RNA-RNA interaction is NP-complete and some approximation algorithms

Saad Mneimneh
Visiting Professor, Computer Science, Hunter College of CUNY
695 Park Avenue, New York, NY 10021
saad@alum.mit.edu

dedicated to my beautiful wife Nora for her encouragement and to Agnel (a biologist) for putting up with a "computer scientist who talks biology"

Abstract

We formulate the problem of RNA-RNA interaction which consists of finding the optimal structure of two RNAs that are folding and binding to each other simultaneously. We show that the problem is NP-complete, provide some basic approximation algorithms, and show some experimental results with satisfactory structure prediction.

Keywords: RNA-RNA interaction, approximation algorithms.

1 Introduction

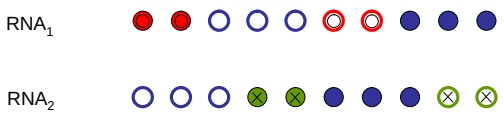
RNA-RNA interaction is a very important phenomenon in molecular biology and plays a crucial role in various biological processes. An example of RNA-RNA interaction is RNA interference (RNAi), where a small interfering RNA (known as siRNA) can be used to silence a given gene by targeting its messenger RNA: The siRNA binds to the RNA and triggers a cascade of events that would eventually destroy it (Post Transcriptional Gene Silencing by RNAi), e.g. [2] and [4].

Although RNAs are single-stranded linear molecules, they usually fold. The problem of RNA folding has been studied extensively in the literature, and many polynomial time algorithms for determining the optimal folding (e.g. with maximum number of bonds, or more generally with minimum energy) of an RNA molecule have been developed [10], [12], [13]. However, the simultaneous folding and binding of two RNAs has not been studied yet. This interplay between individual RNA folding and RNA-RNA binding is very common. For instance, while RNAi may be viewed as a special case of RNA-RNA interaction since siRNAs are 19-21 nucleotide long and do not generally fold, other complex examples of RNA-RNA interaction exist where both RNAs fold (see Section 7). Therefore, the general problem of RNA-RNA interaction where folding and binding occur simultaneously deserves great attention.

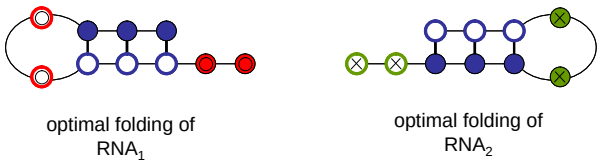
Section 2 motivates the RNA-RNA interaction problem through a toy example. Section 3 gives a precise formulation of the problem to be solved. Section 4 shows that this problem is NP-complete. Section 5 describes some basic approximation algorithms to the problem. Section 6 provides experimental results for siRNAs that support the simultaneous folding and binding of RNA-RNA interaction. Section 7 provides experimental results for RNA-RNA interaction using the algorithms of Section 5 and shows satisfactory structure prediction. Finally, we conclude in Section 8.

2 A toy example

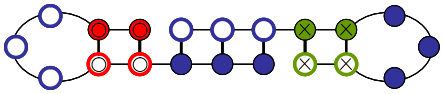
Consider the following two RNAs where nucleotides are abstractly represented by colors/patterns. A solid color/pattern can bond to the same non-solid color/pattern.



The two RNAs can independently fold into two optimal structures as show below. Each RNA can have up to three bonds optimally. The total number of bonds in this case is six.



However, if the two RNAs can interact and form external bonds, then we may obtain a better structure (in terms of the number of bonds) yielding seven bonds in total, as shown below (another possibility is to bind the loops instead of the tails, i.e. kissing loops):



Therefore, to capture the simultaneous folding and binding of this interaction, one can form an RNA-RNA interaction graph where every edge represents a possible bond (either internal to the RNA itself, or external to the other RNA). The RNA-RNA interaction graph for the example above is illustrated in Figure 1 (with edges of the above structure emphasized).

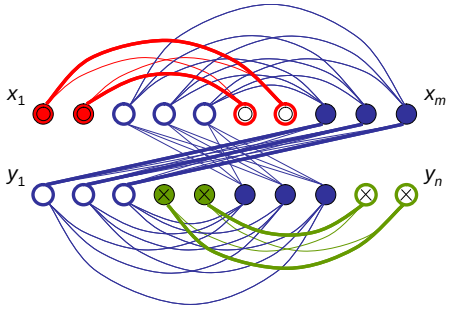


Figure 1: RNA-RNA interaction graph (solution emphasized)

The problem then becomes to identify a maximum cardinality set (or more generally a set with maximum weight as described in Section III) of non-intersecting edges [1] (edges sharing a node are also intersecting). This particular formulation of avoiding intersection is motivated by two facts:

- Pseudoknots are rare in RNA folded structures [10]
- RNA-RNA binding occurs between a sense (5' to 3') molecule and an anti-sense (3' to 5') molecule (so it is also not likely to have knotted interactions)

3 The RNA-RNAi problem

We generalize the idea explored in Section II and include a weight for every possible bond. Therefore, the RNA-RNAi [2] problem is the following: Given an RNA-RNA interaction graph (as in Figure 1) where nodes are partitioned into two ordered sets $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$, and every edge e has a weight $w(e) \in \mathbb{Q}$, find a set of **node disjoint** edges S that maximizes $\sum_{e \in S} w(e)$ such that (intersection is avoided):

- if $(x_i, x_j) \in S$ and $(x_k, x_l) \in S$ then NOT $i < k < j < l$
- if $(y_i, y_j) \in S$ and $(y_k, y_l) \in S$ then NOT $i < k < j < l$

[1] Intersection is interpreted here given the particular realization of the graph, i.e. the graph cannot be redrawn to avoid intersection.

[2] The terminology RNA-RNAi, which stands for RNA-RNA interaction, is not to be confused with RNAi (RNA interference).

- if $(x_i, y_j) \in S$ and $(x_k, y_l) \in S$ then NOT ($i < k$ and $j < l$)

Therefore, the nucleotides of RNA_1 are represented by the ordered elements of X, and the nucleotides of RNA_2 are represented by the ordered elements of Y. Therefore, it is convenient to consider RNA_1 to be the string $x = x_1 \dots x_m$, and RNA_2 to be the string $y = y_1 \dots y_n$. We refer to the edges of the RNA-RNA interaction graph connecting both RNAs, i.e. of the form (x_i, y_j), as binding edges. We also refer to the edges of the RNA-RNA interaction graph internal to a given RNA, i.e. of the form (x_i, x_j) or (y_i, y_j), as folding edges for that RNA. A solution to RNA-RNAi (whether optimal or not) is said to have a weight equal to its achievable sum. As in the case of RNA folding problems, this weighted formulation provides a **basic** model for real RNA-RNA interaction problems where every bond contributes a specific energy. In an RNA world, weights are negative and the objective is to minimize the energy, i.e. the sum of weights, but this is an equivalent formulation.

A special case of RNA-RNAi is the uniformly weighted RNA-RNAi where the weights are all the same. For a uniformly weighted RNA-RNAi, maximizing $\sum_{e \in S} w(e)$ is equivalent to maximizing the cardinality of S (as described in the toy example of Section II). Even the special case of a uniformly weighted RNA-RNAi (the decision version) is NP-complete [8]. For convenience, the following section shows the NP-completeness of RNA-RNAi.

4 RNA-RNAi is NP-complete

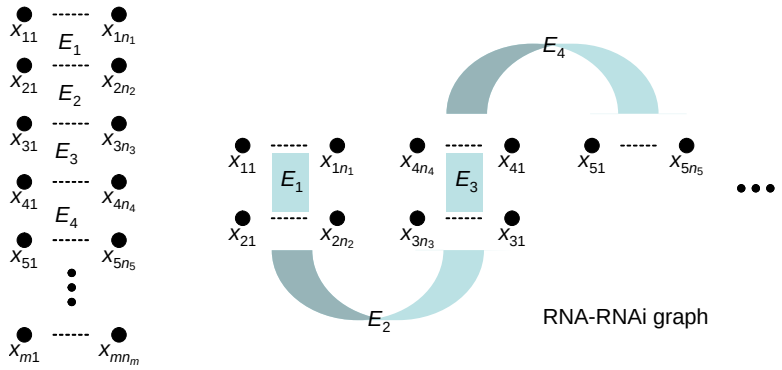
We show that RNA-RNAi (the decision version) is NP-complete by a reduction from 3SAT to a problem that we call simple RNA-RNAi, and a reduction from simple RNA-RNAi to RNA-RNAi. The complete reduction from 3SAT to RNA-RNAi will produce a uniformly weighted RNA-RNAi.

4.1 The simple RNA-RNAi problem

The simple RNA-RNAi problem consists of a graph where nodes are partitioned into m ordered sets $X_1 = \{x_{11}, \dots, x_{1n_1}\}, \dots, X_m = \{x_{m1}, \dots, x_{mn_m}\}$, and every edge e has a weight $w(e) \in \mathbb{Q}$ and connects a node in X_i to a node in X_{i+1} for some i, i.e. of the form $(x_{ij}, x_{(i+1)k})$. The set of all such edges for a given i is denoted by E_i. The objective is to find a set of **node disjoint** edges S that maximizes $\sum_{e \in S} w(e)$ such that (intersection is avoided): if $(x_{pi}, x_{(p+1)j}) \in S$ and $(x_{pk}, x_{(p+1)l}) \in S$ then NOT ($i < k$ and $j < l$).

4.2 A reduction from simple RNA-RNAi to RNA-RNAi

We now describe a polynomial time reduction from simple RNA-RNAi to RNA-RNAi. We place $x_{11}, \dots, x_{1n_1}$ as the first nucleotides of RNA_1, i.e. first batch of nodes in X. We then place $x_{21}, \dots x_{2n_2}$ as the first nucleotides of RNA_2, i.e. first batch of nodes in Y. This is followed by $x_{3n_3}, \dots x_{31}$ (in reverse order) in Y. We then place $x_{4n_4}, \dots x_{41}$ (in reverse order) as the second batch of nodes in X. This is followed by $x_{51} \dots x_{5n_5}$ in X. All edges in E_i are kept intact for all $i = 1 \dots m-1$. We continue in this way until all nodes in $X_1 \dots X_m$ have been placed. We obtain an RNA-RNA interaction graph as illustrated in the figure below, where $X = \{x_{11}, \dots, x_{1n_1}, x_{4n_4}, \dots, x_{41}, x_{51}, \dots, x_{5n_5}, x_{8n_8}, \dots, x_{81}, \dots\}$ and $Y = \{x_{21}, \dots, x_{2n_2}, x_{3n_3}, \dots, x_{31}, x_{61}, \dots, x_{6n_6}, x_{7n_7}, \dots, x_{71}, \dots\}$.



All edges in E_i for odd i become binding edges, and all edges in E_i for even i become folding edges. It can be easily verified that two edges are intersecting in simple RNA-RNAi if and only if they are intersecting in RNA-RNAi obtained above. Therefore, a solution S to RNA-RNAi (that maximizes $\sum_{e \in S} w(e)$) provides a solution to simple RNA-RNAi. Note that when applied to a uniformly weighted simple RNA-RNAi, this construction results in a uniformly weighted RNA-RNAi. Next we show that 3SAT is polynomially reducible to simple RNA-RNAi.

4.3 A reduction from 3SAT to (uniformly weighted) simple RNA-RNAi

For simplicity of illustration, we first present a reduction from 3SAT to simple RNA-RNAi. We then show how to relax the weight assumption and make all the weights equal to 1. The reduction consists of $2n-1$ pairs $\mathbb{X}_1 = (X_1, X_2), \ldots, \mathbb{X}_i = (X_{2i-1}, X_{2i}), \ldots, \mathbb{X}_{2n-1} = (X_{4n-3}, X_{4n-2})$, where n is the number of variables in 3SAT and each X_i is an ordered set of nodes in simple RNA-RNAi.

$\mathbb{X}_i$ for odd i: The odd pairs represent the variables of 3SAT (i.e. n pairs). Therefore, $\mathbb{X}_i$ (i odd) represents variable $x_{(i+1)/2}$ of 3SAT. Moreover, each $\mathbb{X}_i = (X_{2i-1}, X_{2i})$ (i odd) encodes a representation of every clause of 3SAT in the form of a left and a right triplets of pairwise intersecting edges in E_{2i-1} (e.g. $(x_{(2i-1)2}, x_{(2i)3})$, $(x_{(2i-1)3}, x_{(2i)2})$, and $(x_{(2i-1)4}, x_{(2i)1})$ is such a triplet). The left set of triplets represent the clauses when variable $x_{(i+1)/2}$ is set to 0. Similarly, the right set of triplets represent the clauses when variable $x_{(i+1)/2}$ is set to 1. Some edges of the triplets may be missing depending on the structure of the 3SAT formula: if $x_{(i+1)/2}$ is the j^{th} ($j = 1 \ldots 3$) variable of a clause and setting $x_{(i+1)/2}$ to 0 creates only a 0 in the clause, then the j^{th} edge of the corresponding left triplet in E_{2i-1} is missing. Similarly, if setting $x_{(i+1)/2}$ to 1 creates only a 0 in the clause, then the j^{th} edge of the corresponding right triplet in E_{2i-1} is missing.

In addition, two intersecting edges in E_{2i-1} will allow for selecting among the left and right sets of triplets (see Figure 2); this simulates an assignment to the 3SAT variable $x_{(i+1)/2}$. Therefore, these are called assignment edges.

$\mathbb{X}_i$ for even i: The even pairs, i.e. $\mathbb{X}_i = (X_{2i-1}, X_{2i})$ for even i, consist of a complete triplet of pairwise intersecting edges in E_{2i-1} for every clause in 3SAT, and connects to the triplets of $\mathbb{X}_{i-1}$ and $\mathbb{X}_{i+1}$ through edges in $E_{2(i-1)}$ and E_{2i} respectively in a universal way (independent of the 3SAT formula) as shown in Figure 2. These edges are called satisfiability edges. In addition, two edges in $E_{2(i-1)}$ and four edges in E_{2i} will allow for selecting among different sets of satisfiability edges (see Figure 2). These edges are called blocking edges.

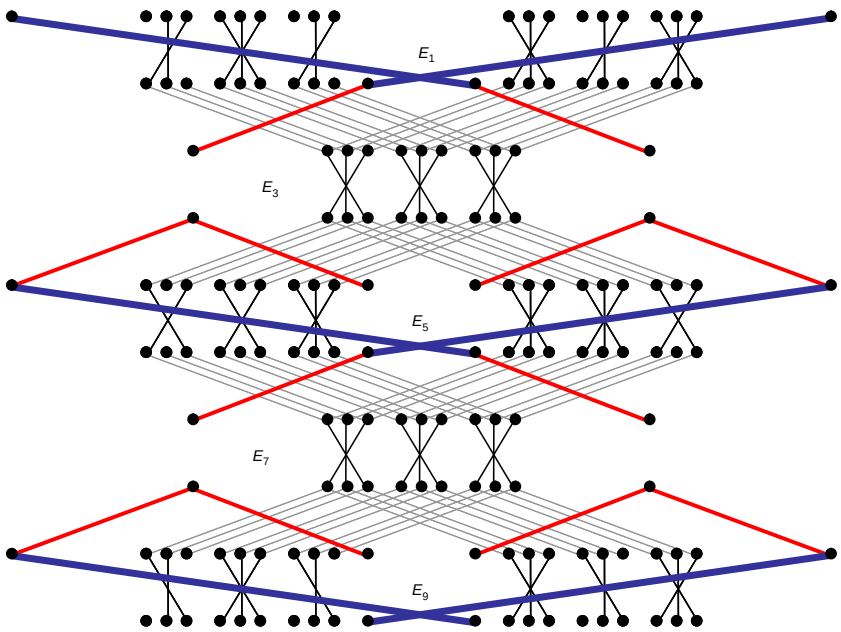


Figure 2: A reduction from $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3)$ to simple RNA-RNAi. The first triplet on each side corresponds to the first clause, and so on. Note that in this example, the numbering of variables and their order of appearance in the clauses coincide, but this needs not be in general.

For now, the edges (triplet, assignment, satisfiability, and blocking) of the obtained simple RNA-RNAi will have different weights. We list them below in decreasing order of weight (decreasing order of thickness in Figure 2).

- assignment edges in E_{2i-1} for odd i (in blue) with weight w_4
- blocking edges in $E_{2(i-1)}$ and E_{2i} for even i (in red) with weight w_3
- triplet edges in E_{2i-1} for all i (in black) with weight w_2
- satisfiability edges in $E_{2(i-1)}$ and E_{2i} for even i (in gray) with weight $w_1 = 1$

The edge weights are constructed such that $w(e) > \sum_{w(e')<w(e)} w(e')$ is true for every edge e. We have $12(n-1)c$ satisfiability edges so $w_2 > 12(n-1)c$. We have at most $(9n-3)c$ triplet edges so $w_3 > [(9n-3)c+1]w_2$. The additional 1 in the expression is to account for all the satisfiability edges. We have $6(n-1)$ blocking edges so $w_4 > [6(n-1)+1]w_3$. Again, the additional 1 in the expression is to account for all triplet and satisfiability edges. Therefore, the weights can be chosen to be integers of order $O(n^3c^2)$.

By construction of the weights, the optimal solution must contain one of the the assignment edges in each E_{2i-1} for odd i. This corresponds to making an assignment for each variable in 3SAT. The choice of assignment edges then dictates the addition of one blocking edge in E_{2i} and two blocking edges in E_{2i+2} for odd i. The total weight achieved so far is $nw_4 + 3(n-1)w_3$. Moreover, WLOG, we have:

- For each variable $x_{(i+1)/2}$ of 3SAT, only one set of free triplets exists in E_{2i-1} (corresponding to its assignment)
- All triplets corresponding to E_{2i-1} with even i are free
- Only one set of free satisfiability edges connecting $\mathbb{X}_i$ to $\mathbb{X}_{i+1}$ exists and these edges connect the free triplets of E_{2i-1} to the free triplets of E_{2i+1}, $i = 1 \ldots 2n-2$

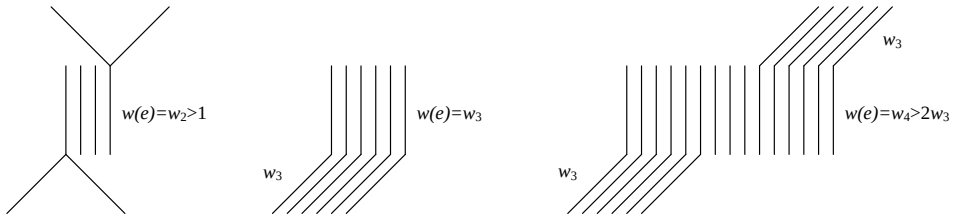
It is easy to verify that a total weight of $nw_4 + 3(n-1)w_3 + (2n-1)cw_2 + 4(n-1)cw_1$ is achievable if and only if the 3SAT formula is satisfiable:

If the 3SAT formula is satisfiable, then consider a satisfying assignment and the corresponding choice of assignment/blocking edges. Let the j^{th} $(j = 1 \ldots 3)$ variable of a given clause be the variable that satisfies the clause under this assignment. We add the j^{th} edge of the corresponding free triplet in each E_{2i-1} for odd i (this edge must exist by construction), and the symmetric edge (i.e. $(4-j)^{th}$) of the corresponding triplet in each E_{2i-1} for even i. This adds a weight of $(2n-1)w_2$ per clause. Then it is possible to add two satisfiability edges on each side of every triplet, achieving the claimed total weight (this is optimal because every free triplet contributes an edge to the solution).

If a weight of $nw_4 + 3(n-1)w_3 + (2n-1)cw_2 + 4(n-1)cw_1$ is achievable, then consider the assignment corresponding to the choice of assignment/blocking edges. By induction on i, all free triplets corresponding to the same clause pick the same edge in all E_{2i-1} for odd i. For a given clause, if that's the j^{th} $(j = 1 \ldots 3)$ edge, then the j^{th} variable in that clause satisfies the clause under the given assignment (otherwise, the corresponding edge would have been missing from one of the triplets).

We now describe a relaxation of the weights. We simply replace an edge of weight w with w non-intersecting edges of weight 1 and replicate the incident nodes as needed; the nodes thus created are placed next to their original nodes. Since all weights are polynomial in the size of the original 3SAT problem, the new (uniformly weighted) simple RNA-RNAi instance is also polynomial in the size of the 3SAT problem.

The only artifact of this transformation is that the solution may now pick *partial* edges. For instance, since an edge e with weight w is transformed to w edges of weight 1, the solution does not guarantee that these w edges are only picked together. However, one can perform the transformation without worrying about this detail because the transformation will always produce in the worst case one of the following three scenarios for edge e:



(a) triplet edge (b) blocking edge (c) assignment edge

Therefore, if any edge among the edges of e is picked, the solution can be changed to pick all of them, without changing the number of edges in the solution. This argument can now be applied recursively starting from the partially picked edge with the largest weight.

Since a solution to simple RNA-RNAi is a solution to the newly obtained (uniformly weighted) simple RNA-RNAi, and any partial solution to the newly obtained (uniformly weighted) simple RNA-RNAi can be made non-partial, the two optimal solutions must have the same value.

Theorem 1 *RNA-RNAi (the decision version) is NP-complete* [3] *(even when uniformly weighted).*

Proof: RNA-RNAi is in NP (a solution can be verified in polynomial time). 3SAT (which is NP-complete) is polynomially reducible to (uniformly weighted) simple RNA-RNAi and (uniformly weighted) simple RNA-RNAi is polynomially reducible to (uniformly weighted) RNA-RNAi. ■

5 Basic approximation algorithms

In this section we provide some basic constant factor approximation algorithms for the RNA-RNAi problem. Recall from Section III the definitions of binding edges and folding edges. Also recall that RNA_1 can be represented as the string $x = x_1 \dots x_m$, and RNA_2 can be represented as the string $y = y_1 \dots y_n$.

5.1 A 1/2 factor approximation algorithm

Consider the structures obtained from performing the following (using the given weight function):

- Optimally solve RNA-RNAi while ignoring binding edges
- Optimally solve RNA-RNAi while ignoring folding edges for both RNAs

The first step corresponds to optimally folding RNA_1 and RNA_2 independently. The second step corresponds to optimally aligning [4] RNA_1 and RNA_2.

Optimal folding and optimal alignment are both well studied problems and can be solved in polynomial time. Optimally folding an RNA of length n takes $O(n^3)$ time and $O(n^2)$ space [10], and optimally aligning two RNAs of lengths m and n respectively takes $O(mn)$ time [9] and linear space [5].

The important observation is that one of the two obtained structures has a weight equal to at least 1/2 the weight of the optimal solution for the corresponding RNA-RNAi problem. Let w_1 and w_2 be the weights achieved by the two structures respectively. Let OPT be the weight of the optimal solution S.

Lemma 1 $\max(w_1, w_2) \geq \frac{1}{2}OPT$.

Proof: Consider the optimal solution. Let A be the sum of weights of edges (bonds) formed in the folded part of RNA_1, i.e. edges in S of the form (x_i, x_j). Let B be the sum of weights of edges (bonds)

[3] Since the RNA-RNA interaction graph is not restricted, the problem assumes infinite nucleotide alphabet; however, it is possible to slightly change the reduction to make the alphabet finite.

[4] By alignment, we signify the binding resulting from aligning RNA_1 and the complement of RNA_2 with a zero scoring gap function. For instance *cgga* and *gccu* align perfectly. This is equivalent to finding the largest weight common subsequence of RNA_1 (of string $x = x_1 \dots x_m$) and the complement of RNA_2 (of string $y = y_1 \dots y_n$).

formed by the alignment part of RNA_1 and RNA_2, i.e. edges in S of the form (x_i, y_j). Let C be the sum of weights of edges (bonds) formed in the folded part of RNA_2, i.e. edges in S of the form (y_i, y_j). Then $OPT = A + B + C$ (the weight of the optimal solution). By the optimality of the two structures, $w_1 \geq A + C$ and $w_2 \geq B$. Therefore, $2\max(w_1, w_2) \geq w_1 + w_2 \geq A + C + B = OPT$.

■

Obviously, in the independent folding of the RNAs, the remaining (non-folded) nucleotides of the RNAs may be aligned. On the other hand, in the alignment, the remaining (non-binding) nucleotides of the RNAs may be folded independently.

A 1/2 factor approximation algorithm:

- optimally fold RNA_1 and RNA_2 independently
  (optimally align their remainders)
- optimally align RNA_1 and RNA_2
  (optimally fold their remainders independently)
- choose the structure with the maximum weight

5.2 A 2/3 factor approximation algorithm

Consider the structures obtained from performing the following (using the given weight function):

- Optimally solve RNA-RNAi while ignoring binding edges
- Optimally solve RNA-RNAi while ignoring the folding edges for RNA_2
- Optimally solve RNA-RNAi while ignoring the folding edges for RNA_1

As before, the first step corresponds to optimally folding RNA_1 and RNA_2 independently. The second step corresponds to optimally folding RNA_1 while interacting with the non-folding RNA_2. Similarly, the third step corresponds to optimally folding RNA_2 while interacting with the non-folding RNA_1.

One of the three obtained structures has a weight equal to at least 2/3 of the weight of the optimal solution for the corresponding RNA-RNAi problem. Let w_i, for $i = 1 \ldots 3$, be the weight achieved by the three structures respectively. Let OPT be the weight of the optimal solution S.

Lemma 2 $\max(w_1, w_2, w_3) \geq \frac{2}{3}OPT$.

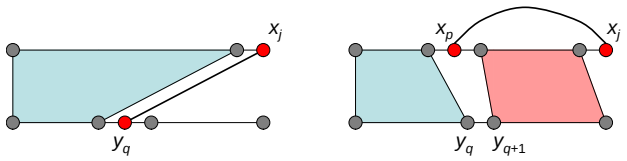
Proof: Let A, B, and C be defined as in the proof of Lemma 1 where $OPT = A + B + C$. By the optimality of the three structures, $w_1 \geq A + C$, $w_2 \geq A + B$, and $w_3 \geq B + C$. Therefore, $3\max(w_1, w_2, w_3) \geq w_1 + w_2 + w_3 \geq A + C + A + B + B + C = 2(A + B + C) = 2OPT$.

■

As stated in Section V.A, optimal folding is a well studied problem and can be solved in $O(n^3)$ time for an RNA of length n [10]. Therefore, the only concern now is to show that it is possible to optimally fold RNA_1 while interacting with a non-folding RNA_2 in polynomial time. This can be done by a dynamic programming algorithm. Let the strings $x = x_1 \ldots x_m$ and $y = y_1 \ldots y_n$ denote the folding and non-folding RNAs respectively. Let $V(i, j, k, l)$ denote the weight that can be achieved in the optimal solution for the substrings $x[i..j]$ and $y[k..l]$. Then we have three possibilities for x_j: x_j does not bond, x_j bonds with some y_q (edge $(x_j, y_q) \in S$) and $k \leq q \leq l$, or x_j bonds with some x_p (edge $(x_p, x_j) \in S$) and $i \leq p < j$. Therefore, we have the following dynamic programming algorithm to compute $V(1, m, 1, n)$. The last two cases are also illustrated pictorially.

$$V(i,j,k,l) = \max \begin{cases} V(i, j-1, k, l) \\ V(i, j-1, k, q-1) + w(x_j, y_q) \\ V(i, p-1, k, q) + \\ V(p+1, j-1, q+1, l) + w(x_{p \neq j}, x_j) \end{cases}$$

where $i \leq p \leq j$ and $k \leq q \leq l$ and w is the weight function. If $k < l$ we set $V(i, j, k, l) = F_x(i, j)$, the weight of optimally folding $x[i..j]$. The actual structure (i.e. S) can be obtained by standard dynamic programming bookkeeping/backtracking methods.



Noting that each case divides the problem into two independent sub-problems i.e. substrings (with one of them being possibly empty), where folding binds only the extremities, the formulation above can be simplified as follows:

$$V(i,j,k,l) = \max \begin{cases} V(i+1,j-1,k,l) + w(x_{i\neq j}, x_j) \\ V(i,p,k,q) + V(p+1,j,q+1,l) \end{cases}$$

where $i-1 \leq p \leq j$ and $k-1 \leq q \leq l$ and $(p \neq i-1 \vee q \neq k-1) \wedge (p \neq j \vee q \neq l)$, $V(i,j,k,k-1) = F_x(i,j)$, $V(i,i-1,k,l) = 0$, and $V(i,i,k,k) = \max(0, w(x_i, y_k))$.

We have $O(m)$ values for p and $O(n)$ values for q and hence $V(i,j,k,l)$ requires $O(mn)$ time to compute. Since we have $O(m^2)$ substrings of x and $O(n^2)$ substrings of y, this algorithm runs in $O(m^3n^3)$ time and $O(m^2n^2)$ space.

In the independent folding of the RNAs, the remaining (non-folded) nucleotides of the RNAs may be aligned. On the other hand, in a folding/alignment, the remaining (non-binding) nucleotides of the non-folding RNA may be folded.

A 2/3 factor approximation algorithm:

- optimally fold RNA_1 and RNA_2 independently (optimally align their remainders)
- optimally fold RNA_1 while interacting with RNA_2 and ignore folding for RNA_2 (optimally fold the remainder of RNA_2)
- optimally fold RNA_2 while interacting with RNA_1 and ignore folding for RNA_1 (optimally fold the remainder of RNA_1)
- choose the structure with the maximum weight

5.3 A note on the approximation factor of dynamic programming

In the previous sections, we relied on dynamic programming algorithms (through alignments and foldings) to obtain constant factor approximations for RNA-RNAi. Therefore, a legitimate question is whether better constant approximation factors can be obtained using such algorithms. The answer to this question is negative. Any dynamic programming algorithm for RNA-RNAi that recursively divides the problem into independent sub-problems (i.e. substrings) cannot achieve a constant approximation factor better than 2/3 (even for uniformly weighted RNA-RNAi). To establish this fact, we introduce the concept of an *entangler*.

Definition 1 (Entangler) *An entangler is a set of five edges that contains two folding edges (x_i, x_j) and (y_k, y_l), and three binding edges e_1, e_2, and e_3, such that:*

- $e_1 = (x_p, y_q) \Rightarrow p \in (i,j), q \notin (k,l)$
- $e2 = (x_p, y_q) \Rightarrow p \in (i,j), q \in (k,l)$
- $e3 = (x_p, y_q) \Rightarrow p \notin (i,j), q \in (k,l)$

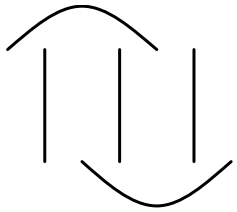


Figure 3: Entangler

A dynamic programming algorithm that recursively divides the problem into independent sub-problems cannot generate a solution with an entangler. The argument is simple: there is no way to break an entangler into independent sub-problems (at least one edge must be excluded).

One can definitely design a dynamic programming algorithm that computes the optimal entangler-free solution. It will be similar to the dynamic programming formulation described in the previous section, but performing symmetrically on both RNAs and allowing both RNAs to fold (and interact):

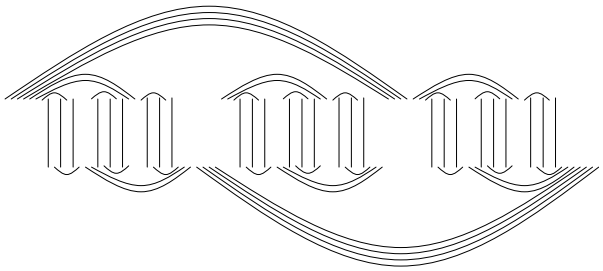
$$V(i,j,k,l) = \max \begin{cases} V(i+1,j-1,k,l) + w(x_{i\neq j}, x_j) \\ V(i,p,k,q) + V(p+1,j,q+1,l) \\ V(i,j,k+1,l-1) + w(y_{k\neq l}, y_l) \end{cases}$$

where $i-1 \leq p \leq j$ and $k-1 \leq q \leq l$ and $(p \neq i-1 \vee q \neq k-1) \wedge (p \neq j \vee q \neq l)$, $V(i,j,k,k-1) = F_x(i,j)$, $V(i,i-1,k,l) = F_y(k,l)$, and $V(i,i,k,k) = \max(0, w(x_i, y_k))$.

The running time and space requirements of the above algorithm are still $O(m^3n^3)$ and $O(m^2n^2)$ respectively. It is easy to show that any entangler-free solution can be recursively broken into independent sub-problems as dictated by the above dynamic programming formulation, and hence, this algorithm computes the optimal entangler-free solution. This algorithm is also a 2/3 factor approximation algorithm because all three solutions described at the beginning of Section V.B are entangler-free.

Now we exhibit an instance of the RNA-RNAi problem where every entangler-free solution is asymptotically at most a 2/3 factor approximation. This proves the claim stated at the beginning of this section.

Given an integer $r > 0$, the instance consists of 3^r non-intersecting binding edges partitioned into three groups (of 3^{r-1} edges each) by 2^{r-1} non-intersecting folding edges on each side, i.e. of the form (x_i, x_j) and (y_k, y_l) respectively. Then each of the three groups is recursively partitioned in the same way. The partitioning stops when we obtain a single entangler, i.e. when $r = 1$. We assume all edges have the same weight (i.e. an instance of uniformly weighted RNA-RNAi). The following figure illustrates the instance for $r = 3$.



It is easy to show that the number of folding edges on each side is given by the following expression:

$$\sum_{i=0}^{r-1} 3^{r-1-i}2^i = 3^{r-1}\sum_{i=0}^{r-1}(\frac{2}{3})^i = 3^{r-1}\frac{1-(2/3)^r}{1-2/3} = 3^r - 2^r$$

Therefore, the optimal solution contains $3^r + (3^r - 2^r) + (3^r - 2^r) = 3^{r+1} - 2^{r+1}$ edges (all of them). Obviously, there is an entangler-free solution with $3^r + (3^r - 2^r) = 2 \cdot 3^r - 2^r$ edges (it excludes the folding edges on one side). This is not the only possible entangler-free solution; however, we claim that any entangler-free solution must exclude at least $3^r - 2^r$ edges. Assuming this claim is true, the

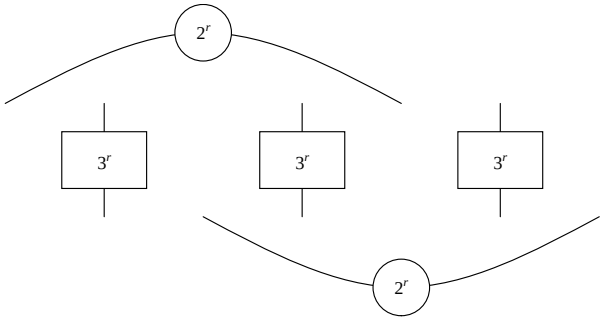
approximation factor of an entangler-free solution is at most

$$\frac{2 \cdot 3^r - 2^r}{3^{r+1} - 2^{r+1}} = \frac{2}{3} + \epsilon$$

where $\lim_{r \to \infty} \epsilon = 0$.

Theorem 2 *An entangler-free solution for the (uniformly weighted) RNA-RNAi problem is asymptotically at most a* $2/3$ *factor approximation.*

Proof: For the instance corresponding to a given r, we prove that any entangler-free solution must exclude at least $3^r - 2^r$ edges, by induction on r. The base case is when $r = 1$, i.e. the instance is just an entangler. Therefore, at least $3^1 - 2^1 = 1$ edge must be excluded. Now assume the claim is true up to some value r. The instance corresponding to $r + 1$ can be viewed as follows:



The three rectangular sets represent the binding edges. The two circular sets represent the folding edges. Since the solution is entangler-free, at least one of these five sets must be excluded. If a circular set is excluded, the number of excluded edges is at least $3^r - 2^r$ for each of the three sub-problems (inductive hypothesis) in addition to 2^r edges for a circular set. Hence, the number of excluded edges is at least $3(3^r - 2^r) + 2^r = 3^{r+1} - 2^{r+1}$. If a rectangular set is excluded, the number of excluded edges is at least $3^r - 2^r$ for two sub-problems (inductive hypothesis) in addition to 3^r edges for one rectangular set. Hence, the number of excluded edges is at least $2(3^r - 2^r) + 3^r = 3^{r+1} - 2^{r+1}$. This proves the induction, and hence the theorem.

■

Note that this theorem is a tight characterization of entangler-free solutions because there is always an entangler-free solution for the RNA-RNAi problem that achieves a $2/3$ factor approximation.

6 RNA interference (RNAi) as an RNA-RNA interaction

Since siRNAs are small RNAs (19-21 nucleotide long) and do not generally fold, one may view RNAi as an RNA-RNA interaction where one of the two RNAs (the siRNA) does not fold. Therefore, one may project further that folding and binding occur simultaneously in RNAi. For that matter, we considered siRNA efficiencies reported in [11] for the human cyclophilin RNA. A number of optimal and near optimal folding structures for this RNA is then obtained using mfold [12], [13]. Based on these structures, and for each nucleotide, the probability of bonding in the folded structure is computed (assuming for simplicity that all structures are equally likely). Therefore, the higher the probability for a given nucleotide, the more drastic the effect of that nucleotide binding an siRNA on the structure of the RNA. On the other hand, the energy of binding is computed for each individual siRNA (identified by its first $3'$ end nucleotide) reported in [11]. The lower the energy, the more stable the binding of the siRNA to the RNA molecule. This energy is used (after being scaled appropriately) to adjust the probability computed above (using averaging). As shown in Figure 4, the result (solid line) is a good approximation for the actual siRNA efficiency pattern (dashed line) reported experimentally in [11] [5]. We also assessed the significance of this approximation by performing alignment on random vectors in the interval $[0, 1]$. After ignoring the very low efficiency values (see Footnote 4), the probability that two random vectors align better than the two shown below is about 0.005.

[5] The very low efficiency values, that do not agree with their corresponding computed values, seem to be due to inherent properties of the siRNA sequence itself, such as a high GC content or a long stretch of A/T.

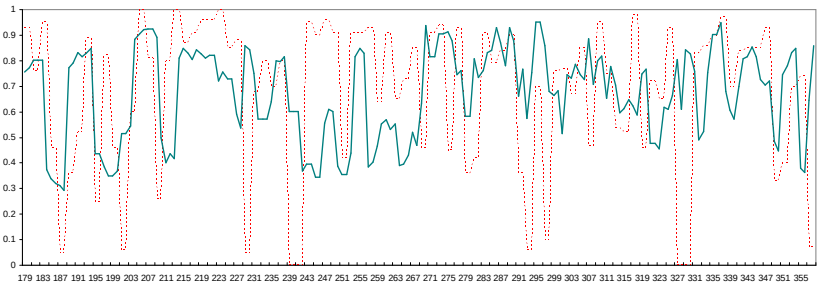
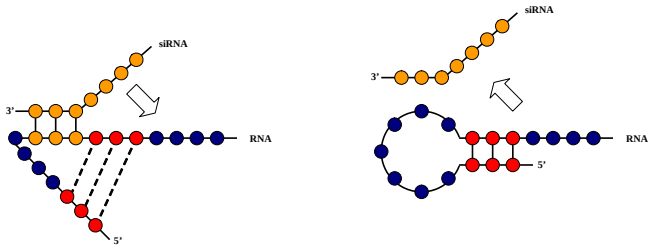
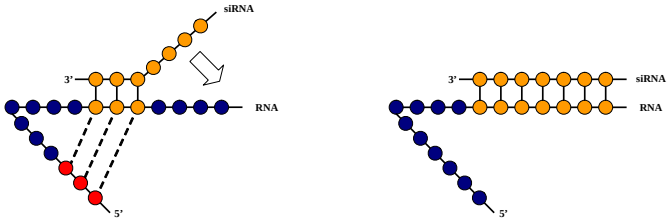


Figure 4: Approximation of siRNA efficiency pattern

One possible interpretation of the result is the following: The stability of the siRNA binding is crucial to trigger the destruction of the RNA. Therefore, to be effective, the siRNA requires a certain amount of binding time, starting at its $3'$ end. Meanwhile, if the RNA can still fold optimally, it forces a rapid dissociation from the siRNA to form the optimal structure, hence not giving the siRNA the time needed to stabilize.



On the other hand, if the siRNA binds to a nucleotide that is supposed to bond (with high probability) in the optimal structure of the RNA, the optimality of the RNA structure is perturbed. The RNA is not stable enough to rapidly dissociate from the siRNA. Therefore, the siRNA will find enough time (if the binding energy is low enough) to trigger the cascade of events that would eventually lead to the destruction of the RNA (while possibly still folding non-optimally).



Therefore, not only our result shows that the computational technique described above is a good rationale for siRNA design, but also it suggests a possibility of simultaneous folding and binding in RNAi as one would expect in a general RNA-RNA interaction.

7 Experimental results for RNA-RNAi

We performed the basic algorithms of Section V on two example RNA-RNAi problems; fhlA-OxyS interaction [3] and CopA-CopT interaction [6] in the Escherichia coli bacteria. Although the algorithms theoretically achieve constant approximation factors, not every solution obtained by the algorithms is realistic. For instance, RNAs do not fold sharply and tend to fold locally. Moreover, two RNAs are likely to interact using complementary blocks of certain sizes. As heuristics, we constrain the folding and alignment in the following ways: if x_i binds to x_j (edge $(x_i, x_j) \in S$), then $4 \leq |i - j| \leq 50$. Moreover, if x_p binds to y_q (edge $(x_p, y_q) \in S$), then $p \in [i, j]$ and $q \in [k, l]$ such that:

- $j - i = l - k = B - 1$
- x_{i+r} binds to y_{k+r} for all $r = 0 \dots B - 1$
- x_{i-1} and x_{j+1} do not bind to y, and y_{k-1} and y_{l+1} do not bind to x

Therefore, the alignment is modified to compute an optimal block alignment. The details of the modified algorithms are not included (a variation on the first dynamic programming formulation of Section V.B to allow lower and upper bounds on B), but the modifications are simple and do not affect the theoretical complexity of the algorithms.

For weights, we used $w(g, u) = 1$, $w(a, u) = 2$, and $w(g, c) = 3$ (which are reasonably proportional to the energy values at 37° [14]). We performed the algorithm of Section V.A on fhlA-OxyS with $7 \leq B < \infty$ as acceptable block sizes. We obtained the structure illustrated in Figure 4 which is almost identical to the known structure of fhlA-OxyS [3] (small differences in folding around the first binding site). Stretches in the middle of the RNAs (9 nucleotides for fhlA, and 43 nucleotides for OxyS) were ignored for better prediction, because they were not reported to fold or bind [3]. Keeping those stretches maintains the same binding sites and loops of Figure 4; however, results in one additional binding site and a number of additional loops, which cannot be avoided computationally due to the optimization nature of the problem.

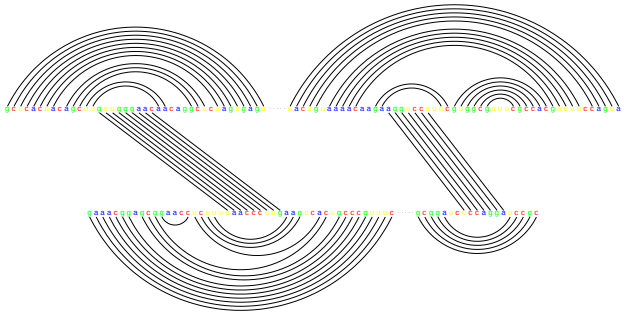


Figure 5: fhlA-OxyS

Since CopA and CopT are complementary, performing the algorithm of Section V.A will produce the trivial solution where both RNAs bind completely to form a double strand. One can possibly multiply $w(x_i, y_j)$ by an appropriate value $\alpha < 1$ (reducing the weight contribution of external bonds) to minimize this effect; however, the solution will switch from the trivial double strand to the trivial two separate folded RNAs. Therefore, the algorithm of Section V.B (or that of Section V.C) is more appropriate.

Even with such algorithm, we still have to use a multiplicative factor $\alpha < 1$ as described above; however, doing so will now generate a non-trivial solution. We performed the algorithm of Section V.B with $\alpha = 1/3$ and $4 \leq B < \infty$. We allowed smaller block sizes here because the two RNAs are complementary (binding is more likely). We obtained the structure illustrated in Figure 5 which is very close to the known structure of CopA-CopT [6]. Namely, the folding in the middle parts should be replaced by binding, and the folding and binding of the extremities should be ignored. Again, the latter cannot be avoided computationally due to the optimization nature of the problem.

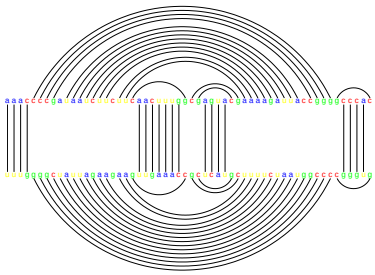


Figure 6: CopA-CopT

Although the choice of a block size B is important, several block sizes may be tried in the neighborhood of some expected or desired size, and the best structure may be picked. Note that a smaller block size (less constrained) does not necessarily imply a better result because the algorithms perform a local optimization

followed by a completion on the remaining parts (see description of algorithms in Section V.A and Section V.B). Figure 6 below shows the variation in weight for the solutions of fhlA-OxyS and CopA-CopT (using the corresponding algorithms described above) when changing the block size (the lower bound) from 1 to 10, and gives a justification for the choices ($B = 7$ and $B = 4$) made above.

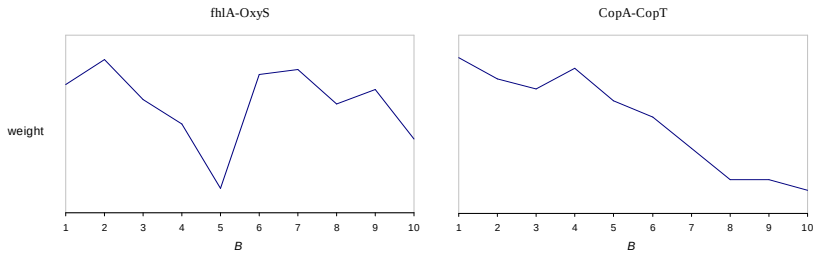


Figure 7: Effect of block size B on total weight

More generally, the stacked pair energy model [7] may be used instead, which favors block formation in both the alignment and the folding, and improves prediction of RNA secondary structure [7]. In principle, the dynamic programming algorithms can be changed to reflect the stacked pair energy model. But the main focus of this paper is on the approximability of the **basic** RNA-RNAi problem described in Section III (but the results can be extended to other formulations).

8 Conclusion

We formulated the RNA-RNA interaction problem and proved it is NP-complete in general. Some constant factor approximation algorithms exist but there is a need for better algorithms in terms of running time, space, and approximability. In particular, we proved an upper bound of 2/3 on the approximation factor of dynamic programming based algorithms. Experimental results validate the concept of RNA-RNA interaction, and the described approximation algorithms provide satisfactory structure prediction, which is crucial for understanding the effect of RNA-RNA interaction on many biological processes.

Acknowledgement

The author would like to thank: The CSE department at SMU for allowing him to develop and teach a course on computational biology, William Westerman for early discussions, Nassim Sohaee from SMU for helping with initial experimentation, Steve Crozier and Skip Garner from UTSW for valuable discussions, Virginia Teller from Hunter College of CUNY for office space, and Ioannis Stamos from Hunter College of CUNY for computer resources.

Disclaimer

Most of this work appears in the SMU CSE Technical Report 04-CSE-03 [8]. While writing this manuscript, similar results were published [1].

References

[1] Alkan C, Karakoc E, Nadeau JH, Sahinalp C, Zhang K. RNA-RNA interaction prediction and antisense RNA target search. *RECOMB 2005*, Cambridge MA, May 14-18, 2005.

[2] Ambion.com, The mechanism of RNA interference. *http://www.ambion.com/techlib/RNAi_mechanism.html*.

[3] Argaman L, Altuvia S. fhlA repression by OxyS RNA: Kissing complex formation at two sites results in stable antisense target RNA complex. *Journal of Molecular Biology* July 2000; **28 300(5)**: 1101-1112.

[4] Hammond SM, Caudy AA, Hannon GJ. Post transcriptional gene silencing by double stranded RNA. *Nature Rev Gen* 2001; **2**: 110-119.

[5] Hirschberg D. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM* 1975; **18(6)**: 341-343.

[6] Kolb FA, Mamgren C, Westhof E, Ehresmann B, Wagner EG, Romby P. An unusual structure formed by antisense target RNA binding involves an extended kissing complex with a four-way junction and a side-by-side helical alignment. *RNA* March 2000; **6(3)**: 311-324.

[7] Mathews D, Sabina J, Zuker M, Turner D. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology* 1999; **288**: 911-940.

[8] Mneimneh S. RNA-RNA interaction is NP-complete. *SMU CSE Technical Report 04-CSE-03* July 2004.

[9] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 1970; **48**: 443-453.

[10] Nussinov R, Pieczenik G, Griggs JR, Kleitman DJ. Algorithms for loop matchings. *SIAM Journal of Applied Mathematics* 1978; **35**: 68-82.

[11] Reynolds A., Leake D., Boese Q., Scaringe S., Marshall WS., Khvorova A., Rational siRNA design for RNA interference. *Nature Biotechnology* **22(3)**, pp. 326-330, March 2004.

[12] Zuker M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research* 2003; **31(13)**: 3406-3415.

[13] Zuker M. On finding all suboptimal foldings of an RNA molecule. *Science* 1989; **244**: 48-52.

[14] Zuker M. RNA folding lecture notes. *http://www.bioinfo.rpi.edu/~zukerm/lectures/RNAfold-html/index.html*.