RNA Pseudoknot Prediction in Energy-Based Models

Article *in* Journal of Computational Biology: a Journal of Computational Molecular Cell Biology · February 2000

DOI: 10.1089/10665270750050862 · Source: PubMed

2 authors, including:

Rune Lyngsø

75 PUBLICATIONS **2,286** CITATIONS

SEE PROFILE

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 7, Numbers 3/4, 2000
Mary Ann Liebert, Inc.
Pp. 409–427

RNA Pseudoknot Prediction in Energy-Based Models

RUNE B. LYNGSØ[1] and CHRISTIAN N.S. PEDERSEN[2]

ABSTRACT

RNA molecules are sequences of nucleotides that serve as more than mere intermediaries between DNA and proteins, e.g., as catalytic molecules. Computational prediction of RNA secondary structure is among the few structure prediction problems that can be solved satisfactorily in polynomial time. Most work has been done to predict structures that do not contain pseudoknots. Allowing pseudoknots introduces modeling and computational problems. In this paper we consider the problem of predicting RNA secondary structures with pseudoknots based on free energy minimization. We first give a brief comparison of energy-based methods for predicting RNA secondary structures with pseudoknots. We then prove that the general problem of predicting RNA secondary structures containing pseudoknots is NP complete for a large class of reasonable models of pseudoknots.

Key words: RNA folding, pseudoknots, energy models, exact algorithms, NP completeness.

1. INTRODUCTION

An RNA molecule is a sequence of nucleotides that often is just an intermediary between DNA and proteins. Some RNA molecules do, however, have vital importance, e.g., in translation of mRNA to proteins. The three-dimensional structure of an RNA molecule is to a large extent determined by interactions between pairs of nucleotides, called base pairings. The secondary structure of an RNA molecule is the set of base pairings in the three-dimensional structure of the molecule. The secondary structure can thus be used in its own right to look for information, e.g., active sites, or as a stepping stone towards prediction of higher structural levels.

If the three-dimensional, or tertiary, structure of an RNA molecule is available, it is of course easy to determine the secondary structure, but determining the tertiary structure is a complicated and time-consuming task. When the tertiary structure of an RNA molecule is not available, the authoritative way of determining the secondary structure of an RNA molecule is by comparative modeling: Given a number of related RNA sequences, the common secondary structure is inferred by identifying compensatory mutations, that is, by identifying pairs of positions where mutations of the base in one of the positions is accompanied by a mutation of the base in the other position to retain their base-pairing capability. The drawback of this technique is that it requires several related RNA sequences to be available. Moreover, since expert intervention is often necessary to identify the compensatory mutations, it is difficult to automate comparative modeling fully. Computational methods for predicting the secondary structure of an RNA sequence are thus in demand.

[1]Baskin Center for Computer Science and Engineering, University of California, Santa Cruz, CA 95064.
[2]Basic Research In Computer Science (BRICS), Department of Computer Science, University of Aarhus, DK8000 Århus, Denmark.

To construct such methods, it is necessary to model the biological reality that governs structure formation. Inspired by the laws of thermodynamics, this is often done in terms of energy minimization. Using a model that describes how to assign free energies to legal secondary structures, the secondary structure of an RNA sequence is predicted as the structure of least free energy. The biological relevance of the predicted structure and the computational resources, such as time and space that are needed to compute it, depend entirely on the choice of legal structures and free energies. Most work has been devoted to construct algorithms for RNA secondary structure prediction when the legal structures are limited to secondary structures that do not contain pseudoknots, that is, do not contain overlapping base pairs. In Nussinov *et al.*, (1978), an algorithm using a simple free-energy function that is minimized when the secondary structure contains the maximum number of complementary base pairs is presented. The algorithm takes time $O(|s|^3)$ for predicting the secondary structure of an RNA sequence s. A more complex model for the free energy of secondary structures is proposed in Tinoco *et al.* (1973). This model states that the free energy of a secondary structure is the sum of independent energies for each loop in the structure. Based on this model of free energy, Nussinov *et al.* (1980) and Zuker *et al.* (1981) present algorithms that take time $O(|s|^3)$ for predicting the secondary structure of an RNA sequence s. Since the ideas of these algorithms form the basis of the widely used `mfold` server (Zuker, 2000) for RNA secondary structure prediction, they are commonly referred to as `mfold` algorithms, or algorithms of the `mfold` type.

The reason that legal structures are often required to not contain pseudoknots is not that pseudoknots do not occur in real world structures, but rather because of modeling and computational considerations. It is still an open question how to construct a reasonable model of free energy for structures containing pseudoknots that also makes it possible to construct efficient structure prediction algorithms. Rivas *et al.* (1999) and Uemura *et al.* (1999) report on successful experiments of energy-based predictions of structures containing certain kinds pseudoknots for short RNA sequences. In Rivas *et al.* (1999) energy parameters for pseudoknots are, in part, derived from energy parameters for similar structural elements not containing pseudoknots, while Uemura *et al.* (1999) simply assign an energy of zero to all structural elements of pseudoknots except stacking base pairs. It is thus likely that the success of these experiments is due to the fact that in short sequences there are not many competing long, energetically favorable helices as much as it is to the actual energy parameters chosen.

In this paper, we study further the problem of predicting RNA secondary structure containing pseudoknots. In the next section, we briefly review the ideas of the `mfold` algorithms. In section 3 we present and compare the algorithms and pseudoknot models of Rivas *et al.* (1999), Uemura *et al.* (1999), and Akutsu (to appear). In the fourth section, we show that predicting RNA secondary structures containing pseudoknots of arbitrary types is **NP** complete for a large class of reasonable free-energy functions. Finally, in the fifth section, we discuss the implications of the **NP** completeness result and compare it to the **NP** completeness result of Akutsu (to appear).

2. TERMINOLOGY

For an RNA sequence s a secondary structure is a set S of base pairs $i \cdot j$ with $1 \leq i < j \leq |s|$, such that $\forall i \cdot j, i' \cdot j' \in S : i = i' \Leftrightarrow j = j'$. Each base can thus take part in at most one base pair. The base pairs of a secondary structure describe the base pairing interactions formed by hydrogen bonding in a corresponding tertiary structure. It is usually assumed that RNA secondary structures do not contain pseudoknots. Two base pairs form a pseudoknot if they are overlapping, i.e., two base pairs $i \cdot j, i' \cdot j' \in S$ form a pseudoknot if $i < i' < j < j'$. The term "pseudoknot" is also used as a shorthand for overlapping structures other than base pairs, e.g., two helices of stacking base pairs, when the base pairs of these structures form pseudoknots.

There are of course good reasons for introducing this restriction, prominent among which is a simplification of legal structures. The simplification of not allowing pseudoknots ensures that two base pairs $i \cdot j, i' \cdot j' \in S$ are either nested, i.e., $i < i' < j' < j$, or disjoint, i.e., $i < j < i' < j'$. In many situations this allows us to handle first one base pair and then the other (if they are nested) or handle them independently (if they are disjoint). The pseudoknot restriction is thus crucial in algorithms, e.g., structure prediction (Eddy *et al.*, 1994; Knudsen *et al.*, 1999; Nussinov *et al.*, 1980; Sakakibara *et al.*, 1994; Zuker *et al.*, 1981), partition function computations (McCaskill, 1990), comparing secondary structures (Zhang

et al., 1989), and simultaneous alignment and structure prediction of RNA sequences (Gorodkin *et al.*, 1997; Sankoff, 1985). In the following, we will exemplify this by giving a brief summary of an algorithm of the `mfold` type for secondary structure prediction. The summary is also aimed at introducing the terminology we will use in the third section. A more detailed summary can be found in Turner *et al.* (1988).

An `mfold` algorithm predicts secondary structures without pseudoknots by computing minimum (or close to minimum) energy structures in the model proposed in Tinoco *et al.* (1973). By this model the free energy of a secondary structure S is the sum of independent energy contributions from each of the structural elements, or *loops*, of S. Each loop of S can be identified with a base pair that is called the *exterior*, or closing, base pair of that loop. The loop consists of all bases *accessible* from the exterior base pair and the exterior base pair itself. For a base pair $i \cdot j \in S$, a base k is accessible from $i \cdot j$ if $i < k < j$ and there are no base pairs $i' \cdot j' \in S$ such that $i < i' < k < j' < j$. More informally the accessible bases are the bases we can get to without having to cross other base pairs. For structures without pseudoknots one can observe that a base can be accessible from at most one base pair (if it is not accessible from any base pairs it is called an external base) and that if one base of a base pair is accessible from another base pair then so is the other. Base pairs accessible from the exterior base pair of a loop are called *interior* base pairs of the loop and the type of a loop depends on the number of interior base pairs in the loop. A loop with no interior base pairs is called a *hairpin* loop. With one interior base pair the type further depends on the relative positions of the exterior base pair $i \cdot j$ and the interior base pair $i' \cdot j'$. If $i' = i + 1$ and $j' = j - 1$ the two base pairs are said to *stack*. If $i' = i + 1$ or $j' = j - 1$ but not both, the loop is called a *bulge*. Otherwise the loop is called an *interior* loop. With more than one interior base pair the loop is called a *multibranched* loop. A segment of consecutive stacking base pairs, i.e., a subset $\{i + l \cdot j - l\}_{0 \le l < k}$ of k base pairs of S, is called a *helix*. The term "helix" is also occasionally used for a number of helices only separated by bulges and interior loops, but not by multibranched loops. These structures appear in Figure 1.

The Tinoco model (Tinoco *et al.*, 1973) is the basic scaffold for `mfold` algorithms, describing the recursive decomposition of a structure into independent loops. To make the problem of computing the structure of minimum energy tractable, a further simplifying assumption about the nature of the energy function for multibranched loops is needed. By this extra assumption the energy of a multibranched loop is an affine function in the number of unpaired bases and the number of interior base pairs of the loop, i.e., the energy of a multibranched loop with k interior base pairs and k' unpaired bases is $eM(k, k') = a + bk' + ck$. It should be noted that current energy functions (see Mathews *et al.* [1999] for the most recently published energy parameters) also assign stacking contributions to the base pairs of a multibranched loop. These contributions only depend on the local neighborhood of the base pairs, however, and can be handled without increasing the resource requirements of algorithms of the `mfold` type.

We are now ready to specify a basic `mfold` algorithm. Three arrays, $V(i, j)$ holding the minimum energy of a secondary structure on $s[i \,..\, j]$ with bases i and j forming a base pair, $WM(i, j)$ holding

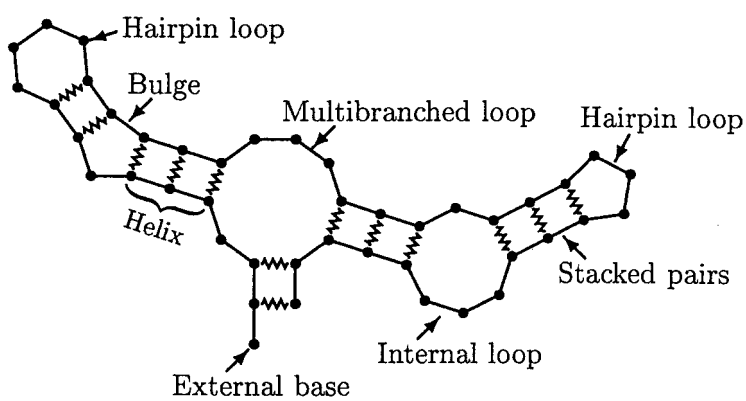


FIG. 1. An example RNA secondary structure illustrating the different types of structural elements, or loops, of pseudoknot free structures. Bases are depicted by circles, the RNA backbone by straight lines, and base pairings by zigzagged lines.

the minimum energy of a structure on $s[i \,..\, j]$ that is part of a multibranched loop, and $W(i, j)$ holding the minimum energy of a structure on $s[i \,..\, j]$, are computed based on the recursions

$$\begin{aligned} V(i, j) = \min \Big\{ & eH(i, j), eS(i, j, i+1, j-1) + V(i+1, j-1), \\ & \min_{\substack{i<i'<j'<j \\ i'-i+j-j'>2}} \{eL(i, j, i', j') + V(i', j')\}, \\ & \min_{i+1<k<j} \{WM(i+1, k-1) + WM(k, j-1) + a\} \Big\}, \end{aligned} \tag{1}$$

$$\begin{aligned} WM(i, j) = \min \Big\{ & V(i, j) + b, WM(i, j-1) + c, WM(i+1, j) + c, \\ & \min_{i<k\leq j} \{WM(i, k-1) + WM(k, j)\} \Big\}, \end{aligned} \tag{2}$$

$$W(i, j) = \min \Big\{ W(i+1, j), W(i, j-1), \min_{i\leq k<j} \{W(i, k) + V(k+1, j)\} \Big\}. \tag{3}$$

These recursions employ energy functions for hairpin loops (eH), stacking base pairs (eS), internal loops and bulges (eL), and multibranched loops (eM as described above). With the currently used parameters for the energy functions, these recursions allow for an $O(|s|^3)$ time algorithm, (Lyngsø *et al.*, 1999; Turner *et al.*, 1988) for computing secondary structures of minimum energy for an RNA sequence s.

The only part of the above recursions where the time requirements of $O(|s|^3)$ might not be obvious is the part pertaining to internal loops and bulges. Here we need to find the minimum of $eL(i, j, i', j') + V(i', j')$ for all $i < i' < j' < j$ with $i' - i + j - j' > 2$. If the function $eL(i, j, i', j')$ is a sum of contributions from the size of the loop, i.e., $i' - i + j - j'$, the lopsidedness of the loop, i.e., $|(i' - i) - (j - j')|$, and contributions from stacking effects at the two base pairs, i.e., functions that only depends on i, j and on i', j' respectively, it is shown in Lyngsø *et al.* (1999) how to handle all interior loops and bulges in time $O(|s|^3)$. This is done by observing that the best choice of interior base pair for interior loops of size k closed by the base pair $i \cdot j$ is either the same as for interior loops of size $k - 2$ closed by the base pair $i + 1 \cdot j - 1$ or one of the two base pairs yielding a loop of maximum lopsidedness (i.e., a bulge). If the assumption about the energy function only holds for $\min\{i' - i, j - j'\} \geq c$, the method can still be applied with an increase in time requirements by a factor of c, i.e., to a time complexity of $O(c|s|^3)$. Energy functions in current use assign a special energy function to bulges, but for interior loops the energy function is a sum of stacking effects at the interior and exterior base pairs, a size-dependent term, and a Ninio type asymmetry penalty, cf. Papanicolaou *et al.* (1984). Asymmetry penalties of the Ninio type only depend on lopsidedness for $\min\{i' - i, j - j'\} \geq c$, where $c = 2$ for currently used energy parameters.

3. ENERGY-BASED PSEUDOKNOT PREDICTION

The Tinoco model (Tinoco *et al.*, 1973) describes how to assign energies to secondary structures not containing pseudoknots, but does not address how to handle secondary structures containing pseudoknots. There is no authoritative model generally agreed on for assigning energies to structures containing pseudoknots. Still, for a model to qualify as an energy model we would expect it to reduce to something similar to the Tinoco model when restricted to structures not containing pseudoknots, i.e., allowing the energy contribution of a base pair to depend on nearby base pairs and stretches of consecutive unpaired bases to have a length dependent energy contribution. Methods relying on the contributions from each base pair being independent, e.g., like the maximum weighted matching approach of Tabaska *et al.* (1998), will thus not be considered here. Furthermore, we will only discuss rigorous (and polynomial time) algorithms. The reason for this is twofold. First, the **NP** completeness result presented in the fourth section only have direct implications for attempts to develop algorithms guaranteed to find a structure of minimum energy. Secondly, for nonrigorous algorithms the set of structures minimized over is determined by heuristics, e.g., depending on the actual RNA sequence as in Brown *et al.* (1996) or random choices as in van Batenburg *et al.* (1995), making it difficult, if not impossible, to compare classes of structures considered by different algorithms.

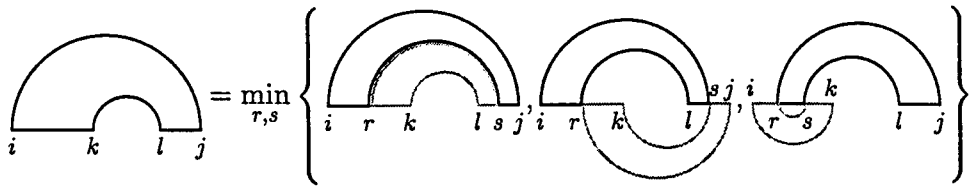


FIG. 2. General recursion scheme for the RNA secondary structure prediction algorithm in Rivas *et al.* (1999).

To our knowledge, four algorithms have been published for rigorous, energy-based, polynomial time prediction of RNA secondary structures including some class of pseudoknots. We will briefly describe and compare—in terms of resource requirements and generality of the class of structures minimized over—the algorithms of Rivas *et al.* (1999), Uemura *et al.* (1999), and Akutsu (to appear). The algorithm of Lyngsø *et al.* (2000) has the same resource requirements as that of Akutsu (to appear) while minimizing over a smaller set of structures and will thus be ignored.

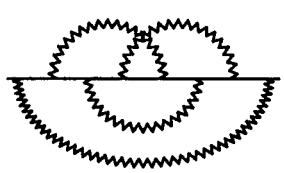
In Figure 2 we briefly sketch the idea of the algorithm presented in Rivas *et al.* (1999). Arrays holding energies of optimal structures for the subsequence from i through j are maintained similar to Equations 1 to 3, but with the further restriction that the bases from k through l are yet unpaired (to allow for future pseudoknot interactions). The general recursion scheme for an entry in one of these matrices is to minimize over all possible ways of splitting the subsequence with an unpaired region into two new subsequences with unpaired regions. The requirements of time $O(|s|^6)$ and space $O(|s|^4)$ for this algorithm are observations that follow directly from Figure 2.

The pseudoknot model, i.e., the class of structures the algorithm minimizes over, in Rivas *et al.* (1999) is only defined implicitly by the specification of the algorithm. The same holds for the pseudoknot model in Uemura *et al.* (1999). We have therefore chosen to compare the generality of the pseudoknot models of the three papers by presenting some structures exemplifying differences between the models. The pseudoknot model in Rivas *et al.* (1999) allows all structures that can be recursively built up by the scheme in Figure 2. This allows for rather complex structures, even some nonplanar structures, cf. Figure 3(a). By a structure being planar, we mean that the graph defined by the base pairs of the structure and the backbone connections of adjacent bases is planar. On the other hand, not all planar structures are included in this model as illustrated by Figure 3(b).

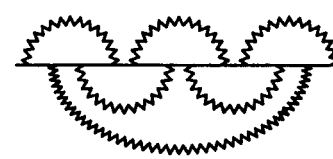
One type of structure that is particularly useful for comparing the pseudoknot models of the three papers is what we will denote a chain of pseudoknots, cf. Figure 4. A set $C = \{i_l \cdot j_l\}_{1 \le l \le k}$ of base pairs is a chain of length k if

- $i_{l+1} < j_l$ for $1 \le l < k$.
- $j_l < i_{l+2}$ for $1 \le l < k - 1$.
- $i_1 < i_2$ and $j_{k-1} < j_k$.

Single base pairs that do not stack with neighboring base pairs usually have an unfavorable energy contribution. Chains might thus seem uninteresting structures. For the pseudoknot models of the three algorithms considered here, however, if a chain belongs to one of the models, so does any structure that can be

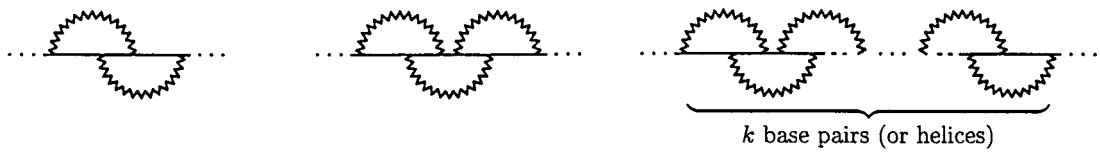


(a) A non-planar structure in the class of structures minimized over.



(b) A planar structure not in the class of structures minimized over.

FIG. 3. Structures exemplifying the class of structures the algorithm in Rivas *et al.* (1999) minimizes over. The RNA backbone is drawn as a straight line and base pairs (or more complex structures, e.g. helices, forming the base pairing interactions) are drawn as zigzagged semi-circles.



(a) A chain of length two. (b) A chain of length three. (c) A chain of length k.

FIG. 4. Chains of pseudoknots. Two base pairs (or helices) forming a pseudoknot is a chain of length two. If the last base pair additionally forms a pseudoknot with a base pair disjoint to the first base pair we get a chain of length three. This can be generalized to chains of pseudoknots of any length.

obtained by replacing the base pairs with structures that do not themselves contain pseudoknots. So the chains can be viewed as simple representatives of classes of more complex structures. With this said, the pseudoknot model in Rivas *et al.* (1999) allows chains of any length. We can start from one end of the chain adding the next base pair in the chain until we reach the other end. At any time, we only need to keep one region free for future base pairings, namely, the region between the rightmost bases of the two most recently added base pairs.

The close correspondence between context-free grammars and RNA secondary structures without pseudoknots is well known (Sakakibara *et al.*, 1994; Searls, 1992). In Uemura *et al.* (1999), tree-adjoining grammars are proposed as a tool for modeling RNA secondary structures with pseudoknots. Tree-adjoining grammars generate trees by repeatedly replacing internal nodes of the current tree with a tree according to the rules of the grammar, similar to the way a context-free grammar generates strings by replacing a nonterminal with a string in the current string. The tree being adjoined has a special leaf, called the foot node, where the subtree previously rooted at the node being replaced is moved to (cf. Figure 5). The path from the root to the foot node of a tree being adjoined is called the backbone of that tree. The string of a tree generated by a tree-adjoining grammar is the concatenation of the labels of the leaves in left-to-right order. To make the problem of parsing a given RNA sequence tractable, the grammars considered in Uemura *et al.* (1999) are restricted to be what they denote as extended simple linear tree-adjoining grammars. A dynamic programming algorithm to parse strings according to such grammars in time $O(|s|^5)$ and space $O(|s|^4)$ is presented. Thus they obtain a speedup by a factor of $|s|$ compared to the algorithm presented in Rivas *et al.* (1999) but still use a similar amount of space.

The pseudoknot model of Uemura *et al.* (1999) is only specified in terms of the tree-adjoining grammar. Tree-adjoining grammars are not the easiest things to comprehend, but at least we can make some qualitative statements about the model by looking at the trees that can be adjoined and the way they introduce base pairs. These trees have at most two leaves labeled with bases. A base introduced to the string of the tree by adjoining a tree with only one leaf labeled with a base is unpaired, while two bases introduced by the

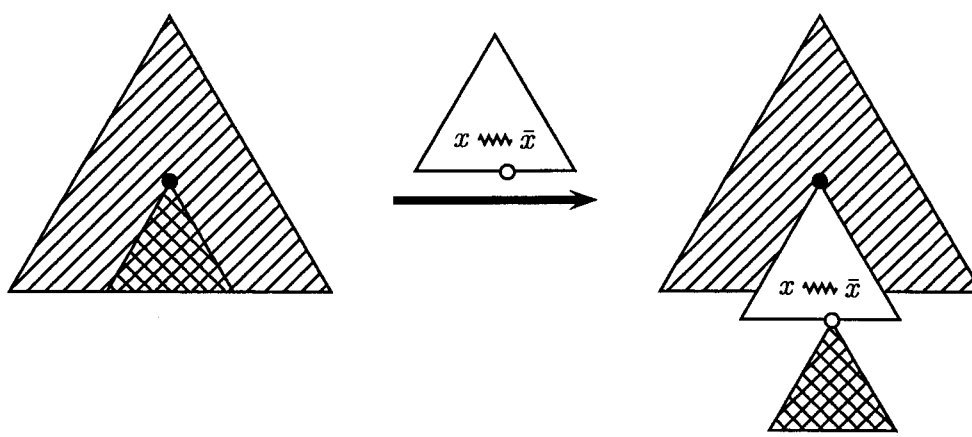


FIG. 5. General idea of the tree adjoining grammar description of RNA secondary structures used in Uemura *et al.* (1999). Base pairs are introduced by adjoining trees with two complementary bases as leaf labels. The foot node is represented by a hollow circle in the tree being adjoined.

same tree form a base pair (cf. Figure 5). It is thus the trees with two leaves labeled with bases that are interesting. There are basically three types of such trees:

- Trees where the two leaves labeled with bases are both to the left of the backbone of the tree but descendants of different internal nodes. We will call these *left-trees.*
- Trees where the two leaves labeled with bases are both to the right of the backbone of the tree but descendants of different internal nodes. We will call these *right-trees.*
- Trees where the two leaves labeled with bases are on different sides of the backbone but descendants of the same internal node. We will call these *center-trees.*

Two bases introduced by the same left-tree can eventually be separated by an arbitrarily long path as we can adjoin new trees to internal nodes on the path between them. We can, however, never form a pseudoknot with two base pairs both introduced by a left-tree. If we adjoin a left-tree somewhere on the path between two bases introduced by another left-tree, the new base pair will be nested inside the other. If we adjoin it somewhere else, the two base pairs will be disjoint. The same holds for two base pairs both introduced by right-trees. Furthermore, by similar reasoning, one can observe that two base pairs both introduced by center-trees or two base pairs introduced by, respectively, a left-tree and a right-tree will also always be either nested or disjoint. This shows that the pseudoknot model of Uemura *et al.* (1999) only allows planar structures (draw base pairs from left- and right-trees on one side of the backbone and base pairs from center-trees on the other).

By the preceding observations, all pseudoknots in the model of Uemura *et al.* (1999) are thus formed between a base pair introduced by a center-tree with a base pair introduced by either a left- or a right-tree. One can further observe that the leftmost base in a pseudoknot will be from the base pair introduced by the left-tree when the pseudoknot is formed between base pairs introduced by a left-tree and a center-tree and will be from the base pair introduced by the center-tree when the pseudoknot is formed between base pairs introduced by a center-tree and a right-tree. By this observation, the pseudoknot model does not allow chains of pseudoknots of length more than three (but it does allow chains of length three). The speedup of the algorithm in Uemura *et al.* (1999) compared to the algorithm in Rivas *et al.* (1999) is thus at the cost of some generality of the pseudoknot model (it could, of course, be possible that the two models were incomparable, i.e., neither being a subset of the other. An inspection of the parsing algorithm in Uemura *et al.* (1999) does, however, show that it is basically a restricted version of the algorithm in Rivas *et al.* (1999), using a similar four dimensional array but simpler combination rules.)

The pseudoknot model defined in Akutsu (to appear) can best be illustrated as folding the affected part of the backbone into an S-like shape. Simple pseudoknots are then formed by base pairings of a base in the middle stem and a base in one of the outer stems. These base pairs are not allowed to cross, i.e., any two base pairs formed between bases from the middle stem and the same outer stem do not themselves form a pseudoknot. For the more complex recursive pseudoknots, base pairs between two bases in the same stem are also made possible by allowing any substring of a stem to be assigned a structure that is itself a recursive pseudoknot.

This definition allows a dynamic programming solution to find the structure of minimum energy. The basic recurrence element in this solution is the minimum energy of a structure involving the bases between i_0 and i in one of the outer stems, and the bases between j and k in the other outer stem and the middle stem (cf. Figure 6). This value is computed by minimizing over adding either a base pair between the middle stem and one of the outer stems or adding a minimum-energy structure to either one of the stems (cf. Figure 6). The time requirement of $O(|s|^5)$ is immediately deduced from the recursion scheme. As for the space requirement, Akutsu (to appear) observes that a recurrence element can only depend directly on another recurrence element if they both have the same leftmost base, i.e., the same value of i_0 (there can be

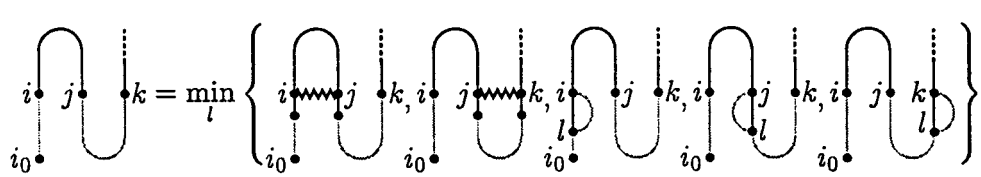


FIG. 6. General recurrence scheme for the algorithm presented in (Akutsu, to appear). The grey areas have already been assigned structure while the black areas are still free for adding extra base pairs.

an indirect dependency through a minimum energy structure added to a stem, but this dependence will be through an array holding the minimum energies of a structure on an entire substring of the RNA sequence, i.e., an array similar to the W array of Equation 3). The recurrences can thus be computed for a fixed i_0 at a time limiting the space requirements to $O(|s|^3)$.

Akutsu (to appear) basically uses the same recurrence element as Rivas *et al.* (1999), the minimum energy of a structure on a substring of the RNA sequence with a region restricted yet to be unpaired. The recurrence relations of Akutsu (to appear) are, however, more restricted. So the pseudoknot model of Akutsu cannot include structures not included in the pseudoknot model of Rivas *et al.* (1999). That there are some structures it does not include follows from the fact that it can be proven that it is at least as restricted as the pseudoknot model of Uemura *et al.* (1999). It can be simulated by the tree adjoining grammar of Uemura *et al.* (1999), basically using left-trees for introducing base pairings between bases in the middle stem and the leftmost stem and center-trees for introducing base pairings between bases in the middle stem and the rightmost stem. Furthermore, it can be observed that the pseudoknot model of Akutsu (to appear) only allows chains of length two. Thus it is actually even more restricted than the pseudoknot model of Uemura *et al.* (1999). The pseudoknot models of Rivas *et al.* (1999), Uemura *et al.* (1999), and Akutsu (to appear) thus exhibit a decreasing order of generality. The decreasing resource requirements of the presented algorithms to find structures of minimum energy in the three pseudoknot models should thus be of no surprise.

Before concluding this section, it should be mentioned that Akutsu (to appear) also presents an algorithm for structure prediction in the restricted model of simple pseudoknots that requires time $O(|s|^4)$ and space $O(|s|^2)$ (in Akutsu [to appear] the space requirements are stated as $O(|s|^3)$, but they can be reduced to $O(|s|^2)$ by a trick similar to reducing the space requirements of computing an alignment of two sequences by filling out the table column by column; Furthermore, if the method of, e.g., Hirschberg (1975) is used in combination with this trick, the time requirements of the traceback to determine a structure with the computed minimum energy can be reduced from $O(|s|^4)$ to $O(|s|^3)$). This algorithm is very similar to the algorithm for recursive pseudoknots, using the same basic recurrence element but simpler recursions. In the presented version, the algorithm assumes that energies are assigned only to stacking base pairs, but it can easily be modified to handle a pseudoknot energy function that is affine in the number of unpaired bases and helices of the pseudoknot, i.e., an energy function similar to the eM energy function for multibranched loops. Despite the limitations of simple pseudoknots, this does allow prediction of interesting structures, e.g., structures with an unlimited number of H type pseudoknots (cf. Figure 7), within resource bounds that are close to being reasonable. Furthermore, though the algorithm cannot easily be modified to allow adding a minimum energy structure to one of the three stems of the S-shaped recurrence element (cf. Figure 6), without increasing the resource requirements, it can be modified to allow connecting two stems with a minimum energy structure. This will make pseudoknot interactions between regions far apart in the sequence possible while still allowing a rich structure in the intervening regions.

The game of modifying models and algorithms to obtain the best possible combination of a fast algorithm and broad class of legal secondary structures could be perpetuated. But for any class of secondary structures with pseudoknots we should probably not expect to do better than the requirements of time $O(|s|^3)$ and space $O(|s|^2)$ of the classic `mfold` algorithm. Furthermore, in the following section, we provide evidence that we should not set hopes too high for developing efficient algorithms handling secondary structures with general types of pseudoknots.

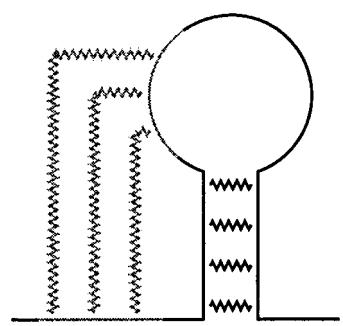


FIG. 7. A pseudoknot of type H, cf. (Pleij, 1993, Figure 1). Zigzagged lines indicate base pairings.

4. COMPLEXITY RESULTS

In this section we prove that RNA secondary structure prediction with pseudoknots is **NP** complete in a simple nearest neighbor model, cf. definition 1. This model might seem too simple, and probably would be if we wanted to base a secondary structure prediction algorithm on it. But when proving complexity results, we want to do so in a model that is as simple as possible. If the problem in the simple model is **NP** complete, it will remain so in any more complex and realistic model as long as fixing some of the parameters in the complex model turns it into the simple model. Akutsu (to appear) also provides an **NP** completeness proof for predicting RNA secondary structures with pseudoknots but in a somewhat different model. We discuss the differences between the two models in the next section.

Definition 1 (The Nearest-Neighbor Pseudoknot Model). *Let $\mathcal{S}$ be a secondary structure on a sequence $s \in \{A, C, G, U\}^*$, with $|s| = n$, that is, $\mathcal{S}$ is a set of base pairs $i \cdot j$ where $1 \leq i < j \leq n$ and $\forall i \cdot j, i' \cdot j' \in \mathcal{S} : i = i' \Leftrightarrow j = j'$. The energy of $\mathcal{S}$ is an independent sum of energies of each of the base pairs in $\mathcal{S}$,*

$$E(\mathcal{S}) = \sum_{i \cdot j \in \mathcal{S}} E(i \cdot j, i+1, j-1),$$

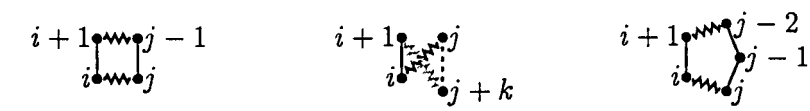
where the energy of a base pair $i \cdot j$ only depends on

- *the base pair itself, that is, the types of bases forming the pair.*
- *the two neighboring bases $i+1$ and $j-1$, that is, the types of these two bases. Furthermore, if $i+1 \cdot j' \in \mathcal{S}$ (or $i' \cdot j-1 \in \mathcal{S}$) the energy can depend on the type of base at position j' (or at position i') and on the position of j' (or of i') relative to i and j.*

The energy of a base pair should be computable in polynomial time.

Note that the Nearest-Neighbor Pseudoknot Model allows arbitrarily complex pseudoknots as there is no restriction that base pairs are not allowed to overlap. The energy of a base pair in the Nearest Neighbor Pseudoknot Model is allowed to depend on nonneighboring bases, but only through a base pairing with a neighboring base. If we compare this to the Tinoco model (Tinoco *et al.*, 1973), the Tinoco model allows the energy of a base pair to depend not only on the neighboring bases and the base pairs they might participate in, but on all bases and base pairs in the loop it closes. If we consider the model assumed by the `mfold` server, this is more restricted than the Tinoco model. Still, it allows the energy of a base pair to depend on the type of loop it closes, the size of the loop, and coaxial stacking of base pairs involving neighboring bases. The Nearest Neighbor Pseudoknot Model can be seen as a further restriction of this where we allow the energy of a base pair to depend only on stacking effects with unpaired neighboring bases and base pairs involving neighboring bases. The value of these stacking effects can, however, depend on whether the involved base pairs form a helix, an ordinary loop (a bulge, an internal loop, or multibranched loop), or a pseudoknot. Specifically, in this section we will use the ability to discern between the three different cases depicted in Figure 8 in the Nearest Neighbor Pseudoknot Model.

If we compare the Nearest Neighbor Pseudoknot Model, e.g., to the energy model used in Rivas *et al.* (1999), it should be of little surprise that the Nearest Neighbor Pseudoknot Model is a restriction of the model used by Rivas *et al.* (1999). The Nearest Neighbor Pseudoknot Model can be obtained from the energy model used by Rivas and Eddy by fixing some of the parameters. Thus an **NP** hardness result for secondary structure prediction in the Nearest Neighbor Pseudoknot Model immediately implies that



(a) A stacking base pair (b) A pseudoknot forming base pair (c) Anything else

FIG. 8. The three cases of neighboring base pairs we need to be able to discern.

secondary structure prediction allowing general pseudoknotted structures in the energy model used by Rivas *et al.* (1999) is **NP** hard.

Proposition 1. *The problem of determining whether an optimal secondary structure in the Nearest Neighbor Pseudoknot Model has energy lower than some energy value E is* **NP** *complete.*

As the problem trivially is in **NP** (guess an optimal secondary structure and verify in polynomial time that it has an energy value lower than E), all we need to do is to prove that it is **NP** hard. We will do this by a reduction to the special case of 3SAT where each literal occurs at most twice (cf. Papadimitriou, 1994, Proposition 9.3). The further restriction in Papadimitriou (1994, Proposition 9.3) that each variable occurs at most three times will not be of importance in the reduction. We will split the reduction into two parts. In the first part we will sketch the basic idea by using an alphabet of unlimited size. Then in the second part we will illustrate how to use the RNA alphabet to emulate an unlimited alphabet.

4.1. Reduction using an unlimited size alphabet

To elucidate the idea of the reduction, we will start by assuming that we have an alphabet of unlimited size at our disposal but otherwise assume the Nearest Neighbor Pseudoknot Model (cf. Definition 1). The symbols used will be separated into pairs of complementary symbols. Only base pairs between complementary symbols will be energetically favorable, and then only if they do not form a pseudoknot with a base pair involving a neighboring base. More precisely, we will use the base pair energy function

$$E(X_i \cdot Y_j, V_{i+1}, W_{j-1}) = \begin{cases} -1 & \text{if } X \text{ and } Y \text{ are complementary symbols and for } j' \notin \{i+1, \ldots j-1\} \\ & \text{we have } V_{i+1} \cdot Z_{j'}, W_{j-1} \cdot Z_{j'}, Z_{j'} \cdot V_{i+1},\ Z_{j'} \cdot W_{j-1} \notin \mathcal{S}. \\ 0 & \text{otherwise.} \end{cases}$$

The notation X_i is used as a shorthand to indicate that the i'th base is of type X.

Given a Boolean formula ϕ in the restricted conjunctive normal form of the restricted 3SAT problem, we need to specify how to construct a sequence s_ϕ such that the energy of an optimal secondary structure of s_ϕ in the Nearest Neighbor Energy Model, using the above energy function, tells whether ϕ is satisfiable. The alphabet we use will consist of

- for each literal l occuring in ϕ, one or two pairs of complementary symbols $(l)_1, \overline{(l)}_1$ and $(l)_2, \overline{(l)}_2$.
- for the i'th clause in ϕ, two pairs of complementary symbols $\mathtt{c}_{i,1}, \overline{\mathtt{c}}_{i,1}$ and $\mathtt{c}_{i,2}, \overline{\mathtt{c}}_{i,2}$.
- for the i'th variable, one pair of complementary symbols $\mathtt{v}_{i,1}, \overline{\mathtt{v}}_{i,1}$.

We will now describe how to construct substrings corresponding to clauses and variables. These substrings will then be combined to construct s_ϕ.

Definition 2. *Let $C_i = l_1 \vee l_2 \vee l_3$ be the i'th clause of ϕ. The* clause substring *$\mathcal{C}_i$ corresponding to C_i is the string*

$$\mathtt{c}_{i,1}(l_1)_{i_1}\overline{\mathtt{c}}_{i,1}\mathtt{c}_{i,2}(l_2)_{i_2}\mathtt{c}_{i,1}\overline{\mathtt{c}}_{i,2}(l_3)_{i_3}\mathtt{c}_{i,2}$$

where $i_j = 1$ if the leftmost occurrence of the literal l_j in ϕ is the occurrence in C_i. Otherwise $i_j = 2$. We will refer to $(l_j)_{i_j}$ (and $\overline{(l_j)}_{i_j}$, (cf. Definition 3) as the symbol corresponding to l_j occuring in C_i and to $\mathtt{c}_{i,1}$, $\overline{\mathtt{c}}_{i,1}$, $\mathtt{c}_{i,2}$, and $\overline{\mathtt{c}}_{i,2}$ as control symbols.

The idea behind this construction is that a base pair formed between the $\overline{\mathtt{c}}_{i,1}$ symbol and one of the $\mathtt{c}_{i,1}$ symbols either to the left or to the right will *screen* either $(l_1)_{i_1}$ or $(l_2)_{i_2}$. By a symbol being screened we mean that it cannot form a base pair with its complementary symbol without forming a pseudoknot with a base pair involving a neighboring symbol. Similarly, a base pair formed between the $\overline{\mathtt{c}}_{i,2}$ symbol and one of the $\mathtt{c}_{i,2}$ symbols either to the left or to the right will screen either $(l_2)_{i_2}$ or $(l_3)_{i_3}$. Furthermore, if the $\overline{\mathtt{c}}_{i,1}$ forms a base pair with the $\mathtt{c}_{i,1}$ symbol to the right and the $\overline{\mathtt{c}}_{i,2}$ symbol forms a base pair with the $\mathtt{c}_{i,2}$ symbol to the left, these two base pairs will form a pseudoknot involving neighboring bases. Thus, the symbols in a clause substring can be involved in at most three energetically favorable base pairs,

one of which must be the symbol corresponding to a literal in the clause forming a base pair with its complementary symbol.

It should be noted that the restricted version of 3SAT does not remain **NP** complete if all clauses are required to have exactly three literals (cf. Papadimitriou, 1994, p. 184). Some clauses may only have two literals. There are several ways to modify Definition 2 to handle this. One simple way to do this is to let $(l_3)_{i_j}$ be a dummy symbol not having a complementary symbol if the clause C only has two symbols.

Definition 3. *Let x_i be a variable occurring twice positively and twice negatively in ϕ. The* variable substring $\mathcal{V}_i$ *corresponding to x_i is the string*

$$\mathtt{v}_i\overline{(x_i)}_2\overline{(x_i)}_1\overline{\mathtt{v}}_i\overline{(\neg x_i)}_2\overline{(\neg x_i)}_1\mathtt{v}_i$$

where $\overline{(x_i)}_j$ is the complementary symbol to the symbol corresponding to the j'th occurrence of the literal x_i in ϕ and $\overline{(\neg x_i)}_j$ is the complementary symbol to the symbol corresponding to the j'th occurrence of the literal $\neg x_i$ in ϕ. We will refer to $\mathtt{v}_i$ and $\overline{\mathtt{v}}_i$ as control symbols.

For a variable x_i appearing positively only once (appearing negatively only once) the variable substring $\mathcal{V}_i$ *corresponding to x_i is the string obtained by removing the symbol $\overline{(x_i)}_2$ (by removing the symbol $\overline{(\neg x_i)}_2$) from the above string.*

The idea behind this construction is that a base pair formed between $\overline{\mathtt{v}}_i$ and the $\mathtt{v}_i$ to either the left or right will screen the symbols corresponding to either the positive or the negative occurrences of x_i in ϕ. Thus, one energetically favorable base pair involving the control symbols in the variable substring can be formed. This base pair will screen the barred symbols corresponding to either all positive or all negative occurrences of x_i. We cannot at the same time form energetically favorable base pairs involving symbols corresponding to both positive and negative occurrences of x_i and form an energetically favorable base pair involving the control symbols in $\mathcal{V}_i$.

Definition 4. *Let ϕ be a Boolean formula on conjunctive normal form where each clause has at most three literals and each literal occurs at most twice. Assume that ϕ consists of c clauses and uses v variables. The sequence corresponding to ϕ is the sequence*

$$s_\phi = \mathcal{C}_1\mathcal{C}_2\ldots\mathcal{C}_c\mathcal{V}_1\mathcal{V}_2\ldots\mathcal{V}_v,$$

where $\mathcal{C}_i$ is the clause substring corresponding to the i'th clause of ϕ and $\mathcal{V}_i$ is the variable substring corresponding to the i'th variable of ϕ.

It should be noted that the ordering of the clauses in the string representation of ϕ and of the symbols corresponding to occurrences of literals in variable substrings is of importance. For any literal l in ϕ, base pairs formed between $(l)_1$ and $\overline{(l)}_1$ and between $(l)_2$ and $\overline{(l)}_2$ will not form a pseudoknot with each other. An example of a Boolean formula and the string representing it is shown in Figure 9. With these definitions we are now ready to prove Proposition 1.

$$\begin{aligned}
\phi &= (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4)\\
s_\phi &= \left.\begin{array}{l}
\mathtt{c}_{1,1}(x_1)_1\overline{\mathtt{c}}_{1,1}\mathtt{c}_{1,2}(x_2)_1\mathtt{c}_{1,1}\overline{\mathtt{c}}_{1,2}(x_3)_1\mathtt{c}_{1,2}\\
\mathtt{c}_{2,1}(\neg x_1)_1\overline{\mathtt{c}}_{2,1}\mathtt{c}_{2,2}(\neg x_2)_1\mathtt{c}_{2,1}\overline{\mathtt{c}}_{2,2}(\neg x_4)_1\mathtt{c}_{2,2}\\
\mathtt{c}_{3,1}(\neg x_1)_2\overline{\mathtt{c}}_{3,1}\mathtt{c}_{3,2}(\neg x_3)_1\mathtt{c}_{3,1}\overline{\mathtt{c}}_{3,2}(x_4)_1\mathtt{c}_{3,2}
\end{array}\right\}\text{Clause substrings}\\
&\quad\left.\begin{array}{l}
\mathtt{v}_1\overline{(x_1)}_1\overline{\mathtt{v}}_1\overline{(\neg x_1)}_2\overline{(\neg x_1)}_1\mathtt{v}_1\\
\mathtt{v}_2\overline{(x_2)}_2\overline{\mathtt{v}}_2\overline{(\neg x_2)}_1\mathtt{v}_2\\
\mathtt{v}_3\overline{(x_3)}_1\overline{\mathtt{v}}_3\overline{(\neg x_3)}_1\mathtt{v}_3\\
\mathtt{v}_4\overline{(x_4)}_1\overline{\mathtt{v}}_4\overline{(\neg x_4)}_1\mathtt{v}_4
\end{array}\right\}\text{Variable substrings}
\end{aligned}$$

FIG. 9. A Boolean formula on the restricted normal form and its corresponding string, cf. definition 4.

Proof of Proposition 1 using an unlimited alphabet. As mentioned, the proof will be by a reduction from 3SAT with the restriction that no literal appears more than twice. So let ϕ be a valid Boolean formula for this restriction of 3SAT with c clauses and v variables. The base pair energy function specified at the beginning of this section and s_ϕ (cf. Definition 4) can both be constructed in polynomial time. We claim that an optimal secondary structure for s_ϕ with the specified energy function has energy $-(3c + v)$ if and only if ϕ is satisfiable; otherwise an optimal secondary structure will have energy larger that $-(3c + v)$. The intuition behind this is illustrated in Figure 10. Base pairs formed between symbols corresponding to literals indicate that the literal should be **true**. Base pairs formed between control symbols in variable substrings prevent a literal and its negation from both being **true** at the same time. Finally, the base pairs that can be formed between control symbols in clause substrings eliminate the advantage of having more than one literal being **true** in a clause.

So let us first assume that there is an assignment of truth values to the variables of ϕ satisfying ϕ. We will use this assignment to construct a structure with $3c + v$ base pairs, all of which have an energy contribution of -1.

- In each of the v variable substrings, we form the base pair between $\overline{\mathtt{v}}_i$ and one of the $\mathtt{v}_i$'s that will screen the symbols corresponding to literals that become **false** by the truth assignment.
- In each of the c clause substrings, we form two base pairs between complementary control symbols, base pairs that do not form a pseudoknot with each other and that leave the symbol corresponding to a literal that becomes **true** by the truth assignment unscreened. That such a symbol exists follows from the truth assignment satisfying ϕ, thus making at least one literal in each clause true.
- For each of the symbols corresponding to a literal left unscreened in the clause substrings, we form the base pair with its complementary symbol in a variable substring. As the literal becomes **true** by the truth assignment, the complementary symbol in the variable substring will also have been left unscreened. Thus, these base pairs will not form pseudoknots with base pairs involving neighboring bases.

These base pairs together form a structure with energy $-(3c + v)$. That no structure can have an energy lower than $-(3c+v)$ follows from the discussions following Definitions 2 and 3. At most three energetically favorable base pairs can be formed using symbols from a clause substring and at most one energetically favorable base pair can be formed using only symbols from a variable substring.

Now assume that an optimal structure of s_ϕ has energy $-(3c + v)$. By the discussions following definitions 2 and 3, we can conclude that any optimal structure must contain a base pair formed between complementary control symbols in each of the variable substrings. Thus, either the symbols corresponding to the positive or the negative occurrences of each variable will be screened and the others unscreened. We will use this to construct a truth assignment by assigning each variable the truth value that makes the literal to which the unscreened symbols correspond **true**. We now have to argue that this truth assignment satisfies ϕ, that is, that there is in each clause a literal that becomes true by the truth assignment. But this follows from the fact that for each clause substring one of the symbols corresponding to a literal has to form a base pair with an unscreened complementary symbol in a variable substring as the control symbols in a clause substring can only account for two energetically favorable base pairs. ■

In the above proof for Proposition 1, we made extensible use of the unlimited size of the alphabet, with each symbol occuring only once or twice in the constructed sequence. However, if $\mathbf{P} \neq \mathbf{NP}$ it still tells us that a polynomial time algorithm for finding the optimal energy of a secondary structure for an RNA

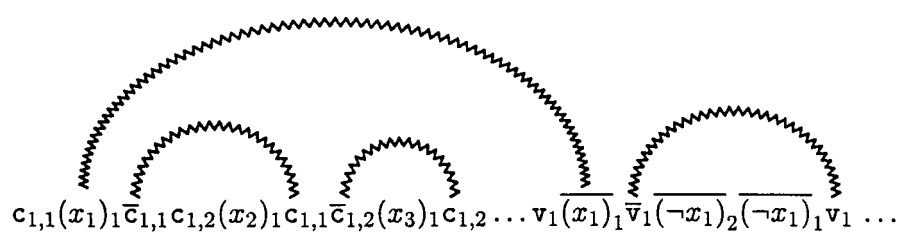


FIG. 10. Part of a base pairing for the sequence constructed in figure 9 illustrating the correspondence between an optimal base pairing and a satisfying truth assignment for the Boolean formula.

sequence in the Nearest Neighbor Pseudoknot Model would have to exploit the fixed size of the RNA alphabet.

4.2. Reduction using the RNA alphabet

We will now prove that even with the four-letter RNA alphabet the structure prediction problem using the Nearest Neighbor Pseudoknot Model remains **NP** complete. In the proof we will use a reduction to the same restricted 3SAT problem as in the previous section, so in the following ϕ we will still refer to the specific instance of the restricted 3SAT problem we want to determine the satisfiability of. A preliminary version of the proof using a base pair energy function making only A, U base pairs stacking with another A, U base pair and C, G base pairs stacking with another C, G base pair energetically favorable has been published in Lyngsø *et al.* (2000). In this section we present a proof of Proposition 1 using a base pair energy function making the stacking of any combination of Watson-Crick and G, U wobble base pairs energetically favorable. More precisely, we will be using the base pair energy function

$$E(X_i \cdot Y_j, V_{i+1}, W_{j-1}) = \begin{cases} -1 & \text{if } V_{i+1} \cdot W_{j-1} \in \mathcal{S} \text{ and } X \cdot Y, V \cdot W \in \{A \cdot U, U \cdot A, C \cdot G, G \cdot C, G \cdot U, U \cdot G\} \\ 4d+7 & \text{if } X \cdot Y \in \{A \cdot U, U \cdot A, C \cdot G, G \cdot C, G \cdot U, U \cdot G\} \text{ and for} \\ & j' \notin \{i+1, \ldots j-1\} \text{ we have } V_{i+1} \cdot Z_{j'}, W_{j-1} \cdot Z_{j'}, Z_{j'} \cdot V_{i+1}, \\ & Z_{j'} \cdot W_{j-1} \notin \mathcal{S} \\ 4d+8 & \text{otherwise} \end{cases}$$

where d is an integer that will depend on the number of clauses and variables in the specific instance ϕ of the restricted 3SAT problem. We thus assume the energy function as well as the RNA sequence to be input to the problem of determining an optimal secondary structure in the Nearest Neighbor Pseudoknot Model.

The basic idea of the proof is simply to replace symbols in the proof using an unlimited alphabet of the previous section with strings over the RNA alphabet. A symbol and its complementary symbol are replaced by complementary RNA strings. We will use strings with $4d + 9$ bases for this replacement. A helix formed between a string replacing a symbol and a string replacing the complementary symbol will thus contain $4d + 8$ pairs of stacking base pairs. Therefore, the base pair energy function specified above assigns a total score of -1 to such a helix, except if this helix forms a pseudoknot with a base pair involving a neighboring base—in this case the score is 0. The energy function thus mimics the energy function used in the proof using an unlimited alphabet.

There are, however, a couple of potential pitfalls when replacing symbols that are scored by base pairing with strings that are scored by stacking effects. First, because of the G, U wobble base pairs a given string can form a helix with numerous different strings and not only its complementary string. Second, we have to consider strings that straddle two strings replacing neighboring symbols, too, as nothing prevents these from forming helices if they exhibit base complementarity. Third, when claiming that a helix formed between a string replacing a symbol and a string replacing the complementary symbol has a score of -1, it is assumed that the helix is not part of a larger helix, i.e., there are no base pairs stacking with the base pairs at either end of the helix. These pitfalls could easily be avoided if we were allowed to insert extra bases between the strings replacing the symbols, but as our energy model only allows the energy of a base pair to depend on base pairs involving neighboring bases this is not generally an option. A helix formed between strings replacing two complementary control symbols needs to neighbor the intervening strings in order to screen them. We do, however, have a little leeway in the clause substrings. A base pair formed between the leftmost $\mathsf{c}_{i,1}$ (or the rightmost $\mathsf{c}_{i,2}$) and its complementary symbol will neighbor the literal symbol it screens on both sides.

In the rest of this section we start by describing the strings we will use to replace the symbols from the proof using an unlimited alphabet and how to construct the RNA sequence corresponding to ϕ. We then prove that by this construction we avoid the pitfalls mentioned above. After that, the proof of Proposition 1 using the RNA alphabet is just in observing that "symbol" and "base pair" in the proof using an unlimited alphabet can be replaced with "string replacing a symbol" and "helix."

Definition 5. *The d digit* binary representation *of k, where* $0 \leq k \leq 2^d - 1$*, over the alphabet* $\{A, U\}$*, is the string* $b_{\{A,U\}}(k, d)$ *of length d that interpreted as a binary number with A representing* 0 *and U*

representing 1 *has the value k. Similarly* $b_{\{C,G\}}(k,d)$ *is the d digit binary representation of k over the alphabet* $\{C, G\}$.

The k'th distinct $\{A, U\}$ pattern *using d digit binary representations is the string*

$$\underbrace{A \ldots A}_{d+2} U b_{\{A,U\}}(k,d) A U A b_{\{A,U\}}(k,d) U \underbrace{A \ldots A}_{d+2}.$$

The k'th distinct $\{C, G\}$ *pattern using d digit binary representations is defined similarly.*

Definition 6. *For a string s the* complementary string $\overline{s}$ *is the string constructed by reversing s and replacing each A with a U, each U with an A, each C with a G, and each G with a C.*

The purpose for these distinct patterns is to circumvent the fact that we only have four letters, or bases, in the RNA alphabet. We will use these distinct patterns to emulate the symbols used in the proof with an alphabet of arbitrary size presented in the previous section. More precisely, we will use distinct $\{A, U\}$ patterns to emulate the control symbols used in the clause substrings and distinct $\{C, G\}$ patterns to emulate the control symbols used in the variable substrings and the symbols corresponding to literals. As symbols are thus replaced by distinct patterns, base pairings between complementary symbols are replaced by helices formed between patterns. A distinct pattern using d digit binary representations consists of $4d + 9$ bases. As mentioned previously, one can thus observe that a helix formed between a distinct pattern with d digit binary representations and its complementary string will have a total energy of -1 using the above base pair energy function, provided that the innermost base pair does not form a pseudoknot with a base pair involving a neighboring base. In that case, the total energy will be 0.

Using patterns instead of single characters, we need to be a little careful though. First of all, one pattern could occur unexpectedly somewhere else in the constructed sequence as the concatenation of a postfix of one pattern and a prefix of the following pattern. However, as we will show later, this is prevented by the design of the distinct patterns in Definition 5. In fact, we will show that, even allowing G, U base pairings, the only helix a distinct pattern can form is with deliberate occurrences of its complement string. We will use the term *complementary pattern* to refer to deliberate occurrences of complementary strings to distinct patterns.

Another problem we have to deal with is the possible extension of helices. When using an alphabet of unlimited size, we could simply make a base pair between complementary symbols energetically favorable regardless of whether it is stacking with another base pair. With the restriction to the RNA alphabet, we have to rely on helix initiation costs combined with stacking effects to make the helix formed between a distinct pattern and its complementary pattern energetically favorable, while not making any shorter helices favorable. But a stacking base pair extending such a helix will not be subject to any initiation penalty and thus in itself be energetically favorable. We thus have to ensure that no helices formed between a distinct pattern and its complementary pattern can be extended by a stacking Watson-Crick or G, U wobble base pair at either end. This means that symbols in the proof using an unlimited alphabet cannot simply be replaced with arbitrary distinct patterns over the RNA alphabet. We have to be a little careful choosing between $\{A, U\}$ and $\{C, G\}$ patterns. Furthermore, a few extra bases have to be inserted at key positions.

Definition 2′. *Let* $C_i = l_1 \vee l_2 \vee l_3$ *be the i'th clause of* ϕ. *The* clause block $\mathcal{C}_i$ *corresponding to* C_i *using d digit binary representations is the string*

$$\underbrace{\blacksquare}_{S_1}\underbrace{\blacksquare}_{L_1} AC \underbrace{\blacksquare}_{\overline{S}_1}\underbrace{\blacksquare}_{S_2}\underbrace{\blacksquare}_{L_2}\underbrace{\blacksquare}_{S_1}\underbrace{\blacksquare}_{\overline{S}_2} CA \underbrace{\blacksquare}_{L_3}\underbrace{\blacksquare}_{S_2},$$

where the $S_{i,j}$*'s are distinct* $\{A, U\}$ *patterns using d digit binary representations for two different k's, and the* L_i*'s are distinct* $\{C, G\}$ *patterns using d digit binary representations for three different k's.*

One observes the close correspondence between the clause substrings of Definition 2 and clause blocks. Mostly symbols have just been replaced with distinct patterns. A few extra bases have been inserted, however, primarily to ensure that all distinct patterns corresponding to literals are flanked by A's. But to prevent these A's from forming an energetically favorable base pair extending a helix formed between $\overline{S}_1$ and the rightmost S_1 (between $\overline{S}_2$ and the leftmost S_2) we furthermore have to insert a C to the right

(to the left) of these A's. It should be noted, though, that the helix formed between $\overline{S}_1$ and the leftmost S_1 (between $\overline{S}_2$ and the rightmost S_2) will still screen the L_1 pattern (the L_3 pattern) as the S_1 pattern (the S_2 pattern) still neighbors the L_1 pattern (the L_3 pattern). By a pattern being screened by a helix we mean that a helix formed between the pattern and its complementary pattern will cause the formation of a pseudoknot involving base pairs with neighboring bases. So, as in the unlimited alphabet case, we can form at most two energetically favorable helices between the $S_{i,j}$ patterns and their complementary patterns, each energetically favorable helix screening one of the patterns corresponding to a literal. If a clause has only two literals we will let L_3 be a dummy pattern not having a complementary pattern occuring anywhere in the sequence we are about to construct, similar to the unlimited alphabet case.

Definition 3′. *Let x_i be a variable occurring in ϕ. The* variable block $\mathcal{V}_i$ *of x_i using d digit binary representations is the string*

$$\underbrace{}_{S_1}\underbrace{\phantom{\overline{P}_1}}_{\overline{P}_1}\underbrace{\phantom{\overline{P}_2}}_{\overline{P}_2}\underbrace{\phantom{\overline{S}_1}}_{\overline{S}_1}\underbrace{\phantom{\overline{N}_1}}_{\overline{N}_1}\underbrace{\phantom{\overline{N}_2}}_{\overline{N}_2}\underbrace{}_{S_1}A,$$

where S_1 is a distinct $\{C, G\}$ pattern for some k, the $\overline{P}_j$'s are complementary strings to the distinct $\{C, G\}$ patterns used for the at most two positive occurrences of x_i (if x_i occurs as positive only once, one of the $\overline{P}$ patterns is omitted from $\mathcal{V}_i$) and the $\overline{N}_j$'s are complementary strings to the distinct $\{C, G\}$ patterns used for the, at most, two negative occurrences of x_i (if x_i occurs negative only once, one of the $\overline{N}$ patterns is omitted from $\mathcal{V}_i$). The order of the complementary patterns corresponding to the positive (negative) occurrences of the variable are reversed compared to their occurrences in the clauses.

The addition of a single A at the end of a variable block ensures that a helix formed between an S_1 pattern and its complementary string cannot be extended with base pairs involving bases from a neighboring variable block. As helices intended to screen other strings are formed only within a clause or variable block, inserting bases between blocks does not cause problems. Apart from this, we have just replaced symbols in Definition 2 with distinct patterns and their complementary patterns. Similarly to the unlimited alphabet case, a helix formed between one of the S_i patterns and its complementary pattern $\overline{S}_i$ will screen the complementary patterns corresponding to either the positive or the negative occurrences of the variable. We are now ready to construct the RNA sequence representing the Boolean formula ϕ.

Definition 4′. *Let ϕ be a Boolean formula on conjunctive normal form where each clause has three literals and each literal occurs at most twice. Assume that ϕ consists of c clauses and uses v variables. The RNA sequence corresponding to ϕ is the sequence*

$$s_\phi = \mathcal{C}_1\mathcal{C}_2 \ldots \mathcal{C}_c\mathcal{V}_1\mathcal{V}_2 \ldots \mathcal{V}_v,$$

where $\mathcal{C}_i$ is the clause block using $\lceil \log_2(3c + v) \rceil$ digit binary representations corresponding to the i'th clause of ϕ, $\mathcal{V}_i$ is the variable block using $\lceil \log_2(3c + v) \rceil$ digit binary representations corresponding to the i'th variable of ϕ, and no distinct pattern is used in more than one connection.

The choice of number of digits we use in the binary representations ensures that we can choose at least $3c + v$ different values for distinct patterns. Each clause block uses two distinct $\{A, U\}$ patterns and three distinct $\{C, G\}$ patterns, while each variable block uses one distinct $\{C, G\}$ pattern. Thus we do not run out of patterns. It should be mentioned here that the integer d appearing in the base pair energy function specified above should be chosen equal to the number of digits used in the binary representations of the constructed sequence, i.e., $d = \lceil \log_2(3c + v) \rceil$.

So far we have assumed that helices only form between a distinct pattern and the complementary string designed to form a helix with it. Helices can of course form between parts of distinct patterns designed not to form helices together, but the following lemma limits the length of such helices, thus establishing the fact that distinct patterns can form helices only with their complementary patterns.

Lemma 1. *Let s_ϕ be an RNA sequence constructed from a Boolean formula ϕ according to definition 4′. In any structure $\mathcal{S}$ of s_ϕ, any helix of consecutively stacking base pairs, all of which are Watson-Crick and G, U wobble base pairs, of length at least $4d + 9$, where d is the number of digits used for the binary*

representations, will have at least $2d + 4$ bases at the end of a distinct pattern forming base pairs with the intended bases of a complementary pattern to this distinct pattern.

Proof. We observe that by construction any substring of s_ϕ of length at least $4d + 9$ will contain either the first $2d + 4$ bases or the last $2d + 4$ bases of a distinct pattern or the complement of a distinct pattern. This follows from the fact that any substring of s_ϕ of length $4d + 9$ will consist of parts of at most two distinct patterns or complements of distinct patterns and at most two bases not part of a distinct pattern or a complement of a distinct pattern.

If one of the two substrings forming the helix contains at least $2d + 4$ bases of a distinct $\{A, U\}$ pattern (or of a distinct $\{C, G\}$ pattern), the claim of the lemma is easy to establish as we do not have to worry about G, U wobble base pairs. Assume that one of the strings forming the helix contains the first $2d + 4$ bases of the k'th distinct $\{A, U\}$ pattern, i.e., it contains the substring $A^{d+2}Ub_{\{A,U\}}(k, d)A$. The $A^{d+2}U$ part of this substring has to form a helix with a stretch of $d + 2$ U's preceded by either an A or a G. The only places in s_ϕ where this occurs are at the ends of complementary patterns to distinct $\{A, U\}$ patterns and at the beginning of the $\overline{S}_2$ complementary pattern in a clause block (cf. Definition 2′). In the latter case, the stretch of $d + 2$ U's will be preceded by a stretch of $d + 2$ A's. The first of these will then form a base pair with the A at the end of $A^{d+2}Ub_{\{A,U\}}(k, d)A$. But this violates the assumption that the helix only contains Watson-Crick and G, U wobble base pairs. In the former case, the stretch of $d + 2$ U's will be preceded by $\overline{b}_{\{A,U\}}(k', d)$ for some k'. As $b_{\{A,U\}}(k, d)$ and $\overline{b}_{\{A,U\}}(k', d)$ must form a helix, k and k' are identical and the complementary $\{A, U\}$ pattern is actually a complementary pattern to the k'th distinct $\{A, U\}$ pattern. Similar arguments can be used if one of the strings forming the helix contains the last $2d + 4$ bases of a distinct $\{A, U\}$ pattern or either the first or the last $2d + 4$ bases of a distinct $\{C, G\}$ pattern.

Assume now that neither of the two substrings forming the helix contain $2d + 4$ bases of a distinct $\{A, U\}$ pattern or a distinct $\{C, G\}$ pattern. They must then contain $2d + 4$ bases of the complementary pattern either to a distinct $\{A, U\}$ pattern or to a distinct $\{C, G\}$ pattern. We will argue the case where one of the substrings contains the first $2d + 4$ bases of the complementary pattern to a distinct $\{C, G\}$ pattern, that is, it contains $G^{d+2}C\overline{b}_{\{C,G\}}(k, d)G$. The other three cases can be handled similarly.

A string forming a helix with the string $G^{d+2}C\overline{b}_{\{C,G\}}(k, d)G$ must have a G at position $d + 2$ to base pair with the C. This G has to be followed by $d+2$ bases each of which is either a C or a U. In s_ϕ, no G is within $d + 2$ bases of a U and we can thus restrict our attention to stretches of $d + 2$ consecutive C's. This stretch cannot be at the end of a distinct $\{C, G\}$ pattern, as we assumed that none of the two substrings contained $2d + 4$ bases of a distinct $\{C, G\}$ pattern. The only other place in s_ϕ where there is a G followed by $d + 2$ consecutive C's is at the junction of $\overline{N}_2$ and S_1 in a variable block (cf. Definition 3′). But then the G will be the last of a stretch of $d + 2$ consecutive G's. The first G in this stretch of consecutive G's must then form a base pair with the terminal G in $G^{d+2}C\overline{b}_{\{C,G\}}(k, d)G$, violating the assumption that the helix does not contain base pairs that are neither Watson-Crick nor G, U wobble base pairs. ■

Lemma 1 ensures that any energetically favorable helix either contains or can be extended to a helix containing the designed helix formed between a distinct pattern and its complementary pattern. To establish that the distinct patterns and their complementary patterns successfully replace symbols from an unlimited alphabet, we now need only to prove that a helix formed between a distinct pattern and its complementary pattern cannot be extended with a base pair at either end. The only energetically favorable helices that can be formed are between distinct patterns and their complementary patterns, and these helices all have an energy contribution of -1.

Lemma 2. *Let s_ϕ be an RNA sequence constructed from a Boolean formula ϕ according to definition 4′ using d digit binary representations. In any structure $\mathcal{S}$ of s_ϕ, there are no helices of more than $4d + 9$ consecutively stacking base pairs containing only Watson-Crick and G, U wobble base pairs. The only helices of $4d + 9$ consecutively stacking base pairs containing only Watson-Crick and G, U wobble base pairs are helices formed between distinct patterns and their complementary pattern.*

Proof. The only thing we need to prove is that we cannot extend a helix formed between a distinct pattern and its complementary pattern with a Watson-Crick or G, U wobble base pair in either end. Lemma 1 ensures that a helix of at least $4d + 9$ consecutively stacking base pairs containing only Watson-

Crick and G, U wobble base pairs will contain at least one end of a helix formed between a distinct pattern and its complementary pattern.

The distinct patterns corresponding to literals in the clause blocks are all flanked by A's on either side. Their complementary patterns are flanked by only C's and G's. Thus a helix formed between the distinct pattern corresponding to a literal and its complementary pattern can only be extended with A, C and A, G base pairs. A helix formed between one of the S_1 patterns and the complementary $\overline{S}_1$ pattern in a variable block (cf. Definition 3′), can be extended only with a G, G base pair (on the inside) or a G, A base pair (on the outside). Finally there are the S_i patterns and their complements in the clause blocks (cf. Definition 2′). If the $\overline{S}_1$ complementary pattern forms a helix with the leftmost S_1 pattern, extending it on the inside would result in a C, C base pair and extending it on the outside would result in an A, A base pair (except for the first clause block where it cannot be extended on the outside). If the $\overline{S}_1$ complementary pattern forms a helix with the rightmost S_1 pattern, extending it on the inside would result in an A, C base pair and extending it on the outside would result in a C, U base pair. The argument for the $\overline{S}_2$ complementary pattern is symmetrical. ■

Proof of Proposition 1. After having established Lemmas 1 and 2, the proof of the proposition more or less reduces to just summarizing the results of these lemmas. As mentioned prior to Lemma 2, the only helices in any structure of s_ϕ that can be energetically favorable are those formed between a distinct pattern and its complementary pattern. Furthermore, though a major part of a distinct pattern screened by a helix can still form a helix with the corresponding part of its complementary pattern without having a base pair form a pseudoknot with a base pair involving a neighboring base, at least one of the flanking bases will have to be left out from this helix (cf. the note about the extra A and C bases in clause blocks). Thus, a helix involving (parts of) a screened pattern cannot be energetically favorable as it either does not include the entire pattern or has a base pair forming a pseudoknot with a base pair involving a neighboring base. The proof is now simply taking the proof using an unlimited size alphabet and replacing (complementary) symbols with (complementary) patterns and base pairs with helices – s_ϕ will have a structure with energy $-(3c + v)$ if and only if ϕ is satisfiable. ■

Thus, finding an optimal structure including pseudoknots of an RNA sequence, even using the simple Nearest Neighbor Pseudoknot Model, is **NP** complete. The energy function used to establish this result does not deviate dramatically from the currently used energy functions for RNA secondary structure prediction except for some simplifications. Stacking of Watson-Crick and G, U wobble base pairs is energetically favorable, loops have a destabilising effect and pseudoknots are even further penalized. The only noticeable weakness of the **NP** completeness proof is that we do not use a fixed energy function, but have the energy function – or at least the relative difference between the stabilising effect of a stacking base pair and the destabilising effect of a loop – be part of the problem instance.

As a closing remark it should be mentioned that though the energy function at a first glance seems to prevent the presence of pseudoknots in an optimal structure this is not the case. The energy function only prevents helices forming a pseudoknot from being energetically favorable if they are *neighboring*, i.e., if a base involved in a base pair in one of the helices is neighboring a base involved in a base pair in the other helix. For sequences constructed on the basis of a Boolean formula as described above, an optimal structure is likely to contain numerous nonneighboring pseudoknots formed by helices of distinct patterns corresponding to literals base pairing with their complementary patterns. All these will have a favorable energy contribution. A structure not containing pseudoknots, i.e., an optimal structure determined by an `mfold` type algorithm, will not be able to obtain the favorable energy contributions from all these helices. Thus our proof does rely on the ability to form pseudoknots.

5. DISCUSSION

The **NP** completeness of the RNA secondary structure prediction problem in the Nearest Neighbor Pseudoknot Model tells us that any algorithm allowing energy functions sufficiently general to be specialized to the energy functions in the Nearest Neighbor Pseudoknot Model, and running in worst case polynomial time, would imply $\mathbf{P} = \mathbf{NP}$. The question whether or not **P** is equal to **NP** is one of the fundamental open problems in computer science. Based on strong evidence, the large majority of computer scientists believe that $\mathbf{P} \neq \mathbf{NP}$. The **NP** completeness of the RNA secondary structure prediction problem in the Nearest

Neighbor Pseudoknot Model thus hints that there is only little hope for a worst case polynomial time algorithm for RNA secondary structure prediction in the Nearest Neighbor Pseudoknot Model, or models extending it. Moreover, it hints that any algorithm for predicting RNA secondary structures with general pseudoknots most likely has to exploit specific properties, e.g., a fixed alphabet size and a fixed energy function, to obtain polynomial running time.

That even demanding the alphabet size and energy function to be fixed is not sufficient to circumvent **NP** completeness follows from the **NP** completeness proof in Akutsu (to appear). In this proof, both the alphabet and the energy function are fixed (though the energy function is somewhat unnatural, e.g., allowing base pairs involving identical bases and not making the stacking of all allowed types of base pairs energetically favorable). It should be mentioned, though, that fixing the energy function in the Nearest Neighbor Pseudoknot Model does not turn it into the model used in Akutsu (to appear) or a model that can be specialized to this, as the two models are rather different. The most notable difference is how certain pseudoknot structures are discouraged. In Akutsu (to appear) this is handled by the model of legal structures by accepting only planar structures as legal structures; a structure like the one illustrated in Figure 3(a) can thus not be assigned an energy in the model of Akutsu (to appear) as it is not a legal structure. In the Nearest Neighbor Pseudoknot Model, all structures are legal as long as no base participates in more than one base pair. It is left to the energy function to discourage certain pseudoknot structures. In the **NP** completeness proof, this discouragement takes the form of penalizing base pairs if they form a pseudoknot with a base pair involving a neighboring base.

The RNA secondary structure prediction problem is basically a weighted matching problem. Thus, one way to elucidate the differences between the two **NP** completeness proofs is to look at what underlying matching problems are being proven **NP** complete. In Akutsu (to appear) one can identify the underlying matching problem as that of finding a maximum weighted 3-matching (i.e., every element is allowed to be matched to at most three other elements) that is planar. In the proof presented in this paper one can identify the underlying matching problem as that of finding a maximum weighted matching when the weight of a pair of elements is allowed to depend on the matching of neighboring elements, where neighboring is with respect to a predefined ordering.

One approach to obtain a polynomial time algorithm for RNA secondary structure prediction with pseudoknots is to limit the types of legal pseudoknots. This is the approach taken in the algorithms discussed in the third section. Another approach is taken in Tabaska *et al.* (1998), where interactions between neighboring base pairs are ignored, thus reducing the problem of RNA secondary structure prediction (with pseudoknots) to compute a maximal weighted matching. If we are satisfied with finding structures that are not necessarily of least free energy, then heuristics can be applied to search for structures of low energy. For example, van Batenburg *et al.* (1995) report on successful experiments with applying genetic algorithms to the problem of finding low energy RNA secondary structures containing pseudoknots.

ACKNOWLEDGMENTS

R.L. was supported by a grant from Carlsbergfondet and C.P. was in part supported by the Danish National Research Foundation.

REFERENCES

Akutsu, T. To appear. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics.*

Brown, M., and Wilson, C. 1996. RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. In *Biocomputing: Proceedings of the 1996 Pacific Symposium*, 109–125. World Scientific Publishing Co.

Eddy, S.R., and Durbin, R. 1994. RNA sequence analysis using covariance models. *Nucl. Acids Res.* 22(11), 2079–2088.

Gorodkin, J., Heyer, L.J., and Stormo, G.D. 1997. Finding common sequence and structure motifs in a set of RNA sequences. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 120–123. AAAI Press.

Hirschberg, D.S. 1975. A linear space algorithm for computing maximal common subsequence. *Communications of the ACM* 18(6), 341–343.

Knudsen, B., and Hein, J.J. 1999. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics* 15(6), 446–454.

Lyngsø, R.B., and Pedersen, C.N.S. 2000. Pseudoknots in RNA secondary structures. In *Proceedings of the 4th Annual International Conference on Computational Molecular Biology (RECOMB)*, 201–209. ACM Press.

Lyngsø, R.B., Zuker, M., and Pedersen, C.N.S. 1999. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics* 15(6), 440–445.

Mathews, D.H., Sabina, J., Zuker, M., and Turner, D.H. 1999. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.* 288, 911–940.

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. *Proceedings of the National Academy of Sciences of the United States of America* 77(11), 6309–6313.

Nussinov, R., Pieczenik, G., Griggs, J.R., and Kleitman, D.J. 1978. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics* 35(1), 68–82.

Papadimitriou, C.M. 1994. *Computational Complexity.* Addison-Wesley Publishing Company, Inc.

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. *Nucl. Acids Res.* 12, 31–44.

Pleij, C.W.A. 1993. RNA pseudoknots. In Gesteland, R.F., and Atkins, J.F., eds., *The RNA World.* Cold Spring Harbor Laboratory Press.

Rivas, E., and Eddy, S. 1999. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.* 285(5), 2053–2068.

Sakakibara, Y., Brown, M., Hughey, R., Mian, I.S., Sjölander, K., Underwood, R.C., and Haussler, D. 1994. Stochastic context-free grammars for tRNA modeling. *Nucl. Acids Res.* 22(23), 5112–5120.

Sankoff, D. 1985. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics* 45(5), 810–825.

Searls, D.B. 1992. The linguistics of DNA. *American Scientist* 80, 579–591.

Tabaska, J.E., Cary, R.B., Gabow, H.N., and Stormo, G.D. 1998. An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics* 14(8), 691–699.

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 Biology* 246, 40–41.

Turner, D.H., Sugimoto, N., and Freier, S.M. 1988. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry* 17, 167–192.

Uemura, Y., Hasegawa, A., Kobayashi, S., and Yokomori, T. 1999. Tree adjoining grammars for RNA structure prediction. *Theoretical Computer Science* 210, 277–303.

van Batenburg, F.H.D., Gultyaev, A.P., and Pleij, C.W.A. 1995. An APL-programmed genetic algorithm for the prediction of RNA secondary structure. *Journal of Theoretical Biology* 174(3), 269–280.

Zhang, K., and Shasha, D. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing* 18(6), 1245–1262.

Zuker, M. 2000. *mfold* server at *http://mfold2.wustl.edu/~mfold/rna/form1.cgi.*

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

Address correspondence to:
Rune B. Lyngsø
Baskin Center for Computer Sciences and Engineering
University of California
1156 High Street
Santa Cruz, CA 95064

E-mail: rlyngsoe@cse.ucsc.edu