BIOINFORMATICS

Vol. 29 ISMB/ECCB 2013, pages i300–i307
doi:10.1093/bioinformatics/btt226

The RNA Newton polytope and learnability of energy parameters

Elmirasadat Forouzmand and Hamidreza Chitsaz*

Department of Computer Science, Wayne State University, Detroit, MI 48202, USA

ABSTRACT

Motivation: Computational RNA structure prediction is a mature important problem that has received a new wave of attention with the discovery of regulatory non-coding RNAs and the advent of high-throughput transcriptome sequencing. Despite nearly two score years of research on RNA secondary structure and RNA–RNA interaction prediction, the accuracy of the state-of-the-art algorithms are still far from satisfactory. So far, researchers have proposed increasingly complex energy models and improved parameter estimation methods, experimental and/or computational, in anticipation of endowing their methods with enough power to solve the problem. The output has disappointingly been only modest improvements, not matching the expectations. Even recent massively featured machine learning approaches were not able to break the barrier. Why is that?

Approach: The first step toward high-accuracy structure prediction is to pick an energy model that is inherently capable of predicting each and every one of known structures to date. In this article, we introduce the notion of *learnability* of the parameters of an energy model as a measure of such an inherent capability. We say that the parameters of an energy model are *learnable* iff there exists at least one set of such parameters that renders *every* known RNA structure to date the minimum free energy structure. We derive a necessary condition for the learnability and give a dynamic programming algorithm to assess it. Our algorithm computes the convex hull of the feature vectors of all feasible structures in the ensemble of a given input sequence. Interestingly, that convex hull coincides with the *Newton polytope* of the partition function as a polynomial in energy parameters. To the best of our knowledge, this is the first approach toward computing the RNA Newton polytope and a systematic assessment of the inherent capabilities of an energy model. The worst case complexity of our algorithm is exponential in the number of features. However, dimensionality reduction techniques can provide approximate solutions to avoid the curse of dimensionality.

Results: We demonstrated the application of our theory to a simple energy model consisting of a weighted count of A-U, C-G and G-U base pairs. Our results show that this simple energy model satisfies the necessary condition for more than half of the input unpseudoknotted sequence–structure pairs (55%) chosen from the RNA STRAND v2.0 database and severely violates the condition for ~13%, which provide a set of hard cases that require further investigation. From 1350 RNA strands, the observed 3D feature vector for 749 strands is on the surface of the computed polytope. For 289 RNA strands, the observed feature vector is not on the boundary of the polytope but its distance from the boundary is not more than one. A distance of one essentially means one base pair difference between the observed structure and the closest point on the boundary of the polytope, which need not be the feature vector of a structure. For 171 sequences, this distance is larger than two, and for only 11 sequences, this distance is larger than five.

Availability: The source code is available on http://compbio.cs.wayne.edu/software/rna-newton-polytope.

Contact: chitsaz@wayne.edu

1 INTRODUCTION

Computational RNA structure and RNA–RNA interaction prediction have always been important problems, particularly now that RNA has been shown to have key regulatory roles in the cell (Bartel, 2004; Brantl, 2002; Gottesman, 2005; Hannon, 2002; Storz, 2002; Wagner and Flardh, 2002; Zamore and Haley, 2005). Furthermore, with the advent of synthetic biology at the whole organism level (Gibson *et al*., 2010), high-throughput accurate RNA engineering algorithms are required for both *in vivo* and *in vitro* applications (Seeman, 2005; Seeman and Lukeman, 2005; Simmel and Dittmer, 2005; Venkataraman *et al*., 2007; Yin *et al*., 2008). Since the dawn of RNA secondary structure prediction nearly two score years ago (Tinoco *et al*., 1973), the research community has proposed increasingly complex models and algorithms, hoping that refined features together with better methods to estimate their parameters would solve the problem. Early approaches considered mere base pair counting, followed by the Turner thermodynamics model, which was a significant leap forward. Recently, massively feature-rich models empowered by parameter estimation algorithms have been proposed, but they provide only modest improvements.

Despite significant progress in the last three decades, made possible by the work of Mathews *et al*. (1999) on measuring RNA thermodynamic energy parameters and the work of several groups on novel algorithms (Bernhart *et al*., 2006; Chitsaz *et al*., 2009a, b; Dirks and Pierce, 2003; McCaskill, 1990; Nussinov *et al*., 1978; Rivas and Eddy, 1999; Waterman and Smith, 1978; Zuker and Stiegler, 1981) and machine learning approaches (Andronescu *et al*., 2010; Do *et al*., 2006; Zakov *et al*., 2011), the RNA structure prediction accuracy has not reached a satisfactory level yet (Rivas *et al*., 2012). Why is it so? Up to now, human intuition and computational convenience have led the way. We believe that human intuition has to be equipped with systematic methods to assess the suitability of a given energy model. Surprisingly, there is not a single method to assess whether the parameters of an energy model are *learnable*. We say that the parameters of an energy model are *learnable* iff there exists at least one set of such parameters that renders *every* known RNA structure to date, determined through radiograph or NMR, the minimum free energy structure. Equivalently, we say that the parameters of an energy model are learnable iff 100% structure prediction accuracy can be achieved when the training and test sets are identical. The first step toward high-accuracy structure prediction is to make sure that the

*To whom correspondence should be addressed.

energy model is inherently capable, i.e. its parameters are learnable. In this work, we provide a necessary condition for the learnability and an algorithm to verify it. The problem of learnability was previously studied for sequence alignment using similar methods (Dewey *et al*., 2006). To the best of our knowledge, this is the first approach toward a systematic assessment of the suitability of an energy model for RNA structure prediction. Note that a successful RNA folding algorithm needs to have the generalization power to predict unseen structures as well. However, if an energy model is inherently incapable of predicting known structures correctly, then it does not matter if it has the generalization power. We leave analysis of the generalization power for future work.

2 BACKGROUND

2.1 RNA secondary structure models

An RNA secondary structure model is often a context-free grammar together with a scoring function for either the rules, in the case of stochastic context-free grammars (SCFGs) (Eddy and Durbin, 1994), or the alphabet, in the case of thermodynamics models (Mathews *et al*., 1999). Such scoring functions induce scoring on the entire generated language. The word with optimal score then yields a predicted structure for the given sequence. For the sake of brevity, we focus on thermodynamics models in this article, but it is obvious that our methods apply to other models including SCFG as well. In our context, the scoring function is the thermodynamics free energy. A secondary structure y of a nucleic acid is decomposed into loops, a free energy is associated with every loop in y and the total free energy G for y is the sum of loop free energies (Mathews *et al*., 1999). The same loop decomposition principle applies to interacting nucleic acids such that the total free energy G is still the sum of the free energies of loops and interaction components (Chitsaz *et al*., 2009b).

2.2 Estimation of energy parameters

Generally, RNA structure prediction algorithms are divided into probabilistic and non-probabilistic categories. Besides machine learning approaches for the probabilistic methods, e.g. SCFGs, which are evaluated in (Rivas *et al*., 2012), existing machine learning algorithms for parameter estimation in the non-probabilistic RNA structure prediction can be grouped into two categories:

- Likelihood-based methods, where the maximum likelihood or maximum conditional likelihood principle is used to estimate the parameters of the model (e.g. Do *et al*., 2006); and
- Large-margin methods, where the model parameters are estimated to maximize the margin between the score of the true structure and the second best structure. This has been done using an online passive-aggressive training algorithm (Zakov *et al*., 2011) and Iterative Constraint Generation (Andronescu *et al*., 2007).

The likelihood-based techniques estimate the best *Gibbs* distribution, which not only assists in predicting the best secondary structure but also is used in determining the thermodynamic parameters. Besides some probabilistic methods (Rivas *et al*., 2012), one of the most successful methods for learning the thermodynamics of RNA has been the MCL method, as in CONTRAfold (Do *et al*., 2006), which maximizes the probability of RNA structures y given RNA sequences x for the training set D. That is, the conditional log likelihood of the training data (using the Boltzmann distribution) is maximized to estimate the best model parameters $\mathbf{h}^* \in \mathbb{R}^k$:

$$\mathbf{h}^* := \arg\max_{\mathbf{h}} L(D;\mathbf{h}) = \max_{\mathbf{h}} \sum_{(x,y)\in D} \log p(y|x,\mathbf{h}) \tag{1}$$

$$p(y|x,\mathbf{h}) := \frac{e^{-G(x,y,\mathbf{h})/RT}}{Q(x,\mathbf{h})} \tag{2}$$

where k denotes the number of different motifs defined in the energy model, R is the gas constant, T is the absolute temperature, $G(x,y,\mathbf{h})$ is the free energy and

$$Q(x,\mathbf{h}) := \sum_{s\in\mathcal{E}(x)} e^{-G(x,s,\mathbf{h})/RT} \tag{3}$$

is the *partition function* (Chitsaz *et al*., 2009b; Dirks and Pierce, 2003; McCaskill, 1990) with $\mathcal{E}(x)$ being the ensemble of possible structures of x. The free energy

$$G(x,s,\mathbf{h}) := \langle c(x,s),\mathbf{h}\rangle \tag{4}$$

is a linear function of the parameters $\mathbf{h}$ where $c(x,s) \in \mathbb{Z}^k$ is the features vector.

The best performing method for RNA STRAND v2.0 is Contextfold (Zakov *et al*., 2011). However, Contextfold performs worse than other methods in other datasets that are structurally different from RNA STRAND v2.0 (Rivas *et al*., 2012). Essentially, it seems that Contextfold suffers from overfitting because it uses a myriad of features in its model. A systematic evaluation of the learnability using our algorithm in this article may help design a more concise set of features that is powerful enough to achieve high accuracy but not too powerful to suffer from overfitting.

3 LEARNABILITY

The question that we ask before parameter estimation is 'does there ever exist parameters $\mathbf{h}^\dagger$ such that for every $(x,y)\in D$, $y = \arg\min_s G(x,s,\mathbf{h}^\dagger)$?' If the answer to this question is no, then there is no hope that one can ever achieve 100% accuracy using the given model. The answer reveals inherent limitations of the model, which can be used to design improved models. We provide a necessary condition for the existence of $\mathbf{h}^\dagger$ and a dynamic programming algorithm to verify it through computing the Newton polytope for every x in D. We will define the RNA Newton polytope below. Not only our algorithm provides a binary answer, it also quantifies the distance from the boundary. For our 3D model, this distance roughly captures the difference between the number of base pairs in the observed structure and the number of base pairs in the closest point on the boundary of the polytope. Note that the closest point on the boundary of the polytope need not be the feature vector of a structure.

4 METHODS

4.1 Necessary condition for learnability

Let $(x, y) \in D$ and $\mathbf{h}^\dagger \in \mathbb{R}^k$. Assume y minimizes $G(x, s, \mathbf{h}^\dagger)$ as a function of s. In that case

$$G(x, y, \mathbf{h}^\dagger) \le G(x, s, \mathbf{h}^\dagger),\ \forall\, s \in \mathcal{E}(x). \tag{5}$$

Replacing equation (4) above,

$$\left\langle c(x, y), \mathbf{h}^\dagger \right\rangle \le \left\langle c(x, s), \mathbf{h}^\dagger \right\rangle,\ \forall\, s \in \mathcal{E}(x) \tag{6}$$

$$0 \le \left\langle c(x, s) - c(x, y), \mathbf{h}^\dagger \right\rangle,\ \forall\, s \in \mathcal{E}(x). \tag{7}$$

Define the *feature ensemble* of sequence x by

$$\mathcal{F}(x) := \left\{ c(x, s) \mid s \in \mathcal{E}(x) \right\} \subset \mathbb{Z}^k. \tag{8}$$

In that case, equation (7) implies that

$$0 \le \left\langle \mathcal{F}(x) - c(x, y), \mathbf{h}^\dagger \right\rangle. \tag{9}$$

We call the convex hull of $\mathcal{F}(x)$ the *Newton polytope* of x,

$$\mathcal{N}(x) := \mathrm{conv}\left\{ \mathcal{F}(x) \right\} \subset \mathbb{R}^k. \tag{10}$$

We remind the reader that the convex hull of a set, denoted by 'conv' hereby, is the minimal convex set that fully contains the set. The reason for naming this polytope the Newton polytope will be made clear below. Inequality equation (9) implies that $c(x, y) \in \partial\mathcal{N}(x)$ is on the boundary of the convex hull of the feature ensemble of x with a support hyperplane normal to $\mathbf{h}^\dagger$. Therefore, we have the following theorem.

Theorem 1. Let $(x, y) \in D$ and $0 \ne \mathbf{h}^\dagger \in \mathbb{R}^k$. Assume y minimizes $G(x, s, \mathbf{h}^\dagger)$ as a function of s. In that case, $c(x, y) \in \partial\mathcal{N}(x)$, i.e. the feature vector of (x, y) is on the boundary of the Newton polytope of x.

Proof. To the contrary, suppose $c(x, y)$ is in the interior of $\mathcal{N}(x)$. Therefore, there is an open ball of radius $\delta > 0$ centered at $c(x, y)$ completely contained in $\mathcal{N}(x)$, i.e.

$$B_\delta(c(x, y)) \subset \mathcal{N}(x). \tag{11}$$

Let

$$p = c(x, y) - (\delta/2)\frac{\mathbf{h}^\dagger}{||\mathbf{h}^\dagger||}.$$

It is clear that $p \in B_\delta(c(x, y)) \subset \mathcal{N}(x)$ since $||p - c(x, y)|| = \delta/2 < \delta$. Therefore, p can be written as a convex linear combination of the feature vectors in $\mathcal{F}(x) = \{v_1, \ldots, v_N\}$, i.e.

$$\exists\, \alpha_1, \ldots \alpha_N \ge 0 :\ \alpha_1 v_1 + \cdots + \alpha_N v_N = p \tag{12}$$

$$\alpha_1 + \cdots + \alpha_N = 1. \tag{13}$$

Note that

$$\left\langle p - c(x, y), \mathbf{h}^\dagger \right\rangle = -(\delta/2)||\mathbf{h}^\dagger|| < 0. \tag{14}$$

Therefore, there is $1 \le i \le N$, such that $\left\langle v_i - c(x, y), \mathbf{h}^\dagger \right\rangle < 0$ for otherwise,

$$\left\langle p - c(x, y), \mathbf{h}^\dagger \right\rangle = \sum_{i=1}^{N} \alpha_i \left\langle v_i - c(x, y), \mathbf{h}^\dagger \right\rangle \ge 0 \tag{15}$$

which would be a contradiction with equation (14). It is now sufficient to note that $v_i \in \mathcal{F}(x)$ and $\left\langle v_i - c(x, y), \mathbf{h}^\dagger \right\rangle < 0$, which is a contradiction with equation (9). □

Corollary 1. *(Necessary Condition for the Learnability)*. For $(x, y) \in D$, a necessary but not sufficient condition for the existence of $\mathbf{h}^\dagger$ such that y minimizes $G(x, s, \mathbf{h}^\dagger)$ as a function of s is that $c(x, y)$ lies on the boundary of $\mathcal{N}(x)$ the Newton polytope of x.

4.2 Relation to the Newton polytope

In addition to D, the set of experimentally determined structures, we often have a repository of thermodynamic measurements, e.g. melting curves, which can help better estimate the energy parameters. Currently, optical melting measurements are analyzed using a two-state model (Siegfried and Bevilacqua, 2009). However, a more accurate analysis of such melting experiments can be done through relating the measurements to the energy parameters through equations involving the partition function and its derivatives with respect to temperature (Chitsaz *et al.*, 2009b). We show that with a change of variables, the partition function becomes a polynomial. Therefore, such equations become a system of polynomial equations the solving of which algebraically requires computation of the Newton polytope of each polynomial (Emiris, 1994; Emiris and Canny, 1995). Recall the partition function defined in equation (3) and energy in equation (4), and conclude

$$Q(x, \mathbf{h}) = \sum_{s \in \mathcal{E}(x)} e^{-\left\langle c(x, s), \mathbf{h} \right\rangle / RT}. \tag{16}$$

Let $c(x, s) = (c_1(x, s), \ldots, c_k(x, s))$ and $\mathbf{h} = (\mathbf{h}_1, \ldots, \mathbf{h}_k)$. Define new variables

$$Z_i := e^{-\mathbf{h}_i / RT},\ 1 \le i \le k \tag{17}$$

and replace them in equation (16). We obtain the partition function

$$Q(x, Z) = \sum_{s \in \mathcal{E}(x)} Z^{c(x, s)} \tag{18}$$

in the form of a polynomial in $\mathbb{R}[Z]$ where

$$Z^{c(x, s)} := \prod_{i=1}^{k} Z_i^{c_i(x, s)} \tag{19}$$

is a monomial as $0 \le c_i(x, s) \in \mathbb{Z}$. The Newton polytope of Q is defined to be the convex hull of the monomials power vectors, i.e.

$$\mathrm{Newton}\left\{ Q(x, Z) \right\} := \mathrm{conv}\left(\left\{ c(x, s) \mid s \in \mathcal{E}(x) \right\} \right) = \mathcal{N}(x). \tag{20}$$

That is why we call $\mathcal{N}(x)$ the Newton polytope of x.

4.3 RNA Newton polytope algorithm

We give a dynamic programming algorithm to compute the Newton polytope for a given nucleic acid sequence x. Denote the length of x by L and the i^{th} nucleotide in x by n_i. Denote the subsequence of x from the i^{th} to the j^{th} nucleotide, inclusive of ends, by $n_i \cdots n_j$. The following lemma allows us to formulate a divide-and-conquer strategy for computing the Newton polytope, which will in turn lead to our dynamic programming algorithm.

Lemma 1. Let f and g be two polynomials in $\mathbb{R}[Z]$. The Newton polytope of the product of f and g is the Minkowski sum of individual Newton polytopes, and the Newton polytope of the sum of f and g is the convex hull of the union of individual Newton polytopes, i.e.

$$Newton\ (fg) = Newton\ (f) \oplus Newton\ (g) \tag{21}$$

$$Newton\ (f + g) = conv\left\{ Newton\ (f) \cup Newton\ (g) \right\} \tag{22}$$

in which $\oplus$ *represents the Minkowski sum of two polytopes* (Emiris, 1994).

This lemma allows us to use the same divide-and-conquer strategy that was used for calculating the partition function (Chitsaz *et al.*, 2009b; Dirks and Pierce, 2003; McCaskill, 1990). We can use the same recursions (grammar) as in the partition function algorithm but with the Minkowski sum $\oplus$ instead of multiplication, convex hull of union instead of summation and the corresponding feature vector c instead of $e^{-\langle c, \mathbf{h} \rangle / RT}$. Furthermore, because union is invariant with respect to repetition of points, the dynamic programming is allowed to be redundant, or equivalently the grammar is allowed to be ambiguous. Hence, any complete

RNA structure or RNA–RNA interaction prediction dynamic programming algorithm can be transformed into a Newton polytope algorithm by replacing the energy with the corresponding feature vector, summation with the Minkowski sum $\oplus$ and minimization with the convex hull of union.

As explained above, we transform any complete partition function or structure prediction dynamic programming algorithm, for single RNA, RNA–RNA interaction or multiple interacting RNAs, into a Newton polytope algorithm. For the sake of illustration, we explicitly spell below only the case of single RNA with separate A-U, C-G and G-U base pair counting energy model. All the other cases are trivially obtained following the transformations above.

In this case, the feature vector

$$c(x,s) = (c_1(x,s), c_2(x,s), c_3(x,s))$$

is 3D: $c_1(x,s)$ is the number of A-U, $c_2(x,s)$ the number of C-G and $c_3(x,s)$ the number of G-U base pairs in s. Our dynamic programming algorithm starts by computing the Newton polytope for all unit length subsequences, followed by all length two subsequences, ..., up to the Newton polytope for the entire sequence x. We denote the Newton polytope of the subsequence $n_i \cdots n_j$ by $\mathcal{N}(i,j)$, i.e.

$$\mathcal{N}(i,j) := \mathcal{N}(n_i \cdots n_j) \tag{23}$$

The following dynamic programming will yield the result

$$\mathcal{N}(i,j) = \\ \text{conv}\left[\cup \begin{cases} \mathcal{N}(i,\ell) \oplus \mathcal{N}(\ell+1,j), & i \le \ell \le j-1 \\ \{(1,0,0)\} \oplus \mathcal{N}(i+1,j-1) & \text{if } n_i n_j = \text{AU}|\text{UA} \\ \{(0,1,0)\} \oplus \mathcal{N}(i+1,j-1) & \text{if } n_i n_j = \text{CG}|\text{GC} \\ \{(0,0,1)\} \oplus \mathcal{N}(i+1,j-1) & \text{if } n_i n_j = \text{GU}|\text{UG} \end{cases} \right] \tag{24}$$

with the base case $\mathcal{N}(i,i) = \{(0,0,0)\}$. To compare with only (AU, CG) base pair counting, we also compute the 2D Newton polygon by just projecting the 3D polytope onto the first two coordinates.

There are two different approaches for polytope representation: (i) vertex representation, which is a set of points, and (ii) half plane representation, which is a set of linear inequalities. The former is often called $\mathcal{V}$-representation and the latter $\mathcal{H}$-representation. Although they are equivalent, and there are algorithms to transform one into the other, computing Minkowski sum is more convenient with the $\mathcal{V}$-representation, and convex hull of union works more efficiently with the $\mathcal{H}$-representation. The choice of representation and algorithms will affect the running time. In this article, we use the $\mathcal{V}$-representation. Owing to the complexity of the convex hull problem, the worst case complexity of our algorithm is exponential in the number of features.

4.4 Verification of the necessary condition

On computation of $\mathcal{N}(x)$ and $c(x,y)$, the feature vector of the experimentally determined structure, it remains to verify whether $c(x,y) \in \partial\mathcal{N}(x)$. Often, $\mathcal{N}(x)$ is represented by its vertices ($\mathcal{V}$-representation) or its confining half planes ($\mathcal{H}$-representation), two equivalent representations that can be transformed into one another. In an $\mathcal{H}$-representation, $c(x,y)$ is on the boundary of $\mathcal{N}(x)$ iff there is at least one confining plane on which $c(x,y)$ lies. This is true because $c(x,y) \in \mathcal{N}(x)$ anyways. Therefore, the necessary condition can be easily checked by checking membership of $c(x,y)$ in every confining plane. Because the vertices of $\mathcal{N}(x)$ are on the integer lattice, all calculations are rational and hence can be performed exactly.

4.5 Dataset

We used 1350 unpseudoknotted RNA sequence–structure pairs from RNA STRAND v2.0 database as our dataset D. RNA STRAND v2.0 contains known RNA secondary structures of any type and organism, particularly with and without pseudoknots. RNA STRAND v2.0 (Andronescu *et al.*, 2008) is a convenient source of RNA sequences and structures selected from various Rfam families (Burge *et al.*, 2013). There are 2334 pseudoknot-free RNAs in the RNA STRAND database. We sorted them based on their length and selected the first 1350 ones, whose lengths vary between 4 and 121 nt. We excluded pseudoknotted structures because our current implementation is incapable of considering pseudoknots. Some sequences in the dataset allow only A-U base pairs (not

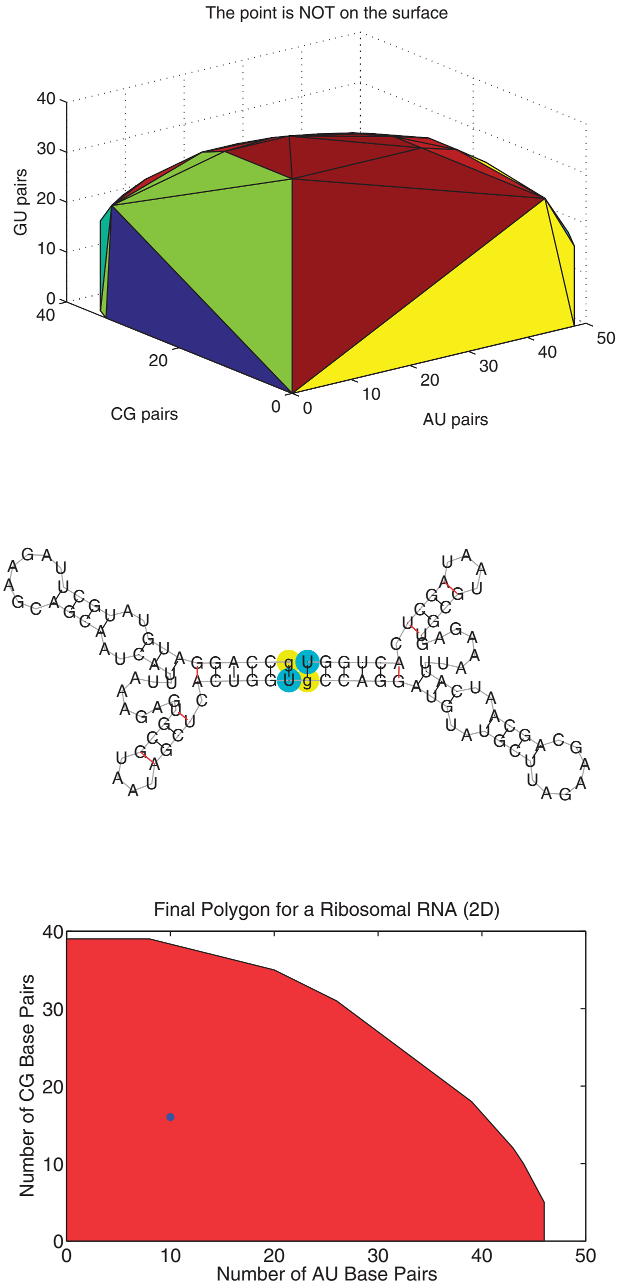

Fig. 1. (Top) The 3D Newton polytope of a ribosomal RNA. The observed feature vector is not on the surface, $r(x) = 2$. (Middle) The observed secondary structure of the ribosomal RNA. (Bottom) The 2D Newton polygon of the ribosomal RNA, $r(x) = 10$. The blue point represents the observed 2D feature vector

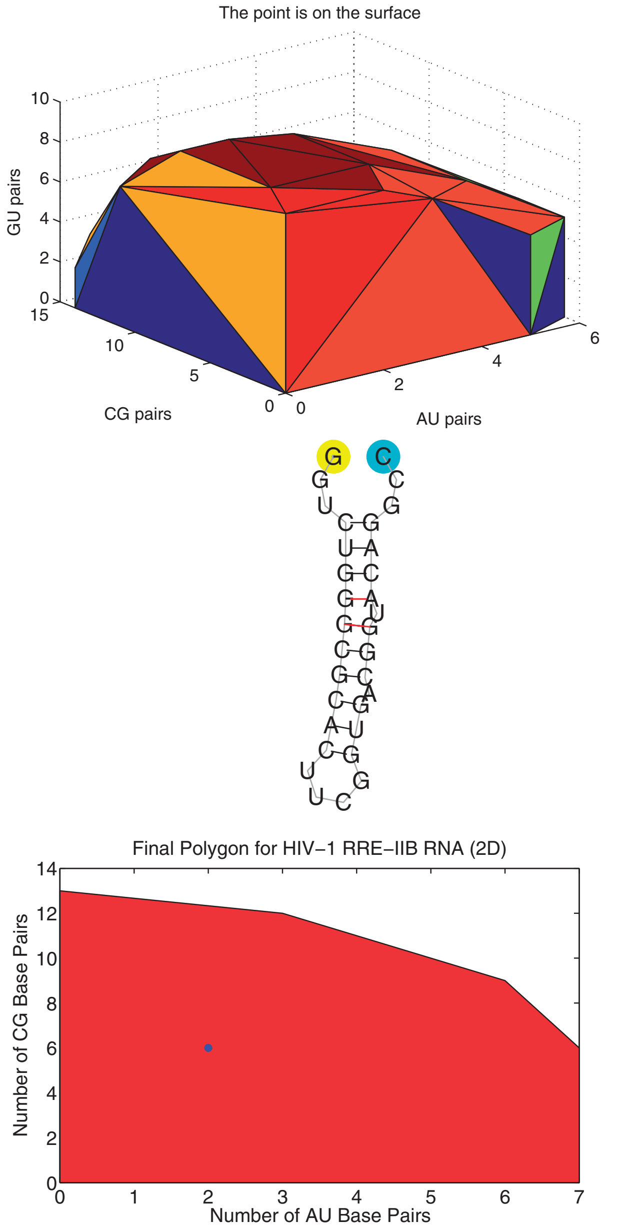

Fig. 2. (Top) The 3D Newton polytope of HIV-1 RRE-IIB 32 NUCLEOTIDE RNA. The observed feature vector is on the surface, $r(x) = 0$. (Middle) The observed secondary structure of HIV-1 RRE-IIB 32 NUCLEOTIDE RNA. (Bottom) The 2D Newton polygon of HIV-1 RRE-IIB 32 NUCLEOTIDE RNA, $r(x) = 2$. The blue point represents the observed 2D feature vector. Note that even though the necessary condition is not satisfied in 2D, it is in 3D

a single C-G or G-U pair), in which case the Newton polytope degenerates into a line.

4.6 Implementation

We implemented the dynamic programming in equation (24) using MATLAB convex hull function, which is based on the quickhull algorithm (Barber *et al.*, 1996). As mentioned above, we used the $\mathcal{V}$-representation and computed the Minkowski sum by direct pairwise summation of vertices. More precisely, for two convex polytopes P with vertices $p_1, \ldots, p_a$ and Q with vertices $q_1, \ldots, q_b$, the vertices of $P \oplus Q$ are $p_i + q_j$ for $1 \leq i \leq a$ and $1 \leq j \leq b$. To verify the

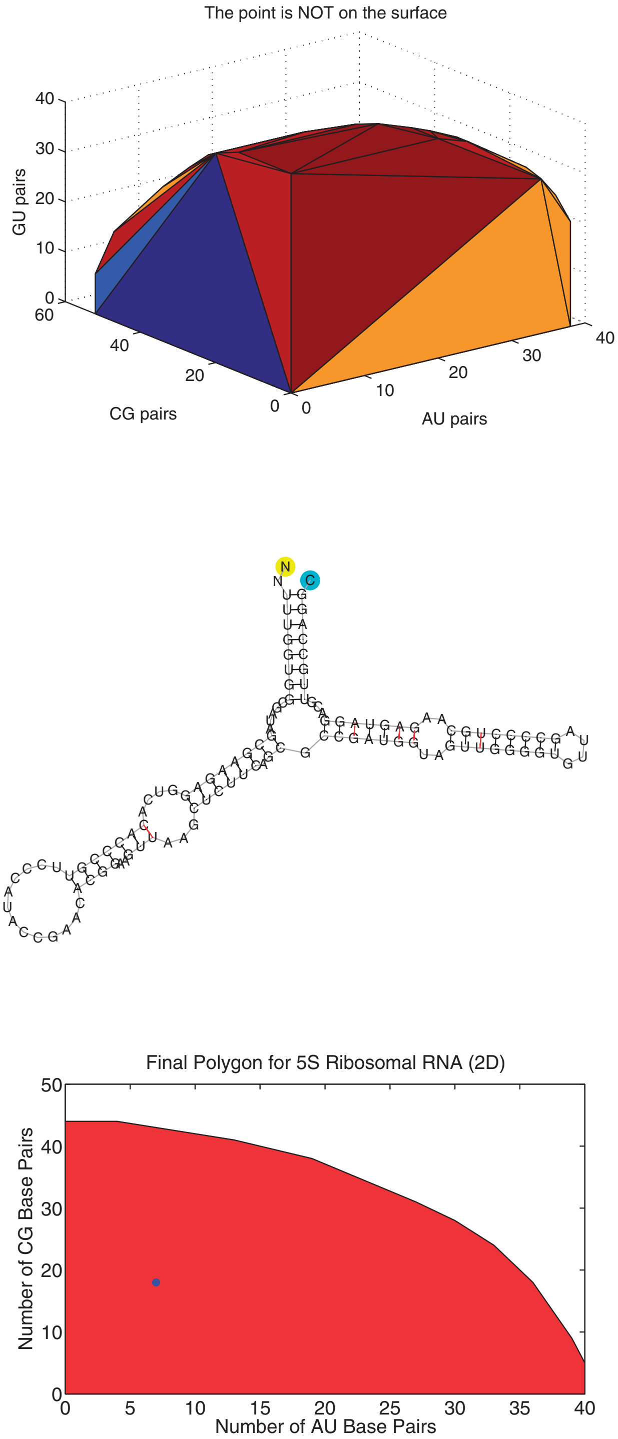

Fig. 3. (Top) The 3D Newton polytope of *E.coli* 5S rRNA. The observed feature vector is not on the surface, $r(x) = 7$. (Middle) The observed secondary structure of *E.coli* 5S rRNA. (Bottom) The 2D Newton polygon of *E.coli* 5S rRNA, $r(x) = 7$. The blue point represents the observed 2D feature vector

necessary condition, i.e. whether the experimentally determined feature vector lies on the boundary of the Newton polytope, we calculated the distance of the feature vector from the boundary of the polytope using the ‘p_poly_dist’ MATLAB function (Yoshpe, 2006). A zero distance corresponds to the case where the feature vector lies on the boundary, i.e. the condition is satisfied, and a positive distance to the case where the feature vector is in the interior of the Newton polytope. We normalized the distance by third root of the volume of the polytope (square root of the area in the case of a polygon). The normalized distance quantifies how far the feature vector is from the boundary. We parallelized our MATLAB code using MATLAB ‘parfor’. The length of input RNA sequences varied between 4 and ~120 nt. For the smallest ones, our program took a fraction of a second and for the longest ones it took < 10 min to run on a 2.5 GHz 12-Core AMD Opteron CPU.

5 RESULTS

For each strand of RNA, the distance between $c(x, y)$, the real feature vector of the secondary structure and the computed convex hull, $\mathcal{N}(x)$, is calculated using (Yoshpe, 2006). We denote this distance by $r(x)$ here. The necessary condition for the learnability is satisfied if $r(x) = 0$ for all x in the dataset, which shows that the observed feature vector lies on the boundary of $\mathcal{N}(x)$.

Figure 1 illustrates the secondary structure of a ribosomal RNA and its 2D and 3D Newton polytopes. This RNA is 116 nt long, and the distance between the polytope and the observed feature vector for this RNA is 2. In the 2D (AU, CG) model, this distance is 10. The observed feature vector has moved closer to the boundary, going from 2D to 3D, as GU pairs are accounted for in the 3D model. Figure 2 shows a shorter HIV-1 RNA with the length of 32 nt. The distance between the 3D polytope and the observed feature vector for this RNA is 0, while there is a distance of 2 in the 2D model. In this case also, the observed feature vector has moved closer to the boundary, going from 2D to 3D. Because there is no GU pair in this observed structure, the observed feature vector lies on the $c_3(x, y) = 0$ face, which is on the boundary of the polytope. Figure 3 illustrates *Escherichia coli* 5S ribosomal RNA, which is 121 nt long, and the distance between the feature vector and the polytope is 7 in this case. The distance has not changed in the 3D model.

Figure 4 demonstrates the histogram of $r(x)$ for the input dataset. Out of 1350 strands of RNA, the observed feature vector is on the surface of the 3D Newton polytope for 749 (55%) sequences. The distance r is not zero, but not more than one, for 289 (21%) sequences. For 141 (~10%) strands, this

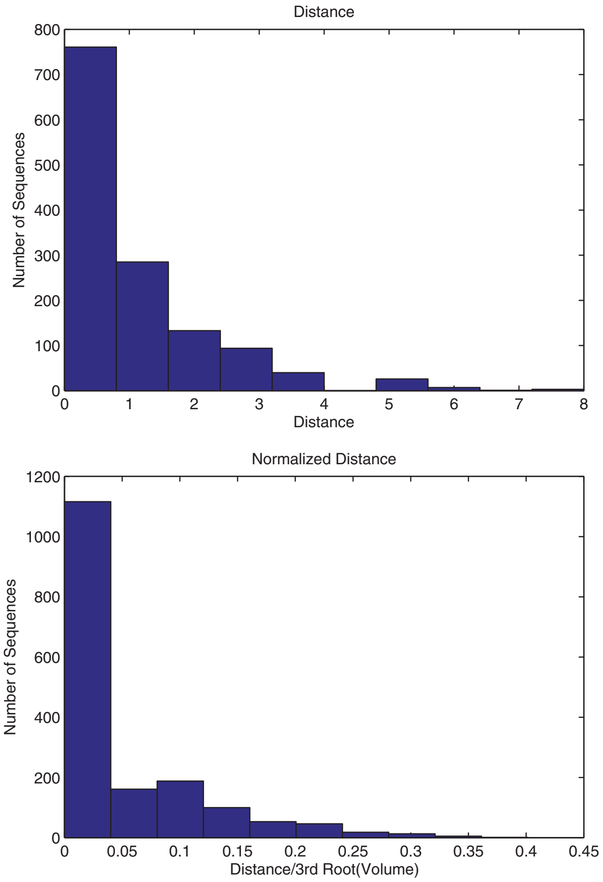

Fig. 4. (Top) Histogram of $r(x)$ in the 3D model. (Bottom) Histogram of $r(x)/\sqrt[3]{\mathrm{Vol}(\mathcal{N}(x))}$

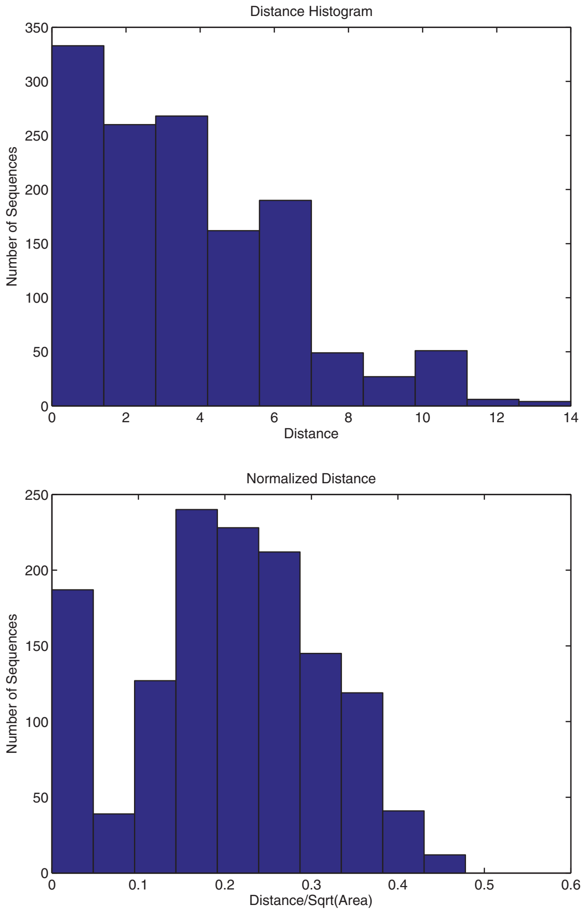

Fig. 5. (Top) Histogram of $r(x)$ in the 2D (AU, CG) model. (Bottom) Histogram of $r(x)/\sqrt{\mathrm{Area}(\mathcal{N}(x))}$

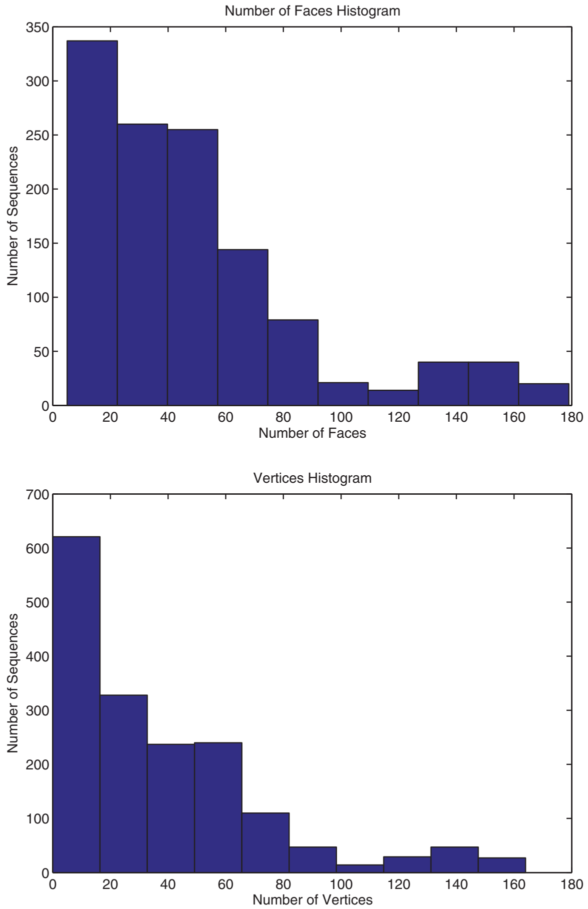

Fig. 6. (Top) Histogram of the number of faces of the 3D polytope. (Bottom) Histogram of the number of vetices of the 3D polytope

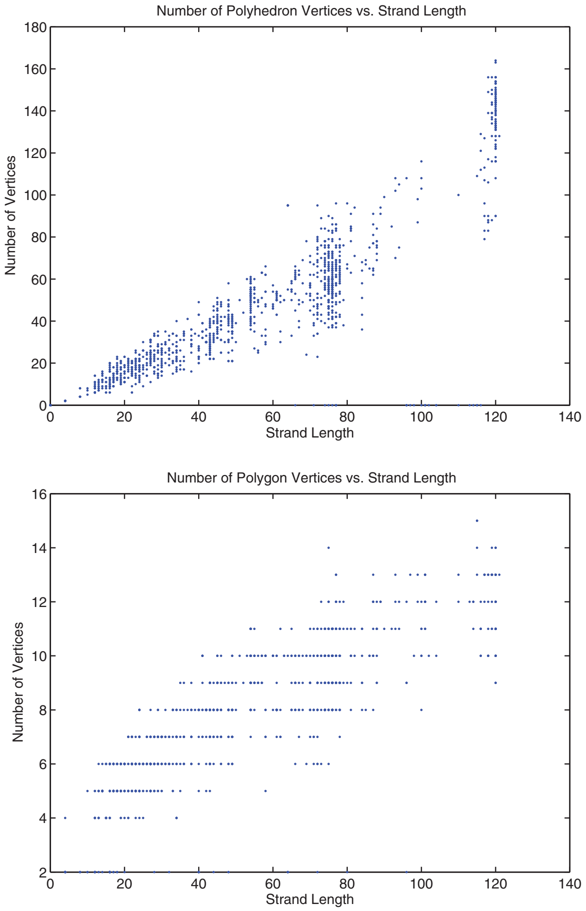

Fig. 7. (Top) Scatter plot of the number of vertices versus strand length in the 3D model. (Bottom) Scatter plot of the number of vertices versus strand length in the 2D model

distance is between 1 and 2, and for 171 (13%) strands, it is more than 2, and just for 11 (~1%) sequences, this distance is above 5. As it is clear from Figure 4, the largest *r* is 8 bp in the 3D model. The second plot in Figure 4 shows the normalized distance histogram. The third root of the polytope's volume is used as the normalization factor. For the same set of strands, the observed feature vector of only 176 strands lies on the boundary of the 2D Newton polygon. For 327 sequence, this distance is larger than 5. As you can see in Figure 5, the maximum distance between an observed feature vector and the polygon is 14 in the 2D model. The second plot in Figure 5 shows the normalized distance histogram. The square root of the polygon's area is used as the normalization factor.

Figure 6 shows the histogram of the number of faces of the 3D Newton polytope; 79 strands have a polytope with less than 10 faces. The minimum number of faces is five, and 118 strands yield a polytope with more than 100 faces. For 211 strands, the polytope has more than 10 but no more than 20 faces. Only 47 strands have more than 150 faces. The second plot in Figure 6 depicts the histogram of the number of vertices of the Newton polytope. Figure 7 shows the relation between the length and number of vertices for the 2D and 3D models. The number of vertices in 2D is not more than 15, but that number increases to about 180 in 3D.

6 CONCLUSION AND FUTURE WORK

We introduced the notion of learnability of the parameters of an energy model as a measure of its inherent capability. We derived a necessary condition for the learnability and gave a dynamic programming algorithm to assess it. Our algorithm computes the convex hull of the feature vectors of all feasible structures in the ensemble of a given input sequence. Also, that convex hull coincides with the *Newton polytope* of the partition function as a polynomial in transformed energy parameters.

Our theory applied to a simple energy model that counts A-U, C-G and G-U base pairs separately revealed that about half of chosen known structures could potentially be predicted using this simple energy model. For another one-fifth, the necessary condition is barely violated, which suggests that augmenting this energy model with more features is expected to satisfy the necessary condition for them. The condition is severely violated for 13% of sequences, which will be the subject of future investigation. The twilight zone (~10%) is also interesting and requires deeper examination.

The Newton polytope lies in the core of computer algebra for solving polynomial equations. Therefore, we envision applications of our RNA Newton polytope in symbolic estimation of energy parameters. Our algorithm has a worst case exponential complexity due to the computation of the convex hull. However, we envision efficient approximation of the Newton ploytope through topology-preserving dimensionality reduction methods. Applying our method to increasingly complex models as well as design of energy models using our analysis method will be pursued in the future. Sufficient conditions for the learnability, and also assessing the generalization power of an energy model, remain for future work.

Conflict of Interest: none declared.

REFERENCES

Andronescu,M. *et al.* (2007) Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics*, **23**, 19–28.

Andronescu,M. *et al.* (2008) RNA STRAND: the RNA secondary structure and statistical analysis database. *BMC Bioinformatics*, **9**, 340.

Andronescu,M. *et al.* (2010) Computational approaches for RNA energy parameter estimation. *RNA*, **16**, 2304–2318.

Barber,C.B. *et al.* (1996) The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, **22**, 469–483.

Bartel,D.P. (2004) MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, **116**, 281–297.

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

Brantl,S. (2002) Antisense-RNA regulation and RNA interference. *Biochim. Biophys. Acta*, **1575**, 15–25.

Burge,S.W. *et al.* (2013) Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res.*, **41**, D226–D232.

Chitsaz,H. *et al.* (2009a) biRNA: fast RNA-RNA binding sites prediction. In: *Workshop on Algorithms in Bioinformatics (WABI)*. Vol. 5724, LNBI, Springer-Verlag, Berlin, Heidelberg.

Chitsaz,H. *et al.* (2009b) A partition function algorithm for interacting nucleic acid strands. *Bioinformatics*, **25**, i365–i373.

Dewey,C.N. *et al.* (2006) Parametric alignment of Drosophila genomes. *PLoS Comput. Biol.*, **2**, e73.

Dirks,R.M. and Pierce,N.A. (2003) A partition function algorithm for nucleic acid secondary structure including pseudoknots. *J. Comput. Chem.*, **24**, 1664–1677.

Do,C.B. *et al.* (2006) CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, **22**, 90–98.

Eddy,S.R. and Durbin,R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res.*, **22**, 2079–2088.

Emiris,I. (1994) Sparse elimination and applications in kinematics. Ph.D. Thesis, UC Berkeley, Berkeley, CA.

Emiris,I.Z. and Canny,J.F. (1995) Efficient incremental algorithms for the sparse resultant and the mixed volume. *J. Symbolic Comput.*, **20**, 14–19.

Gibson,D.G. *et al.* (2010) Creation of a bacterial cell controlled by a chemically synthesized genome. *Science*, **329**, 52–56.

Gottesman,S. (2005) Micros for microbes: non-coding regulatory RNAs in bacteria. *Trends Genet.*, **21**, 399–404.

Hannon,G.J. (2002) RNA interference. *Nature*, **418**, 244–251.

Mathews,D. *et al.* (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.

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

Nussinov,R. *et al.* (1978) Algorithms for loop matchings. *SIAM J. Appl. Math.*, **35**, 68–82.

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

Rivas,E. *et al.* (2012) A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. *RNA*, **18**, 193–212.

Seeman,N. (2005) From genes to machines: DNA nanomechanical devices. *Trends Biochem. Sci.*, **30**, 119–125.

Seeman,N.C. and Lukeman,P.S. (2005) Nucleic acid nanostructures: bottom-up control of geometry on the nanoscale. *Rep. Prog. Phys.*, **68**, 237–270.

Siegfried,N.A. and Bevilacqua,P.C. (2009) Thinking inside the box: designing, implementing, and interpreting thermodynamic cycles to dissect cooperativity in RNA and DNA folding. *Methods Enzymol.*, **455**, 365–393.

Simmel,F. and Dittmer,W. (2005) DNA nanodevices. *Small*, **1**, 284–299.

Storz,G. (2002) An expanding universe of noncoding RNAs. *Science*, **296**, 1260–1263.

Tinoco,I. *et al.* (1973) Improved estimation of secondary structure in ribonucleic acids. *Nat. New Biol.*, **246**, 40–41.

Venkataraman,S. *et al.* (2007) An autonomous polymerization motor powered by DNA hybridization. *Nat. Nanotechnol.*, **2**, 490–494.

Wagner,E. and Flardh,K. (2002) Antisense RNAs everywhere? *Trends Genet.*, **18**, 223–226.

Waterman,M.S. and Smith,T.F. (1978) RNA secondary structure: a complete mathematical analysis. *Math. Biosci.*, **42**, 257–266.

Yin,P. *et al.* (2008) Programming DNA tube circumferences. *Science*, **321**, 824–826.

Yoshpe,M. (2006) *Distance from a point to a 2D polygon*. http://www.mathworks.com/matlabcentral/fileexchange/12744-distance-from-a-point-to-polygon (14 May 2013, date last accessed).

Zakov,S. *et al.* (2011) Rich parameterization improves RNA structure prediction. In: Bafna,V. and Sahinalp,S. (eds.) *Proceedings of the 15th Annual International Conference on Research in Computational Molecular Biology*, Vol. 6577, Lecture Notes in Computer Science. Springer, Berlin-Heidelberg, pp. 546–562.

Zamore,P.D. and Haley,B. (2005) Ribo-gnome: the big world of small RNAs. *Science*, **309**, 1519–1524.

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