An IP Algorithm for RNA Folding Trajectories[*]

Amir H. Bayegan[1] and Peter Clote[2]

1 **Boston College, Biology Department, Chestnut Hill, MA, USA**
a.h.bayegan@gmail.com
2 **Boston College, Biology Department, Chestnut Hill, MA, USA**
clote@bc.edu

Abstract

Vienna RNA Package software `Kinfold` implements the Gillespie algorithm for RNA secondary structure folding kinetics, for the move sets MS_1 [resp. MS_2], consisting of base pair additions and removals [resp. base pair addition, removals and shifts]. In this paper, for arbitrary secondary structures s, t of a given RNA sequence, we present the first optimal algorithm to compute the shortest MS_2 folding trajectory $s = s_0, s_1, \ldots, s_m = t$, where each intermediate structure s_{i+1} is obtained from its predecessor by the addition, removal or shift of a single base pair. The shortest MS_1 trajectory between s and t is trivially equal to the number of base pairs belonging to s but not t, plus the number of base pairs belonging to t but not s. Our optimal algorithm applies integer programming (IP) to solve (essentially) the minimum feedback vertex set (FVS) problem for the "conflict digraph" associated with input secondary structures s, t, and then applies topological sort, in order to generate an optimal MS_2 folding pathway from s to t that maximizes the use of shift moves. Since the optimal algorithm may require excessive run time, we also sketch a fast, near-optimal algorithm (details to appear elsewhere). Software for our algorithm will be publicly available at `http://bioinformatics.bc.edu/clotelab/MS2distance/`.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Integer programming, RNA secondary structure, folding trajectory, feedback vertex problem, conflict digraph

Digital Object Identifier 10.4230/LIPIcs.WABI.2017.6

1 Introduction

In this paper, we introduce the first algorithm to compute the MS_2 distance between two secondary structures s and t of a given RNA sequence $a_1, \ldots, a_n$; i.e. the length m of the shortest refolding trajectory $s = s_0, s_1, \ldots, s_m = t$, in which each intermediate secondary structure s_{i+1} is obtained from s_i by a single base pair addition, removal or shift. Here a *shift* transforms a base pair (x, y) to the base pair (x', y'), where either $x \in \{x', y'\}$ or $y \in \{x', y'\}$, but not both; see Figure 1 for an illustration of all possible types of shift moves. Although shifts are considered in the secondary structure folding kinetics program `Kinfold` [12] as well as in theoretical work on RNA molecular structure evolution [15], most papers on RNA secondary structure do not consider shift moves, presumably due to the sometimes tremendous additional complications even though the shift moves for helix zippering and defect diffusion are supported by experimental data [14]. Indeed, while our algorithm to compute the expected number of nearest neighbors with respect to MS_1

[*] This research was supported in part by National Science Foundation grant DBI-1262439 (PC). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

17th International Workshop on Algorithms in Bioinformatics (WABI 2017).
Editors: Russell Schwartz and Knut Reinert; Article No. 6; pp. 6:1–6:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

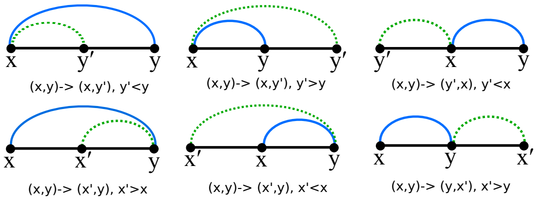


Figure 1 Shift moves from solid base pair to dotted base pair. Image taken from [2].

moves [1] is highly non-trivial, our analogous algorithm for MS_2 moves is far more complex [2]. Moreover, the current paper illustrates the enormous computational complexity that arises when considering MS_2 distance rather than MS_1 distance – while MS_1 distance, also known as *base pair distance*, is trivial to compute, we conjecture that MS_2 distance is NP-complete, where this problem can be formalized as a decision problem to determine, for any given secondary structures s, t and integer m, whether there is an MS_2 trajectory $s = s_0, s_1, \ldots, s_m = t$ of length $\leq m$, in which each intermediate secondary structure s_{i+1} is obtained from s_i by a single base pair addition, removal or shift. Here, we describe an exact (possibly exponential time) integer programming (IP) algorithm, and in a sequel to this paper, we will describe a fast, *near-optimal* algorithm, a greedy algorithm and a slow, exact branch-and-bound algorithm (details of these algorithm cannot be given here, due to space constraints). We conclude the current paper by a benchmarking comparison between the optimal IP algorithm and the near-optimal algorithm, and compare the values of MS_1, MS_2 and Hamming distance on a data set of 3800 random RNA sequences having random initial and random target structures of length n, computed for a range of values of n.

Since our algorithms involve the *feedback vertex set* problem for *RNA conflict digraphs*, we now provide a bit of background about this problem. Given a directed graph, or digraph, $G = (V, E)$, a *feedback vertex set* (FVS) is a subset $V' \subseteq V$ which contains at least one vertex from every directed cycle in G, thus rendering G acyclic. Similarly, a *feedback arc set* (FAS) is a subset $E' \subseteq E$ which contains at least one directed edge (arc) from every directed cycle in G. The FVS [resp. FAS] problem is the problem to determine a minimum size feedback vertex set [resp. feedback arc set] which renders G acyclic. Both the FVS and FAS are NP-complete for arbitrary digraphs, as well as for tournaments; indeed the FVS and FAS problems both appear in the list of 21 problems shown by R.M. Karp to be NP-complete [10]. We now introduce some necessary definitions.

Although the notion of secondary structure is well-known, we give three distinct but equivalent definitions, that will allow us to overload secondary structure notation to simplify presentation of our algorithms.

▶ **Definition 1** (Secondary structure as set of ordered base pairs)**.** Let $[1, n]$ denote the set $\{1, 2, \ldots, n\}$. A secondary structure for a given RNA sequence $a_1, \ldots, a_n$ of length n is defined to be a set s of ordered pairs (i, j), with $1 \leq i < j \leq n$, such that the following conditions are satisfied.

1. *Watson-Crick and wobble pairs:* If $(i, j) \in s$, then $a_i a_j \in \{GC, CG, AU, UA, GU, UG\}$.
2. *No base triples:* If (i, j) and (i, k) belong to s, then $j = k$; if (i, j) and (k, j) belong to s, then $i = k$.
3. *Nonexistence of pseudoknots:* If (i, j) and (k, ℓ) belong to s, then it is not the case that $i < k < j < \ell$.
4. *Threshold requirement for hairpins:* If (i, j) belongs to s, then $j - i > \theta$, for a fixed value $\theta \geq 0$; i.e. there must be at least θ unpaired bases in a hairpin loop. Following standard convention, we set $\theta = 3$ for steric constraints.

If s is a secondary structure (set of ordered pairs), then $|s|$ denotes the size of s, i.e. the number of base pairs belonging to s.

Without risk of confusion, it will be convenient to overload the concept of secondary structure s with two alternative, equivalent notations, for which context will determine the intended meaning.

▶ **Definition 2** (Secondary structure as set of unordered base pairs)**.** A secondary structure s for the RNA sequence $a_1, \ldots, a_n$ is a set of unordered pairs $\{i, j\}$, with $1 \leq i, j \leq n$, such that the corresponding set of ordered pairs

$$\{i, j\}_< \overset{\text{def}}{=} (\min(i, j), \max(i, j)) \tag{1}$$

satisfies Definition 1. If s is a secondary structure (set of unordered pairs), then $|s|$ denotes the size of s, i.e. the number of base pairs belonging to s.

▶ **Definition 3** (Secondary structure as an integer-valued function)**.** A secondary structure s for $a_1, \ldots, a_n$ is a function $s : [1, \ldots, n] \to [0, \ldots, n]$, such that $\Big\{\{i, s[i]\}_< : 1 \leq i \leq n, s[i] \neq 0\Big\}$ satisfies Definition 1; i.e.

$$s[i] = \begin{cases} 0 & \text{if } i \text{ is unpaired in } s \\ j & \text{if } (i, j) \in s \text{ or } (j, i) \in s \end{cases} \tag{2}$$

▶ **Definition 4** (Secondary structure distance measures)**.** Let s, t be secondary structures of length n. Base pair distance is defined by equation (3) below, and Hamming distance is defined by equation (4) below.

$$d_{BP}(s,t) = |\{(x,y) : ((x,y) \in s \wedge (x,y) \notin t) \vee ((x,y) \in t \wedge (x,y) \notin s)\}| \tag{3}$$

$$d_H(s,t) = |\{i \in [1,n] : s[i] \neq t[i]\}| \tag{4}$$

We next define some primitive notions used later to define a central concept of *RNA conflict directed graph* (digraph). Let $[1, n]$ denote the set $\{1, \ldots, n\}$. Given secondary structure s on RNA sequence $\{a_1, \ldots, a_n\}$, we say that a position $x \in [1, n]$ is *touched* by s, or equivalently that the structure s *touches* the position x, if x belongs to a base pair of s, or equivalently $s[x] \neq 0$. Let BP_1 [resp. BP_2] denote the set of base pairs of s [resp. t] which are not touched by any base pair of t [resp. s]; i.e.

$$BP_1 = \{(i,j) \in s : t[i] = 0 = t[j]\} \tag{5}$$

$$BP_2 = \{(i,j) \in t : s[i] = 0 = s[j]\} \tag{6}$$

2 MS_2 distance between secondary structures

In this section, we present an integer programming (IP) algorithm to compute the MS_2 distance between any two secondary structures s, t, i.e. the minimum length of a trajectory from s to t, involving only base pair additions, removals and shifts. Since any shift move, such as $(x, y) \to (x, z)$ can be simulated by removal of the base pair (x, y) followed by addition of the base pair (x, z), our strategy to produce a minimum length MS_2 trajectory is to use graph-theoretic methods to *maximize* the number of shift moves and *minimize* the number of base pair additions and removals. The validity of this approach is formalized in the following simple theorem, whose proof is straightforward.

▶ **Theorem 5.** *Suppose that the MS_1 distance between secondary structures s, t is k, i.e. base pair distance $d_{BP}(s,t) = |s - t| + |t - s| = k$. Suppose that ℓ is the number of shift moves occurring in the shortest MS_2 trajectory $s = s_0, s_1, \dots, s_m = t$ from s to t. Then the MS_2 distance between s and t equals*

$$d_{MS_2}(s,t) = \ell + (k - 2\ell) = k - \ell\,. \tag{7}$$

2.1 RNA conflict digraph

Throughout this section, we take s, t to be two arbitrary, distinct, but fixed secondary structures of the RNA sequence $a_1, \dots, a_n$. To determine a minimum length MS_2 folding trajectory from secondary structure s to secondary structure t, we must maximize the number of shift moves and minimize the number of base pair additions and removals. To that end, note that the base pairs in s that do not touch any base pair of t must be removed in any MS_2 path from s to t, since there is no shift of such base pairs to a base pair of t – such base pairs are exactly those in BP_1, defined in equation (5). Similarly, note that the base pairs in t that do not touch any base pair of s must occur must be added, in the transformation of s to t, since there is no shift of any base pair from s to obtain such base pairs of t – such base pairs are exactly those in BP_2, defined in equation (6). We now focus on the remaining base pairs of s, all of which touch a base pair of t, and hence could theoretically allow a shift move in transforming s to t, *provided* that there is no base triple or pseudoknot introduced by performing such a shift move. Examples of all six possible types of shift move are illustrated in Figure 1. To handle such cases, we define the notion of *RNA conflict digraph*, solve the *feedback vertex set* (FVS) problem by integer programming (IP), apply topological sorting [3] to the acyclic digraph obtained by removing a minimum set of vertices occurring in feedback loops, then apply shift moves in topologically sorted order. We now formalize this argument.

Define the digraph $G = (V, E)$, whose vertices (or nodes) $n \in V$ are defined in the following Definition 6 and whose directed edges are defined in Definition 7.

▶ **Definition 6** (Vertex in an RNA conflict digraph). If s, t are distinct secondary structures for the RNA sequence $a_1, \dots, a_n$, then a vertex in the RNA conflict digraph $G = G(s,t)$ is a triplet node, or more simply, node $v = (x, y, z)$ consisting of integers x, y, z, such that the base pair $\{x, y\}_< = (\min(x,y), \max(x,y))$ belongs to t, and the base pair $\{y, z\}_< = (\min(y,z), \max(y,z))$ belongs to s. Let $v.t$ [resp. $v.s$] denote the base pair $\{x, y\}_<$ [resp. $\{y, z\}_<$] belonging to t [resp. s]. The middle integer y of node $v = (x, y, z)$ is called the *pivot* position, since it is common to both s and t. Nodes are ordered by the integer ordering of their pivot positions: $(x, y, z) \preceq (x', y', z')$ if and only if $y \le y'$ (or $y = y'$ and $x < x'$, or $y = y'$, $x = x'$, and $z < z'$). If $v = (x, y, z)$ is a node, then $flatten(v)$ is defined to be the set $\{x, y, z\}$ of its coordinates.

Nodes are representations of a potential shift move, and can be categorized into six types, as shown in Figure 1.

▶ **Definition 7** (Directed edge in an RNA conflict digraph). The base pair $\{a, b\}_<$ is said to *cross* the base pair $\{c, d\}_<$ if either $\min(a,b) < \min(c,d) < \max(a,b) < \max(c,d)$ or $\min(c,d) < \min(a,b) < \max(c,d) < \max(a,b)$; in other words, base pairs cross if they form a pseudoknot. Base pairs $\{a, b\}_<$ and $\{c, d\}_<$ are said to touch if $|\{a, b\} \cap \{c, d\}| = 1$; in other words, base pairs touch if they form a base triple. There is a directed edge from node $n_1 = (x_1, y_1, z_1)$ to node $n_2 = (x_2, y_2, z_2)$, denoted $n_1 \to n_2$, if either $z_1 = x_2$ or the base pair $\{y_1, z_1\}_< \in s$ from n_1 crosses base pair $\{x_2, y_2\}_< \in t$ from n_2.

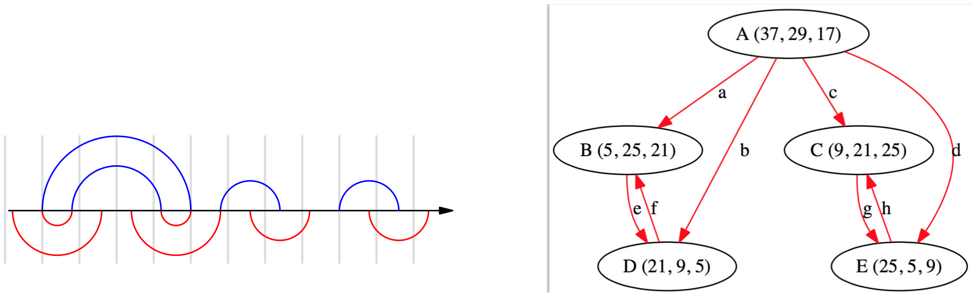


Figure 2 Rainbow diagram (left) and conflict digraph (right) for the toy example of initial structure s consisting of the six base pairs $(1, 13)$, $(5, 9)$, $(17, 29)$, $(21, 25)$, $(33, 41)$, $(49, 57)$ and of the target structure t consisting of the four base pairs $(5, 25)$, $(9, 21)$, $(29, 37)$, $(45, 53)$. The corresponding conflict digraph consists of 5 vertices, 8 directed edges, and 2 directed cycles. Edges are labeled for discussion in the text.

The motivation behind the definition of *edge* $\mathbf{v_1} \to \mathbf{v_2}$ is that either a base triple or pseudoknot will result if one applies the shift corresponding to $\mathbf{v_2}$ before the shift corresponding to $\mathbf{v_1}$. To that end, it is natural to define the edge $\mathbf{v_1} \to \mathbf{v_2}$ if the base pair $\mathbf{v_1}.s$ either touches or crosses the base pair $\mathbf{v_2}.t$. This approach is valid, but may lead to larger conflict digraphs with many more cycles than necessary, resulting in additional computational cost in the enumeration of all directed cycles as well as in application of the IP solver. Consider the example of initial structure $s = \{(5, 9)\}$ and target structure $t = \{(1, 5), (9, 13)\}$, for which an optimal 2-step MS_2 trajectory consists of shifting base pair $(5, 9)$ to $(1, 5)$, followed by adding the base pair $(9, 13)$. Vertices of the conflict digraph $G = (V, E)$ are clearly $\mathbf{v_1} = (1, 5, 9)$ and $\mathbf{v_2} = (13, 9, 5)$. Since $\mathbf{v_1}.s = (5, 9)$ touches $\mathbf{v_2}.t = (9, 13)$, we would have $\mathbf{v_1} \to \mathbf{v_2}$; since $\mathbf{v_2}.s = (5, 9)$ touches $\mathbf{v_1}.t = (1, 5)$, we would have $\mathbf{v_2} \to \mathbf{v_1}$. However, if we apply the shift move corresponding to $\mathbf{v_1}$, then we cannot subsequently apply the shift move corresponding to $\mathbf{v_2}$, since $\mathbf{v_1}.s = \mathbf{v_2}.s$. A similar issue arises when $\mathbf{v_1}.t = \mathbf{v_2}.t$.

For another example, consider the initial structure $s = \{(5, 9), (21, 25)\}$ and target structure $t = \{(5, 25), (9, 21)\}$. Vertices of the conflict digraph are B, C, D, E, where B is $(5, 25, 21)$, D is $(21, 9, 5)$, C is $(9, 21, 25)$, E is $(25, 5, 9)$. Supposing that $\mathbf{v_1} = (x_1, y_1, z_1)$ and $\mathbf{v_2} = (x_2, y_2, z_2)$, if $\mathbf{v_1} \to \mathbf{v_2}$ were defined by $\mathbf{v_1}.s$ touches or crosses $\mathbf{v_2}.t$, then the resulting conflict digraph would be the complete digraph on vertices B, C, D, E, leading to many more cycles than necessary. By defining $\mathbf{v_1} \to \mathbf{v_2}$ by either $z_1 = x_2$ or $\mathbf{v_1}.s$ crosses $\mathbf{v_2}.t$, the resulting conflict digraph consists of only $B \to C$, $C \to B$, $D \to E$, $E \to D$, which appears as a portion of Figure 2.

▶ **Definition 8** (Conflict digraph $G = (V, E)$)**.** Let s, t be distinct secondary structures for the RNA sequence $a_1, \ldots, a_n$. The RNA confict digraph $G(s, t) = (V(s, t), E(s, t))$, or $G = (V, E)$ when s, t are clear from context, is defined by

$$V = \{(x, y, z) : x, y, z \in [1, n] \wedge \{x, y\} \in t \wedge \{y, z\} \in s\}, \tag{8}$$

$$\begin{aligned} E = \Big\{&(n_1, n_2) : n_1 = (x_1, y_1, z_1) \in V \wedge n_2 = (x_2, y_2, z_2) \in V \wedge \Big(z_1 = x_2 \vee \\ &\Big([\min(y_1, z_1) < \min(x_2, y_2) < \max(y_1, z_1) < \max(x_2, y_2)] \vee \\ &[\min(x_2, y_2) < \min(y_1, z_1) < \max(x_2, y_2) < \max(y_1, z_1)]\Big)\Big\}. \end{aligned} \tag{9}$$

Notice that Definition 8 establishes a partial ordering on vertices of the conflict digraph $G = (V, E)$, in that edges determine the order in which shift moves should be performed.

Indeed, if $n_1 = (x, y, z)$, $n_2 = (u, v, w)$ and $(n_1, n_2) \in E$, which we denote from now on by $n_1 \to n_2$, then the shift move in which $\{y, z\} \in s$ shifts to $\{x, y\} \in t$ *must* be performed *before* the shift move where $\{v, w\} \in s$ shifts to $\{u, v\} \in t$ – indeed, if shifts are performed in the opposite order, then after shifting $\{v, w\} \in s$ to $\{u, v\} \in t$ and before shifting $\{y, z\} \in s$ to $\{x, y\} \in t$, we would create either a base triple or a pseudoknot. Our strategy to efficiently compute the MS_2 distance between secondary structures s and t will be to (1) enumerate all simple cycles in the conflict digraph $G = (V, E)$ and to (2) apply an integer programming (IP) solver to solve the minimum feedback arc set problem $V' \subset V$. Noticing that the *induced digraph* $\overline{G} = (\overline{V}, \overline{E})$, where $\overline{V} = V - V'$ and $\overline{E} = E \cap (\overline{V} \times \overline{V})$, is acyclic, we then (3) topologically sort $\overline{G}$, and (4) perform shift moves from $\overline{V}$ in topologically sorted order. In an initial implementation of our algorithms, we used the `simple_cycles()` function from the `NetworkX` python library to enumerate all simple cycles.

There is a first important technical deviation from this strategy, corresponding to an additional IP constraint (‡) in line 7 of the pseudocode below, necessary to address a possible *overlap* between triplet nodes. It can happen, for instance, that base pair $(x, y) \in t$, and that base pairs $(u, x) \in s$ and $(y, z) \in s$, for which we have triplet nodes $v_1 = (y, x, u)$ and $v_2 = (x, y, z)$. If we detect node v_1 [resp. v_2] in a simple cycle C_1 [resp. C_2], then in the absence of (‡), the first IP constraint (†) would *remove* both nodes v_1 and v_2, whereby IP variables x_{v_1} and x_{v_2} would both be set to 0, resulting in the *removal of both* base pairs (u, x) and (y, z) from s in line 16 of the pseudocode. This causes an additional base pair *addition* of (x, y) to the folding pathway in line 18 of the pseudocode. In contrast, if (for instance) only the node v_1 had been removed, resulting in the base pair removal of (u, x) from s, then it would have been possible to shift base pair (y, z) to (x, y), rather than removing both (u, x) and (y, z) from s with subsequent base pair addition of (x, y). Such situations are avoided by the IP constraint (‡) below.

One might (incorrectly) surmise that it is possible to immediately remove the base pair $\{y, z\}$ from s for every node $v = (x, y, z) \notin \overline{V}$. The fallacy of doing this can be illustrated as follows. Suppose, for instance, that base pair $(x, y) \in s$, and that base pairs $(u, x) \in t$ and $(y, z) \in t$, for which we have triplet nodes $v_1 = (u, x, y)$ and $v_2 = (z, y, x)$. Since v_1, v_2 overlap in two positions, by the constraint (‡) in line 7 of the pseudocode below, it cannot be that both v_1 and v_2 both belong to $\overline{V}$. If neither v_1 nor v_2 belongs to $\overline{V}$, then it is safe to immediately remove base pair $\{x, y\}$ from s. However, if (say) $v_1 \in \overline{V}$ and $v_2 \notin \overline{V}$, then it would be a mistake to remove $\{x, y\}$ from s if we could instead later shift base pair $\{x, y\}$ to base pair $\{u, v\}$, *provided* that so doing does not create a base triple. Such a shift is possible if position u is not touched by s. A clean treatment of such situations requires the following definition.

▶ **Definition 9** (Base pair (x, y) is covered by $\overline{V}$)**.** Suppose that $G = (V, E)$ is the RNA conflict digraph for RNA sequence $a_1, \ldots, a_n$ and secondary structures s, t. Let $\overline{V} \subset V$. Base pair $(x, y) \in t$ is covered by $\overline{V}$ if there exists $v \in \overline{V}$ such that $v.t = (x, y)$, i.e. the t base pair portion of v equals (x, y). Base pair $(x, y) \in s$ is covered by $\overline{V}$ if there exists a base pair $(u, v) \in t$ such that (x, y) touches (u, v) and (u, v) is covered.

It will now follow that we can remove all base pairs $(x, y) \in s$ that are not covered by $\overline{V}$, as indicated in line 11 of the following pseudocode.

▶ **Algorithm 1** (MS_2 distance from s to t)**.**
Input: *Secondary structures s, t for RNA sequence $a_1, \ldots, a_n$*
Output: *Folding trajectory $s = s_0, s_1, \ldots, s_m = t$, where $s_0, \ldots, s_m$ are secondary structures, m is the minimum possible value for which s_i is obtained from s_{i-1} by a single base pair addition, removal or shift for each $i = 1, \ldots, m$.*

First, initialize the variable numMoves to 0, and the list moveSequence to the empty list [ ]. Recall that $BP_2 = \{(x,y) : (x,y) \in t, (s-t)[x] = 0, (s-t)[y] = 0\}$. Bear in mind that s is constantly being updated, so actions performed on s depend on its current value.

```
 //remove base pairs from $s$ that are untouched by $t$
 $BP_1 = \{(x,y) : (x,y) \in s, (t-s)[x] = 0, (t-s)[y] = 0\}$
 for $(x,y) \in BP_1$
     remove $(x,y)$ from $s$; numMoves = numMoves+1
 //define conflict digraph $G = (V,E)$ on updated $s$ and unchanged $t$
 define $V$ by equation (8)
 define $E$ by equation (9)
 define conflict digraph $G = (V,E)$
 //IP solution of minimum feedback arc set problem
 maximize $\sum_{v \in V} x_v$ where $x_v \in \{0,1\}$, subject to constraints (†) and (‡)
 //first constraint removes vertex from each simple cycle of $G$
 (†) $\sum_{v \in C} x_v < |C|$ for each simple directed cycle $C$ of $G$
 //ensure shift moves cannot be applied if they share same base pair from $s$ or $t$
 (‡) $x_v + x_{v'} \leq 1$, for all pairs of vertices $v = (x,y,z)$ and $v' = (x',y',z')$
   with $|\{x,y,z\} \cap \{x',y',z'\}| = 2$
 $\overline{V} = \{v \in V : x_v = 1\}$
 $\overline{E} = \{(v,v') : v,v' \in \overline{V} \wedge (v,v') \in E\}$
 let $\overline{G} = (\overline{V}, \overline{E})$
  //remove all base pairs of $s$ not covered by $\overline{V}$
 let $\overline{\text{Cover}} = \{(x,y) \in s : (x,y)$ is not covered by $\overline{V}\}$
 for $(x,y) \in \overline{\text{Cover}}$
     remove $(x,y)$ from $s$; numMoves = numMoves+1
 //topological sort of IP solution $\overline{V}$
 topological sort of $\overline{G}$ to determine total ordering $\prec$ on $\overline{V}$
 for $v = (x,y,z) \in \overline{V}$ in topologically sorted order $\prec$
     shift $\{y,z\}$ to $\{x,y\}$ in $s$; numMoves = numMoves+1
    //add remaining base pairs from $t-s$
 for $(x,y) \in t-s$
     add $(x,y)$ to $s$; numMoves = numMoves+1
 return folding trajectory, numMoves
```

Toy example

Consider the following toy 48 nt example of initial structure s consisting of the six base pairs $(1,13)$, $(5,9)$, $(17,29)$, $(21,25)$, $(33,41)$, $(49,57)$ and the target structure t consisting of the four base pairs $(5,25)$, $(9,21)$, $(29,37)$, $(45,53)$ with dot-bracket structures given by

```
>s
1234567890123456789012345678901234567890123456789012345678
GAAAGAAAUAAACAAAGAAAGAAACAAACAAAGAAAGAAACAAAGAAAGAAACAAACA
(...(...)...)...(...(...)...)...(.......).......(.......).
>t
1234567890123456789012345678901234567890123456789012345678
GAAAGAAAUAAACAAAGAAAGAAACAAACAAAGAAAGAAACAAAGAAAGAAACAAACA
....(...(...........)...)...(.......).......(.......).....
```

Figure 2a depicts the “rainbow” diagram of initial structure s shown below the line in red, and of target structure t shown above the line in blue. If $G = (V,E)$ denotes the corresponding conflict digraph of s,t, as depicted in Figure 2b, then the vertices $\mathbf{v} = (x,y,z)$

belonging to V are exactly those triples (x,y,z) such that blue arc $\mathbf{v}.t = (x,y)_<$ touches red arc $\mathbf{v}.s = (y,z)_<$. Moreover, for vertices $\mathbf{v}_1 = (x_1,y_1,z_1)$ and $\mathbf{v}_2 = (x_2,y_2,z_2)$, there is a directed edge $\mathbf{u} \to \mathbf{v}$ exactly when either (1) $z_1 = x_2$ or (2) $\mathbf{v}_1.s$ crosses $\mathbf{v}_2.t$. For the current example, it is straightforward for the user to derive the conflict digraph from the rainbow diagram, and to confirm that the conflict digraph $G = (V,E)$ consists of 5 vertices, 8 directed edges, and 2 directed cycles, as shown in Figure 2b.

A solution feedback vector set (FVS) problem is given by $\overline{V} = \{A,B,C\} = \{(37,29,17), (5,25,21), (9,21,25)\}$, since $\overline{V}$ is a maximum size subset of V such that the induced digraph $\overline{G} = (\overline{V},\overline{E})$ is acyclic, where edge set $\overline{E} = E \cap (\overline{V} \times \overline{V}) = \{a,c\}$ – in more intuitive terms, $\overline{G}$ is obtained by deleting the bottom two nodes $(21,9,5)$, $(25,5,9)$ of Figure 2b, and deleting all edges b,e,f,d,g,h incident to these two nodes. In contrast, a solution feedback arc set (FAS) problem is given by $\overline{E} = E - \{f,h\} = \{a,b,c,d,e,g\}$ obtained by deleting edges f,h from the conflict digraph $G = (V,E)$ of Figure 2b. The resulting digraph $\overline{G} = (V, E - \{f,g\})$ is acyclic.

We now trace Algorithm 1. The set BP_1 from equation (5) consists of the base pairs $(1,13)$, $(33,41)$, $(49,57)$ in s that do not touch any base pair of t, and hence cannot be removed by applying a shift move. Lines 1–3 of Algorithm 1 result in updating the value of the variable s by removing these three base pairs. Lines 4–6 construct the conflict digraph $G = (V,E)$ shown in Figure 2. Line 7 invokes the IP solver to maximize the number of vertices of V subject to constraints (†) and (‡). The maximum size subset of V that satisfies (†) is simply a solution of VAS. Constraint (‡) additionally requires that $|flatten(\mathbf{v}_1 \cap flatten(\mathbf{v}_2| \leq 1$ for all $\mathbf{v}_1, \mathbf{v}_2 \in \overline{V}$. Observe that from the original vertex set V in the conflict digraph of Figure 2b, $|flatten(B) \cap flatten(C)| = |\{21,25\}| = 2$, $|flatten(B) \cap flatten(D)| = |\{5,21\}| = 2$, $|flatten(B) \cap flatten(E)| = |\{5,25\}| = 2$, $|flatten(C) \cap flatten(D)| = |\{9,21\}| = 2$, $|flatten(C) \cap flatten(E)| = |\{9,25\}| = 2$. Thus, although $\{A,B,C\}$ is a solution of VAS, it is <u>not</u> a solution of both constraints (†) and (‡) – a maximum size set $\overline{V} \subseteq V$ that satisfies both (†) and (‡) is $\overline{V} = \{A,B\}$.

Line 11 of Algorithm 1 computes $\overline{\text{Cover}} = \{(5,9)\}$, i.e. base pair $(5,9)$ is the only <u>uncovered</u> base pair of s, which is removed from s by lines 12,13. Line 14 applies topological sorting to establish the following total ordering of vertices in $\overline{V}$: (1) vertex A or $(37,29,17)$, (2) vertex B or $(5,25,21)$. Lines 15–16 result in the shift $(17,29)$ to $(29,37)$, followed by the shift $(21,25)$ to $(5,25)$. Lines 17–18 add any remaining base pairs of $t - s$ to s, resulting in adding base pairs $(9,21)$ and $(45,53)$. This yields an 8-step MS_2 trajectory consisting of 4 base pair removals, 2 shifts and 2 base pair additions (not shown due to space constraints).

```
     GAAAGAAAUAAACAAAGAAAGAAACAAACAAAGAAAGAAACAAAGAAAGAAACAAACA
     1234567890123456789012345678901234567890123456789012345678

0.   (...(...)...)...(...(...)...)...(.......).......(.......).  initial structure
1.   ................(...(...)...)...(.......).......(.......).  remove (1,13)
2.   ................(...(...)...)...................(.......).  remove (33,41)
3.   ................(...(...)...).............................  remove (49,57)
4.   (...........)...(...(...)...)...(.......).......(.......).  remove (5,9)
5.   ....................(...)...(.......).....................  shift (17,29) -> (29,37)
6.   ....(...................)...(.......).....................  shift (21,25) -> (5,25)
7.   ....(...(...........)...)...(.......).....................  add (9,21)
8.   ....(...(...........)...)...(.......).......(.......).....  add (45,53)
```

Bistable switch

As nontrivial example, consider the 34 nt bistable switch with RNA sequence ACAGGUUCGC CUGUGUUGCG AACCUGCGGG UUCG taken from Figure 1(b).2 of [8], in which the authors performed structural probing by comparative imino proton NMR spectroscopy. Figures 3a, 3b, and 3c respectively depict the metastable secondary structure s having free energy -14.00 kcal/mol, the minimum free energy (MFE) secondary structure t having free energy of -14.70 kcal/mol, and the MFE conflict digraph. In the MFE conflict digraph $G = (V, E)$, vertices are triplet nodes (x, y, z), where (unordered) base pair $\{y, z\} \in s$ belongs to the metastable structure, and (unordered) base pair $\{x, y\} \in t$ belongs to the MFE structure. A direct edge $(x, y, z) \to (u, v, w)$ occurs if $\{y, z\} \in s$ touches or crosses $\{u, v\} \in t$. The conflict digraph $G = (V, E)$ for this bistable switch contains 11 vertices, 71 directed edges, and 92,114 directed cycles. The MS_2 distance is 13, consisting of 4 base pair removals, 7 shifts and 2 base pair additions. The the corresponding minimum length trajectory follows. This optimal MS_2 trajectory contains 4 base pair removals, 2 base pair additions, and 7 base pair shifts.

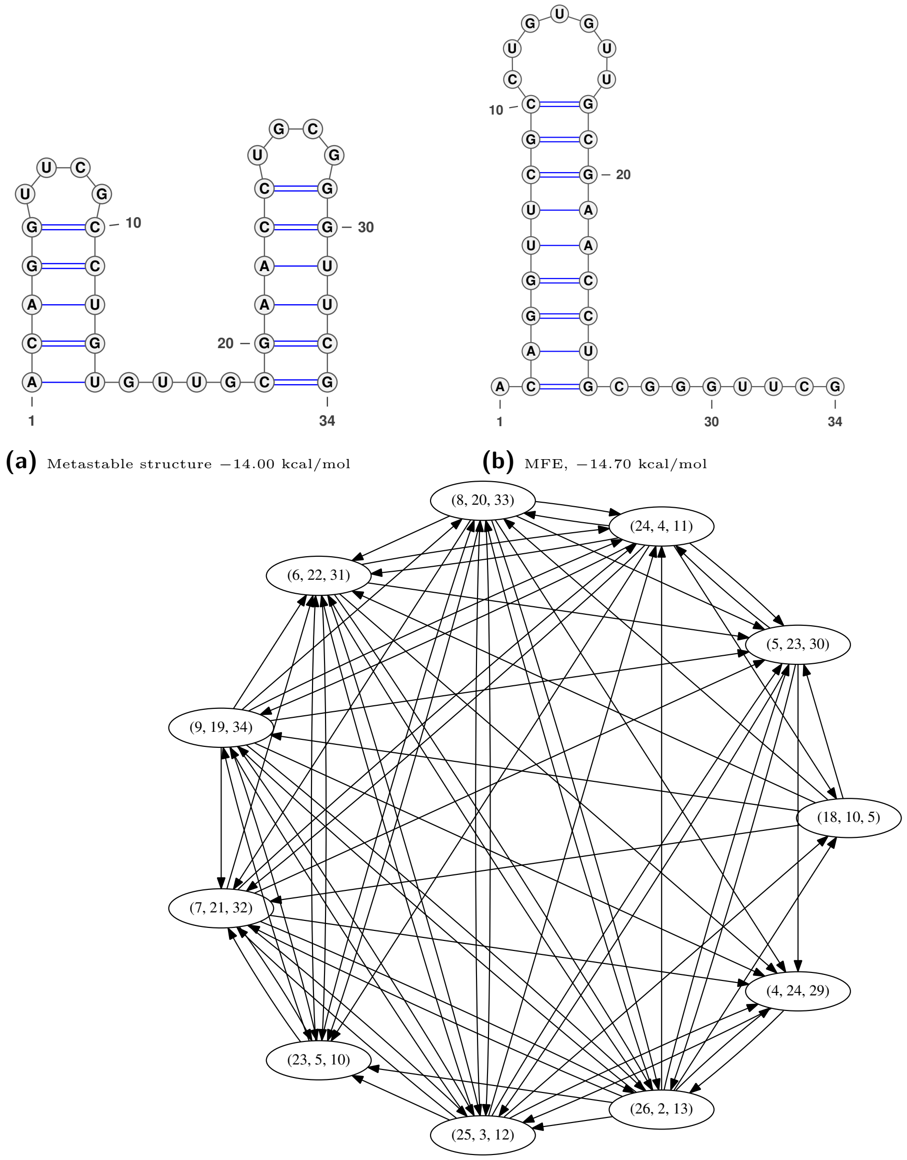
```
     ACAGGUUCGCCUGUGUUGCGAACCUGCGGGUUCG
     1234567890123456789012345678901234

 0.  (((((....)))))....((((((....))))))    initial structure
 1.  .((((....)))).....((((((....))))))    remove (1,14)
 2.  .((.(....).)).....((((((....))))))    remove (4,11)
 3.  ..(.(....).)......((((((....))))))    remove (2,13)
 4.  ....(....)........((((((....))))))    remove (3,12)
 5.  .........(.......)((((((....))))))    shift (5,10)  -> (10,18)
 6.  ........((.......))(((((....))))).    shift (19,34) -> (9,19)
 7.  .......(((.......)))((((....))))..    shift (20,33) -> (8,20)
 8.  ......((((.......))))(((....)))...    shift (21,32) -> (7,21)
 9.  .....(((((.......)))))((....))....    shift (22,31) -> (6,22)
10.  ....((((((.......))))))(....).....    shift (23,30) -> (5,23)
11.  ...(((((((.......))))))).........    shift (24,29) -> (4,24)
12.  .(.(((((((.......))))))).)........    add (2,26)
13.  .(((((((((.......)))))))))........    add (3,25)
```

Details concerning our fast, near-optimal algorithm will be presented elsewhere; however, since Figures 4 and 5 compare the performance of the exact IP (optimal) algorithm with that of the near-optimal algorithm, we briefly sketch the idea behind the method.

▶ **Algorithm 2** (Near-optimal MS_2 distance from s to t).
Input: *Secondary structures* s, t *for RNA sequence* $a_1, \ldots, a_n$
Output: *Folding trajectory* $s = s_0, s_1, \ldots, s_m = t$*, where* $s_0, \ldots, s_m$ *are secondary structures, for each* $i = 1, \ldots, m$*,* s_i *is obtained from* s_{i-1} *by a single base pair addition, removal or shift, and* m *is an approximation to* MS_2 *distance between* s *and* t*.*

The idea is to generate all equivalence classes $[x]$ with respect to equivalence relation $\equiv$, defined to be the reflexive, transitive closure of $\sim$, where for $x, y \in [1, n]$, we say $x \sim y$ if $\{x, y\} \in s$ or $\{x, y\} \in t$. We consider a *coarse-grain* digraph, whose vertices are the equivalence classes $[x]$, and whose directed edges $[x] \to [y]$ are defined if a base pair from s that lies in $[x]$ crosses a base pair from t that lies in $[y]$. Solve the *feedback arc* problem (not feedback vertex problem) for this coarse-grained digraph using IP, where we note that the number of cycles is dramatically smaller than that for the exact IP algorithm. Apply topological sorting on the coarse-grained acyclic digraph after removal of feedback arcs. Subsequently process each equivalence class by using the exact IP algorithm. Due to space



(a) Metastable structure −14.00 kcal/mol

(b) MFE, −14.70 kcal/mol

(c) RNA conflict digraph, where s is metastable, t is MFE structure

Figure 3 Conflict digraph for toy example (a) and for the 34 nt bistable switch (b,c,d) with sequence ACAGGUUCGC CUGUGUUGCG AACCUGCGGG UUCG taken from Figure 1(b).2 of [8], in which the authors performed structural probing by comparative imino proton NMR spectroscopy. (a) Toy example used in a first example of Algorithm 1. (b) Metastable structure having next lowest free free energy (after that of minimum free energy structure) of −14.00 kcal/mol. (c) Minimum free energy (MFE) structure having −14.70 kcal/mol. (d) RNA conflict digraph $G = (V, E)$, having directed edges $(x, y, z) \to (u, v, w)$ if the (unordered) base pair $\{y, z\} \in s$ touches or crosses the (unordered) base pair $\{u, v\} \in t$. Here, s is in the metastable structure shown in (a) having −14.00 kcal/mol, while t is the MFE structure shown in (b) having −14.70 kcal/mol. The conflict digraph represents a necessary order of application of shift moves, in order to avoid the creation of base triples or pseudoknots in the optimal trajetory being constructed. The conflict digraph G has 11 vertices, 71 directed edges and 92,114 directed cycles.

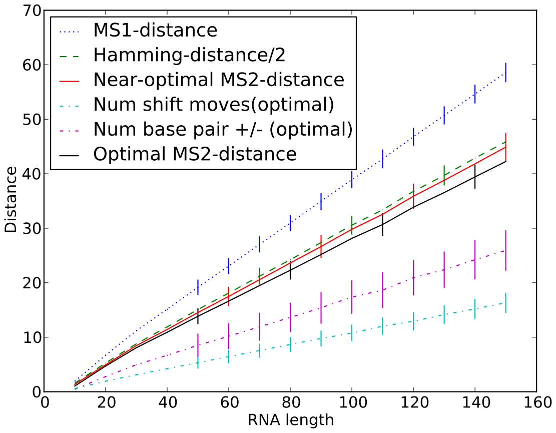


Figure 4 Benchmarking statistics for optimal and near-optimal algorithm to compute minimum length MS_2 folding trajectories between random secondary structures s, t of random RNA sequences of variable lengths. For each sequence length $n = 10, 15, 20, \cdots, 150$ nt, twenty random RNA sequences were generated of length n, with probability of 1/4 for each nucleotide. For each RNA sequence, twenty secondary structures s, t were uniformly randomly generated so that 40% of the nucleotides are base-paired. It follows that the benchmarking dataset consisted of $20 \cdot \binom{20}{2} = 3800$ many triples $\mathbf{a}, s, t$, where $\mathbf{a} = a_1, \ldots, a_n$ denotes an RNA sequence of length n, and s, t are random secondary structures of $\mathbf{a}$. Using this dataset, consisting of $20 \cdot \binom{20}{2} = 3800$ many triples $\mathbf{a}, s, t$, where $\mathbf{a} = a_1, \ldots, a_n$ denotes an RNA sequence of length n, and s, t are random secondary structures of $\mathbf{a}$, the average MS_2 distance was computed for both the exact IP Algorithm 1 and the near-optimal algorithm, whose details cannot be described due to space constraints. In addition to MS_2 distance computed by the exact IP and the near-optimal algorithm, the figure displays MS_1 distance (also known as base pair distance), Hamming distance over 2, and provides a breakdown of the MS_1 distance in terms of the number of base pair addition/removal moves "num base pair +/- (optimal)" and the shift moves "num shift moves (optimal)".

constraints, we cannot provide additional details for the near-optimal algorithm, which will be described elsewhere.

3 Discussion and an application

Computational approaches to the problem of RNA secondary structure folding kinetics involve one of three approaches: (1) computation of energy-optimal folding pathways [13, 5, 4], (2) solution of the master equation [11] to determine the time necessary to reach equilibrium [19, 16], (3) repeated simulations using the Gillespie algorithm [6] as in the software `Kinfold` [5] and `KINEFOLD` [20].

An *energy-optimal* folding pathway is a sequence $s = s_0, s_1, \ldots, s_m = t$ of secondary structures from initial structure s to target structure t, such that each intermediate structure s_i is obtained from its predecessor s_{i-1} by application of a move from a specified move set, and such that the *maximum* energy difference $E(s,t) = \max_{i=1,\ldots,m} (E(s_i) - E(s_0))$ between an intermediate structure and the initial structure is the *minimum* possible value, when taken over all possible folding trajectories – this energy difference $E(s,t)$ is called the *barrier energy*. Intuitively, an energy-optimal folding trajectory is analogous to an alpine walk between two points A and B, for which the walker reaches the minimum possible intermediate altitude, and the barrier energy is analogous to the difference between the altitude at the

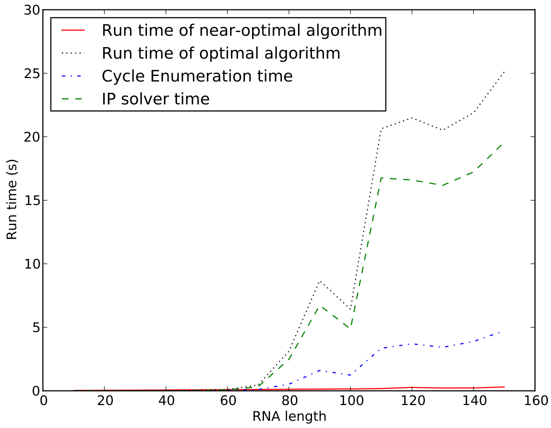


Figure 5 Run time for the exact IP (optimal) algorithm 1 and the near-optimal algorithm 2 to compute minimum length MS_2 folding trajectories for the same data set from previous Figure 4. Each data point represents the average $\mu \pm \sigma$ where error bars indicate one standard deviation, taken over 3800 sequence, structure pairs. Run time of the optimal algorithm depends almost entirely on the time to enumerate all directed cycles, using our C++ implementation of Johnson's algorithm [9] as well as time for the Gurobi IP solver.

mountain pass and that at ground camp A. The problem of computing the barrier energy is NP-complete, even for the trivial energy model of -1 per base pair [17]. After calling the program `RNAsubopt -e E` to generate all secondary structures, whose free energy is within E kcal/mol of the minimum free energy (MFE), the program `barriers` [5] uses a "flooding" procedure to determine an energy-optimal folding trajectory (run time of `RNAsubopt -e E` is exponential in the user-input energy parameter E). Note that `barriers` allows move sets MS_1 and MS_2, but that both the initial and target structure must be *locally optimal*, where a locally optimal structure has the property that no structure obained by applying one move from the move set yields a structure with strictly lower energy. The structures s, t for the previous toy example RNA are not locally optimal, in contrast to the structures s, t for the the previous bistable switch.

The program `RNAtabupath` [4] is a local search method using the *tabu* heuristic [7] which provides a very fast, near-optimal solution for the barrier energy and energy-optimal folding trajectory. Note that `RNAtabupath` does not require that initial and target structures be locally optimal, but at present only computes near-optimal MS_1 trajectories.de Another application of `RNAtabupath` is that the true MS_1 barrier energy is bounded above by the `RNAtabupath` barrier energy estimate, and hence can be used as energy parameter for `barriers`; i.e. ethe user need not use trial-and-error when entering an energy parameter that exceeds the barrier energy in order to generate an energy-optimal folding trajectory – a very time-consuming, manual procedure. Analogously, one can use Algorithm 1 to provide an energy parameter that exceeds the MS_2 barrier energy in order to generate an energy-optimal MS_2 folding trajectory using `barriers`.

Figure 6 shows the free energy profile of the shortest MS_2 folding trajectory returned by Algorithm 1 for the 34 nt bistable switch with sequence ACAGGUUCGC CUGUGUUGCG AACCUGCGGG UUCG, which sequence comes from Figure 1(b).2 of [8]. The barrier energy for the shortest MS_2 trajectory computed by Algorithm 1 is 10.8 kcal/mol with trajectory length 13. The figure also shows the nearly energy-optimal folding MS_2 trajectory

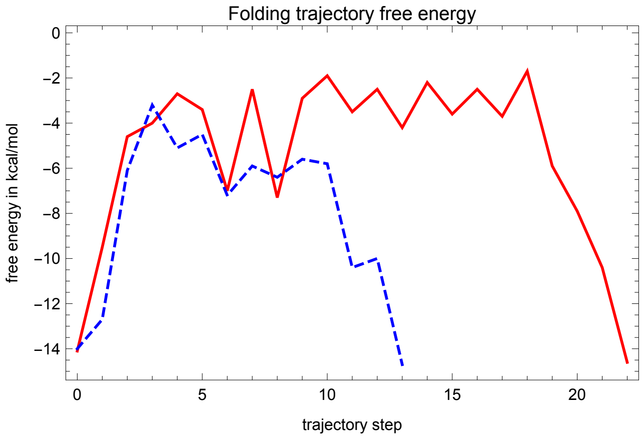


Figure 6 Free energy in kcal/mol of secondary structures appearing in RNA folding trajectories of the bistable switch, whose sequence is given in Figure 1(b).2 of [8]. The dashed blue line corresponds to the minimum length MS_2 trajectory computed by Algorithm 1; the solid red line corresponds to the lowest energy MS_1 folding trajectory found by the program `tabuPath` described in [4] in 100 folding attempts. The barrier energy for MS_1 trajectories estimated by the near-optimal software `tabuPath` (without shift) is 12.4 kcal/mol with trajectory length 23. The barrier energy for the shortest MS_2 trajectory computed by Algorithm 1 is 10.8 kcal/mol with trajectory length 13.

of `RNAtabupath`. Figure 7 shows the Arrhenius tree returned by `barriers` when run with energy parameter $10.8 + (14.7 - 14.0) = 11.5$ kcal/mol, where the value 10.8 is the energy barrier for the trajectory returned by Algorithm 1, -14.7 [resp. -14.0] is the free energy of initial [resp. target] structure s [resp. t]. Since t is the minimum free energy (MFE) structure, the program `RNAsubopt -e 11.5` will generate close to the smallest set of structures which guarantee that the program `barriers` can find an energy-optimal folding trajectory from s to t. Figures 6 and 7 consider the small 34 nt bistable switch sequence, for display purposes; clearly this approach becomes much more practical for long RNA sequences when using the near-optimal Algorithm 2.

4 Conclusion and discussion

In this paper, we have presented the first algorithms to compute the MS_2 distance between any two secondary structures s, t of a given RNA sequence. Despite the impressive speed and (approximate) accuracy of our near-optimal algorithm 2, we conjecture that the problem of computing a minimum-length MS_2 trajectory is NP-hard. This is due to several reasons: (1) the complexity of the exact IP algorithm, (2) the dramatic increase in the number of simple cycles in RNA conflict digraphs, as sequence length increases (not shown), (3) the dramatic increase in run time required by the Gurobi IP solver for sequences of increasing length (not shown), (4) the fact that FVS and FAS are NP-complete problems for several known families of digraphs. Initial investigations (omitted here) have shown the that family of RNA conflict digraphs is distinct from a host of graph families, for which the computational complexity of FVS/FAS is known; however, at present it is unclear whether there is a polynomial time algorithm to determine whether a given digraph is representable as an RNA conflict digraph.

The complexity of MS_2 distance suggests that simple simulation studies of RNA structural evolution and robustness [18] are unlikely to be extended to consider shift moves, despite the experimental evidence for particular shift moves such as helix zippering and defect

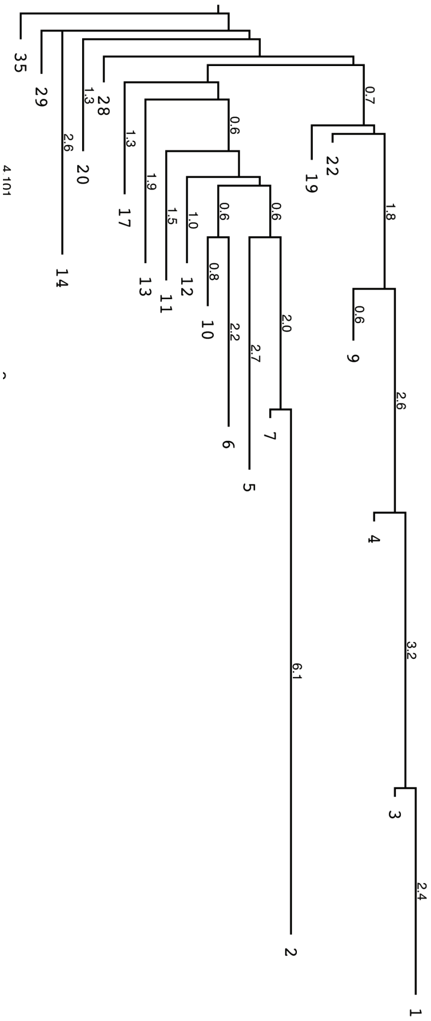


Figure 7 Arrhenius tree produced by running Vienna RNA Package programs `RNAsubopt -s -e 11.5` and `barriers` to obtain an optimal folding pathway. The energy bound of 11.5 kcal/mol was selected, because the free energy of initial structure s [resp. target structure t] is -14.0 [resp. -14.6] kcal/mol, and the barrier energy of the shortest MS_2 trajectory from the left panel of this figure is 10.8 kcal/mol. It follows that we know there exists a folding trajectory within $10.8 + (14.7 - 14.0) = 11.5$ kcal/mol of the MFE structure t. The advantage of first running Algorithm 1 is that knowledge of the energy barrier for the shortest MS_2 path allows an efficient computation of `RNAsubopt` and `barriers` – in the current case, `RNAsubopt` only needed to generate 1556 structures, and to find 28 saddle structures. The number of structures for this 34 nt bistable switch is 845,139,060,165 $\approx 8.45 \cdot 10^{11}$.

diffusion [14]. Moreover, studies of RNA structural evolution from [15] used Hamming distance as a simple approximation to MS_2 distance, which we now know from Figure 4 not to be particularly accurate. Finally, some very interesting, yet complex questions are raised concerning graph theory (which digraphs are representable as conflict digraphs), computational complexity (whether MS_2 distance is NP-hard), and potentially related group theoretic questions.

References

1 P. Clote. Expected degree for RNA secondary structure networks. *J. Comput. Chem.*, 0(O):O, November 2014.

2 P. Clote and A. Bayegan. Network Properties of the Ensemble of RNA Structures. *PLoS One*, 10(10):e0139476, 2015.

3 T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Algorithms.* McGraw-Hill, 1990. 1028 pages.

4 I. Dotu, W. A. Lorenz, P. VAN Hentenryck, and P. Clote. Computing folding pathways between RNA secondary structures. *Nucleic. Acids. Res.*, 38(5):1711–1722, 2010.

5 C. Flamm, I. L. Hofacker, P. F. Stadler, and M. Wolfinger. Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, 216:155–173, 2002.

6 D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.*, 22(403):403–434, 1976.

7 F. W. Glover and M. Laguna. *Tabu Search.* Springer-Verlag, 1998. 408 p.

8 C. Hobartner and R. Micura. Bistable secondary structures of small RNAs and their structural probing by comparative imino proton NMR spectroscopy. *J. Mol. Biol.*, 325(3):421–431, January 2003.

9 D. B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.*, 4:77–84, 1975.

10 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York*, pages 85–103, 1972. URL: `http://www.cs.berkeley.edu/~luca/cs172/karp.pdf`.

11 A. Kolmogoroff. Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung. *Mathematische Annalen*, 104:415–458, 1931.

12 R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker. Viennarna Package 2.0. *Algorithms. Mol. Biol.*, 6:26, 2011.

13 S. R. Morgan and P. G. Higgs. Barrier heights between ground states in a model of RNA secondary structure. *J. Phys. A: Math. Gen.*, 31:3153–3170, 1998.

14 D. Pörschke. Model calculations on the kinetics of oligonucleotide double-helix coil transitions: Evidence for a fast chain sliding reaction. *Biophys Chem*, 2(2):83–96, August 1974.

15 P. Schuster and P. F. Stadler. Modeling conformational flexibility and evolution of structure: RNA as an example. In U. Bastille, M. Roman, and M. Vendruscolo, editors, *Structural Approaches to Sequence-Evolution*, page 3–36. Springer, Heidelberg, 2007.

16 X. Tang, B. Kirkpatrick, S. Thomas, G. Song, and N. M. Amato. Using motion planning to study RNA folding kinetics. *J. Comput. Biol.*, 12(6):862–881, July/August 2005.

17 C. Thachuk, J. Maňuch, L. Stacho, and A. Condon. NP-completeness of the direct energy barrier height problem. *Natural Computing*, 10(1):391–405, 2011.

18 A. Wagner. Robustness and evolvability: a paradox resolved. *Proc. Biol Sci.*, 275(1630):91–100, January 2008.

19 Michael T. Wolfinger, W. Andreas Svrcek-Seiler, Christoph Flamm, Ivo L. Hofacker, and Peter F. Stadler. Efficient folding dynamics of RNA secondary structures. *J. Phys. A: Math. Gen.*, 37:4731–4741, 2004.

20 A. Xayaphoummine, T. Bucher, and H. Isambert. Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots. *Nucleic. Acids. Res.*, 33(Web):W605–W610, July 2005.