**A partition function algorithm for interacting nucleic acid strands**

Hamidreza Chitsaz¹,†, Raheleh Salari¹,†, S. Cenk Sahinalp¹,‡ and Rolf Backofen²,§

¹Lab for Computational Biology, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada and ²Department of Bioinformatics, Institute of Computer Science, Albert-Ludwigs-Universität, Freiburg, Germany

**ABSTRACT**

Recent interests, such as RNA interference and antisense RNA regulation, strongly motivate the problem of predicting whether two nucleic acid strands interact.

**Motivation:** Regulatory non-coding RNAs (ncRNAs) such as microRNAs play an important role in gene regulation. Studies on both prokaryotic and eukaryotic cells show that such ncRNAs usually bind to their target mRNA to regulate the translation of corresponding genes. The specificity of these interactions depends on the stability of intermolecular and intramolecular base pairing. While methods like deep sequencing allow to discover an ever increasing set of ncRNAs, there are no high-throughput methods available to detect their associated targets. Hence, there is an increasing need for precise computational target prediction. In order to predict base-pairing probability of any two bases in interacting nucleic acids, it is necessary to compute the interaction partition function over the whole ensemble. The partition function is a scalar value from which various thermodynamic quantities can be derived. For example, the equilibrium concentration of each complex nucleic acid species and also the melting temperature of interacting nucleic acids can be calculated based on the partition function of the complex.

**Results:** We present a model for analyzing the thermodynamics of two interacting nucleic acid strands considering the most general type of interactions studied in the literature. We also present a corresponding dynamic programming algorithm that computes the partition function over (almost) all physically possible joint secondary structures formed by two interacting nucleic acids in O(n³) time. We verify the predictive power of our algorithm by computing (i) the melting temperature for interacting RNA pairs studied in the literature and (ii) the equilibrium concentration for several variants of the OxyS–fhlA complex. In both experiments, our algorithm shows high accuracy and outperforms competitors.

**Availability:** Software and web server is available at http://compbio.cs.sfu.ca/taverna/pirna/

**Contact:** cenk@cs.sfu.ca; backofen@informatik.uni-freiburg.de

**Supplementary information:** Supplementary data are available at Bioinformatics online.

**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. Regulatory non-coding RNAs (ncRNAs) such as microRNAs play an important role in gene regulation. Studies on both prokaryotic and eukaryotic cells show that such ncRNAs usually bind to their target mRNA to regulate the translation of corresponding genes. The specificity of these interactions depends on the stability of intermolecular and intramolecular base pairing. While methods like deep sequencing allow to discover an ever increasing set of ncRNAs, there are no high-throughput methods available to detect their associated targets. Hence, there is an increasing need for precise computational target prediction. In order to predict base-pairing probability of any two bases in interacting nucleic acids, it is necessary to compute the interaction partition function over the whole ensemble. The partition function is a scalar value from which various thermodynamic quantities can be derived. For example, the equilibrium concentration of each complex nucleic acid species and also the melting temperature of interacting nucleic acids can be calculated based on the partition function of the complex.

Motivation:

Regulatory non-coding RNAs (ncRNAs) such as microRNAs play an important role in gene regulation. Studies on both prokaryotic and eukaryotic cells show that such ncRNAs usually bind to their target mRNA to regulate the translation of corresponding genes. The specificity of these interactions depends on the stability of intermolecular and intramolecular base pairing. While methods like deep sequencing allow to discover an ever increasing set of ncRNAs, there are no high-throughput methods available to detect their associated targets. Hence, there is an increasing need for precise computational target prediction. In order to predict base-pairing probability of any two bases in interacting nucleic acids, it is necessary to compute the interaction partition function over the whole ensemble. The partition function is a scalar value from which various thermodynamic quantities can be derived. For example, the equilibrium concentration of each complex nucleic acid species and also the melting temperature of interacting nucleic acids can be calculated based on the partition function of the complex.

Results:

We present a model for analyzing the thermodynamics of two interacting nucleic acid strands considering the most general type of interactions studied in the literature. We also present a corresponding dynamic programming algorithm that computes the partition function over (almost) all physically possible joint secondary structures formed by two interacting nucleic acids in O(n³) time. We verify the predictive power of our algorithm by computing (i) the melting temperature for interacting RNA pairs studied in the literature and (ii) the equilibrium concentration for several variants of the OxyS–fhlA complex. In both experiments, our algorithm shows high accuracy and outperforms competitors.

Availability:

Software and web server is available at http://compbio.cs.sfu.ca/taverna/pirna/

Contact: cenk@cs.sfu.ca; backofen@informatik.uni-freiburg.de

Supplementary information: Supplementary data are available at Bioinformatics online.

© 2009 The Author(s)

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

© 2009 The Author(s)

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/2.0/uk/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

secondary structure: a partition function algorithm should guarantee that every joint structure is considered exactly once.

In this article, we present an $O(n^5)$ time algorithm to compute the partition function over the type of interactions that Alkan et al. (2006) considered. We extend the standard energy model for a single RNA model to an energy model for the joint secondary structure of interacting strands by considering new types of (joint) structural components. We verify our algorithm (and the associated software we developed) by computing the melting temperature for RNA pairs available (Diamond et al., 2001; Mathews and Turner, 2002; Xia et al., 1998) and the equilibrium concentration for OxyS–IHa complexes for wild-type IHa and four other mutants reported in the literature (Argaman and Altuvia, 2000). In both experiments, our algorithm shows high accuracy and outperforms existing alternatives.

1.1 Related work
During the last few decades, several computational methods emerged to study the secondary structure thermodynamics of a single nucleic acid strand. Nearest neighbor thermodynamic model has become the standard energy model for a nucleic acid secondary structure (Mathews et al., 1999). The standard energy model is based on the assumption that stacking base pairs and loop entropies contribute additively to the free energy of a nucleic acid secondary structure. More recently, the standard energy model has been extended for pseudoknots (Cao and Chen, 2006; Dirks and Pierce, 2003). Based on additivity of the energy, efficient dynamic programming algorithms for predicting the minimum free energy secondary structure (Nussinov et al., 1978; Rivas and Eddy, 1999; Waterman and Smith, 1978; Zuker and Stiegler, 1981) and computing the partition function of a single strand (Dirks and Pierce, 2003; McCaskill, 1990) have been developed.

Some previous attempts to analyze the thermodynamics of multiple interacting strands concatenate input sequences in some order and consider them as a single strand. For example, pairfold (Andronescu et al., 2005) and RNAfold from Vienna package (Bernhart et al., 2006) concatenate the two input sequences into a single strand and predict its minimum free-energy structure. Dirks et al. (2007) present a method, as a part of NUPACK, that concatenates the input sequences in some order, carefully considering symmetry and sequence multiplicities, and computes the partition function for the whole ensemble of complex species. However, concatenating the sequences is not accurate at all as even if pseudoknots are considered, some useful interactions are excluded while many physically impossible interactions are included (e.g. physically impossible crossing interactions).

Alternatively, several methods avoid internal base-pairing in either strand, and compute the minimum free energy secondary structure for their hybridization under this constraint [RNAhybrid (Rehmenson et al., 2004), RNAfold (Dmitrov and Zuker, 2004; Markham and Zuker, 2008), and RNAduplex from Vienna package (Bernhart et al., 2006)]. These approaches naturally work only for simple cases involving typically very short strands.

A third set of methods predict the secondary structure of each individual RNA independently, and predict the (most likely)

hybridization between the unpaired regions of the two molecules. More sophisticated alternatives view interaction as a multi-step process (Busch et al., 2008; Mückstein et al., 2006; Walton et al., 2002): (i) unfolding of the two molecules to expose bases needed for hybridization, (ii) the hybridization at the binding site and (iii) restructuring of the complex to a new minimum free-energy conformation.

In addition to the above approaches, a number of studies aimed to compute the minimum total energy joint structure between two interacting strands under energy models with growing complexity. For instance, Pervouchine devised a dynamic programming algorithm to maximize the number of base pairs among interacting strands (Pervouchine, 2004). A follow-up work by Kato et al. (2009) proposed a grammar-based approach to RNA–RNA interaction prediction. More generally, Alkan et al. (2006) studied the joint secondary structure prediction problem under three different models: (i) base pair counting, (ii) stacked pair energy model and (iii) loop energy model. Alkan et al. (2006) proved that the general RNA–RNA interaction prediction under all three energy models is an NP-hard problem. Therefore, they suggested some natural constraints on the topology of possible joint secondary structures that are satisfied by all examples of complex RNA–RNA interactions in the literature. The resulting algorithms compute the minimum free energy secondary structure among all possible joint secondary structures that do not contain (internal) pseudoknots, crossing interactions (i.e. external pseudoknots) and zigzags (see Section 2.1 for the exact definition).

2 METHODS

2.1 Preliminaries
Throughout this article, 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. Each nucleotide is paired with at most one nucleotide in the same or the other strand. We refer to the i-th nucleotide in R and S by $i_R$ and $i_S$, respectively. The subsequence from the i-th nucleotide in 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 $\bullet$. An intramolecular base pair between the nucleotides $i_R$ and $i_S$ is called a bond and denoted by a circle $\bigcirc$. An arc $i_R\bullet j_R$ covers a bond $i_R\bigcirc j_R$ if $i_R < j_R$ and $i_R \leq j_S$. We call $i_R\bullet j_R$ an interaction arc if there is a bond $i_R\bigcirc j_R$ covered by $i_R\bullet j_R$. A kissing arc $i_R\bullet j_R$ is an interaction arc that directly covers a bond. More precisely, we call $i_R\bullet j_R$ a kissing arc if it covers a bond $i_R\bigcirc j_R$ such that if $i_R\bigcirc j_R$ covers the same bond $i_R\bigcirc j_R$, then $i_R \leq i_S$ and $j_R \leq j_S$. A subsequence $[i_R,j_R]$ contains a direct bond, $i_R\bigcirc j_R$, if $i_R \leq i_S \leq j_R$ and no arc within $[i_R,j_R]$ covers $i_R\bigcirc j_R$. Assuming $i_R < j_R$, two bonds $i_R\bigcirc j_R$ and $i_S\bigcirc j_S$ are called crossing bonds if $i_S < j_R$. An interaction arc $i_R\bullet j_R$ in a strand subsumes a subsequence $[i_R,j_R]$ in the other strand if for all bonds $i_R\bigcirc j_R$, if $i_S \leq i_R \leq j_S$ then $i_R < j_R$. Two interaction arcs are equivalent if they subsume one another. Two interaction arcs $i_R\bullet j_R$ and $i_S\bullet j_S$ are part of a zigzag if neither $i_R\bullet j_R$ nor $i_S\bullet j_S$ subsumes $[i_R,j_R]$.

In this article, 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.

2.2 Interaction energy model
A unpseudoknotted secondary structure $\s$ of a single nucleic acid, in the standard energy model (Mathews et al., 1999), is decomposed into loops,
These components and their free energy contributions are the standard case, an interaction secondary structure model by defining those new kinds of interaction components. Similar to (remember we do not allow pseudoknots, crossing bonds and zigzags in this total free energy into intramolecular loops and the new interaction components such that the structure into the interaction components (Argaman and Altuvia, 2000).

In an interaction, secondary structure of two strands under our assumptions consisting of a series of bonds, \( \sum_{i=1}^{\ell} A_i \) and zigzags. We call such a component hybrid. Given a pair of nucleic acid strands \( R \) and \( S \), and a temperature \( T \), compute the partition function, \( Q(T) \), over \( S^R \) the set of all possible single or duplex secondary structures that do not contain pseudoknots, crossing bonds and zigzags.

### Interaction partition function (IPF) problem

Given a pair of nucleic acid strands \( R \) and \( S \), a temperature \( T \), compute the partition function, \( Q(T) \), over \( S^R \) the set of all possible single or duplex secondary structures that do not contain pseudoknots, crossing bonds and zigzags.

**Input:** nucleic acid strands \( R \) and \( S \).

**Output:**

\[
Q(T) = \sum_{S^R} e^{-G_{S^R}/RT}
\]

We give a recursive algorithm, called Partition function for Interacting Nucleic Acids (piRNA), for the IPF problem. In all of our recursions, the considered cases are disjoint. This fact shows that every possible secondary structure is reached by exactly one trajectory in the recursion process. Our algorithm guarantees to consider all possible secondary structures exactly.
once. Since our algorithm covers so many cases, we do not include all the details here. A comprehensive description of our algorithm is available in our Supplementary Material.

We present our algorithm using recursion diagrams (Dirks and Pierce, 2003; Rivas and Eddy, 1999). Our algorithm computes two types of recursive quantities: (i) the partition function of a subsequence \([i, j]\) in one strand, and (ii) the joint partition function of subsequences \([i_a, j_a]\) and \([i_b, j_b]\). A region is the domain over which a partition function is computed. Terminal bases are the boundaries of a region. For the first type, region is \([i, j]\) with \(i\) and \(j\) terminal bases. For the second type, region is \([i_a, j_a] \times [i_b, j_b]\) with \(i_a, j_a, i_b\) and \(j_b\) terminal bases. The length pair of region \([i_a, j_a] \times [i_b, j_b]\) is \((i_a - i_a + 1, i_b - i_b + 1)\). Our algorithm starts with \((i_a = 1, j_a = 1)\) and considers all length pairs incrementally up to \((i_a = L_a, j_a = L_a)\). For a fixed length pair \((i_a, j_a)\), recursive quantities for all the regions \([i_a, i_b + 1 - j_b] \times [i_a, i_a + 1 - j_a]\) are computed.

For computing the partition function of a subsequence in one strand we use McCaskill’s (1990) algorithm. McCaskill’s algorithm is shown in Figure 4, in which \(Q_{ij}\) is the partition function for the subsequence \([i, j]\). Throughout this article, 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 that may be paired or unpaired. Letter(s) in 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 their energy to the partition function as defined by the energy model. Green and red regions have the same recursion level and contribute their energy to the partition function as indicated by the arrows (see Supplementary Material for a full definition).

In the following, we present all cases of \(Q_{ij}^{a, b}\) which is the interaction partition function for the region \([i_a, j_a] \times [i_b, j_b]\). 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. For the interaction partition functions, gray regions indicate a reference to the partition functions for the single sequences. Figure 5 shows the cases of \(Q_{ij}\). (i) there is no bond between the two subsequences, (ii) the leftmost bond is a direct bond in both subsequences and (iii) the leftmost bond is covered by an arc in at least one subsequence.

\begin{align}
Q_{ij} = & 1 + \sum_{i_a + 1 \geq k \geq j} Q_{i_a k} Q_{k j + 1}.
\end{align}

The second line shows the cases of \(Q_{ij}^{a, b}\) which is the partition function for the subsequence \([i, j]\) assuming \(i\) and \(j\) are base paired. The arc \(\ast\) can close different substructures: hairpin, interior or multiloop. The energy contribution of each substructure is calculated based on the standard thermodynamics energy model. The third line shows cases of \(Q_{ij}^{a, b}\) which is the partition function for the subsequence \([i, j]\) assuming the region constitutes at least one arc. A region tagged by \(b\) and colored by green is contained in a multiloop and the penalty of multiloop should be applied to it. Explicit equations for \(Q_{ij}\) and \(Q_{ij}^{a, b}\) are given in the Supplementary Material.

\[Q_{ij}^{a, b} = Q_{i_a k} Q_{k j + 1} + \sum_{i_a + 1 \geq k \geq j} Q_{i_a k} Q_{k j + 1} + \sum_{i_a + 1 \geq k \geq j} Q_{i_a k} Q_{k j + 1} Q_{i_a k} Q_{k j + 1}\]

Fig. 4. McCaskill’s algorithm: recursion for \(Q_{ij}\), the partition function for the subsequence \([i, j]\). Above, \(Q_{ij}^{a, b}\) is the partition function for the subsequence \([i_a, j_a]\) assuming \(i\) and \(j\) are base paired, and \(Q_{ij}^{b, b}\) is the partition function for the subsequence \([i_b, j_b]\) assuming there is at least one arc in the region.

Fig. 5. Cases of the interaction partition function \(Q_{ij}^{a, b}\). The first case constitutes no bonds. In the second case, the leftmost bond is a direct bond on both subsequences. In the third case, the leftmost bond is covered by an interaction arc in at least one subsequence.

Fig. 6. Recursion for \(Q_{ij}^{a, b}\) assuming \(i_a \geq j_b\) is a bond. We show a version of the recursion that contains two split points in each sequence for simplicity reasons. However, this would increase the complexity and can easily be resolved by introducing two additional matrices \(Q_{i_a k}^{a, a}\) and \(Q_{i_a k}^{b, b}\) for the region \([i_a, k] \times [k, j_b]\) as indicated by the arrows (see Supplementary Material for a full definition).

In the following, we present all cases of \(Q_{ij}^{a, b}\) which is the interaction partition function for the region \([i_a, j_a] \times [i_b, j_b]\). 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. For the interaction partition functions, gray regions indicate a reference to the partition functions for the single sequences. Figure 5 shows the cases of \(Q_{ij}\). (i) there is no bond between the two subsequences, (ii) the leftmost bond is a direct bond in both subsequences and (iii) the leftmost bond is covered by an arc in at least one subsequence.

\[Q_{ij}^{a, b} = Q_{i_a k} Q_{k j + 1} + \sum_{i_a + 1 \geq k \geq j} Q_{i_a k} Q_{k j + 1} Q_{i_a k} Q_{k j + 1}\]

\[Q_{ij}^{a, b} = Q_{i_a k} Q_{k j + 1} + \sum_{i_a + 1 \geq k \geq j} Q_{i_a k} Q_{k j + 1} Q_{i_a k} Q_{k j + 1}\]

\[Q_{ij}^{a, b} = Q_{i_a k} Q_{k j + 1} + \sum_{i_a + 1 \geq k \geq j} Q_{i_a k} Q_{k j + 1} Q_{i_a k} Q_{k j + 1}\]
all cases, is the end point of an interaction arc: (i) $iR$.

Cases of Fig. 7. with a bond, then the interaction arc subsumes the other subsequence. If both

are end points of interaction arcs, then one of the arcs subsumes the

-leftmost of which is $iR$.

Fig. 7 shows the cases of $Q^0_{iR}$, for which we assume at least one of $iR$ and $jS$ is the end point of an interaction arc.

Figure 6 shows the recursion for $Q^0_{iR}$, the interaction partition function for the region $\{iR, jS\}$ assuming $iR$ is a bond. Since we have a $\beta i$ penalty for each hybrid component, the recursion for $Q^0$ has to determine whether the region contains one or several hybrid components. In all cases, $Q^0$ contains the full hybrid component containing the bond $iR/jS$ (see Fig. 7 for $Q^0$ recursion). The first possibility reflects the case where we have only one hybrid component. In the other cases, we have at least two hybrid components. The subsequent intermolecular bond starts a new hybrid component iff (i) it is not direct in at least one subsequence, i.e. it is covered by an arc in the associated regions (Case 2 of the $Q^0$ recursion), or (ii) there is at least one arc between the two subsequent intermolecular bonds (Cases 3 and 4 of the $Q^0$ recursion). Using the additional matrices $Q^{0\phi}$ and $Q^{0\phi}$, we get

$$Q^0_{iR}(iR, jS) = Q^0_{iR}(iR, jS) + \sum_{iR, jS, \leq 2} Q^0_{iR}(iR, jS) Q^0_{iR}(iR, jS)$$

$$+ \sum_{iR, jS, \leq 2} Q^0_{iR}(iR, jS) Q^0_{iR}(iR, jS)$$

where

Figure 7 shows the cases of $Q^0$: (i) there is no bond other than $iR/jS$ and $iR/jS$, the leftmost of which is $iR$.

Figure 8 shows the cases of $Q^0_{iR}$, for which at least one of $iR$ and $jS$ is the end point of an interaction arc: (i) $iR$ is a terminal subsequence of $iS$, $jR$, $iS$ $jS$, and $jS$ is not base paired with $jS$, and (ii) $iR$ is base paired with $jS$, and $iR$ is the end point of an interaction arc while the other one is the end point of a bond, then the interaction arc subsumes the other subsequence. If both $iR$ and $jS$ are end points of interaction arcs, then one of the arcs subsumes the other one or they are equivalent. Therefore,

$$Q^0_{iR}(iR, jS) = \sum_{iR, jS, \leq 2} Q^0_{iR}(iR, jS) Q^0_{iR}(iR, jS)$$

$$Q^0_{iR}(iR, jS) + \sum_{iR, jS, \leq 2} Q^0_{iR}(iR, jS) Q^0_{iR}(iR, jS)$$

in which $Q^0_{iR}(iR, jS)$ is the interaction partition function of $\{iR, jS\}$ assuming $iR$ is an interaction arc that subsumes $jS$. Then $Q^0_{iR}(iR, jS)$ is the symmetric counterpart of $Q^0_{iR}$ and $Q^0_{iR}(iR, jS)$ is the interaction partition

function of $\{iR, jS\}$, assuming $iR$ and $jS$ are equivalent interaction arcs.

For $Q^0$, it does not make any difference which one of the covering arcs, $iR$ or $jS$, is extracted first. We first extract the covering arc from $S$ (Fig. 9). Extracting the covering arc, the remaining subsequence of $S$ contains either at least one direct bond, in which case kissing loop penalty should be applied, or multiple interaction arcs, in which case multiloop penalty should be applied. Hence, Figure 9 is appropriately colored by green and red to remind the type of penalty.

Note that $Q^0_{iR}$ and $Q^0_{jS}$ are the partition functions for the $\{iR\}$ and $\{jS\}$ subsequence of $S$ which have no more spanning interaction arc in the region, and (ii) there is at least another innermost spanning interaction arc $iS, jS$. In both groups, there could be some additional intramolecular structure in the region. The quantity $Q^0_{iR}$ is the partition function for the subsequence $\{iR\}$, the gap in Fig. 10 is appropriately colored by green and red to remind the type of penalty.

The union of the cases of $Q^0$ and $Q^0$ comprises the cases of $Q^0$. Similar to the cases of $Q^0$, we extract the covering arc from $Q^0$ to obtain $Q^0$, $Q^0$, $Q^0$, and $Q^0$, where $k$ stands for kissing (or equivalently containing a direct bond) and $m$ for multiple interaction arcs. Note that all four terminal bases of the region of these four quantities are paired, i.e. each terminal base is either the end point of a bond or of an interaction arc. These four quantities have complicated cases. Due to lack of space, we explain them in our Supplementary Material.

3 EXPERIMENTS

Here, we report our implementation of the algorithm and two types of experiments we performed to test the predictive power of our algorithm.

(1) Predicting the melting temperature of RNA duplexes is an important application of the partition function for interacting nucleic acid pairs (Dimitrov and Zuker, 2004); our first
experiment tests how accurately our algorithm predicts the melting temperature of RNA pairs collected from several sources in the literature with respect to the accuracy of available alternatives. RNAcofold from Vienna package v1.7.2 (Bernhart et al., 2006) and UNAFold v3.6 which is a new version of former mfold (Markham and Zuker, 2008). We remind the reader that RNAcofold concatenates the two RNA strands and computes the partition function for the resulting single strand. Therefore, it does not consider many cases that our algorithm considers. UNAFold v3.6, on the other hand, simplifies the problem by forbidding intramolecular base pairing. It computes the partition function of the two strands over just hybridization structures. As can be expected, our algorithm consistently outperforms the alternatives in all three datasets:

(1) A novel experiment (which, to our knowledge has not been performed successfully by any other program to date) uses our algorithm to predict the equilibrium concentration of an RNA–RNA complex, in particular the OxyS–fhlA interaction (Argaman and Altuvia, 2000). We successfully predicted the equilibrium concentrations for OxyS with wild-type fhlA and four other fhlA mutants.

Note that the parameters used by our program in the above experiments have been manually optimized as computational learning methods for fine tuning the parameters require prohibitive computational resources. It may be possible to improve the accuracy of our program through a better selection of parameters.

3.1 Implementation

We remind the reader that the time and space complexity of our algorithm are $O(n^6)$ and $O(n^4)$, respectively, here $n=\max(L_1,L_2)$ is the maximum length of the two input strands. We implemented our algorithm in C++, and used the energy functions and energy parameters of UNAFold v3.6 for a single strand (Markham and Zuker, 2008). The parameters used by our program for our own interaction energy model are given in the next section. As per Section 2.2, we use a different $\beta_1$ penalty and $\sigma$ for a hybrid component that is covered by a kissing loop. The parameters for a hybrid component that is not covered by a kissing loop is denoted by $\beta_1'$ and $\sigma'$. We add an Adenine-Uracil base pair (AU) penalty to the energy of a hybrid component per each terminal AU base pair; this penalty is motivated by Xia et al. (1998). Similar to RNAhybrid, the interior loops in a hybrid component are restricted to a constant maximum length, in either sequence, which is set to 15 in this work.

Since our algorithm considers many more possible secondary structures in comparison to alternative methods, our program has a higher running time. Fortunately, our algorithm can be easily parallelized as the dynamic programming tables computed by our program on subsequence pairs depend only on their (proper) subregions. We parallelized our program using OpenMP 3.0. Our experiments were performed on a large-scale shared memory parallel platform with 64PPC 1.9 GHz processors with 256 GB RAM. We ran our program for strands of length between 5 nt to 120 nt. The running time of our program for short strands (<~20 nt) was ~1 min.; for longer strands (~120 nt) it was ~10 h.

3.2 Datasets

The first dataset that we used for predicting melting temperature contains all nine different RNA pairs reported in Table 3 of Xiang et al. (1998). It contains almost complementary 5- to 7- nt RNA pairs that were designed to optimize the thermodynamic parameters for terminal base pairs. Their melting temperatures vary from 29.8°C to 53.7°C.

The second dataset that we used for computing melting temperature contains all 12 different RNA pairs reported in Table 1 of Diamond et al. (2001). These RNA pairs are designed to optimize the thermodynamic parameters for three-way multi loops. In each pair of this dataset, the first RNA has ~20 nt and the second one has ~10 nt. The experimental melting temperatures were determined from heat absorbance measurements by two different methods that are explained as ‘Method 3’ and ‘Method 4’ in Puglisi and Tinoco (1989). Although these pairs are very similar, the average difference of the two methods for this dataset is 2.49°C. This suggests that there may exist RNA pairs with exceptional features in this set.

The third dataset that we used for computing melting temperature contains all 62 different RNA pairs reported in Tables 3 and 4 of Mathews and Turner (2002). These pairs are designed to optimize the thermodynamic parameters for three- and four-way multi loops. In each pair of this dataset, the first RNA has 22–40 nt and the second one has 10–14 nt. Again, the experimental melting temperatures were determined by two different methods. This dataset is large enough with longer sequences, and the average difference of the two methods for this dataset is 0.7°C, smaller than that for the second dataset. Moreover, the variance and maximum of the difference is smaller than those of the second dataset. Overall, this dataset is more reliable than the previous one. These three datasets are all we were able to collect from the literature.

3.3 Melting temperature

As mentioned before, predicting the melting temperature of RNA duplexes is one of the most important applications of the partition function for interacting nucleic acid pairs (Dimitrov and Zuker, 2004). Table 1 shows the melting temperatures computed by our program, RNAcofold, and UNAFold v3.6 for the first dataset. In this set, the strands are short, and as we expected, our algorithm is highly accurate with only 1.48°C absolute difference from experimental values on average. It can be seen that RNAcofold and UNAFold perform relatively poorly, and their predicted melting temperatures differ from the experimental values by about 9°C on average.

Table 2 shows the melting temperatures predicted by the three programs for the second dataset. Each pair is referred to by an identifier (A, B, …, L). Please refer to our Supplementary Material or Diamond et al. (2001) to see the exact sequences of each pair. As mentioned before, the experimental melting temperatures were determined from heat absorbance measurements by two different
Table 1. Experimental and predicted melting temperatures for the first dataset [see Section 3.2 and Xia et al. (1998)].

<table>
<thead>
<tr>
<th>Pairs</th>
<th>Experiment</th>
<th>piRNA</th>
<th>RNAcofold</th>
<th>UNAFold</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACGA/ACGU</td>
<td>29.8</td>
<td>29.41</td>
<td>42.64</td>
<td>46.14</td>
</tr>
<tr>
<td>GACACA/ACGCA</td>
<td>36.07</td>
<td>46.61</td>
<td>43.91</td>
<td></td>
</tr>
<tr>
<td>ACAGA/ACGCU</td>
<td>30.2</td>
<td>30.38</td>
<td>42.68</td>
<td>45.15</td>
</tr>
<tr>
<td>GCUGA/ACGAC</td>
<td>37.2</td>
<td>36.88</td>
<td>47.75</td>
<td>44.71</td>
</tr>
<tr>
<td>AGCGA/UCGCU</td>
<td>48.2</td>
<td>44.91</td>
<td>56.8</td>
<td>57.59</td>
</tr>
<tr>
<td>GUCACUA/GAGACU</td>
<td>51.1</td>
<td>49.4</td>
<td>58.44</td>
<td>55.91</td>
</tr>
<tr>
<td>AGCGA/UCGCU</td>
<td>45.7</td>
<td>45.47</td>
<td>56.4</td>
<td>56.68</td>
</tr>
<tr>
<td>GUCACUA/GAGACU</td>
<td>52</td>
<td>49.96</td>
<td>59.11</td>
<td>56.25</td>
</tr>
<tr>
<td>GA2GCA2G/CAGCU</td>
<td>53.7</td>
<td>49.07</td>
<td>59.07</td>
<td>56.00</td>
</tr>
</tbody>
</table>

Avg. error 1.48
Spearman rank corr. 0.97

All values, except Spearman's rank correlation, are in degree centigrade. Bold entries are the most accurate predictions. In other words, they have the least difference from experimental measurements.

Table 2. Experimental and predicted melting temperatures for the set of RNA pairs reported in Diamond et al. (2001).

<table>
<thead>
<tr>
<th>Pairs</th>
<th>Experiment</th>
<th>piRNA</th>
<th>RNAcofold</th>
<th>UNAFold</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>28.7</td>
<td>30.3</td>
<td>32.44</td>
<td>50.99</td>
</tr>
<tr>
<td>B</td>
<td>19</td>
<td>20.5</td>
<td>31.05</td>
<td>52.55</td>
</tr>
<tr>
<td>C</td>
<td>33.6</td>
<td>33.6</td>
<td>32.94</td>
<td>51.11</td>
</tr>
<tr>
<td>D</td>
<td>33.9</td>
<td>33.6</td>
<td>32.43</td>
<td>51.02</td>
</tr>
<tr>
<td>E</td>
<td>23</td>
<td>24.4</td>
<td>31.06</td>
<td>52.48</td>
</tr>
<tr>
<td>F</td>
<td>34.9</td>
<td>36.9</td>
<td>33.28</td>
<td>54.7</td>
</tr>
<tr>
<td>G</td>
<td>32.4</td>
<td>33.6</td>
<td>32.76</td>
<td>49.76</td>
</tr>
<tr>
<td>H</td>
<td>16.1</td>
<td>18.9</td>
<td>36.41</td>
<td>57.72</td>
</tr>
<tr>
<td>I</td>
<td>29</td>
<td>32.2</td>
<td>32.32</td>
<td>50.99</td>
</tr>
<tr>
<td>J</td>
<td>32.3</td>
<td>37.01</td>
<td>56.52</td>
<td>28.8</td>
</tr>
<tr>
<td>K</td>
<td>23.4</td>
<td>30.7</td>
<td>31.45</td>
<td>49.36</td>
</tr>
<tr>
<td>L</td>
<td>33.5</td>
<td>35.4</td>
<td>32.64</td>
<td>50.51</td>
</tr>
</tbody>
</table>

Avg. difference 8.05
Spearman rank corr. 0.97

Each pair is referred to by an identifier (A, B, ... L). Please refer to our Supplementary Material or Puglisi et al. (1998) to see the exact sequences of each pair. All values, except Spearman's rank correlation, are in degree centigrade.

Methods which are explained as 'Method 3' and 'Method 4' in Puglisi and Tinoco (1989). We refer to the melting temperature values computed by 'Method 3' and 'Method 4' by $T_3$ and $T_4$, respectively. RNAcofold accuracy obviously dropped in this case, whereas UNAFold accuracy did not change much in comparison to the results for the first dataset. The accuracy of our method has also dropped a bit, which may be because of some RNA pairs with exceptional features.

Table 3. Experimental and predicted melting temperatures for the set of RNA pairs reported in Mathews and Turner (2002).

<table>
<thead>
<tr>
<th>Pairs</th>
<th>Experiment</th>
<th>piRNA</th>
<th>RNAcofold</th>
<th>UNAFold</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>54.4</td>
<td>46</td>
<td>56.81</td>
<td>37</td>
</tr>
<tr>
<td>B</td>
<td>51.8</td>
<td>52.2</td>
<td>56.84</td>
<td>37</td>
</tr>
<tr>
<td>C</td>
<td>55.9</td>
<td>56</td>
<td>56.86</td>
<td>37</td>
</tr>
<tr>
<td>D</td>
<td>58.4</td>
<td>57.3</td>
<td>56.85</td>
<td>37</td>
</tr>
<tr>
<td>E</td>
<td>57.3</td>
<td>56.9</td>
<td>56.84</td>
<td>37</td>
</tr>
<tr>
<td>F</td>
<td>56.7</td>
<td>57.1</td>
<td>56.85</td>
<td>37</td>
</tr>
<tr>
<td>G</td>
<td>51.2</td>
<td>49.3</td>
<td>56.94</td>
<td>37.1</td>
</tr>
<tr>
<td>H</td>
<td>55.2</td>
<td>54.7</td>
<td>56.96</td>
<td>41</td>
</tr>
<tr>
<td>I</td>
<td>56.1</td>
<td>55.6</td>
<td>56.98</td>
<td>41</td>
</tr>
<tr>
<td>J</td>
<td>54.8</td>
<td>54.3</td>
<td>56.98</td>
<td>37.06</td>
</tr>
<tr>
<td>K</td>
<td>51.3</td>
<td>54.8</td>
<td>56.98</td>
<td>37</td>
</tr>
<tr>
<td>L</td>
<td>53.2</td>
<td>52</td>
<td>57.17</td>
<td>50.17</td>
</tr>
<tr>
<td>M</td>
<td>57</td>
<td>56.4</td>
<td>57.83</td>
<td>48.01</td>
</tr>
<tr>
<td>N</td>
<td>55.6</td>
<td>55.4</td>
<td>57.85</td>
<td>41.04</td>
</tr>
<tr>
<td>O</td>
<td>55.4</td>
<td>57.8</td>
<td>57.83</td>
<td>41.84</td>
</tr>
<tr>
<td>P</td>
<td>55.3</td>
<td>54.5</td>
<td>57.3</td>
<td>52.17</td>
</tr>
<tr>
<td>Q</td>
<td>51.4</td>
<td>53.9</td>
<td>57.85</td>
<td>48.01</td>
</tr>
<tr>
<td>R</td>
<td>56.6</td>
<td>56.6</td>
<td>57.18</td>
<td>47.01</td>
</tr>
<tr>
<td>S</td>
<td>58.7</td>
<td>58.9</td>
<td>57.18</td>
<td>44.1</td>
</tr>
<tr>
<td>T</td>
<td>58.8</td>
<td>57.2</td>
<td>57.17</td>
<td>44.1</td>
</tr>
<tr>
<td>U</td>
<td>54.6</td>
<td>57.5</td>
<td>57.15</td>
<td>44.1</td>
</tr>
<tr>
<td>V</td>
<td>57.2</td>
<td>56.9</td>
<td>57.48</td>
<td>43</td>
</tr>
<tr>
<td>W</td>
<td>57.9</td>
<td>57.9</td>
<td>57.17</td>
<td>44.1</td>
</tr>
<tr>
<td>X</td>
<td>56.5</td>
<td>57.5</td>
<td>57.14</td>
<td>44.1</td>
</tr>
<tr>
<td>Y</td>
<td>57.2</td>
<td>56.9</td>
<td>57.48</td>
<td>43</td>
</tr>
<tr>
<td>Z</td>
<td>57.9</td>
<td>57.9</td>
<td>57.17</td>
<td>44.1</td>
</tr>
</tbody>
</table>

Avg. difference 0.04
Spearman rank corr. 0.2 0.28

Each pair is referred to by an identifier. Please refer to our Supplementary Material or Mathews and Turner (2003) to see the exact sequences of each pair. All values, except Spearman's rank correlation, are in degree centigrade. Bold entries are the most accurate predictions. In other words, they have the least difference from experimental measurements.
Altuvia (2000). Those plots can be predicted from the partition functions for OxyS, fhlA, OxyS–OxyS, fhlA–fhlA and OxyS-fhlA. Therefore based on (8), a method can compute the concentrations from partition functions, and why it is difficult to correctly predict those equilibrium concentrations.

Given two nucleic acid strands \( R \) and \( S \), we can compute the equilibrium concentrations of \( R, S, RR, SS \) and \( RS \) species, denoted by \( N_R, N_S, N_{RR}, N_{SS} \) and \( N_{RS} \), respectively, from their partition functions (Dimitrov and Zuker, 2004). In the equilibrium, the free energy of a closed system at constant temperature, volume and pressure tends toward a minimum (Landau and Lifshitz, 1969). Equilibrium concentrations are computed from the chemical equilibrium constants

\[
K_R = \frac{Q_{RR}}{Q_R^{0}} N_{RR} N_{S} \sum_{R}^{0} R \\
K_S = \frac{Q_{SS}}{Q_S^{0}} N_{SS} N_{R} \sum_{S}^{0} S \\
K_{RS} = \frac{Q_{RS}}{Q_R Q_S} N_{RS} N_{RS}^{0}
\]

under the constraint \( N_{RS} = N_{R}^{0} - 2N_{RR} - N_{R} = N_{S}^{0} - 2N_{SS} - N_{S} \), in which \( N^{0} \) are the initial concentrations of single strands. We noticed that \( Q_{R} \) and \( Q_{S} \) computed by the three programs are very close because they use the same algorithm for a single strand (i.e. McCaskill’s algorithm). Therefore based on (8), a method can compute equilibrium concentrations correctly only if it computes each individual \( Q^{0} \) accurately. As one can observe in Figure 11, our program has been able to predict OxyS–fhlA complex concentrations accurately, thus we can conclude that our program computes all \( Q^{0} \) accurately.

As mentioned above, the parameters used by our program on this dataset have been manually optimized. Our energy parameters in this experiment are \( \beta_1 = 6.6, \beta_2 = 0.1, \sigma = 0.9, \beta_1' = 4.5 \) and \( \sigma' = 0.9 \).

The running time of our program for the first dataset was about a few seconds, for the second dataset about 10 min and for the third dataset \( \sim 72 \) h on a Linux PC with Pentium-D 3.6 GHz CPU and 4 GB of RAM. Note that we did not use any learning methods for tuning our six interaction energy parameters because of the running time of our program. Our interaction energy parameters in melting temperature experiments are \( \beta_1 = 5.1, \beta_2 = 0.1, \sigma = 0.92, \beta_1' = 4.1 \) and \( \sigma' = 0.95 \), which were manually optimized using only the first data set. The second and the third datasets were used as test sets.

### 3.4 Equilibrium concentration

Our second set of experiments, to the best of our knowledge, have not been successfully performed by the use of any available program to date. Here we predict the equilibrium concentrations for OxyS with wild-type fhlA and four other fhlA mutants. OxyS is a small untranslated RNA (109 nt) that is induced in response to oxidative stress in *E. coli*. It acts as a regulator affecting the expression of multiple genes. In particular, OxyS represses metabolism, by binding to it. Argaman and Altuvia (2000) carried the translation of fhlA, a transcriptional activator for formate metabolism, as we had expected.

Figure 11 shows the experimental measurements and our results. Interestingly, our algorithm predicted the equilibrium concentration of OxyS–fhlA complex quite accurately for the wild-type fhlA and all of its mutants. We also experimented with RNAcofold and UNAFold in this case. Both RNAcofold and UNAFold predict that the percentage of OxyS in complex is approximately 0 in all five cases for the considered fhlA concentrations. This is probably not very surprising as correctly predicting the equilibrium concentrations is a very difficult task and is highly sensitive to the accuracy of the partition functions. We describe below how to compute the concentrations from partition functions, and why it is difficult to correctly predict those equilibrium concentrations.

Given two nucleic acid strands \( R \) and \( S \), we can compute the equilibrium concentrations of \( R, S, RR, SS \) and \( RS \) species, denoted by \( N_R, N_S, N_{RR}, N_{SS} \) and \( N_{RS} \), respectively, from their partition functions (Dimitrov and Zuker, 2004). In the equilibrium, the free energy of a closed system at constant temperature, volume and pressure tends toward a minimum (Landau and Lifshitz, 1969). Equilibrium concentrations are computed from the chemical equilibrium constants

\[
K_R = \frac{Q_{RR}}{Q_R^{0}} N_{RR} N_{S} \sum_{R}^{0} R \\
K_S = \frac{Q_{SS}}{Q_S^{0}} N_{SS} N_{R} \sum_{S}^{0} S \\
K_{RS} = \frac{Q_{RS}}{Q_R Q_S} N_{RS} N_{RS}^{0}
\]

under the constraint \( N_{RS} = N_{R}^{0} - 2N_{RR} - N_{R} = N_{S}^{0} - 2N_{SS} - N_{S} \), in which \( N^{0} \) are the initial concentrations of single strands. We noticed that \( Q_{R} \) and \( Q_{S} \) computed by the three programs are very close because they use the same algorithm for a single strand (i.e. McCaskill’s algorithm). Therefore based on (8), a method can compute equilibrium concentrations correctly only if it computes each individual \( Q^{0} \) accurately. As one can observe in Figure 11, our program has been able to predict OxyS–fhlA complex concentrations accurately, thus we can conclude that our program computes all \( Q^{0} \) accurately.

As mentioned above, the parameters used by our program on this dataset have been manually optimized. Our energy parameters in this experiment are \( \beta_1 = 6.6, \beta_2 = 0.1, \sigma = 0.9, \beta_1' = 4.5 \) and \( \sigma' = 0.9 \).
4 CONCLUSION AND FUTURE WORK

In this article, we present piRNA, an efficient algorithm to compute a partition function of two interacting nucleic acid strands. Our algorithm considers all almost physically possible secondary structures that do not contain pseudoknots, crossing interactions and ‘zigzag’s. In order to specify the free energy of a joint structure established by interacting strands, we extend the standard neighbor single-strand thermodynamic energy model to an energy model for two interacting strands by introducing three new components: (i) hybrid component, (ii) kissing loop and (iii) inter-hybrid loops that are modified versions of hybridization, multi loop and inter-hybrid energy models. We verified our algorithm by computing the melting temperature for RNA pairs available in the literature and the equilibrium concentration for OxyS-fhlA complex. In both experiments our algorithm provides high accuracy and it is several times faster than available alternatives.

We computed the melting temperature for RNA pairs in (Diamond et al., 2001; Mathews and Turner, 2002; Xia et al., 1998) (Tables 1–3). On average, the predicted melting temperature by our program is ~2°C different from experimental values. Our program is >103°C more accurate than the alternatives, RNAcofold and UNAFold, on average. It is important to note that RNAcofold and UNAFold both perform poorly in at least one of the three datasets, while our program is consistently accurate across all three datasets. Therefore, neither RNAcofold nor UNAFold is as reliable as our program for melting temperature prediction. In addition, our algorithm is able to compute the OxyS-fhlA complex equilibrium concentrations for wild-type and mutated fhlA accurately. Both RNAcofold and UNAFold predict those equilibrium concentrations to be approximately 0, which does not even roughly follow the experimental measurements.

Although our algorithm is fairly efficient, improving the generality and complexity of our algorithm will be one of our priorities in the near future. In particular, we aim to explore whether it is possible to cover more general interactions without increasing the computational complexity of the algorithm.

Funding: Bioinformatics for Combatting Infectious Diseases (BCID) initiative (to H.C.); MITACS Research Grant (to R.S.); Michael Smith Foundation for Health Research Career Award (to S.C.S.). 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) (to R.B.).

Conflict of Interest: none declared.

REFERENCES
