loops or the method described here, and whether only to consider loops smaller than a specified size. By this we hope to have eliminated most of the noise due to differences in implementations so as to get a comparison of the underlying methods.

We decided to test our method against the complete enumeration method, both when using a cut-off size of 30 for internal loops (a commonly used cut-off size) and when allowing loops of any size. All four methods were tested with random sequences of length 500 and 1000, respectively, and the results are summarized in Table 1. As expected, a huge increase in performance is obtained when allowing internal loops of any size, but even when limiting internal loops to size at most 30, our method obtains a speed-up of 30–40% compared to the complete enumeration method. We are currently working on implementing the method for calculating the optimal substructure on the parts of the sequence excluding the substring from i through j, thus allowing the prediction of suboptimal structures as described in Zuker (1989) and calculation of base pair probabilities based on partition functions as described in McCaskill (1990).

Table 1. Comparison of different methods to evaluate internal loops. The running times are as reported by the Unix time command on a Silicon Graphics Indigo 2

Sequence length	500	1000
Complete enumeration, unlimited loop size	2119 s	35 988 s
Our method, unlimited loop size	127 s	1123 s
Complete enumeration, loop size ≤ 30	48 s	264 s
Our method, loop size ≤ 30	30 s	182 s

Discussion

It is well known that heuristics may speed up the evaluation of internal loops in practice. One way to do this is for all subsequences to keep track of the most stable structure of any of its subsequences. This is then used to cut off the evaluation of large loops closed by some base pair when it is evident that they cannot be more stable than the most stable structure closed by that base pair found so far.

As the method described above actually evaluates internal loops closed by some base pair in order of decreasing size, this heuristic cannot be combined with our method. We have instead implemented a heuristic based on determining upper bounds for the free energy of the optimal multi-branched loop closed by some base pair. This heuristic unfortunately does not seem to have a positive effect for sequences shorter than 1000 nucleotides, as the time spent determining when to stop further evaluation exceeds the time that would have been spent evaluating the rest of the loops for all but very long sequences.

It would, of course, be more interesting to obtain further improvements on the worst-case behavior of the algorithm, possibly by applying some advanced search techniques similar to those described in Eppstein *et al.* (1988). This is not a straightforward task though, as our method has shifted the focus from the exterior (closing) base pair to the interior base pair of an internal loop. The same interior base pair might be optimal for several choices of exterior base pairs. Furthermore, the exterior base pair that yields the most stable substructure with a specified interior base pair might not even be one of them. Thus, it is of no use just to search for the exterior base pair yielding the most stable substructure.

Finally, it should be mentioned that current methods for energy-based RNA secondary structure prediction only consider structures that do not contain pseudoknots. Probably the open question of RNA secondary structure prediction is to put forward a model including pseudoknots that allows fair predictions in reasonable time.

Acknowledgement

This work was supported, in part, by NIGMS grant GM54250 to M.Z.

References

Eppstein,D., Galil,Z. and Giancarlo,R. (1988) Speeding up dynamic programming. In *Proceedings of the 29th Symposium on Foundations of Computer Science*. Assoc. Comput. Mach., pp. 488–496

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

Hopkin,K. (1998) When RNA ruled—another lost world? *HMS Beagle, The BioMedNet Magazine,* 27. http://biomednet.com/hmsbeagle/1998/27/resnews/meeting.htm.

McCaskill,J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

Nussinov,R. and Jacobson,A.B. (1980) Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl Acad. Sci. USA*, **77**, 6309–6313.

Papanicolaou,C., Gouy,M. and Ninio,J. (1984) An energy model that predicts the correct folding of both the tRNA and the 5S RNA molecules. *Nucleic Acids Res.*, **12**, 31–44.

Peritz,A.E., Kierzek,R., Sugimoto,N. and Turner,D.H. (1991) Thermodynamic study of internal loops in oligoribonucleotides: symmetric loops are more stable than asymmetric loops. *Biochemistry*, **30**, 6428–6436.

Tinoco,I., Uhlenbeck,O.C. and Levine,M.D. (1971) Estimation of secondary structure in ribonucleic acids. *Nature*, **230**, 362–367.

Tinoco,I., Borer,P.N., Dengler,B., Levine,M.D., Uhlenbeck,O.C., Crothers,D.M. and Gralla,J. (1973) Improved estimation of secondary structure in ribonucleic acids. *Nature New Biol.*, **246**, 40–41.

Turner,D.H., Sugimoto,N. and Freier,S.M. (1988) RNA structure prediction. *Annu. Rev. Biophys. Biophys. Chem.*, **17**, 167–192.

Waterman,M.S. and Smith,T.F. (1986) Rapid dynamic programming methods for RNA secondary structure. *Adv. Appl. Math.*, **7**, 455–464.

Zuker,M. (1989) On finding all suboptimal foldings of an RNA molecule. *Science*, **244**, 48–52.

Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, **9**, 133–148.