Multiple RNA Interaction: Beyond Two

Saad Mneimneh* and Syed Ali Ahmed

***Abstract*—The interaction of two RNA molecules involves a complex interplay between folding and binding that warranted recent developments in RNA-RNA interaction algorithms. However, biological mechanisms in which more than two RNAs take part in an interaction also exist. It is reasonable to believe that interactions involving multiple RNAs are generally more complex to be treated pairwise. In addition, given a pool of RNAs, it is not trivial to predict which RNAs interact without sufficient biological knowledge. Therefore, structures resulting from multiple RNA interactions often cannot be predicted by the existing algorithms that handle RNAs pairwise and may simply favor the best interacting pair. We propose a system for multiple RNA interaction that overcomes the difficulties mentioned above by formulating a combinatorial optimization problem called *Pegs and Rubber Bands*. A solution to this problem encodes a structure of interacting RNAs. The problem, not surprisingly, is NP-hard. However, our experiments with approximation algorithms and heuristics for the problem suggest that this formulation is adequate to predict known interaction patterns of multiple RNAs. In general, however, the optimal solution obtained does not necessarily correspond to the actual structure observed in biological experiments. Moreover, a structure produced by interacting RNAs may not be unique. We extend our approach to generate multiple suboptimal solutions. By clustering these solutions, we are able to reveal representatives that correspond to realistic structures. Specifically, our results on the U2-U6 complex with introns in the spliceosome of human/yeast and the CopA-CopT complex in *E. coli* are consistent with published biological structures.**

***Index Terms*—Approximation algorithm, clustering, dynamic programming, multiple RNA interaction, structure prediction, suboptimal solution.**

I. Introduction

THE interaction of two RNA molecules has been independently formulated as a computational problem in several works, e.g., [1]–[3]. In their most general form, these formulations lead to NP-hard problems (which means computationally intractable, i.e., the running time of the algorithm that produces an optimal solution increases exponentially with the problem size). To overcome this hurdle, researchers have been either reverting to approximation algorithms, or imposing algorithmic restrictions; for instance, the avoidance of the formation of certain structures computationally.

Manuscript received October 05, 2014; revised November 20, 2014; accepted December 23, 2014. Date of publication February 11, 2015; date of current version April 07, 2015. The work of S. Mneimneh was supported by NSF Award CCF-1049902. The work of S. A. Ahmed was supported by NSF Award CCF-1049902 and a Graduate Center Fellowship. *Asterisk indicates corresponding author.*

*S. Mneimneh is with Hunter College of the City University of New York, New York NY 10065 (e-mail: saad@hunter.cuny.edu).

S. A. Ahmed is with The Graduate Center of the City University of New York, New York NY 10016 (e-mail: sahmed3@gc.cuny.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNB.2015.2402591

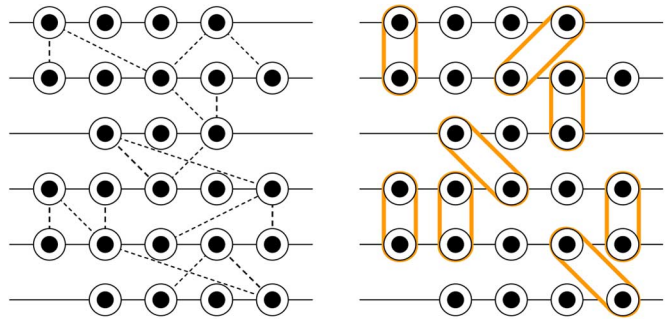

Fig. 1. Pegs and Rubber Bands. All positive weights are equal to 1 and are represented by dashed lines. The optimal solution achieves a total weight of 8.

While these algorithms had limited use in the beginning, they became important venues for (and in fact popularized) an interesting biological fact: RNAs interact. For instance, micro-RNAs (miRNAs) bind to a complementary part of messenger RNAs (mRNAs) and inhibit their translation [4].

But more complex forms of RNA-RNA interaction exist. In *E. coli*, CopA binds to the ribosome binding site of CopT, also as a regulation mechanism to prevent translation [5]; so does OxyS to fhlA [6]. In both of these structures, the simultaneous folding (within the RNA) and binding (to the other RNA) are non-trivial to be predicted as separate events. To account for this, most of the RNA-RNA interaction algorithms are based on the probability for a pair of subsequences, one of each RNA, to participate in the interaction, and in doing so they generalize the energy model used for the partition function of a single RNA to the case of two RNAs [7]–[12]. This generalization takes into consideration the simultaneous aspect of folding and binding.

Not surprisingly, however, there exist other mechanisms in which more than two RNA molecules take part in an interaction. Typical scenarios involve the interaction of multiple small nucleolar RNAs (snoRNAs) with ribosomal RNAs (rRNAs) in guiding the methylation of the rRNAs [4], and multiple small nuclear RNAs (snRNA) with mRNAs in the splicing of introns [13]. Some early attempts for multiple RNA interaction have been considered, e.g., [14], [15], but they only generalize the partition function algorithm of [16] applied to the concatenation of all RNAs as one, and so can only produce structures with no pseudoknots. While pseudoknots are rare in folded structures, they translate to kissing loops when spanning multiple RNAs, which are quite frequent. Even though algorithms for kissing loops exist, e.g., [17], advances in pairwise interaction of RNAs suggest that the latter is more adequate. Nevertheless, even with the existence of a computational framework for a single RNA-RNA interaction, it is reasonable to believe that interactions involving multiple RNAs are generally more complex to be treated pairwise. In addition, given a pool of RNAs, it is not trivial to predict which RNAs interact without some prior

1536-1241 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

$$W(i_1, i_2, \ldots, i_m) = \max \begin{cases} W(i_1 - 1, i_2, \ldots, i_m) \\ W(i_1, i_2 - 1, i_3, \ldots, i_m) \\ \vdots \\ W(i_1, \ldots, i_{m-1}, i_m - 1) \\ W(i_1 - 1, i_2 - 1, i_3, \ldots, i_m) + w(1, i_1, i_2) \\ W(i_1, i_2 - 1, i_3 - 1, i_4, \ldots, i_m) + w(2, i_2, i_3) \\ \vdots \\ W(i_1, \ldots, i_{m-2}, i_{m-1} - 1, i_m - 1) + w(m-1, i_{m-1}, i_m) \end{cases}$$

where $W(0, 0, \ldots, 0) = 0$.

Fig. 2. Dynamic programming algorithm for Pegs and Rubber Bands.

biological information. This may lead to the following computational hurdle. When the interaction pattern is unknown, and RNAs are treated pairwise, an immediate consequence is the greedy nature of the algorithm: the best interacting pair of RNAs will often dominate the solution; this in turn will lock the interaction pattern of the whole ensemble into a suboptimal state; thus preventing the correct structure from presenting itself as a solution.

We propose a mathematical formulation based on combinatorial optimization that overcomes the issues outlined above. The model derives from the premise that multiple RNA interaction may be the result of competing pairwise interactions that settle; and yet a simple pairwise handling of the RNAs will not capture that competition. And while the resulting problem is NP-hard (any reasonable formulation for multiple RNA interaction will be), it admits an approximation algorithm with provable bounds on optimality, i.e., $(1-\epsilon)$-factor of optimal for every $\epsilon > 0$. The important feature is that we capitalize on existing pairwise interaction algorithms while avoiding the possible locking problem mentioned earlier. The approach gives satisfactory prediction of the interactions in the spliceosome involving the U2-U6 complex and introns, in addition to some known pairwise interactions, such as CopA-CopT in *E. coli*. Extending the approach to generate multiple suboptimal solutions (as opposed to just one $(1-\epsilon)$-optimal solution), shows that these solutions correspond to alternative observed structures in the spliceosome of yeast and of the complex CopA-CopT in *E. coli* (and sometimes a suboptimal solution is the correct one).

II. Formulation: Pegs and Rubber Bands

We present a combinatorial optimization problem that we call Pegs and Rubber Bands [18], [19] as a framework for multiple RNA interaction. The link between the two will be made shortly following a formal description of Pegs and Rubber Bands. Consider m rows numbers 1 to m with n_l pegs in row l numbered 1 to n_l. There is an infinite supply of rubber bands, and a rubber band can be placed around two pegs in consecutive rows. For instance, we may choose to place a rubber band around peg i in row l and peg j in row $l+1$; we call it a rubber band at $[l, i, j]$. In this case, the rubber band contributes a given weight $w(l, i, j)$. The Pegs and Rubber Bands problem is to maximize the total weight by placing rubber bands around pegs in such a way that no two rubber bands overlap. Formally, each peg can have at most one rubber band around it, and if a rubber band is placed at $[l, i_1, j_1]$ and another at $[l, i_2, j_2]$, then $i_1 < i_2 \Leftrightarrow j_1 < j_2$. We assume without loss of generality that $w(l, i, j) \neq 0$ to avoid the unnecessary placement of rubber bands and, therefore, either $w(l, i, j) > 0$ or $w(l, i, j) = -\infty$. Fig. 1 shows an example.

Given an optimal solution, it can always be reconstructed from left to right by repeatedly placing some rubber band at $[l, i, j]$ such that, at the time of this placement, no rubber band is around peg $[l, k]$ for $k > i$ and no rubber band is around peg $[l+1, k]$ for $k > j$. This process can be carried out by a dynamic programming algorithm to compute the maximum weight (in exponential time), by defining $W(i_1, i_2, \ldots, i_m)$ to be the maximum weight when we truncate the rows at pegs $[1, i_1], [2, i_2], \ldots, [m, i_m]$ (see Fig. 2).

The maximum weight is given by $W(n_1, n_2, \ldots, n_m)$ and the optimal solution can be obtained by standard backtracking. When all rows have $O(n)$ pegs, this algorithm runs in $O(mn^m)$ time and $O(n^m)$ space.

A. Multiple RNA Interaction as Pegs and Rubber Bands

To provide some initial context, we describe how the formulation of Pegs and Rubber Bands, though in a primitive way now, can capture the problem of multiple RNA interaction. To make the connection, RNA sequences become the rows of pegs, the ordered pegs represent RNA bases $\{A, G, C, U\}$ in the order of occurrence in their sequence, a rubber band around pegs $[l, i]$ and $[l+1, j]$ is an interaction between the corresponding base pair, the ith base of RNA l and the jth base of RNA $l+1$, and the weights are chosen based on the (negative) energies of interacting bases (as base pairs). Those energies are obtained using pairwise interaction algorithms on the $m(m-1)/2$ pairs of RNAs, and in their own merit account for both intra- and inter- molecular energies. Therefore, $w(l, i, j)$ corresponds to the energy of binding the ith base of RNA l to the jth base of RNA $l+1$ while breaking any possible folding. This is why we do not explicitly consider the folding within RNAs, which is factored in the weights.

With the above analogy in mind, it should be clear that an optimal solution for Pegs and Rubber Bands represents the lowest energy conformation in a base-pair energy model, when a pseudoknot-like restriction is imposed on the RNA interaction (rubber bands cannot overlap). Such a restriction is natural for interacting RNA molecules. We obviously assume that an order on the RNAs is given, that they alternate in sense and antisense, and that the first RNA interacts with the second RNA, which in turn interacts with the third RNA, and so on. We later relax this ordering and condition on the interaction pattern of the RNAs. While a simple base-pairing model is not likely to give realistic results, our goal for the moment is simply to establish

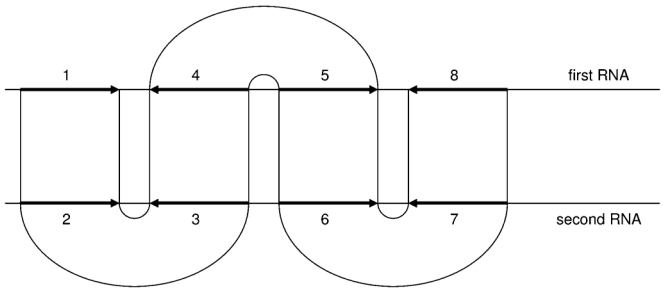

Fig. 3. Pegs and Rubber Bands as a special instance of RNA-RNA interaction, vertical lines indicate regions where only interaction (binding of the two RNAs) is allowed, and curved lines indicate regions where only folding within each RNA is allowed.

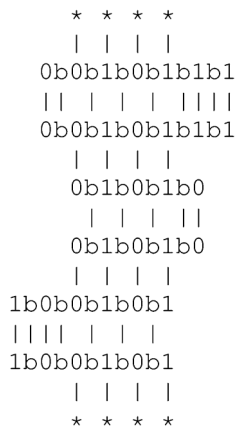

Fig. 4. Reduction from LCS for {0010111, 01010, 100101} to Pegs and Rubber Bands (the symbol | denotes a rubber band). The optimal solution with weight $2(7+5+6)-3+4=37$ corresponds to a common subsequence of length 4, namely 0101.

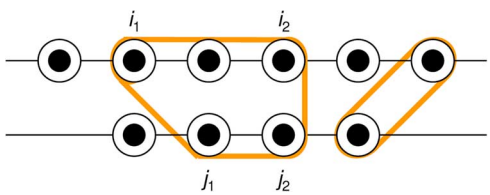

Fig. 5. A rubber band can now be placed around a window of pegs, here $u=3$ and $v=2$ in the window on the left.

a correspondence between the two problems. We will improve on this in Section III.

B. Complexity of the Problem

With the above correspondence in mind, the problem of Pegs and Rubber Bands can be viewed as a instance of a classical RNA-RNA interaction, involving only two RNAs: We construct the first as RNA 1, followed by RNA 4 reversed, followed by RNA 5, followed by RNA 8 reversed, and so on; and the second as RNA 2, followed by RNA 3 reversed, followed by RNA 6, followed by RNA 7 reversed, and so on, as shown in Fig. 3.

Therefore, Pegs and Rubber Bands can be solved as an RNA-RNA interaction problem. While this RNA-RNA interaction represents a restricted instance of the more general NP-hard problem, it is still NP-hard. In fact, Pegs and Rubber Bands itself is NP-hard.

Theorem 1: Pegs and Rubber Bands is NP-hard.

Proof: We make a reduction from the longest common subsequence (LCS) for a set of binary strings, which is an NP-hard problem. In this reduction, pegs are labeled and $w(l,i,j)$ depends only on the label of peg $[l,i]$ and the label of peg $[l+1,j]$. We describe this weight as a function of labels shortly. Each binary string is modified by adding the symbol b between every two consecutive bits. A string of original length n is then transformed into two consecutive (identical) rows of $2n-1$ pegs each, where each peg is labeled by the corresponding symbol in $\{0,1,b\}$. For any given integer k, the first and last rows consist of k pegs labeled $*$. We now define the weight as a function of labels: $w(0,0)=w(1,1)=w(b,b)=w(*,0)=w(*,1)=w(0,*)=w(1,*)=1$ and $w(x,y)=-\infty$ otherwise. It is easy to verify that the strings have a common subsequence of length k if and only if the optimal solution has a weight of $\sum_i(2n_i-1)+k=2\sum_i n_i-m+k$ (when every peg has a rubber band around it), where n_i is the original length of string i and m is the number of strings. ■

C. An Approximation Algorithm

While our problem is NP-hard, we can show that the same formulation can be adapted to obtain a polynomial time approximation. A maximization problem admits a polynomial time approximation scheme (PTAS) iff for every fixed $\epsilon>0$ there is an algorithm with a running time polynomial in the size of the input that finds a solution within $(1-\epsilon)$ of optimal [20]. We show below that we can find a solution within $(1-\epsilon)$ of optimal in time $O(m\lceil 1/\epsilon\rceil n^{\lceil 1/\epsilon\rceil})$, where m is the number of rows and each row has $O(n)$ pegs.

Theorem 2: Pegs and Rubber Bands admits a PTAS.

Proof: Let OPT be the weight of the optimal solution and denote by $W[i\ldots j]$ the weight of the optimal solution when the problem is restricted to rows $i,i+1,\ldots,j$ (a subproblem). For a given $\epsilon>0$, let $k=\lceil 1/\epsilon\rceil$. Consider the following k solutions (weights), each obtained by a concatenation of optimal solutions for subproblems consisting of at most k rows.

$$W_1 = W[1\ldots 1]+W[2\ldots k+1]+W[k+2\ldots 2k+1]+\ldots$$
$$W_2 = W[1\ldots 2]+W[3\ldots k+2]+W[k+3\ldots 2k+2]+\ldots$$
$$\vdots$$
$$W_k = W[1\ldots k]+W[k+1\ldots 2k]+W[2k+1\ldots 3k]+\ldots$$

It is easy to verify that every pair of consecutive rows appear in exactly $k-1$ of the above subproblems. Therefore, if we sum up $W_1,\ldots,W_k$, we cover the optimal solution $k-1$ times; and since each $W[i\ldots j]$ is optimal for the subproblem given by rows $i,i+1,\ldots,j$, we have:

$$\sum_{i=1}^{k} W_i \geq (k-1)OPT$$
$$\Rightarrow \max_i W_i \geq \frac{k-1}{k}OPT \geq (1-\epsilon)OPT$$

If m is the total number of rows, then there are $O(m)$ subproblems of at most k rows each and, therefore, the running time required to find $\max_i W_i$ when every row has $O(n)$ pegs is $O(mkn^k)=O(m\lceil 1/\epsilon\rceil n^{\lceil 1/\epsilon\rceil})$. ■

For a given integer k, the $(1-1/k)$-factor approximation algorithm is to simply choose the best $W_i = W[1\ldots i]+W[i+1\ldots i+k]+W[i+k+1\ldots i+2k]+\ldots$ as a solution. Some

$$W(i_1,i_2,\ldots,i_m)=\max\begin{cases} W(i_1-1,i_2,\ldots,i_m)\\ W(i_1,i_2-1,i_3,\ldots,i_m)\\ \vdots\\ W(i_1,\ldots,i_{m-1},i_m-1)\\ \\ W((i_1-u-g)^+,(i_2-v-g)^+,i_3,\ldots,i_m)+w(1,i_1,i_2,u,v)\\ W(i_1,(i_2-u-g)^+,(i_3-v-g)^+,i_4,\ldots,i_m)+w(2,i_2,i_3,u,v)\\ \vdots\\ W(i_1,\ldots,i_{m-2},(i_{m-1}-u-g)^+,(i_m-v-g)^+)+w(m-1,i_{m-1},i_m,u,v)\end{cases}$$

where x^+ denotes $\max(0,x)$, $w(l,i,j,u,v)=-\infty$ if $u>i$ or $v>j$, $0<u,v\leq w$ (the maximum window size), $g\geq 0$ (the gap), and $W(0,0,\ldots,0)=0$.

Fig. 6. Modified dynamic programming algorithm for Pegs and Rubber Bands with the windows and gaps formulation.

more theoretical results on approximation based on our formulation are obtained in [21] (but they potentially suffer from the "locking" problem mentioned in the Introduction because they rely on matching the RNAs into pairs in such a way to maximize the weight of the full pairwise interactions.)

As a practical step, and prior to choosing the best of the W_i's, we can fill in for each W_i some additional rubber bands (interactions) between (RNAs) row i and row $i+1$, between row $i+k$ and row $i+k+1$, and so on, by identifying the pegs of these rows (regions of RNAs) that are not part of the solution. This does not affect the theoretical guarantee but gives a larger weight to the solution. We call it *gap filling*.

III. Windows and Gaps: A Better Formulation for RNA Interaction

In the previous section, we described our initial attempt to view the interaction of m RNAs as a Pegs and Rubber Bands problem with m rows, where the first RNA interacts with the second RNA, and the second with the third, and so on (so they alternate in sense and antisense). This used a simple base-pair energy model, which is not too realistic. We now address this issue (and leave the issues of the ordering and the interaction pattern to Section IV). A better model for RNA interaction will consider windows of interaction instead of single bases. In terms of our Pegs and Rubber Bands problem, this translates to placing rubber bands around a stretch of contiguous pegs in two consecutive rows, e.g., around pegs $[l,i_1]$, $[l,i_2]$, $[l+1,j_1]$, and $[l+1,j_2]$, where $i_2\geq i_1$ and $j_2\geq j_1$. The weight contribution of placing such a rubber band is now given by $w(l,i_2,j_2,u,v)$, where i_2 and j_2 are the last two pegs covered by the rubber band in row l and row $l+1$, and $u=i_2-i_1+1$ and $v=j_2-j_1+1$ represent the length of the two windows covered in row l and row $l+1$, respectively.

A window with weight $w(l,i,j,u,v)$ represents a potential interaction between a stretch of length u ending at the ith base of RNA l and a stretch of length v ending at the jth base of RNA $l+1$. As before, the weight can be obtained as the negative of the energy of the corresponding interaction using a generalized energy model.

As a heuristic, we also allow for the possibility of imposing a gap $g\geq 0$ between windows to establish a distance at which windows may be considered energetically separate. This gap is also taken into consideration when we perform the gap filling procedure described at the end of Section II-B. The modified algorithm is shown in Fig. 6, and if we set $u=v=1$ and $g=0$, then we retrieve the original algorithm of Fig. 2.

The running time of the modified algorithm is $O(mw^2n^m)$ and $O(mw^2\lceil 1/\epsilon\rceil n^{\lceil 1/\epsilon\rceil})$ for the approximation scheme, where w is the maximum window length. If we impose that $u=v$, then those running times become $O(mwn^m)$ and $O(mw\lceil 1/\epsilon\rceil n^{\lceil 1/\epsilon\rceil})$ respectively.

For the correctness of the algorithm, we now have to assume that windows are *subadditive* (energy wise). In other words, we require the following condition (otherwise, the algorithm may compute an incorrect optimum due to the possibility of achieving the same window by two or more smaller ones with larger total weight):

$$w(l,i,j,u_1,v_1)+w(l,i-u_1,j-v_1,u_2,v_2)\leq w(l,i,j,u_1+u_2,v_1+v_2)$$

In our experience, most existing RNA-RNA interaction algorithms produce weights (the negative of the energy values) of RNA interaction windows that mostly conform to the above condition. At any rate, we filter the windows to eliminate those that are not subadditive. For instance, if the above condition is not met, we set $w(l,i,j,u_1,v_1)=w(l,i-u_1,j-v_1,u_2,v_2)=-\infty$ (recursively starting with smaller windows).

We use windows satisfying $2\leq u,v\leq w=26$. The weights $w(l,i,j,u,v)$ are obtained from RNAup, a tool to compute energies of pairwise interactions [7], as (negative of energy values):

$$w(l,i,j,u,v)\propto \log p_l(i-u+1,i)+\log p_{l+1}(j-v+1,j)+\log Z_l^I(i-u+1,i,j-v+1,j)$$

where $p_l(i_1,i_2)$ is the probability that subsequence $[i_1,i_2]$ is free (does not fold) in RNA l, and $Z_l^I(i_1,i_2,j_1,j_2)$ is the partition function (as computed in [7]) of the interaction of subsequences $[i_1,i_2]$ in RNA l and $[j_1,j_2]$ in RNA $l+1$ (subject to no folding within RNAs). As such, the weight considers intra-molecular and inter-molecular energies.

IV. Order and Interaction Pattern via Permutations: A Heuristic Algorithm

Viewing the interaction of m RNAs as Pegs and Rubber Bands with m rows dictates that the first RNA interacts with the second RNA, and the second with the third, and so on. This not only imposes a specific order on the interaction, but it also restricts each RNA to interact with at most two others. Therefore, this rather arbitrary choice in the model is eliminated. We first identify each RNA as being *even* (sense) or *odd* (antisense). Given m RNAs and a permutation (order) on the

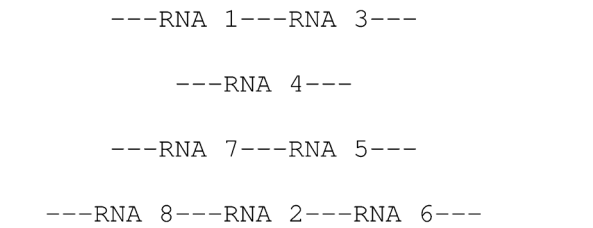

Fig. 7. Placement of the permutation {1,3,4,7,5,8,2,6} where the RNA number also indicates its parity. The interaction pattern is less restrictive then before; for instance, RNA 7 can interact with RNA 2, RNA 4, RNA 6, and RNA 8.

Given $\epsilon = 1/k$ and m RNAs
  produce a random permutation π on $\{1, \ldots, m\}$
  let W be the weight of the PTAS $(1-\epsilon)$-optimal solution given π
  repeat
    better←false
    generate a set Π of neighboring permutations for π
    for every $\pi' \in \Pi$ (in any order)
      do let W' be the weight of the PTAS $(1-\epsilon)$-optimal solution given π'
        if $W' > W$
          then $W \leftarrow W'$
            $\pi \leftarrow \pi'$
            better←true
  until not better

Fig. 8. A heuristic for multiple RNA interaction using the PTAS algorithm of Section II-C.

set $\{1, \ldots, m\}$, we map the RNAs onto the rows as follows: Starting with the first RNA, and moving in order, we place RNAs on the first row as long as they have the same parity. We then move to the next row, and perform this process for the remaining set. This is repeated until all RNAs have been placed. RNAs that end up on the same row are *virtually* considered as one RNA that is the concatenation of all. However, in the corresponding Pegs and Rubber Bands problem, we do not allow a window to span multiple RNAs on the same row, nor do we enforce a gap between two windows in different RNAs.

For example, if we consider the following permutation of RNAs {1,3,4,7,5,8,2,6}, where the RNA number also indicates its parity (for the sake of illustration), then we end up with the following placement: RNA 1 and RNA 3 in that order on the first row, followed by RNA 4 on the second row, followed by RNA 7 and RNA 5 in that order on the third row, followed by RNA 8, RNA 2, and RNA 6 in that order on the fourth row, resulting in four virtual RNAs on four rows of pegs as shown in Fig. 7.

Given a solution, random perturbations of the permutation are then used to find better solutions. Fig. 8 shows a possible heuristic that searches for the best permutation via neighboring permutations (recall that the permutation uniquely determines the placement).

To generate neighboring permutations for this heuristic algorithm one could adapt a standard 2-opt method used in the Traveling Salesman Problem (or other techniques). For instance, given permutation π, a neighboring permutation π' can be obtained by dividing π into three parts and making π' the concatenation of the first part, the reverse of the second part, and the third part. In other words, if $\pi = (\alpha, \beta, \gamma)$, then $\pi' = (\alpha, \beta^R, \gamma)$ is a neighbor of π, where β^R is the reverse of β.

V. Multiple Suboptimal Solutions

RNAs often interact in more than one way. We describe how to generate (all) solutions with a weight of at least some threshold T. The solutions are then clustered to limit their number. The clustering requires a distance metric which is also described below.

Process(S)
  $(i_1, \ldots, i_m) \leftarrow B(S)$
  if $W(i_1 - g - 1, \ldots, i_m - g - 1) + w(S) < T$
    then return
    else for every window $w(l, i, j, u, v)$ that is terminal is S
        with $i_l - i > g$ and $i_{l+1} - j > g$
        do Process($S + w(l, i, j, u, v)$)
  if $w(S) \geq T$
    then output S

Fig. 9. Generating multiple suboptimal solutions.

A. Generation

To explore the generation process, we assume that the order and interaction pattern have already been determined, e.g., the permutation given by the algorithm of Fig. 8. We then seek suboptimal solutions. Define the boundary $B(S)$ of a solution as $(i_1, \ldots, i_m)$, where i_l is the smallest index at row l such that peg $[l, i_l]$ is covered by a window, $l = 1 \ldots m$. We will also use $w(S)$ to denote the weight of that solution. We will use $w(l, i, j, u, v)$ interchangeably to denote a window and its weight. We denote by $S + w(l, i, j, u, v)$ an extension of solution S by the addition of window w.

We say that a window $w(l, i, j, u, v)$ in S with $B(S) = (i_1, \ldots, i_m)$ is a *terminal* window iff:

- $i - u + 1 = i_l$,
- $j - v + 1 = i_{l+1}$, and
- no other window $w(l', i', j', u', v')$ in S satisfies $i' - u' + 1 = i_{l'}$, $j' - v' + 1 = i_{l'+1}$, and $l' > l$.

This imposes some order on the windows to prevent generating the same solution in multiple ways. To that end, we can only extend a solution by adding to it a terminal window (a window that becomes the terminal for the extended solution). Observe that whenever $W(i_1 - g - 1, \ldots, i_m - g - 1) + w(S) < T$, where $B(S) = (i_1, \ldots, i_m)$, and g is the gap parameter as described in Section III, S cannot be extended in anyway to meet the threshold.

Let ϕ with $B(\phi) = (\infty, \ldots, \infty)$ represent the empty solution (with zero weight). We have the algorithm of Fig. 9 for generating every solution with weight at least T, starting with Process(ϕ). Because windows are considered in order, the running time of the algorithm is linear in the size of its output plus a crude $O(2^{|\mathbb{W}|})$ bound (all possible solutions), where $\mathbb{W}$ is the set of windows.

Further pruning is possible to speedup the process. For instance, if S with $B(S) = (i_1, \ldots, i_m)$ satisfies $w(S) + w(l, i, j, u, v) < T$, every window $w(l, i', j', u', v')$ such that $i' \leq i - u$ and $j' \leq j - v$ can be skipped from being added to S.

B. Distance Metric

We use a Jaccard metric [22] to quantify how dissimilar two solutions are. To motivate the approach below, consider two solutions generated by the process function of Fig. 9. If the two solutions are similar, we expect to have added a similar set of windows to each; furthermore, these windows are added in the same order.

Given a solution S, define $|S|$ as the number of windows in S, and let $w(l_1, i_1, j_1, u_1, v_1), \ldots, w(l_{|S|}, i_{|S|}, j_{|S|}, u_{|S|}, v_{|S|})$ be the $|S|$ windows in the order by which they were added to S using the process function of Fig. 9. Each of these windows, say $w(l, i, j, u, v)$, represents two intervals on two rows, namely $[i-u+1, i]$ on row l and $[j-v+1, j]$ on row $l+1$. Therefore, define the set of interaction intervals

$$I(S) = \big(I_1, \ldots, I_{2|S|}\big) = ([i_1-u_1+1, i_1], [j_1-v_1+1, j_1], \ldots \\ \ldots, \left[i_{|S|} - u_{|S|} + 1, i_{|S|}\right], \left[j_{|S|} - v_{|S|} + 1, j_{|S|}\right])$$

as an ordered sequence of $2|S|$ intervals, and $L(S) = (l_1, \ldots, l_{|S|})$ as an ordered sequence of $|S|$ rows, where l_i is the row defining the ith window. Therefore, $L(S)$ means that we have the following set of pairwise interactions (not necessarily unique in terms of RNAs): RNA l_1 with RNA l_1+1, RNA l_2 with RNA l_2+1, …, RNA $l_{|S|}$ with RNA $l_{|S|}+1$. Two solutions that do not agree on this set, are considered completely dissimilar; otherwise, their distance is given by the amount of overlap in their interaction intervals, hence the following definition of distance:

Given two solutions S_1 with $I(S_1) = (I_1, I_2, \ldots)$ and S_2 with $I(S_2) = (T_1, T_2, \ldots)$, the distance between S_1 and S_2 is

$$d(S_1, S_2) = \begin{cases} 1 & L(S_1) \neq L(S_2) \\ 1 - \frac{\sum_i |I_i \cap T_i|}{\sum_i |I_i \cup T_i|} & L(S_1) = L(S_2) \end{cases}$$

where $\cap$ and $\cup$ represent the standard intersection and union operations on sets respectively, and intervals are treated as sets of integers. This distance is a metric in [0,1].

Lemma 1: The distance defined above is a metric.

Proof: The distance is non-negative and symmetric. Furthermore, $d(S_1, S_2) = 0$ iff $S_1 = S_2$ because $|I_i \cap T_i| \leq |I_i \cup T_i|$ with equality iff $I_i = T_i$. Therefore, we only worry about proving subadditivity (triangular inequality). Consider the two solutions S_1 and S_2, and a third solution S_3. If $L(S_1) \neq L(S_2)$, then $L(S_1) \neq L(S_3)$ or $L(S_2) \neq L(S_3)$. Without loss of generality, let $L(S_1) \neq L(S_3)$. This means $d(S_1, S_2) = 1$ and $d(S_1, S_3) = 1$ and thus $d(S_1, S_2) \leq d(S_1, S_3) + d(S_3, S_2)$. If $L(S_1) = L(S_2)$, then either $L(S_1) = L(S_2) \neq L(S_3)$ or $L(S_1) = L(S_2) = L(S_3)$. In the former case $d(S_1, S_3) = d(S_3, S_2) = 1$, and thus $d(S_1, S_2) \leq d(S_1, S_3) + d(S_3, S_2)$. In the latter case, consider the sets

$$A = \bigcup_i I_i \times \{i\}$$
$$B = \bigcup_i T_i \times \{i\}$$

where the intervals are treated as sets of integers. Observe that $d(S_1, S_2) = 1 - ((A \cap B)/(A \cup B))$, which is a Jaccard distance and is known to be a metric. ∎

C. Clustering

The generated suboptimal solutions may be a lot more than what we need. In addition, many of them will be similar. Therefore, we use clustering to reduce their number. We adopt hierarchical agglomerative clustering with single linkage and the silhouette index [23] to determine the optimal number of clusters. Given a solution S, let $c(S)$ be its cluster. Let $b_j(S)$ be the average distance from S to all solutions in cluster j, and let $b(S) = \min_{j \neq c(S)} b_j(S)$. We assume that the number of clusters is at least 2, so $b(S)$ is defined. Let $a(S)$ be the average distance from S to all other solutions in $c(S)$. If S is a singleton in its cluster, we make $a(S) = b(S)$. The silhouette of a solution S is given by

$$\frac{b(S) - a(S)}{\max\left[a(S), b(S)\right]}$$

and is always in the interval $[-1, 1]$. A silhouette close to 1 means that solution S is well situated in its cluster since $a(S) \ll b(S)$. The silhouette of a cluster is the average silhouette of all the solutions in the cluster. The silhouette index is the average of all the cluster silhouettes. We seek the number of clusters that maximizes this index. The beauty of this index follows from that it is always bounded, works for arbitrary notions of distance (dissimilarity), and does not require the use of a cluster centroid, which is typically not trivial to find for non-Euclidean distances.

Given a number of clusters, the optimal solution in each cluster acts as a "representative" of the cluster. We sort the representatives (solutions) by their weight in a decreasing order (best to worst). The premise of this approach, in the case where alternative structures may exist, is that the first few representatives should reveal some of the realistic structures that are observed in biological experiments [24].

```
I1  3′ UGUAUG 5′
         ||||
U6    5′ AUACA.....GAUUa...       ...cGUGAAGCGU 3′
                   ||||              |||||||||
U2   3′ UAUGAUg....CUAGAAu..........gCACUUCGCA 5′
         |||||
I2    5′ UACUAAc 3′
```

Fig. 10. A modified human snRNA U2–U6 complex in the splicing of an intron, as reported in [25]. Bases indicated by small letters are missing from the interaction. From left to right: g-c and a-u are missing due to the condition $2 \leq u = v \leq 26$ for every window $w(l, i, j, u, v)$, but also due to the added instability of a bulge loop when this condition is relaxed; c-g ends up being not favored by RNAup. I1 is shifted (UGU should interact with ACA instead) but this is a computational artifact of optimization that is hard to avoid. Overall, the structure is accurate and cannot be predicted by a pairwise handling of the RNAs.

VI. Experimental Results

For all of our experiments, we will only show the interaction pattern among the RNAs. The folding within the individual RNAs is omitted. The windows and weights are obtained as described in Section III using the RNAup tool [7]. As a practical step, we filter all windows $w(l, i, j, u, v)$ to keep only those that satisfy $u = v$. We call these *balanced* windows.

A. Single Solutions

The heuristic algorithm of Fig. 8 powered by the PTAS algorithm is used to pick the largest weight solution among several runs. We set a window gap $g = 4$ as described in Section III. The value of $k = 1/\epsilon$ and the gap filling criterion (described at the end of Section II) depend on the scenarios, as described below:

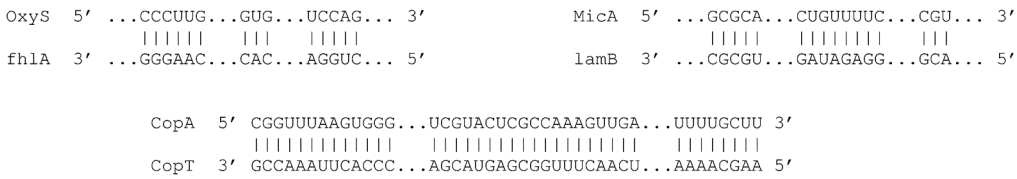

Fig. 11. Known pairs of interacting RNAs.

1) Structure Prediction in the Spliceosome: The human snRNA complex U2–U6 (two snRNAs) of the spliceosome is necessary for the splicing of an mRNA intron [13], [25]. This intron is thousands of nucleotides long; therefore, we only consider its functional regions, which consist of two structurally autonomous parts. This gives a total of four RNAs: two snRNAs U2 and U6; and two introns I1 and I2. We performed the algorithm with gap filling for $k = 2, 3, 4$. In all three cases, the solution with the largest weight consistently finds the structure shown in Fig. 10. This structure is in agreement with the pattern described in [13], [25]. Observe that our algorithm avoids the "locking" problem described in the Introduction, which would have favored the binding of U2–U6 at their left extremities in Fig. 10 when they fully interact, leaving I1 and I2 completely detached.

2) Fishing for Pairs: Six RNAs in *E. coli* of which three pairs are known to interact are used [8]. The interest here is to see whether the algorithm can identify the three pairs. For this purpose, it will suffice to set $k = 2$ and to ignore gap filling. Furthermore, we only consider solutions in which each RNA interacts with at most one other RNA. The solution with the largest weight identifies the three pairs correctly (Fig. 11). In addition, the interacting sites in each pair are consistent with the predictions of existing RNA-RNA interaction algorithms, e.g., [10].

3) Structural Separation: Six RNAs are used: CopA, CopT, and the four RNAs of the human spliceosome described above. The algorithm is performed with $k = 3$ and gap filling. The solution with the largest weight results in a successful prediction that separates the RNA structure shown in Fig. 10 from the RNA complex CopA-CopT of Fig. 11.

B. Multiple Suboptimal Solutions

The algorithm of Fig. 9 is modified to prevent the possibility of generating "technically" different solutions that may be "practically" the same. This can happen when the gap $g = 0$ so that a window can be generated as a combination of two or more adjacent windows. Therefore, we extend the condition for adding window $w(l, i, j, u, u)$ to a solution as follows (compare to Fig. 9):

$$(i_l - i > g \text{ and } i_{l+1} - j > g) \text{ and } (i_l - i > 1 \text{ or } i_{l+1} - j > 1)$$

We use $g = 0$ and set the threshold T low enough to generate at least 1000 solutions (when they exist). The 1000 (or fewer) solutions with the highest weights are then considered for clustering as described in Section V-C to produce representative solutions.

1) Structural Variation: The U2–U6 complex in the spliceosome of yeast has been reported to have two distinct experimental structures, e.g., [26]. In one conformation, U2 and U6 interact to form a helix known as helix Ia. In another conformation, the interaction reveals a structure containing an additional helix, known as helix Ib. It has been conjectured in [27] that coaxial stacking is essential for the stabilization of helix Ia in U2–U6 and, therefore, inhibition of the coaxial stacking, possibly by protein binding, may activate the second conformation. Regardless of what underlying mechanisms are responsible for this conformational switch, our suboptimal solutions cluster in a way that reveal the two conformations among the first few representatives (Fig. 12 and Fig. 13). In this example, the sequences of U2 and U6 have been truncated up to helix Ib, thus eliminating the Intramolecular Stem Loop ISL of U6 as in [28]. Without this truncation, helix Ib does not give rise to a balanced window with positive weight (and hence is dropped from the input set). Nevertheless, it can still appear as part of an unbalanced window that includes both helix Ia and helix Ib. When unbalanced windows are allowed in the input set, the corresponding solution containing both helices joins the cluster containing the solution with helix Ia alone. Therefore, the two will not be revealed as separate solutions as only one of them can be the representative of the cluster.

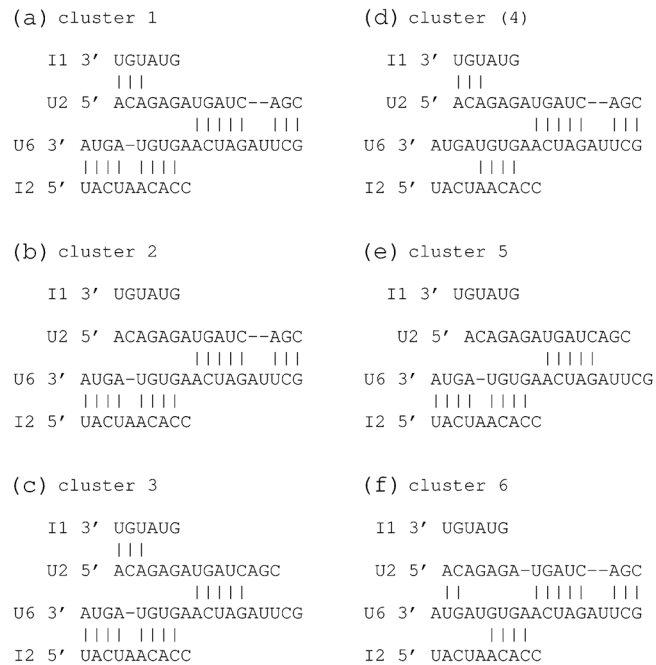

Fig. 12. (a) Helix Ia and helix Ib with both introns attached. (b) Helix Ia and helix Ib with I1 detached. (c) Helix Ia with both introns attached. (d) Helix Ia and helix Ib with I2 partially detached. (e) Helix Ia with I1 detached. (f) Helix Ia and helix Ib with I1 detached and I2 partially detached, moving towards detaching both introns, as would happen when U2 and U6 are bound optimally in a full pariwise interaction.

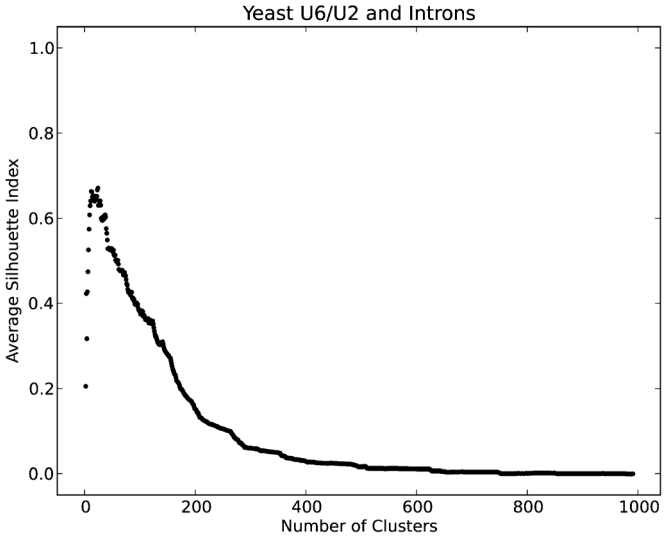

Fig. 13. Silhouette index for clustering 1000 yeast solutions, revealing a peack at 24 clusters.

```
(a) cluster 1

CopA 5′ CGGUUUAAGUGGG...UCGUACUCGCCAAAGUUGA...UUUUGCUU 3′
        |||||||||||||   ||||||||||||||||||   ||||||||
CopT 3′ GCCAAAUUCACCC...AGCAUGAGCGGUUUCAACU...AAAACGAA 5′


(b) cluster 2

CopA 5′ CGGUUUAAGUGGG...UCGUACUCGCCAAAGUUGA...UUUUGCUU 3′
        |||||||||||||   ||||||||||||||||||
CopT 3′ GCCAAAUUCACCC...AGCAUGAGCGGUUUCAACU...AAAACGAA 5′
```

Fig. 14. (a) The optimal solution. (b) A solution closer to the one observed in biological experiments in which the third interaction window in non-existent.

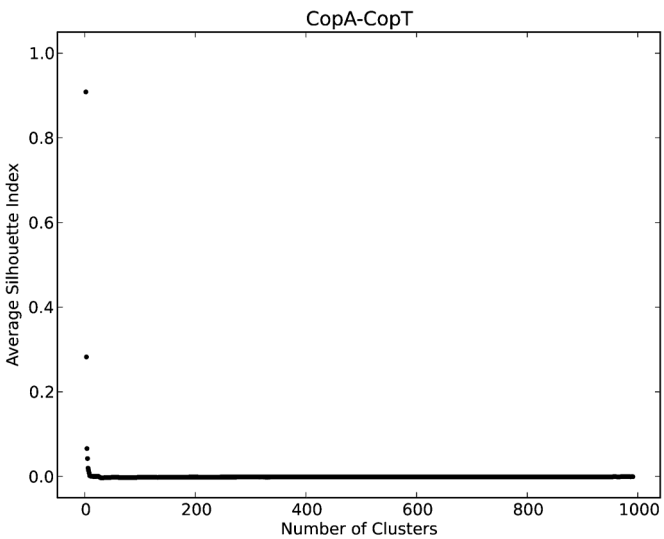

Fig. 15. Silhouette index for clustering 1000 CopA-CopT solutions, revealing a peak at two clusters.

are not biologically real. This is because dropping these interactions from the solution would make it less optimal. The third interaction window of CopA-CopT in Fig. 11 is an example of such an artifact. As shown in Figs. 14 and 15, clustering produces two clusters, thus two representatives. The second representative succeeds in dropping the undesired window.

3) Reversible Kissing Loops: Reversible kissing loops represent an even harder mechanism to capture by optimization. With this mechanism, the initial kissing complex occurs between a subset of loop bases in both RNAs, but this interaction is fully reversible and very unstable, [29]. Therefore, in the final interaction, the kissing loop will be missing few bases towards its center. A known example of this scenario is the middle interaction window of CopA-CopT in Figs. 11 and 14 (considering the folding pattern of CopA and CopT reveals that this interaction window is a kissing loop). By isolating this window and generating suboptimal solutions, we obtain two clusters with two representatives. The second representative reveals a separation of the interaction close to the center, as shown in Fig. 16 and Fig. 17.

```
(a) cluster 1

CopA 5′ ...UCGUACUCGCCAAAGUUG... 3′
           |||||||||||||||||
CopT 3′ ...AGCAUGAGCGGUUUCAAC... 5′


(b) cluster 2

CopA 5′ ...UCGUACUCGCCAAAGUUG... 3′
           ||||||  |||||||
CopT 3′ ...AGCAUGAGCGGUUUCAAC... 5′
```

Fig. 16. (a) The optimal solution for the middle window of CopA-CopT. A solution for the middle window that mimics the behavior of reversible kissing loops.

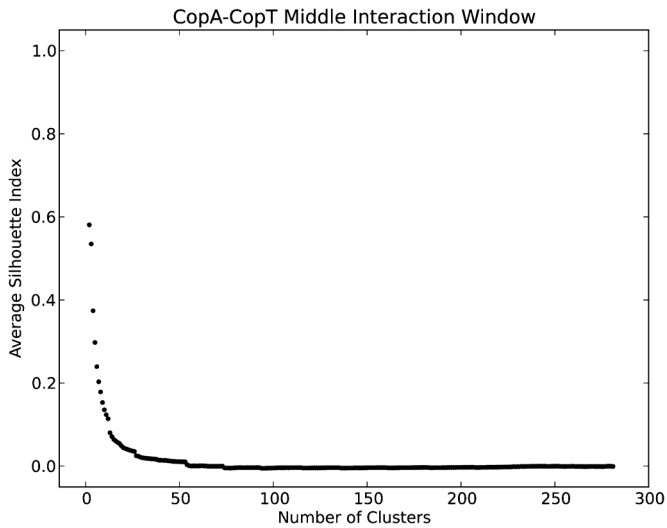

Fig. 17. Silhouette index for clustering 295 solutions of the middle window of CopA-CopT, revealing a peak at two clusters.

VII. Conclusion

While RNA-RNA interaction algorithms exist, they are not suitable for predicting RNA structures in which more than two RNA molecules interact. For one thing, the interaction pattern may not be known. Moreover, even with some existing knowledge on the pattern of interaction, treating the RNAs pairwise may not lead to the best global structure. We formulate multiple RNA interaction as an optimization problem, characterize its complexity (NP-hard), and provide approximation and heuristic algorithms.

We explore three scenarios: 1) structure prediction: we predict a correct complex of two snRNAs (modified human U2 and U6) and two structurally autonomous parts of an intron, a total of four RNAs; 2) fishing for pairs: given a pool of RNAs, we

identify the pairs that are known to interact; and 3) structural separation: we successfully divide the RNAs into independent groups of multiple interacting RNAs.

In practice, however, the best structure may not be the real one that is observed in biological experiments, and that in turn may not be unique. We extend the formulation to produce suboptimal solutions. Clustering those solutions can 1) provide several representative structures when they exist, e.g., two conformations of the U2-U6 complex in the spliceosome of yeast, and 2) find realistic structures that are not necessarily optimal in the computational sense, e.g., reversible kissing loops of CopA-CopT in *E. coli*.

Appendix A
RNA Sequences

MicA (even)

5′ GAAAGACGCGCAUUUGUUAUCAUCAUCCCUGUUUUCAGC
GAUGAAAUUUUGGCCACUCCGUGAGUGGCCUUUU 3′

lamB (odd)

5′ GGCAGCGCAUGUCGUCGUCUGCAUCAAGAGCCGGGUGUU
UAAGGCCUCCAUAAAAAAACGAAACGCAAAACCAUUCGC
AGUUUUAGAAGGUGGCAGCGUUUAAAGAAAAGCAAUGAU
CUCAGGAGAUAGAAUGAUGAUUACUCUGCGCAAACUCCC
ACUGGCGGUUGCUGUCGCAGCGG 3′

CopA (even)

5′ CGGUUUAAGUGGGCCCCGGUAAUCUUUUCGUACUCGCCA
AAGUUGAAGAAGAUUAUCGGGGUUUUUGCUU 3′

CopT (odd)

5′ AAGCAAAAACCCCGAUAAUCUUUUCAACUUUGGCGAGUA
CGAAAAGAUUACCGGGGCCCACUUAAACCG 3′

OxyS (even)

5′ GAAACGGAGCGGCACCUCUUUUAACCCUUGAAGUCACUG
CCCGUUUCGAGAGUUUCUCAACUCGAAUAACUAAAGCCA
ACGUGAACUUUUGCGGAUCUCCAGGAUCCGCU 3′

fhlA (odd)

5′ AGUUAGUCAAUGACCUUUUGCACCGCUUUGCGGUGCUUU
CCUGGAAGAACAAAAUGUCAUAUACACCGAUGAGUGAUC
UCGGACAACAAGGGUUGUUCGACAUCACUCGGACA 3′

Human Spliceosome

I1 (odd)

5′ NNNNNNNNNNGUAUGUNNNNNNNNNN 3′

U6 (even)

5′ AUACAGAGAAGAUUAGCAUGGCCCCUGCGCAAGGAUGAC
ACGCAAAUUCGUGAAGCGU 3′

U2 (odd)

5′ ACGCUUCACGGCCUUUUGGCUAAGAUCAAGUGUAGUAU 3′

I2 (even)

5′ NNNNNNNNNNUACUAACNNNNNNNNNN 3′

Yeast Spliceosome

I1 (odd)

5′ NNNNGUAUGUNNNN 3′

U6 (even)

5′ ACAGAGAUGAUCAGCAGUUCCCCUGCAUAAGGAUGAACC
GUUUU 3′

U2 (odd)

5′ CUUUGCCUUUUGGCUUAGAUCAAGUGUAGUA 3′

I2 (even)

5′ NNNNUACUAAUANNNN 3′

References

[1] D. D. Pervouchine, "Iris: Intermolecular rna interaction search," *Genome Informatics Series*, vol. 15, no. 2, p. 92, 2004.

[2] C. Alkan, E. Karakoc, J. H. Nadeau, S. C. Sahinalp, and K. Zhang, "Rna-rna interaction prediction and antisense rna target search," *J. Comput. Biol.*, vol. 13, no. 2, pp. 267–282, 2006.

[3] S. Mneimneh, "On the approximation of optimal structures for rna-rna interaction," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 6, no. 4, pp. 682–688, 2009.

[4] I. M. Meyer, "Predicting novel RNA-RNA interactions," *Current Opinion Struct. Biol.*, vol. 18, no. 3, pp. 387–393, 2008.

[5] F. A. Kolb, C. Malmgren, E. Westhof, C. Ehresmann, B. Ehresmann, E. Wagner, and P. Romby, "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*, vol. 6, no. 3, pp. 311–324, 2000.

[6] L. Argaman and S. Altuvia, "fhla repression by oxys rna: Kissing complex formation at two sites results in a stable antisense-target rna complex," *J. Mol. Biol.*, vol. 300, no. 5, pp. 1101–1112, 2000.

[7] U. Mückstein, H. Tafer, J. Hackermüller, S. H. Bernhart, P. F. Stadler, and I. L. Hofacker, "Thermodynamics of rna-rna binding," *Bioinformatics*, vol. 22, no. 10, pp. 1177–1182, 2006.

[8] H. Chitsaz, R. Backofen, and S. C. Sahinalp, "biRNA: Fast RNA-RNA binding sites prediction," in *Algorithms in Bioinformatics*. New York: Springer, 2009, pp. 25–36.

[9] H. Chitsaz, R. Salari, S. C. Sahinalp, and R. Backofen, "A partition function algorithm for interacting nucleic acid strands," *Bioinformatics*, vol. 25, no. 12, pp. i365–i373, 2009.

[10] R. Salari, R. Backofen, and S. C. Sahinalp, "Fast prediction of rna-rna interaction," *Algorithms Mol. Biol.*, vol. 5, no. 5, 2010.

[11] F. W. Huang, J. Qin, C. M. Reidys, and P. F. Stadler, "Partition function and base pairing probabilities for rna-rna interaction prediction," *Bioinformatics*, vol. 25, no. 20, pp. 2646–2654, 2009.

[12] A. X. Li, M. Marz, J. Qin, and C. M. Reidys, "RNA-RNA interaction prediction based on multiple sequence alignments," *Bioinformatics*, vol. 27, no. 4, pp. 456–463, 2011.

[13] J.-S. Sun and J. L. Manley, "A novel u2–u6 snrna structure is necessary for mammalian mrna splicing," *Genes Develop.*, vol. 9, no. 7, pp. 843–854, 1995.

[14] M. Andronescu, Z. C. Zhang, and A. Condon, "Secondary structure prediction of interacting rna molecules," *J. Mol. Biol.*, vol. 345, no. 5, pp. 987–1001, 2005.

[15] R. M. Dirks, J. S. Bois, J. M. Schaeffer, E. Winfree, and N. A. Pierce, "Thermodynamic analysis of interacting nucleic acid strands," *SIAM Review*, vol. 49, no. 1, pp. 65–88, 2007.

[16] J. S. McCaskill, "The equilibrium partition function and base pair binding probabilities for RNA secondary structure," *Biopolymers*, vol. 29, no. 6–7, pp. 1105–1119, 1990.

[17] H.-L. Chen, A. Condon, and H. Jabbari, "An $o(n^5)$ algorithm for mfe prediction of kissing hairpins and 4-chains in nucleic acids," *J. Comput. Biol.*, vol. 16, no. 6, pp. 803–815, 2009.

[18] S. Mneimneh, S. A. Ahmed, and N. L. Greenbaum, "Multiple RNA interaction—Formulations, approximations, heuristics," in *Proc. Int. Conf. Bioinformat. Models, Methods, Algorithms (Bioinformatics 2013)*, Barcelona, Spain, Feb. 11–14, 2013, pp. 242–249.

[19] S. A. Ahmed, S. Mneimneh, and N. L. Greenbaum, "A combinatorial approach for multiple rna interaction: Formulations, approximations, heuristics," in *Computing and Combinatorics*. Berlin/Heidelberg, Germany: Springer , 2013, pp. 421–433.

[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein *et al., Introduction to Algorithms*. Cambridge, MA, USA: MIT Press , 2001, vol. 2.

[21] W. Tong, R. Goebel, T. Liu, and G. Lin, "Approximation algorithms for the maximum multiple rna interaction problem," in *Combinatorial Optimization and Applications*. New York: Springer, 2013, pp. 49–59.

[22] P. Jaccard, *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Lausanne, Switzerland: Impr. Corbaz, 1901.

[23] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.

[24] S. A. Ahmed and S. Mneimneh, "Multiple rna interaction with sub-optimal solutions," in *Bioinformatics Research and Applications*. Berlin/Heidelberg, Germany: Springer International, 2014, pp. 149–162.

[25] C. Zhao, R. Bachu, M. Popović, M. Devany, M. Brenowitz, J. C. Schlatterer, and N. L. Greenbaum, "Conformational heterogeneity of the protein-free human spliceosomal u2–u6 snrna complex," *RNA*, vol. 19, no. 4, pp. 561–573, 2013.

[26] D. G. Sashital, G. Cornilescu, and S. E. Butcher, "U2–u6 rna folding reveals a group ii intron-like domain and a four-helix junction," *Nature Struct. Mol. Biol.*, vol. 11, no. 12, pp. 1237–1242, 2004.

[27] S. Cao and S.-J. Chen, "Free energy landscapes of rna/rna complexes: With applications to snRNA complexes in spliceosomes," *J. Mol. Biol.*, vol. 357, no. 1, pp. 292–312, 2006.

[28] M. I. Newby and N. L. Greenbaum, "A conserved pseudouridine modification in eukaryotic u2 snrna induces a change in branch-site architecture," *RNA*, vol. 7, no. 06, pp. 833–845, 2001.

[29] F. A. Kolb, H. M. Engdahl, J. G. Slagter-Jäger, B. Ehresmann, C. Ehresmann, E. Westhof, E. G. H. Wagner, and P. Romby, "Progression of a loop-loop complex to a four-way junction is crucial for the activity of a regulatory antisense RNA," *EMBO J.*, vol. 19, no. 21, pp. 5905–5915, 2000.