BPPart and BPMax: RNA-RNA Interaction Partition Function and Structure Prediction for the Base Pair Counting Model

Ali Ebrahimpour-Boroojeny  Sanjay Rajopadhye  Hamidreza Chitsaz*
Department of Computer Science, Colorado State University

Abstract

RNA-RNA interaction (RRI) is ubiquitous and has complex roles in the cellular functions. In human health studies, miRNA-target and lncRNAs are among an elite class of RRIs that have been extensively studied and shown to play significant roles in various diseases including cancer. Bacterial ncRNA-target and RNA interference are other classes of RRIs that have received significant attention. Accordingly, RRI bioinformatics tools tailored for those elite classes have been proposed in the last decade.

Interestingly, there are instances of mRNA-mRNA interactions where both partners appear in the same KEGG pathway without any direct link between them, or any prior knowledge in the literature about their relationship. Those recently discovered cases suggest that RRI scope is much wider than those aforementioned elite classes. Hence, there is a need for high-throughput generic RNA-RNA interaction bioinformatics tools.

In this paper, we revisit our RNA-RNA interaction partition function algorithm, piRNA, which happens to be the most comprehensive, and albeit the most computationally-intensive, thermodynamic model for RNA-RNA interaction. piRNA computes the partition function, base-pairing probabilities, and structure for the comprehensive Turner energy model using 96 different dynamic programming tables. In this study, we embark on a journey to retreat from sophisticated thermodynamic models to much simpler models such as base pair counting. That might seem counter-intuitive at the first glance; however, our idea is to benefit from the advantages of such simple models in terms of implementation, tuning, running time, and memory footprint and compensate for the associated information loss by adding data-oriented machine learning components in the future to the pipeline.

In this work, we simplify the energy model and instead consider only simple weighted base pair counting to obtain BPPart algorithm for Base-pair Partition function and BPMax for Base-pair Maximization, which use 9 and 2 tables respectively. They are empirically 225 and 1350 fold faster than piRNA due to reduction in the number of table look-ups and the fact that the 96 tables of piRNA make the CPU cache much less effective. After a search-based optimization of the weights of base pairs, a correlation of 0.855 and 0.836 was achieved between piRNA and BPPart and between piRNA and BPMax, respectively, in 37°C on 50,500 experimentally characterized RRIs. This correlation increases to 0.920 and 0.904 for those algorithms, respectively, in −180°C due to a decrease in the effect of thermodynamic entropy in lower temperatures.

The results show that simplifying the model does not result in a noticeable loss of the thermodynamic information that piRNA captures. Therefore, the proposed algorithms can be

*Corresponding Author. Email: chitsaz@chitsazlab.org
used along with machine learning methods in a strategic retreat from slower comprehensive physical models such as piRNA.

1 Introduction

Since mid 1990s with the advent of RNA interference discovery, RNA-RNA interaction (RRI) has moved to the spotlight in modern, post-genome biology. RRI is ubiquitous and has increasingly complex roles in cellular functions. In human health studies, miRNA-target and IncRNAs are among an elite class of RRIs that have been extensively studied and shown to play significant roles in various diseases including cancer. Bacterial ncRNA-target and RNA interference are other classes of RRIs that have received significant attention. However, new evidence suggests that other classes of RRI, such as mRNA-mRNA interactions, are biologically important.

The RISE database [1] reports a number of biologically significant instances of mRNA-mRNA interactions. These representative mRNA-mRNA interactions suggest that general RRIs, including mRNA-mRNA interactions, play major roles in human biology. Hence, there is a need for high-throughput generic RNA-RNA interaction bioinformatics tools for all types of RNAs. As an example of this necessity for all types of RNAs, we found 3 cliques of size 4 of interacting protein-coding RNAs in ribosome which conform to what we generally expect from the structure of the ribosome. These cliques are highly entangled together to form an interaction graph as Figure 1. RPS3 which seems to be one of the genes with the highest number of connections interacts with at least 14 other genes in ribosome pathway. Another interesting clique of size 4 that we could find consists of 4 genes in the pathway of regulation of actin cytoskeleton, ACTB, ACTG1, PFN1, and TMSB4X. These genes are involved in vital tasks of proliferation, migration, mobility, and differentiation of the cell. Being able to capture all the interactions that RNAs might have helps us to better understand the post-transcriptional regulation of the genes.

Figure 1: A substructure of the genes in the ribosome pathway. Each node represents a gene and each edge represents an experimentally observed interaction between the corresponding genes.

In this paper, we revisit our RNA-RNA interaction partition function algorithm, piRNA, which happens to be the most comprehensive, albeit the most computationally intensive, thermodynamic
model for RNA-RNA interaction [2]. \textit{piRNA} is a dynamic programming algorithm that computes the partition function, base-pairing probabilities, and structure for the comprehensive Turner energy model in $O(n^4m^2 + n^2m^4)$ time and $O(n^4 + m^4)$ space. Due to intricacies of the energy model, including various loops such as hairpin loop, bulge/internal loop, and multibranch loop, \textit{piRNA} involves 96 different dynamic programming tables and needs multiple table look-ups for computing their values.

In this paper, we introduce a strategic retreat from the slower comprehensive models such as \textit{piRNA} by simplifying the energy model and instead considering only simple weighted base pair counting to obtain \textit{BPPart} algorithm for Base-pair Partition function and \textit{BPMax} for Base-pair Maximization, which are much faster. By the explosion of experimental data which makes us able to use machine learning methods, such as deep learning, for detection of RNA subsequences that interact, this retreat is necessary if one is willing to build physics-guided models by using the features that are derived by an energy model. \textit{BPPart} involves 10 dynamic programming tables, and \textit{BPMax} involves only 2 tables. Both \textit{BPPart} and \textit{BPMax} compared with \textit{piRNA} are much simpler dynamic programming algorithms which are more than 225 fold and 1300 fold faster, respectively, on the 50500 RRI samples we used for our experiments. The reason for this noticeable speed-up is reducing the number of tables and the number of table look-ups for computing the new values and also the fact that the 96 large tables of \textit{piRNA} reduces the efficiency of using the cache. Moreover, from the point of view of code optimization and development/debugging for different hardware platforms, it is much more convenient to work with \textit{BPPart} and \textit{BPMax} because of the significantly reduced memory footprint, and this provides room for further optimization of these methods in the future.

The first question is, how much accuracy do we lose by simplifying the scoring model from the comprehensive Turner model to simply weighted base pair counting? We answer that question by computing both the Pearson and rank correlations in different temperatures between the results of \textit{BPPart}, \textit{BPMax}, and \textit{piRNA} on 50,500 experimentally characterized RRIs in the RISE database [1]. We find that the Pearson correlations between \textit{BPPart} and \textit{piRNA} is 0.957 and \textit{BPMax} and \textit{piRNA} is 0.941 at $-180^\circ C$ after optimizing the weights for base pairs. As the temperature increases, the effect of entropy, which is not taken into account in the simple base pair counting model, increases. Completely conforming with the theoretical expectations, we find that the Pearson correlations between \textit{BPPart} and \textit{piRNA} and also between \textit{BPMax} and \textit{piRNA} is 0.883 at $37^\circ C$. We conclude that both \textit{BPPart} and \textit{BPMax} capture a significant portion of the thermodynamic information that can possibly be complemented with machine learning techniques in the future for more accurate predictions.

\section*{Related work}

During the last few decades, several computational methods emerged to study the secondary structure of single and interacting nucleic acid strands. Most use a thermodynamic model such as the well-known Nearest Neighbor Thermodynamic model [3, 4, 5, 2, 6, 7, 8, 9, 10, 11]. Some previous attempts to analyze the thermodynamics of multiple interacting strands concatenate input sequences in some order and consider them as a single strand [12, 13, 14]. Alternatively, several methods avoid internal base-pairing in either strand and compute the minimum free energy secondary structure for their hybridization under this constraint [15, 16, 17]. The most comprehensive solution is computing the joint structure between two interacting strands under energy models with a growing complexity [18, 19, 20, 21, 22, 23].
Other methods predict the secondary structure of individual RNA independently, and predict the (most likely) hybridization between the unpaired regions of the two interacting molecules as a multistep process: 1) unfolding of the two molecules to expose bases needed for hybridization, 2) the hybridization at the binding site, and 3) restructuring of the complex to a new minimum free energy conformation [24, 25, 26, 27]. The success of such methods, including our biRNA algorithm [27], suggests that the thermodynamic information vested in subsequences and pairs of subsequences of the input RNAs can provide valuable information for predicting features of the entire interaction.

In addition to general RNA-RNA interaction tools, many tools have been developed to predict the secondary structure of interacting RNAs for a specific type of interest which has been shown to be more effective in some cases due to the utilization of certain properties belonging to that type. As mentioned earlier, miRNA-target prediction is one such class of high interest for which such specialized tools have been created to incorporate various properties specific to miRNAs; some of these tools use the seed region of a miRNA which is highly conserved [28, 29, 30, 31], some consider the free energy to compute accessibility to the binding site in 3’ UTR [32, 20, 29], some utilize the conservation level which is derived using the phylogenetic distance [33, 34, 35, 36, 28, 29], and some others consider other target sites as well, such as the 5’ UTR, Open Reading Frames (ORF), and the coding sequence (CDS) for mRNAs [37, 38, 39, 40].

There are also several other tools developed for other specific types of RNA; IntaRNA [41, 42] is one such tool that although is used for RNA-RNA interaction in general, it is primarily designed for predicting target sites of non-coding RNAs (ncRNAs) on mRNAs. There are many other examples, such as PLEXY [43] which is a tool designed for C/D snoRNAs, RNAsnoop [44] that is designed for H/ACA snoRNAs, TargetRNA [45] which is a tool aimed at predicting interaction of bacterial sRNAs [46].

2 Methods

Here we describe how our algorithm, BPPart, utilizes a dynamic programming approach to compute the partition function for RNA-RNA interaction when entropy is ignored and only a weighted score for pairing different nucleotides is considered. This algorithm guarantees to be mutually exclusive on the set of structures; in other words, it counts each structure exactly once.

2.1 Preliminaries

In this paper, we mostly follow the notations and definition of the authors of piRNA [2]. We denote the two nucleic acid strands by \( R \) and \( S \). Strand \( R \) is indexed from 1 to \( L_R \), and \( S \) is indexed from 1 to \( L_S \) both in 5’ to 3’ direction. Note that the two strands interact in opposite directions, e.g. \( R \) in 5’ → 3’ with \( S \) in 3’ ← 5’ direction; however, we consider the reverse of \( S \) in the figures and equations for the sake of easier illustration and convenience. Each nucleotide is paired with at most one nucleotide in the same or the other strand. The subsequence from the \( i^{th} \) nucleotide in one strand to the \( j^{th} \) nucleotide in a strand is denoted by \([i, j]\).

An intramolecular base pair between the nucleotides \( i \) and \( j \) in a strand is called an arc and denoted by a bullet \( i \bullet j \). We represent the score of such arc by \( \text{score}(i, j) \). An intermolecular base pair between the nucleotides \( i_1 \) and \( i_2 \) is called a bond and denoted by a circle \( i_1 \circ i_2 \). We represent the score of such bond by \( \text{iscore}(i_1, i_2) \). An arc \( i_1 \bullet j_1 \) covers a bond \( k_1 \circ k_2 \) if \( i_1 < k_1 < j_1 \). We call \( i_1 \bullet j_1 \) an interaction arc if there is a bond \( k_1 \circ k_2 \) covered by \( i_1 \bullet j_1 \). We call a base on either
strand an event if it is either the end-point of a bond or an interaction arc.

Assuming \( i_1 < j_1 \), two bonds \( i_1 \circ i_2 \) and \( j_1 \circ j_2 \) are called crossing bonds if \( i_2 > j_2 \). An interaction arc \( i_1 \bullet j_1 \) in a strand subsumes a subsequence \([i_2, j_2]\) in the other strand if for all bonds \( k_1 \circ k_2 \), if \( i_2 \leq k_2 \leq j_2 \) then \( i_1 < k_1 < j_1 \). In other words, none of the bases in \([i_2, j_2]\) has a bond with a base outside the \( i_1 \bullet j_1 \) arc. Two interaction arcs are equivalent if they subsume one another. Two interaction arcs \( i_1 \bullet j_1 \) and \( i_2 \bullet j_2 \) are part of a zigzag, if neither \( i_1 \bullet j_1 \) subsumes \([i_2, j_2]\) nor \( i_2 \bullet j_2 \) subsumes \([i_1, j_1]\).

In this work, we assume there are no pseudoknots in individual secondary structures of \( R \) and \( S \), and also there are no crossing bonds and zigzags between \( R \) and \( S \). These constraints are being made to make the problem a polynomial problem rather than an NP-hard one as the general case of considering all possible structures [19].

We denote the ensemble of unpseudoknotted structures of \( R \) and \( S \) by \( S(R) \) and \( S(S) \) respectively. The ensemble of unpseudoknotted, crossing-free, and zigzag-free joint interaction structures in denoted by \( S_I(R, S) \).

For a given structure \( s \in S(R) \cup S(S) \), let \( AU(s) \) denote the number of A-U base pairs in \( s \). Similarly, \( CG(s) \) and \( GU(s) \) denote the number of C-G and G-U base pairs in \( s \) respectively. We define

\[
\text{bpcount}(s) = c_1 GU(s) + c_2 AU(s) + c_3 CG(s),
\]

in which \( c \)'s are constants. In this study, we try a range of values of these constants. The details and results of using these different values can be found in Section 3.

For a given joint interaction structure \( s \in S_I(R, S) \), let \( AU(s), CG(s), \) and \( GU(s) \) denote the number of corresponding intramolecular base pairs in \( s \) as defined above. Let \( AU^I(s), CG^I(s), \) and \( GU^I(s) \) denote the number of corresponding intermolecular base pairs in \( s \). We define

\[
\text{bpcount}^I(s) = c'_1 GU^I(s) + c'_2 AU^I(s) + c'_3 CG^I(s)
\]

and

\[
\text{bpcount}(s) = c_1 GU(s) + c_2 AU(s) + c_3 CG(s) + \text{bpcount}^I(s),
\]

in which \( c \)'s and \( c' \)'s are the tunable weights for base pairs.

### 2.2 Problem Definition

In this paper, we solve two problems:

1. **Base Pair Counting Partition Function:** we give a dynamic programming algorithm \( \text{BPPart} \) to compute the partition function

\[
Q(R, S) = \sum_{s \in S_I(R, S)} e^{bpcount(s)},
\]

2. **Base Pair Maximization:** we give a dynamic programming algorithm \( \text{BPMax} \) to find the structure that has the maximum weighted base pair count, i.e.

\[
\text{bpmax}(R, S) = \arg\max_{s \in S_I(R, S)} \text{bpcount}(s).
\]
This problem was previously studied by D. Pervouchine in an algorithm called IRIS. However, there is no publicly available functional implementation of IRIS. Moreover, we define a novel interaction score

\[
\text{interaction-score}(R, S) = \max \{ \text{bpcount}^f(s) \mid \text{bpcount}(s) = \text{bpcount}(\text{bpmax}(R, S)) \}
\]  

(6)

and compute it by backtracing all possible optimal structures and selecting the one that has maximum interaction portion.

2.3 BPPart Algorithm

First, we start with the recursions for computing the partition function on a single strand which is going to occur in many cases of the double-stranded version. Let represent the partition function of the subsequence from the \(i^{th}\) nucleotide to the \(j^{th}\) one, inclusive, as \(Q_{i,j}\). As shown in Figure 2, there are two mutually exclusive cases; either there is no arc (the left case) or there is a unique leftmost arc (the right case) which starts at the \(k^{th}\) position. We show the structure that starts at the \(k^{th}\) base in the second case by \(Qz\).

The property of the \(Qz_{i,j}\) is that it has to have at least one arc starting at its first nucleotide, \(i\). Therefore, due to the assumption that no pairing is allowed between two bases that are less than 3 bases apart, for the subsequences of a length less than 5, the value of \(Qz\) is 0. Otherwise, assuming the first nucleotide is paired with the \(k\)th base, as Figure 3 shows, we can split the \(Qz_{i,j}\) structure into a \(Q\) structure inside \(i \cdot k\) and a segment after \(k\), \([k+1,j]\), which is a \(Q\) structure again. Therefore, the value of \(Qz_{i,j}\) can be computed using the equation 8.

According to the explanation and corresponding recursion formulas, equations 7 and 8, we need two 2-dimensional tables for solving the base pair counting partition function on each strand. In the following equations, we distinguish these tables by using superscripts \(^{(1)}\) and \(^{(2)}\) for the first strand (the one that appears at the top in the figures) and the second one (the one at the bottom) respectively.

\[
Q_{i,j} = \begin{cases} 
1 & j \leq i \\
1 + \sum_{k=i}^{j-1} Qz_{k,j} & \text{otherwise}
\end{cases}
\]

(7)

Figure 2: For computing \(Q\), notice that either there is no pairing or there is at least one arc which starts at some index \(k\) and results in a case of \(Qz\).
Figure 3: Computing $Q_z$ can be achieved by considering the base $k$ that is paired with $i$ and the two $Q$ substructures it forms, one between $i$ and $k$ and one after $k$.

$$Q_z(i,j) = \begin{cases} 
0 & j - i < 4 \\
\sum_{k=i+4}^j Q_{i+1,k-1} \times \text{score}(i,k) \times Q_{k+1,j} & \text{otherwise}
\end{cases} \quad (8)$$

Now, for the partition function of a pair of RNA sequences, we consider a 4-dimensional table $Q_I$ in which $Q_{I_{i_1,j_1,i_2,j_2}}$ is the value of base pair counting partition function for the subsequences $[i_1,j_1]$ on $R$ and $[i_2,j_2]$ on $S$. As Figure 4 shows, we can split the set of all possible structures of $Q_I$ into 3 mutually exclusive subsets. The leftmost case shows the structures in which there exist no bonds. Therefore the value of $Q_{I_{i_1,j_1,i_2,j_2}}$ can be computed using the first case of equation (9). The other two cases occur when there is at least one bond; so, there is at least one event on both $R$ and $S$ which we call $k_1$ and $k_2$, respectively. In the second case, these left-most events are end-points of a bond; hence, this case can be broken into a bond-free section on the left side of $k_1 \circ k_2$, and a section called $Q_{Ib}$ which contains the bond itself and a general case of $Q_I$ on the right side of the bond, $(Q_{I_{k_1+1,j_1,k_2+1,j_2}})$. Therefore, we do not need a separate table for $Q_{Ib}$. The third case occurs when $k_1$ and $k_2$ are not end-points of a bond. We call this structure $Q_{Ia}$.

For computing $Q_{Ia_{i_1,j_1,i_2,j_2}}$, we have to consider the property of this structure that the leftmost bases on both $R$ and $S$ have to be events, but they cannot both be the end-points of a bond. Therefore, either one or both of them have to be end-points of an interaction arc. For the case where both $i_1$ and $i_2$ are end-points of some interaction arcs $i_1 \bullet k_1$ and $i_2 \bullet k_2$ such that those arcs are equivalent, $Q_{Ia}$ splits to two exclusive substructures $Q_{Ie_{i_1,j_1,i_2,j_2}}$ and $Q_{I_{k_1+1,j_1,k_2+1,j_2}}$ where $Q_{Ie}$ is a structure in which first and last bases on each strand are paired and the two arcs are equivalent.

In $Q_{Ie_{i_1,j_1,i_2,j_2}}$, if we remove the arcs $i_1 \bullet j_1$ and $i_2 \bullet j_2$, we will get the general case of $Q_{I_{i_1+1,j_1-1,i_2+1,j_2-1}}$ for the inner-section with the constraint that there has be at least one bond in that region because the assumption is that the extracted arcs where interaction arcs. To fulfill this constraint we can exclude all the cases where no bond exists as shown in equation (10). Since this case can be reduced to a special case of $Q_I$, we do not need a separate table for that and we can directly replace it in all other equation with the formula of equation (10).
Figure 4: Each case of $QI$ structure (left side of the equation) can lead to 3 cases. It is clear that either no bonds exist (leftmost case), or at least one bond exists. If the first event on both of the sequences is a bond we have the case $QIb$ (the middle case) which is actually $QI$ with one bond on the left; if not, we will have a case of $QIa$ (the rightmost case).

\[
QI_{i_1,j_1,i_2,j_2} = \begin{cases} 
Q^{(1)}_{i_1,j_1} \times Q^{(2)}_{i_2,j_2} & j_1 < i_1 \text{ or } j_2 < i_2 \\
Q^{(1)}_{i_1,j_1} \times Q^{(2)}_{i_2,j_2} \\
+ \sum_{j_1} \sum_{j_2} Q^{(1)}_{i_1,k_1-1} \times Q^{(2)}_{i_2,k_2-1} \times iscore(k_1,k_2) \times QI_{k_1+1,j_1,k_2+1,j_2} \\
+ \sum_{j_1} \sum_{j_2} Q^{(1)}_{i_1,k_1-1} \times Q^{(2)}_{i_2,k_2-1} \times QIa_{k_1,j_1,k_2,j_2} & \text{otherwise} 
\end{cases}
\]  

\[
QI_{i_1,j_1,i_2,j_2} = \begin{cases} 
0 & j_1 < i_1 + 4 \text{ or } j_2 < i_2 + 4 \\
(QI_{i_1+1,j_1-1,i_2+1,j_2-1} - Q^{(1)}_{i_1+1,j_1-1}) \times iscore(i_1,j_1) \times iscore(i_2,j_2) & \text{otherwise} 
\end{cases}
\]  

Now, for the other cases of $QIa$, we have to consider all the structures where either exactly one of the left-most events on $R$ and $S$ are end-point of a bond or both of them are end-points of some arc where the arcs are not equivalent. Then, $QIa$ can be split into one such structure and a general case of $QI$ on its right side. The set of such structure can be split into two symmetric set of cases for which we will explain the structures covering them.

Let consider $QIs_{i_1,j_1,i_2,j_2}^{(1)}$ as the structure in which we have arc $i_1 \bullet j_1$ and $i_2$ is either the end-point of a bond with the other end at some $k_1$ where $i_1 < k_1 < j_1$, or is the end-point of some arc $i_2 \bullet k_2$ that does not subsume $i_1 \bullet j_1$ and $i_2 < k_2 < j_2$. The other constraint of this set of structures is that $j_2$ is the right-most event on $S$ that is subsumed by $i_1 \bullet j_1$. Also, let consider the symmetric case of this structure as $QIs_{i_1,j_1,i_2,j_2}^{(2)}$. Hence, all other cases of $QIa_{i_1,j_1,i_2,j_2}$ can be
represented as $QIs_{i_1,k_1,i_2,k_2}^{(1)}$ and $QI_{k_1+1,j_1,k_2+1,j_2}^{(1)}$ or $QIs_{i_1,k_1,i_2,k_2}^{(2)}$ and $QI_{k_1+1,j_1,k_2+1,j_2}^{(2)}$.

Figure 5: There are 3 cases for computing the $QIa$ structure; either the leftmost base of only one of the strands is an end point of an arc or both of them.

\[
QIa_{i_1,j_1,i_2,j_2} = \sum_{k_1=i_1}^{j_1} \sum_{k_2=i_2}^{j_2} QIac_{i_1,k_1,i_2,k_2} \times QI_{k_1+1,j_1,k_2+1,j_2}
\]  

(11)

\[
QIac_{i_1,j_1,i_2,j_2} = QIs_{i_1,j_1,i_2,j_2}^{(1)} + QIs_{i_1,j_1,i_2,j_2}^{(2)} + QI_{i_1,j_1,i_2,j_2}
\]

(12)

Since $QIs^{(1)}$ and $QIs^{(2)}$ are symmetric, here we only explain the computation of $QIs^{(1)}$. For computing $QIs_{i_1,j_1,i_2,j_2}^{(1)}$, if we extract $i_1 \cdot j_1$, there remains the subsequence $[i_1+1,j_1-1]$ on $R$ which is known to have at least one event since $i_1 \cdot j_1$ is an interaction arc. Let call the left-most event on $[i_1+1,j_1-1]$ as $k_1$. Therefore, $QIs_{i_1,j_1,i_2,j_2}^{(1)}$ can be split into $QI_{i_1,k_1-1}^{(1)}$ and $QIaux_{k_1,j_1,i_2,j_2}^{(1)}$ which is a structure with the property of having event on $k_1$, $i_2$, and $j_2$, and $i_2 \cdot j_2$ is not allowed.

Figure 6: $QIs^{(1)}$ has one arc that can be extracted and the structure derived will have the property that the two end bases of the bottom strand cannot be paired (the new structure inherits this property from $QIs^{(1)}$). On the top strand, we consider the leftmost event. This new structure is $QIaux^{(1)}$. 

9
\[ QI_{i_1,j_1,i_2,j_2}^{(1)} = \begin{cases} 
0 & j_1 < i_1 + 4 \text{ or } j_2 < i_2 \\
Q_{i_1,j_1}^{(1)} \times Q_{i_2,j_2}^{(2)} & \\
+ \sum_{k_1=i_1}^{j_1} \sum_{k_2=i_2}^{j_2} Q_{i_1,k_1-1}^{(1)} \times Q_{i_2,k_2-1}^{(2)} \\
\times iscore(k_1,k_2) \times QI_{k_1+1,j_1,k_2+1,j_2}^{(1)} \\
+ \sum_{k_1=i_1}^{j_1} \sum_{k_2=i_2}^{j_2} Q_{i_1,k_1-1}^{(1)} \times Q_{i_2,k_2-1}^{(2)} \\
\times QI_{k_1,j_1,k_2,j_2}^{(1)} 
\end{cases} \]

(13)

To compute \( QI_{aux,i_1,j_1,i_2,j_2} \), by considering the right-most event on \( R \), \( k_1 \), we have a \( QI_{k_1+1,j_1}^{(1)} \) structure on the right side of \( k_1 \) on \( R \) and the remaining part is a structure in which all four corners are events, and \( i_2 \circ j_2 \) is not allowed. If there is an arc from \( i_1 \) to \( k_1 \), then we will have another \( QI_{s}^{(1)} \) structure; if not, we call the structure \( QIm \). The property of this new structure is that all the corners are events, but neither the two corners on \( R \) nor the two ones on \( S \) form an arc with one another.

![Figure 7](image)

Figure 7: Two cases must be considered for the \( QI_{aux}^{(1)} \) structure, in which the 2 end points of the bottom strand are events. For the top strand, only the leftmost end point is required to be an event. It can either be the end point of an arc (rightmost case) or not (leftmost case).

\[ QI_{aux,i_1,j_1,i_2,j_2}^{(1)} = \sum_{k_1=i_1}^{j_1} QI_{s,i_1,k_1,j_2}^{(1)} \times QI_{k_1+1,j_1}^{(1)} + \sum_{k_1=i_1}^{j_1} QIm_{i_1,k_1,j_2}^{(1)} \times QI_{k_1+1,j_1}^{(1)} \]

(14)

For computing \( QIm_{i_1,j_1,i_2,j_2} \), we have to consider 3 mutual exclusive cases in Figure 8. The first one shows the case where \( i_1 \circ i_2 \) and \( j_1 \circ j_2 \) and the remaining part will be \( QI_{i_1+1,j_1-1,i_2+1,j_2-1}^{(1)} \). In the second case, \( i_1 \circ i_2 \), but \( j_1 \) and \( j_2 \) do not form a bond. Since \( j_1 \) and \( j_2 \) are both events but do not form a bond, we can form a \( QIac \) structure on the right side the current structure which starts at index \( k_1 \) on \( R \) and index \( k_2 \) on \( S \). Therefore, we will end up with \( QI_{i_1+1,k_1-1,i_2+1,k_2-1}^{(1)} \) in the middle. The third case is symmetric to the second case. For the fourth case, neither \( i_1 \) and \( j_1 \) nor \( i_2 \) and \( j_2 \) can form a bond. By extracting a \( QIac \) structure from the left starting at indices \( k_1 \) on \( R \) and \( k_2 \) on \( S \), we will end up with a \( QI_{aux,i_1,k_1-1,i_2,k_2-1}^{(1)} \) structure on the left.
\[ QIm_{i_1,j_1,i_2,j_2} = \begin{cases} 
 QI_{i_1+1,j_1-1,i_2+1,j_2-1} 	imes iscore(i_1,i_2) \times iscore(j_1,j_2) \\
 + iscore(i_1,i_2) \\
 + \sum_{k_1=i_1+1}^{j_1} \sum_{k_2=i_2+1}^{j_2} QI_{i_1+1,k_1-1,i_2+1,k_2-1} \\
 \times QIac_{k_1,j_1,k_2,j_2} \\
 + \sum_{k_1=i_1+1}^{j_1} \sum_{k_2=i_2+1}^{j_2} QIa_{i_1,k_1-1,i_2,k_2-1} \\
 \times QIac_{k_1,j_1,k_2,j_2} \\
 + iscore(j_1,j_2) \times QIa_{i_1,j_1-1,i_2,j_2-1} & i_1 < j_1 \& i_2 < j_2 \\
 iscore(i_1,i_2) & i_1 = j_1 \& i_2 = j_2 \\
 0 & \text{otherwise} 
\end{cases} \] (15)

\[ \text{Figure 8: For computing } QIm, \text{ since we know the four end points are events, but none of the two end points in one strand can form an arc, we must consider the 3 different cases shown above.} \]

2.4 BPMax Algorithm

Here, we explain BPMax algorithm which is the first implementation (as far as we know) of the base pair counting method explained in [18], with some small tweaks to emphasize the interaction between two RNAs by letting the score of bonds to be different from that of arcs. It also generates normalized interaction score so that it becomes independent from the length of the two interacting sequences which can directly affect the number of pairings otherwise. In the rest of this section, we explain the algorithm implemented.

Here, we use the same notation as before. In addition, for a single strand of nucleotides, \( S_{i,j} \) represents the maximum number of base pairs that we can have on subsequence \([i,j]\). For each strand we need to make such table; to distinguish these tables from one another, we will use superscripts (1) and (2) for \( R \) and \( S \) strand respectively. \( F_{i_1,j_1,i_2,j_2} \) represents the number of maximum pairings (considering both intra- and inter-pairings) on subsequences \([i_1,j_1]\) and \([i_2,j_2]\) from \( R \) and \( S \) respectively.
To compute $S_{i,j}$, since the pairing of two bases that are less than 3 nucleotides apart are not allowed, the value of $S$ for sequences of length less than 5 is considered as 0. Otherwise, the recursion in the second case of equation 16 can be utilized. It considers the case where we have arc $i \bullet j$ and recurs on $[i+1, j-1]$, and also other cases in which $i^{th}$ and $j^{th}$ bases are not paired and the $[i, j]$ is split into two smaller subsequences.

$$S_{i,j} = \max(S_{i+1,j-1} + \text{score}(i, j), \max_{k=1}^{j-1} S_{i,k} + S_{k+1,j})$$ (16)

Now, to compute $F_{i_1,j_1,i_2,j_2}$, as you can see in Figure 9 and (17), conceptually similar to what we had for $S$, 3 cases have to be considered: $i_1 \bullet j_1$ and $F_{i_1+1,j_1-1,i_2,j_2}$, arc $i_2 \bullet j_2$ and $F_{i_1,j_1,i_2+1,j_2-1}$, or none of these arcs and two smaller cases of $F_{i_1,k_1,i_2,k_2}$ and $F_{k_1+1,j_1,k_2+1,j_2}$.

$$F_{i_1,j_1,i_2,j_2} = \begin{cases} 
    S_{i_2,j_2}^{(2)} & j_1 < i_1 \\
    S_{i_1,j_1}^{(1)} & j_2 < i_2 \\
    \text{iscore}(i_1, i_2) & i_1 = j_1 \text{ and } i_2 = j_2 \\
    \max(F_{i_1+1,j_1-1,i_2,j_2} + \text{score}(i_1,j_1), \\ F_{i_1,j_1,i_2+1,j_2-1} + \text{score}(i_2,j_2), \\ \max_{k_1=i_1}^{j_1} \max_{k_2=i_2}^{j_2} (F_{i_1,k_1,i_2,k_2} + F_{k_1+1,j_1,k_2+1,j_2})) & \text{otherwise}
\end{cases}$$ (17)

3 Results

To investigate to what extent the scores of $\text{BPPart}$ and $\text{BPMax}$ are correlated with that of $\text{piRNA}$, we used the RISE database which combines the information about interacting RNAs from multiple experiments. For human dataset, we extracted all the interaction windows for those pairs that have that data and eliminated the ones that contained a window with length less than 15 because they are too short to provide us with an unbiased comparison. Then, the remaining pairs were sorted based on the product of the lengths of the interacting windows. Finally, the first 50500 pairs
of sequences were chosen as our dataset for different experiments and analysis. Figure 10 shows the distribution of the lengths of the sequences present in our dataset and also the product of the lengths of the RNA subsequences in each pair.

First, piRNA was ran on our dataset at 8 different temperatures, 37, 25, 13, 0, −40, −80, −130, and −180 degrees celcius. BPPart and BPMax were ran on the dataset with different weights for each base-pair combination. In general, we want to use the stack energies of the base pairs in the Turner model to tune their weights. We considered a fixed weight of 3 for CG (and GC). Using the experimentally computed stack energies of the Turner model, a minimum and maximum value for the weights of AU and GU were computed. As an example, to compute the maximum weight of AU (and UA), we consider the maximum released energy when AU (or UA) is stacked with another pair; this happens when UA is stacked with CG and 2.4 kcal/mol energy is released. Then, we consider the minimum value of released energy in an stack for CG or GC (for which we assumed a constant weight of 3). Using the experimentally computed stack energies of the Turner model, a minimum and maximum value for the weights of AU and GU were computed. As an example, to compute the maximum weight of AU (and UA), we consider the maximum released energy when AU (or UA) is stacked with another pair; this happens when UA is stacked with CG and 2.4 kcal/mol energy is released. Then, we consider the minimum value of released energy in an stack for CG or GC (for which we assumed a constant weight of 3). Now, given the maximum energy of CG, which is 3.4 kcal/mol, the value of interest is computed as $0.6 \times \frac{3}{4} = 0.529$. However, for the sake of comprehensiveness and exploring the shape of the plots, a much lesser lower-bound of −4.5 and −3 were used for BPPart and BPMax, respectively.

Finally, for all the combinations of weights of AU and GU, in steps of 0.5, the Pearson and Spearman’s Rank correlations with the scores from piRNA at different temperatures were computed. When computing the correlations, we divide the scores from all algorithms by the sum of the lengths of corresponding sequences to normalize them. This normalization mitigates the effect of length on the computed correlations. This step is necessary because, generally, as the length of the pair of sequences increases the scores of all three algorithms increases, and if unnormalized scores are used, a biased higher correlation will be derived. Notice that for the scores of partition functions, piRNA and BPPart, we used the log of the scores; that is why we factor out the sum of the lengths for normalization. If the original values were used, we had to divide the scores by $\exp(L_R + L_S)$. Figures 11 and 12 show the final correlation values. The optimum value of correlation for each temperature is presented in Tables 1 and 2. Figure 13 shows the scatter plots of the scores of BPPart and piRNA at 37°C and −180°C. The red line shows the regression line that is fitted to the points by minimizing the mean squared error (MSE). These plots conform with our expectations and findings that there is a high Pearson and Spearman’s Rank correlation between the two and these correlations are higher for lower temperatures.

As the tables show, there is a high correlation between piRNA and BPPart as well as between piRNA and BPMax, especially when the temperature decreases which is due to a decrease in the role of thermodynamic entropy in the lower temperatures. Also, the Pearson and Spearman’s Rank correlation between BPPart and BPMax were computed with their optimum weights at 37°C and values 0.971 and 0.968 were derived, respectively. It is evident that the correlations for BPPart and BPMax are very high which is expected because of the similar nature of them that is being based on the principle of Minimizing Free Energy (MFE).

Finally, to understand better the behavior of the surface around the higher values in the correlations plots of Figures 11 and 12, the Shannon entropy for the values above a threshold was computed. Figure 15 shows the value of the Shannon entropy for the top 30 values of Pearson and Spearman’s Rank correlation at each temperature.
4 Analysis

The Gibbs free energy

\[ \Delta G = \Delta H - T \Delta S \]

is composed of a term \( \Delta H \) called enthalpy that does not depend on temperature and a term \( T \Delta S \) called entropy that linearly depends on temperature \( T \). Intuitively, enthalpy is the chemical energy that is often released upon formation of chemical bonds such as base pairing. Entropy, on the other hand, captures the size of all possible spatial conformations for a fixed secondary structure. In other words, entropy captures the amount of 3D freedom of the molecule. A base pair brings enthalpy down, hence favorable from enthalpy point of view, and decreases freedom (entropy), hence unfavorable from entropy point of view. These two opposing objectives are combined linearly...
through the temperature coefficient.

In the full thermodynamic model, we consider both terms. In the base pair counting, we consider only a simplistic enthalpy term. Partition function for the full thermodynamic model is

\[ \sum_{s \in S^f} e^{-\Delta G/RT}, \quad (19) \]
in which $R$ is the gas constant. Note that

$$-rac{\Delta G}{T} = -\frac{\Delta H}{T} + \Delta S,$$

(20)

and as $T \to 0$, $-\Delta H/T \to \infty$ and the contribution of $\Delta S$ is diminished to 0 since it is finite. Hence in low temperatures, the effect of entropy becomes negligible, and we expect to see strong correlation between the base pair counting model and full thermodynamic model.
Figure 15: Shannon entropy for the top 30 Pearson (left) and Spearman’s Rank (right) correlation values at different temperatures for \texttt{BPPart} and \texttt{BPMax}.

Figure 14 shows the Pearson correlations between \texttt{BPPart} and \texttt{BPMax} scores and that of \texttt{piRNA} for a fixed combination of weights that results in the highest correlation at 37 (°C). For \texttt{BPPart} the chosen weights are 0.5, 1.0, and 3 for \texttt{AU}, \texttt{GU}, and \texttt{CG}, respectively, while the corresponding weights for \texttt{BPMax} are 1.0, 1.5, and 3.

Perfectly conforming with the theory, we see higher correlations at low temperatures. That somewhat validates our implementations as \texttt{piRNA} was written totally independently about 10 years ago. Moreover, as can be seen in Figures 11 and 12, the surface around the optimum value for higher temperatures becomes flatter. Figure 15, which shows the entropy of the top 30 correlation values, confirms this observation. This means the correlation values are less sensitive to a change in the weights of the base pairs as the temperature increases; this conforms with the theory because at higher temperatures, the thermodynamic entropy increases and the total score of \texttt{piRNA} becomes less sensitive to the energy released by pairings. It is worth mentioning that having less Shannon entropy for the top values at higher temperatures decreases the possibility of having universal optimum values for the weights of the base pairs.

Another noticeable characteristics of the plots 11 and 12 is the region in which the scores of both \texttt{AU} and \texttt{GU} are non-positive. This region for \texttt{BPMax} is flat because when both of these pairs are penalized (or not rewarded when their score is zero), the algorithm simply avoids making such pairs because it is trying to maximize the score. Therefore, it only tries to maximize the number of \texttt{CG} pairs, which is independent of the score (penalty in this case) of the other two types of base pairs. This also applies to the case where one of the base pairs has a non-positive score; in that case, \texttt{BPMax} works independently of the score of that base pair. So, as soon as any of the scores becomes zero or less than zero, \texttt{BPMAX} remains constant along the corresponding axis. For \texttt{BPPart}, however, the story is different because it simply counts all the possible pairings and even if the score of a base pair becomes negative, it does not ignore counting that.

Moreover, \texttt{BPPart} has a higher correlation than \texttt{BPMax} does which comes with the price of a 6 fold increase in computation time. Also, as Figure 15 shows, the Shannon entropy for the top 30 values is less in \texttt{BPMax} and the gap between them grows as temperature decreases; this shows that \texttt{BPPart} has a flatter region around the optimum value and its optimum value is less sensitive
to changes in the weights. Meanwhile, having a curvier surface in **BPMax** which has less entropy increases the possibility of having more stable and universal optimum values for the weights. As mentioned earlier, the running time difference between the two is noticeable: **BPMax** is about 6 fold faster than **BPPart**. Hence, we now have three choices in increasing order of computational cost: **BPMax**, **BPPart**, and **piRNA**. Running time increases about 6 and 225 fold, respectively, from one to the next.

5 Conclusions

We revisited the problems of partition function and structure prediction for interacting RNAs. We simplified the energy model and instead considered only simple weighted base pair counting to obtain **BPPart** for the partition function and **BPMax** for structure prediction. As a result, **BPPart** runs about 225 fold and **BPMax** runs about 1300 fold faster than **piRNA** does. Hence, we gained significant speedup by potentially sacrificing accuracy.

To evaluate practical accuracy of both new algorithms, we computed the Pearson and rank correlations in different temperatures between the results of **BPPart**, **BPMax**, and **piRNA** on 50,500 experimentally characterized RRIs in the RISE database [1]. **BPPart** and **BPMax** results correlate well with those of **piRNA** at low temperatures. At the room and body temperatures, there is considerable correlation and therefore, significant information in the results of **BPPart** and **BPMax**.

We conclude that both **BPPart** and **BPMax** capture a significant portion of the thermodynamic information. Both tools can be used as filtering steps in more sophisticated RRI prediction pipelines. Also, the information captured by **BPPart** and **BPMax** can possibly be complemented with machine learning techniques in the future for more accurate predictions. We now have three choices for RRI thermodynamics in increasing computational cost: **BPMax**, **BPPart**, and **piRNA**. Depending on the application and the trade-off between time and accuracy, one can be chosen.

References

[1] Gong, J. *et al.* RISE: a database of RNA interactome from sequencing experiments. *Nucleic Acids Res.* (2017).

[2] Chitsaz, H., Salari, R., Sahinalp, S. & Backofen, R. A partition function algorithm for interacting nucleic acid strands. *Bioinformatics* **25**, i365–i373 (2009). Also ISMB/ECCB proceedings.

[3] Mathews, D., Sabina, J., Zuker, M. & Turner, D. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.* **288**, 911–940 (1999).

[4] Cao, S. & Chen, S. Predicting RNA pseudoknot folding thermodynamics. *Nucleic Acids Res.* **34**, 2634–2652 (2006).

[5] Dirks, R. M. & Pierce, N. A. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry* **24**, 1664–1677 (2003).

[6] Nussinov, R., Piecznik, G., Grigg, J. R. & Kleitman, D. J. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics* **35**, 68–82 (1978).
[7] Waterman, M. S. & Smith, T. F. RNA secondary structure: A complete mathematical analysis. *Math. Biosc.* **42**, 257–266 (1978).

[8] Zuker, M. & Stiegler, P. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research* **9**, 133–148 (1981).

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

[10] McCaskill, J. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers* **29**, 1105–1119 (1990).

[11] Wenzel, A., Akbaşlı, E. & Gorodkin, J. Research: fast rna–rna interaction search using a simplified nearest-neighbor energy model. *Bioinformatics* **28**, 2738–2746 (2012).

[12] Andronescu, M., Zhang, Z. & Condon, A. Secondary structure prediction of interacting RNA molecules. *J. Mol. Biol.* **345**, 987–1001 (2005).

[13] Bernhart, S. et al. Partition function and base pairing probabilities of RNA heterodimers. *Algorithms Mol Biol* **1**, 3 (2006).

[14] Dirks, R. M., Bois, J. S., Schaeffer, J. M., Winfree, E. & Pierce, N. A. Thermodynamic analysis of interacting nucleic acid strands. *SIAM Review* **49**, 65–88 (2007).

[15] Rehmsmeier, M., Steffen, P., Hochsmann, M. & Giegerich, R. Fast and effective prediction of microRNA/target duplexes. *RNA* **10**, 1507–1517 (2004).

[16] Dimitrov, R. A. & Zuker, M. Prediction of hybridization and melting for double-stranded nucleic acids. *Biophysical Journal* **87**, 215–226 (2004).

[17] Markham, N. & Zuker, M. UNAFold: software for nucleic acid folding and hybridization. *Methods Mol. Biol.* **453**, 3–31 (2008).

[18] Pervouchine, D. IRIS: intermolecular RNA interaction search. *Genome Inform* **15**, 92–101 (2004).

[19] 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* **13**, 267–282 (2006).

[20] Lorenz, R. et al. Viennarna package 2.0. *Algorithms for Molecular Biology* **6**, 26 (2011).

[21] DiChiacchio, L., Sloma, M. F. & Mathews, D. H. Accessfold: predicting rna–rna interactions with consideration for competing self-structure. *Bioinformatics* **32**, 1033–1039 (2015).

[22] Kato, Y., Akutsu, T. & Seki, H. A grammatical approach to RNA-RNA interaction prediction. *Pattern Recognition* **42**, 531–538 (2009).

[23] Huang, F. W. D., Qin, J., Reidys, C. M. & Stadler, P. F. Partition function and base pairing probabilities for RNA-RNA interaction prediction. *Bioinformatics* **25**, 2646–2654 (2009).
[24] Mückstein, U. et al. Thermodynamics of RNA-RNA binding. Bioinformatics 22, 1177–1182 (2006).

[25] Walton, S., Stephanopoulos, G., Yarmush, M. & Roth, C. Thermodynamic and kinetic characterization of antisense oligodeoxynucleotide binding to a structured mRNA. Biophys. J. 82, 366–377 (2002).

[26] Busch, A., Richter, A. S. & Backofen, R. IntaRNA: Efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. Bioinformatics 24, 2849–2856 (2008).

[27] Chitsaz, H., Backofen, R. & Sahinalp, S. biRNA: Fast RNA-RNA binding sites prediction. In Salzberg, S. & Warnow, T. (eds.) Workshop on Algorithms in Bioinformatics (WABI), vol. 5724 of LNBI, 25–36 (Springer-Verlag, Berlin, Heidelberg, 2009).

[28] Krek, A. et al. Combinatorial microrna target predictions. Nature genetics 37, 495 (2005).

[29] Kertesz, M., Iovino, N., Unnerstall, U., Gaul, U. & Segal, E. The role of site accessibility in microrna target recognition. Nature genetics 39, 1278 (2007).

[30] Krüger, J. & Rehmsmeier, M. Rnahybrid: microrna target prediction easy, fast and flexible. Nucleic acids research 34, W451–W454 (2006).

[31] Zhang, Y. miru: an automated plant mirna target prediction server. Nucleic acids research 33, W701–W704 (2005).

[32] Hofacker, I. L. et al. Fast folding and comparison of rna secondary structures. Monatshefte für Chemie/Chemical Monthly 125, 167–188 (1994).

[33] Nam, J.-W. et al. Global analyses of the effect of different cellular contexts on microrna targeting. Molecular cell 53, 1031–1043 (2014).

[34] Betel, D., Koppal, A., Agius, P., Sander, C. & Leslie, C. Comprehensive modeling of microrna targets predicts functional non-conserved and non-canonical sites. Genome biology 11, R90 (2010).

[35] Reczko, M., Maragkakis, M., Alexiou, P., Grosse, I. & Hatzigeorgiou, A. G. Functional microrna targets in protein coding sequences. Bioinformatics 28, 771–776 (2012).

[36] Gaidatzis, D., van Nimwegen, E., Hausser, J. & Zavolan, M. Inference of mirna targets using evolutionary conservation and pathway analysis. BMC bioinformatics 8, 69 (2007).

[37] Riffo-Campos, Á., Riquelme, I. & Brebi-Mieville, P. Tools for sequence-based mirna target prediction: what to choose? International journal of molecular sciences 17, 1987 (2016).

[38] Miranda, K. C. et al. A pattern-based method for the identification of microrna binding sites and their corresponding heteroduplexes. Cell 126, 1203–1217 (2006).

[39] Hsu, J. B.-K. et al. mirtar: an integrated system for identifying mirna-target interactions in human. BMC bioinformatics 12, 300 (2011).
[40] Xu, W., San Lucas, A., Wang, Z. & Liu, Y. Identifying microRNA targets in different gene regions. *BMC bioinformatics* **15**, S4 (2014).

[41] Busch, A., Richter, A. S. & Backofen, R. Intarna: efficient prediction of bacterial srna targets incorporating target site accessibility and seed regions. *Bioinformatics* **24**, 2849–2856 (2008).

[42] Mann, M., Wright, P. R. & Backofen, R. Intarna 2.0: enhanced and customizable prediction of rna–rna interactions. *Nucleic acids research* **45**, W435–W439 (2017).

[43] Kehr, S., Bartschat, S., Stadler, P. F. & Tafer, H. Plexy: efficient target prediction for box c/d snornas. *Bioinformatics* **27**, 279–280 (2010).

[44] Tafer, H., Kehr, S., Hertel, J., Hofacker, I. L. & Stadler, P. F. Rnasnoop: efficient target prediction for h/aca snornas. *Bioinformatics* **26**, 610–616 (2010).

[45] Tjaden, B. Targetrna: a tool for predicting targets of small rna action in bacteria. *Nucleic acids research* **36**, W109–W113 (2008).

[46] Umu, S. U. & Gardner, P. P. A comprehensive benchmark of rna–rna interaction prediction tools for all domains of life. *Bioinformatics* **33**, 988–996 (2017).