BRICS

Basic Research in Computer Science

BRICS RS-99-15 Lyngsø et al.: An Improved Algorithm for RNA Secondary Structure Prediction

An Improved Algorithm for RNA Secondary Structure Prediction

Rune B. Lyngsø
Michael Zuker
Christian N. S. Pedersen

BRICS Report Series **RS-99-15**

ISSN 0909-0878 **May 1999**

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/99/15/`

An improved algorithm for RNA secondary structure prediction

R. B. Lyngsø* Michael Zuker† C. N. S. Pedersen‡

Abstract

Though not as abundant in known biological processes as proteins, RNA molecules serve as more than mere intermediaries between DNA and proteins, e.g. as catalytic molecules. 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. We present a new method to evaluate all possible internal loops of size at most k in an RNA sequence, s, in time $O(k|s|^2)$; this is an improvement from the previously used method that uses time $O(k^2|s|^2)$. For unlimited loop size this improves the overall complexity of evaluating RNA secondary structures from $O(|s|^4)$ to $O(|s|^3)$ and the method applies equally well to finding the optimal structure and calculating the equilibrium partition function. We use our method to examine the soundness of setting $k = 30$, a commonly used heuristic.

1 Introduction

Structure prediction remains one of the most compelling, yet elusive areas of computational biology. Not yielding to overwhelming numbers and

*Department of Computer Science, University of Aarhus, Denmark. E-mail: rlyngsoe@daimi.au.dk. Work done while visiting the Institute for Biomedical Computing at Washington University, St. Louis.

†Institute for Biomedical Computing, Washington University, USA. E-mail: zuker@ibc.wustl.edu

‡Basic Research In Computer Science, Centre of the Danish National Research Foundation, University of Aarhus, Denmark. Supported by the ESPRIT Long Term Research Programme of the EU under project number 20244 (ALCOM-IT). E-mail: cstorm@brics.dk.

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, much progress has been achieved. Dynamic programming algorithms combined with the nearest neighbour model and experimentally determined free energy parameters give rigorous solutions to the problems of computing minimum free energy structures, structures that are usually close to real world 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. According to the theory of thermodynamics the optimal foldings of an RNA sequence are those of minimum free energy, and thus the native foldings, i.e. the foldings encountered in the real world, should correspond to the optimal foldings. Furthermore, thermodynamics tells us that the folding of an RNA sequence in the real world is actually a probability distribution over all possible structures, where the probability of a specific structure is proportional to an exponential of the free energy of the structure. For a set of structures, the partition function is the sum over all structures of the set of the exponentials of the free energies.

Information on the secondary structure of an RNA molecule can be used as a stepping-stone to modelling 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 [6], the prediction of RNA secondary structure should continue to be important for biomolecule engineering.

A model was proposed in [16, 15] to calculate the stability (in terms of free energy) of a folded RNA molecule by adding independent contributions from base pair stacking and loop destabilising terms from the secondary structure. This model has proven 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. in [23, 10]. Zuker [21] proposes a method to determine all base pairs that can participate in structures with a free energy within a specified range from the optimal. McCaskill [9] demonstrates how a related dynamic programming algorithm can be used to calculate equilibrium partition functions, which lead to exact calculations of base pair probabilities in the model.

A major problem for these algorithms is the time required to evaluate possible internal loops. In general, this requires time $O(|s|^4)$ which is often circumvented by assuming that only 'small' loops need to be considered (e.g. [9]). 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 $O(|s|^2)$ thus reducing the overall complexity to $O(|s|^3)$. If the stability of an internal loop can be assumed only to depend on the size of the internal loop, Waterman et. al. [18] describes how to reduce the time requirement to $O(|s|^3)$[1]. This is further improved to $O(|s|^2 \log^2 |s|)$ for convex free energy functions by Eppstein et.al. [1]. Affine free energy functions (i.e. of the form $a + bn$, where n is the size of the loop) allows for $O(|s|^2)$ computation time by borrowing a simple method used in sequence alignment [2].

Unfortunately the currently used free energy functions for internal loops are not convex, let alone affine. Furthermore, the technique described in [1] 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 [9] 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 $O(|s|^3)$. 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 section 2 we briefly review the basic dynamic programming algorithm for RNA secondary structure prediction and introduce the notation we will be using. In section 3 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 in section 3.2, and describe how the same technique can be used to calculate the contributions to the partition function from structures with internal loops in section 3.3. In section 4 we compare our method to the previously used method, and in section 5 we present an experiment using the new algorithm to analyse a hitherto commonly used heuristic. In section 6 we discuss some future directions for improvements.

[1] This method is also referred to by [9] 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.

2 Basic dynamic programming algorithm

A secondary structure of a sequence s is a set S of base pairs $i \cdot j$ with $1 \leq i < j \leq |s|$, 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 further assume that the structure does not contain pseudo-knots. A pseudo-knot is two "overlapping" base pairs, that is, base pairs $i \cdot j$ and $i' \cdot j'$ with $i < i' < j < j'$.

One can view a pseudo-knot free secondary structure S as a collection of *loops* together with some *external* unpaired bases (see figure 1). Let $i < k < j$ with $i \cdot j \in S$. Then k is said to be *accessible* from $i \cdot j$ if for all $i' \cdot j' \in S$ it is not the case that $i < i' < k < j' < j$. The base pair $i \cdot j$ is said to be the *exterior* base pair of (or *closing*) the loop consisting of $i \cdot j$ and all bases accessible from it. If i' and j' are accessible from $i \cdot j$ and $i' \cdot j' \in S$ – observe that for a structure without pseudo-knots either both or none of i' and j' will be accessible from $i \cdot j$ if $i' \cdot j' \in S$ – then $i' \cdot j'$ is called an *interior* base pair of the loop and is said to be accessible from $i \cdot j$. If there are no interior base pairs the loop is called a *hairpin* loop. With one interior base pair it is called a *stacked pair* if $i' = i + 1$ and $j' = j - 1$, and otherwise it is called an *internal* loop (*bulges* are a special kind of internal loops with either $i' = i + 1$ or $j' = j - 1$). Loops with more than one interior base pair are called *multibranched* loops. Unpaired bases and base pairs not accessible from any base pair are called external.

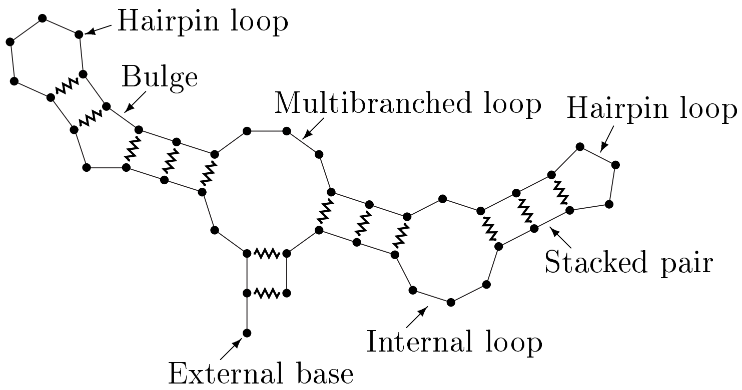


Figure 1: An example RNA structure. Bases are depicted by circles, the RNA backbone by straight lines and base pairings by zigzagged lines.

RNA secondary structure prediction is the problem of determining the most stable structure for a given sequence. We measure stability in terms of the free energy of the structure. Thus we want to find a structure of minimal free energy which we will also call an optimal structure. The energy of a secondary structure is assumed to be the sum of the energies of the loops of the structure and furthermore the loops are assumed to be independent, that is, the energy of a loop only depends on the loop and not on the rest of the structure[15].

Based on these assumptions one can specify a recursion to calculate the energy of the optimal structure for a sequence s [23, 10]. Before presenting our improvement to the part of the algorithm dealing with internal loops, we will briefly review the hitherto used method. We use the same notation as in [17]. Four arrays[2] – W, V, VBI and VM – are used to hold the minimal free energy of certain restricted structures of subsequences of s. The entries of these arrays are interdependent and can be calculated recursively using pre-specified free energy functions – eS, eH, eL and eM – for the contributions from the various types of loops as follows.

- The energy of an optimal structure of the subsequence from 1 through i:

$$W(i) = \min\{W(i-1), \min_{1<j\leq i}\{W(j-1) + V(j,i)\}\}.$$

- The energy of an optimal structure of the subsequence from i through j closed by $i \cdot j$:

$$V(i,j) = \min\{eH(i,j), eS(i,j) + V(i+1, j-1), \\ VBI(i,j), VM(i,j)\}$$

  where $eH(i,j)$ is the energy of a hairpin loop closed by $i \cdot j$ and $eS(i,j)$ is the energy of stacking base pair $i \cdot j$ with $i+1 \cdot j-1$.

- The energy of an optimal structure of the subsequence from i through j where $i \cdot j$ closes a bulge or an internal loop:

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

[2] Actually two arrays – V and W – suffices, but we will use four arrays to simplify the description. Below we will introduce a fifth array WM that will also be needed in an efficient implementation.

where $eL(i,j,i',j')$ is the energy of a bulge or internal loop with exterior base pair $i \cdot j$ and interior base pair $i' \cdot j'$.

- The energy of an optimal structure of the subsequence from i through j where $i \cdot j$ closes a multibranched loop:

$$VM(i,j) = \min_{\substack{i<i_1<j_1< \\ \dots \\ <i_k<j_k<j}} \{eM(i,j,i_1,j_1,\dots,i_k,j_k) + \sum_{l=1}^{k} V(i_l,j_l)\}$$

where $k > 1$ and $eM(i,j,i_1,j_1,\dots,i_k,j_k)$ is the energy of a multibranched loop with exterior base pair $i \cdot j$ and interior base pairs $i_1 \cdot j_1, \dots, i_k \cdot j_k$.

When all entries of these arrays have been filled out, $W(|s|)$ contains the free energy for optimal structures and an optimal structure can be determined by backtracking the calculations that led to this free energy.

To make the problem of determining the optimal secondary structure tractable the following simplifying assumption is often made. The energy of multibranched loops can be decomposed into linear contributions from the number of unpaired bases in the loop, the number of branches in the loop and a constant[22][3], that is

$$eM(i,j,i_1,j_1,\dots,i_k,j_k) = \\ a + bk + c\left(i_1 - i - 1 + j - j_k - 1 + \sum_{l=1}^{k-1}(i_{l+1} - j_l - 1) \right). \quad (1)$$

We introduce an extra array

- The energy of an optimal structure of the subsequence from i through j that constitutes part of a multibranched loop structure, that is, where unpaired bases and external base pairs are penalised according to equation 1:

$$WM(i,j) = \min\{V(i,j) + b, WM(i,j-1) + c, WM(i+1,j) + c, \\ \min_{i<k\leq j}\{WM(i,k-1) + WM(k,j)\}\}$$

[3]It is known that the stability of a multibranched loop also depends on the stacking effects of the base pairs in the loop and their neighbouring unpaired bases. These effects can also be handled efficiently, but for simplicity we have omitted the details here.

which enables us to restate the calculation of the energy of the optimal multibranched loop as

$$VM(i,j) = \min_{i+1<k\leq j-1} \{WM(i+1,k-1) + WM(k,j-1) + a\}.$$

Based on these recurrence relations we can by dynamic programming calculate the energy of the optimal structure in time $\mathrm{O}(|s|^3)$ – assuming that the free energy functions can be evaluated in constant time – except for the calculation of the entries of VBI which requires $\mathrm{O}(|s|^4)$ in total. The bottleneck of finding the optimal structures is thus the evaluation of internal loops. In the following section we will present a method to reduce the time used calculating the entries of VBI from $\mathrm{O}(|s|^4)$ to $\mathrm{O}(|s|^3)$, thereby improving the time complexity of the overall RNA secondary structure prediction algorithm from $\mathrm{O}(|s|^4)$ to $\mathrm{O}(|s|^3)$.

3 Efficient evaluation of internal loops

Examining the recursion for internal loops one observes that two base pairs, $i \cdot j$ and $i' \cdot j'$, may be compared as candidates for the interior base pair for numerous exterior base pairs. If $V(i,j) \ll V(i',j')$, it is evident that we would not have to consider $i' \cdot j'$ as a candidate interior base pair for any entry of VBI where $i \cdot j$ would also be a candidate interior base pair.

Though it would often in practice be the case that we could a priori discard many candidate interior base pairs by the above observation, we can not in general guarantee this to be the case. To get an improvement in the worst case performance of the evaluation of internal loops, we thus have to examine properties of the energy functions for internal loop stability that will allow us to group base pairs and entries of VBI, such that we only have to make one comparison between $i \cdot j$ and $i' \cdot j'$ to determine which one would yield the more stable structure for the entire group of entries. In this section we will exploit such properties of currently used energy functions leading to an algorithm for evaluating internal loops requiring worst case time $\mathrm{O}(|s|^3)$.

Currently used energy rules for internal loop stability (cf. [20]) split the contributions into three parts:

- An entropic term that depends on the size of the loop.
- Stacking energies for the mismatched base pairs adjacent to the enclosing (exterior *and* interior) base pairs.

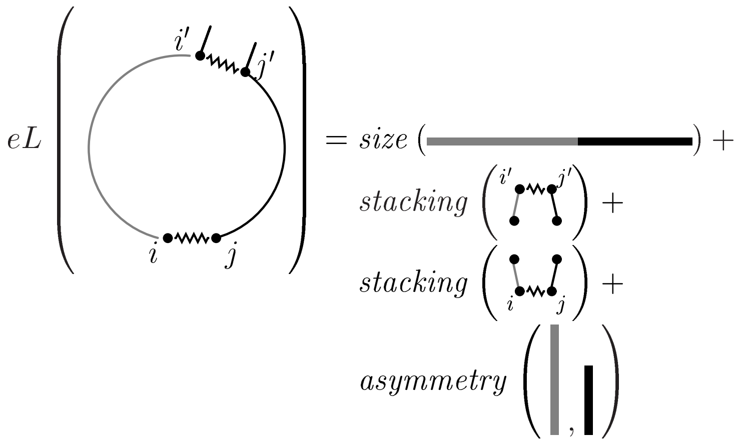


Figure 2: The energy function for internal loops can be split into a sum of independent contributions.

- An asymmetry penalty for asymmetric loops.

With this separation we can rewrite the internal loop energy function as

$$\begin{aligned} eL(i,j,i',j') = &\,\mathrm{size}(i'-i+j-j'-2)+ \\ &\,\mathrm{stacking}(i\cdot j)+\mathrm{stacking}(i'\cdot j')+ \\ &\,\mathrm{asymmetry}(i'-i-1,j-j'-1). \end{aligned} \tag{2}$$

Figure 2 gives a graphical representation of these components of the internal loop energy function. 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

$$\mathrm{asymmetry}(k+1,l+1) = \mathrm{asymmetry}(k,l) + g(k+l) \tag{3}$$

holds. The change of the asymmetry function when varying the size while maintaining lopsidedness thus only depends on the size of the loop. This is equivalent to assuming that

$$\mathrm{asymmetry}(k,l) = \mathrm{lopsidedness}(|k-l|) + \mathrm{size}'(k+l), \tag{4}$$

where one can observe that the g term in equation 3 corresponds to changes in the size$'$ term in equation 4. This size-dependence of the asymmetry function can be moved to the size-function of the overall internal loop energy function, thus allowing us to restate the assumption of equation 3 as

$$\text{asymmetry}(k+1, l+1) = \text{asymmetry}(k, l). \tag{5}$$

In the rest of this paper we will therefore omit the g term, but the formulation of equation 3 might be useful when specifying or recognising an asymmetry function obeying the assumption.

3.1 Finding optimal internal loops

If the assumption of equation 3 holds, we propose algorithm 1 as an efficient alternative to compute the $VBI(i, j)$ entries in the dynamic programming algorithm for predicting RNA secondary structure. The algorithm is an extension of the ideas in [18] where an $\mathrm{O}(n^3)$ method for calculating the entries of VBI, assuming that the stability of an internal loop only depends on the size of the loop, was presented. The rationale behind the algorithm is, that when we extend loops while retaining lopsidedness we can reuse comparisons as depicted in Figure 3. Thus for a pair of indices, i and j, the algorithm *does not* compute the $VBI(i, j)$ entry. Instead, if we denote all internal loops with a specific size and exterior base pair as a *class* of internal loops, the algorithm evaluates all classes of internal loops where $i \cdot j$ is the middle candidate base pair, that is, choosing $i \cdot j$ as the interior base pair results in a symmetric loop (or almost symmetric – loops of odd size will always have a lopsidedness of at least one).

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

The time complexity of $\mathrm{O}(n^3)$ is easy to see, since the algorithm for each of the $\mathrm{O}(n^2)$ pairs of indices, i and j, uses time $\mathrm{O}(n)$. To prove the correctness of the algorithm, we will start by sketching a simpler algorithm for which the correctness is obvious, but that has the drawback of using space $\mathrm{O}(n^3)$. Then we will argue that algorithm 1 is similar to this algorithm except for the order in which the computations are carried out, that is, the order in which the different candidate interior loops for a specific entry of VBI are evaluated. Hence, the correctness of the simpler algorithm implies the correctness of algorithm 1.

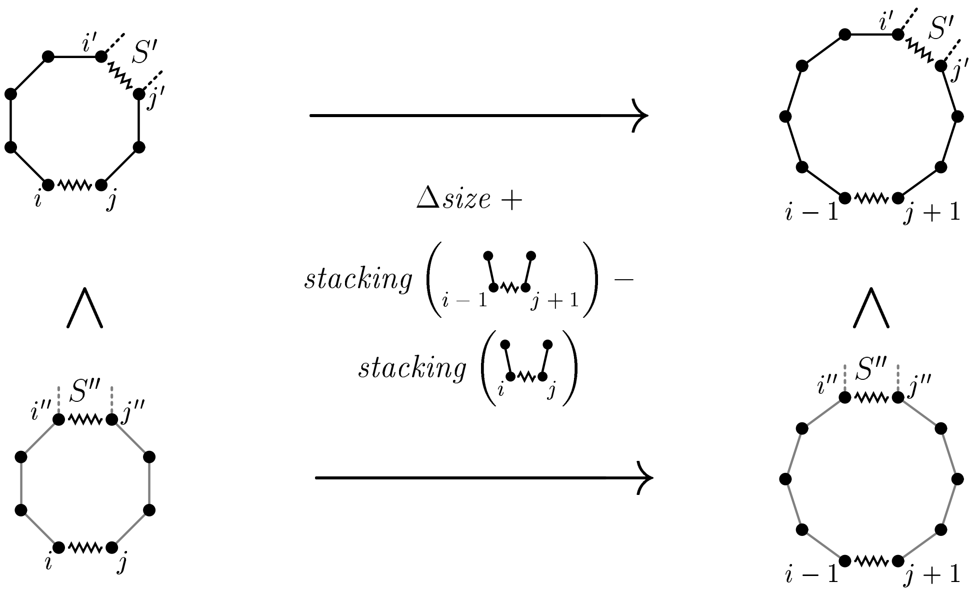


Figure 3: The difference in destabilising energy when extending a loop from being closed by $i \cdot j$ to being closed by $i-1 \cdot j+1$ is determined solely by the size of the loop and the change in stacking stability of the closing base pair. We can thus reuse comparisons between different choices of interior base pairs, e.g. $i' \cdot j'$ and $i'' \cdot j''$.

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'.

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

$$VBI'(i,j,l) = \min \begin{cases} VBI'(i+1, j-1, l-2) + \\ \qquad \text{size}(l) - \text{size}(l-2) + \\ \qquad \text{stacking}(i \cdot j) - \text{stacking}(i+1 \cdot j-1) \\ V(i+1, j-l-1) + eL(i,j,i+1,j-l-1) \\ V(i+l+1, j-1) + eL(i,j,i+l+1,j-1). \end{cases} \tag{6}$$

Proof. By definition

$$VBI'(i,j,l) = \min_{\substack{i<i'<j'<j \\ i'-i+j-j'-2=l}} \{eL(i,j,i',j') + V(i',j')\}. \tag{7}$$

The last two entries of equation 6 handle the cases where this minimum is obtained by a bulge, that is at $i' = i+1$ or $j' = j-1$. Otherwise the minimum is the minimum over

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

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. $\square$

Lemma 1 yields the basic recursion needed to compute each entry of VBI' in constant time[4]. It is easily observed that VBI' contains $O(n^3)$ entries and that VBI can be calculated from VBI' as

$$VBI(i,j) = \min_{l}\{VBI'(i,j,l)\}, \tag{8}$$

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 $VBI'(i-1, j+1, l+2)$.

[4] 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 don't have to consider it here.

Algorithm 1 Evaluation of classes of internal loops with size $2l + a$ and exterior base pair $i - l \cdot j + l + a$.

/* *When $a = 0$ loops of even size are handled and when $a = 1$ loops of odd size are handled; this is necessary as we increase the loop size by two in each iteration.* */

for $a = 0$ **to** 1 **do**

  /* *E maintains the energy of the optimal loop except for* size *and external stacking contributions.* */

  $E = \infty$

  /* *Iterate through the exterior base pairs. For even sized loops we skip $l = 1$ as this yields a stacked base pair.* */

  for $l = 2 - a$ **to** $\min\{i - 1, |s| - j - a\}$ **do**

    /* *Examine the two new candidate interior base pairs, i.e. the interior base pairs next to the currently considered exterior base pair.* */

$$
\begin{aligned}
E = \min\{E, & V(i - l + 1, j - l + 1) + \\
& \quad \text{asymmetry}(0, 2l + a - 2) + \\
& \quad \text{stacking}(i - l + 1, j - l + 1), \\
& V(i + a + l - 1, j + a + l - 1) + \\
& \quad \text{asymmetry}(2l + a - 2, 0) + \\
& \quad \text{stacking}(i + a + l - 1, j + a + l - 1)\}
\end{aligned}
$$

    /* *Update VBI for the currently considered exterior base pair.* */

$$
\begin{aligned}
VBI(i - l, j + a + l) = \min\{& VBI(i - l, j + a + l), \\
& E + \text{size}(2l + a - 2) + \text{stacking}(i - l, j + a + l)\}
\end{aligned}
$$

  end for

end for

This is used in algorithm 1 to avoid maintaining the VBI' table. Instead we use E to hold the value[5] that should otherwise be stored in one of the entries of VBI'. We use this value to check it as a candidate for the relevant entry of VBI, according to equation 8, in the second minimum of the **for**-loop in algorithm 1. After this check we only need the value to calculate the value corresponding to another entry of VBI'; this is done in the first minimum in the next iteration of the **for**-loop. Now the value can safely be discarded as it is no longer needed. It is straightforward to verify

[5] To avoid having to keep adding and subtracting the size and external stacking terms in algorithm 1 we defer adding these terms until the value is considered as a candidate for one of the VBI entries.

that the value that should otherwise have been stored in $VBI'(i', j', l)$ is handled when algorithm 1 is invoked with $i = i' + \lfloor \frac{l}{2} \rfloor$ and $j = j' - \lceil \frac{l}{2} \rceil$. The correctness of the value maintained in E can easily be proved by induction, using lemma 1.

3.2 The Asymmetry Function Assumption

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 only holds for $\min(k, l) \geq c - 1$ we can however modify the algorithm to handle this situation, a modification that does lead to 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, starting the iteration at $l = c$ and adding or subtracting $c - 1$ from the indices of the interior base pairs considered, including where they partake in the parameters of the asymmetry function. Thus bulges can be treated specially while only doubling the time complexity.

Papanicolaou et. al. [11] propose an asymmetry penalty function on the form

$$\text{asymmetry}(k, l) = \min\{K, N_{k,l} f(M_{k,l})\}, \tag{9}$$

usually called Ninio type asymmetry penalty functions, with $N_{k,l} = |k - l|$ and $M_{k,l} = \min\{k, l, c\}$. The constants K and c and the function f are parameters of the penalty function. We observe that $N_{k+1,l+1} = N_{k,l}$ and that $M_{k+1,l+1} = M_{k,l}$ if $\min\{k, l\} \geq c$. For $\min\{k, l\} \geq c$ it thus follows that $\text{asymmetry}(k + 1, l + 1) = \text{asymmetry}(k, l)$, and thus asymmetry functions on this form adheres to the above relaxed assumption, allowing us to solve the RNA secondary structure prediction problem using Ninio type asymmetry penalty functions in time $O(cn^3)$. In [11] an asymmetry function with $c = 5$ was proposed. A modification of the parameters based on thermodynamic studies was proposed in [12]. With these parameters $c = 1$ thus allowing us to treat only bulges specially[6].

[6]Sequence dependent destabilising 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.

3.3 Computing the partition function

In [9] it is described how to compute the full equilibrium partition functions 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 in this paper also applies to the calculation of the partition functions. In this section we will briefly sketch how to compute the internal loops' contribution to the partition functions. The reader is refered to [9] for the full details on how to calculate the partition functions.

In [9] $Q_{i,j}$ 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 constraint that bases i and j form a base pair[7]. We will specify how to calculate the contributions from structures with an internal loop closed by $i \cdot j$.

From [9, equations 4 and 7] it is seen that the contributions from these structures – if we consider a stacked pair to be an internal loop of size 0 – are

$$\sum_{i<h<l<j} e^{-eL(i,j,h,l)/kT} Q^b_{h,l}\,, \tag{10}$$

where [9, 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 approach in section 3.1 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 energy calculations in section 3.1. Now it can be proved that

$$\begin{aligned} Q^{il}_{i,j,l} = {} & Q^{il}_{i+1,j-1,l-2} e^{(\mathrm{size}(l-2)-\mathrm{size}(l)+\mathrm{stacking}(i+1\cdot j-1)-\mathrm{stacking}(i\cdot j))/kT} \\ & + Q^b_{i+1,j-l-1} e^{-eL(i,j,i+1,j-l-1)/kT} + Q^b_{i+l+1,j-1} e^{-eL(i,j,i+l+1,j-1)/kT} \end{aligned} \tag{11}$$

by similar arguments as in the proof of lemma 1. There is a slight problem if $\mathrm{stacking}(i \cdot j) = \infty$ or $\mathrm{stacking}(i+1 \cdot j-1) = \infty$ – that is, if bases i and j or bases $i+1$ and $j-1$ does not form a base pair – but in the proof of equation 11 this can be handled by assuming that all stacking energies are finite. In the algorithm we handle it by postponing the

[7] Thus $Q^b_{i,j}$ corresponds to $V(i,j)$ in energy calculations.

Algorithm 2 Evaluation of classes of internal loops with size $2l + a$ and exterior base pair $i - l \cdot j + l + a$.

/* *Make sure to handle both even sized and odd sized loops.* */
for $a = 0$ **to** 1 **do**
  /* *Q maintains the partition function contribution for the current class of internal loops except for* size *and external stacking factors.* */
  $Q = 0$
  /* *Iterate through the exterior base pairs. For even sized loops we skip $l = 1$ as this yields a stacked base pair.* */
  for $l = 2 - a$ **to** $\min\{i - 1, |s| - j - a\}$ **do**
    /* *Add contributions from the two new interior base pairs, i.e. the interior base pairs next to the currently considered exterior base pair.* */
    $$Q = Q + Q^b_{i-l+1,j-l+1} e^{-(\text{asymmetry}(0,2l+a-2)+\text{stacking}(i-l+1 \cdot j-l+1))/kT} + Q^b_{i+a+l-1,j+a+l-1} e^{-(\text{asymmetry}(2l+a-2,0)+\text{stacking}(i+a+l-1 \cdot j+a+l-1))/kT}$$
    /* *Update Q^b with contributions from the currently considered class of internal loops.* */
    $$Q^b_{i-l,j+a+l} = Q^b_{i-l,j+a+l} + Qe^{-(\text{size}(2l+a-2)+\text{stacking}(i-l \cdot j+a+l))/kT}$$
  end for
end for

multiplication with the exponential of the stacking energies until adding the contribution of $Q^{il}_{i,j,l}$ to $Q^b_{i,j}$. We can now rewrite equation 10 as

$$\sum_{l=0}^{j-i-2} Q^{il}_{i,j,l}\,, \tag{12}$$

and based on equations 11 and 12 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 $\mathrm{O}(n^3)$, and the correctness of algorithm 2 can be proven by arguments similar to the proof of the correctness of algorithm 1.

4 Implementation

The method described in this paper has been implemented in *ZUKER*[8], 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 determines 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.

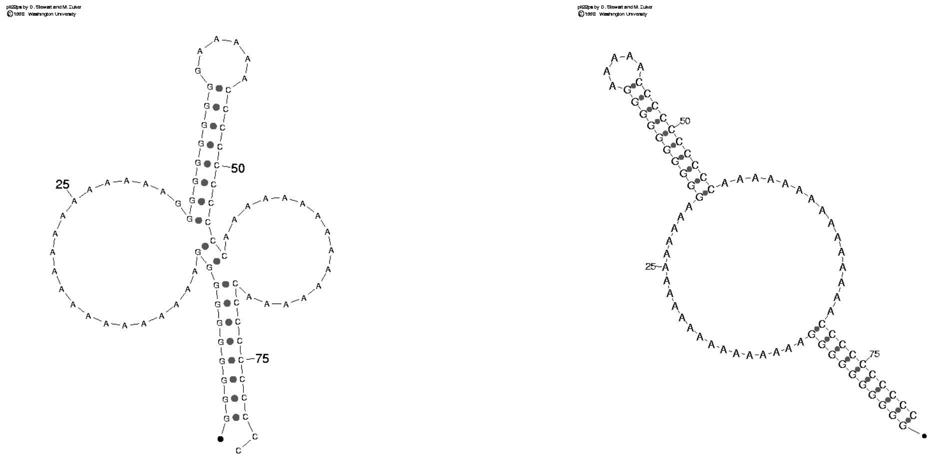
We decided to test our method against the complete enumeration method, both when using a cutoff size of 30 for internal loops (a commonly used cutoff size) and when allowing loops of any size. All four methods were tested with random sequences of length 500 and 1000, respectively, and the results are summarised 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 speedup of 30 – 40 % compared to the complete enumeration method.

Sequence length	500	1000
Complete enumeration, unlimited loop size	2,119 s	35,988 s
Our method, unlimited loop size	127 s	1,123 s
Complete enumeration, loop size ≤ 30	48 s	264 s
Our method, loop size ≤ 30	30 s	182 s

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.

The current implementation encompasses 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 [21] and calculation of base pair probabilities based on partition functions as described in [9]. We are currently working on adding coaxial stacking modifications to the multibranched loop evaluations, and on extending the program to take other param-

[8]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) Maximum loop size 30; Energy: -29.6 kcal/mol

(b) No maximum loop size; Energy: -42.9 kcal/mol

Figure 4: Foldings of the sequence GGGGGGGGGGAAAAAAAAAAAAAAAAAAAA GGGGGGGGGGAAAAACCCCCCCCCCAAAAAAAAAAAAAAACCCCCCCCCC

eters, e.g. mutual information or base pair confidences obtained from alignments, into account.

5 Experiments

To make the problem of determining the optimal secondary structure for an RNA sequence more tractable it has hitherto been common practice to limit the size of internal loops. The `mfold` server has a built-in limit of 30 and in [5] a limit of 30 is also hinted at. With the ability to make a rigorous search for the optimal structure, we decided to see whether this limit has been reasonable.

5.1 A constructed 'mean' sequence

The easiest way to find a loop of size larger than 30 is of course to construct it yourself. We constructed a sequence of length 80 consisting only of C's, G's and A's (but no U's), designed to fold into two stems of 10 base pairing C's and G's separated by an internal loop of 35 unpaired

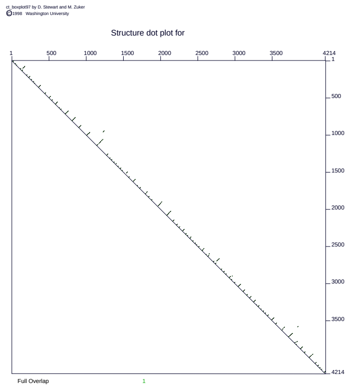


Figure 5: Dot-plot of the prediction of the Qβ structure at 65 °C. The absence of long range base pairings (dots far away from the diagonal) is apparent.

A's, and with a hairpin loop consisting of 5 A's. The result of folding this sequence at 37 °C with and without a size limit of 30, respectively, is shown in figure 4

One can observe that the prediction with a cutoff size of 30 does in fact pair most of the C's with G's – but instead of having the A's in one big internal loop they are folded out as two bulges. A further observation is that there can indeed be a major increase in stability by choosing one large internal loop instead of two smaller bulges.

Though this example may be cute, the interesting question of course is whether RNA sequences for which the optimal structure contains a large internal loop occur naturally. The reason that a cutoff size of 30 has been deemed reasonable is of course that no internal loops even close to this size are observed in a standard structure prediction at 37 °C. But when the temperature is increased, base pairs become less stable which may cause short stems of stacking base pairs to break up. We thus decided to look at a couple of sequences for which structure prediction at higher temperatures would be interesting.

5.2 Qβ

Jacobson [7] reported on some experiments on determining structural features in Qβ denatured to various extents. It is believed that denaturing effects relates to temperature effects, and we thus chose to fold this sequence at nine different temperatures in the range from 45 °C to 100 °C to see whether we would find any of the structural features reported by Jacobson.

None of these predicted foldings showed any signs of the features Jacobson reported – at higher temperatures the structure simply came apart as small structural fragments, usually covering less than 100 nucleotides. Furthermore we did not observe any internal loops larger than size 25. An example prediction is shown in figure 5.

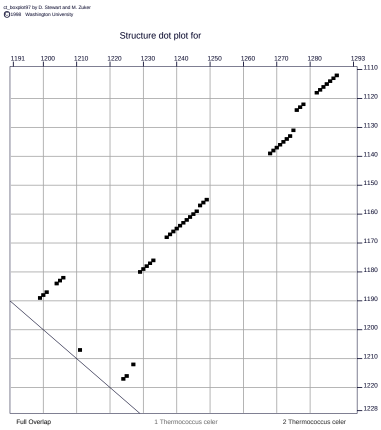
5.3 *Thermococcus celer*

Thermococcus celer is an organism that lives in solfataric marine water holes of Vulcano, Italy, at temperatures around 90 °C; its optimal growth temperature is reported to be around 88 °C [19]. Furthermore, the structure of the 23S subunit exhibits an internal loop of size 33 closed by base pairs $1139 \cdot 1268$ and $1155 \cdot 1249$, cf. [4, 3].

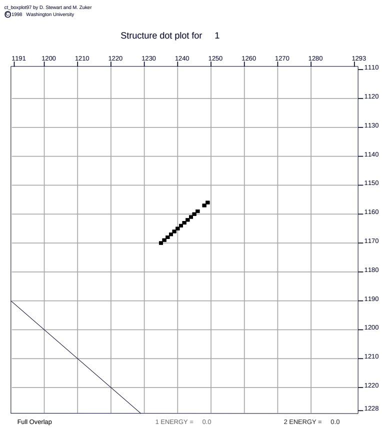
Folding this sequence at 88 °C we did (almost) get the inner stem of this internal loop but the outer stem came apart as two single strands (cf. figure 6(b)). When lowering the temperature to 75 °C we did get both stems, but the internal loop was split into two loops of size 2 and 27, respectively, by a short stem consisting of the base pairs $1141 \cdot 1266$ and $1142 \cdot 1265$ (cf. figure 6(c)).

We then tried to search the range of temperatures between 75 °C and 88 °C, and at 82 °C we did in fact correctly predict the internal loop of size 33 (cf. figure 6(d)). At this temperature we on the other hand missed the structure inside the inner stem, a structure that is quite well predicted at 75 °C; no temperature thus seemed decisively best for predicting this structural fragment. Generally, as with the Qβ predictions, these predictions missed long-range base pairings and predicted structures consisting of fragments covering less than 300 bases.

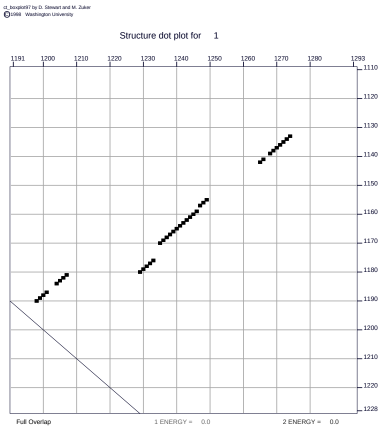
It should however be mentioned that a prediction at 82 °C with a cutoff size of 30 completely misses the outer stem and thus makes a prediction of this fragment identical to the prediction at 88 °C. Thus we get a decisively better prediction at this temperature when examining internal loops of all sizes than when using a cutoff size of 30.



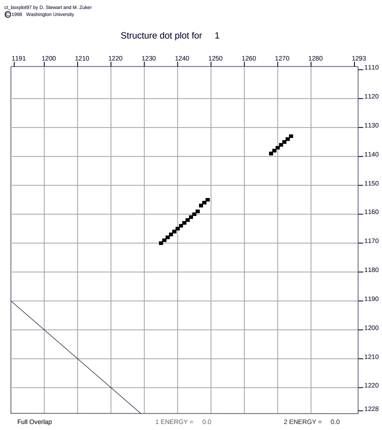
(a) Fragment of the structure between bases 1112 and 1288.



(b) Prediction of the same fragment at 88 °C.



(c) Prediction of the same fragment at 75 °C.



(d) Prediction of the same fragment at 82 °C.

Figure 6: Known and Predicted structures for *thermococcus celer*.

6 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 a specific base pair, when it is evident that they can not be more stable than the most stable structure closed by that base pair found so far.

As the method described in section 3 actually evaluates the internal loops closed by a specific base pair in order of decreasing size, the above heuristic can not 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, for all but very long sequences, the time spent determining when to stop further evaluation exceeds the time that would have been spent evaluating the rest of the loops.

It would of course be more interesting to obtain further improvements on the worst-case behaviour of the algorithm, possibly by applying some advanced search techniques similar to those described in [1]. 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 specific 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.

Our studies of structure predictions at high temperatures did not show an abundance of internal loops larger than the hitherto used cutoff size. There is thus no reason to suspect that predictions using this cutoff size are generally erroneous. We were however able to predict one internal loop that exceeds this size limit. Furthermore we predicted a number of internal loops with size larger than 20. This indicates that the cutoff size of 30 is probably a little bit to small for safe predictions at high temperatures. Especially if also suboptimal foldings, cf. [21], are sought for, or if calculating the partition functions as in [9], the cutoff size – if used at all – should be set somewhat higher.

Another observation is that the energy parameters estimated for higher temperatures by extrapolation of parameters experimentally determined at lower temperatures do not seem to allow for a prediction of the long range base pairings. One reason for this might be that structures at higher temperatures tend to have more unpaired bases in multibranched loops. The effect of the number of unpaired bases on the stability of multibranched loops should theoretically be logarithmic but are modelled by a linear function for reasons of computational efficiency. This might be acceptable for multibranched loops with only a few unpaired bases but becomes prohibitive as the number of unpaired bases grows.

Finally it should be mentioned that current methods for energy based RNA secondary structure prediction only consider structures that do not contain pseudo knots. Probably *the* open question of RNA secondary structure prediction is to put forth a model including pseudo knots that allows fair predictions within reasonable resources. Currently known methods suffer from either being too time- and space-consuming (time $O(n^6)$ and space $O(n^4)$ for the method presented in [13] and time $O(n^5)$ and space $O(n^3)$ for a restricted class of pseudo knots presented in [8]) or shifting the focus from stability of loops back to stability of pairs, cf. [14].

7 Acknowledgements

The authors would like to thank Darrin Stewart for his help with generating the figures for this article. This work was supported, in part, by NIGMS grant GM54250 to MZ.

References

[1] David Eppstein, Zvi Galil, and Raffaele Giancarlo. Speeding up dynamic programming. In *Proc. 29th Symp. Foundations of Computer Science*, pages 488–496. Assoc. Comput. Mach., October 1988.

[2] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162:705–708, 1982.

[3] R. R. Gutell. http://pundit.icmb.utexas.edu/. RNA secondary structures.

[4] R. R. Gutell, M. W. Gray, and M. N. Schnare. A compilation of large subunit (23S and 23S-like) ribosomal RNA structures. *Nucleic Acids Res.*, 21:3055–3074, 1993. Database issue.

[5] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie*, (125):167–188, 1994.

[6] Karen Hopkin. When RNA ruled – another lost world? *HMS Beagle, The BioMedNet Magazine*, 27, March 1998. http://biomednet.com/hmsbeagle/1998/27/ resnews/meeting.htm.

[7] Ann B. Jacobson. Secondary structure of coliphage Qβ RNA. *Journal of Molecular Biology*, 221:557–570, 1991.

[8] Rune Lyngsø. Computational aspects of biological sequences and structures. Master's thesis, University of Aarhus, Department of Computer Science, DK-8000 Århus C, Denmark, May 1997.

[9] John S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.

[10] Ruth Nussinov and Ann B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. USA*, 77(11):6309–6313, nov 1980.

[11] Chaterine Papanicolaou, Manolo Gouy, and Jacques Ninio. An energy model that predicts the correct folding of both the tRNA and the 5S RNA molecules. *Nucleic Acids Research*, 12:31–44, 1984.

[12] Adam E. Peritz, Ryszard Kierzek, Naoki Sugimoto, and Douglas H. Turner. Thermodynamic study of internal loops in oligoribonucleotides: symmetric loops are more stable than asymmetric loops. *Biochemistry*, 30:6428–6436, 1991.

[13] Elena Rivas and Sean Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, 1999.

[14] Jack E. Tabaska, Robert B. Cary, Harold N. Gabow, and Gary D. Stormo. An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics*, 14(8):691–699, September 1998.

[15] Ignacio Tinoco, Philip N. Borer, Barbara Dengler, Mark D. Levine, Olke C. Uhlenbeck, Donald M. Crothers, and Jay Gralla. Improved estimation of secondary structure in ribonucleic acids. *Nature New Biology*, 246:40–41, 1973.

[16] Ignacio Tinoco, Olke C. Uhlenbeck, and Mark D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[17] D. H. Turner, N. Sugimoto, and S. M. Freier. RNA structure prediction. *Ann. Rev. Biophys. Biophys. Chem*, 17:167–192, 1988.

[18] Michael S. Waterman and T. F. Smith. Rapid dynamic programming methods for RNA secondary structure. *Advances in Applied Mathematics*, 7:455–464, 1986.

[19] W. Zillig, I. Holz, D. Janekovic, W. Schäfer, and W. D. Reiter. The archaebacterium *Thermococcus celer* represents, a novel genus within the thermophilic branch of archaebacteria. *Systematic and Applied Microbiology*, 4:88–94, 1983.

[20] Michael Zuker. http://www.ibc.wustl.edu/~zuker/rna/energy/node2.html. RNA web-page.

[21] Michael Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

[22] Michael Zuker and David Sankoff. RNA secondary structures and their prediction. *Bull. Mathematical Biology*, 46:591–621, 1984.

[23] Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9:133–148, 1981.

Recent BRICS Report Series Publications

RS-99-15 **Rune B. Lyngsø, Michael Zuker, and Christian N. S. Pedersen.** ***An Improved Algorithm for RNA Secondary Structure Prediction*****. May 1999. 24 pp. An alloy of two articles appearing in Istrail, Pevzner and Waterman, editors,** ***Third Annual International Conference on Computational Molecular Biology*****, RECOMB 99 Proceedings, 1999, pages 260–267, and** ***Bioinformatics*****, 15, 1999.**

RS-99-14 **Marcelo P. Fiore, Gian Luca Cattani, and Glynn Winskel.** ***Weak Bisimulation and Open Maps*****. May 1999. To appear in Longo, editor,** ***Fourteenth Annual IEEE Symposium on Logic in Computer Science*****, LICS '99 Proceedings, 1999.**

RS-99-13 **Rasmus Pagh.** ***Hash and Displace: Efficient Evaluation of Minimal Perfect Hash Functions*****. May 1999. 11 pp. A short version to appear in** ***Algorithms and Data Structures: 6th International Workshop*****, WADS '99 Proceedings, LNCS, 1999.**

RS-99-12 **Gerth Stølting Brodal, Rune B. Lyngsø, Christian N. S. Pedersen, and Jens Stoye.** ***Finding Maximal Pairs with Bounded Gap*****. April 1999. 31 pp. To appear in** ***Combinatorial Pattern Matching: 10th Annual Symposium*****, CPM '99 Proceedings, LNCS, 1999.**

RS-99-11 **Ulrich Kohlenbach.** ***On the Uniform Weak König's Lemma*****. March 1999. 13 pp.**

RS-99-10 **Jon G. Riecke and Anders B. Sandholm.** ***A Relational Account of Call-by-Value Sequentiality*****. March 1999. 51 pp. To appear in** ***Information and Computation*****, LICS '97 Special Issue. Extended version of an article appearing in** ***Twelfth Annual IEEE Symposium on Logic in Computer Science*****, LICS '97 Proceedings, 1997, pages 258–267. This report supersedes the earlier report BRICS RS-97-41.**

RS-99-9 **Claus Brabrand, Anders Møller, Anders B. Sandholm, and Michael I. Schwartzbach.** ***A Runtime System for Interactive Web Services*****. March 1999. 21 pp. Appears in Mendelzon, editor,** ***Eighth International World Wide Web Conference*****, WWW8 Proceedings, 1999, pages 313–323 and** ***Computer Networks*****, 31:1391–1401, 1999.**