

An efficient algorithm to compute the landscape of locally optimal
RNA secondary structures with respect to the Nussinov-Jacobson
energy model

P. Clote

Department of Biology

Department of Computer Science (courtesy appt.)

Higgins 355, Boston College, Chestnut Hill, MA 02467, USA

Tel: 617 552 1332, Fax: 617 552 2011, clote@bc.edu.

18 July 2004

Running Title: Efficiently computing the landscape of RNA structures

Key words: RNA, secondary structure, energy landscape, local optimum.

Abstract

We make a novel contribution to the theory of biopolymer folding, by developing an efficient algorithm to compute the number of locally optimal secondary structures of an RNA molecule, with respect to the Nussinov-Jacobson energy model. Additionally, we apply our algorithm to analyze the folding landscape of selenocysteine insertion sequence (SECIS) elements from A. Böck (personal communication), hammerhead ribozymes from Rfam [17], and tRNAs from Sprinzl’s database [30]. It had previously been reported that tRNA has lower minimum free energy than random RNA of the same compositional frequency [7, 24], although the situation is less clear for mRNA [27, 33, 8],^a which plays no structural role. Applications of our algorithm extend knowledge of the energy landscape differences between naturally occurring and random RNA.

Given an RNA molecule a_1, \dots, a_n and an integer $k \geq 0$, a k -locally optimal secondary structure S is a secondary structure on a_1, \dots, a_n which has k fewer base pairs than the maximum possible number, yet for which no basepairs can be added without violation of the definition of secondary structure (e.g. introducing a pseudoknot). Despite the fact that the number $\text{numStr}(k)$ of k -locally optimal structures for a given RNA molecule in general is exponential in n , we present an algorithm running in time $O(n^4)$ and space $O(n^3)$, which computes $\text{numStr}(k)$ for each k . Structurally important RNA, such as SECIS elements, hammerhead ribozymes, tRNA, all have a markedly smaller number of k -locally optimal structures than that of random RNA of the same dinucleotide frequency, for small and moderate values of k . This suggests a potential future role of our algorithm as a tool to detect noncoding RNA genes.

^aSeffens and Digby [27] report that mRNA has lower folding energy than that of random RNA having the same mononucleotide frequency, while Workman and Krogh [33] report the opposite conclusion with respect to random RNA having the same dinucleotide frequency.

1 Motivation

In 1973 C.B. Anfinsen postulated that a biopolymer’s native state is in a global free energy minimum, based on experimental work [1] where bovine pancreatic ribonuclease was denatured (disulfide bonds broken) by the addition of urea, while the protein reconstituted its original disulfide bonding pattern upon the subsequent removal of the denaturant. Even earlier, in 1968, C. Levinthal [20] raised the question of how macromolecules can fold into their native state relatively rapidly (milliseconds to seconds), even though the number of possible conformations is exponential. More recently, Bryngelson and Wolynes [3] applied the theory of spin glasses to explain the dynamics of protein folding. Trying to understand the landscape and kinetics of biopolymer (RNA, protein) folding has led to many motivating questions. Are special folding pathways necessary? Is there an energy funnel? What distinguishes naturally occurring macromolecules from random macromolecules? Can one computationally detect or engineer macromolecular switches? Can one engineer artificial macromolecules with even better folding properties?

Monte-Carlo simulations of Šali et al. [25, 26], as reported in *Nature* (1994), suggested that proteins fold rapidly if and only if there is a large difference Δ between the global energy minimum of the native state and the energy of the next misfolded state. See the article by P. Clote [5] for an application of Sinclair’s method [29] of rapidly folding Markov chains to explain the result of Šali et al. [25, 26] in terms of the second eigenvalue of the underlying Markov chain transition matrix; also see E. Shakhnovich’s survey [28] on protein folding, thermodynamics and kinetics.

Although computing the native state for even the simplest HP (hydrophobic-polar) protein model [4] is *NP*-complete [9, 2], computing the secondary structure of RNA is tractable [36, 34, 19, 13], using the $O(n^3)$ time Zuker algorithm, as implemented in mFold or Vienna RNA package.

In [10] Cupal et al. designed a $O(n^3m^2)$ time and $O(n^2m)$ space algorithm to compute the *density of states* of RNA secondary structures for a given RNA molecule. Here, n is the length of the RNA sequence, and m is the number of energy bins. The left image of Figure 1 illustrates the density of states, which graphs the proportion of secondary structures in the k -th energy bin, as a function of k . The right image of Figure 1 displays the energy spectrum with a prominent energy gap Δ between the native state and next lowest energy structure.

In [15, 16] Flamm et al. investigated energy landscapes near the global energy minimum for RNA secondary structures. In contrast to the assertions of [25, 26] for protein structure, they found that mean first passage time for RNA to reach the native state (i.e. folding time) appeared to be correlated with the existence of *bottlenecks in folding pathways* from local optima to the global optimum, rather than the existence of a large energy gap Δ between the energy of the optimum secondary structure and that of the next lowest suboptimum.

The algorithms of Cupal et al. [10] and Flamm et al. [15, 16] use the so-called Turner energy model [21] for RNA with experimentally derived energy tables for stacked basepairs, hairpin loops, special tetraloops, bulges and internal loops, as well as an affine approximation for multiloops. In contrast, the algorithms presented in this paper use the simpler Nussinov-Jacobson energy model [23], where an optimal structure has the maximum number of base pairs. With substantial additional work, it may be possible to lift our current algorithm to handle the full energy model [21]; however, given the complexity of our main algorithm, it seems fully justified to consider the simple Nussinov-Jacobson energy model in this paper.

In [11, 12], Y. Ding and C.E. Lawrence describe how to *sample* RNA secondary structures for a given nucleotide sequence, according to the Boltzmann probability [22]. A striking application of their work to RNAi, as given on the web server Sfold, is to predict potential target regions $a_i, a_{i+1}, a_{i+2}, a_{i+3}$ of mRNA which satisfy Tuschl’s rules [18, 32] and have high Boltzmann probability of not being basepaired. Though this work does sample low energy secondary structures, it

does not give any information about k -locally optimal structures.

In this paper, by a radically different approach explained in Algorithm 11, we design and implement an $O(n^4)$ time and $O(n^3)$ space procedure, which computes the number $\text{numStr}(k)$ of k -locally optimal secondary structures of a given RNA molecule with respect to the Nussinov-Jacobson energy model [23]. As mentioned above, Cupal et al. [10] designed a $O(n^3m^2)$ time and $O(n^2m)$ space algorithm to compute the density of states of RNA secondary structures for the full energy model with Turner’s rules [21], where n is the length of the RNA sequence, and m is the number of energy bins. From their algorithm, however, one cannot obtain any information about how many of these structures are actually k -locally optimal in our sense, i.e. that no basepairs can be added without violating conditions of secondary structure. Additionally, the work of Flamm et al. [15, 16] allows one to explore the energy landscape near the native state; however, it would require exponential time, as opposed to our algorithm’s time $O(n^4)$, to explore the entire energy landscape. Moreover, the algorithm of Flamm et al. gives no information about k -locally optimal structures, as defined in this paper.

Applications of our algorithm clearly indicate a fundamentally different energy landscape for structurally important classes of RNA, than that of random RNA of the same dinucleotide frequency. See Figures 3, 4, 5 and 6.

In [14], Evers and Giegerich designed an $O(n^3)$ algorithm to compute the number of *saturated* secondary structures for a given RNA molecule, thus answering a question of [35]. Here, a secondary structure is *saturated* if “stacking regions extend maximally in both directions” and there are no isolated base pairs (i.e. not adjacent to a stacked base pair). While the notion of *saturated* is related to the notion of k -locally optimal defined in Definition 2, the two concepts are distinct. There exist saturated structures which are not optimal or k -locally optimal, and there exist optimal structures which are not saturated. It seems possible, however, that by adapting techniques given in this paper, one could develop an algorithm to compute the number of k -saturated secondary structures, where the latter might designate a secondary structure which has k fewer base pairs than that of a saturated structure.

2 Introduction

A *secondary structure* for an RNA sequence $a = a_1 \cdots a_n \in \{A, C, G, U\}^n$ is an expression $s = s_1 \cdots s_n$ involving dot, left and right parenthesis, which is well-balanced, such that nucleotides corresponding to matching parentheses are either Watson-Crick complements or GU wobble pairs. We say that a secondary structure has threshold θ , if hairpin loops have at least θ unpaired bases.

More formally, we define a secondary structure as follows.

Definition 1 *A secondary structure S on RNA sequence a_1, \dots, a_n is defined to be a set of ordered pairs corresponding to basepair positions, which satisfies the following requirements.*

1. Watson-Crick or GU wobble pairs: *If (i, j) belongs to S , then pair (a_i, a_j) must be one of the following canonical basepairs: (A, U) , (U, A) , (G, C) , (C, G) , (G, U) , (U, G) .*
2. Threshold requirement: *If (i, j) belongs to S , then $j - i > \theta$.*
3. Nonexistence of pseudoknots: *If (i, j) and (k, ℓ) belong to S , then it is not the case that $i < k < j < \ell$.*
4. No base triples: *If (i, j) and (i, k) belong to S , then $j = k$; if (i, j) and (k, j) belong to S , then $i = k$.*

With respect to a fixed energy model, either simply maximizing the number of basepairs (Nussinov-Jacobson algorithm [23, 6]) or by more realistically computing the stabilizing energy of stacked basepairs along with destabilizing energy of hairpin loops, bulges, internal loops and multiloops (Zuker's algorithm [36, 34]), one computes an optimal secondary structure. Figure 2 illustrates the secondary structure of the the hammerhead ribozyme with Rfam accession number 23AF170503. Further details about RNA secondary structure, etc., may be found in the the text of P. Clote and R. Backofen [6].

3 Nussinov-Jacobson energy landscape of an RNA molecule

We begin by recalling the Nussinov-Jacobson algorithm [23], which computes a secondary structure by dynamic programming which has the maximum number of base pairs. Though this may appear very simple, the Nussinov-Jacobson algorithm, originally applied to compute the optimal secondary structure of the bacteriophage MS2, laid the foundations for later dynamic programming algorithms [36, 34, 19, 13] using a more realistic energy model.

Similar to the situation in sequence alignment, the maximum number of base pairs is uniquely defined; however there may be several secondary structures having this maximum number.

Definition 2 Let a_1, \dots, a_n be an RNA sequence, and S a secondary structure on a_1, \dots, a_n . In the following, indices i, j satisfy $1 \leq i < j \leq n$.

1. Define the interval $[i, j]$ to denote the set $\{i, i + 1, \dots, j\}$.
2. Define the restriction $S_{[i,j]}$ to be $S \cap \{(x, y) : i \leq x < y \leq j, (x, y) \in S\}$.
3. Define S to be optimal on $[i, j]$ if $S_{[i,j]}$ has the maximum possible number of base pairs; i.e. there is no secondary structure T , with $S_{[i,j]} \subseteq T_{[i,j]}$ and $|S_{[i,j]}| < |T_{[i,j]}|$. If the context permits, or especially if $i = 1$ and $j = n$, we may simply say that S is optimal.
4. Define S to be locally optimal if for all $i < x < y < j$, $S \cup \{(x, y)\}$ violates the definition of secondary structure, as given in Definition 1.
5. For $1 \leq k \leq n$, define S to be k -locally optimal on $[i, j]$ if S is locally optimal on $[i, j]$ and $|S_{[i,j]}| + k = |T_{[i,j]}|$, where T is an optimal secondary structure on $[i, j]$. If the context permits, or especially if $i = 1$ and $j = n$, we may simply say that S is k -locally optimal. Note that k -locally optimal secondary structures, especially for small values of k , are potential kinetic traps in the folding process.
6. Define $BP(i, j, 0) = |T_{[i,j]}|$, where T is an optimal secondary structure on $[i, j]$. For $1 \leq k \leq n$, define $BP(i, j, k) = BP(i, j, 0) - k$ if there exists a k -locally optimal secondary structure on $[i, j]$, and otherwise $BP(i, j, k)$ is undefined. Note that BP stands for number of base pairs; thus $BP(i, j, k)$ is the number of base pairs in a k -locally optimal secondary structure on the interval $[i, j]$.
7. Define the boolean-valued function $bp(i, j)$ by

$$bp(i, j) = \begin{cases} 1 & \text{if } a_i, a_j \text{ can basepair} \\ 0 & \text{else.} \end{cases}$$

The following algorithm provides a method to compute the maximum number $M(1, n)$ of base pairs in a secondary structure on a_1, \dots, a_n .

Algorithm 3 (Nussinov-Jacobson [23]) Given RNA nucleotide sequence a_1, \dots, a_n , recall that for $1 \leq i < j \leq n$, $bp(i, j)$ is defined by $bp(i, j) = 1$ if i, j can basepair, else 0. Define $M_{i,j} = 0$ if $j - i \leq \theta$, and otherwise

$$M_{i,j} = \max \left\{ M_{i,j-1}, \max_{i \leq \ell < j-\theta} \{ bp(a_\ell, a_j) \cdot (1 + M_{i,\ell-1} + M_{\ell+1,j-1}) \} \right\}.$$

Then $M_{i,j}$ equals the maximum number of basepairs for any secondary structure on a_i, \dots, a_j , for all $1 \leq i \leq j \leq n$.

The Nussinov-Jacobson algorithm uses dynamic programming to compute the matrix M in time $O(n^3)$ and space $O(n^2)$, and additionally uses a linear time backtracking method to output an optimal secondary structure on a_1, \dots, a_n having $M(1, n)$ many base pairs. Details of this algorithm and its implementation are given in the text by P. Clote and R. Backofen [6].

The Nussinov-Jacobson algorithm treats three separate cases, when considering $M_{i,j}$. In case 1, j is considered to (potentially) basepair with i , while in case 2, j is considered to (potentially) basepair with an intermediate $i < k < j$. In case 3, j is considered not to basepair with any position in the interval $[i, j]$. We follow this schema in both the heuristic Algorithm 5 as well as the main Algorithm 11. First, we require several definitions.

Definition 4 Define the locally optimal energy spectrum, usually abbreviated to energy spectrum, of a_1, \dots, a_n by $\{k : BP(1, n, k) \text{ is defined}\}$; i.e. the set of integers k for which there exists a k -locally optimal secondary structure on a_1, \dots, a_n . Additionally, for $1 \leq i < j \leq n$ and $0 \leq k \leq n$, define $numStr(i, j, k)$ to be the number of distinct k -locally optimal secondary structures on a_i, \dots, a_j . By energy landscape of RNA nucleotide sequence a_1, \dots, a_n , we mean the function $NumStr : \{0, \dots, n\} \rightarrow \mathbf{N}$, defined by $NumStr(k) = numStr(1, n, k)$. Finally, define the locally optimal state density function

$$\rho(k) = \frac{NumStr(k)}{\sum_{\ell} NumStr(\ell)}.$$

Our definition of (locally optimal) state density gives the proportion of k -locally optimal structures with respect to the total number of locally optimal structures, and hence is analogous to the state density function of [10] illustrated in Figure 1.^b Similarly, our definition of *locally optimal energy spectrum* is motivated by the notion of energy spectrum, as given in [25, 26] and [10].

To clarify the previous definition, here is a small example. Suppose that $\theta = 3$, and that a_1, \dots, a_n is AAAAUUUUU. The maximum number of basepairs is 3. There is one 0-locally optimal structure (with 3 basepairs) given by (((...))). There are 12 1-locally optimal structures, given by

AAAAUUUUU	AAAAUUUUU	AAAAUUUUU	AAAAUUUUU
..((...))	.((...))	.((...))	((...)..)
(..(...))	.((...).)	(. (...))	(. (...)).
((...).)	((...)).	((...)).	((...))...

Finally, there are three 2-locally optimal secondary structures for AAAAUUUUU, given by

AAAAUUUUU	AAAAUUUUU	AAAAUUUUU
(...)....	(...)...	... (...)

Note that all of the previously given structures have positive energy, according to the Turner energy model [21], and so the empty structure, with minimum free energy of 0, would be that occurring in nature. The previous example illustrates as well the fact that it can happen that the free energy of secondary structure $S \cup \{(x, y)\}$ is actually higher than that of S ; i.e. a structure

^bNote that whereas [10] considers the space of all secondary secondary structures, we consider only locally optimal secondary structures. Thus to be pedantic, in place of density, we should write *relative density*; we nevertheless continue to use the term density in this paper. It is a routine programming exercise to compute the total number of (not necessarily k -locally optimal) secondary structures for a given RNA nucleotide sequence. Using such an algorithm, along with Algorithm 11, we could give a density of states for k -locally optimal structures with respect to all secondary structures.

having more base pairs does not guarantee that its energy will be lower (though this is usually the case).

The idea of the both Algorithm 5 and 11 is to compute $BP(i, j, k)$ and $numStr(i, j, k)$ for all $0 \leq k \leq n$ and all $1 \leq i < j \leq n$, where we consider all i, j with $j - i = d$ before proceeding to consider all i, j with $j - i = d + 1$. As in any dynamic programming method, our implementation stores previously computed values $BP(i, j, k)$ and $numStr(i, j, k)$ for all i, j with $j - i = d$, so that these values are available when considering i, j with $j - i = d + 1$. Finally, note that we need only consider values $k \leq \lfloor n/2 \rfloor$, since there are clearly at most $n/2$ potential base pairs in a secondary structure on a_1, \dots, a_n . For simplicity, our pseudocode ignores this point.

In both Algorithm 5 and 11, when considering (for instance) the case that j basepairs with an intermediate $i < k < j$, we consider all possible values $0 \leq \ell, \ell' \leq n$ for which $BP(i, k - 1, \ell)$ and $BP(k + 1, j - 1, \ell')$ are defined. It then follows that $BP(i, j, k)$ must include a contribution $1 + BP(i, k - 1, \ell) + BP(k + 1, j - 1, \ell')$ for all such ℓ, ℓ' . Nonexistence of pseudoknots, guaranteed by Definition 1, ensures that the problem of computing $BP(i, j, k)$ can be thus decomposed into the above-mentioned three cases, as in the Nussinov-Jacobson algorithm.

In both Algorithm 5 and 11, we use the notation $numStr^*(i, j, k)$ resp. $BP^*(i, j, k)$ to denote the number of k -locally optimal structures on a_i, \dots, a_j resp. the number of base pairs in every k -locally optimal structure on a_i, \dots, a_j , as calculated by the algorithm. In the heuristic Algorithm 5, both values may be underestimations of the real values, as defined in Definition 2; i.e. $numStr^*(i, j, k) \leq numStr(i, j, k)$ and $BP^*(i, j, k) \leq BP(i, j, k)$. Later, for our exact Algorithm 11, we show that the values computed by the algorithm are the correct values, as defined Definition 2; i.e. $numStr^*(i, j, k) = numStr(i, j, k)$ and $BP^*(i, j, k) = BP(i, j, k)$.

Algorithm 5 (Heuristic for energy spectrum) *Given RNA nucleotide sequence a_1, \dots, a_n , the following pseudocode computes a subset of the energy spectrum of a_1, \dots, a_n , in that any value k generated below does in fact belong to the spectrum. Moreover, if the algorithm defines $BP^*(i, j, k)$, then $BP(i, j, k)$ is defined, and $numStr^*(i, j, k) \leq numStr(i, j, k)$.*

```

run Nussinov-Jacobson to compute  $BP^*(i, j, 0)$  and
 $numStr^*(i, j, 0)$  for  $1 \leq i < j \leq n$ 
for  $1 \leq i \leq j \leq n$ 
  for  $k = 1$  to  $n$ 
    initialize  $BP^*(i, j, k)$  to be undefined and  $numStr^*(i, j, k) = 0$ 
for  $d = \theta$  to  $n - 1$ 
  for  $i = 0$  to  $n - 1$ 
     $j = i + d$  // skip if  $j > n$ 
    for  $r = i$  to  $j - \theta - 1$ {
      if  $r, j$  basepair{
         $jCanBasePair = TRUE$ 
        if  $r = i$  //CASE 1:  $j$  basepairs with  $i$ 
          for all  $0 \leq \ell \leq BP^*(i, j, 0)$ 
            if  $BP^*(i + 1, j - 1, \ell)$  is defined
               $k = BP^*(i, j, 0) - [1 + BP^*(i + 1, j - 1, \ell)]$ 
               $BP^*(i, j, k) = 1 + BP^*(i + 1, j - 1, \ell)$ 
               $numStr^*(i, j, k) = 1 + numStr^*(i + 1, j - 1, \ell)$ 

```

```

if  $i < r < j$  //CASE 2:  $j$  basepairs with intermediate  $r$ 
  for all  $0 \leq \ell \leq BP^*(i, r - 1, 0)$ 
    for all  $0 \leq \ell' \leq BP^*(r + 1, j - 1, 0)$ 
      if  $BP^*(i, r - 1, \ell), BP^*(r + 1, j - 1, \ell')$  are defined
         $bp0 = 1 + BP^*(i, r - 1, \ell) + BP^*(r + 1, j - 1, \ell')$ 
         $k = BP^*(i, j, 0) - bp0$ 
         $BP^*(i, j, k) = 1 + BP^*(i, r - 1, \ell) + BP^*(r + 1, j - 1, \ell')$ 
         $numStr^*(i, j, k) = 1 + numStr^*(i, r - 1, \ell) + numStr^*(r + 1, j - 1, \ell')$ 
      }
    }
  }
if  $jCanBasePair = FALSE$ 
  //CASE 3:  $j$  cannot basepair in interval  $[i, j]$ 
  for all  $0 \leq \ell \leq BP^*(i, r - 1, 0)$ 
     $k = BP^*(i, j, 0) - BP^*(i, j - 1, \ell)$ 
     $BP^*(i, j, k) = BP^*(i, j - 1, \ell)$ 
     $numStr^*(i, j, k) = numStr^*(i, j - 1, \ell)$ 
  }
for  $k = 1$  to  $n$ 
  if  $BP^*(1, n, k)$  is defined
    print  $k, numStr^*(k)$ 

```

In addition to outputting that k belongs to the energy spectrum, as well as (a lower bound for) the number of k -locally optimal secondary structures, our implementation includes a backtracking procedure to output an example k -locally optimal secondary structure. For this, additional details necessary for a traceback are implemented, in order to decide which of Case 1,2,3 is obtained when considering i, j, k ; such details, omitted for brevity, are important for the backtracking algorithm (not shown). By using either `mfold` [36] or `RNAeval` [19], one can additionally compute the Turner [21] free energy associated with the example k -locally optimal structure, as illustrated in Figure 2.

The pseudocode in Algorithm 5 is only a heuristic, since it can indeed happen that though j can basepair with some $i \leq r \leq j - \theta - 1$, there is a k -locally suboptimal structure T on $[i, j - 1]$ where r is not “visible” to j ; i.e. there is a base pair $(x, y) \in T$, with $i \leq x < r < y < j$. In such a case, T , interpreted as a structure on $[i, j]$ is indeed locally optimal, although T may be k' -locally optimal on $[i, j]$, for some $k' \neq k$. Thus it can happen that $BP^*(i, j - 1, k)$ is not defined, whereas $BP^*(i, j, k)$ is defined, and $numStr^*(i, j, k) < numStr(i, j, k)$. This leads us naturally to consider how to parametrize the notion of *visibility*, as defined in Definitions 7 and 8.

Recall that the restriction $S_{[i,j]}$ of secondary structure S to interval $[i, j]$, where $1 \leq i \leq j \leq n$, is defined by

$$S_{[i,j]} = \{(x, y) : i \leq x \leq y \leq j, (x, y) \in S\}.$$

For brevity, instead of saying that S is a secondary structure on a_i, \dots, a_j , we will usually say that S is a secondary structure on $[i, j]$, where the nucleotide sequence a_1, \dots, a_n is fixed.

Definition 6 For RNA nucleotide sequence a_1, \dots, a_n , and for $1 \leq i \leq x < y \leq j \leq n$, a basepair (x, y) is an exterior basepair in $[i, j]$ if there is no basepair $(x', y') \in S$ where $i \leq x' < x$ and $y < y' \leq j$. Similarly a position $z \in [i, j]$ is interior to a basepair $(x, y) \in S$, if $x \leq z \leq y$. The position z is free or visible in $[i, j]$ if it is not interior to any basepair of S in interval $[i, j]$.

Consideration of why Algorithm 5 fails to compute the entire energy spectrum and why it gives only a lower bound on $numStr(i, j, k) \leq numStr(i, j, k)$ leads to the critical notions of $VisNuc(S, m)$, $VisPos(S, i, j)$, and the correct algorithm, as expressed in Theorem 10.

Definition 7 Let S be a secondary structure on RNA sequence a_i, \dots, a_j , and let θ denote the minimum number of unpaired bases in a hairpin loop. Define

$$VisNuc(S) = \{a_z : \text{for all } (x, y) \in S [(z < x \text{ or } z > y) \text{ and } z < j - \theta]\}.$$

In words, $VisNuc(S)$ is the set of *visible* nucleotides $x \in \{A, C, G, U\}$, which appear at a position z , where $i \leq z < j - \theta$, and which are exterior to all basepairs of S .^c We additionally need to consider which of the positions $j - \theta, \dots, j$ in the interval $[i, j]$ is *free* on $[i, j]$, i.e. not within the scope of a basepair of $S_{[i, j]}$. This is done as follows.

Definition 8 With the same assumptions as in the previous definition, define the *visible position* $VisPos(S, i, j) = b$, where $0 \leq b \leq \theta + 1$, and b is the greatest value in $[0, \theta + 1]$, such that for all $0 \leq x < b$, $j - x$ is visible.^d

Note that $VisPos(i, j, k) = 0$ if and only if none of $a_{j-\theta}, \dots, a_j$ is visible, whereas $Vis(i, j, k) = \theta + 1$ if and only if each of $a_{j-\theta}, \dots, a_j$ is visible.

The rather technical definitions of both $VisNuc$ and $VisPos$ are crucial in Algorithm 11, and its proof of correctness Theorem 10, in computing the energy spectrum of RNA nucleotide sequence a_1, \dots, a_n . For this reason, we give several examples. Consider RNA nucleotide sequence AUUCCGGCA, and let $\theta = 1$.

1. The secondary structure

AUUCCGGCA
(.)(.)...

is $\{G\}$, 2-visible.

2. The secondary structure

AUUCCGGCA
... (.) ...

is $\{A, G, U\}$, 2-visible.

3. The secondary structure

AUUCCGGCA
.... (.) ..

is $\{A, C, U\}$, 2-visible.

4. The secondary structure

AUUCCGGCA
(.).. (.)

^cN.B. that $VisNuc(S) \subseteq \{A, C, G, U\}$ is a set, not list.

^dRecall that a position k is visible in $[i, j]$ if k is not interior to any basepair in interval $[i, j]$.

is $\{C\}$, 1-visible.

5. The secondary structure

AUCCGGCA
 ..(.(.))

is $\{A, U\}$, 0-visible.

Definition 9 Given integers $1 \leq i \leq j \leq n$, $k \geq 0$, $s \subseteq \{A, C, G, U\}$, $0 \leq b \leq \theta + 1$, define $\text{SecStr}(i, j, k, s, b)$ to be the set of secondary structures S on a_i, \dots, a_j which are k -locally optimal, where $\text{VisNuc}(S, \theta) = s$ and $\text{VisPos}(S, i, j) = b$, and let $\text{numStr}(i, j, k, s, b) = |\text{SecStr}(i, j, k, s, b)|$. Additionally, let $\text{BP}(i, j, k, s, b)$ denote the number of base pairs in a k -locally optimal s, b -visible secondary structure on a_i, \dots, a_j . Instead of writing $S \in \text{SecStr}(i, j, k, s, b)$, we may at times say that S is a k -locally optimal (s, b) -visible secondary structure on $[i, j]$.

Recall that $\text{numStr}(i, j, k)$ was defined in Definition 4. Note that

$$\text{numStr}(i, j, k) = \sum_{s, b} \text{numStr}(i, j, k, s, b)$$

and

$$\text{BP}(i, j, k) = \sum_{s, b} \text{BP}(i, j, k, s, b)$$

where the sum is over all $s \subseteq \{A, C, G, U\}$ and $0 \leq b \leq \theta + 1$.

Theorem 10 Given RNA nucleotide sequence a_1, \dots, a_n , let $\text{BP}^*(\cdot)$ and $\text{numStr}^*(\cdot)$ be the functions computed in Algorithm 11 for the number of base pairs and number of structures. Then $\text{BP}^*(i, j, k, s, b) = \text{BP}(i, j, k, s, b)$ and $\text{numStr}^*(i, j, k, s, b) = \text{numStr}(i, j, k, s, b)$ for all $1 \leq i \leq j \leq n$, $0 \leq k \leq n$, $s \subseteq \{A, C, G, U\}$ and $0 \leq b \leq \theta + 1$, where $\text{BP}(i, j, k, s, b)$ and $\text{numStr}(i, j, k, s, b)$ are defined in Definition 9. Hence Algorithm 11 correctly computes the number of k -locally optimal structures on interval $[i, j]$, as well as the number of base pairs in each k -locally optimal structure on interval $[i, j]$; i.e.

$$\begin{aligned} \text{numStr}(i, j, k) &= \sum_{s, b} \text{numStr}(i, j, k, s, b) = \sum_{s, b} \text{numStr}^*(i, j, k, s, b) = \text{numStr}^*(i, j, k) \\ \text{BP}(i, j, k) &= \sum_{s, b} \text{BP}(i, j, k, s, b) = \sum_{s, b} \text{BP}^*(i, j, k, s, b) = \text{BP}^*(i, j, k) \end{aligned}$$

where $\text{numStr}(i, j, k)$ and $\text{BP}(i, j, k)$ are defined in Definition 2. Moreover, the algorithm's runtime is $O(n^4)$ and its space requirement is $O(n^3)$. If loops in Algorithm 11 involving locally optimality level k are bound by m , then the runtime is $O(mn^3)$ and space is $O(mn^2)$.

PROOF. Since the time and space bound are obvious by looking at the structure of the pseudocode, only the first assertion needs to be proved. Before beginning the proof, note that since the Nussinov-Jacobson algorithm correctly computes the number of optimal (i.e. 0-locally optimal) structures, functions mentioned in the statement of the theorem are correctly computed for $k = 0$; i.e. $\text{BP}^*(i, j, 0) = \text{BP}(i, j, 0)$, $\text{BP}^*(i, j, 0, s, b) = \text{BP}(i, j, 0, s, b)$, $\text{numStr}^*(i, j, 0) = \text{numStr}(i, j, 0)$, $\text{numStr}^*(i, j, 0, s, b) = \text{numStr}(i, j, 0, s, b)$.

For any i, j, k, s, b , let $SecStr^*(i, j, k, s, b)$ be the collection of structures accounted for in the term $numStr^*(i, j, k, s, b)$ in Algorithm 11. We show that $SecStr^*(i, j, k, s, b) = SecStr(i, j, k, s, b)$ by induction on $j - i \geq 0$, for all k, s, b . Since $numStr^*(i, j, k, s, b) = |SecStr^*(i, j, k, s, b)|$ and $numStr(i, j, k, s, b) = |SecStr(i, j, k, s, b)|$, the result will follow.

BASE CASE: $i \leq j \leq i + \theta$.

In this case, S must be \emptyset , k must be 0, $s = \{a_i, \dots, a_j\}$,^e and $b = \min(j - i + 2, \theta + 1)$. By the initialization step of Algorithm 11, $BP^*(i, j, 0, s, b) = 0 = BP(i, j, 0, s, b)$, $SecStr^*(i, j, 0, s, b) = \{\emptyset\} = SecStr(i, j, 0, s, b)$ and $numStr^*(i, j, 0, s, b) = 1 = numStr(i, j, 0, s, b)$.

INDUCTIVE CASE: Assume that the statement of the theorem holds for all i, j, k, s, b , provided that $j - i \leq d_0$. Consider i, j such that $0 \leq j - i = d_0 + 1$, and fix k, s, b . Consider secondary structure $S \in SecStr(i, j, k, s, b)$. Since S is k -locally optimal, $|S| + k = BP(i, j, 0)$. Now S uniquely satisfies one of the following three cases.

CASE 1: $(i, j) \in S$.

It must be that parameters $s = 0 = b$, since no nucleotides or positions in $[i, j]$ are visible if i, j basepair. Let $S_0 = S_{[i+1, j-1]}$, and let k_0, s_0, b_0 be such that $S_0 \in SecStr(i + 1, j - 1, k_0, s_0, b_0)$. By the induction hypothesis, $SecStr^*(i + 1, j - 1, k_0, s_0, b_0) = SecStr(i + 1, j - 1, k_0, s_0, b_0)$, and so $S \in SecStr^*(i, j, k, s, b)$. We now show that every structure in $SecStr^*(i, j, k, s, b)$ arising from Case 1 is indeed a k -locally optimal s, b -visible structure on $[i, j]$

Let T_0 be any k_0 -locally optimal s_0, b_0 -visible structures on a_{i+1}, \dots, a_{j-1} . We claim that $T = T_0 \cup \{(i, j)\}$ is a k -locally optimal s, b -visible structure on a_i, \dots, a_j .

If T were not locally optimal, then T_0 would not be locally optimal, a contradiction. Since $|S_0| + k_0 = BP^*(i + 1, j - 1, k_0) = BP(i + 1, j - 1, k_0) = |T_0| + k_0$, we have $|S_0| = |T_0|$, and so $|S| = |S_0| + 1 = |T_0| + 1 = |T|$. It follows that T must be k -locally optimal. Since all positions $k \in [i, j]$ are interior to the base pair (i, j) , it follows that $VisNuc(T) = \emptyset = VisNuc(S)$, and $VisPos(T) = 0 = VisPos(S)$. Thus T is a k -locally optimal s, b -visible structure on $[i, j]$.

By the induction hypothesis $numStr^*(i + 1, j - 1, k_0, s_0, b_0) = numStr(i + 1, j - 1, k_0, s_0, b_0)$, hence Case 1 of Algorithm 11 correctly accounts for all secondary structures arising in this case.

CASE 2: $(r, j) \in S$, for some $i < r < j$.

Let S_0 resp. S_1 denote $S_{[i, r-1]}$ resp. $S_{[r+1, j-1]}$. Let $k_0, k_1, s_0, s_1, b_0, b_1$ be such that $S_0 \in SecStr(i, r - 1, k_0, s_0, b_0)$ and $S_1 \in SecStr(r + 1, j - 1, k_1, s_1, b_1)$.

By the induction hypothesis, $SecStr^*(i, r - 1, k_0, s_0, b_0) = SecStr(i, r - 1, k_0, s_0, b_0)$ and $SecStr^*(r + 1, j - 1, k_1, s_1, b_1) = SecStr(r + 1, j - 1, k_1, s_1, b_1)$, so $S \in SecStr^*(i, j, k, s, b)$. We now show that every structure in $SecStr^*(i, j, k, s, b)$ arising from Case 2 is indeed a k -locally optimal s, b -visible structure on $[i, j]$

Consider any $T_0 \in SecStr(i, r - 1, k_0, s_0, b_0)$ and $T_1 \in SecStr(r + 1, j - 1, k_1, s_1, b_1)$ and let $T = T_0 \cup T_1 \cup \{(r, j)\}$. We claim that T is a k -locally optimal s, b -visible structure on $[i, j]$.

If T were not locally optimal, then either T_0 or T_1 would not be locally optimal, a contradiction. Since $|S_0| + k_0 = BP^*(i, r - 1, k_0) = BP(i, r - 1, k_0) = |T_0| + k_0$, and $|S_1| + k_1 = BP^*(r + 1, j - 1, k_1) = BP(r + 1, j - 1, k_1) = |T_1| + k_1$, we have $|S_0| = |T_0|$, $|S_1| = |T_1|$, and so $|S| = |S_0| + |S_1| + 1 = |T_0| + |T_1| + 1 = |T|$. It follows that T is k -locally optimal. Clearly

$$VisNuc(S) = VisNuc(S_0) \cup \{a_{j-x} : 0 \leq x < b\}.$$

Since $VisNuc(S_0) = VisNuc(T_0)$, $VisPos(S_0) = VisPos(T_0)$ and

$$VisNuc(T) = VisNuc(T_0) \cup \{a_{j-x} : 0 \leq x < b_0\}$$

^eNote that s is a set, rather than list or sequence, and so as sets $\{A, C, A, A, G\} = \{A, C, G\}$.

it follows that T is (s, b) -visible.

By the induction hypothesis $numStr^*(i, r-1, k_0, s_0, b_0) = numStr(i, r-1, k_0, s_0, b_0)$, $numStr^*(r+1, j-1, k_1, s_1, b_1) = numStr(r+1, j-1, k_1, s_1, b_1)$, hence Case 2 of Algorithm 11 correctly accounts for all secondary structures arising in this case.

CASE 3: For all $i \leq r \leq j$, $(r, j) \notin S$; i.e. j is unpaired in $[i, j]$.

Let $S_0 = S_{[i, j-1]}$, so $S_0 = S$, when considered as a secondary structure on $[i, j-1]$. Let k_0, s_0, b_0 be such that $S_0 \in SecStr(i, j-1, k_0, s_0, b_0)$.^f

By the induction hypothesis, $SecStr^*(i, j-1, k_0, s_0, b_0) = SecStr(i, j-1, k_0, s_0, b_0)$ so $S \in SecStr^*(i, j, k, s, b)$. We now show that every structure in $SecStr^*(i, j, k, s, b)$ arising from Case 3 is indeed a k -locally optimal s, b -visible structure on $[i, j]$

Let $T_0 \in SecStr(i, j-1, k_0, s_0, b_0)$, and define $T = T_0$, but when considered as a secondary structure on $[i, j]$.

CLAIM: T is locally optimal.

PROOF OF CLAIM. If T were not locally optimal, then either T_0 is locally optimal, a contradiction, or it must be that a_j can basepair with some a_x , where $i \leq x < j - \theta$, and which is external to all basepairs of T_0 . Since $VisNuc(T_0)$ by definition is the set

$$\{a_z : \text{for all } (x, y) \in T_0 [(z < x \text{ or } z > y) \text{ and } z < (j-1) - \theta]\}$$

it must be that either $a_x \in VisNuc(T_0)$ or $x = j - \theta - 1$ and $b_0 = \theta + 1$.

Assume first that the former holds, i.e. that $a_x \in VisNuc(T_0)$. Now $VisNuc(S_0) = VisNuc(T_0)$, so there is some x' , where $i \leq x' < (j-1) - \theta$, which is external to all basepairs of S_0 , and the nucleotides $a_{x'}, a_x$ at positions x', x are the same. But then (x', j) could be added as a basepair to S , violating the locally optimality of S , a contradiction.

Now assume that the latter holds, i.e. that $x = j - \theta - 1$ and $b_0 = \theta + 1$. By definition of b_0 , this means that position x is external to any basepair of T_0 , and since $S_0 \in SecStr(i, j, k_0, s_0, b_0)$, it must be that position x is external to any basepair of S_0 . But then (x, j) could be added as a basepair to S , violating the locally optimality of S , a contradiction. Q.E.D. claim.

We have just shown that T is locally optimal.^g Now since $|S_0| + k_0 = BP^*(i, r-1, k_0) = BP(i, r-1, k_0) = |T_0| + k_0$, we have $|S_0| = |T_0|$, and so $|S| = |T|$. It follows that T is k -locally optimal. Clearly

$$VisNuc(S) = \begin{cases} VisNuc(S_0) \cup \{a_{j-\theta-1}\} & \text{if } b_0 = \theta + 1 \\ VisNuc(S_0) \cup \emptyset & \text{else} \end{cases}$$

By the induction hypothesis, $VisNuc(T_0) = VisNuc(S_0)$ and $VisPos(S_0) = b_0 = VisPos(T_0)$, so

$$VisNuc(T) = \begin{cases} VisNuc(T_0) \cup \{a_{j-\theta-1}\} & \text{if } b_0 = \theta + 1 \\ VisNuc(T_0) \cup \emptyset & \text{else} \end{cases}$$

hence $VisNuc(S) = s = VisNuc(T)$. Finally, since $VisPos(S_0) = b_0 = VisPos(T_0)$ and position j is unpaired in both S and T , it must be that $VisPos(S) = b = VisPos(T)$. Thus $T \in SecStr(i, j, k, s, b)$. This concludes the proof of the theorem. \blacksquare

Using dynamic programming, we store related values $BP(i, j, k, s, b)$, $VisPosLeft(i, j, k, s, b)$, $VisPosRight(i, j, k, s, b)$, $VisNucLeft(i, j, k, s, b)$, $VisNucRight(i, j, k, s, b)$, $kLeft(i, j, k, s, b)$, and

^fNote that it is possible that $k_0 \neq k$, since it can happen that $BP(i, j, 0) > BP(i, j-1, 0)$.

^gIndeed, the entire complication of considering parameters s, b is solely to ensure that T is locally optimal in Case 3.

$kRight(i, j, k, s, b)$, where the first records the number of basepairs in a k -locally optimal secondary structure on $[i, j]$ with visibility parameters $VisNuc(i, j, k) = s$, $VisPos(i, j, k) = b$, while the next four correspond to visibility parameters for left/right components of a k -locally optimal secondary structure, and the last two correspond to the local optimality parameter of the left/right components. These values, with further details omitted, are necessary in our implementation of Algorithm 11, which additionally outputs for each value k , if appropriate, an example k -locally optimal secondary structure on a_1, \dots, a_n .

One area for potential improvement of our current work is to determine, for each i, j, k, s, b , that s, b -visible k -locally optimal secondary structure on a_i, \dots, a_j having minimum free energy, or alternatively to compute the average free energy of all s, b -visible k -locally optimal secondary structures on a_i, \dots, a_j . This would, however, require substantial reorganization of our current code.

Note that it is an easy and well-known exercise to compute the partition function $Z = \sum_S e^{-E(S)/RT}$ for the Nussinov-Jacobson energy function, following ideas of [22]. Algorithm 11 then allows the computation of the Boltzmann probability of k -locally optimal structures

$$Pr[k\text{-locally optimal structures}] = \frac{numStr(k) \cdot e^{-(BP(1,n,0)-k)/RT}}{Z}.$$

In future work, we plan to pursue this application as a potential tool for the detection of noncoding RNA genes.

4 Results

Our results are summarized graphically in Figures 3, 4, 5, each consisting of two graphs, where the left plot is for k -locally optimal sample minimum free energy, and more importantly, the right plot is for the number of k -locally optimal structures. The latter can easily be transformed into *density of locally optimal state* graphs, as in Figure 6. Figure 3 concerns a typical SECIS element, Figure 4 concerns the typical hammerhead type III ribozyme 23AF170503 from Rfam, and Figure 5 concerns Sprinzl’s collection of length 76 nucleotide tRNA. Figures 3 and 4 were produced by plotting values corresponding to a single, structurally important RNA with average values of 100 RNAs of the same dinucleotide frequency. Figure 5 was produced by plotting average values for Sprinzl’s collection of length 76 nucleotide tRNA along with average values from a collection of ten times that number of random RNAs, obtained by generating 10 random RNAs per tRNA, each with the same dinucleotide frequency of the given tRNA. Figures 3, 4 and 5 each contain both a curve from structurally important RNAs (SECIS, hammerhead, tRNA) and a curve from summary data on 100 random RNA sequences having the same dinucleotide frequency.^h Table 7 displays data, where the last column is the ratio of the density of locally optimal states of random RNA over that of SECIS element `formy1MRF`. For instance, the density of 0-locally optimal states is 6717 times greater for random RNA than for `formy1MRF`, 754 times greater for 1-locally optimal states of random RNA than for `formy1MRF`, etc. The third column of this table gives the ratio of the number of k -locally optimal structures for random RNA over that for `formy1MRF`. Random RNA has many times more k -locally optimal structures, for small k , for random RNA than for biological RNA, creating more potential “decoys” in the folding landscape. By applying our algorithm, it seems promising that our method might be useful in determining whether engineered (artificial) RNA is likely to fold in a rapid and stable manner.

We implemented Algorithm 11 (along with backtracking and omitted parts) in about 1500 lines of C, and additionally wrote about 500 lines of Python code for the analysis of the data given here and for cgi scripts for the web server for this algorithm, available at

<http://clavius.bc.edu/~clotelab/>.

^hWorkman and Krogh [33] showed that, in contrast to the results reported by Seffens and Digby [27], mRNA has no lower free energy than that of random RNA of the same dinucleotide frequency.

5 Acknowledgements

I'd like to thank Ivo Hofacker, for some remarks, and Vince Moulton, who raised the question of the number of secondary structures having base pair distance of δ from a given structure. This question, answered in a joint paper in preparation, led to reflections on the RNA energy spectrum. The current work emanated from the Benasque RNA workshop in summer 2003, organized by Elena Rivas and Eric Westhof, both of whom I'd like to thank warmly.

References

- [1] Anfinsen, C.B. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.
- [2] Berger, B. and Leighton, T. Protein folding in the hydrophobic-hydrophilic (hp) model is NP-complete. *Journal of Computational Biology*, 5:27–40, 1998.
- [3] Bryngelson, J.D. and Wolynes, P.G. Spin glasses and the statistical mechanics of protein folding. *Proc. Natl. Acad. Sci. USA*, 84:7524–7528, 1987.
- [4] Chan, H.S. and Dill, K.A. Compact polymers. *Macromolecules*, 22:4559–4573, 1989.
- [5] Clote, P. Protein folding, the Levinthal paradox and rapidly mixing Markov chains. In *Automata, Languages and Programming, 26th International Colloquium, ICALP'99*, pages 240–249. Springer Verlag, 1999. Springer Lecture Notes in Computer Science, **1644**.
- [6] Clote, P. and Backofen, R. *Computational Molecular Biology: An Introduction*. John Wiley & Sons, 2000. 279 pages.
- [7] Clote, P., Kranakis, E., and Krizanc, D. Asymptotics of random RNA. In R. Spang, P. Béziat, and M. Vingron, editors, *Currents in Computational Molecular Biology 2003*, pages 149–150. IEEE, 2003.
- [8] Cohen, B. and Skienna, S. Natural selection and algorithmic design of mRNA. *Journal of Computational Biology*, 10(3-4):419–432, 2002.
- [9] Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., and Yannakakis, M. On the complexity of protein folding. *J. Comp. Biol.*, 5(3):523–466, 1998.
- [10] Cupal, J., Hofacker, I., and Stadler, P. Dynamic programming algorithm for the density of states of RNA secondary structures. In R. Hofstädt, T. Lengauer, M. Löffler, and D. Schomburg, editors, *Computer Science and Biology 96 (Proceedings of the German Conference on Bioinformatics)*, pages 184–186. Univ. Leipzig, 1996.
- [11] Ding, Y. and Lawrence, C.E. Statistical prediction of single-stranded regions in RNA secondary structure and application to predicting effective antisense target sites and beyond. *Nucleic Acids Res.*, 29:1034–1046, 2001.
- [12] Ding, Y. and Lawrence, C.E. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res.*, 31(24):7280–7301, 2003.
- [13] Hofacker I. et al. Vienna RNA Package. <http://www.tbi.univie.ac.at/~ivo/RNA/>
- [14] Evers, D.J. and Giegerich, R. Reducing the conformation space in RNA structure prediction. In *German Conference on Bioinformatics (GCB'01)*, 2001.
- [15] Flamm, C., Fontana, W., Hofacker, I.L., and Schuster, P.F. RNA folding at elementary step resolution. *RNA*, 6:325–338, 2000.
- [16] Flamm, C., Hofacker, I.L., Stadler, P.F., and Wolfinger, M. Barrier trees of degenerate landscapes. *Z. Phys. Chem.*, 216:155–173, 2002.
- [17] Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A., and Eddy, S.R. Rfam: an RNA family database. *Nucleic Acids Res.*, 31(1):439–441, 2003.
- [18] Harborth, J., Elbashir, S.M., Vandeburgh, K., Manninga, H., Scaringe, S.A., Weber, K., and Tuschl, T. Sequence, chemical, and structural variation of small interfering RNAs and short hairpin RNAs and the effect on mammalian gene silencing. *Antisense Nucleic Acid Drug Dev.*, 13:83–106, 2003.
- [19] Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., and Schuster, P. Fast folding and comparison of RNA secondary structures. *Monatsch. Chem.*, 125:167–188, 1994.
- [20] Levinthal, C. Are there pathways for protein folding? *Journal de Chimie Physique*, 65:44–45, 1968.
- [21] Mathews, D.H., Sabina, J., Zuker, M., and Turner, D.H. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, 288:911–940, 1999.
- [22] McCaskill, J.S. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.
- [23] Nussinov, R. and Jacobson, A.B. Fast algorithm for predicting the secondary structure of single stranded RNA. *Proceedings of the National Academy of Sciences, USA*, 77(11):6309–6313, 1980.
- [24] Rivas, E. and Eddy, S. Secondary structure alone is generally not statistically significant for the detection of noncoding RNA. *Bioinformatics*, 16:573–585, 2000.
- [25] Šali, A., Shakhnovich, E., and Karplus, M. How does a protein fold? *Nature*, 369:248–251, May 1994.

- [26] Šali, A., Shakhnovich, E., and Karplus, M. Kinetics of protein folding: A lattice model study of the requirements for folding to the native state. *Journal of Molecular Biology*, 235:1614–1636, 1994.
- [27] Seffens, W. and Digby, D. mRNAs have greater negative folding free energies than shuffled or codon choice randomized sequences. *Nucl. Acids. Res.*, 27:1578, 1999.
- [28] Shakhnovich, E. Theoretical studies of protein-folding thermodynamics and kinetics. *Current Opinion in Structural Biology*, 7:29–40, 1997.
- [29] Sinclair, A. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Birkhäuser, 1993.
- [30] Sprinzl, M., Horn, C., Brown, M., Ioudovitch, A., and Steinberg, S. Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res.*, 26:148–153, 1998.
- [31] Sprinzl, M., Vassilenko, K.S., Emmerich, J., and Bauer, F. tRNA Database. <http://www.staff.uni-bayreuth.de/~btc914/search/index.html>
- [32] Tuschl, T. Functional genomics: RNA sets the standard. *Nature*, 421:220–221, 2003.
- [33] Workman, C. and Krogh, A. No evidence that mRNAs have lower folding free energies than random sequences with the same dinucleotide distribution. *Nucl. Acids. Res.*, 27:4816–4822, 1999.
- [34] Zuker, M. and Sankhoff, D. RNA secondary structures and their prediction. *Bulletin of Biology*, 46(4):591–621, 1984.
- [35] Zuker, M. and Sankhoff, D. RNA secondary structures and their prediction. *Bull. Math. Biol.*, 46:591–621, 1984.
- [36] Zuker, M. and Stiegler, P. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, 9:133–148, 1981.

Appendix

In the following, an assignment of the form $x += y$ is shorthand for $x=x+y$.

Algorithm 11 (Energy spectrum) *Given RNA nucleotide sequence a_1, \dots, a_n , the following pseudocode computes the number of k -locally optimal secondary structures of a_1, \dots, a_n , for all $0 \leq k \leq n$.*

1. Run Nussinov-Jacobson to compute $BP^*(i, j, 0) = BP(i, j, 0)$ for $1 \leq i < j \leq n$.
2. Compute $numStr^*(i, j, 0) = numStr(i, j, 0)$ and $numStr^*(i, j, 0, s, b) = numStr(i, j, 0, s, b)$, and set $SecStr^*(i, j, 0, s, b) = SecStr(i, j, 0, s, b)$ for all $1 \leq i < j \leq n$, $s \subseteq \{A, C, G, U\}$, and $0 \leq b \leq \theta + 1$ (details omitted, but idea is to modify Nussinov-Jacobson).
3. For $1 \leq i < j \leq n$ and $1 \leq k \leq n$, set $BP^*(i, j, k, s, b)$ to be undefined, initialize $numStr^*(i, j, k, s, b) = numStr^*(i, j, k) = 0$, $SecStr^*(i, j, k, s, b) = SecStr^*(i, j, k)$ and execute the following.
 - (a) If a_i, a_j can basepair, then for all $1 \leq \ell \leq \lfloor \frac{n}{2} \rfloor$, $s \subseteq \{A, C, G, U\}$ and $0 \leq b \leq \theta + 1$, if $BP^*(i+1, j-1, \ell) = BP^*(i+1, j-1, \ell, s, b)$, then:
$$k = BP^*(i, j, 0) - [1 + BP^*(i+1, j-1, \ell)]$$
 if $k > 0$ {
$$BP^*(i, j, k) = 1 + BP^*(i+1, j-1, \ell)$$

$$s_0 = 0; b_0 = 0; //when a_i, a_j basepair, nothing in [i, j] is visible$$

$$BP^*(i, j, k, s_0, b_0) = 1 + BP^*(i+1, j-1, \ell)$$

$$SecStr^*(i, j, k, s_0, b_0) = \{S \cup \{(i, j)\} : S \in SecStr^*(i+1, j-1, \ell, s, b)\}$$

$$numStr^*(i, j, k, s_0, b_0) += numStr^*(i+1, j-1, \ell, s, b)$$

$$numStr^*(i, j, k) += numStr^*(i+1, j-1, \ell, s, b)$$

$$SecStr^*(i, j, k) = SecStr^*(i, j, k) \cup SecStr^*(i, j, k, s_0, b_0)$$
 }
 - (b) For $i < r < j - \theta$, if a_r, a_j can basepair then for all $1 \leq \ell, \ell' \leq \lfloor \frac{n}{2} \rfloor$, $s, s' \subseteq \{A, C, G, U\}$ and $0 \leq b, b' \leq \theta + 1$, if $BP^*(i, r-1, \ell) = BP^*(i, r-1, \ell, s, b)$ and $BP^*(r+1, j-1, \ell') = BP^*(r+1, j-1, \ell', s', b')$, then:
$$k = BP^*(i, j, 0) - [1 + BP^*(i, r-1, \ell) + BP^*(r+1, j-1, \ell')]$$
 if $k > 0$ {
$$BP^*(i, j, k) = 1 + BP^*(i, r-1, \ell) + BP^*(r+1, j-1, \ell')$$

$$s_0 = s \cup \{a_{j-x} : 0 \leq x \leq b\}; b_0 = 0;$$

$$BP^*(i, j, k, s_0, b_0) = 1 + BP^*(i, r-1, \ell) + BP^*(r+1, j-1, \ell')$$

$$numStr_0^* = numStr^*(i, r-1, \ell, s, b)$$

$$numStr_1^* = numStr^*(r+1, j-1, \ell', s', b')$$

$$S_0^* = SecStr^*(i, r-1, \ell, s, b)$$

$$S_1^* = SecStr^*(r+1, j-1, \ell', s', b')$$

$$SecStr^*(i, j, k, s_0, b_0) = \{S \cup \{(r, j)\} : S \in S_0^* \cup S_1^*\}$$

$$numStr^*(i, j, k, s_0, b_0) += numStr_0^* + numStr_1^*$$

$$numStr^*(i, j, k) += numStr_0^* + numStr_1^*$$

$$SecStr^*(i, j, k) = SecStr^*(i, j, k) \cup SecStr^*(i, j, k, s_0, b_0)$$
 } //Special case when either term on right of last assignment equals 0
 //Such cases omitted for brevity
 - (c) For all $1 \leq \ell \leq \lfloor \frac{n}{2} \rfloor$, $s \subseteq \{A, C, G, U\}$ and $0 \leq b \leq \theta + 1$, if $BP^*(i, j-1, \ell) = BP^*(i, j-1, \ell, s, b)$, then:
$$k = BP^*(i, j, 0) - BP^*(i, j-1, \ell)$$
 if $k > 0$ {

```

if  $b = \theta + 1$  and  $a_j$  can basepair with  $a_{j-\theta-1}$ 
  //  $a_{j-\theta-1}$  is visible and can basepair with  $a_j$  (bad case)
  skip
else if  $b = \theta + 1$ 
  //  $a_{j-\theta-1}$  is visible, but can't basepair with  $a_j$ 
   $s_0 = s \cup \{a_{j-x}\}$ 
   $b_0 = \min(b + 1, \theta + 1)$ 
else //  $b \leq \theta$ , i.e.  $a_{j-\theta-1}$  not visible
   $s_0 = s$ 
   $b_0 = \min(b + 1, \theta + 1)$ 
 $BP^*(i, j, k) = BP^*(i, j - 1, \ell)$ 
 $BP^*(i, j, k, s_0, b_0) = BP^*(i, j - 1, \ell, s, b)$ 
 $numStr^*(i, j, k, s_0, b_0) += numStr^*(i, j - 1, \ell, s, b)$ 
 $numStr^*(i, j, k) += numStr^*(i, j - 1, \ell)$ 
 $SecStr^*(i, j, k, s_0, b_0) = SecStr^*(i, j, k, s_0, b_0) \cup SecStr^*(i, j - 1, \ell, s, b)$ 
 $SecStr^*(i, j, k) = SecStr^*(i, j, k) \cup SecStr^*(i, j, k, s_0, b_0)$ 
}

```

The previous pseudocode outlines the gist of our algorithm, which is implemented using dynamic programming (i.e. for values $d = 1, \dots, n - 1$ and $i = 1, \dots, n - 1$, one sets $j = i + d$ and for this fixed pair i, j one handles all k, s, b). Our implementation requires additional variables such as $kLeft$, $kRight$, $sLeft$, $sRight$, $bLeft$, $bRight$, etc. which are necessary for the traceback used in backtracking to produce a sample k -locally optimal secondary structure, for each k . Further details are omitted.

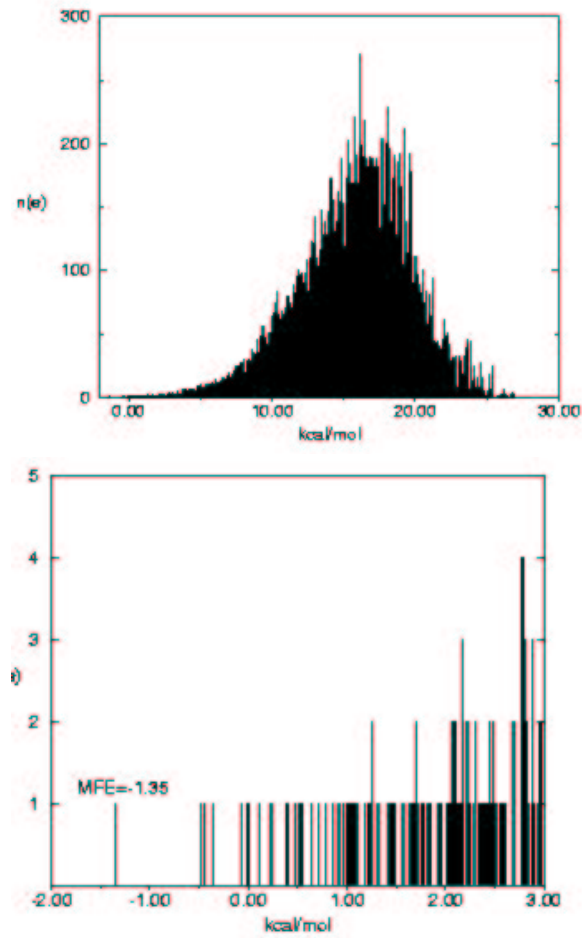


Figure 1: Density of states (left) and energy spectrum (right) for sequence GUCGUAGUCGAUGCUCUAGCUG, reproduced from Cupal, Hofacker and Stadler [10]. The minimum free energy structure is $E = -1.35$ kcal/mol and there are 70727 possible secondary structures. The energy spectrum illustrates an energy gap of 0.87 between the native state and the next lowest energy state. (Reproduced with permission.)

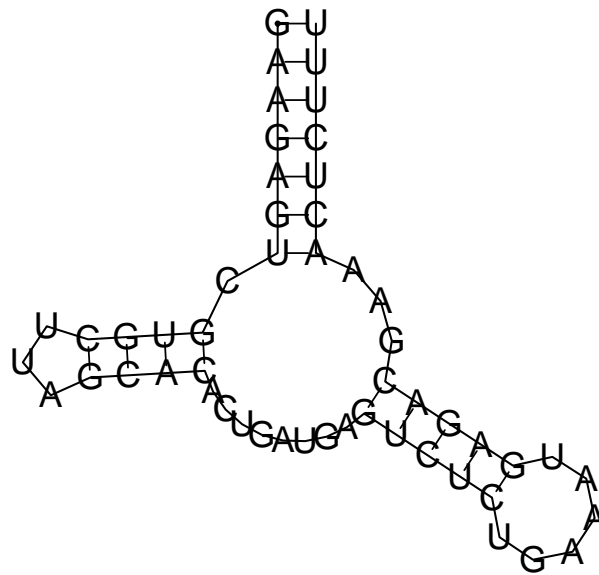


Figure 2: Secondary structure of hammerhead ribozyme GAAAGGUCUGUGCUUAGCACACUGACGAGUCCUGAAAUGGAACGAAACCUUUU with Rfam accession number AF170503 having balanced parenthesis formatted secondary structure ((((((((((((.....)))))).....((((.....)))))).....)))))) and minimum free energy -16.90 kcal/mol. (Image produced by RNAfold.)

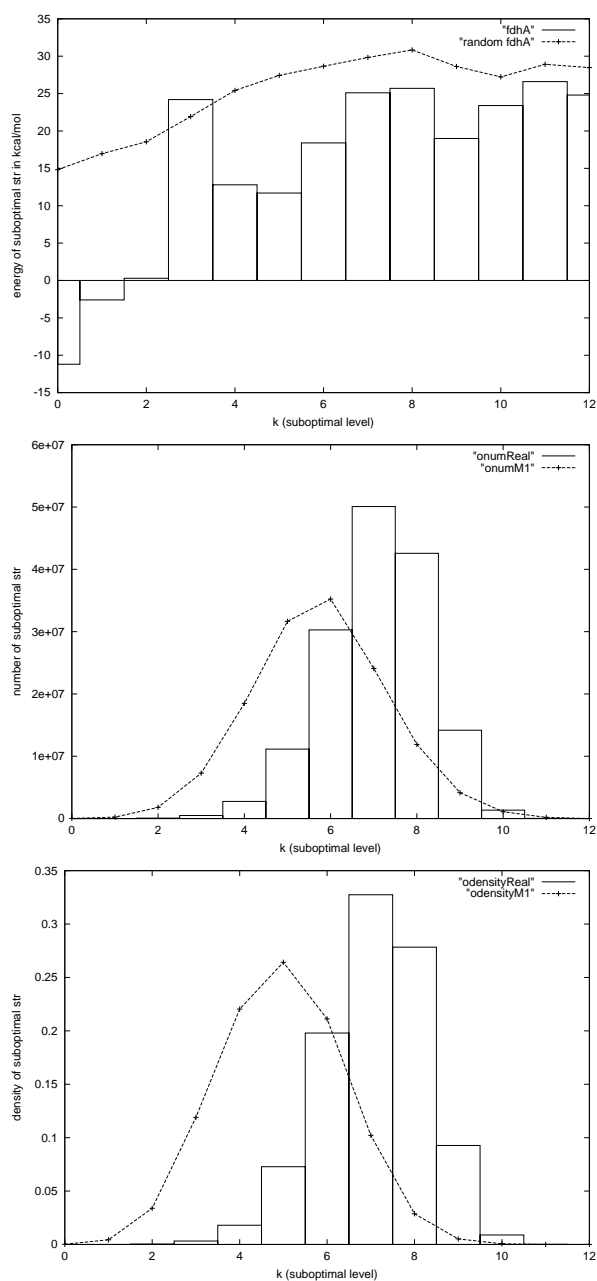


Figure 3: (i) Minimum free energy (mfe) of sample k -locally optimal secondary structures, using RNAeval, of 49 nt. SECIS element fdhA, from A. Böck (Ludwig-Maximilians-Universität München, personal communication), versus average of sample k -locally optimal mfe of 100 49 nt. random RNAs of same dinucleotide frequency. (Top curve for random RNA.) (ii) Average number of k -locally optimal secondary structures of 49 nt. SECIS element fdhA, versus average number of k -locally optimal secondary structures of 100 49 nt. random RNAs of same dinucleotide frequency. (Left curve for random RNA.) (iii) Density of k -locally optimal secondary structures of 49 nt. SECIS element fdhA, versus average density of k -locally optimal secondary structures of 100 49 nt. random RNAs of same dinucleotide frequency. (Left curve for random RNA.)

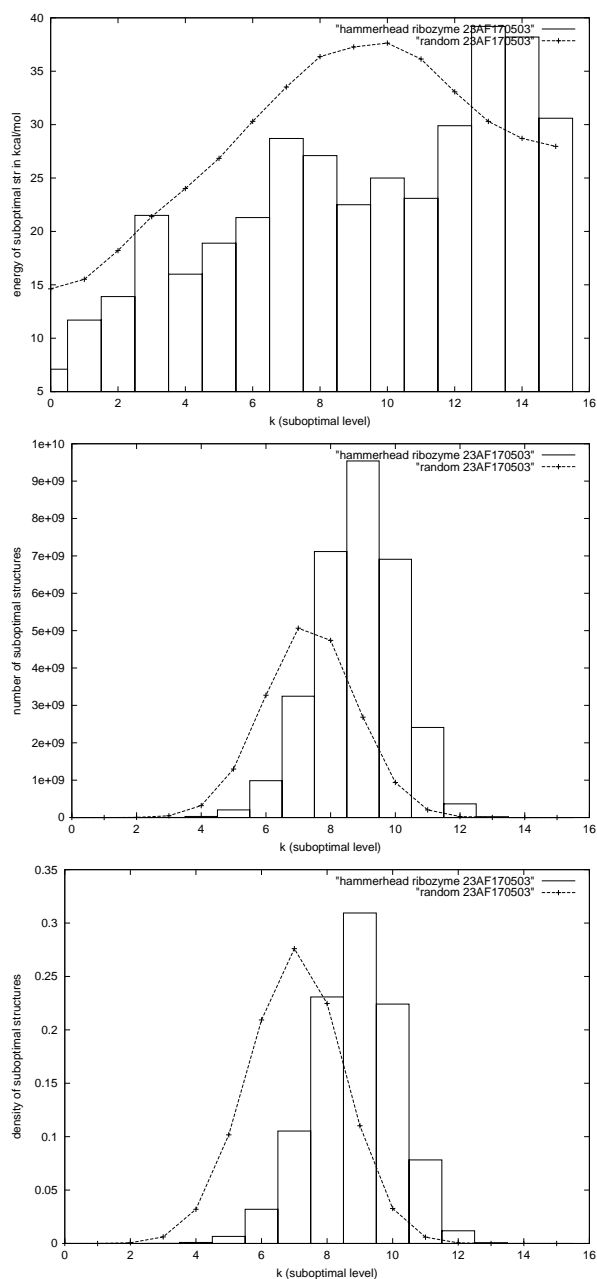


Figure 4: (i) Minimum free energy (mfe) of sample k -locally optimal secondary structures, using RNAeval, of 54 nt. hammerhead type III ribozyme 23AF170503, from Rfam, versus average of sample k -locally optimal mfe of 100 54 nt. random RNAs of same dinucleotide frequency. (Top curve for random RNA.) (ii) Average number of k -locally optimal secondary structures of 54 nt. hammerhead type III ribozyme 23AF170503, from Rfam, versus average number of k -locally optimal secondary structures of 100 54 nt. random RNAs of same dinucleotide frequency. (Left curve for random RNA.) (iii) Density of k -locally optimal secondary structures of 54 nt. hammerhead type III ribozyme 23AF170503, from Rfam, versus average number of k -locally optimal secondary structures of 100 54 nt. random RNAs of same dinucleotide frequency. (Left curve for random RNA.)

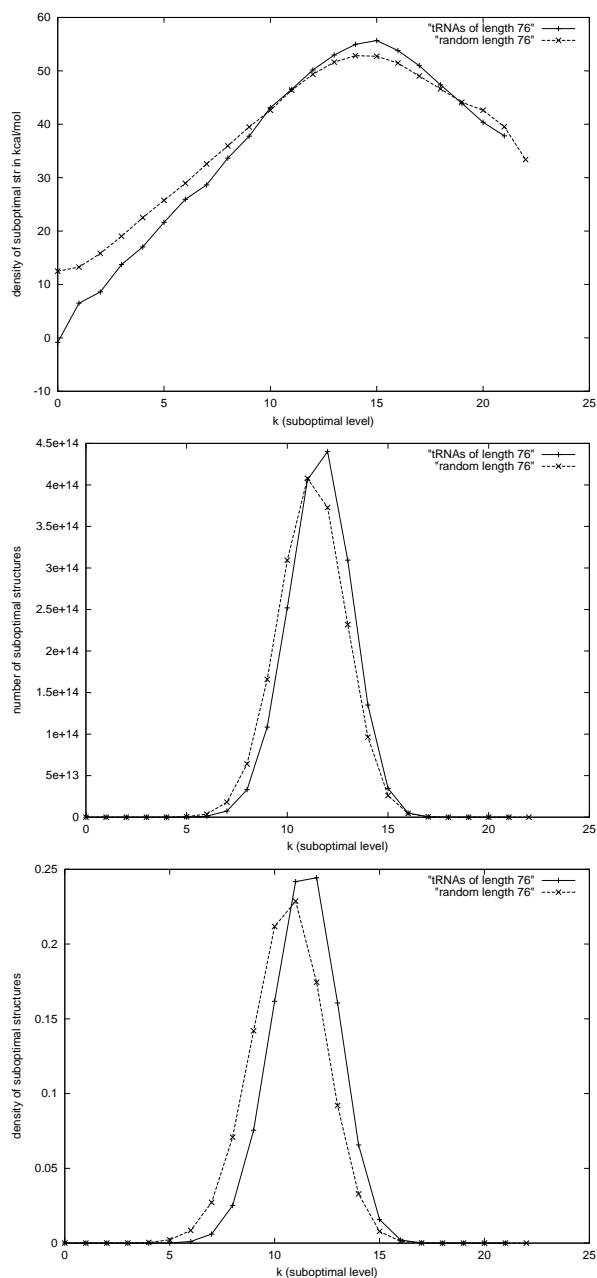


Figure 5: (i) Average minimum free energy (mfe), using `RNAeval`, of sample k -locally optimal secondary structures from Sprinzl's collection [31] of 76 nt. tRNAs, versus ten times that number of random 76 nt. RNAs having same dinucleotide frequency of the former. (ii) Average number of k -locally optimal secondary structures from Sprinzl's collection [31] of 76 nt. tRNAs, versus random 76 nt. RNAs having same dinucleotide frequency. (iii) Density of k -locally optimal secondary structures from Sprinzl's collection [31] of 76 nt. tRNAs, versus random 76 nt. RNAs having same dinucleotide frequency.

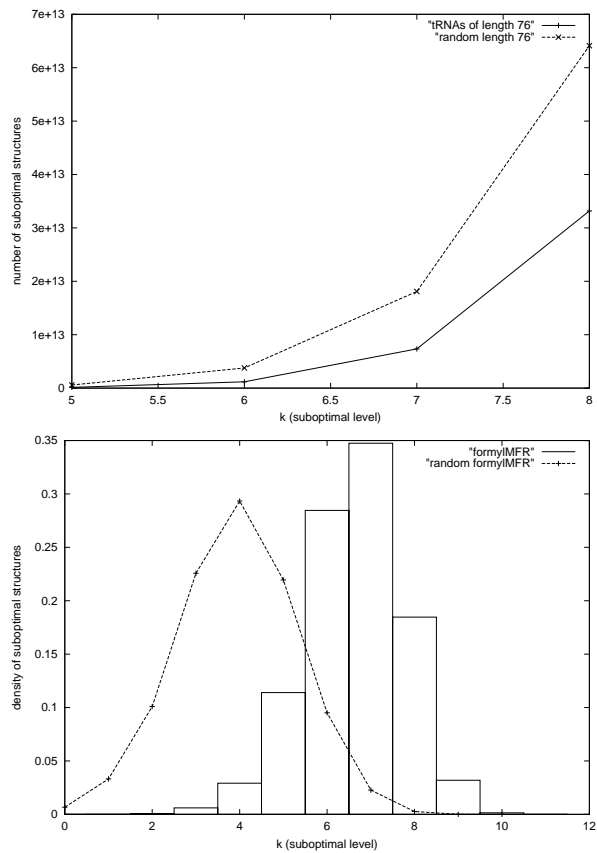


Figure 6: (i) Enlarged initial segment of Figure 5 (ii). (ii) Density of locally optimal states for SECIS element formylMFR and 100 random RNAs of same length and dinucleotide frequency. (Left curve for random RNA.)

k	$N(f)$	$N(r)$	$N(r)/N(f)$	$\rho(f)$	$\rho(r)$	$\rho(r)/\rho(f)$
0	5.000000	3165.060000	633.012000	0.000001	0.006717	6717.000000
1	187.000000	30694.000000	164.139037	0.000044	0.033205	754.659091
2	3038.000000	149776.000000	49.300856	0.000711	0.100938	141.966245
3	25765.000000	457407.000000	17.753037	0.006031	0.225671	37.418504
4	124712.000000	816999.000000	6.551086	0.029192	0.293425	10.051555
5	487115.000000	789899.000000	1.621586	0.114022	0.219475	1.924848
6	1215720.000000	425506.000000	0.350003	0.284572	0.095083	0.334126
7	1484400.000000	142984.000000	0.096324	0.347462	0.022720	0.065388
8	788851.000000	28985.700000	0.036744	0.184651	0.002659	0.014400
9	136190.000000	3812.730000	0.027996	0.031879	0.000105	0.003294
10	6119.000000	76.500000	0.012502	0.001432	0.000000	0.000000
11	15.000000	0.000000	0.000000	0.000004	0.000000	0.000000

Figure 7: Number of k -locally structures and density of k -locally optimal structures for SECIS element `formylMRF` with nucleotide sequence `AUGUUGGAGGGGAACCCUGUAAGGGACCCUCCAACAU`, together with average values of 100 random RNA of same dinucleotide frequency. Here $N(f)$ resp. $N(r)$ denotes the number of k -locally structures for `formylMRF` resp. random RNA, while $\rho(f)$ resp. $\rho(r)$ denotes the density of k -locally structures for `formylMRF` resp. random RNA. The corresponding ratios $N(r)/N(f)$ and $\rho(r)/\rho(f)$ are displayed. While the Nussinov-Jacobson optimal secondary structure for `formylMRF` has 15 base pairs, statistics for the average number of base pairs for the optimal secondary structure of 100 random RNAs of the same dinucleotide frequency as that of `formylMRF` are as follows: mean 11.33, standard deviation 1.50, maximum 14, minimum 6 (here the max,min are taken over 100 random sequences). Though `formylMRF` has 11-locally optimal structures, i.e. having $15 - 11 = 4$ base pairs, there are no 12-locally optimal structures.