A Partition Function Algorithm for Nucleic Acid Secondary Structure Including Pseudoknots

ROBERT M. DIRKS,[1] NILES A. PIERCE[2]
[1]*Department of Chemistry, California Institute of Technology, Pasadena, California 91125*
[2]*Department of Applied & Computational Mathematics and Department of Bioengineering, California Institute of Technology, Pasadena, California 91125*

Received 16 December 2002; Accepted 5 March 2003

Abstract: Nucleic acid secondary structure models usually exclude pseudoknots due to the difficulty of treating these nonnested structures efficiently in structure prediction and partition function algorithms. Here, the standard secondary structure energy model is extended to include the most physically relevant pseudoknots. We describe an $O(N^5)$ dynamic programming algorithm, where N is the length of the strand, for computing the partition function and minimum energy structure over this class of secondary structures. Hence, it is possible to determine the probability of sampling the lowest energy structure, or any other structure of particular interest. This capability motivates the use of the partition function for the design of DNA or RNA molecules for bioengineering applications.

 J Comput Chem 24: 1664–1677, 2003

Key words: DNA; RNA; partition function; secondary structure; pseudoknots

Introduction

The problem of predicting the minimum energy secondary structure of an RNA or single-stranded DNA (ssDNA) molecule has been studied extensively for the past two decades. Secondary structure is described by a list of base pairs $i \cdot j$ in which each base appears at most once. Using a loop-based nearest-neighbor energy function and experimentally derived parameters, numerous algorithms have been implemented to predict the secondary structure or structures that a given nucleic acid sequence will adopt. The first dynamic programming algorithms for secondary structure prediction were proposed by Waterman and Smith[1,2] and Nussinov et al.[3] In 1981, Zuker and Stiegler[4] introduced an improved dynamic programming algorithm that explores all possible unpseudoknotted secondary structures in $O(N^4)$ time, where N is the sequence length. Several reviews describe subsequent progress on methods for secondary structure prediction.[5–7] In 1990, McCaskill[8] described a different $O(N^4)$ dynamic programming algorithm for computing the partition function of a given sequence over all possible unpseudoknotted secondary structures. Both the Zuker and Stiegler structure prediction algorithm and the McCaskill partition function algorithm can be reduced to $O(N^3)$ complexity using a simplified energy model.[4,8] The partition function can be used to derive thermodynamic properties of the equilibrium conformational ensemble including the base-pairing probability of any two bases.[8,9] As a result, the partition function holds promise as a means of evaluating and improving sequence designs for molecules that are intended to adopt a specified secondary structure.[10,11]

In the absence of pseudoknots, thermodynamic models for nucleic acid secondary structure are based on a decomposition of the base-pairing graph for a molecule into distinct loops that are associated with empirically measured enthalpic and entropic terms that depend on loop sequence, length and type.[12,13] Starting with the work of Tinoco,[14] the development of these physical models has involved the work of many researchers.[5–7] The canonical loop types are illustrated in Figure 1, where the base-pairing graph incorporates stacked bases, hairpin loops, a bulge loop, an interior loop and a multiloop. Depicted as a polymer graph with the polymer backbone drawn as a straight line and paired bases connected by arcs, all of these loop types appear as nested structures

Correspondence to: N. A. Pierce; e-mail: niles@caltech.edu

Contract/grant sponsor: NSF graduate research fellowship (to R.M.D.)

Contract/grant sponsor: Defense Advanced Research Projects Agency (DARPA) Air Force Research Laboratory; contract/grant number: F30602-010200561 (to N.A.P.)

Contract/grant sponsor: Ralph M. Parsons Foundation (to N.A.P.)

This article includes Supplementary Material available from the authors upon request or via the Internet at ftp://ftp.wiley.com/public/journals/jcc/suppmat/24/1664 or http://www.interscience.wiley.com/jpages/0192-8651/suppmat/24/v24.1664.html.

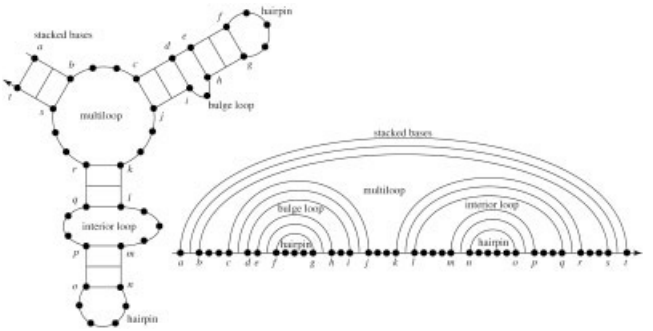

Figure 1. Canonical loops of nucleic acid secondary structure: hairpin loops (one closing base pair), stacked base pairs (two closing base pairs with both loop sides of length zero), a bulge loop (two closing base pairs with one loop side of length greater than zero), an interior loop (two closing base pairs with both loop sides of length greater than zero), and a multiloop (more than two closing base pairs). These loop structures are all nested so there are no crossing arcs in the polymer graph with the polymer backbone drawn as a straight line.

with no crossing arcs. The energy of the structure is the sum of the energies of its constituent loops.

One deficiency in most RNA or ssDNA secondary structure models has been the assumption that these structures do not contain pseudoknots. Pseudoknots are formed when two base pairs $i \cdot j$ and $d \cdot e$, with $i < d$, fail to satisfy the nesting property $i < d < e < j$ as illustrated in Figure 2. The omission of pseudoknots from secondary structure models removes an exponentially large subset of all possible secondary structures from consideration. Pseudoknots are known to exist in ribosomal RNA, viral RNA and a number of ribozymes.[15] Currently, pseudoknots have been identified in over 200 naturally occurring RNAs, as cataloged in the Pseudobase database.[16] Pseudoknotted structures also arise in engineering efforts to design new molecular structures and machines using nucleic acids.[17]

Pseudoknots present a major obstacle to the algorithms commonly used to predict RNA and ssDNA structures. Dynamic programming approaches solve large problems by breaking them up into smaller, self-contained subproblems. For example, to find the minimum energy fold of a sequence containing N nucleotides, Zuker and Stiegler's algorithm[4] calculates the minimum energy fold for each subsequence $[i, j]$, for all $1 \leq i < j \leq N$. Using the standard energy model, in the special case where bases i and j are paired, the assumption that there are no pseudoknots ensures that this subproblem is self-contained; no base between i and j can base-pair with anything outside of this region, and no secondary structure outside of this region will affect the loop energy of this subsequence. As a consequence, the minimum energy fold for this region (still assuming i and j are paired) can be determined independently of the rest of the sequence, and the solution can be applied wherever this subsequence occurs. However, when pseudoknots are allowed, forcing i to be paired with j is not sufficient to define a self-contained subproblem, as neither the structure nor the energy of the region between i and j is independent of the rest of the sequence. Thus, to use a dynamic programming algorithm for pseudoknots, simplifying assumptions about the complexity of pseudoknots must be made, and additional, more intricate recursions must be adopted.

Owing to these difficulties, alternative approaches have been attempted for predicting pseudoknotted secondary structures. Maximum weighted matching[18] has been applied to this problem using a nonloop-based energy model. Heuristics have also been used to include some pseudoknots in structure searches based on the standard energy model.[19–22] However, there is currently no known efficient algorithm that considers all possible secondary structures and produces a minimum energy structure or the partition function. In fact, Lyngso and Pedersen[23] and Akutsu[24] proved that finding a minimum energy structure among all possible pseudoknots is *NP*-hard when using the standard nearest-neighbor energy function.

One strategy for bypassing the inherent intractability of a complete search of secondary structure space is to limit the class of pseudoknots to those that are physically most likely to occur. Rivas and Eddy[25] have attempted to do this by expanding the dynamic programming scheme for structure prediction to include a restricted set of pseudoknots. Owing to a dearth of experimental data on pseudoknot energetics, Rivas and Eddy parameterize a plausible and computationally expedient energy function for pseudoknots. Their algorithm is slower than Zuker and Stiegler's original dynamic program,[4] running in $O(N^6)$ time, but it does successfully capture many possible pseudoknots. Akutsu[24] describes an $O(N^5)$ dynamic program for secondary structure prediction over a different class of pseudoknots. Unfortunately, the recursions defined by Rivas and Eddy and by Akutsu contain many redundancies, and are hence unsuitable for partition function calculations.

A recursion is redundant if a single secondary structure is reached by multiple trajectories in the recursion process. For structure prediction algorithms, redundancy is not a fundamental problem; the goal is to evaluate the minimum energy structure, and hence, it is inconsequential whether the same structure is examined more than once (except for efficiency concerns). Partition function algorithms determine a weighted sum of all configurations, so repetition implies overcounting. The goal of the present work is to design nonredundant pseudoknot recursions that allow for partition function calculations on a restricted set of physically important pseudoknots. The partition function algorithm can be modified in a straightforward way to obtain an algorithm for computing the minimum energy structure over the same class of secondary structures.

This article proceeds by summarizing the standard physical model and introducing McCaskill's partition function algorithm for the unpseudoknotted case, which requires $O(N^4)$ computation and $O(N^2)$ storage.[8] With the exception of interior loop terms, this

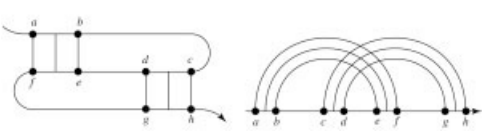

Figure 2. A sample pseudoknot with base pairs $a \cdot f$ and $c \cdot h$ (with $a < c$) that fail to satisfy the nesting property $a < c < h < f$. This leads to crossing arcs in the polymer graph.

algorithm can be reduced to $O(N^3)$ following ideas presented by McCaskill.[8] However, Lyngso et al.[26] have shown that the complexity of the interior loop evaluations can also be reduced to $O(N^3)$ by exploiting certain features of the standard energy model. Following a similar strategy, we present an $O(N^3)$ partition function algorithm for the unpseudoknotted case that requires no approximation to the standard energy model.

A pseudoknot energy model is then introduced that resembles the standard multiloop treatment. A nonredundant $O(N^8)$ algorithm requiring $O(N^4)$ storage is described that computes the partition function for structures including the most common pseudoknots in nature and in engineering applications. By increasing the number of $O(N^4)$ storage arrays, the computational complexity of the algorithm can be reduced to $O(N^6)$. Furthermore, by generalizing the special interior loop treatment to the pseudoknotted case, it is possible to further reduce the computational complexity to $O(N^5)$ without approximation.

The dynamic programming algorithms are described using two compatible representations. Recursion diagrams facilitate the invention, modification, and interpretation of the algorithms by illuminating the relationships between the various recursive quantities. Each diagram corresponds to a mathematical recursion equation. For clarity and conciseness, we present these equations in the form of compact pseudocode.

We perform a preliminary parameterization of our pseudoknot model for RNA using 200 known RNA pseudoknots and 400 unpseudoknotted tRNAs. We then demonstrate the use of the partition function as a tool for sequence design. Given a physical model and a target secondary structure, we are curious whether it is possible to select a sequence that will adopt the desired structure with high probability. This issue is examined for the design of a multiloop and a pseudoknot using the new pseudoknot model. These case studies suggest that most sequences selected using the standard approach of sequence symmetry minimization[27] do not adopt the target secondary structure with high probability. However, by direct optimization of the probability using repeated evaluations of the partition function, it is possible to obtain sequences that are predicted to fold to the target structure with high probability.

Partition Function without Pseudoknots

Standard Energy Model

In the standard energy model for unpseudoknotted secondary structures,[12,13] a loop free energy G_L is associated with each loop L in a secondary structure s, so that the total free energy G_s is

$$G_s = \sum_{L \in s} G_L. \tag{1}$$

The partition function is then a weighted sum over the set of all possible secondary structures S

$$Q = \sum_{s \in S} e^{-G_s/RT} \tag{2}$$

where R is the universal gas constant and T is the temperature.

A base pair $d \cdot e$ is *interior* to another base pair $i \cdot j$ if $i < d < e < j$. In the standard energy model, the energy associated with an empty subsequence $[i, j]$ that contains no base pairs and is external to all loops is assumed to be zero

$$G_{i,j}^{\text{empty}} = 0. \tag{3}$$

The energy associated with a *hairpin loop* closed by base pair $i \cdot j$ is represented by a two-dimensional array

$$G_{i,j}^{\text{hairpin}} \tag{4}$$

that depends on sequence and loop size. The energy of an *interior loop* defined by closing base pair $i \cdot j$ and an interior base pair $d \cdot e$ is represented in a four-dimensional array

$$G_{i,d,e,j}^{\text{interior}} \tag{5}$$

that depends on sequence, loop size and loop asymmetry. *Bulge loops* are treated as special cases of interior loops (where either $d = i + 1$ or $e = j - 1$). *Stacked pairs* are represented by interior loops with both $d = i + 1$ and $e = j - 1$. In treating *multiloops,* it is impractical to incorporate sequence dependence for all of the defining base pairs. This is true both because there is a lack of experimental data and because the energy array would continue to increase in size by a factor of $O(N^2)$ with the addition of each interior base pair to the loop. Instead, the multiloop energetics are approximated by the expression

$$G^{\text{multi}} = \alpha_1 + \alpha_2 B + \alpha_3 U \tag{6}$$

where α_1 is the penalty for the formation of a multiloop, B is the number of base pairs that define the multiloop (including the closing pair $i \cdot j$), and U is the number of unpaired bases in the multiloop. This energy expression is illustrated in Supplementary Material Figure S1. The total energy for a multiloop must be introduced incrementally, as each interior base pair defining the loop is encountered during the multiloop recursions. The form of these incremental pieces of G^{multi} will be stated as the recursions are defined.

$O(N^4)$ Algorithm

To determine the partition function Q for an unpseudoknotted strand of length N, McCaskill's algorithm[8] starts by considering all continuous subsequences of length $l = 1$ and explores all subsequences of incrementally increasing length until $l = N$. The $O(N^4)$ form of the algorithm requires the calculation and storage of three terms $Q_{i,j}$, $Q_{i,j}^b$, and $Q_{i,j}^m$ for each subsequence. These quantities ignore the portions of the structure that are exterior to the subsequence $[i, j]$.

$Q_{i,j}$ represents the full partition function for subsequence $[i, j]$ and is defined recursively by the equation

$$Q_{i,j} = 1 + \sum_{\substack{d,e \\ i \le d < e \le j}} Q_{i,d-1} Q_{d,e}^b \tag{7}$$

Figure 3. $O(N^4)$ Algorithm: recursion for $Q_{i,j}$, the full partition function for the subsequence $[i, j]$. Either the subsequence $[i, j]$ is empty with recursion energy $G_{i,j}^{\text{empty}} = 0$, or there exists one or more pairs with rightmost base pair $d \cdot e$ and recursion energy $G_{e+1,j}^{\text{empty}} = 0$.

where $Q_{i,j}^b$ requires its own recursive definition and represents the partition function for subsequence $[i, j]$ assuming that i and j are base-paired. The definition of $Q_{i,j}$ may be equivalently represented by the recursion diagram of Figure 3. Recursion diagrams[25] are a useful tool for describing the relationships between recursive quantities. A horizontal line indicates the phosphate backbone, a solid curved line indicates that two bases are paired, and a dashed curved line denotes a subsequence with terminal bases that may be paired or unpaired. The letter under the curve matches the superscript of the quantity defining the contribution (e.g., "b" corresponds to Q^b). Shaded regions indicate portions of secondary structure that are fixed at the current recursion level and contribute a *recursion energy* to the partition function as defined by the standard energy model (3)–(6). Unshaded regions under curves have partition function contributions based on recursive quantities previously evaluated for shorter subsequences.

In Eq. (7) and Figure 3, the first possibility is that the subsequence $[i, j]$ is empty, contributing the term $\exp(-G_{i,j}^{\text{empty}}/RT) = 1$. Otherwise, there must exist a rightmost base pair $d \cdot e$ on the subsequence $[i, j]$ denoted by a solid b-curve with an associated partition function contribution given by a previous evaluation of $Q_{d,e}^b$. The term *rightmost* implies that no other base on the subsequence $[e + 1, j]$ is involved in a base pair, so the shaded region is associated with a recursion energy $G_{e+1,j}^{\text{empty}} = 0$. The subsequence $[i, d - 1]$ may, however, contain additional base-pairs, and its partition function is given by a previous evaluation of $Q_{i,d-1}$. Every possible base pair $d \cdot e$ that can be formed in subsequence $[i, j]$ must be considered as a possible rightmost pair, and for each of these, the product $Q_{i,d-1}Q_{d,e}^b \exp(-G_{e+1,j}^{\text{empty}}/RT)$ is added to Q_{ij}. The reliance on the concept of a rightmost pair ensures that the recursions are nonredundant and is a key distinction between McCaskill's partition function approach and the redundant energy minimization recursions of Zuker and Stiegler,[4] Rivas and Eddy,[25] and Akutsu[24] (the use of a *leftmost* extremal convention is equally valid).

The above recursion relied on the calculation of $Q_{i,j}^b$, representing the partition function for subsequence $[i, j]$ assuming i and j are base-paired. This quantity is defined by the recursive equation

$$Q_{i,j}^b = \exp(-G_{i,j}^{\text{hairpin}}/RT) + \sum_{\substack{d,e \\ i<d<e<j}} \exp(-G_{i,d,e,j}^{\text{interior}}/RT)Q_{d,e}^b + \sum_{\substack{d,e \\ i<d<e<j}} Q_{i+1,d-1}^m Q_{d,e}^b \exp\{-[\alpha_1 + 2\alpha_2 + \alpha_3(j - e - 1)]/RT\}, \quad (8)$$

or equivalently by the recursion diagram of Figure 4. Although similar in spirit, there are important differences from the recursions for $Q_{i,j}$ defined above. First, because i and j are paired, the empty recursion becomes a hairpin loop, with a recursion energy given by (4) and a corresponding partition function contribution $\exp(-G_{i,j}^{\text{hairpin}}/RT)$. Second, placing a rightmost pair $d \cdot e$ can lead to two types of structures with very different energy functions. If $[i, j]$ contains only the single interior base pair $d \cdot e$, then an interior loop is formed, with a recursion energy given by (5). The partition function contribution associated with the subinterval $[d, e]$ is given by a previous evaluation of $Q_{d,e}^b$ so that the total contribution for each interior loop structure is $\exp(-G_{i,d,e,j}^{\text{interior}}/RT)Q_{d,e}^b$. Otherwise, in addition to the rightmost pair $d \cdot e$, there must be at least one base pair in the interval $[i + 1, d - 1]$, and a multiloop is formed. This requirement is depicted in Figure 4 by a dashed m-curve which implies that there is at least one base pair in the subinterval that may or may not involve the terminal bases. Rather than explicitly enumerating all possible base pairing scenarios for the interval $[i + 1, d - 1]$ at this level in the recursion (an approach that would increase the time complexity by a factor of $O(N^2)$ for every additional base pair), the influence of these additional pairs may be obtained more efficiently by evaluating the recursive quantity $Q_{i+1,d-1}^m$. This is possible because the energetic model for multiloops (6) depends only on the number of interior base pairs and the number of unpaired bases and not on simultaneous knowledge of all the base pairs that define the multiloop. The multiloop recursion energy for this diagram is then $\alpha_1 + 2\alpha_2 + \alpha_3(j - e - 1)$, accounting for initiating a multiloop, the closing base pair $i \cdot j$, the interior base pair $d \cdot e$ and the number of unpaired bases $j - e - 1$. The corresponding partition function contribution is $Q_{i+1,d-1}^m Q_{d,e}^b \exp\{-[\alpha_1 + 2\alpha_2 + \alpha_3(j - e - 1)]/RT\}$.

The quantity $Q_{i,j}^m$ is used to examine all multiloop structures, and is defined by the recursive equation

$$Q_{i,j}^m = \sum_{\substack{d,e \\ i\le d<e\le j}} [\exp\{-[\alpha_2 + \alpha_3(d - i) + \alpha_3(j - e)]/RT\}Q_{d,e}^b + Q_{i,d-1}^m Q_{d,e}^b \exp\{-[\alpha_2 + \alpha_3(j - e)]/RT\}] \quad (9)$$

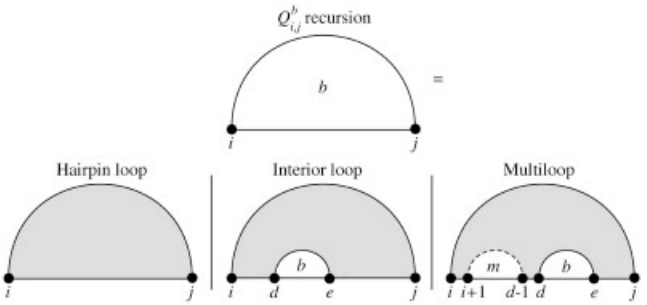

Figure 4. $O(N^4)$ Algorithm: recursion for $Q_{i,j}^b$, the partition function for the subsequence $[i, j]$ assuming i and j are base-paired. Either the subsequence $[i, j]$ is a hairpin loop with recursion energy $G_{i,j}^{\text{hairpin}}$, or there exists one internal base pair $d \cdot e$ forming an interior loop with recursion energy $G_{i,d,e,j}^{\text{internal}}$, or there are at least two interior base pairs forming a multiloop with rightmost pair $d \cdot e$ and recursion energy $\alpha_1 + 2\alpha_2 + \alpha_3(j - e - 1)$.

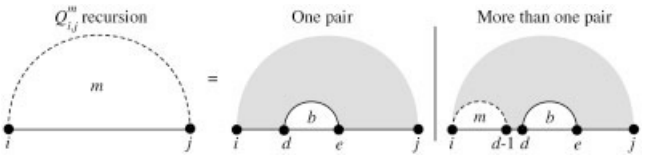

Figure 5. $O(N^4)$ Algorithm: recursion for $Q^m_{i,j}$, the partition function for the subsequence $[i, j]$ inside a multiloop when there is at least one base pair in the subsequence. Either there is only one more base pair $d \cdot e$ defining the multiloop and the recursion energy is $\alpha_2 + \alpha_3(d - i) + \alpha_3(j - e)$, or there is more than one pair with rightmost pair $d \cdot e$ and recursion energy $\alpha_2 + \alpha_3(j - e)$.

or the recursion diagram of Figure 5. Again, we consider the placement of a rightmost base pair $d \cdot e$ with partition function contributions given by a previous evaluation of $Q^b_{d,e}$. Inside a multiloop, there are exactly two possibilities. The pair $d \cdot e$ may complete the definition of the multiloop, in which case the recursion energy $\alpha_2 + \alpha_3(d - i) + \alpha_3(j - e)$ accounts for the single new pair and the remaining bases. Otherwise, there is at least one more base pair in the subsequence $[i, d - 1]$ to be accounted for by a previous evaluation of $Q^m_{i,d-1}$. The recursion energy then accounts for the new pair $d \cdot e$ and the newly identified unpaired bases to give $\alpha_2 + \alpha_3(j - e)$.

Pseudocode for the algorithm is shown in Figure 6, where the recursion Eqs. (7)–(9) lead to $O(N^4)$ computational complexity, as reflected in the programming loops that are nested four deep to compute Q, Q^b, and Q^m. Note that Q^b must be computed prior to Q and Q^m for each subsequence. The bounds for each programming loop are chosen so as to exclude hairpins with fewer than three unpaired bases. These sterically impossible structures have infinite energies in the standard physical model, and so do not contribute to the partition function.

```
Initialize (Q, Q^b, Q^m) // O(N^2) space
Set all values to 0 except Q_{i,i-1} = 1
for l = 1, N
  for i = 1, N-l+1
    j = i+l-1
    // Q^b recursion
    Q^b_{i,j} = exp{-G^hairpin_{i,j}/RT}
    for d = i+1, j-5 // loop over all possible rightmost pairs d·e
      for e = d+4, j-1
        Q^b_{i,j} += exp{-G^interior_{i,d,e,j}/RT} Q^b_{d,e}
        Q^m_{i,j} += Q^m_{i+1,d-1} Q^b_{d,e} exp{-[α1 + 2α2 + α3(j-e-1)]/RT}
    // Q, Q^m recursions
    Q_{i,j} = 1 //empty recursion
    for d = i, j-4 // loop over all possible rightmost pairs d·e
      for e = d+4, j
        Q_{i,j} += Q_{i,d-1} Q^b_{d,e}
        Q^m_{i,j} += exp{-[α2 + α3(d-i) + α3(j-e)]/RT} Q^b_{d,e}
        Q^m_{i,j} += Q^m_{i,d-1} Q^b_{d,e} exp{-[α2 + α3(j - e)]/RT}
//Partition function is Q_{1,N}
```

Figure 6. Pseudocode implementation of McCaskill's $O(N^4)$ dynamic programming partition function algorithm for nucleic acids without pseudoknots. Here, N is the length of the strand and $l = j - i + 1$ is the length of the substrand under consideration at any given point during the recursive process. The recursions are described schematically in Figures 3–5.

Figure 7. $O(N^3)$ Algorithm: recursion for $Q_{i,j}$, the full partition function for the subsequence $[i, j]$. Either the subsequence $[i, j]$ is empty with recursion energy $G^{\text{empty}}_{i,j} = 0$, or there exists one or more pairs with a rightmost base pair that involves d and some other base on the subinterval $[d + 4, j]$.

O(N³) Algorithm

As noted by McCaskill in his original article,[8] this algorithm can be improved to run in time $O(N^3)$ if the standard energy model for interior loops is simplified and extra memory is used to store intermediate values in computing Q and Q^m. Lyngso et al.[26] exploit the form of the standard interior loop energy expression to calculate interior loop contributions in $O(N^3)$. Following McCaskill[8] and Lyngso et al.,[26] we now describe an $O(N^3)$ algorithm that reproduces the results of the $O(N^4)$ algorithm without approximation. The modifications necessary to obtain this improvement will provide a useful precedent for achieving similar gains in the pseudoknotted case.

Recursion diagrams are presented in Figures 7–10 and pseudocode is shown in Figure 11. The recursion for $Q_{i,j}$ described in Figure 7 no longer explicitly considers a rightmost base pair $d \cdot e$. Instead, the secondary recursive quantity $Q^s_{d,j}$ is used to evaluate all possible rightmost base pairs that can form with base d. Note that the s-curve is solid on one half and dotted on the other, because base d is known only to be paired to some base in the interval $[d + 4, j]$.

The recursions for $Q^b_{i,j}$ and $Q^m_{i,j}$ are modified in a similar way in Figures 8 and 9 by introducing the secondary recursion quantity Q^{ms} to compute the multiloop contributions. The $Q^s_{i,j}$ and $Q^{ms}_{i,j}$

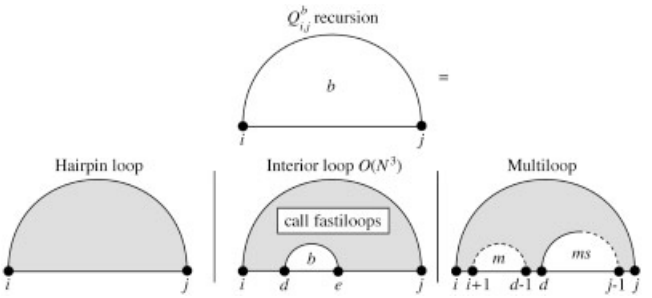

Figure 8. $O(N^3)$ Algorithm: recursion for $Q^b_{i,j}$, the partition function for the subsequence $[i, j]$ assuming i and j are base-paired. Either the subsequence $[i, j]$ is a hairpin loop with recursion energy $G^{\text{hairpin}}_{i,j}$, or there exists one internal base pair $d \cdot e$ forming an interior loop with recursion energy $G^{\text{internal}}_{i,d,e,j}$, or there are at least two interior base pairs with rightmost base pair that involves d and some other base on the subinterval $[d + 4, j - 1]$. The interior loop contributions are obtained in $O(N^3)$ using the function "fastiloops" described in Supplementary Material Figures S2 and S3.

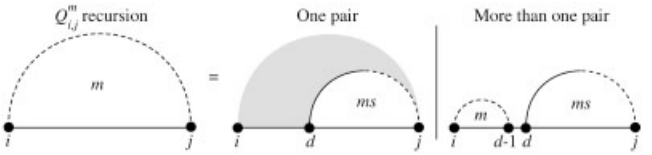

Figure 9. $O(N^3)$ Algorithm: recursion for $Q^m_{i,j}$, the partition function for the subsequence $[i, j]$ inside a multiloop when there is at least one base pair in the subsequence. One possibility is that there is only one more base pair defining the multiloop. In this case, the unpaired subinterval $[i, d-1]$ is associated with a recursion energy $\alpha_3(d-i)$ and the rightmost pair involves d and some other base on the subinterval $[d+4, j]$. The other possibility is that there are at least two interior base pairs with a rightmost base pair that involves d and some other base on the subinterval $[d+4, j]$.

recursions have exactly the same structure but different recursion energies, and are depicted by the recursion diagram in Figure 10. As suggested by the solid and dashed halves of the *s*- and *ms*-curves, this recursion considers all possible base-pairing partners for base i. For $Q^s_{i,j}$, the bases on the subinterval $[d+1, j]$ are external to all loops so the recursion energy is G^{empty}. For $Q^{ms}_{i,j}$, pair $i \cdot d$ is inside a multiloop and the bases $[d+1, j]$ are unpaired bases inside a multiloop so the recursion energy is $\alpha_2 + \alpha_3(j-d)$.

Using these recursions, the pseudocode in Figure 11 now describes an algorithm that is $O(N^3)$ with the exception of the interior loop contributions, which are computed in the function "fastiloops." To this point, the interior loop energy has been described by the black-box function $G^{\text{interior}}_{i,d,e,j}$ of Eq. (5), whose four subscripts imply an $O(N^4)$ computational complexity for computing the interior loop contributions to Q^b. To reduce the complexity, it will be necessary to examine the definition of $G^{\text{interior}}_{i,d,e,j}$.

An interior loop with closing pair $i \cdot j$ and interior pair $d \cdot e$ has sides of length

$$L_1 \equiv d - i - 1, \quad L_2 \equiv j - e - 1 \tag{10}$$

so that the loop size and asymmetry may be expressed as

$$\text{size} \equiv L_1 + L_2, \quad \text{asymmetry} \equiv |L_1 - L_2|. \tag{11}$$

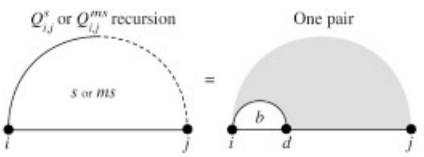

Figure 10. $O(N^3)$ Algorithm: recursion for $Q^s_{i,j}$ and $Q^{ms}_{i,j}$, secondary partition functions for considering all possible rightmost base pairs that involve base i. For $Q^s_{i,j}$, the subsequence $[d, j]$ is external to all base pairs so the recursion energy is $G^{\text{empty}}_{d+1,j} = 0$. For $Q^{ms}_{i,j}$, the subsequence $[d, j]$ is inside a multiloop so the recursion energy is $\alpha_2 + \alpha_3(j-d)$.

```
Initialize (Q, Q^b, Q^m, Q^s, Q^ms, Q^x, Q^x1, Q^x2) //O(N^2) space
Set all values to 0 except Q_{i,i-1} = 1
for l = 1, N // subsequence length
  Initialize Q^x = Q^x1, Q^x1 = Q^x2, Q^x2 = 0
  for i = 1, N-l+1
    j = i+l-1
    // Q^b recursion
    Q^b_{i,j} = exp(-G^hairpin_{i,j}/RT)
    //Compute internal loop contributions to Q^b in O(N^3)
    call fastiloops(i, j, l, Q^b, Q^x, Q^x2)
    for d = i+6, j-5
      Q^b_{i,j} += Q^m_{i+1,d-1} Q^ms_{d,j-1} exp{-[α1 + α2]/RT}
    // Q^s, Q^ms recursion
    for d = i+4, j // loop over all possible rightmost pairs i·d
      Q^s_{i,j} += Q^b_{i,d}
      Q^ms_{i,j} += Q^b_{i,d} exp{-[α2 + α3(j-d)]/RT}
    // Q, Q^m recursions
    Q_{i,j} = 1 //empty recursion
    for d = i, j-4
      Q_{i,j} += Q_{i,d-1} Q^s_{d,j}
      Q^m_{i,j} += exp{-α3[d-i]/RT} Q^ms_{d,j}
      Q^m_{i,j} += Q^m_{i,d-1} Q^ms_{d,j}
//Partition function is Q_{1,N}
```

Figure 11. Pseudocode implementation of an $O(N^3)$ dynamic programming partition function algorithm for nucleic acids without pseudoknots. Here, N is the length of the strand and $l = j - i + 1$ is the length of the substrand under consideration at any given point during the recursive process. The recursions are described schematically in Figures 7–10. The function "fastiloops" computes the interior loop contributions to $Q^b_{i,j}$ for all i and j in $O(N^3)$ as detailed in the pseudocode and schematic of Supplementary Material Figures S2 and S3.

Stacked bases correspond to the special case $L_1 = L_2 = 0$ and bulge loops to the case where either $L_1 = 0$ or $L_2 = 0$. In the standard model, special energy expressions are used for stacked pairs and bulge loops as well as for those interior loops with either $L_1 \le 3$ or $L_2 \le 3$. However, for all cases when both $L_1 \ge 4$ and $L_2 \ge 4$, the form of the energy function becomes

$$G^{\text{interior}}_{i,d,e,j} = \gamma_1(L_1 + L_2) + \gamma_2(|L_1 - L_2|) + \gamma_3(i, j, i+1, j-1) + \gamma_3(e, d, e+1, d-1) \tag{12}$$

corresponding to functions of loop size, loop asymmetry, and the identity of the closing base pairs and nearest neighbors. We term these structures "extensible loops" and the related structures in which i and j are not required to base pair "possible extensible loops." For subsequences of length $l = j - i + 1$, we now define the quantity

$$Q^x_{i,s} \equiv \sum_{\substack{\text{possible extensible loops} \\ \text{with size } L_1+L_2=s}} \exp\{-[\gamma_1(s) + \gamma_2(|L_1 - L_2|) + \gamma_3(e, d, e+1, d-1)]/RT\} Q^b_{d,e}, \tag{13}$$

where d and e are expressed in terms of L_1 and L_2 using (10). If the nucleotides at i and j can form a base pair, then the partition function contributions to $Q^b_{i,j}$ associated with the extensible interior loops of size s can be computed as the product

$$Q^x_{i,s}\exp\{-\gamma_3(i, j, i+1, j-1)/RT\} \quad (14)$$

because all of the loops in the summation are closed by $i \cdot j$. Note that the value of j is implied by i and l. Whether or not i and j can base pair, the quantity $Q^x_{i,s}$ remains useful because it satisfies the following recursive extension property[26]

$$\begin{aligned} Q^x_{i-1,s+2} = {} & Q^x_{i,s}\exp\{-[\gamma_1(s+2) - \gamma_1(s)]/RT\} + \exp\{-[\gamma_1(s+2) \\ & + \gamma_2(|L_1 - L_2|) + \gamma_3(e, d, e+1, d-1)]/RT\}Q^b_{d,e}\big|^{L_1=4}_{L_2=s-2} \\ & + \exp\{-[\gamma_1(s+2) + \gamma_2(|L_1 - L_2|) \\ & + \gamma_3(e, d, e+1, d-1)]/RT\}Q^b_{d,e}\big|^{L_1=s-2}_{L_2=4}. \end{aligned} \quad (15)$$

Hence, possible extensible loops for which i and j cannot base pair can still be used to compute the partition function contributions for larger loops when the sequence does permit the closing base pair to form. The first line of the extension property expresses the fact that extending each side of the possible extensible loop by one base requires a change in the size contribution from $\gamma_1(s)$ to $\gamma_1(s+2)$ but otherwise leaves the asymmetry and interior base pair contributions of each of these structures unchanged. The subsequent lines add the new contributions from possible extensible loops of size $s+2$ with either $L_1 = 4$ or $L_2 = 4$. These are the only two possible extensible loops of length $s+2$ that cannot be obtained by extending smaller loops because these smaller loops do not use the energy expression (12). Exploiting the extension property (15) and making use of (14), the contributions of all extensible interior loops to each $Q^b_{i,j}$ can be computed in $O(N^3)$. For each of $O(N^2)$ closing $i \cdot j$ pairs, the remaining $O(N)$ nonextensible interior loops are evaluated as special cases using expressions contained in the black box function $G^{\text{interior}}_{i,d,e,j}$. The total complexity of the interior loop evaluations is thus $O(N^3)$. Using these ideas, the algorithm for computing the interior loop contributions in $O(N^3)$ is described in the pseudocode and schematic of Supplementary Material Figures S2 and S3.

Minimum Energy Structure Modifications

Recurrence relations that generate each secondary structure exactly once can be applied equally well to either energy minimization or partition function calculations by treating the loop energies differently in the two cases. When the partition function scheme calculates the term $\exp(-G/RT)$ for a loop, the energy minimization scheme considers the loop energy G. When the exponentiated energies are multiplied in the partition function algorithm, the loop energies are added for energy minimization. Finally, when the contributions from alternative structures are added in the partition function scheme, a minimum is taken over these structures in the energy minimization scheme. After fully applying the recursions, the structure prediction scheme identifies the energy of the most stable structure, while the partition function scheme produces a sum with one exponentiated energy term for every possible structure.

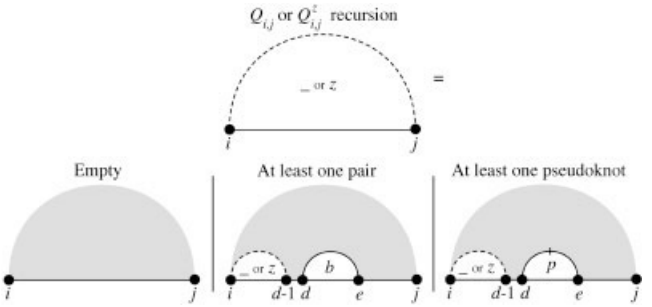

Figure 12. $O(N^8)$ Algorithm: recursion for $Q_{i,j}$, the full partition function for the subsequence $[i, j]$. Either the subsequence $[i, j]$ is empty with recursion energy $G^{\text{empty}}_{i,j}$, or there is at least one pair (with rightmost base pair $d \cdot e$) with recursion energy $G^{\text{empty}}_{e+1,j}$, or there is at least one pseudoknot (with rightmost pseudoknot filling the subsequence $[d, e]$) with recursion energy $\beta_1 + G^{\text{empty}}_{e+1,j}$. The same recursion is used (with modified recursion energies) for $Q^z_{i,j}$, the full partition function for the subsequence $[i, j]$ inside a pseudoknot. In this context, either the subsequence $[i, j]$ is empty with recursion energy $\beta_3(j - i + 1)$, or there is at least one pair (with rightmost base pair $d \cdot e$) with recursion energy $\beta_2 + \beta_3(j - e)$, or there is at least one pseudoknot (with rightmost pseudoknot filling the subsequence $[d, e]$) with recursion energy $\beta^p_1 + 2\beta_2 + \beta_3(j - e)$.

Partition Function with Pseudoknots

Pseudoknot Energy Model

We now introduce an energy model for pseudoknots that is motivated by the standard treatment of multiloops. The energy associated with an exterior pseudoknot is given by

$$G^{\text{pseudo}} = \beta_1 + \beta_2 B^p + \beta_3 U^p, \quad (16)$$

where β_1 is the penalty for introducing a pseudoknot, B^p is the number of base pairs that border the interior of the pseudoknot, and U^p is the number of unpaired bases inside the pseudoknot. If the pseudoknot is inside a multiloop, β_1 is replaced by β^m_1, and if the pseudoknot is inside another pseudoknot, β_1 is replaced by β^p_1. Several features of this potential function are illustrated in Supplementary Material Figure S4.

O(N^8) *Algorithm*

We now introduce pseudoknots into the partition function recursions while maintaining the property that each structure contributes to the partition function exactly once. First, we consider a relatively straightforward but inefficient approach that increases the complexity of the algorithm to $O(N^8)$ and the storage requirements to $O(N^4)$.

In the unpseudoknotted case, Q, Q^b, and Q^m were defined in a nonredundant manner by using the extremal convention of introducing rightmost base pairs or b-curves. The same approach can be followed for pseudoknots, introducing rightmost pseudoknots or p-curves. In Figures 12–14, p-curves are introduced in a completely analagous manner to b-curves. Each p-curve represents the

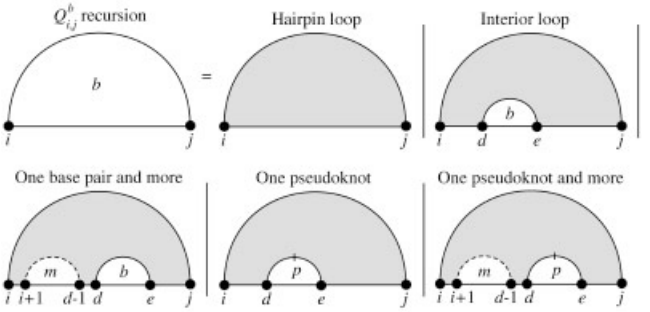

Figure 13. $O(N^8)$ Algorithm: recursion for $Q^b_{i,j}$, the partition function for the subsequence $[i, j]$ assuming i and j are base-paired. Either the subsequence $[i, j]$ is a hairpin loop with recursion energy $G^{\text{hairpin}}_{i,j}$, or there exists one internal base pair $d \cdot e$ forming an interior loop with recursion energy $G^{\text{internal}}_{i,d,e,j}$, or there is more than one base pair or pseudoknot (with rightmost pair $d \cdot e$) forming a multiloop with recursion energy $\alpha_1 + 2\alpha_2 + \alpha_3(j - e - 1)$, or there is one pseudoknot filling the subsequence $[d, e]$ with recursion energy $\alpha_1 + \beta^m_1 + 3\alpha_2 + \alpha_3(d - i - 1) + \alpha_3(j - e - 1)$, or there is more than one pseudoknot or base pair (with rightmost pseudoknot filling the subsequence $[d, e]$) with recursion energy $\alpha_1 + \beta^m_1 + 3\alpha_2 + \alpha_3(j - e - 1)$.

boundaries of a pseudoknot, so d and e are paired to some bases in the subinterval $[d - 1, e - 1]$, but not to each other, as reflected in the solid divided arc of the p-curve. Using a nonredundant definition of the partition function contribution of the p-curve, Q^p, will ensure that the algorithm never visits a structure twice.

The quantity $Q^p_{i,j}$ is defined recursively by the diagram in Figure 15, where the pseudoknot interior is specified. Arcs that would cross in this diagram are reflected across the horizontal axis

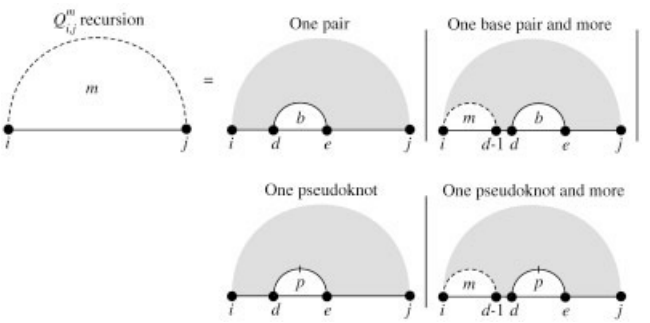

Figure 14. $O(N^8)$ Algorithm: recursion for $Q^m_{i,j}$, the partition function for the subsequence $[i, j]$ inside a multiloop when there is at least one base pair or pseudoknot in the subsequence. Either there is one final base pair $d \cdot e$ in the multiloop with recursion energy $\alpha_2 + \alpha_3(d - i) + \alpha_3(j - e)$, or there is more than one base pair or pseudoknot (with rightmost pair $d \cdot e$) and recursion energy $\alpha_2 + \alpha_3(j - e)$, or there is one pseudoknot contributing two base pairs to the multiloop with recursion energy $\beta^m_1 + 2\alpha_2 + \alpha_3(d - i) + \alpha_3(j - e)$, or there is more than one pseudoknot or base pair (with rightmost pseudoknot filling the subsequence $[d, e]$) and recursion energy $\beta^m_1 + 2\alpha_2 + \alpha_3(j - e)$.

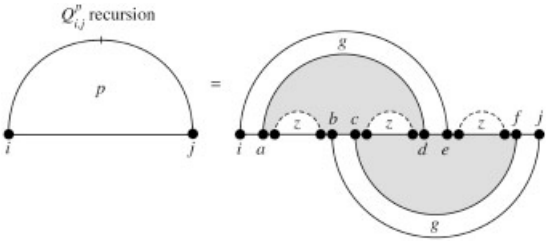

Figure 15. $O(N^8)$ Algorithm: recursion for $Q^p_{i,j}$, the partition function for the pseudoknot filling the subsequence $[i, j]$. The recursion energy is $2\beta_2$, where β_2 is the penalty associated with each base pair bordering the interior of the pseudoknot.

for clarity. The structure of the spanning regions is described by another recursive quantity Q^g, which requires four subscripts and hence $O(N^4)$ storage. The three interior regions of the pseudoknot are depicted as dashed z-curves to indicate that the right and left bases may or may not be paired. The corresponding quantity Q^z is defined by exactly the same recursive process as Q (see Fig. 12) but with recursion energies that reflect the fact that Q^z is inside a pseudoknot.

The gap partition function $Q^g_{i,d,e,j}$ is defined by the recursion in Figure 16. There are two types of cases: either $i \cdot j$ and $d \cdot e$ are the only spanning pairs or there is another outermost spanning pair $c \cdot f$ inside the spanning region. In either case, if there is no additional structure inside the spanning region then an interior loop is formed. If there is at least one additional base pair or pseudoknot

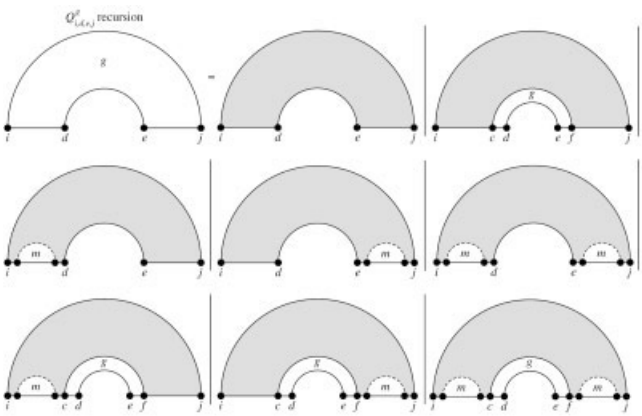

Figure 16. $O(N^8)$ Algorithm: recursion for $Q^g_{i,d,e,j}$, the partition function for the pseudoknot spanning region filling subsequence $[i, j]$ excluding the gap $[d + 1, e - 1]$. There are two types of cases: i. either $i \cdot j$ and $d \cdot e$ are the only spanning base pairs; ii. there is another outermost spanning pair $c \cdot f$ inside the spanning region. An interior loop may be formed (with recursion energy i. $G^{\text{interior}}_{i,d,e,j}$ or ii. $G^{\text{interior}}_{i,c,f,j}$) or else a multiloop is formed due to at least one more base pair or pseudoknot in the spanning region. There may be additional structure to the left of the gap [with recursion energy i. $\alpha_1 + 2\alpha_2 + \alpha_3(j - e - 1)$ or ii. $\alpha_1 + 2\alpha_2 + \alpha_3(j - f - 1)$], to the right of the gap [with recursion energy i. $\alpha_1 + 2\alpha_2 + \alpha_3(d - i - 1)$ or ii. $\alpha_1 + 2\alpha_2 + \alpha_3(c - i - 1)$], or on both sides of the gap (with recursion energy i. or ii. $\alpha_1 + 2\alpha_2$).

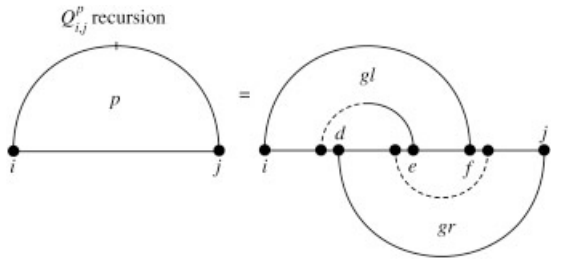

Figure 17. $O(N^5)$ Algorithm: recursion for $Q^p_{i,j}$, the partition function for the pseudoknot filling the subsequence $[i, j]$. There is no recursion energy associated with this diagram.

in the spanning region to the left, right, or on both sides of the gap, then a multiloop is formed inside the spanning region.

The precise mathematical form of the recursions for the $O(N^8)$ algorithm is given in Supplementary Material Figure S5. The computational complexity of the algorithm increases to $O(N^8)$ because it requires the eight indices i, a, b, c, d, e, f, j to specify the structure of the pseudoknot.

The pseudoknot recursion of Figure 15 describes pseudoknots that have precisely two spanning regions (i.e., g-curves). Note that pseudoknots may be nested within the interior (using z-curves) or the spanning regions (using m-curves) of pseudoknots to as many levels as are allowed by the length of the strand. This class of pseudoknots includes 98% of the pseudoknots in the Pseudobase database of known RNA pseudoknots.[15,16] However, this class is somewhat more restrictive than those treated by the redundant structure prediction recursions of Rivas and Eddy[25] and Akutsu[24] (see Supplementary Material Figure S6, for examples).

$O(N^5)$ *Algorithm*

The previous $O(N^8)$ algorithm can be improved to $O(N^5)$ complexity by defining additional intermediate recursions and generalizing the "fastiloops" approach to calculate interior loops inside the spanning regions of pseudoknots. Noting the number of indices on the recursion diagrams in Figures 12–16 or the nesting depth of the loops in the $O(N^8)$ pseudocode of Supplementary Material Figure S5, it is apparent that the two parts of the algorithm that require modification are the calculation of Q^p, which is $O(N^8)$, and the calculation of Q^g, which has four contributions that are $O(N^6)$. The other recursions for Q, Q^b, Q^m and Q^z depicted in Figures 12–14 are $O(N^4)$ and need not be modified. The challenge is to find a way of specifying the pseudoknot internal structure in stages so as to recurse over exactly the same set of nonredundant structures using exactly the same recursion energies.

A new Q^p recursion for the pseudoknot interior is shown in Figure 17. In comparison to Figure 15, the interior z-curve regions of the pseudoknot have been subsumed into left and right gap recursions Q^{gl} and Q^{gr}. Q^{gl} has an outer spanning pair $i \cdot f$ and an inner spanning pair between e and some base in the subsequence $[i + 1, d - 1]$. Q^{gr} has an outer spanning pair $d \cdot j$ and an inner spanning pair with one end in the subsequence $[d + 1, e - 1]$ and the other in the subsequence $[f + 1, j - 1]$. There are now five subscripts corresponding to an $O(N^5)$ complexity.

The recursions for Q^{gl} and Q^{gr} are defined in Figure 18, where the pseudoknot interior regions are specified by introducing z-curves. The definitions of Q^{gr} and Q^{gl} were chosen specifically so that Q^{gr} could recurse to Q^{gl}, which in turn, recurses to Q^g. This approach is more efficient in terms of operation count and storage than alternative formulations in which Q^{gr} recurses to Q^g through a different intermediate quantity.

The new $O(N^5)$ recursion for Q^g is shown in Figure 19. The only cases that require modification are the ones where there is an additional spanning pair. If there is an interior loop, the use of the black-box potential $G^{\text{interior}}_{i,c,f,j}$ would lead to an $O(N^6)$ computational complexity. However, a similar "fastiloops" treatment of "possible extensible loops" can be used to compute these contributions in $O(N^5)$ as detailed in the pseudocode of Supplementary Material Figure S7. For the three multiloop cases where there is an additional spanning pair, we introduce the left and right supplementary gap recursions Q^{gls} and Q^{grs} defined in Figure 20. These recursions define the spanning region using g-curves and introduce the multiloop interior using m-curves.

It is critical to employ these recursions in the correct order so that all quantities are available when needed. Pseudocode describing the mathematical formulation of this $O(N^5)$ algorithm for computing the partition function of a nucleic acid strand is shown in Figure 21. This is the main result of the article.

Methods

The partition function algorithms and minimum energy structure prediction algorithms described in this article were implemented in the C programming language using recursion energies based on standard secondary structure energy models for unpseudoknotted ssDNA and RNA.[12,13] A new pseudoknot energy model was introduced and a preliminary parameterization for RNA is described under Results.

One difference from the published form of the unpseudoknotted RNA energy potential[13] is the exclusion of the special bonus

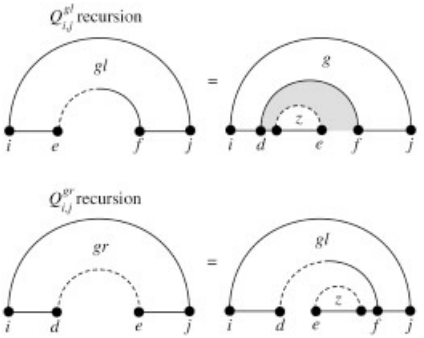

Figure 18. $O(N^5)$ Algorithm: recursions for $Q^{gl}_{i,e,f,j}$ and $Q^{gr}_{i,d,e,j}$, the partition functions for the left and right spanning regions of a pseudoknot. The recursion energy for Q^{gl} is β_2 corresponding to the penalty for base pair $d \cdot f$ bordering the interior of a pseudoknot. There is no recursion energy for Q^{gr}.

for hairpins with GGG on the 5′ side of the stem because this term violates the loop decomposition paradigm. Coaxial stacking terms[13] are also excluded, although these could be incorporated with the same computational complexity by using additional memory. However, from the point of view of partition function redundancy, it is unclear how to treat different coaxial stackings of the same secondary structure.

All other energy terms in the standard model,[13] including dangle energies[28] and penalties for helices not terminated by G · C are included in the implementation. These terms are also applied to the pseudoknot energy model. These details do not change the structure of the recursions described in the pseudocode, and are omitted for clarity. The dangle terms can be implemented exactly without the use of additional recursions if helices are not allowed to terminate with a G · U wobble pair (this is the method used for the results presented here). This case can also be handled exactly at the expense of storing and computing multiple copies of some of the recursive quantities. Another alternative is to allow G · U wobble pairs to terminate helices but to treat the associated dangle energies approximately.

The structure prediction algorithms identify the energy of the most stable structure. Once the minimum energy is known, a separate backtrack routine is used to identify a corresponding minimum energy structure.

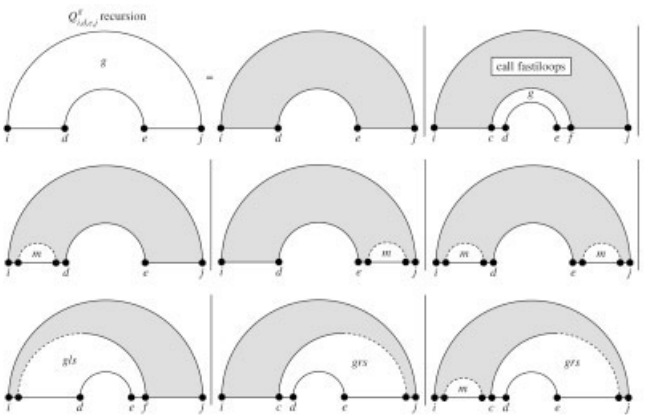

Figure 19. $O(N^5)$ Algorithm: recursion for $Q^g_{i,d,e,j}$, the partition function for the pseudoknot spanning region filling subsequence $[i, j]$ excluding the gap $[d + 1, e - 1]$. There are two types of cases: i. either $i \cdot j$ and $d \cdot e$ are the only spanning base pairs, ii. there is another outermost spanning pair $c \cdot f$ inside the spanning region. All cases of type i. are $O(N^4)$ and the treatment is identical to Figure 16. The contribution for the interior loop case of type ii. may be calculated in $O(N^5)$ using the function "fastiloops" detailed in the pseudocode of Supplementary Material Figure S7. The three other type ii. cases correspond to multiloops with at least one more base pair or pseudoknot in the spanning region. There may be additional structure to the left of the gap [with recursion energy $\alpha_1 + \alpha_2 + \alpha_3(j - f - 1)$], to the right of the gap [with recursion energy $\alpha_1 + \alpha_2 + \alpha_3(c - i - 1)$], or on both sides of the gap (with recursion energy $\alpha_1 + \alpha_2$). The boundaries of the spanning regions are specified in the left and right supplementary gap recursions Q^{gls} and Q^{grs}.

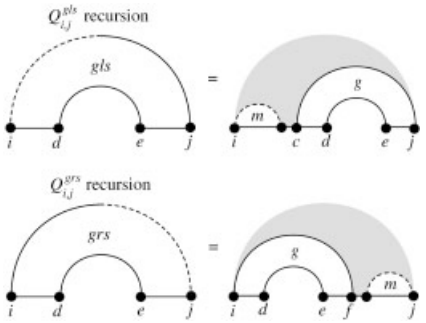

Figure 20. $O(N^5)$ Algorithm: recursions for $Q^{gls}_{i,d,e,j}$ and $Q^{grs}_{i,d,e,j}$, supplementary gap partition functions used in computing Q^g. The recursion energy for both diagrams is α_2, corresponding to the introduction of a base pair bordering the interior of a multiloop.

Results

Pseudoknot Model Parameterization

Although it is not the emphasis of the present work, we provide a preliminary parameterization of our pseudoknot model for RNA secondary structure by suggesting values for β_1, β_1^m, β_1^p, β_2, and β_3. We focus on RNA rather than ssDNA because of the large number of pseudoknotted and unpseudoknotted secondary structures that are known for RNAs. The standard RNA parameters for unpseudoknotted structures are taken directly from mfold3.1 by Zuker and coworkers.[13] In selecting values for the pseudoknot parameters, there are two competing objectives. A negative control monitors the introduction of spurious pseudoknots into structures that are known to be unpseudoknotted. A positive control monitors the correct prediction of pseudoknots in known pseudoknotted structures. For both controls, the selected cases are divided into a "working" set that is used during the parameter search and a "free" set that is used to provide an independent evaluation of the parameters after the search is completed. Model parameterization is performed by comparing the predicted minimum energy structure with the experimentally determined secondary structure.

For the negative control, we rely on a database of over 3000 known tRNA sequences that are believed to be unpseudoknotted based on experimental structures or sequence alignment.[29] From these sequences we randomly select a working set and a free set with 200 sequences each. For the purposes of this study, the only concern of the negative control was whether or not a spurious pseudoknot is predicted in the lowest energy structure. Correct prediction of the unpseudoknotted secondary structures for these tRNAs lies in the purview of the mfold3.1 parameters,[13] which were not altered in this study. Two potential drawbacks to using tRNAs as the negative control are that tRNAs all have roughly the same cloverleaf structure, and that most tRNAs involve modified nucleotides. The first problem may result in a bias towards parameters that preserve cloverleaf structures, while the second may produce errors if the modified nucleotides significantly affect the minimum energy fold. In the future, it would be advantageous if secondary structure information were provided in RNA crystal and NMR structure databases to provide more information for secondary structure studies.

```
Initialize (Q, Q^b, Q^m, Q^p, Q^z) // O(N^2) space
Initialize (Q^g, Q^gl, Q^gr, Q^gls, Q^grs, Q^x, Q^x1, Q^x2) // O(N^4) space
Set all values to 0 except Q_{i,i-1} = Q^z_{i,i-1} = 1
for l = 1, N // examine subsequences of increasing length
  Initialize Q^x = Q^x1, Q^x1 = Q^x2, Q^x2 = 0
  for i = 1, N-l+1
    j = i+l-1
    // Q^b recursion
    Q^b_{i,j} = exp(-G^hairpin_{i,j}/RT)
    for d = i+1, j-5 // all possible rightmost pairs d·e
      for e = d+4, j-1
        Q^b_{i,j} += exp(-G^interior_{i,d,e,j}/RT) Q^b_{d,e}
        Q^b_{i,j} += Q^m_{i+1,d-1} Q^b_{d,e} exp{-[α1 + 2α2 + α3 (j-e-1)]/RT}
    for d = i+1, j-9 // all possible rightmost pseudoknots filling [d, e]
      for e = d+8, j-1
        G^recursion = α1 + β1^m + 3α2 + α3 (j-e-1)
        Q^b_{i,j} += exp{-[G^recursion + α3 (d-i-1)]/RT} Q^p_{d,e}
        Q^b_{i,j} += Q^m_{i+1,d-1} Q^p_{d,e} exp{-G^recursion/RT}
    // Q^g recursion
    for d = i+1, j-5 // set inner pair d·e
      for e = d+4, j-1
        Q^g_{i,d,e,j} += exp(-G^interior_{i,d,e,j}/RT)
    call fastiloops(i, j, l, Q^g, Q^x, Q^x2)
    for d = i+6, j-5
      for e = d+4, j-1
        Q^g_{i,d,e,j} += Q^m_{i+1,d-1} exp{-[α1+2α2+α3(j-e-1)]/RT}
    for d = i+1, j-10
      for e = d+4, j-6
        Q^g_{i,d,e,j} += exp{-[α1+2α2+α3(d-i-1)]/RT} Q^m_{e+1,j-1}
    for d = i+6, j-10
      for e = d+4, j-6
        Q^g_{i,d,e,j} += Q^m_{i+1,d-1} exp{-[α1+2α2]/RT} Q^m_{e+1,j-1}
    for d = i+7, j-6
      for e = d+4, j-2
        for f = e+1, j-1
          Q^g_{i,d,e,j} += Q^gls_{i+1,d,e,f} exp{-[α1+α2+α3(j-f-1)]/RT}
    for d = i+2, j-11
      for e = d+4, j-7
        for c = i+1, d-1
          Q^g_{i,d,e,j} += exp{-[α1+α2+α3(c-i-1)]/RT} Q^grs_{c,d,e,j-1}
    for d = i+7, j-11
      for e = d+4, j-7
        for c = i+6, d-1
          Q^g_{i,d,e,j} += Q^m_{i+1,c-1} Q^grs_{c,d,e,j-1} exp{-[α1 + α2]/RT}
    // Q^gls, Q^grs recursions
    for c = i+5, j-6
      for d = c+1, j-5
        for e = d+4, j-1
          Q^gls_{i,d,e,j} += exp(-α2/RT) Q^m_{i,c-1} Q^g_{c,d,e,j}
    for d = i+1, j-10
      for e = d+4, j-6
        for f = e+1, j-5
          Q^grs_{i,d,e,j} += Q^g_{i,d,e,f} Q^m_{f+1,j} exp(-α2/RT)
    // Q^gl, Q^gr recursions
    for d = i+1, j-5
      for f = d+4, j-1
        for c = d, f-3
          Q^gl_{i,e,f,j} += Q^g_{i,d,f,j} Q^z_{d+1,e} exp(-β2/RT)
    for d = i+1, j-4
      for e = d+3, j-1
        for f = e, j-1
          Q^gr_{i,d,e,j} += Q^gl_{i,d,f,j} Q^z_{e,f-1}
    // Q^p recursion
    for d = i+2, j-4 // set points left to right
      for e = max{d+2, i+5}, j-3
        for f = e+1, j-2
          Q^p_{i,j} += Q^gl_{i,d-1,e,f} Q^gr_{d,e-1,f+1,j}
    // Q, Q^m, Q^z recursions
    Q_{i,j} = 1 //empty recursion
    Q^z_{i,j} = exp(-[β3(j-i+1)]/RT)
    for d = i, j-4 // all possible rightmost pairs d·e
      for e = d+4, j
        Q_{i,j} += Q_{i,d-1} Q^b_{d,e}
        Q^m_{i,j} += exp{-[α2 + α3 (d-i) + α3 (j-e)]/RT} Q^b_{d,e}
        Q^m_{i,j} += Q^m_{i,d-1} Q^b_{d,e} exp{-[α2 + α3 (j-e)]/RT}
        Q^z_{i,j} += Q^z_{i,d-1} Q^b_{d,e} exp{-[β2 + β3(j-e)]/RT}
    for d = i, j-8 // all possible rightmost pseudoknots filling [d, e]
      for e = d+8, j
        Q_{i,j} += Q_{i,d-1} Q^p_{d,e} exp{-β1/RT}
        Q^m_{i,j} += exp{-[β1^m + 2α2 + α3 (d-i) + α3 (j-e)]/RT} Q^p_{d,e}
        Q^m_{i,j} += Q^m_{i,d-1} Q^p_{d,e} exp{-[β1^m + 2α2 + α3 (j-e)]/RT}
        Q^z_{i,j} += Q^z_{i,d-1} Q^p_{d,e} exp{-[β1^p + 2β2 + β3(j-e)]/RT}
//Partition function is Q_{1,N}
```

For the positive control, we draw from a database of 212 RNA pseudoknots, each described as a single stretch of consecutive nucleotides.[15,16] From this sample, we exclude the five structures that contain chains of pseudoknots that cannot be modeled using the present recursions (see, e.g., Supplementary Material Fig. S6). From the remaining sequences, we randomly select a working set and a free set of 100 pseudoknots each. To determine if the predicted and experimental structures are equivalent, the structures are reduced to a basic pseudoknot topology. This is accomplished by considering only the base pairs in the spanning regions that define the pseudoknot (i.e., the pairs in the Q^g recursion that span the gap). If both the experimental and the predicted structures have the same pseudoknot topology, and the end of each spanning region in the predicted structure overlaps with the corresponding region in the experimental structure, then the prediction is considered a match.

As a starting point, we began by interpolating the parameters of a recent pseudoknot model[30] to obtain the partial specification: $\beta_1 + 2\beta_2 = 9.6$ and $\beta_3 = 0.15$. After a search of the nearby parameter space we selected the values

$$\beta_1 = 9.6, \quad \beta_1^m = \beta_1^p = 15.0, \quad \beta_2 = 0.1, \quad \beta_3 = 0.1.$$

The success rates for the working set and the free set for both the negative and the positive controls are summarized in Table 1. The $O(N^5)$ structure prediction algorithm avoids introducing spurious pseudoknots in 92% of the negative controls and correctly predicts 61% of the pseudoknots in the positive controls. By comparison, running Rivas and Eddy's structure prediction algorithm with their parameters[25] on the same sequences, there are no spurious pseudoknots predicted for 98% of the negative controls, and the correct pseudoknots are predicted for 43% of the positive controls. Our experience suggests that for our model, there is a clear tradeoff between avoiding spurious pseudoknots and predicting correct ones. We chose to balance our parameters so as to obtain as many correct pseudoknots as possible while maintaining at least a 90% rate on the negative control.

Additional assistance in parameterizing the pseudoknot model using computational or experimental studies would be most welcome. Modifications to the formulation of the pseudoknot energy expression can be accommodated to the extent that the dynamic programming framework allows.

Algorithm Complexity

The computational complexity of all four partition function algorithms is demonstrated empirically in Figure 22. The $O(N^4)$ and

Figure 21. Pseudocode implementation of an $O(N^5)$ dynamic programming partition function algorithm for nucleic acids with pseudoknots. Here, N is the length of the strand and $l = j - i + 1$ is the length of the substrand under consideration at any given point during the recursive process. The recursions are described schematically in Figures 12–14 and 17–20. The function "fastiloops" computes certain interior loop contributions to Q^g in $O(N^5)$ as detailed in the pseudocode of Supplementary Material Figure S7.

Table 1. RNA Pseudoknot Parameterization.

	Negative control			Positive control		
Model	Working	Free	Overall	Working	Free	Overall
Dirks & Pierce	179/200	187/200	92%	60/100	61/100	61%
Rivas & Eddy[25]	195/200	197/200	98%	40/100	46/100	43%

$O(N^3)$ algorithms excluding pseudoknots and the $O(N^8)$ and $O(N^5)$ algorithms including pseudoknots are each tested on three random sequences for each of the depicted sequence lengths. The slopes of the least-squares fits to this data closely follow the expected theoretical complexity estimates with the exception of the $O(N^5)$ algorithm, which scales somewhat worse than the expected estimate. This effect occurs because some of the nested loops that dominate the software execution time are shorter than length N, so the slope is greater than the predicted slope for small N. The complexity estimates for all four algorithms should increase in accuracy as the length of the test molecules increases.

To illustrate the impact of lower computational complexity, note that calculation of the unpseudoknotted partition function for a sequence of length 500 requires roughly 120 s using the $O(N^3)$ algorithm and 2900 s using the $O(N^4)$ method. Shorter test molecules must be considered for the algorithms that incorporate pseudoknots. For a sequence of length 100, the $O(N^5)$ algorithm requires roughly 80 s while the $O(N^8)$ algorithm requires roughly 1200 s. By comparison, the $O(N^6)$ structure prediction algorithm of Rivas and Eddy[26] runs in approximately 2300 s on sequences of length 100. Complexity benchmarks were performed on a 700MHz Pentium Xeon processor.

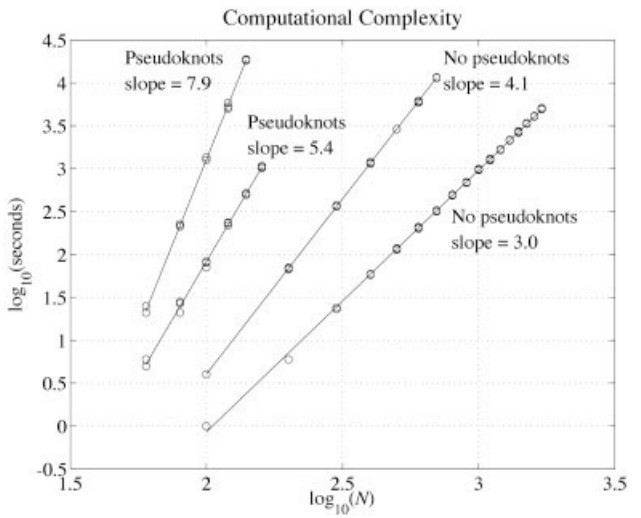

Figure 22. Comparison of the observed computational complexity for $O(N^4)$ and $O(N^3)$ partition function algorithms without pseudoknots and $O(N^8)$ and $O(N^5)$ partition function algorithms that include pseudoknots. Each algorithm is run on three randomly selected sequences for each of the depicted sequence lengths N. Timings are performed on a 700MHz Pentium Xeon processor.

Sequence Design

In designing nucleic acid structures for nanotechnology applications, the objective is to select sequences that adopt the desired secondary structure with both high affinity and high specificity. Currently, most designs are performed using sequence symmetry minimization (SSM),[27] an approach that has proven quite useful for designing branched structures[31] including DNAs with the connectivity of a cube[32] and a two-dimensional DNA crystal lattice.[17] SSM attempts to ensure specificity by prohibiting repeated subsequences (of a specified length) in double-stranded regions of the target graph and by prohibiting repeated subsequences and their complements in single-stranded or branched regions of the target graph. This simple and flexible strategy can be employed for single or multiple strands with or without pseudoknots. Affinity is optimized only weakly by ensuring compatibility with the base-pairing graph so it is unclear on theoretical grounds whether SSM should produce sequences that adopt the desired graph with high probability.

The partition function provides an ideal framework for evaluating the performance of design algorithms. At equilibrium, the probability of sampling a secondary structure s with energy G_s may be obtained from the partition function Q using

$$p(s) = \exp(-G_s/RT)/Q. \tag{17}$$

If the probability of adopting the desired graph is close to unity, then within the context of the approximate physical model, the sequence design achieves both high affinity and high specificity for the target graph. This observation suggests that direct optimization of the sequence so as to maximize the probability of folding to the target graph represents an attractive design strategy.[10,11] In the past, this approach has only been applicable to the design of single strands without pseudoknots due to the inability to compute the partition function for more general cases.

Consider the sequence selection problem for the RNA multiloop of Figure 23. It is an interesting question as to whether it is possible to design a sequence that adopts this entropically unfavorable structure with high probability. In Figure 23, we compare the probabilities of folding to the target graph for 1000 sequences that satisfy the base pairing graph and are either random or satisfy SSM for subsequences of length four. These probabilities were computed using the $O(N^5)$ partition function algorithm including pseudoknots. Approximately 94% of the random sequences and 87% of the SSM sequences fold to the desired secondary structure with less than 10% probability. The best random design has a probability of 61% and the best SSM design has a probability of

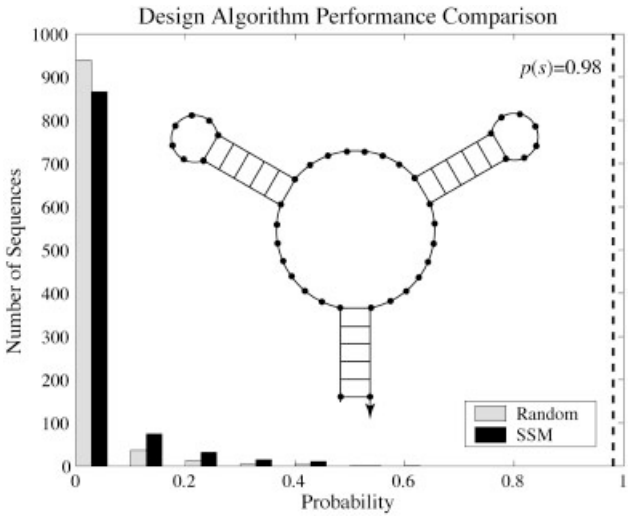

Figure 23. RNA multiloop design test case with 64 nucleotides. The multiloop comprises three closing base pairs separated by single stranded regions of length six. The three stems contain six base pairs each and the hairpins are of length five. The partition function is used to evaluate the probability of folding to the desired secondary structure for 1000 sequences that satisfy the base pairing graph and are either random or satisfy perfect sequence symmetry minimization with a word length of four. The highest probability achieved by a random sequence is 61%, and the highest probability achieved by an SSM sequence is 51%. Direct optimization of the probability using an adaptive walk leads to a single sequence with a probability of folding to the desired graph of 98%.

51%. Note that this performance assessment is based on the stringent criterion that every nucleotide must exactly match the target graph. In practice, molecules may still have utility even if the secondary structure deviates to some degree from the desired target.

To attempt to determine whether it is possible to design a sequence that samples this multiloop structure with high probability, we used an adaptive walk in sequence space to optimize the probability directly. The sequence at a randomly selected position was subjected to a random perturbation (matched by a compatible mutation to a paired base) and the move was accepted if the partition function indicated an increased probability of folding to the target graph. After only a few hundred iterations, this procedure yielded a sequence with a 98% probability of folding to the intended secondary structure.

If the probabilities are instead evaluated using the standard energy model[13] and the $O(N^3)$ partition function algorithm without pseudoknots, there is virtually no change in the histogram. The maximum probabilities achieved by random and SSM sequences remain 61% and 51%, respectively. The sequence obtained by an adaptive walk maintains a probability of 98%.

As another interesting design test case, consider the RNA pseudoknot of Figure 24. Approximately 98% of random and 93% of SSM designs have less than a 10% probability of folding to the target structure. The best random and SSM designs have probabilities of 42 and 44%, respectively. Using an adaptive walk and repeated evaluation of the partition function, we obtained a sequence that folds to the target graph with 98% probability.

Conclusions

We describe a nonredundant dynamic programming algorithm that computes the partition function of an RNA or ssDNA strand over secondary structure space. For the first time, this space is extended to include the most physically relevant pseudoknots. The algorithm has a time complexity of $O(N^5)$ and requires $O(N^4)$ memory, where N is the length of the strand. An algorithm for identifying the minimum energy structure is obtained by a straightforward modification of the partition function algorithm. A preliminary parameterization of the model is performed for RNA using 200 RNA pseudoknots and 400 unpseudoknotted tRNAs.

The partition function is useful for studying the conformational ensembles of both naturally occuring and designed nucleic acid sequences. Tests on an RNA multiloop and an RNA pseudoknot indicate that sequences designed using the standard approach of sequence symmetry minimization tend to adopt the target base pairing graph with low probability. Sequences designed by direct optimization of the probability demonstrated high affinity and specificity for the target secondary structures. These conclusions are subject to the limitations of the approximate physical model on which the partition function is based. Future work will explore the

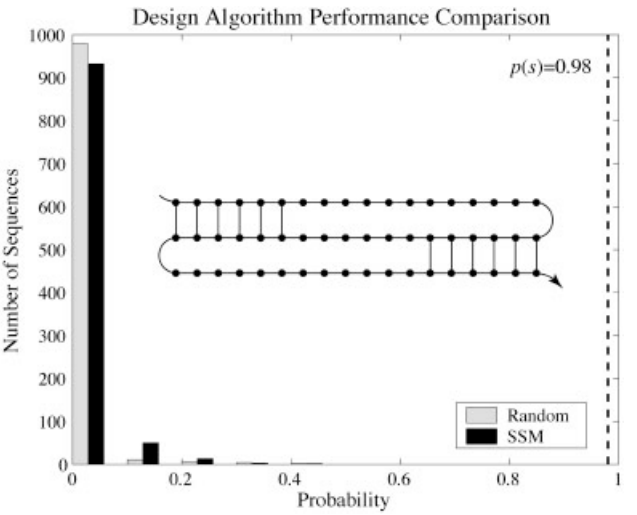

Figure 24. RNA pseudoknot design test case with 54 nucleotides. The pseudoknot comprises two stems of length six with two interior single stranded regions of length 12 and one of length six. The partition function is used to evaluate the probability of folding to the desired secondary structure for 1000 sequences that satisfy the base pairing graph and are either random or satisfy perfect sequence symmetry minimization with a word length of four. The highest probability achieved by a random sequence is 42%, and the highest probability achieved by an SSM sequence is 44%. Direct optimization of the probability using an adaptive walk leads to a single sequence with a probability of folding to the desired graph of 98%.

experimental behavior of sequences designed using the partition function.

Acknowledgments

We wish to thank Dr. E. Winfree, our close collaborator in the ongoing effort to design and build functional nucleic acid systems, for many interesting discussions during the course of this work.

References

1. Waterman, M. S. In Studies in Foundations and Combinatorics: Advances in Mathematics Supplement Studies; Academic Press: New York, 1978; p 167, vol. 1.
2. Waterman, M. S.; Smith, T. F. Math Biosci 1978, 42, 257.
3. Nussinov, R.; Pieczenik, J. R.; Griggs, J. R.; Kleitman, D. J. SIAM J Appl Math 1978, 35, 68.
4. Zuker, M.; Stiegler, P. Nucleic Acids Res 1981, 9, 133.
5. Zuker, M.; Sankoff, D. Bull Math Biol 1984, 46, 591.
6. Turner, D. H.; Sugimoto, N. Annu Rev Biophys Biophys Chem 1988, 17, 167.
7. Zuker, M. Curr Opin Struct Biol 2000, 10, 303.
8. McCaskill, J. S. Biopolymers 1990, 29, 1105.
9. Bonhoeffer, S.; McCaskill, J. S.; Stadler, P. F.; Schuster, P. Eur Biophys J 1993, 22, 13.
10. Hofacker, I. L.; Fontana, W.; Stadler, P. F.; Bonhoeffer, L. S.; Tacker, M.; Schuster, P. Chem Monthly 1994, 125, 167.
11. Flamm, C.; Hofacker, I. L.; Maurer–Stroh, S.; Stadler, P. F.; Zehl, M. RNA 2001, 7, 254.
12. SantaLucia, J., Jr. Biochemistry 1996, 35, 3555.
13. Mathews, D. H.; Sabina, J.; Zuker, M.; Turner, D. H. J Mol Biol 1999, 288, 911.
14. Tinoco, I., Jr.; Uhlenbec, D. C.; Levine, M. D. Nature 1971, 230, 362.
15. van Batenburg, F. H. D.; Gultyaev, A. P.; Pleij, C. W. A.; Ng, J. Nucleic Acids Res 2000, 28, 201.
16. van Batenburg, F. H. D.; Gultyaev, A. P.; Pleij, C. W. A. Nucleic Acids Res 2001, 29, 194.
17. Winfree, E.; Liu, F.; Wenzler, L. A.; Seeman, N. C. Nature 1998, 394, 539.
18. Tabaska, J. E.; Cary, R. B.; Gabow, H. N.; Stormo, G. D. Bioinformatics 1998, 14, 691.
19. Chen, J. H.; Le, S. Y.; Maizel, J. V. Comput Appl Biosci 1992, 8, 243.
20. Bouthinon, D.; Soldano, H. Bioinformatics 1999, 15, 785.
21. Uemura, Y.; Hasegawa, A.; Kobayashi, S.; Yokomori, T. Theor Comput Sci 1999, 210, 277.
22. Isambert, H.; Siggia, E. D. Proc Natl Acad Sci USA 2000, 97, 6515.
23. Lyngso, R. B.; Pedersen, C. N. S. J Comput Biol 2000, 7, 409.
24. Akutsu, T. Discrete Appl Math 2000, 104, 45.
25. Rivas, E.; Eddy, S. R. J Mol Biol 1999, 285, 2053.
26. Lyngso, R. B.; Zuker, M.; Pedersen, C. N. S. Bioinformatics 1999, 15, 440.
27. Seeman, N. C. J Theor Biol 1982, 99, 237.
28. Serra, M. J.; Turner, D. H. Methods Enzymol 1995, 259, 242.
29. Sprinzl, M.; Horn, C.; Brown, M.; Ioudovitch, A.; Steinberg, S. Nucleic Acids Res 1998, 26, 148.
30. Gultyaev, A. P.; van Batenburg, F. H. D.; Pleij, C. W. A. RNA 1999, 5, 609.
31. Seeman, N. C. Trends Biotechnol 1999, 17, 437.
32. Chen, J.; Seeman, N. C. Nature 1991, 350, 631.