Time and space efficient RNA-RNA interaction prediction via sparse folding

Raheleh Salari[1]*, Mathias Möhl[2]*, Sebastian Will[2]*,
S. Cenk Sahinalp[1]**, Rolf Backofen [2]**

[1] Lab for Computational Biology, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
{*rsalaris,cenk*}*@cs.sfu.ca*

[2] Bioinformatics, Institute of Computer Science, Albert-Ludwigs-Universität, Freiburg, Germany
{*mmohl,will,backofen*}*@informatik.uni-freiburg.de*

Abstract. In the past years, a large set of new regulatory ncRNAs have been identified, but the number of experimentally verified targets is considerably low. Thus, computational target prediction methods are on high demand. Whereas all previous approaches for predicting a general joint structure have a complexity of $O(n^6)$ running time and $O(n^4)$ space, a more time and space efficient interaction prediction that is able to handle complex joint structures is necessary for genome-wide target prediction problems. In this paper we show how to reduce both the time and space complexity of the RNA-RNA interaction prediction problem as described by Alkan et al. [1] via dynamic programming sparsification - which allows to discard large portions of DP tables without loosing optimality. Applying sparsification techniques reduces the complexity of the original algorithm from $O(n^6)$ time and $O(n^4)$ space to $O(n^4\psi(n))$ time and $O(n^2\psi(n) + n^3)$ space for some function $\psi(n)$, which turns out to have small values for the range of n that we encounter in practice. Under the assumption that the polymer-zeta property holds for RNA-structures, we demonstrate that $\psi(n) = O(n)$ on average, resulting in a linear time and space complexity improvement over the original algorithm. We evaluate our sparsified algorithm for RNA-RNA interaction prediction by total free energy minimization, based on the energy model of Chitsaz et al. [2], on a set of known interactions. Our results confirm the significant reduction of time and space requirements in practice.

1 Introduction

Starting with the discovery of microRNAs (miRNAs) and the advent of genome-wide transcriptomics, it has become clear that RNA plays a large variety of important roles in living organisms that extend far beyond being a mere intermediate in protein biosynthesis [3]. Several of these non-coding RNAs (ncRNAs)

* Joint first authors

** to whom correspondence should be addressed regulate gene expression post-transcriptionally through base pairing (and establishing a joint structure) with a target mRNA, as per the eukaryotic miRNAs and small interfering RNAs (siRNAs) [4–6], antisense RNAs [7, 8] or bacterial small regulatory RNAs (sRNAs) [9]. In addition to such endogenous regulatory ncRNAs, antisense oligonucleotides have been used as exogenous inhibitors of gene expression; antisense technology is now commonly used as a research tool as well as for therapeutic purposes. Furthermore, synthetic nucleic acids systems have been engineered to self assemble into complex structures performing various dynamic mechanical motions [10–14].

Despite all the above advances, the first set of computational methods for predicting ncRNA-target mRNA interactions suffered from over-simplifying the types of interactions allowed. As a result they could not accurately predict many known interactions, especially those involving long ncRNAs. More precisely, these methods either restricted the interactions to external positions, or they allowed interactions with at most one interaction site. These restrictions were lifted by two independently developed methods, which provided the first set of algorithms for predicting a precise interaction structure of two RNA strands: (i) the algorithm by Pervouchine [15], for example, maximizes the total number of base pairs, and (ii) a more general method by Alkan et al. [1], minimizes the total free energy of the interacting RNA strands using a nearest neighbor energy model. Alkan *et al.* also provide a proof of the NP-completeness of the general problem, together with a precise definition of interaction types that can be handled, as well as the first experimental confirmation of the total free energy minimization approach via correctly predicting the joint structure formed by a number of interacting RNA pairs.

More recently, two approaches [2, 16] independently solved the problem of calculating the partition function for the interaction model introduced by Alkan *et al.*, allowing to determine important thermodynamic quantities like melting temperatures. As demonstrated in [2], the computed melting temperatures are in a good agreement with experimentally measured ones.

One key problem with the above approaches for predicting a general joint structure [15, 1, 2, 16] is that they all have a worst case running time of $O(n^6)$ and a space complexity of $O(n^4)$. While this complexity might be acceptable when analyzing only a few putative sRNA-target interaction pairs, we are now faced with the situation that the amount of data to be analyzed is vastly increasing. To give an example, a recent mapping of transcripts using tiling arrays in the budding yeast *S. cerevisiae* [17] with 5,654 annotated open reading frames (ORF) has found 1555 antisense RNAs that overlap at least partially with the ORFs at the opposite strand. Currently, it is completely unclear what these antisense RNAs are doing - whether they target only their associated sense mRNA or have also other mRNA targets, and whether they always form a complete duplex or more complex joint structures such as multiple kissing hairpins if they overlap only partially is not known. The same situation appears in many other species. Thus, there is urgent need for a more time and space efficient interaction prediction method that is able to handle complex joint structures.

In this paper we present a new method for calculating the joint structure of interacting RNAs by minimizing their total free energy, which improves time and space efficiency over previous approaches. As first in its class, the method is sufficiently fast to be applied in large scale screening approaches. We suggest to refine putative interacting pairs with even more accurate RNA-RNA-interaction prediction approaches [2, 16]. Because these approaches compute a partition function for RNA-RNA-interaction, they can determine important thermodynamic parameters such as melting temperatures, however their efficiency cannot be improved in the same way.

We show how to reduce both time and space complexity using an approach called *sparsification*, which uses the observation that the resulting DP-matrices are sparse. As previous applications of sparsification to problems related to RNA folding, our approach exploits a triangle inequation on the dynamic programming matrix. Assuming the *polymer-zeta* property for interacting RNAs, we show an efficiency gain by a linear factor. This *polymer-zeta* property basically states that the probability of a base pair decreases with its size, i.e. there are only few long range base pairs.

In this paper we consider a version of the polymer-zeta property for interacting RNAs and develop novel algorithmic approaches as (1) we cannot assume the standard polymer-zeta property for all base pairs as for intermolecular base pairs there is no clear notion of a distance between the bases; (2) the joint interaction prediction problem does not allow to split only at arcs in the recursion, which was crucial in the demonstration of a linear (asymptotic) speed up for problems involving the folding of a single RNA.

We sparsify the dynamic programming tables involved in total free energy minimization first described in Alkan *et al.* [1] on the more general energy model of Chitsaz *et al.* [2] resulting in a significant reduction in time and space complexity. There are four different cases that need to be sped up, which results in a total of four different candidate lists; for each sequence and each region, we have to consider folding with interaction or without interaction, which gives rise to two candidate lists per sequence. We emphasize that beyond reducing time complexity, we obtain a similar space reduction even in the intricate setting of four independent candidate lists.

Sparsification in RNA folding. The general technique of DP sparsification has been used in the context of RNA-folding, to reduce the time and space complexity of two central problems in this domain, namely (i) the calculation of the MFE structure of a single RNA sequence folding [18, 19], and (ii) the Sankoff approach [20] of simultaneous folding and alignment of two RNAs [21, 19]. In both cases, a (roughly) linear reduction in the time complexity was achieved on average.[1] The time/space reduction is based on the assumption that RNA-structures or consensus structures - in the simultaneous alignment and folding of RNAs,

[1] To be more precise, the time complexity of RNA-folding was reduced from $O(n^3)$ to $O(nZ)$ and the space complexity was reduced from $O(n^2)$ to $O(Z)$, where Z is a sparsity factor satisfying $n \leq Z \leq n^2$. An estimation [18] of the expected value of a parameter related to Z, based on a probabilistic model for polymer folding and

satisfy the polymer-zeta behavior, which is an assumption that we employ in predicting the intramolecular base-pairs observed in RNA joint structures. The above approaches for RNA folding as well as simultaneous folding and alignment use the polymer-zeta property for either a single RNA sequence and structure, or for a consensus structure of two (structurally similar) RNAs, leading to a single candidate list.

RNA-RNA interaction prediction methods. The first set of computational methods to calculate joint structures formed by interacting RNAs (e.g., RNAhybrid [22] or TargetRNA [23]) considered only the base-pairs between the two different strands that form a duplex structure. Since this ignores the intramolecular structures, later approaches aimed to predict a joint structure for both interacting RNAs. This second generation of RNA-RNA interaction prediction methods, which include pairfold [24], RNAcofold [25] and the method presented by Dirks *et al.* as part of the NUpack package [26], consider joint structures of mRNA and sRNA that are generated by concatenating the two sequences using a special linker character. Then, a modified version of the standard RNA-folding algorithms (such as Mfold [27] or RNAfold [28]) which preserve the basic recursive structure of standard RNA-folding but specially treat loops that contain the linker symbol, is applied. Unfortunately, none of the above approaches can predict joint structures with kissing hairpin interactions. For that reason, a third generation of RNA-RNA interaction prediction algorithms (in particular, RNAup [29] and IntaRNA [30]) were recently introduced. These approaches first determine the accessibility of all putative interaction sites, from which an energy to make the sites free of intramolecular base-pairs can be calculated. Later, this energy is combined with the energy of the duplex that can be formed between different interaction sites.

Clearly, the third generation methods can only handle one interaction site per sequence - which may not include any intramolecular base-pairs. As a result, two or more kissing hairpins as per the interaction between OxyS and fhlA [31] cannot be treated by these approaches. For the purpose of handling such complex joint structures, more sophisticated DP-methods of Pervouchine [15] and Alkan et al. [1], as well as the partition function variants by Chitsaz et al. [2] and Huang et al. [16] were introduced. Finally, more recent methods introduced in [32, 33] can be seen as heuristic approximations to the full model of [2], or as an extension of the accessibility approaches (RNAup/IntaRNA) to several interaction sites.

2 Preliminaries

Throughout this paper, we denote the two nucleic acid strands by $\mathbf{R}$ and $\mathbf{S}$. Strand $\mathbf{R}$ is indexed from 1 to L_R in $5'$ to $3'$ direction and $\mathbf{S}$ is indexed from 1 to L_S in $3'$ to $5'$ direction. Note that the two strands interact in opposite directions, e.g. $\mathbf{R}$ in $5' \rightarrow 3'$ with $\mathbf{S}$ in $3' \leftarrow 5'$ direction. Each nucleotide is paired with at most one nucleotide in the same or the other strand. The subsequence from the

measured by simulations, shows that Z is significantly smaller than $O(n^2)$. Similar results are given for the co-folding problem.

i^{th} nucleotide to the j^{th} nucleotide in a strand is denoted by $[i, j]$. We refer to the i^{th} nucleotide in $\mathbf{R}$ and $\mathbf{S}$ by i_R and i_S respectively. An intramolecular base pair between the nucleotides i and j in a strand is called an *arc* and denoted by a bullet $i \bullet j$. An intermolecular base pair between the nucleotides i_R and i_S is called a *bond* and denoted by a circle $i_R \circ i_S$. An arc $i_R \bullet j_R$ (or respectively $i_S \bullet j_S$) *covers* a bond $k_R \circ k_S$ if $i_R < k_R < j_R$ (or $i_S < k_S < j_S$). An arc is called *interaction arc* if it covers a bond. A subsequence $[i_R, j_R]$ (or $[i_S, j_S]$, analogously) contains a *direct bond*, $k_R \circ k_S$, if $i_R \leq k_R \leq j_R$ and no arc within $[i_R, j_R]$ covers $k_R \circ k_S$. Two bonds $i_R \circ i_S$ and $j_R \circ j_S$ are called *crossing bonds* if $i_R < j_R$ and $i_S > j_S$ or $i_R > j_R$ and $i_S < j_S$. An interaction arc $i_R \bullet j_R$ in R *subsumes* a subsequence $[i_S, j_S]$ in S if there is at least one bond $k_R \circ k_S$, where $i_R < k_R < j_R$ and $i_S < k_S < j_S$, and for all bonds $k_R \circ k_S$, if $i_S \leq k_S \leq j_S$ then $i_R < k_R < j_R$. Analogously, interaction arcs in S can subsume subsequences in R. Two interaction arcs $i_R \bullet j_R$ and $i_S \bullet j_S$ are part of a *zigzag*, if there is a bond $k_R \circ k_S$, where $i_R < k_R < j_R$ and $i_S < k_S < j_S$, but neither $i_R \bullet j_R$ subsumes $[i_S, j_S]$ nor $i_S \bullet j_S$ subsumes $[i_R, j_R]$.

We represent the recursions of our dynamic programming (DP) algorithm in a graphical notation using the recursion diagrams introduced in [2]. Within the recursion diagrams, a horizontal line indicates the phosphate backbone, a solid curved line indicates an arc, and a dashed curved line encloses a region and denotes its two terminal bases which may be paired or unpaired. Letters within a region specify a recursive quantity. White regions are recursed over and blue regions indicate those portions of the secondary structure that are fixed at the current recursion level and contribute to the energy as defined by the energy model. Green and red regions have the same recursion cases as the corresponding white regions, except that for the green regions multiloop energy and for red regions kissing loop energy is applied, i.e. the corresponding penalties for each unpaired base and base pair should be applied. A solid vertical line indicates a bond, a dashed vertical line denotes two terminal bases of a region which may be base paired or unpaired, and a dotted vertical line denotes two terminal bases of a region which are assumed to be unpaired. A terminal determined by $\bullet$ is starting point of either an interaction arc or a bond.

3 Methods

In this section we discuss an algorithm for RNA-RNA interaction prediction via total free energy minimization, under the assumption that there are no (internal) pseudoknots, crossing bonds (i.e. external pseudoknots), or zigzags in the joint structure. The algorithm is similar to the one introduced by Alkan et al. [1] on a simpler energy model. We use sparsification techniques to reduce the complexity of the original algorithm from $O(n^6)$ time and $O(n^4)$ space to $O(n^4\psi(n))$ time and $O(n^2\psi(n)+n^3)$ space for some function $\psi(n) = O(n)$ on average. To simplify the presentation, we discuss the sparsification for the joint structure prediction via total base pair maximization. Note that RNA-RNA interaction based on base pair maximization is the generalized version of the Nussinov model [34] for single RNA folding and was employed by Pervouchine [15] as well as Alkan et al. [1] for RNA-RNA interaction prediction. Later in the paper we also provide all concepts for generalizing the algorithm to capture a more realistic energy model provided by Chitsaz et al. [2].

3.1 Sparsification for Maximizing Base Pairs

Given two RNA sequences $\mathbf{R}$ and $\mathbf{S}$, $N(i_R, j_R, i_S, j_S)$ denotes the maximum number of base pairs in the joint structure of $[i_R, j_R]$ and $[i_S, j_S]$, and $N^{\mathbf{X}}(i,j)$ (for $\mathbf{X} \in \{\mathbf{R}, \mathbf{S}\}$) denotes the maximum number of base pairs of the subsequence $[i,j]$ of the single sequence $\mathbf{X}$. The recursion cases for computing the maximum number of base pairs for RNA-RNA interaction are illustrated in Fig. 1. $N(i_R, j_R, i_S, j_S)$ and $N^{\mathbf{X}}(i,j)$ for $\mathbf{X} \in \{\mathbf{R}, \mathbf{S}\}$ are calculated by the following recursions

$$N(i_R, j_R, i_S, j_S) = \max \begin{cases} N(i_R+1, j_R, i_S, j_S) & (a) \\ N(i_R, j_R, i_S+1, j_S) & (b) \\ N(i_R+1, j_R, i_S+1, j_S)+1 & (c) \\ \max\limits_{\substack{i_R<k\leq j_R \\ R[i_R],R[k] \text{ compl.}}} \begin{pmatrix} 1+N^{\mathbf{R}}(i_R+1, k-1) \\ \quad +N(k+1, j_R, i_S, j_S) \end{pmatrix} & (d) \\ \max\limits_{\substack{i_S\leq k<j_S \\ S[i_S],S[k] \text{ compl.}}} \begin{pmatrix} 1+N^{\mathbf{S}}(i_S+1, k-1) \\ \quad +N(i_R, j_R, k+1, j_S) \end{pmatrix} & (e) \\ \max\limits_{\substack{i_R<k_R\leq j_R \\ i_S<k_S\leq j_S \\ R[i_R],R[k_R] \text{ compl.}}} \begin{pmatrix} 1+N(i_R+1, k_R-1, i_S, k_S) \\ \quad +N(k_R+1, j_R, k_S+1, j_S) \end{pmatrix} & (f) \\ \max\limits_{\substack{i_R<k_R\leq j_R \\ i_S<k_S\leq j_S \\ S[i_S],S[k_S] \text{ compl.}}} \begin{pmatrix} 1+N(i_R, k_R, i_S+1, k_S-1) \\ \quad +N(k_R+1, j_R, k_S+1, j_S) \end{pmatrix} & (g) \end{cases} \tag{1}$$

$$N^{\mathbf{X}}(i,j) = \max \begin{cases} N^{\mathbf{X}}(i+1, j) & (a) \\ \max\limits_{\substack{i<k\leq j \\ X[i],X[k] \text{ compl.}}} \begin{pmatrix} 1+N^{\mathbf{X}}(i+1, k-1) \\ \quad +N^{\mathbf{X}}(k+1, j) \end{pmatrix} & (b) \end{cases} \tag{2}$$

In Eq. 1, the cases (a) and (b) introduce an unpaired base at positions i_R and i_S respectively, and case (c) introduces a bond $i_R \circ i_S$. Cases (d) and (f) introduce an arc at $i_R \bullet k$ and cases (e) and (g) at $i_S \bullet k$, where cases (f) and (g) assume that the arc is an interaction arc and cases (d) and (e) assume that this is not the case.

Time reduction by sparsification We will apply a sparsification technique to reduce the number of cases necessary to be considered for Eq 1(d)-(g), as well as Eq 2(b).

Concerning sparsification, the simple cases are Eq 1(d),(e), and Eq 2(b), which correspond to the folding of a single sequence. The sparsification of these

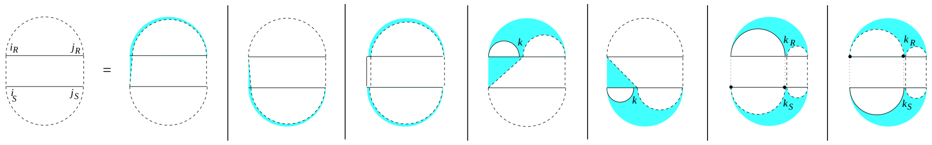


Fig. 1. Recursion cases for computing the maximum base pairing joint structure of $[i_R, j_R]$ and $[i_S, j_S]$.

cases works in close analogy to the sparsification of RNA structure prediction as described by Wexler et al. [18]. We will briefly review their approach adapted to case Eq 2(b). Thereafter, we describe sparsification of the complex cases.

Sparsifying recursion cases for single structure folding The key to sparsification is a triangle inequality property of the DP matrix. In the case of $N^{\mathbf{X}}$, for every subsequence $[i, j]$ and $i < k \leq j$ the following inequality holds:

$$N^{\mathbf{X}}(i,j) \geq N^{\mathbf{X}}(i,k) + N^{\mathbf{X}}(k+1,j).$$

Due to this property, it is sufficient to maximize in Eq. 2(b) for each i only over certain candidates k instead of all k with $i < k \leq j$. In this case, k is a candidate for i, iff $N^{\mathbf{X}}(i+1,k) < N^{\mathbf{X}}(i,k)$ and for all $i < k' < k$, $1 + N^{\mathbf{X}}(i+1, k'-1) + N^{\mathbf{X}}(k'+1,k) < N^{\mathbf{X}}(i,k)$. Operationally, during the computation of $N^{\mathbf{X}}(i,k)$ we detect that k is a candidate for i by checking that the instance $1+N^{\mathbf{X}}(i+1,k-1)+N^{\mathbf{X}}(k+1,k)$ of recursion case Eq. 2(b) is the only maximal case.

For non-candidates k there exists some k', $i \leq k' < k$, where $N^{\mathbf{X}}(i,k) = N^{\mathbf{X}}(i,k') + N^{\mathbf{X}}(k'+1,k)$. Then for all $j > k$, $N^{\mathbf{X}}(i,k) + N^{\mathbf{X}}(k+1,j) = N^{\mathbf{X}}(i,k') + N^{\mathbf{X}}(k'+1,k) + N^{\mathbf{X}}(k+1,j)$, and by triangle inequality $N^{\mathbf{X}}(i,k) + N^{\mathbf{X}}(k+1,j) \leq N^{\mathbf{X}}(i,k') + N^{\mathbf{X}}(k'+1,j)$. This means that, whenever a non-candidate k yields a maximal value, then there is already a $k' < k$ that yields the same value. Therefore k does not need to be considered, because the smallest such k' is taken into account.

Wexler et al. showed that sparsification reduces the expected time complexity of RNA folding by a linear factor, since the expected number of candidates for each i is constant. The transfer of sparsification to cases Eq 1(d) and (e) is straightforward, because only one subsequence is decomposed and the indices of the other subsequence remain fixed.

Sparsifying recursion cases for joint structure folding We extend the sparsification idea to the recursion cases Eq 1(f) and (g), which split both sequences and therefore minimize over a pair of split points (k_R, k_S). For the four dimensional matrix $N(i_R, j_R, i_S, j_S)$, the following generalization of the triangle inequality holds.

Observation 1 (Triangle inequality for $N(i_R, j_R, i_S, j_S)$) *For every subsequence $[i_R, j_R]$ and $[i_S, j_S]$ and for every $i_R < k_R \leq j_R$ and $i_S \leq k_S < j_S$, $N(i_R, j_R, i_S, j_S) \geq N(i_R, k_R, i_S, k_S) + N(k_R+1, j_R, k_S+1, j_S)$.*

Note that in principle both cases Eq 1(f) and (g) split the two subsequences at k_R and k_S, respectively, into the pairs $[i_R, k_R]$, $[i_S, k_S]$ and $[k_R+1, j_R]$, $[k_S+1, j_S]$. The only difference is that within the first pair of subsequences, $[i_R, k_R]$, $[i_S, k_S]$, case (f) assumes an arc $i_R \bullet k_R$ and case (g) assumes an arc $i_S \bullet k_S$. We consider only the case Eq 1(f), the case (g) is analogous.

Definition 1. (Candidate for case Eq. 1(f)) *For case Eq. 1(f), a pair (k_R, k_S) is a* candidate *for (i_R, i_S), iff i_R and k_R are complementary and for all $(k'_R, k'_S) \neq (k_R, k_S)$ with $i_R < k'_R \leq k_R$, $i_S < k'_S \leq k_S$,*

$$\begin{aligned} &1 + N(i_R+1, k_R-1, i_S, k_S) + N(k_R+1, k_R, k_S+1, k_S) \\ &> 1 + N(i_R+1, k'_R-1, k'_S, k_S) + N(k'_R+1, k_R, k'_S+1, k_S), \end{aligned}$$

With respect to the recursion case (f) a candidate (k_R, k_S) implies that the instance with $k_R = j_R$ and $k_S = j_S$ (i.e. $1 + N(i_R+1, k_R-1, i_S, k_S) + N(k_R+1, k_R, k_S+1, k_S)$) is the only maximal instance in the maximization of (f). Furthermore, it implies that none of the cases (a)-(e) in the computation of $N(i_R, k_R, i_S, k_S)$ yields a larger value than case (f).

Lemma 1. *For correctness of the recursion of Eq. 1, in the maximization of Eq. 1(f) it suffices to consider only the set of candidates given above.*

Proof. For any non-candidate (k_R, k_S), there exists some (k'_R, k'_S) with $i_R - 1 \leq k'_R \leq k_R$, $i_S - 1 \leq k'_S \leq k_S$, $(k'_R, k'_S) \neq (k_R, k_S)$, $(k'_R, k'_S) \neq (i_R-1, i_S-1)$, and

$$1 + N(i_R+1, k_R-1, i_S, k_S) \leq N(i_R, k'_R, i_S, k'_S) + N(k'_R+1, k_R, k'_S+1, k_S). \quad (3)$$

Note that $k'_R = i_R - 1$ or $k'_S = i_S - 1$ in Eq. 3 occurs when (k_R, k_S) is not a candidate due to one of the recursion cases (a)-(e).

Eq. 3 and the triangle inequality imply that for all $j_R > k_R$ and $j_S > k_S$

$$\begin{aligned} &1 + N(i_R+1, k_R-1, k_S, j_S) + N(k_R+1, j_R, k_S+1, j_S) \\ &\leq N(i_R, k'_R, i_S, k'_S) + N(k'_R+1, k_R, k'_S+1, k_S) + N(k_R+1, j_R, k_S+1, j_S) \\ &\leq N(i_R, k'_R, i_S, k'_S) + N(k'_R+1, j_R, k'_S+1, j_S). \end{aligned} \quad (4)$$

Non-candidates (k_R, k_S) for (i_R, i_S) do not need to be considered in the recursions of all $N(i_R, j_R, i_S, j_S)$, because there exists a recursion case splitting at (k'_R, k'_S) that yields the same or better score for $N(i_R, k_R, i_S, k_S)$. The equivalent case is considered in the recursion of $N(i_R, j_R, i_S, j_S)$ and, due to Eq. 4, yields a greater or equal score. □

Therefore the recursion case Eq. 1(f) can be updated such that the maximization runs only over the candidates for this case.

$$\max_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ (k_R, k_S) \text{ candidate for } (i_R, i_S)}} \begin{pmatrix} 1 + N(i_R+1, k_R-1, i_S, k_S) \\ + N(k_R+1, j_R, k_S+1, j_S) \end{pmatrix} \quad (5)$$

Analogously, we define candidates for case Eq. 1(g). The candidate criterion for Eq. 1(g) is stricter than for Eq. 1(f), since we require that a candidate for Eq. 1(g) is better than all cases Eq. 1(a)-(e) and (f).

Definition 2 (Expected number of candidates). *$\psi_1(n)$ denotes the expected number of candidates $k \leq n+i$ for some i in cases Eq. 1(d),(e), and Eq. 2(b). $\psi_2(n)$ is the expected number of candidates (k_R, k_S), $k_R \leq i_R+n$, $k_S \leq i_S+n$, for some (i_R, i_S) in cases Eq. 1(f) and (g).*

Applying the described sparsification to all non-constant cases in recursions Eq. 1 and Eq. 2, yields the following.

Theorem 2. *$N(1, L_R, 1, L_S)$ can be computed in $O((\psi_1(n)+\psi_2(n))n^4)$ expected time, where $n = max(L_R, L_S)$.*

For a theoretical bound on $\psi_1(n)$ and $\psi_2(n)$, we assume the polymer-zeta property holds for each one of the RNA sequences that are involved in the interaction (with the other RNA sequence). The polymer-zeta property states that in any long polymer chain the probability of having arc between two monomers with distance m converges to $b.m^{-c}$, where $b, c > 0$ are some constants. For a polymer as a self-avoiding random walk on a square lattice, it has been known that $c > 1$ [35]. The exponent c for the denaturation transition of DNA in both 2D and 3D models is found to be larger than 2 [36]. Since RNA folds similar to other polymers, one can assume that RNA folding obeys the polymer-zeta property; i.e. the probability that a structure is formed over the subsequence of length m converges to $b.m^{-c}$, where $c > 1$. Although the property is not proven for RNA molecules, there is empirical evidence, as shown by Wexler et al. [18], that a version of polymer-zeta property holds for RNA molecules as well.

Lemma 2. *Assume that the two interacting RNAs independently satisfy the* polymer-zeta *property with $c > 1$, i.e. there exist constants $b > 0$ and $c > 1$ such that the probability for any internal base pair $i \bullet (i+m)$ is bounded by $b \cdot m^{-c}$ - even when two RNAs interact. Then $\psi_1(n) = O(1)$ and $\psi_2(n) = O(n)$.*

Proof. $\psi_1(n) = O(1)$ follows from Wexler et al. [18]. For $\psi_2(n) = O(n)$, consider all candidates (k_R, k_S) for (i_R, i_S) and case Eq. 1(f). (Case Eq. 1(g) is symmetric.) Note that in Eq. 1(f), $i_R \bullet k_R$. For a fixed k_S analogously to Wexler et al. [18], the expected number of k_R with $i_R \bullet k_R$ is $b\sum_{i=1}^{n} i^{-c} < b\sum_{i=1}^{\infty} i^{-c}$ which converges to a constant for $c > 1$. Hence for each of the $O(n)$ possible values of k_S, k_R takes only a constant number of different values and hence on average we have $O(n)$ such candidates. □

Space efficient strategy The space complexity of the algorithm can be reduced from $O(n^4)$ to $O(n^3+\psi(n)n^2)$ as follows. The matrices $N^{\mathbf{R}}$ and $N^{\mathbf{S}}$ only require $O(n^2)$ space. All cases for the computation of an entry $N(i_R, j_R, i_S, j_S)$ only rely on entries $N(i'_R, j'_R, i'_S, j'_S)$ that satisfy one of the following two properties. (i) $j'_R \in \{j_R-1, j_R\}$ and $j'_S \in \{j_S-1, j_S\}$ or (ii) $N(i'_R, j'_R, i'_S, j'_S)$ corresponds to

```
Algorithm: Space efficient evaluation of Eq. 1
precompute matrices $N^{\mathbf{R}}$ and $N^{\mathbf{S}}$ ;
initialize empty lists for candidates ;
for $j_R = 1..L_R$ do
    allocate and init matrix slice $N(\cdot, j_R, \cdot, \cdot)$ ;
    for $j_S = 1..L_S$, $i_R = j_R..1$, $i_S = j_S..1$ do
        compute $N(i_R, j_R, i_S, j_S)$ ;
        if $j_R$ is candidate for $i_R$ and Eq. 1(d) then
            store $N^{\mathbf{R}}(i_R + 1, j_R - 1, i_S, j_S)$ in list for $i_R$ and Eq. 1(d)
        else if $j_S$ is candidate for $i_S$ and Eq. 1(e) then
            store $N^{\mathbf{S}}(i_S + 1, j_S - 1)$ in list for $i_S$ and Eq. 1(e)
        else if candidate for Eq. 1(f) then
            store $N(i_R + 1, j_R - 1, i_S, j_S)$ in list for $(i_R, i_S)$ and Eq. 1(f)
        else if candidate for Eq. 1(g) then
            store $N(i_R, j_R, i_S + 1, j_S + 1)$ in list for $(i_R, i_S)$ and Eq. 1(g)
        end
    end
    free matrix slice $N(\cdot, j_R - 1, \cdot, \cdot)$ ;
end
```

some candidate of the respective case, i.e. in case Eq. 1(d) $j'_R + 1$ is a candidate for $i'_R - 1 = i_R$, in case (e) $j'_S + 1$ is a candidate for $i'_S - 1 = i_S$, in case (f) $(j'_R + 1, j'_S)$ is a candidate for $(i'_R - 1, i'_S) = (i_R, j_R)$, and in case (g) $(j'_R, j'_S + 1)$ is a candidate for $(i'_R, i'_S - 1) = (i_R, j_R)$. As shown in the following algorithm, all values that satisfy (i) can be stored in a three dimensional matrix and all values that satisfy (ii) can be stored in candidate lists of length $\psi(n)$ for each of the $O(n^2)$ instances of (i_R, i_S).

Note that, in the pseudocode, we maintain two three dimensional matrices, namely $N(\cdot, j_R, \cdot, \cdot)$ and $N(\cdot, j_R - 1, \cdot, \cdot)$ during the computation of the values for j_R. In practice, we save half of this memory, because any entry $N(\cdot, j_R - 1, \cdot, j_s)$ can be freed as soon as all $N(\cdot, j_R, \cdot, j_S)$ are computed.

Trace-Back We describe the recursive trace-back starting from a matrix entry (i_R, j_R, i_S, j_S). Computing the Trace-back involves some recomputation. First, the entire matrix slice $N(\cdot, j_R, \cdot, j_S)$ is recomputed unless it is already in memory. This requires access to only entries in the same matrix slice and candidates. Then, the best case in the recursion for $N(i_R, j_R, i_S, j_S)$ is identified. In cases (a)-(c), we recurse to the respective entry. In cases (d)-(g), which split in a first and second entry, we first recurse to the second one, which is in the same matrix slice. Then, we free the memory for the current matrix slice and recurse to the first entry, which will cause recomputation. Since each entry is recomputed at most once, the trace-back does not affect the asymptotic complexity.

3.2 Sparsification for Minimizing Free Energy

Alkan et al. [1] describe minimization of the free energy of RNA-RNA-interaction based on a simple stacked-pair energy model assuming there are no pseudoknots, crossing bonds, and zigzags in the joint structure. Here we discuss an algorithm for RNA-RNA interaction free energy minimization on the same type of interactions based on the interaction energy model of Chitsaz et al. [2]. Since the general recursive structure of this algorithm is identical to base pair maximization, our sparsification technique can be applied to reduce their time and space complexity in the same way. The exact recursions of our sparsified free energy minimization algorithm are given in the appendix. Compared to base pair maximization, these recursions distinguish several matrices representing differently scored substructures. Notably, they are formulated such that all cases that split an entry (i_R, j_R, i_S, j_S) at (k_R, k_S) are of the same form as cases Eq. 1(f) and (g) or k_R and k_S are bounded due to the loop length restriction of the energy model. Achieving the same space complexity requires one additional consideration. For assigning correct energy to internal loops formed by interaction arcs, an entry (i_R, j_R, i_S, j_S) can depend on (i'_R, j'_R, i_S, j_S), where j'_R is neither j_R nor $j_R - 1$. However, $j_R - j'_R$ is still bounded by the maximal loop length ℓ of the energy model, i.e. $j_R - j'_R < \ell$. Hence, it suffices to store ℓ matrix slices $(\cdot, j'_R, \cdot, \cdot)$ for $j_R - \ell < j'_R \leq j_R$.

Theorem 3. *The MFE interaction of two RNAs of maximal length n can be computed in expected time $O((\psi_1(n) + \psi_2(n))n^4)$ and expected space $O((\psi_1(n) + \psi_2(n))n^2 + n^3)$.*

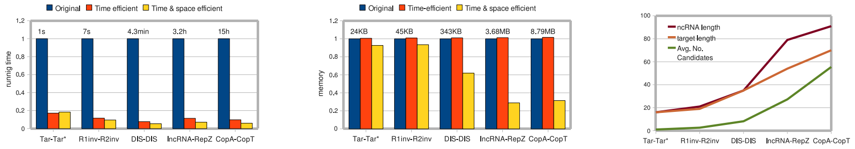
4 Experimental Results

For evaluating the effect of sparsification on RNA-RNA-interaction, we implemented three variants of the total free energy minimization algorithm for RNA-RNA-interaction prediction: the first variant does not perform any sparsification, the second employs sparsification for improving the time complexity, and the third improves both time and space complexity. Below, we first demonstrate that sparsification leads to a significant reduction of the time and space requirements in practice. Then we study the relationship between the sequence length and the number of candidates per each base on a large set of confirmed RNA-RNA interactions and study the average time/space behavior of the algorithms.

Since sparsification does not affect the calculated free energy values (i.e. optimality of the calculated joint free energy of the interaction), the accuracy of the predicted interactions is identical to previous approaches for general RNA-RNA-interactions based on the same scoring scheme [1, 2, 32]. As a result, the reader is referred to Salari et al. [32] for an assessment of sensitivity, positive prediction value, and F-measure of these methods (which will be identical to that of the method presented here) on the data set of Kato et al. [37] which involves five distinct RNA-RNA interactions.

Time and space requirements of total free energy minimization

We applied the three variants of the MFE algorithm to five distinct RNA-RNA interactions reported by Kato et al. [37], which were used by Salari et al. [32] to



(a) Run-time improvement (b) Space improvement (c) Average No. candidates

Fig. 2. Performance of three variants of the RNA-RNA interaction prediction algorithm via total free energy minimization, on a set of interactions compiled by Kato et al. [37]. All values for time and space usage are normalized by the usage of the non-sparsified algorithm, for which absolute time/space usage figures are also given.

assess the accuracy of available RNA-RNA interaction methods with no sparsification. Note that the available methods are not capable of handling interactions involving longer RNAs.

Fig. 2 shows (in absolute terms) time and space usage of the algorithms (with or without sparsification) on a Sun Fire X4600 server with 2.6 GHz processor speed. The results show that sparsification significantly improves the performance of the algorithms. In fact, Fig. 2 demonstrates that as the RNA sequences in question get longer, the relative performance of the sparsified algorithms (with respect to the non-sparsified ones) improve. Although the pure time optimization causes a small space overhead due to maintaining the candidate lists, the time and space optimization not only improve the space utilization, as expected, but also results in further reduction in running time.

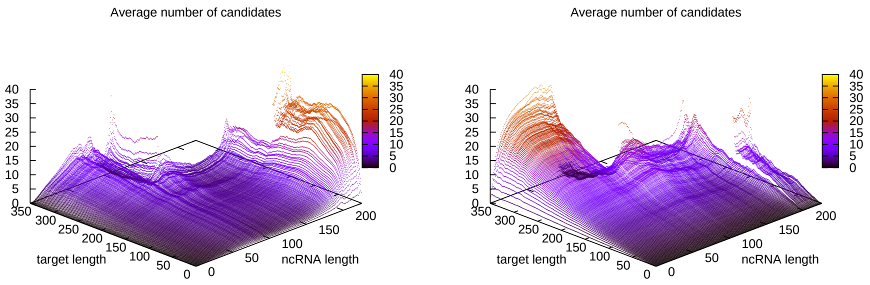
Number of Candidates

The time and space complexity of the (time and space) sparsified RNA-RNA-interaction prediction algorithm is linearly proportional to the (average) number of interaction partner candidates per base. Fig. 2(c) shows how the average number of candidates (k_R, k_S) change as the lengths of the two RNA sequences increase. While the non-sparsified algorithms need to consider a quadratic number of split points (k_R, k_S), the number of candidates (and hence the number of split points) is much lower for the sparsified algorithms.

In order to observe the effects of sparsification on a much larger data set involving longer RNA sequences, we employ the algorithm for RNA-RNA interaction prediction which maximizes the number of (internal and external) base pairs. The data set we use for this purpose includes 43 pairs of ncRNAs and their known target mRNAs. This set not only includes (i) the data set of Kato et al. [37], but also (ii) a recently compiled test set of Busch et al. [30] consisting of 18 sRNA-target pairs, as well as (iii) all ncRNA-target interactions of E.coli from NPinter [38]. Among these interactions 32 are from E.coli, 8 are from Salmonella typhimurium and 3 are from HIV. Since the majority of the known ncRNAs bind to their target mRNAs in close proximity of the start codon, we extracted - as the target region - a subsequence comprising 300nt upstream and 50nt downstream of the first base of the start codon of each mRNA from GenBank [39]. As a result, the maximum sequence length is 227nt for ncRNAs and 350nt for target mRNAs.

The experimental results on this larger data set confirm that the sparsification technique works for a single RNA folding via base pair maximization: the average number of candidates for those cases is low (roughly 5) as previously reported by Wexler et al. [18].

The recursion cases Eq. 1(f) and (g) split both RNAs simultaneously at points (k_R, k_S). Therefore they dominate the running time of the algorithm. For these cases, we counted the candidates that were considered during the computation of (the maximum number of base pairs of) each subsequence pair. The average number of candidates for different subsequence lengths, both for ncRNAs and mRNAs are depicted in Fig. 3 - specific cases that correspond to Eq. 1(f) as well as Eq. 1(g) are provided separately. Note that the average number of candidates are generally low regardless of the sequence lengths: among all possible combinations of split points (k_R, k_S) (respectively in ncRNA and mRNA), even for the longest subsequences (e.g. ncRNA length $l_S = 252$ and mRNA length $l_R = 202$), no more than 40 pairs (of the possible 252 x 202 = $50,904$ combinations for this example) are actual candidates on the average.[2]



(a) for case Eq. 1(f)

(b) for case Eq. 1(g)

Fig. 3. Average number of candidates as a function of subsequence lengths.

Conclusion

In this paper, we consider the problem of predicting the joint structure of two interacting RNAs via minimizing their total free energy as a tool for detecting/verifying mRNA targets of regulatory ncRNAs. Earlier approaches to the

[2] Note that certain combinations of l_R and l_S there is no value for the number of candidates due to the fact that there is no data for $l_R > 111$ and $l_S > 252$ as well as $l_R > 202$ and $l_S > 153$.

problem either use a restricted interaction model, not covering many known joint structures, or require significant computational resources for many practical instances. Here we show that sparsification, a technique that has been applied to single RNA folding, can be applied to the problem of RNA-RNA interaction prediction, to significantly improve both the running time and the space utilization of these approaches. In fact, by employing a version of the polymer-zeta property for interacting RNA-structures (a property generally assumed to be held by many polymers, and has been empirically shown for single RNAs), we show how to reduce the running time and space of RNA-RNA interaction prediction, from $O(n^6)$ time and $O(n^4)$ space to $O(n^4\psi(n))$ time and $O(n^2\psi(n)+n^3)$ space, for a function $\psi(n)=O(n)$ on average. These theoretical predictions are verified by our experiments; as a result it is now possible to employ computational prediction of RNA-RNA interactions to a much wider range of potential regulatory ncRNAs and their targets.

Acknowledgments

R. Salari was supported by SFU-CTEF funded Bioinformatics for Combating Infectious Diseases Project co-lead by Sahinalp. The research of M. Möhl was funded by the German Research Foundation (DFG grant BA 2168/3-1). S.C. Sahinalp was supported by MITACS, NSERC, the CRC program and the Michael Smith Foundation for Health Research. R. Backofen received funding from the German Research Foundation (DFG grant BA 2168/2-1 SPP 1258), and from the German Federal Ministry of Education and Research (BMBF grant 0313921 FRISYS).

References

1. Alkan, C., Karakoc, E., Nadeau, J.H., Sahinalp, S.C., Zhang, K.: RNA-RNA interaction prediction and antisense RNA target search. Journal of Computational Biology (Special RECOMB 2005 Issue) **13**(2) (2006) 267–82
2. Chitsaz, H., Salari, R., Sahinalp, S.C., Backofen, R.: A partition function algorithm for interacting nucleic acid strands. Bioinformatics (Special ISMB/ECCB 2009 Issue) **25**(12) (June 2009) i365–73
3. Storz, G.: An expanding universe of noncoding RNAs. Science **296**(5571) (2002) 1260–3
4. Bartel, D.P.: MicroRNAs: genomics, biogenesis, mechanism, and function. Cell **116**(2) (2004) 281–97
5. Hannon, G.J.: RNA interference. Nature **418**(6894) (2002) 244–51
6. Zamore, P.D., Haley, B.: Ribo-gnome: the big world of small RNAs. Science **309**(5740) (2005) 1519–24
7. Wagner, E., Flardh, K.: Antisense RNAs everywhere? Trends Genet. **18** (May 2002) 223–226
8. Brantl, S.: Antisense-RNA regulation and RNA interference. Bioch. Biophys. Acta **1575**(1-3) (2002) 15–25
9. Gottesman, S.: Micros for microbes: non-coding regulatory RNAs in bacteria. Trends in Genetics **21**(7) (2005) 399–404

10. Seeman, N.: From genes to machines: DNA nanomechanical devices. Trends Biochem. Sci. **30** (Mar 2005) 119–125
11. Seeman, N.C., Lukeman, P.S.: Nucleic acid nanostructures: bottom-up control of geometry on the nanoscale. Reports on Progress in Physics **68** (January 2005) 237–270
12. Simmel, F., Dittmer, W.: DNA nanodevices. Small **1** (Mar 2005) 284–299
13. Venkataraman, S., Dirks, R., Rothemund, P., Winfree, E., Pierce, N.: An autonomous polymerization motor powered by DNA hybridization. Nat Nanotechnol **2** (Aug 2007) 490–494
14. Yin, P., Hariadi, R., Sahu, S., Choi, H., Park, S., Labean, T., Reif, J.: Programming DNA tube circumferences. Science **321** (Aug 2008) 824–826
15. Pervouchine, D.D.: IRIS: intermolecular RNA interaction search. Genome Inform **15**(2) (December 2004) 92–101
16. Huang, F.W., Qin, J., Reidys, C.M., Stadler, P.F.: Partition Function and Base Pairing Probabilities for RNA-RNA Interaction Prediction. Bioinformatics (2009) In press.
17. David, L., Huber, W., Granovskaia, M., Toedling, J., Palm, C.J., Bofkin, L., Jones, T., Davis, R.W., Steinmetz, L.M.: A high-resolution map of transcription in the yeast genome. Proc. Natl. Acad. Sci. USA **103**(14) (2006) 5320–5
18. Wexler, Y., Zilberstein, C., Ziv-Ukelson, M.: A study of accessible motifs and RNA folding complexity. Journal of Computational Biology (Special RECOMB 2006 Issue) **14**(6) (2007) 856–72
19. Backofen, R., Tsur, D., Zakov, S., Ziv-Ukelson, M.: Sparse RNA folding: Time and space efficient algorithms. In Kucherov, G., Ukkonen, E., eds.: Proc. 20th Symp. Combinatorial Pattern Matching. Volume 5577 of LNCS., Springer (2009) 249–262
20. Sankoff, D.: Simultaneous solution of the RNA folding, alignment and protosequence problems. SIAM J. Appl. Math. **45**(5) (1985) 810–825
21. Ziv-Ukelson, M., Gat-Viks, I., Wexler, Y., Shamir, R.: A faster algorithm for rna co-folding. In Crandall, K.A., Lagergren, J., eds.: WABI 2008. Volume 5251 of Lecture Notes in Computer Science., Springer (2008) 174–185
22. Rehmsmeier, M., Steffen, P., Hochsmann, M., Giegerich, R.: Fast and effective prediction of microRNA/target duplexes. RNA **10**(10) (2004) 1507–17
23. Tjaden, B., Goodwin, S.S., Opdyke, J.A., Guillier, M., Fu, D.X., Gottesman, S., Storz, G.: Target prediction for small, noncoding RNAs in bacteria. Nucleic Acids Research **34**(9) (2006) 2791–802
24. Andronescu, M., Zhang, Z.C., Condon, A.: Secondary structure prediction of interacting RNA molecules. Journal of Molecular Biology **345**(5) (2005) 987–1001
25. Bernhart, S.H., Tafer, H., Muckstein, U., Flamm, C., Stadler, P.F., Hofacker, I.L.: Partition function and base pairing probabilities of RNA heterodimers. Algorithms Mol Biol **1**(1) (2006) 3
26. Dirks, R.M., Bois, J.S., Schaeffer, J.M., Winfree, E., Pierce, N.A.: Thermodynamic analysis of interacting nucleic acid strands. SIAM Review **49**(1) (2007) 65–88
27. Zuker, M.: Prediction of RNA secondary structure by energy minimization. Methods in Molecular Biology **25** (1994) 267–94
28. Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, S., Tacker, M., Schuster, P.: Fast folding and comparison of RNA secondary structures. Monatshefte Chemie **125** (1994) 167–188
29. Muckstein, U., Tafer, H., Hackermuller, J., Bernhart, S.H., Stadler, P.F., Hofacker, I.L.: Thermodynamics of RNA-RNA binding. Bioinformatics **22**(10) (2006) 1177–82

30. Busch, A., Richter, A.S., Backofen, R.: IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. Bioinformatics **24**(24) (2008) 2849–56

31. Argaman, L., Altuvia, S.: fhla repression by OxyS RNA: kissing complex formation at two sites results in a stable antisense-target RNA complex. Journal of Molecular Biology **300**(5) (2000) 1101–12

32. Salari, R., Backofen, R., Sahinalp, S.C.: Fast prediction of RNA-RNA interaction. In: Proc. of the 9th Workshop on Algorithms in Bioinformatics (WABI). (2009) 261–272 Also *Algorithms for Molecular Biology* In press.

33. Chitsaz, H., Backofen, R., Sahinalp, S.C.: biRNA: Fast RNA-RNA binding sites prediction. In: Proc. of the 9th Workshop on Algorithms in Bioinformatics (WABI). (2009) 25–36

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

35. fisher, M.E.: Shape of a self-avoiding walk or polymer chain. JOURNAL OF CHEMICAL PHYSICS **44** (1966) 616–622

36. Kafri, Y., Mukamel, D., Peliti, L.: Why is the DNA denaturation transition first order? Phys. Rev. Lett. **85** (Dec 2000) 4988–4991

37. Kato, Y., Akutsu, T., Seki, H.: A grammatical approach to rna-rna interaction prediction. Pattern Recogn. **42**(4) (2009) 531–538

38. Wu, T., Wang, J., Liu, C., Zhang, Y., Shi, B., Zhu, X., Zhang, Z., Skogerb, G., Chen, L., Lu, H., Zhao, Y., Chen, R.: NPInter: the noncoding RNAs and protein related biomacromolecules interaction database. Nucleic Acids Res. **34** (Jan 2006) D150–152

39. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Wheeler, D.L.: GenBank. Nucleic Acids Research **36**(Database issue) (2008) D25–30

APPENDIX

Total number of fragments for different ncRNA and target subsequence lengths The plot of Fig. 4 shows the total number of fragments for different ncRNA and target subsequence lengths. The white region on top right of the plot in Fig. 4 ($l_R > 111 \wedge l_S > 252$ and $l_R > 202 \wedge l_S > 153$) denotes the area that there are no fragments in our data set.

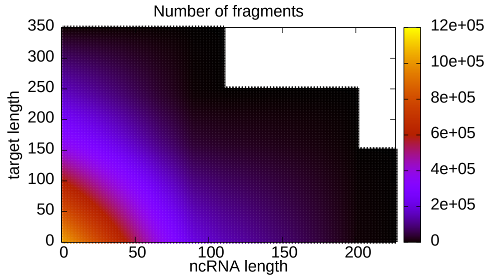


Fig. 4. Total number of fragments for different ncRNA and target lengths.

Sparsification of Energy Minimization RNA-RNA-Interaction Here, we present our sparsified algorithm for RNA-RNA interaction free energy minimization based on the interaction energy model of Chitsaz et al. [2]. The minimum free energy (mfe) joint structure $M(i_R, j_R, i_S, j_S)$ derived from one of the seven possible cases shown in Fig. 4(b). The first two cases are when i_R or i_S is an unpaired base. In third case i_R interacts with i_S, this bond starts a special type of joint structure denoted by Ib and it is explained in Fig. 4(c). The forth and fifth cases are when i_R or i_S is forming intramolecular base pairs. In other possible cases either $i_R \bullet k_R$ is an interaction arc subsuming $[i_S, k_S]$ or $i_S \bullet k_S$ is an interaction arc subsuming $[i_R, k_R]$. The sparsified DP algorithm for free energy minimization, $M(i_R, j_R, i_S, j_S)$, is defined as follows:

$$M(i_R, j_R, i_S, j_S) = \max \begin{cases} M(i_R+1, j_R, i_S, j_S) & (a) \\ M(i_R, j_R, i_S+1, j_S) & (b) \\ M^{Ib}(i_R, j_R, i_S, j) & (c) \\ \max\limits_{\substack{i_R < k \leq j_R \\ k \text{ cand. for } (i_R)}} \begin{pmatrix} M^{\mathbf{R}.b}(i_R, k) \\ \quad + M(k+1, j_R, i_S, j_S) \end{pmatrix} & (d) \\ \max\limits_{\substack{i_S \leq k < j_S \\ k \text{ cand. for } (i_S)}} \begin{pmatrix} M^{\mathbf{S}.b}(i_S, k) \\ \quad + M(i_R, j_R, k+1, j_S) \end{pmatrix} & (e) \\ \max\limits_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ (k_R, k_S) \text{ cand. for } (i_R, i_S)}} \begin{pmatrix} M^{Is}(i_R, k_R, i_S, k_S) \\ \quad + M(k_R+1, j_R, k_S+1, j_S) \end{pmatrix} & (f) \\ \max\limits_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ (k_R, k_S) \text{ cand. for } (i_R, i_S)}} \begin{pmatrix} M^{Is'}(i_R, k_R, i_S, k_S) \\ \quad + M(k_R+1, j_R, k_S+1, j_S) \end{pmatrix} & (g) \end{cases} \tag{6}$$

$$M^{\mathbf{X}}(i, j) = \max \begin{cases} M^{\mathbf{X}}(i+1, j) & (a) \\ \max\limits_{\substack{i < k \leq j \\ (k) \text{ cand. for } (i)}} \begin{pmatrix} M^{\mathbf{X}.b}(i, k) \\ \quad + M^{\mathbf{X}}(k+1, j) \end{pmatrix} & (b) \end{cases} \tag{7}$$

$M^{Ib}(i_R, j_R, i_S, j_S)$ (Fig. 4(c)) is the mfe for the joint structure of $[i_R, j_R]$ and $[i_S, j_S]$ assuming $i_R \cdot j_S$ is an interaction bond, and $M^{Is}(i_R, j_R, i_S, j_S)$ (Fig. 4(d)) is the mfe for the joint structure of $[i_R, j_R]$ and $[i_S, j_S]$ assuming $i_R \circ j_R$ is an interaction arc subsuming $[i_S, j_S]$. $M^{Is'}$ is symmetric to M^{Is} where $i_S \circ j_S$ is an interaction arc subsuming $[i_R, j_R]$. In Q^{Isl}, $[i_S, j_S]$ contains at least interaction arc and in Q^{Isk}, $[i_S, j_S]$ contains at least one direct bond. The other auxiliary matrices are Q^{Ill}, Q^{Ilk}, Q^{Ikl}, and Q^{Ikk} (Fig. 4(g)). Q^{Ill} includes all cases where both $[i_R, j_R]$ and $[i_S, j_S]$ have at least one interaction arc. Q^{Ilk} (symmetric to Ikl) includes all cases where $[i_R, j_R]$ has at least one interaction arc and $[i_S, j_S]$ has at least one direct bond. Q^{Ikk} includes all cases where both $[i_R, j_R]$ and $[i_S, j_S]$ have at least one direct bond.

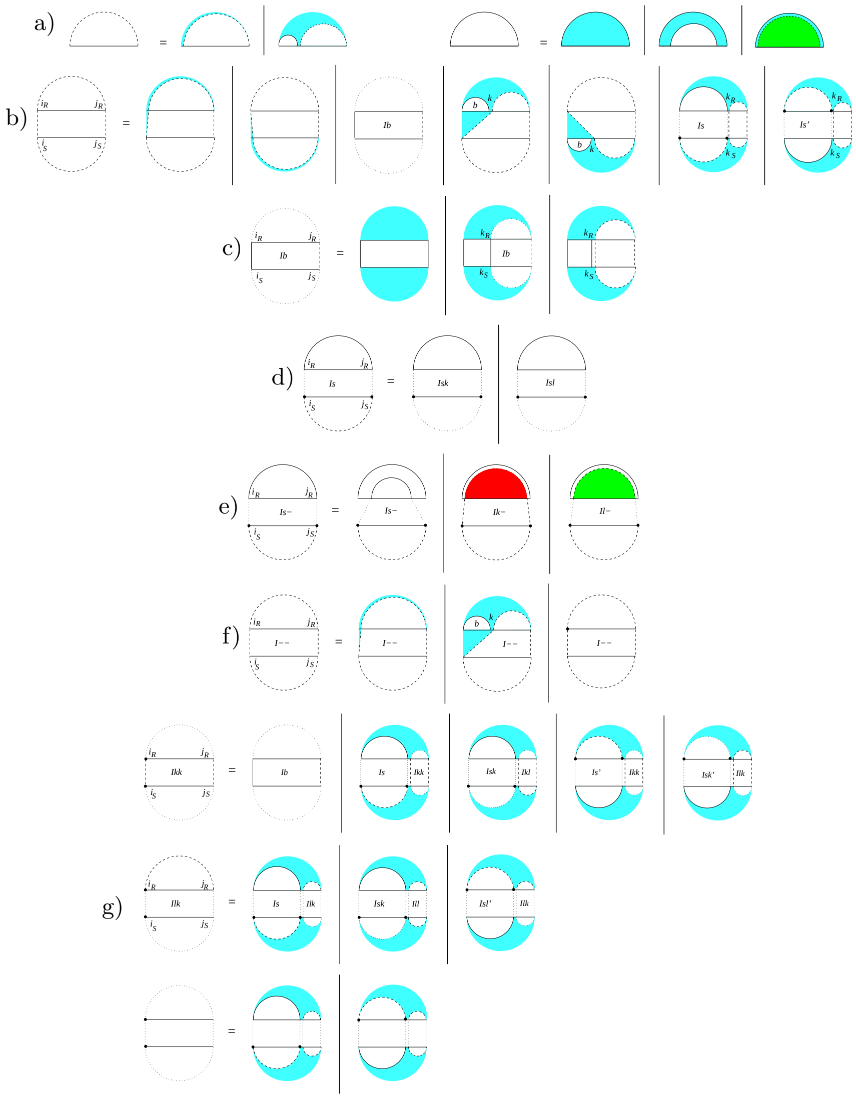


Fig. 5. a) Recursion cases for MFE single structure. b) Recursion cases for MFE joint structure. c) Recursion cases for MFE joint structure while $i_R \circ j_S$ is a bond. Here $i_R < k_R \leq \min i_R + \ell, j_R$ and $i_S < k_S \leq \min i_S + \ell, j_S$ w. ℓ is the maximal loop length. d) In recursive quantity Is, $i_R \bullet j_R$ is an interaction arc which subsumes interval $[i_S, j_S]$. The subsumed area contains at least one direct bond or at least one interaction arcs. e) Recursion cases for Isl or Isk which extract the interaction arc $i_R \bullet j_R$. f) In Ikk, Ikl, Ilk, or Ill, if the terminal point i_R (or j_S) is not an end point of interaction bond or arc, some recursions should be applied to extract the internal structure. g) Recursion for joint structures that has direct interactions on both subsequences (Ikk), direct interaction on one subsequence and interaction arc on the other (Ikl and Ilk which are symmetric), and interaction arcs on both subsequences (Ill).