ANALYSIS, DESIGN, AND CONSTRUCTION OF NUCLEIC ACID DEVICES

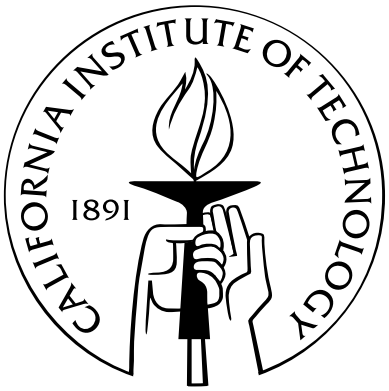
Thesis by

Robert M. Dirks

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2005

(Defended 11 May 2005)

Acknowledgments

First and foremost I would like to thank my advisor, Niles Pierce, and the rest of my lab-mates for creating a wonderful learning environment that fosters teamwork, creativity, and critical thinking. All of the work in this thesis would not have been possible without them.

I would also like to thank my committee, Steve Mayo, Bill Goddard, Nate Lewis, and Erik Winfree, for all of their constructive criticism and suggestions along the way. In addition, the following members of the Mayo lab have been especially helpful in providing both experimental and computational tips: Rhonda DiGiusto, Geoff Hom, Premal Shah and Possu Huang. From the Winfree lab, Paul Rothemund and Joe Schaeffer have been wonderful sources of good ideas and suggestions about all things DNA. My parents Mike and Suree, and my brother Bill, also deserve praise for the support they have given me throughout my life. Finally, I would like to acknowledge my fiancée, Christine Ueda, for helping with experiments, listening to me rant and rave on a wide range of topics, and for filling these last few years with tremendous joy. Thank you all!

Abstract

Nucleic acids present great promise as building blocks for nanoscale devices. To achieve this potential, methods for the analysis and design of DNA and RNA need to be improved. In this thesis, traditional algorithms for analyzing nucleic acids at equilibrium are extended to handle a class of pseudoknots, with examples provided relevant to biologists and bioengineers. With these analytical tools in hand, nucleic acid sequences are designed to maximize the equilibrium probability of a desired fold. Upon analysis, it is concluded that both affinity and specificity are important when choosing a sequence; this conclusion holds for a wide range of target structures and is robust to random perturbations to the energy model. Applying the intuition gained from these studies, a process called hybridization chain reaction (HCR) is invented, and sequences are chosen that experimentally verify this phenomenon. In HCR, a small number of DNA or RNA molecules trigger a system wide configurational change, allowing the amplification and detection of specific, nucleic acid sequences. As an extension, HCR is combined with a pre-existing aptamer domain to successfully construct an ATP sensor, and the groundwork is laid for the future development of sensors for other small molecules. In addition, recent studies on multi-stranded algorithms and improvements to HCR are included in the appendices. Not only will these advancements increase our understanding of biological RNAs, but they will also provide valuable tools for the future development of nucleic acid nanotechnologies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Nucleic Acid Based Devices

Since Ned Seeman's early work on the creation of synthetic DNA shapes and patterns [27, 29, 11, 30], nucleic acids have emerged as a powerful and facile tool for the development of nanoscale, biocompatible technologies. With utility as both programmable scaffolds [28, 27, 4, 15, 42, 12, 43, 16, 17, 32] and dynamic, functional motifs [45, 44, 14, 35, 2, 33, 31, 34], DNA and RNA have potential applications as biosensors [22, 9], drug delivery systems [18], molecular transporters [34, 31], and "lab on a chip" [23] technologies. In addition to the well-established chemical and enzymatic methods for manipulating DNA and RNA, one key reason nucleic acids are popular components for engineered nanodevices is that reasonably accurate and efficient computational methods exist [47, 38, 46, 20] for predicting and analyzing basic DNA and RNA structures. These algorithms allow for the rapid, *in silico* design and improvement of nucleic acids before synthesizing and testing them in the laboratory. Described in this thesis are improvements to these algorithms (chapters 2 and 3), followed by a detailed study of nucleic acid sequence design (chapter 4). This culminates in the creation and experimental validation of DNA devices (chapter 5) that can detect specific nucleic acid sequences, as well as small molecules such as ATP.

1.1 Structural Hierarchy of Nucleic Acids

A nucleic acid can be described in many different ways. For a given problem, choosing the proper level of detail is crucial in distilling key features without being inundated with extraneous information. Three useful representations of nucleic acids are *primary, secondary,* and *tertiary* structure; the strengths and weaknesses of each viewpoint will be described below. While the physical properties, chemical composition, and biological roles of nucleic acids are also important, a description of such details is not necessary to understand the main thrust of this thesis. For a more complete appreciation of the physical, chemical, and biological contexts of nucleic acids, see Appendix A.

1.1.1 Primary Structure

The primary structure of a nucleic acid is simply the sequence, typically given in the 5' to 3' direction. For DNA, sequences are combinations of the four bases: adenine (A), cytosine (C), guanine (G) and thymine (T). In RNA, thymine is replaced by uracil (U). Throughout this thesis, a sequence, such as `CGTGACCCAG`, always represents a single strand as opposed to the double helix normally imagined when referring to DNA (see Appendix A). Primary structure can be incredibly useful, especially when describing the proteins that nucleic acids often encode. However, this level of detail ignores the folded structure of the DNA or RNA, and hence is insufficient for the purposes of this thesis.

1.1.2 Secondary Structure

Numbering the bases of a nucleic acid strand from 1 to N (where N is the sequence length), the secondary structure is simply a list of base pairs between Watson-Crick complements (A·U, C·G for RNA and A·T, C·G for DNA) or wobble pairs (G·U or G·T), where each number appears in at most one pair. These base pairs dominate the energetics of folding for a nucleic acid and provide a rough schematic of the shape of the molecule. Figure 1.1 shows two different representations of a secondary structure, and depicts the canonical base pairing patterns that can arise. It is this level of description that can be efficiently computed and analyzed, providing an *in silico* feedback loop for the development of nucleic acid technologies.

An important secondary structure motif in both engineered DNA nanostructures [42, 43] and biologically important RNAs [39] is the pseudoknot (see figure 1.2). Traditional algorithms [47, 38, 46, 20] exclude pseudoknots (although some do attempt to handle them [1, 25]), making it difficult to rigorously analyze, predict and design pseudoknots. Since McCaskill's original work in 1990 on the equilibirum properties of non-pseudoknotted structures, little to no progress was made on the incorporation of pseudoknots into these calculations until 2003 and 2004, when the papers [7, 8] described in chapters 2 and 3 were published.

Due to the strength of base pairing interactions, secondary structure is often sufficient to describe a useful level of functionality for a DNA or RNA nanostructure. In particular, unpaired bases are often free to hybridize with other nucleic acids and hence represent active sites. In contrast, base paired positions are protected and are much less likely to hybridize with a neighboring RNA or DNA. This easy to describe structure-function relationship makes secondary structure descriptions the best choice for many nucleic acid applications, and was essential in creating the biosensor described in chapter 5.

1.1.3 Tertiary Structure

For the purposes of this thesis, all structural information beyond simple base pairs is lumped into tertiary structure. This includes how different elements of secondary structure orient themselves with respect to one another and describes the overall three-dimensional geometry of the molecule. This also describes how an RNA or DNA aptamer [13] binds to a substrate or how a ribozyme [24] catalyzes chemical reactions. Clearly, for a truly *de novo* design of nucleic acid devices, this level of description is essential in creating diverse functionalities. However, the computational advantages that make nucleic acids especially easy to analyze and design do not yet extend to molecular modeling. Instead, more detailed calculations that take into account such factors as hydrophobic packing, electrostatics, and solvent interactions are required. To take advantage of these atomic level interactions in nucleic acid devices, functional modules can be borrowed from natural ribozymes or from *in vitro* selection experiments [37, 10], and incorporated into a design schematic [22, 9] without directly addressing the

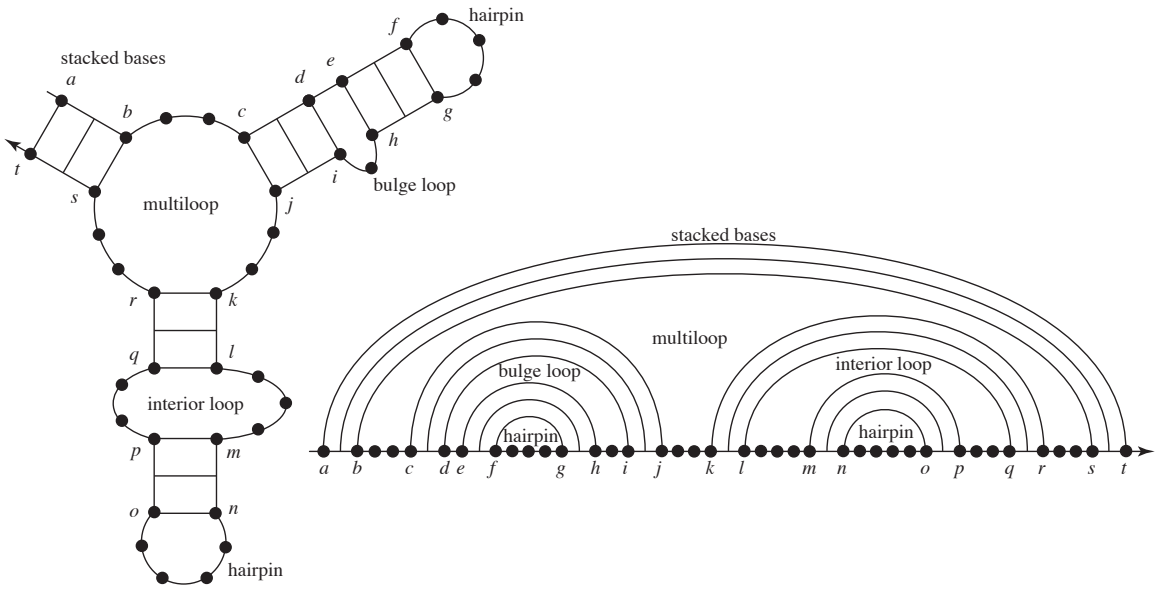


Figure 1.1. Canonical loops of nucleic acid secondary structure: hairpin loops (one closing base pair), stacked base pairs (two closing base pairs with both loop sides of length zero), a bulge loop (two closing base pairs with one loop side of length greater than zero), an interior loop (two closing base pairs with both loop sides of length greater than zero), and a multiloop (more than two closing base pairs). On the right is a polymer graph representation of the same secondary structure, with base pairs represented by arced lines and the polymer backbone drawn as a straight line. The loops in the secondary structures are all nested so there are no crossing arcs.

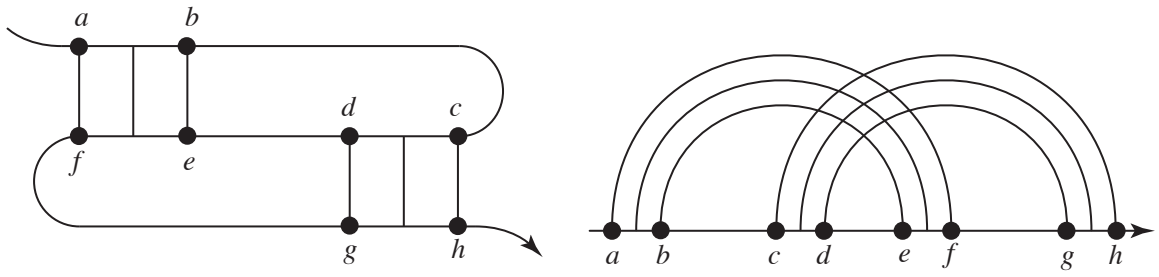


Figure 1.2. A sample pseudoknot with base pairs $a \cdot f$ and $c \cdot h$ (with $a < c$) that fail to satisfy the nesting property $a < c < h < f$. This leads to crossing arcs in the polymer graph. The letters a, c, h, and f represent the numbers associated with each base, counting upwards in the 5' to 3' direction.

tertiary structure that explains how the sequence operates. The computational challenge of directly modeling tertiary structure makes this level of description, while important, not currently necessary for the development of new technologies.

1.2 Thesis Overview

The research done for this thesis can be divided into three distinct categories: analysis, design, and construction.

1.2.1 Analysis

First, computational tools for the analysis of nucleic acid secondary structures will be described. Emphasis is placed on the rigorous incorporation of a class of pseudoknots [7] into both minimum free energy and partition function algorithms [41, 40, 21, 48, 47, 38, 46, 20], while maintaining the fidelity of the nearest neighbor energy model [36, 26, 19]. This is a challenge that had been unsolved since the original algorithms in the late 1970s and early 1980s. These dynamic programming methods can completely explore an exponentially large class of secondary structures in polynomial time, where both the size of the class and the computational cost grow with respect to the length of the sequence. With these algorithms, the minimum free energy structure [47, 38, 46], as well as the partition function [20] describing all possible folds, can be determined for a single stranded RNA or DNA. This allows the equilibrium population of any structure or base pair to be calculated and provides insight into the stability of the dominant fold, as well as any competing configurations [20, 3].

Using the sparse existing data on known pseudoknots, the energy model used for these dynamic programming algorithms has been roughly parameterized [7]. Subsequent applications to both a biologically relevant pseudoknot, as well as an engineered one, demonstrate the utility of these methods. In addition, the algorithms are described in a general way [8] to easily accommodate future modifications.

Work has also been done to analyze the free energy landscapes of multiple, interacting strands

of DNA, and will be published shortly [5]. This work is described in Appendix D.

1.2.2 Design

For a series of target secondary structures, a collection of different design criteria are used to generate sequences. The quality of these sequences is then compared [6] using the partition function analysis developed previously. From these comparisons, it is concluded that both affinity and specificity are important in designing sequences with optimal thermodynamic properties. To determine the robustness of these results, random perturbations are introduced to the energy model parameters and the sequences are reevaluated [6]. The same qualitative trends are observed even as the parameters are varied by as much as 50%. These results should significantly affect how researchers in the field create sequences for nucleic acid devices.

In addition to the thermodynamic properties of the designed sequences, the folding kinetics of these sequences are explored with stochastic simulations. No clear correlation is observed between favorable equilibrium probabilities and fast folding, suggesting that kinetics and thermodynamics are independent issues when designing sequences [6].

1.2.3 Construction

In the final chapter, a DNA system will be described that can undergo a substantial configurational change upon the introduction of a small amount of target DNA (or RNA). This change, termed hybridization chain reaction (HCR), can be easily detected experimentally [9], thereby acting as a sensor for a DNA or RNA signal. By incorporating a pre-existing aptamer into the system, the sensor can be adapted to detect other small molecules. The sequences for HCR were chosen using the analysis tools and design criteria described in the corresponding sections. More sophisticated versions of HCR still under investigation are described in Appendix F. When fully developed, the invention of HCR may prove to be a valuable tool in the creation of practical, nucleic acid technologies.

Bibliography

[1] T. Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104:45–62, 2000.

[2] P. Alberti and J.-L. Mergny. DNA duplex-quadruplex exhange as the basis for a nanomolecular machine. *Proceedings of the National Academy of Sciences of the United States of America*, 100:1569–1573, 2003.

[3] S. Bonhoeffer, J. S. McCaskill, P. F. Stadler, and P. Schuster. RNA multi-structure landscapes. *European Biophysics Journal*, 22:13–24, 1993.

[4] J. Chen and N. C. Seeman. The synthesis from DNAs of a molecule with the connectivity of a cube. *Nature*, 350:631–633, 1991.

[5] R. M. Dirks, J. S. Bois, J. Schaeffer, E. Winfree, and N. A. Pierce. Thermodynamic analysis of multi-stranded nucleic acid systems. *In prep.*, 2005.

[6] R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–1403, 2004.

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

[8] R. M. Dirks and N. A. Pierce. An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots. *Journal of Computational Chemistry*, 25:1295–1304, 2004.

[9] R. M. Dirks and N. A. Pierce. Triggered amplification by hybridization chain reaction. *Proceedings of the National Academy of Sciences of the United States of America*, 101(43):15275–15278, 2004.

[10] A. D. Ellington and J. W. Szostak. In vitro selection of RNA molecules that bind specific ligands. *Nature*, 346:818–822, 1990.

[11] T.-J. Fu and N. C. Seeman. DNA double-crossover molecules. *Biochemistry*, 32:3211–3220, 1993.

[12] T. H. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J. H. Reif, and N. C. Seeman. Construction, analysis, ligation and self-assembly of DNA triple crossover complexes. *Journal of the American Chemical Society*, 122:1848–1869, 2000.

[13] J. F. Lee, J. R. Hesselberth, L. A. Meyers, and A. D. Ellington. Aptamer database. *Nucleic Acids Research*, 32:D95–D100, 2004.

[14] J. J. Li and W. Tan. A single DNA molecule nanomotor. *Nano Letters*, 2(4):315–318, 2002.

[15] X. Li, X. Yang, J. Qi, and N. C. Seeman. Antiparallel DNA double crossover molecules as components for nanoconstruction. *Journal of the American Chemical Society*, 118:6131–6140, 1996.

[16] D. Liu, S. H. Park, J. H. Reif, and T. H. LaBean. DNA nanotubes self-assembled from triple-crossover tiles as templates for conductive nanowires. *Proceedings of the National Academy of Sciences of the United States of America*, 101(3):717–722, 2004.

[17] D. Liu, M. Wang, Z. Deng, R. Walulu, and C. Mao. Tensegrity: Construction of rigid DNA triangles with flexible four-arm DNA junctions. *Journal of the American Chemical Society*, 126:2324–2325, 2004.

[18] Z. Ma and J.-S. Taylor. Nucleic acid-triggered catalytic drug release. *Proceedings of the National Academy of Sciences of the United States of America*, 97:11159–11163, 2000.

[19] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940, 1999.

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

[21] R. Nussinov, J. R. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algorithms for loop matchings. *SIAM Journal of Applied Mathematics*, 35:68–82, 1978.

[22] R. Nutiu and Y. Li. Structure-switching signaling aptamers. *Journal of the American Chemical Society*, 125:4771–4778, 2003.

[23] M. C. Pirrung. How to make a DNA chip. *Angewandte Chemie-International Edition*, 41(8):1277, 2002.

[24] H. Porta and P. M. Lizardi. An allosteric hammerhead ribozyme. *Bio/Technology*, 13:161–164, 1995.

[25] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285:2053–2068, 1999.

[26] J. Santalucia Jr. Improved nearest-neighbor parameters for predicting DNA duplex stability. *Biochemistry*, 35:3555–3562, 1996.

[27] N. C. Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Bioogy.*, 99:237–247, 1982.

[28] N. C. Seeman. DNA in a material world. *Nature*, 421:427–431, 2003.

[29] N. C. Seeman and R. K. Kallenbach. Design of immobile nucleic acid junctions. *Biophysical Journal*, 44:201–209, 1983.

[30] N. C. Seeman, Y. Zhang, and J. H. Chen. DNA nanoconstructions. *Journal of Vacuum Science and Technology a-Vacuum Surfaces and Films*, 12(4):1895–1903, 1994.

[31] W. B. Sherman and N. C. Seeman. A precisely controlled DNA biped walking device. *Nano Letters*, 4(7):1203–1207, 2004.

[32] W. M. Shih, J. D. Quispe, and G. F. Joyce. A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature*, 427:618–621, 2004.

[33] J.-S. Shin and N. A. Pierce. Rewritable memory by controllable nanopatterning of DNA. *Nano Letters*, 4(5):905–909, 2004.

[34] J.-S. Shin and N. A. Pierce. A synthetic DNA walker for molecular transport. *Journal of the American Chemical Society*, 126:10834–10835, 2004.

[35] F. C. Simmel and B. Yurke. A DNA-based molecular device switchable between three distinct mechanical states. *Applied Physics Letters*, 80(5):883–885, 2002.

[36] I. Tinoco Jr., O. C. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[37] C. Tuerk and L. Gold. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage t4 DNA polymerase. *Science*, 249(4968):505–510, 1990.

[38] D. H. Turner, N. Sugimoto, and S. M. Freier. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17:167–192, 1988.

[39] F. H. D. van Batenburg, A. P. Gultyaev, C. W. A. Pleij, and J. Ng. Pseudobase: a database with RNA pseudoknots. *Nucleic Acids Research*, 28:201–204, 2000.

[40] M. S. Waterman. Secondary structure of single-stranded nucleic acids. In *Studies in foundations and combinatorics: Advances in Mathematics Supplemental Studies*, volume 1, pages 167–212. Academic Press, New York, 1978.

[41] M. S. Waterman and T. F. Smith. RNA secondary structure: a complete mathematical analysis. *Mathematical Biosciences*, 42:257–266, 1978.

[42] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

[43] H. Yan, T. H. LaBean, L. Feng, and J. H. Reif. Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices. *Proceedings of the National Academy of Sciences of the United States of America*, 100(14):8103–8108, 2003.

[44] H. Yan, X. Zhang, Z. Shen, and N. C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415(6867):62–5, 2002.

[45] B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, 2000.

[46] M. Zuker. Calculating nucleic acid secondary structure. *Current Opinion in Structural Biology*, 10:303–310, 2000.

[47] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, 1984.

[48] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–147, 1981.

Chapter 2

Analysis of Secondary Structures

The work presented here is heavily based on:

R. M. Dirks and N. A. Pierce, *A partition function algorithm for nucleic acid secondary structure including pseudoknots.* Journal of Computational Chemistry, 2003. **24**(13): pp. 1664-1677. Reprinted with copyright permissions.

2.1 Abstract

Nucleic acid secondary structure models usually exclude pseudoknots due to the difficulty of treating these non-nested 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. An $O(N^5)$ dynamic programming algorithm is described, 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.

2.2 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 $i \cdot j$ and is described in more detail in Chapter 1. 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 [24, 25] and Nussinov et al. [13]. In 1981, Zuker and Stiegler [29] 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 [28, 20, 27]. In 1990, McCaskill [12] 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 [29, 12]. The partition function can be used to derive thermodynamic properties of the equilibrium conformational ensemble including the base-pairing probability of any two bases [12, 2]. 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 [7, 5].

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 [15, 11]. Starting with the work of Tinoco [19], the development of these physical models has involved the work of many researchers [28, 20, 27]. The canonical loop types are illustrated in figure 1.1, 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 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 1.2. 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 [23]. Currently, pseudoknots have been identified in over 200 naturally occurring RNAs, as cataloged in the Pseudobase database [22]. Pseudoknotted structures also arise in engineering efforts to design new molecular structures and machines using nucleic acids [26].

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 [29] 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, in order 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 non-loop-based energy model. Heuristics have also been used to include some pseudoknots in structure searches based on the standard energy model [4, 3, 21, 8]. 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 [9] and Akutsu [1] 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 [14] 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 [29], running in $O(N^6)$ time, but it does successfully capture many possible pseudoknots. Akutsu [1] 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 chapter 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 [12]. With the exception of interior loop terms, this algorithm can be reduced to $O(N^3)$ following ideas presented by McCaskill [12]. However, Lyngso, Zuker and Pedersen [10] 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. Finally, we perform a preliminary parameterization of our pseudoknot model for RNA using 200 known RNA pseudoknots and 400 unpseudoknotted tRNAs.

2.3 Partition Function without Pseudoknots

2.3.1 Standard Energy Model

In the standard energy model for unpseudoknotted secondary structures [15, 11], 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

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

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

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

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

$$G_{i,j}^{\text{empty}} = 0. \tag{2.1}$$

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

$$G_{i,j}^{\text{hairpin}} \tag{2.2}$$

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

$$G_{i,d,e,j}^{\text{interior}} \tag{2.3}$$

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

$$G^{\text{multi}} = \alpha_1 + \alpha_2 B + \alpha_3 U \tag{2.4}$$

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 appendix figure B.1. 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.

2.3.2 $O(N^4)$ Algorithm

To determine the partition function Q for an unspeudoknotted strand of length N, McCaskill's algorithm [12] 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 $Q_{i,j}$, $Q^b_{i,j}$, and $Q^m_{i,j}$ 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

$$Q_{i,j} = 1 + \sum_{\substack{d,\,e \\ i \le d < e \le j}} Q_{i,d-1}\, Q^b_{d,e} \tag{2.5}$$

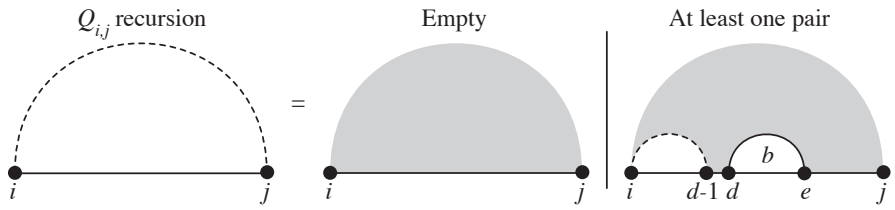


Figure 2.1. $O(N^4)$ Algorithm: Recursion for $Q_{i,j}$, the full partition function for the subsequence $[i,j]$. Either the subsequence $[i,j]$ is empty with recursion energy $G^{\text{empty}}_{i,j}=0$, or there exists one or more pairs with rightmost base pair $d{\cdot}e$ and recursion energy $G^{\text{empty}}_{e+1,j}=0$.

where $Q^b_{i,j}$ 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 2.1. Recursion diagrams [14] are a useful tool for describing the relationships between recursive quantitities. 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 (2.1)–(2.4). Unshaded regions under curves have partition function contributions based on recursive quantities previously evaluated for shorter subsequences.

In equation (2.5) and figure 2.1, the first possibility is that the subsequence $[i,j]$ is empty, contributing the term $\exp(-G^{\text{empty}}_{i,j}/RT)=1$. 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 $Q^b_{d,e}$. 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 $G^{\text{empty}}_{e+1,j}=0$. 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 $Q_{i,d-1}\,Q^b_{d,e}\exp(-G^{\text{empty}}_{e+1,j}/RT)$ 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

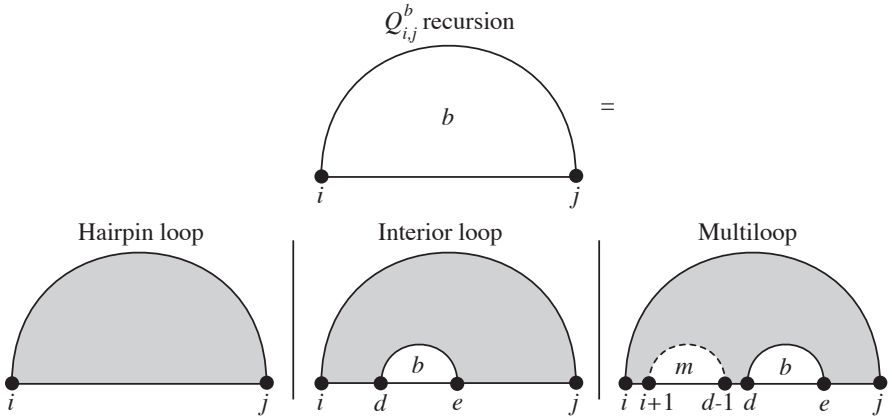


Figure 2.2. $O(N^4)$ Algorithm: Recursion for $Q^b_{i,j}$, the partition function for the subsequence $[i,j]$ assuming i and j are base-paired. Either the subsequence $[i,j]$ is a hairpin loop with recursion energy $G^{\text{hairpin}}_{i,j}$, or there exists one internal base pair $d{\cdot}e$ forming an interior loop with recursion energy $G^{\text{internal}}_{i,d,e,j}$, or there are at least two interior base pairs forming a multiloop with rightmost pair $d{\cdot}e$ and recursion energy $\alpha_1 + 2\alpha_2 + \alpha_3(j-e-1)$.

McCaskill's partition function approach and the redundant energy minimization recursions of Zuker and Stiegler [29], Rivas and Eddy [14], and Akutsu [1]. (The use of a *leftmost* extremal convention is equally valid.)

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

$$\begin{aligned} Q^b_{i,j} = {} & \exp(-G^{\text{hairpin}}_{i,j}/RT) \\ & + \sum_{\substack{d,\,e \\ i<d<e<j}} \exp(-G^{\text{interior}}_{i,d,e,j}/RT)\, Q^b_{d,e} \\ & + \sum_{\substack{d,\,e \\ i<d<e<j}} Q^m_{i+1,d-1} Q^b_{d,e} \exp\{-[\alpha_1+2\alpha_2+\alpha_3(j-e-1)]/RT\}, \end{aligned}$$

or equivalently by the recursion diagram of figure 2.2. Though similar in spirit, there are important differences from the recursions for $Q_{i,j}$ defined above. First, since i and j are paired, the empty recursion becomes a hairpin loop, with a recursion energy given by (2.2) and a corresponding partition function contribution $\exp(-G^{\text{hairpin}}_{i,j}/RT)$. 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 (2.3). The partition function contribution associated with the subinterval $[d, e]$ is given by a previous evaluation of $Q^b_{d,e}$ so that the total contribution for each interior loop structure is $\exp(-G^{\text{interior}}_{i,d,e,j}/RT)\, Q^b_{d,e}$. Otherwise, in addition to the rightmost pair de, 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 2.2 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 $Q^m_{i+1,d-1}$. This is possible because the energetic model for multiloops (2.4) 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 $Q^m_{i+1,d-1}\, Q^b_{d,e}\, \exp\{-[\alpha_1{+}2\alpha_2{+}\alpha_3(j{-}e{-}1)]/RT\}$.

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

$$Q^m_{i,j} = \sum_{\substack{d,\,e \\ i \le d < e \le j}} \Big[\exp\{-[\alpha_2 + \alpha_3(d-i) + \alpha_3(j-e)]/RT\}\, Q^b_{d,e} + Q^m_{i,d-1}\, Q^b_{d,e}\, \exp\{-[\alpha_2 + \alpha_3(j-e)]/RT\} \Big] \tag{2.6}$$

or the recursion diagram of figure 2.3. 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

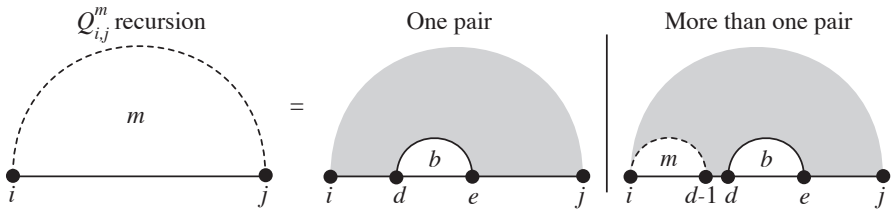


Figure 2.3. $O(N^4)$ Algorithm: Recursion for $Q^m_{i,j}$, the parition function for the subsequence $[i,j]$ inside a multiloop when there is at least one base pair in the subsequence. Either there is only one more base pair $d{\cdot}e$ defining the multiloop and the recursion energy is $\alpha_2 + \alpha_3(d-i) + \alpha_3(j-e)$, or there is more than one pair with rightmost pair $d{\cdot}e$ and recursion energy $\alpha_2 + \alpha_3(j-e)$.

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 2.4, where the recursion equations (2.5)–(2.6) 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.

2.3.3 $O(N^3)$ Algorithm

As noted by McCaskill in his original paper [12], 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, Zuker and Pedersen [10] exploit the form of the standard interior loop energy expression to calculate interior loop contributions in $O(N^3)$. Following McCaskill [12] and Lyngso [10], 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 2.5–2.8 and pseudocode is shown in figure 2.9. The

```
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^b_{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 2.4. 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 2.1–2.3.

recursion for $Q_{i,j}$ described in figure 2.5 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 since base d is known only to be paired to some base in the interval $[d+1, j]$.

The recursions for $Q^b_{i,j}$ and $Q^m_{i,j}$ are modified in a similar way in figures 2.6 and 2.7 by introducing the secondary recursion quantity Q^{ms} to compute the multiloop contributions. The $Q^s_{i,j}$ and $Q^{ms}_{i,j}$ recursions have exactly the same structure but different recursion energies and are depicted by the

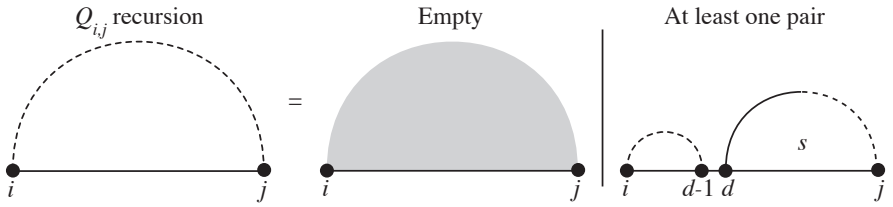


Figure 2.5. $O(N^3)$ Algorithm: Recursion for $Q_{i,j}$, the full partition function for the subsequence $[i, j]$. Either the subsequence $[i, j]$ is empty with recursion energy $G^{\text{empty}}_{i,j} = 0$, or there exists one or more pairs with a rightmost base pair that involves d and some other base on the subinterval $[d+4, j]$.

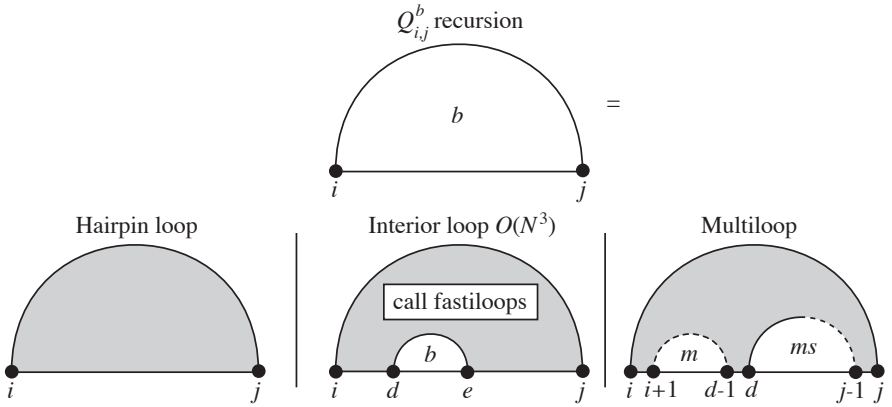


Figure 2.6. $O(N^3)$ Algorithm: Recursion for $Q^b_{i,j}$, the partition function for the subsequence $[i,j]$ assuming i and j are base-paired. Either the subsequence $[i,j]$ is a hairpin loop with recursion energy $G^{\text{hairpin}}_{i,j}$, or there exists one internal base pair $d\cdot e$ forming 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 appendix figures B.2 and B.3.

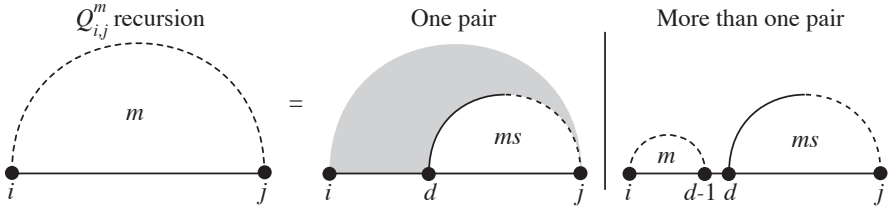


Figure 2.7. $O(N^3)$ Algorithm: Recursion for $Q^m_{i,j}$, the parition function for the subsequence $[i,j]$ inside a multiloop when there is at least one base pair in the subsequence. One possibility is that there is only one more base pair defining the multiloop. In this case, the unpaired subinterval $[i,d{-}1]$ 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 $[d{+}4,j]$.

recursion diagram in figure 2.8. 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 2.9 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 equation (2.3), whose four subscripts imply an $O(N^4)$ computational complexity for computing the

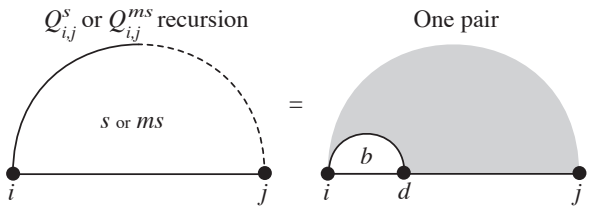


Figure 2.8. $O(N^3)$ Algorithm: Recursion for $Q^s_{i,j}$ and $Q^{ms}_{i,j}$, secondary parition functions for considering all possible rightmost base pairs that involve base i. For $Q^s_{i,j}$, the subsequence $[d, j]$ is external to all base pairs so the recursion energy is $G^{\text{empty}}_{d+1,j} = 0$. 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**
    $Q^b_{i,j} = \exp(-G^{\text{hairpin}}_{i,j}/RT)$
    //Compute internal loop contributions to Q^b in $O(N^3)$
    call fastiloops$(i, j, l, Q^b, Q^x, Q^{x2})$
    for $d = i+6, j-5$
      $Q^b_{i,j}$ += $Q^m_{i+1,d-1}\, Q^{ms}_{d,j-1}\, \exp\{-[\alpha_1 + \alpha_2]/RT\}$
    // Q^s, Q^{ms} **recursion**
    for $d = i+4, j$ // loop over all possible rightmost pairs $i \cdot d$
      $Q^s_{i,j}$ += $Q^b_{i,d}$
      $Q^{ms}_{i,j}$ += $Q^b_{i,d}\, \exp\{-[\alpha_2 + \alpha_3(j-d)]/RT\}$
    // Q, Q^m **recursions**
    $Q_{i,j} = 1$ //empty recursion
    for $d = i, j-4$
      $Q_{i,j}$+= $Q_{i,d-1}\, Q^s_{d,j}$
      $Q^m_{i,j}$ += $\exp\{-\alpha_3[d-i]/RT\}\, Q^{ms}_{d,j}$
      $Q^m_{i,j}$ += $Q^m_{i,d-1}\, Q^{ms}_{d,j}$
//Partition function is $Q_{1,N}$

Figure 2.9. 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 2.5–2.8. 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 appendix figures B.2 and B.3.

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

$$L_1 \equiv d - i - 1, \qquad L_2 \equiv j - e - 1 \tag{2.7}$$

so that the loop size and asymmetry may be expressed as

$$\text{size} \equiv L_1 + L_2, \qquad \text{asymmetry} \equiv |L_1 - L_2|.$$

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 $L_1 \leq 3$ or $L_2 \leq 3$. However, for all cases when both $L_1 \geq 4$ and $L_2 \geq 4$, the form of the energy function becomes

$$\begin{aligned} G^{\text{interior}}_{i,d,e,j} = {} & \gamma_1(L_1{+}L_2) + \gamma_2(|L_1{-}L_2|) \\ & + \gamma_3(i,j,i{+}1,j{-}1) + \gamma_3(e,d,e{+}1,d{-}1) \end{aligned} \tag{2.8}$$

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

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

where d and e are expressed in terms of L_1 and L_2 using (2.7). 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

$$Q^x_{i,s} \exp\{-\gamma_3(i,j,i{+}1,j{-}1)/RT\} \tag{2.9}$$

since 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 [10]

$$\begin{aligned} Q^x_{i-1,s+2} = \quad & Q^x_{i,s} \exp\{-[\gamma_1(s{+}2) - \gamma_1(s)]/RT\} \\ & + \exp\Big\{-\Big[\gamma_1(s+2) + \gamma_2(|L_1{-}L_2|) \\ & + \gamma_3(e,d,e{+}1,d{-}1)\Big]/RT\Big\} Q^b_{d,e}\Big|_{\substack{L_1=4 \\ L_2=s-2}} \\ & + \exp\Big\{-\Big[\gamma_1(s+2) + \gamma_2(|L_1{-}L_2|) \\ & + \gamma_3(e,d,e{+}1,d{-}1)\Big]/RT\Big\} Q^b_{d,e}\Big|_{\substack{L_1=s-2 \\ L_2=4}} \ . \end{aligned} \tag{2.10}$$

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 since these smaller loops do not use the energy expression (2.8). Exploiting the extension property (2.10) and making use of (2.9), 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 $O(N)$ non-extensible 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 appendix figures B.2 and B.3.

2.3.4 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.

2.4 Partition Function with Pseudoknots

2.4.1 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

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

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_1^m, and if the pseudoknot is inside another pseudoknot, β_1 is replaced by β_1^p. Several features of this potential function are illustrated in appendix figure B.4.

2.4.2 $O(N^8)$ 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 2.10–2.12, p-curves are introduced in a completely analagous manner to b-curves. Each p-curve represents the 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 for the paritition 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 2.13, where the pseudoknot interior is specified. Arcs that would cross in this diagram are reflected across the horizontal axis 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 figure 2.10) 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 2.14. There are two types

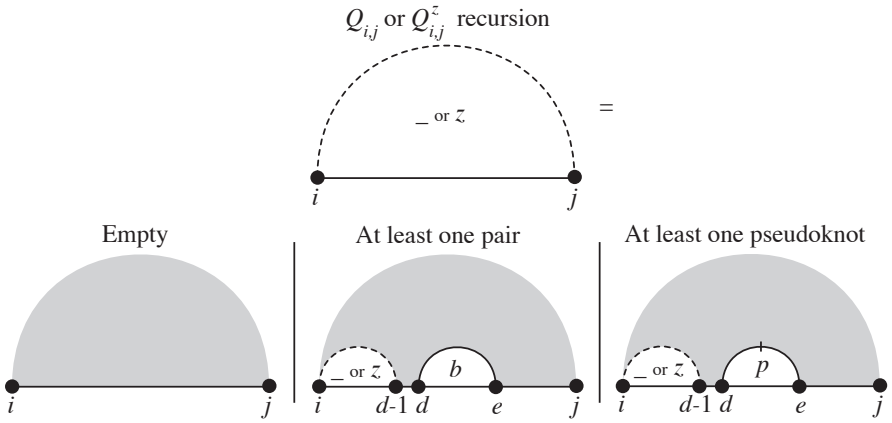


Figure 2.10. $O(N^8)$ Algorithm: Recursion for $Q_{i,j}$, the full partition function for the subsequence $[i,j]$. Either the subsequence $[i,j]$ is empty with recursion energy $G^{\text{empty}}_{i,j}$, or there is at least one pair (with rightmost base pair $d{\cdot}e$) with recursion energy $G^{\text{empty}}_{e+1,j}$, or there is at least one pseudoknot (with rightmost pseudoknot filling 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 subsequence $[d,e]$) with recursion energy $\beta^p_1+2\beta_2+\beta_3(j-e)$.

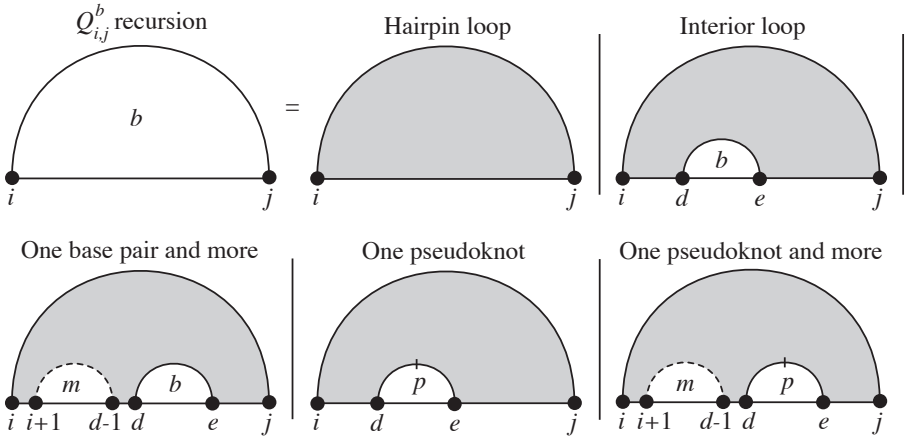


Figure 2.11. $O(N^8)$ Algorithm: Recursion for $Q^b_{i,j}$, the partition function for the subsequence $[i,j]$ assuming i and j are base-paired. Either the subsequence $[i,j]$ is a hairpin loop with recursion energy $G^{\text{hairpin}}_{i,j}$, or there exists one internal base pair $d{\cdot}e$ forming 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 $\alpha_1+\beta^m_1+3\alpha_2+\alpha_3(j-e-1)$.

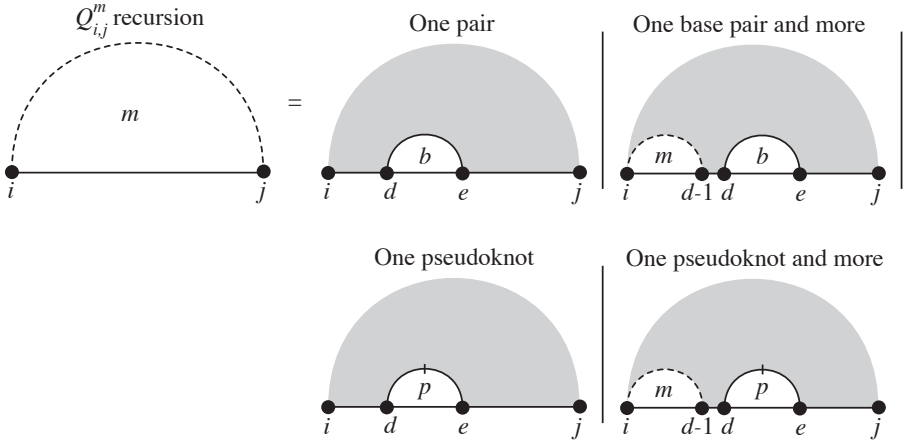


Figure 2.12. $O(N^8)$ Algorithm: Recursion for $Q^m_{i,j}$, the parition function for the subsequence $[i, j]$ inside a multiloop when there is at least one base pair or pseudoknot in the subsequence. Either there is one final base pair $d \cdot e$ in the multiloop with recursion energy $\alpha_2 + \alpha_3(d-i) + \alpha_3(j-e)$, 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 subsequence $[d, e]$) and recursion energy $\beta^m_1 + 2\alpha_2 + \alpha_3(j-e)$.

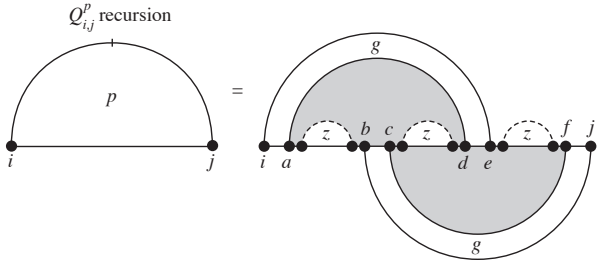


Figure 2.13. $O(N^8)$ Algorithm: Recursion for $Q^p_{i,j}$, the partition function for the pseudoknot filling the subsequence $[i, j]$. The recursion energy is $2\beta_2$, where β_2 is the penalty associated with each base pair bordering the interior of the pseudoknot.

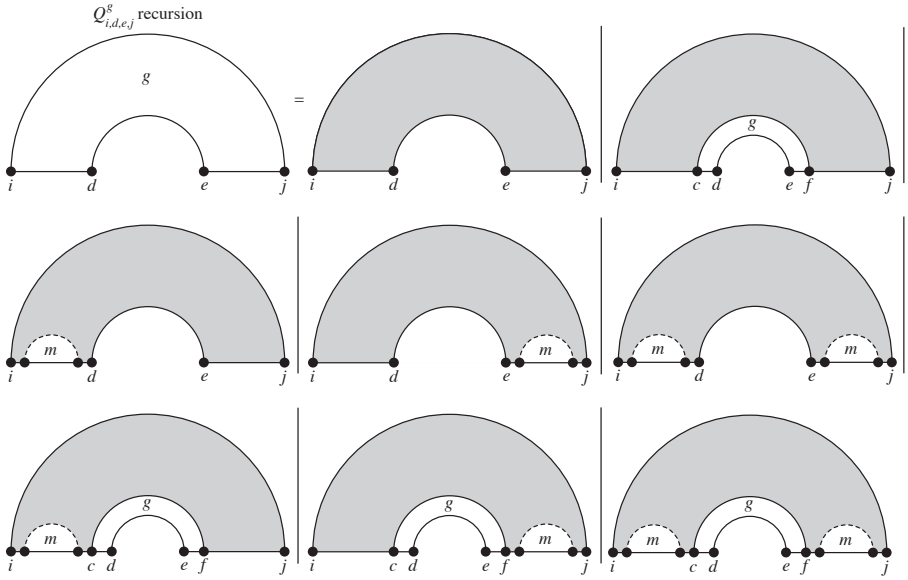


Figure 2.14. $O(N^8)$ Algorithm: Recursion for $Q^g_{i,d,e,j}$, the partition function for the pseudoknot spanning region filling subsequence $[i,j]$ excluding the gap $[d{+}1, e{-}1]$. There are two types of cases: i. either $i{\cdot}j$ and $d{\cdot}e$ are the only spanning base pairs, ii. there is 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. $\alpha_1 + 2\alpha_2$).

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 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 appendix figure B.5. 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 2.13 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

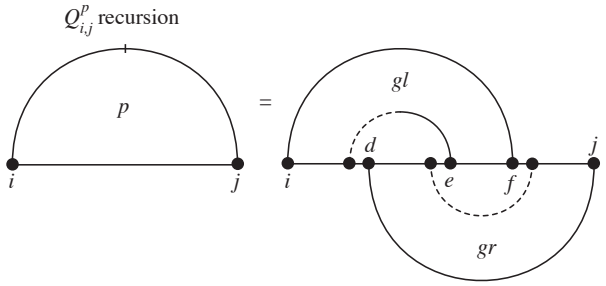


Figure 2.15. $O(N^5)$ Algorithm: Recursion for $Q^p_{i,j}$, the partition function for the pseudoknot filling the subsequence $[i, j]$. There is no recursion energy associated with this diagram.

database of known RNA pseudoknots [23, 22]. However, this class is somewhat more restrictive than those treated by the redundant structure prediction recursions of Rivas and Eddy [14] and Akutsu [1] (see appendix figure B.6).

2.4.3 $O(N^5)$ 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 2.10–2.14 or the nesting depth of the loops in the $O(N^8)$ pseudocode of appendix figure B.5, 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 2.10–2.12 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 show in figure 2.15. In comparison to figure 2.13, the interior z-curve regions of the pseudknot 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

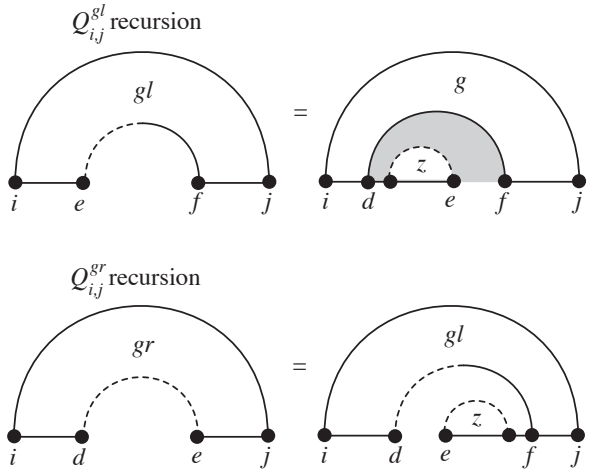


Figure 2.16. $O(N^5)$ Algorithm: Recursions for $Q^{gl}_{i,d,e,j}$ and $Q^{gr}_{i,d,e,j}$, the partition functions for the left and right spanning regions of a pseudoknot. The recursion energy for Q^{gl} is β_2 corresponding to the penalty for base pair $d{\cdot}f$ bordering the interior of a pseudoknot. There is no recursion energy for Q^{gr}.

are now five subscripts corresponding to an $O(N^5)$ complexity.

The recursions for Q^{gl} and Q^{gr} are defined in figure 2.16, 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 2.17. 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 pseudocode of appendix figure B.7. 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 2.18. 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

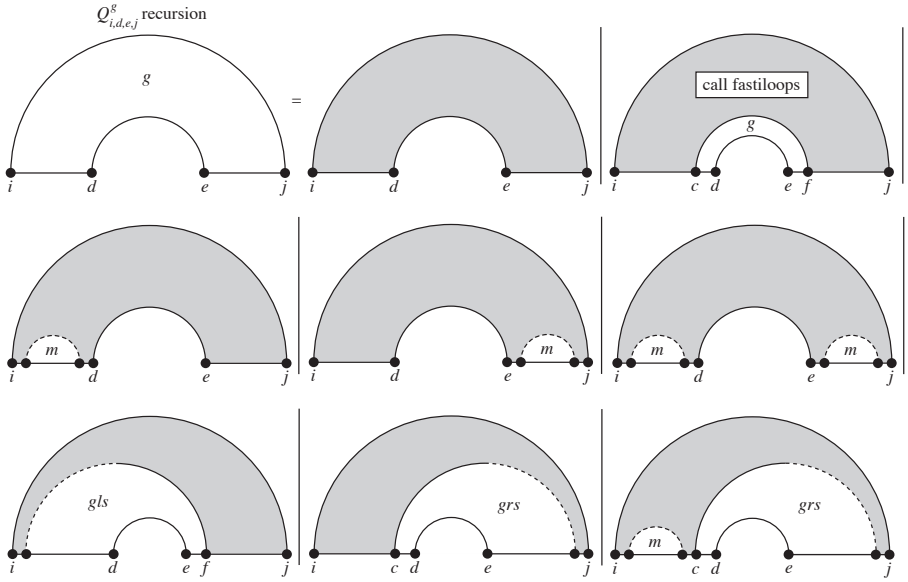


Figure 2.17. $O(N^5)$ Algorithm: Recursion for $Q^{g}_{i,d,e,j}$, the partition function for the pseudoknot spanning region filling subsequence $[i,j]$ excluding the gap $[d{+}1, e{-}1]$. There are two types of cases: i. either $i \cdot j$ and $d \cdot e$ are the only spanning base pairs, ii. there is 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 2.14. 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 appendix figure B.7. 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}.

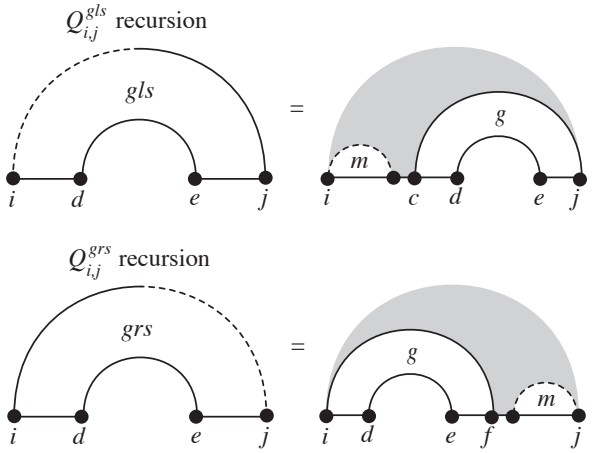


Figure 2.18. $O(N^5)$ Algorithm: Recursions for $Q^{gls}_{i,d,e,j}$ and $Q^{grs}_{i,d,e,j}$, supplementary gap partition functions used in computing Q^{g}. The recursion energy for both diagrams is α_2, corresponding to the introduction of a base pair bordering the interior of a multiloop.

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 2.19. This is the main result of the chapter.

2.5 Methods

The partition function algorithms and minimum energy structure prediction algorithms described in this chapter were implemented in the C programming language using recursion energies based on standard secondary structure energy models for unpseudoknotted ssDNA and RNA [15, 11]. 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 [11] is the exclusion of the special bonus for hairpins with GGG on the 5' side of the stem since this term violates the loop decomposition paradigm. Coaxial stacking terms [11] 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 [11], including dangle energies [16] 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

```
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 + β^m_1 + 3α2 + α3 (d-i-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 e = 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{-[β^m_1 + 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{-[β^m_1 + 2α2 + α3 (j-e)]/RT}
        Q^z_{i,j} += Q^z_{i,d-1} Q^p_{d,e} exp{-[β^p_1 + 2β2 + β3(j-e)]/RT}
//Partition function is Q_{1,N}
```

Figure 2.19. 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 schemataically in figures 2.10–2.12 and 2.15–2.18. The function "fastiloops" computes certain interior loop contributions to Q^g in $O(N^5)$ as detailed in the pseudocode of appendix figure B.7.

Table 2.1. 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 [14]	195/200	197/200	98%	40/100	46/100	43%

energy structure.

2.6 Results

2.6.1 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 co-workers [11]. 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 [17]. 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 [11], 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 was provided in RNA crystal and NMR structure databases to provide more information for secondary structure studies.

For the positive control, we draw from a database of 212 RNA pseudoknots, each described as a single stretch of consecutive nucleotides [23, 22]. From this sample, we exclude the five structures that contain chains of pseudoknots that cannot be modeled using the present recursions (see appendix figure B.6). From the remaining sequences, we randomly select a working set and a free set of 100 pseudoknots each. In order 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 [6] 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 2.1. 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 [14] 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 accomodated to the extent that the dynamic programming framework allows.

2.6.2 Algorithm Complexity

The computational complexity of all four partition function algorithms is demonstrated empirically in figure 2.20. The $O(N^4)$ and $O(N^3)$ 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 seconds using the $O(N^3)$ algorithm and 2900 seconds 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 seconds while the $O(N^8)$ algorithm requires roughly 1200 seconds. By comparison, the $O(N^6)$ structure prediction algorithm of Rivas and Eddy [14] runs in approximately 2300 seconds on sequences of length 100. Complexity benchmarks were performed on a 700MHz Pentium Xeon processor.

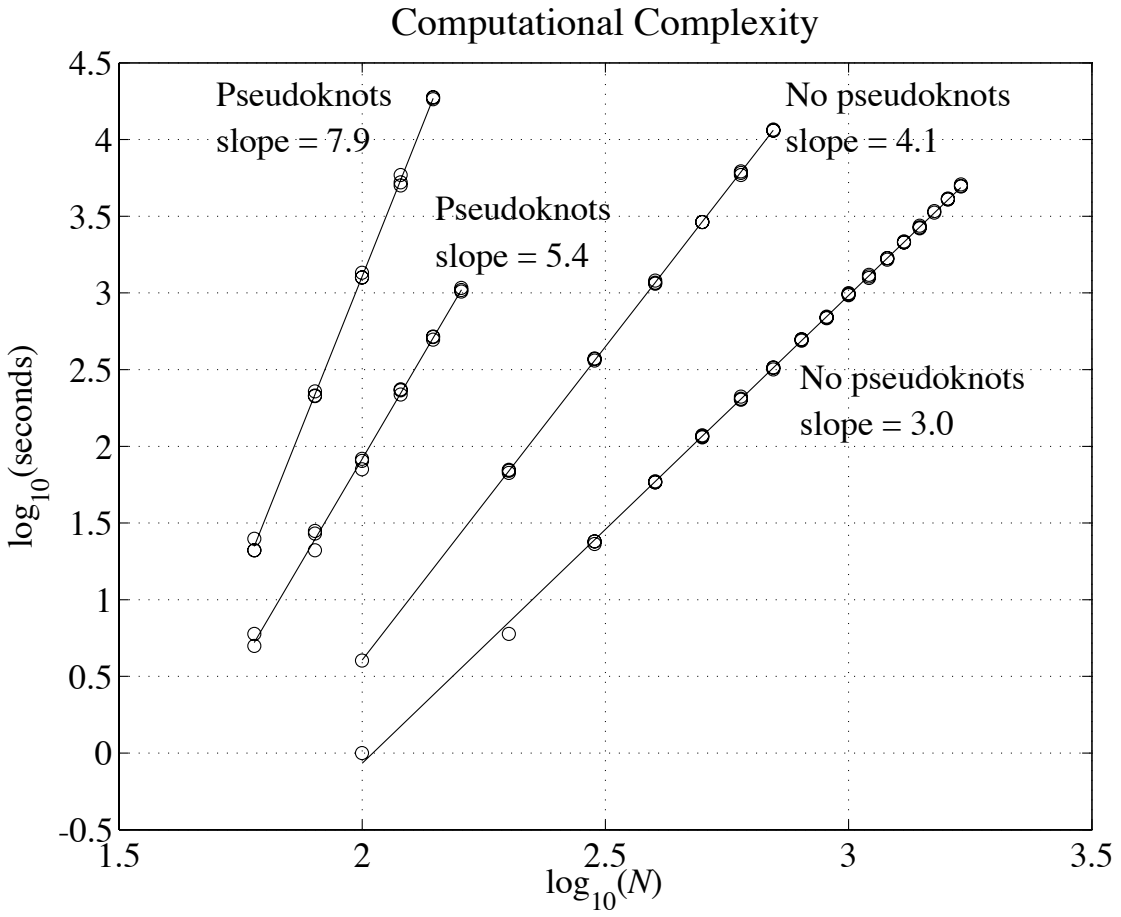


Figure 2.20. Comparison of the observed computational complexity for $O(N^4)$ and $O(N^3)$ partition function algorithms without pseudoknots and $O(N^8)$ and $O(N^5)$ partition function algorithms that include pseudoknots. Each algorithm is run on three randomly selected sequences for each of the depicted sequence lengths N. Timings are performed on a 700MHz Pentium Xeon processor.

2.7 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.

Bibliography

[1] T. Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104:45–62, 2000.

[2] S. Bonhoeffer, J. S. McCaskill, P. F. Stadler, and P. Schuster. RNA multi-structure landscapes. *European Biophysics Journal*, 22:13–24, 1993.

[3] D. Bouthinon and H. Soldano. A new method to predict the consensus secondary structure of a set of unaligned RNA sequences. *Bioinformatics*, 15(10):785–98, 1999.

[4] J. H. Chen, S. Y. Le, and J. V. Maizel. A procedure for RNA pseudoknot prediction. *Computer Applications in the Biosciences*, 8(3):243–8, 1992.

[5] C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7:254–265, 2001.

[6] A. P. Gultyaev, F. H. D. van Batenburg, and C. W. A. Pleij. An approximation of loop free energy values of RNA h-pseudoknots. *RNA*, 5:609–617, 1999.

[7] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Chemical Monthly*, 125:167–188, 1994.

[8] H. Isambert and E. D. Siggia. Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme. *Proceedings of the National Academy of Sciences of the United States of America*, 97(12):6515–20, 2000.

[9] R. B. Lyngso and C. N. S. Pedersen. RNA pseudoknot prediction in energy-based models. *Journal of Computational Biology*, 7(3/4):409–427, 2000.

[10] R. B. Lyngso, M. Zuker, and C. N. S. Pedersen. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, 15(6):440–445, 1999.

[11] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940, 1999.

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

[13] R. Nussinov, J. R. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algorithms for loop matchings. *SIAM Journal of Applied Mathematics*, 35:68–82, 1978.

[14] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285:2053–2068, 1999.

[15] J. Santalucia Jr. Improved nearest-neighbor parameters for predicting DNA duplex stability. *Biochemistry*, 35:3555–3562, 1996.

[16] M. J. Serra and D. H. Turner. Prediction thermodynamic properties of RNA. *Methods Enzymology*, 259:242–261, 1995.

[17] M. Sprinzl, C. Horn, M. Brown, A. Ioudovitch, and S. Steinberg. Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Research*, 26:148–153, 1998.

[18] J. E. Tabaska, R. B. Cary, H. N. Gabow, and G. D. Stormo. An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics*, 14(8):691–9, 1998.

[19] I. Tinoco Jr., O. C. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[20] D. H. Turner, N. Sugimoto, and S. M. Freier. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17:167–192, 1988.

[21] Y. Uemura, A. Hasegawa, S. Kobayashi, and T. Yokomori. Tree adjoining grammars for RNA structure prediction. *Theoretical Computer Science*, 210(2):277–303, 1999.

[22] F. H. D. van Batenburg, A. P. Gultyaev, and C. W. A. Pleij. Pseudobase: structural information on RNA pseudoknots. *Nucleic Acids Research*, 29(1):194–195, 2001.

[23] F. H. D. van Batenburg, A. P. Gultyaev, C. W. A. Pleij, and J. Ng. Pseudobase: a database with RNA pseudoknots. *Nucleic Acids Research*, 28:201–204, 2000.

[24] M. S. Waterman. Secondary structure of single-stranded nucleic acids. In *Studies in foundations and combinatorics: Advances in Mathematics Supplemental Studies*, volume 1, pages 167–212. Academic Press, New York, 1978.

[25] M. S. Waterman and T. F. Smith. RNA secondary structure: a complete mathematical analysis. *Mathematical Biosciences*, 42:257–266, 1978.

[26] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

[27] M. Zuker. Calculating nucleic acid secondary structure. *Current Opinion in Structural Biology*, 10:303–310, 2000.

[28] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, 1984.

[29] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–147, 1981.

Chapter 3

Analysis: Recursion Probabilities

The work presented here is heavily based on:

R.M. Dirks and N.A. Pierce, *An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots.* Journal of Computational Chemistry, 2004. **25**(10): p. 1295-1304. Reprinted with copyright permissions.

3.1 Abstract

Given a nucleic acid sequence, the algorithm in the preceding chapter allows the calculation of the partition function over secondary structure space including a class of physically relevant pseudoknots. Here, a method for computing base-pairing probabilities starting from the output of this partition function algorithm is presented. The approach relies on the calculation of recursion probabilities that are computed by backtracking through the partition function algorithm, applying a particular transformation at each step. This transformation is applicable to any partition function algorithm that follows the same basic dynamic programming paradigm. Base-pairing probabilities are useful for analyzing the equilibrium ensemble properties of natural and engineered nucleic acids, as demonstrated for a human telomerase RNA and a synthetic DNA nanostructure.

3.2 Introduction

Thermodynamic models based on nucleic acid secondary structure and nearest-neighbor identities [17, 18, 14, 10, 24] underly dynamic programming algorithms for predicting the minimum energy secondary structure [20, 21, 12, 25, 8] and calculating the partition function over secondary structure space [11, 2, 8]. In their original forms, these algorithms exclude the possiblity of pseudoknots, a biologically relevant class of secondary structures [19] that also arises in DNA nanotechnology applications [22, 23]. Pseudoknots result when two base pairs $i{\cdot}j$ and $d{\cdot}e$, with $i < d$, fail to satisfy the nesting property $i < d < e < j$ (see figure 3.1). Recent extensions of the structure prediction [13, 1, 6] and partition function [6] algorithms allow the inclusion of certain pseudoknots.

For an ensemble of secondary structures $s \in \Omega$, the partition function

$$Q = \sum_{s \in \Omega} e^{-G_s/RT}$$

may be used to compute the probability

$$p(s^*) = \tfrac{1}{Q} e^{-G_{s^*}/RT} \tag{3.1}$$

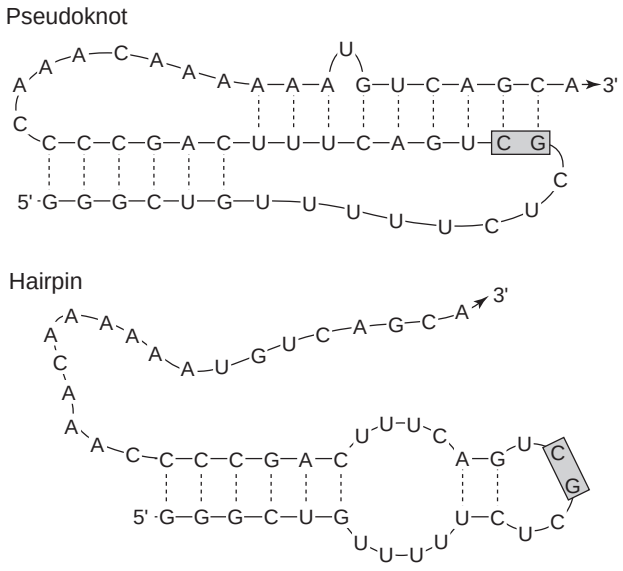


Figure 3.1. Secondary structures of competing pseudoknot and hairpin constructs in human telomerase RNA. The wild-type sequence is shown. For the two-point mutant implicated in dyskeratosis congenita, GC is replaced by AG in the shaded boxes, disrupting two base pairs in the pseudoknot construct. For the experimental studies of the hairpin structure [16], the 18 nucleotides at the 3' end are excluded to prevent formation of the pseudoknot.

that secondary structure s^* is sampled at thermodynamic equilibrium. The ensemble equilibrium can also be characterized by the matrix of base-pairing probabilities with entries $p_{i,j}$ corresponding to the probability that base i is paired with base j in Ω.

McCaskill's original paper [11] defines elegant dynamic programs to compute the partition function and base-pairing probabilities over the ensemble of unpseudoknotted secondary structures. The partition function algorithm builds up recursively from short subsequences to the full strand and then the pair probabilities are computed by working backwards to short subsequences using intermediate results from the partition function calculation. In the absence of pseudoknots, the partition function algorithm is sufficiently succinct that McCaskill is able to determine the form of the pair probability backtrack algorithm simply by considering the few possible forms of enclosing secondary structure for any given base pair. While this approach is simple and efficient, it is not easily generalizable to algorithmic extensions, such as the inclusion of pseudoknots. Here, we describe a general method for mechanically transforming the new pseudoknot partition function algorithm [6] to compute recursion probabilities, which can be used in turn to compute base-pairing probabilities. The

transformation approach is generalizable to any future partition function extensions that follow the same dynamic programming paradigm.

Base-pairing probabilities assist in the analysis of biologically relevant pseudoknots. Here, we examine human telomerase RNA, which exists at equilibrium in both hairpin and pseudoknotted forms [3]. A two-point mutation, implicated in the disease dyskeratosis congenita, alters the thermodynamic balance between these competing structures [16]. This shift in equilibrium is clearly identifiable when the base-pairing probabilities for the two sequences are compared. Base-pairing probabilities that permit pseudoknots are also useful in analyzing synthetic DNA nanostructures [22, 23].

3.3 Algorithm

For clarity, we begin by considering the class of secondary structures excluding pseudoknots and then address the additional complexity that arises when pseudoknots are introduced.

3.3.1 Partition Function Recursions

For a strand of length N, the partition function may be computed over all unpseudoknotted secondary structures in $O(N^4)$ using the algorithm [11, 8] summarized in figure 2.4 (see chapter 2 for a detailed description)[1]. Partition function recursions are non-redundant in the sense that every secondary structure in the ensemble Ω is visited exactly once using a unique sequence of recursions. The algorithm computes the partition function $Q_{i,j}$ for each subsequence $[i,j]$ ignoring all bases exterior to $[i,j]$, starting from subsequences of length $l=1$ and building up incrementally to $l=N$. The recursions that define $Q_{i,j}$ rely on additional restricted partition functions $Q^b_{i,j}$ and $Q^m_{i,j}$. $Q^b_{i,j}$ represents the partition function for subsequence $[i,j]$ given that i and j are base paired and $Q^m_{i,j}$ is used to calculate multiloop contributions. At the end of the recursive process, the full partition function Q is given by $Q_{1,N}$ and the values of $Q_{i,j}, Q^b_{i,j}, Q^m_{i,j}$ are stored in matrices for $1 \leq i,j \leq N$.

[1] The complexity may be reduced to $O(N^3)$ by exploiting the formulation of the nearest-neighbor energy model for long interior loops [9, 6]

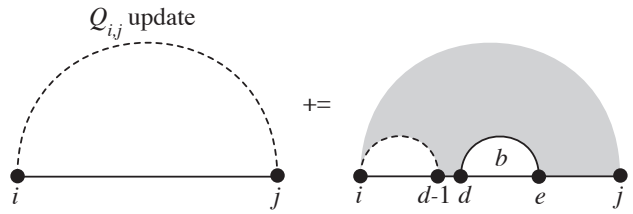


Figure 3.2. Recursion diagram corresponding to recursive update (3.2), depicting the addition to $Q_{i,j}$ of partition function contributions for those structures with rightmost base pair $d{\cdot}e$. See chapter 2 for a thorough description of the partition function algorithm (with or without pseudoknots) in terms of recursion diagrams.

These intermediate results will play a critical role in the new algorithm described below.

3.3.2 Recursion Probabilities

Following the execution of the partition function calculation, a second algorithm can be implemented to calculate probability matrices, P, P^b, P^m, corresponding to the Q, Q^b, Q^m matrices. The values stored in these P-type matrices will be termed *recursion probabilities*.

Recursion probabilities can be intuitively described as follows. Consider sampling the ensemble of secondary structures $s \in \Omega$ where the probability of selecting structure s^* is given by the Boltzmann probability (3.1). For each secondary structure s^*, the contribution to Q is computed by a unique recursion sequence involving specific $Q_{i,j}$, $Q^b_{i,j}$ and $Q^m_{i,j}$ intermediates. Associating these intermediates with structure s^*, the recursion probability $P_{i,j}$, $P^b_{i,j}$ or $P^m_{i,j}$ corresponds to the probability that the sampled structure s^* requires the use of the corresponding intermediate $Q_{i,j}$, $Q^b_{i,j}$ or $Q^m_{i,j}$ to calculate the partition function contribution.

Recent work by Ding and Lawrence [4] exploits quantities related to recursion probabilities to statistically sample the distribution of unpseudoknotted secondary structures for a given sequence. Here, we develop a general approach for computing P-type matrices given a set of Q-type matrices and corresponding partition function recursions.

An algorithm for computing recursion probabilities can be formulated in a mechanical way starting from a set of partition function recursions. The crux of this formulation is the repeated application of a single transformation to the partition function code. In particular, updates of the

form

$$Q_{i,j} \mathrel{+}= Q_{i,d-1}\, Q^b_{d,e} \tag{3.2}$$

(equivalent to the recursion diagram of figure 3.2) are converted to the following series of statements.

$$\begin{aligned} \Delta p \quad &= \quad P_{i,j}\, \overbrace{Q_{i,d-1}\, Q^b_{d,e}/Q_{i,j}}^{\text{conditional probability}} \\ P_{i,d-1} \quad &\mathrel{+}= \quad \Delta p \\ P^b_{d,e} \quad &\mathrel{+}= \quad \Delta p \end{aligned}$$

Specifically, the right hand side (RHS) of each recursive update is divided by the left hand side (LHS), and the P term corresponding to the new denominator is multiplied by this quotient. The resulting probabilities, temporarily stored as Δp, are subsequently added to every P-type value corresponding to the Q-type terms on the RHS of the original statement (3.2).

To understand this transformation, recall that $Q_{i,j}$, $Q^b_{i,j}$ and $Q^m_{i,j}$ are partition functions for structural subclasses of the full sequence. In recursive updates such as (3.2), the ratio of the RHS to the fully-computed LHS corresponds to the probability that a structure drawn from an equilibrium ensemble defined by the LHS partition function is in the sub-ensemble defined by the RHS partition function. As an example, transformation (3.3) states that for any i, d, e, j, the structures represented by $Q_{i,j}$ partially consist of substructures represented by $Q_{i,d-1}$ and $Q^b_{d,e}$. Consequently, once the probability $P_{i,j}$ is determined, it can be used to augment $P_{i,d-1}$ and $P^b_{d,e}$ since the frequencies of the corresponding substructures within the $Q_{i,j}$ ensemble can be derived from $Q_{i,d-1}$ and $Q^b_{d,e}$. By backtracking through the partition function algorithm and transforming all recursive updates analagously to (3.3), probabilities can be calculated for each recursion.

Starting from the partition function algorithm of figure 2.4, the recursion probability algorithm is obtained by performing three modifications: (1) the two outermost loops are altered so that the algorithm starts with the full strand of length $l = N$ and decrements down to subsequences of length $l = 1$, (2) all recursive updates are transformed as for equation (3.3), (3) the order of the recursion blocks $(Q^b, [Q, Q^m])$ is reversed $([P, P^m], P^b)$. This last modification is necessary

Compute Q, Q^b, Q^m using $O(N^4)$ partition function algorithm
Initialize (P, P^b, P^m) // $O(N^2)$ space
Set all P-type values to 0
$P_{1,N} = 1$ //probability of "recursing" to the entire strand is 1
for $l = N, 1$ //decrement
  for $i = 1, N-l+1$
    $j = i+l-1$
    //P, P^m **recursions**
    for $d = i, j-4$ // loop over all possible rightmost pairs $d{\cdot}e$
      for $e = d+4, j$
        $\Delta p = P_{i,j} Q_{i,d-1}\, Q^b_{d,e} / Q_{i,j}$
        $P_{i,d-1} \mathrel{+}= \Delta p$
        $P^b_{d,e} \mathrel{+}= \Delta p$
        $P^b_{d,e} \mathrel{+}= P^m_{i,j} \exp\{-[\alpha_2 + \alpha_3(d-i) + \alpha_3(j-e)]/RT\}\, Q^b_{d,e} / Q^m_{i,j}$
        $\Delta p = P^m_{i,j} Q^m_{i,d-1}\, Q^b_{d,e} \exp\{-[\alpha_2 + \alpha_3(j-e)]/RT\} / Q^m_{i,j}$
        $P^m_{i,d-1} \mathrel{+}= \Delta p$
        $P^b_{d,e} \mathrel{+}= \Delta p$
    // P^b **recursion**
    for $d = i+1, j-5$ // loop over all possible rightmost pairs $d{\cdot}e$
      for $e = d+4, j-1$
        $P^b_{d,e} \mathrel{+}= P^b_{i,j}\, Q^b_{d,e} \exp\{-G^{\text{interior}}_{i,d,e,j}/RT\} / Q^b_{i,j}$
        $\Delta p = P^b_{i,j} Q^m_{i+1,d-1}\, Q^b_{d,e} \exp\{-[\alpha_1 + 2\alpha_2 + \alpha_3(j-e-1)]/RT\} / Q^b_{i,j}$
        $P^m_{i+1,d-1} \mathrel{+}= \Delta p$
        $P^b_{d,e} \mathrel{+}= \Delta p$

Figure 3.3. $O(N^4)$ recursion probability algorithm that excludes pseudoknots. For simplicity, we omit details such as checking for updates with zero in the denominator (in which case the numerator will also evaluate to zero and the expression should be skipped).

because the recursion order in the partition function algorithm ensures that if one quantity (e.g. $Q_{i,j}$) recurses to another quantity of the same length (e.g. $Q^b_{i,j}$) then the "lower level" quantity (i.e. $Q^b_{i,j}$) is calculated first. The reverse ordering is needed for the recursion probability algorithm since $P^b_{i,j}$ cannot be used until it has been fully computed in the $P_{i,j}$ loop.

The pseudocode in figure 3.3 details the outcome of these transformations for the unpseudoknotted case. This modified algorithm reverses the flow of the partition function calculation and incrementally determines all recursion probabilities (frequencies of families of structures), based on the probabilities of all superstructures that directly contain them. Once recursion probabilities are computed for all i and j, the base-pairing probability $p_{i,j}$ is simply $P^b_{i,j}$, since $Q^b_{i,j}$ is associated with every structure s in which $i{\cdot}j$ appears, and $i{\cdot}j$ is associated with exactly one $Q^b_{i,j}$. By starting from a more complicated $O(N^3)$ partition function algorithm [9, 6], the computational complexity of the recursion probability algorithm can also be reduced to $O(N^3)$ as described in Appendix C.

3.3.3 Pseudoknots

The procedure outlined above for obtaining recursion probability algorithms is equally applicable to a new partition function algorithm that includes pseudoknots (see the pseudocode of figure 2.19). For the unpseudoknotted algorithm, all base pairs stem from Q^b recursions, so the values stored in P^b are precisely the desired probabilities (i.e. $p_{i,j} = P^b_{i,j}$). For the pseudoknotted case, $P^b_{i,j}$ only gives the probability that i and j form a nested pair. The full base-pairing probability must also take into consideration those base pairs that are non-nested and lead to pseudoknotted structures (termed gap-spanning pairs in [6]). For these gap-spanning pairs, there is no single recursion probability that represents the contribution to $p_{i,j}$. However, this contribution may be succinctly represented in terms of Q-type and P-type matrices for the full pseudoknotted algorithm.

A new set of quantities, $P^{bg}_{i,j}$, will be used to store the base pairing probabilities of $i \cdot j$ gap-spanning pairs in pseudoknots. The most pertinent recursion probability, $P^g_{i,d,e,j}$, stores the probability of a gap structure with outer gap-spanning pair $i \cdot j$ and inner gap-spanning pair $d \cdot e$ corresponding to the partition function recursion $Q^g_{i,d,e,j}$ (see figure 19 in [6]). Due to the structure of the $Q^g_{i,d,e,j}$ recursion, the sum of $P^g_{i,d,e,j}$ over all values of d, e precisely gives the probability of an outer pair $i \cdot j$

$$P^{bg}_{i,j} \mathrel{+}= \sum_{i<d<e<j} P^g_{i,d,e,j}. \tag{3.3}$$

However, the sum of $P^g_{i,d,e,j}$ over all values of i, j does *not* give the probability of an inner pair $d \cdot e$, since the same inner pair may be present in multiple recursions required to define the same secondary structure. To correctly determine the probabilities of inner gap-spanning pairs, only the portion of P^g that corresponds to calling Q^g directly from Q^{gl} should be included

$$P^{bg}_{d,f} \mathrel{+}= \sum_{i<d\leq e<f<j} P^{gl}_{i,e,f,j} Q^g_{i,d,f,j} Q^z_{d+1,e} \exp(-\beta_2/RT)/Q^{gl}_{i,e,f,j}. \tag{3.4}$$

Here, Q^{gl} and Q^z are partition function recursions used to define the interior structure of a pseudoknot and β_2 is a pseudoknot energy parameter (see figures 18 and 12 in [6]). Allowing pseudoknots,

the total probability of a base pair $i \cdot j$ is then

$$p_{i,j} = P^{b}_{i,j} + P^{bg}_{i,j}.$$

Pseudocode detailing the algorithm for computing recursion probabilities in the pseudoknotted case is provided in figure 3.4, where the calculation of $P^{bg}_{i,j}$ using (3.3) and (3.4) has been embedded at little additional cost.[2] In Appendix C, we describe how to reduce the complexity of the pseudoknotted algorithm from $O(N^6)$ to $O(N^5)$.

3.4 Methods

The standard energy model [10] and pseudoknot extension [6] are implemented as described previously [6], including dangle energies and penalties for helices not terminated by a G·C pair. These terms do not change the structure of the recursions described in the pseudocode and are omitted for clarity. Coaxial stacking contributions are not included in the physical model, as it is unclear how to treat different stackings associated with the same secondary structure in the context of the partition function. To maintain consistency with a recent design study [5], dangle energies are treated analogously to the d2 option in the Vienna package [8]. Following this approach, dangle energies are included even if two helices are separated by one or zero bases, providing some compensation for the neglect of coaxial stacking bonuses.

3.5 Applications

The recursion probability algorithm provides a simple, general method for calculating the frequency of various substructures in the ensemble of states for a given nucleic acid. Base-pairing probabilities derived from the recursion probabilities are particularly useful for analyzing secondary structure via dot plot analyses [11]. A traditional dot plot depicts the probabilities of forming all possible $i \cdot j$ base pairs. In the case of pseudoknots, the dot plot can be decomposed into two dot plots, one for nested

[2] Note that (3.3) and (3.4) use different indices for P^{bg} to maintain consistency with the pseudocode.

Compute $Q, Q^b, Q^m, Q^p, Q^z, Q^g, Q^{gl}, Q^{gr}, Q^{gls}, Q^{grs}$
// using $O(N^5)$ partition function algorithm
Initialize $(P, P^b, P^m, P^p, P^z, P^{bg})$ // $O(N^2)$ space
Initialize $(P^g, P^{gl}, P^{gr}, P^{gls}, P^{grs})$ // $O(N^4)$ space
// Initialize $(Q^x, Q^{x1}, Q^{x2}, P^x, P^{x1}, P^{x2})$ for $O(N^5)$ version
// Set all Q^x-type values to 0 for $O(N^5)$ version
Set all P-type values to 0
$P_{1,N} = 1$ //probability of "recursing" to the entire strand is 1
for $l = N, 1$ //decrement
  // Initialize $Q^x = Q^{x1}$, $Q^{x1} = Q^{x2}$, $Q^{x2} = 0$ for $O(N^5)$ version
  // Initialize $P^x = P^{x1}$, $P^{x1} = P^{x2}$, $P^{x2} = 0$ for $O(N^5)$ version
  for $i = 1, N-l+1$
    $j = i+l-1$
    //P, P^m, P^z **recursions**
    for $d = i, j-4$ // all possible rightmost pairs $d{\cdot}e$
      for $e = d+4, j$
        $\Delta p = P_{i,j} Q_{i,d-1} Q^b_{d,e} / Q_{i,j}$
        $P_{i,d-1}$ += Δp
        $P^b_{d,e}$ += Δp
        $P^b_{d,e}$ += $P^m_{i,j} \exp\{-[\alpha_2 + \alpha_3(d-i) + \alpha_3(j-e)]/RT\}\, Q^b_{d,e} / Q^m_{i,j}$
        $\Delta p = P^m_{i,j} Q^m_{i,d-1}\, Q^b_{d,e} \exp\{-[\alpha_2 + \alpha_3(j-e)]/RT\} / Q^m_{i,j}$
        $P^m_{i,d-1}$ += Δp
        $P^b_{d,e}$ += Δp
        $\Delta p = P^z_{i,j} Q^z_{i,d-1} Q^b_{d,e} \exp\{-[\beta_2 + \beta_3(j-e)]/RT\} / Q^z_{i,j}$
        $P^z_{i,d-1}$ += Δp
        $P^b_{d,e}$ += Δp
      for $d = i, j-8$ // all possible rightmost pseudoknots filling $[d, e]$
        for $e = d+8, j$
          $\Delta p = P_{i,j} Q_{i,d-1} Q^p_{d,e} \exp\{-\beta_1/RT\} / Q_{i,j}$
          $P_{i,d-1}$ += Δp
          $P^p_{d,e}$ += Δp
          $P^p_{d,e}$ += $P^m_{i,j} \exp\{-[\beta_1^m + 2\alpha_2 + \alpha_3(d-i+j-e)]/RT\} Q^p_{d,e} / Q^m_{i,j}$
          $\Delta p = P^m_{i,j} Q^m_{i,d-1} Q^p_{d,e} \exp\{-[\beta_1^m + 2\alpha_2 + \alpha_3(j-e)]/RT\} / Q^m_{i,j}$
          $P^m_{i,d-1}$ += Δp
          $P^p_{d,e}$ += Δp
          $\Delta p = P^z_{i,j} Q^z_{i,d-1} Q^p_{d,e} \exp\{-[\beta_1^p + 2\beta_2 + \beta_3(j-e)]/RT\} / Q^z_{i,j}$
          $P^z_{i,d-1}$ += Δp
          $P^p_{d,e}$ += Δp
    //P^p **recursion**
    for $d = i+2, j-4$
      for $e = \max(d+2, i+5), j-3$
        for $f = e+1, j-2$
          $\Delta p = P^p_{i,j} Q^{gl}_{i,d-1,e,f} Q^{gr}_{d,e-1,f+1,j} / Q^p_{i,j}$
          $P^{gl}_{i,d-1,e,f}$ += Δp
          $P^{gr}_{d,e-1,f+1,j}$ += Δp
    // P^{gr} **recursion**
    for $d = i+1, j-4$
      for $e = d+3, j-1$
        for $f = e, j-1$
          $\Delta p = P^{gr}_{i,d,e,j} Q^{gl}_{i,d,f,j} Q^z_{e,f-1} / Q^{gr}_{i,d,e,j}$
          $P^{gl}_{i,d,f,j}$ += Δp
          $P^z_{e,f-1}$ += Δp
    // P^{gl} **recursion**
    for $d = i+1, j-5$
      for $f = d+4, j-1$
        for $e = d, f-3$
          $\Delta p = P^{gl}_{i,e,f,j} Q^g_{i,d,f,j} Q^z_{d+1,e} \exp(-\beta_2/RT) / Q^{gl}_{i,e,f,j}$
          $P^g_{i,d,f,j}$ += Δp
          $P^z_{d+1,e}$ += Δp
          $P^{bg}_{d,f}$ += Δp // P^{bg} inner gap-spanning base-pairing prob
    // P^{grs} **recursion**
    for $d = i+1, j-10$
      for $e = d+4, j-6$
        for $f = e+1, j-5$
          $\Delta p = P^{grs}_{i,d,e,j} Q^g_{i,d,e,f} Q^m_{f+1,j} \exp(-\alpha_2/RT) / Q^{grs}_{i,d,e,j}$
          $P^g_{i,d,e,f}$ += Δp
          $P^m_{f+1,j}$ += Δp
    // P^{gls} **recursion**
    for $c = i+5, j-6$
      for $d = c+1, j-5$
        for $e = d+4, j-1$
          $\Delta p = P^{gls}_{i,d,e,j} \exp(-\alpha_2/RT) Q^m_{i,c-1} Q^g_{c,d,e,j} / Q^{gls}_{i,d,e,j}$
          $P^m_{i,c-1}$ += Δp
          $P^g_{c,d,e,j}$ += Δp
    // P^g **recursion**
    for $d = i+2, j-6$ // simple interior loops
      for $e = d+4, j-2$
        for $c = i+1, d-1$
          for $f = e+1, j-1$
            $P^g_{c,d,e,f}$ += $P^g_{i,d,e,j} \exp(-G^{\text{interior}}_{i,c,f,j}/RT) Q^g_{c,d,e,f} / Q^g_{i,d,e,j}$
    // for $O(N^5)$ version, replace previous five lines with:
    // **call** fastiloopsN5$(i, j, l, Q^g, Q^x, Q^{x2}, P^g, P^x, P^{x2})$
    for $d = i+6, j-5$
      for $e = d+4, j-1$
        $\Delta p = P^g_{i,d,e,j} Q^m_{i+1,d-1} \exp\{-[\alpha_1 + 2\alpha_2 + \alpha_3(j-e-1)]/RT\} / Q^g_{i,d,e,j}$
        $P^m_{i+1,d-1}$ += Δp
    for $d = i+1, j-10$
      for $e = d+4, j-6$
        $\Delta p = P^g_{i,d,e,j} \exp\{-[\alpha_1 + 2\alpha_2 + \alpha_3(d-i-1)]/RT\} Q^m_{e+1,j-1} / Q^g_{i,d,e,j}$
        $P^m_{e+1,j-1}$ += Δp
    for $d = i+6, j-10$
      for $e = d+4, j-6$
        $\Delta p = P^g_{i,d,e,j} Q^m_{i+1,d-1} \exp\{-[\alpha_1 + 2\alpha_2]/RT\} Q^m_{e+1,j-1} / Q^g_{i,d,e,j}$
        $P^m_{i+1,d-1}$ += Δp
        $P^m_{e+1,j-1}$ += Δp
    for $d = i+7, j-6$
      for $e = d+4, j-2$
        for $f = e+1, j-1$
          $\Delta p = P^g_{i,d,e,j} Q^{gls}_{i+1,d,e,f} \exp\{-[\alpha_1 + \alpha_2 + \alpha_3(j-f-1)]/RT\} / Q^g_{i,d,e,j}$
          $P^{gls}_{i+1,d,e,f}$ += Δp
    for $d = i+2, j-11$
      for $e = d+4, j-7$
        for $c = i+1, d-1$
          $\Delta p = P^g_{i,d,e,j} \exp\{-[\alpha_1 + \alpha_2 + \alpha_3(c-i-1)]/RT\} Q^{grs}_{c,d,e,j-1} / Q^g_{i,d,e,j}$
          $P^{grs}_{c,d,e,j-1}$ += Δp
    for $d = i+7, j-11$
      for $e = d+4, j-7$
        for $c = i+6, d-1$
          $\Delta p = P^g_{i,d,e,j} Q^m_{i+1,c-1} Q^{grs}_{c,d,e,j-1} \exp\{-[\alpha_1 + \alpha_2]/RT\} / Q^g_{i,d,e,j}$
          $P^m_{i+1,c-1}$ += Δp
          $P^{grs}_{c,d,e,j-1}$ += Δp
    // P^{bg} outer gap-spanning base-pairing prob
    for $d = i+1, j-5$
      for $e = d+4, j-1$
        $P^{bg}_{i,j}$ += $P^g_{i,d,e,j}$
    // P^b **recursion**
    for $d = i+1, j-5$ // all possible rightmost pairs $d{\cdot}e$
      for $e = d+4, j-1$
        $P^b_{d,e}$ += $P^b_{i,j} \exp(-G^{\text{interior}}_{i,d,e,j}/RT) Q^b_{d,e} / Q^b_{i,j}$
        $\Delta p = P^b_{i,j} Q^m_{i+1,d-1}\, Q^b_{d,e} \exp\{-[\alpha_1 + 2\alpha_2 + \alpha_3(j-e-1)]/RT\} / Q^b_{i,j}$
        $P^m_{i+1,d-1}$ += Δp
        $P^b_{d,e}$ += Δp
    for $d = i+1, j-9$ // all possible rightmost pseudoknots filling $[d, e]$
      for $e = d+8, j-1$
        $G^{\text{recursion}} = \alpha_1 + \beta_1^m + 3\alpha_2 + \alpha_3(j-e-1)$
        $P^p_{d,e}$ += $P^b_{i,j} \exp\{-[G^{\text{recursion}} + \alpha_3(d-i-1)]/RT\} Q^p_{d,e} / Q^b_{i,j}$
        $\Delta p = P^b_{i,j} Q^m_{i+1,d-1}\, Q^p_{d,e} \exp\{-G^{\text{recursion}}/RT\} / Q^b_{i,j}$
        $P^m_{i+1,d-1}$ += Δp
        $P^p_{d,e}$ += Δp

Figure 3.4. $O(N^6)$ recursion probability algorithm that includes a class of pseudoknots. Modifications required to produce an $O(N^5)$ version of the algorithm are noted in comments. See Appendix C for details.

pairs and one for non-nested gap-spanning pairs.

To see the utility of this decomposition, calculations were run on wild-type and mutant sequences of a pseudoknot construct derived from human telomerase RNA [16]. Experimental evidence suggests that this pseudoknot exists in equilibrium with an alternative, hairpin structure, and that this equilibrium functions as a biological switch [3]. A two-point mutant, found in a small percentage of people, shifts the equilibrium towards the hairpin structure, leading to a disease known as dyskeratosis congenita [3]. Feigon et al. [16] examine this shift in equilibrium for segments of the wild-type and mutant sequences described in figure 3.1, revealing that the pseudoknot conformation dominates the hairpin for the wild-type sequence (∼95% to ∼5%) but competes roughly equally in the mutant sequence (∼50% to ∼50%). Using preliminary pseudoknot parameters [6], energies were computed for both the wild-type sequence and the two-point mutant on the psuedoknotted and hairpin structures. The predicted energies in Table 3.1 match reasonably well with experimental values [16]. For the wildtype sequence, the disparity between the pseudoknot and hairpin energies suggests an equilibrium that favors the more stable pseudoknot. In contrast, the energies for the double mutant sequence suggest a more balanced equilibrium. Figures 3.5 and 3.6 illustrate that the hairpin conformation has a significant impact on the pair probabilities for the mutant, but not for the wild-type sequence.

Base-pairing probabilities can also be used to construct metrics for evaluating nucleic acid designs. The secondary structure s may be described by a symmetric $N \times N$ matrix S with entries $S_{i,j} = 1$ if s contains base pair $i \cdot j$ and $S_{i,j} = 0$ otherwise. We augment this matrix by an additional column with entries $S_{i,N+1} = 1$ if base i is unpaired and $S_{i,N+1} = 0$ otherwise. Hence, each row sum is one. For a given sequence of length N, the metric [5]

$$n(s^*) = N - \sum_{\substack{1 \le i \le N \\ 1 \le j \le N+1}} p_{i,j} S^*_{i,j}$$

represents the average number of nucleotides that differ from the target secondary structure s^* at

Table 3.1. Energy comparisons for human telomerase RNA constructs

RNA	Conformation	Energies (kcal/mol) $\Delta G^\circ_{\text{exp}}$	$\Delta G^\circ_{\text{calc}}$
Wild-type	Pseudoknot	-17.8	-18.5[2]
	Hairpin	-9.8[1]	-11.5[3]
Mutant	Pseudoknot	-11.2	-11.3[2]
	Hairpin	-10.5[1]	-11.5[3]

[1] Experiments were performed on partial sequences that excluded the 18 nucleotides on the 3' end to prevent the formation of pseudoknots [16]. This truncation does not affect the corresponding $\Delta G^\circ_{\text{calc}}$.
[2] A related pseudoknot structure that is otherwise identical but omits the three consecutive $A{\cdot}U$ pairs in the stem with the bulge loop is predicted to be 0.5 kcal/mol more stable.
[3] The secondary structure energy calculation ignores the four consecutive non-canonical base pairs that are observed to close the interior loop in the hairpin stem [16].

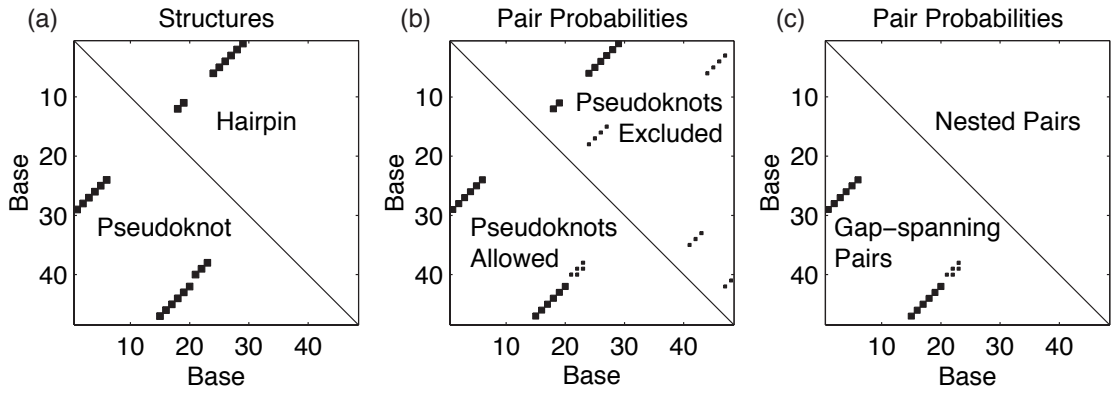


Figure 3.5. Dot plots for wild-type human telomerase RNA. (a) Pseudoknot (bottom left) and hairpin (top right) constructs. For (b) and (c), large dots indicate a $p_{i,j} \geq 0.5$ and small dots indicate $0.5 > p_{i,j} \geq 0.05$. (b) Base-pairing probabilities including pseudoknots (bottom left) and excluding pseudoknots (top right). (c) A decomposition of the full base-pairing probabilities into gap-spanning pairs (bottom left) and nested pairs (top right). Note that there are no nested pairs with significant probability, indicating that pseudoknot conformations are dominating the equilibrium.

thermodynamic equilibrium. This is a less stringent metric than $p(s^*)$, the probability that the sequence exactly adopts structure s^*; even if $p(s^*)$ is not close to unity, $n(s^*)$ can still be small if the equilibrium ensemble is dominated by structures that differ only slightly from s^*.

It is illustrative to compare the two metrics on a real design problem involving pseudoknots. For example, Winfree *et al.* [22] designed and constructed DNA double-crossover molecules [7] that interact to form a two-dimensional lattice with a pseudoknotted unit cell. These sequence designs were performed using sequence symmetry minimization [15] to ensure that incorrectly paired subsequences of length six would always contain at least one mismatch and most incorrectly paired

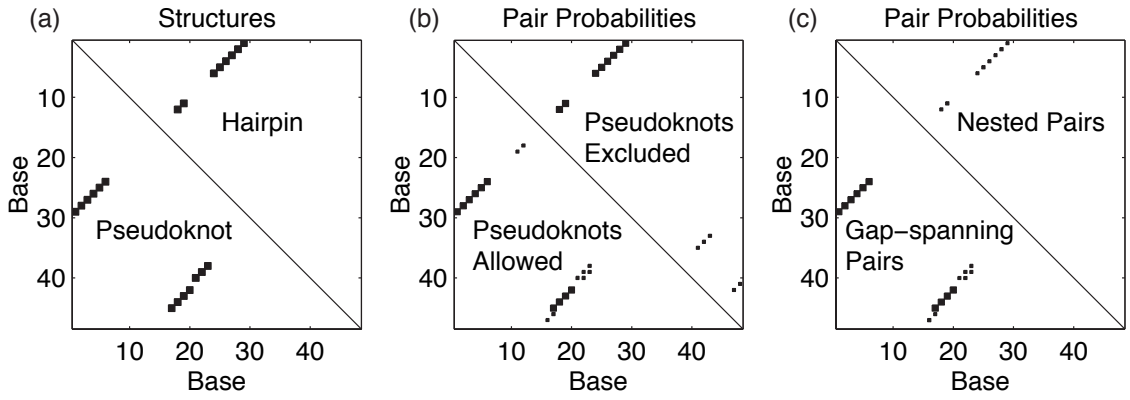


Figure 3.6. Dot plots for double mutant human telomerase RNA. The plots are analagous to those of figure 3.5. The key difference is observed in (c), where the hairpin stem appears as both gap-spanning pairs and nested pairs, indicating the increased significance of hairpin conformations.

subsequences of length five would also contain a mismatch [22]. Lacking DNA pseudoknot parameters, we examine an RNA analog of their sequence for the portion of the pseudoknotted unit cell depicted in figure 3.7a. The probability of adopting the target structure is $p(s^*) = 0.1$ and the average number of incorrect nucleotides is $n(s^*) = 4.0$. The low value of $p(s^*)$ might indicate a cause for concern, but for a structure with 90 nucleotides and helices of length eight, the average number of incorrect nucleotides is relatively small. Hence, it is not surprising that the sequence behaves well experimentally, demonstrating the correct base-pairing topology despite slight predicted variations at the ends of helices. The dot plot in figure 3.7b illustrates the similarity between the average structure and the desired target.

Interestingly, design methods described in previous work [5] can be used, in conjunction with the pseudoknot partition function algorithm, to find sequences that achieve $p(s^*) = 0.98$ and $n(s^*) \ll 1$. It is unclear whether these sequences would provide any experimental benefit for this system (even given a perfect energy model), since the difference between $n(s^*) \approx 4$ and $n(s^*) \ll 1$ may be lost in experimental noise. By contrast, if a sequence produced $p(s^*) = 0.1$ with $n(s^*) \gg 4$, then the equilibrium ensemble could include important structures differing significantly from the target structure.

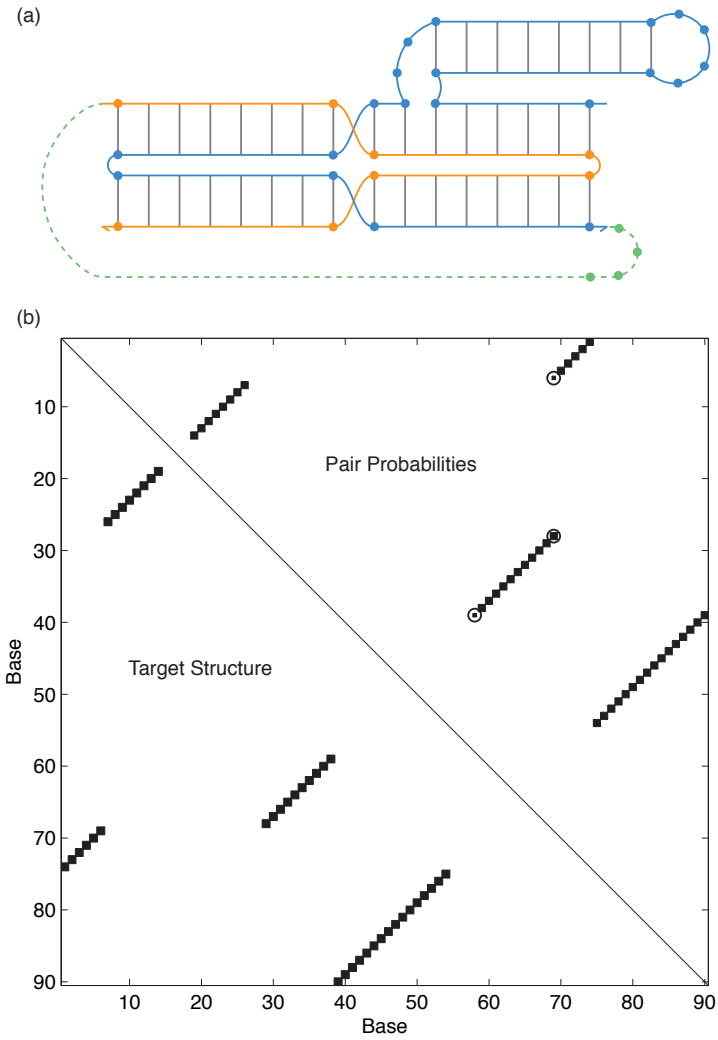


Figure 3.7. Computational examination of a pseudoknotted DNA nanostructure. (a) Secondary structure for a double-crossover molecule that forms a portion of the unit cell in a two-dimensional lattice [22]. For our computational study, we join the blue and orange strands (arrows denote 3') into a single strand using auxiliary nucleotides (green) to facilitate the use of the single-stranded partition function algorithm [6]. In the absence of DNA pseudoknot parameters, we consider the RNA analog 5'-CCAACUCCUAGCGAUUUUUCGCUAGGUUUACCAGAUCCACAAGCCGACGUUACAUUUU- GGAUCUGGUAAGUUGGUGUAACGUCGGCUUGU-3', where the interior hyphens denote the boundaries of the auxiliary linker segment. (b) Dot plot analysis of the designed sequence. The bottom left depicts the base pairs in the target structure, and the upper right depicts the base-pairing probabilities. Large dots indicate a $p_{i,j} \geq 0.5$ and small dots indicate $0.5 > p_{i,j} \geq 0.05$. The circles indicate the major differences between the target structure and the calculated pair probabilities.

3.6 Conclusions

A general transformation rule extends nucleic acid partition function algorithms to calculate recursion probabilities, which in turn can be used to compute base-pairing probabilities. We use this approach to derive an algorithm for computing base-pairing probabilities starting from a partition function algorithm that includes a class of pseudoknots. The same strategy will apply to future partition function extensions that follow the same dynamic programming paradigm.

To demonstrate the utility of base-pairing probabilities, calculations were performed on a pseudoknot/hairpin construct thought to represent an important biological switch. In agreement with experimental evidence, the computational results indicate that the pseudoknot dominates the hairpin for the wild-type sequence, but not for the double mutant. Base-pairing probabilities were also used to examine the ensemble properties of a synthetic nucleic acid sequence designed to assemble into a pseudoknotted double-crossover molecule. The average number of incorrect nucleotides was found to be small, suggesting that the relatively low computed probability of adopting the target secondary structure should not significantly affect the experimental performance of the molecule.

Bibliography

[1] T. Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104:45–62, 2000.

[2] S. Bonhoeffer, J. S. McCaskill, P. F. Stadler, and P. Schuster. RNA multi-structure landscapes. *European Biophysics Journal*, 22:13–24, 1993.

[3] L. R. Comolli, I. Smirnov, L. Xu, E. H. Blackburn, and T. L. James. A molecular switch underlies a human telomerase disease. *Proceedings of the National Academy of Sciences of the United States of America*, 99(26):16998–17003, 2002.

[4] Y. Ding and C. E. Lawrence. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Research*, 31(24):7280–7301, 2003.

[5] R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–1403, 2004.

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

[7] T.-J. Fu and N. C. Seeman. DNA double-crossover molecules. *Biochemistry*, 32:3211–3220, 1993.

[8] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Chemical Monthly*, 125:167–188, 1994.

[9] R. B. Lyngso, M. Zuker, and C. N. S. Pedersen. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, 15(6):440–445, 1999.

[10] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940, 1999.

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

[12] R. Nussinov, J. R. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algorithms for loop matchings. *SIAM Journal of Applied Mathematics*, 35:68–82, 1978.

[13] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285:2053–2068, 1999.

[14] J. Santalucia Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences of the United States of America*, 95(4):1460–1465, 1998.

[15] N. C. Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Bioogy.*, 99:237–247, 1982.

[16] C. A. Theimer, L. D. Finger, L. Trantirek, and J. Feigon. Mutations linked to dyskeratosis congenita cause changes in the structural equilibrium in telomerase RNA. *Proceedings of the National Academy of Sciences of the United States of America*, 100(2):449–454, 2003.

[17] I. Tinoco Jr., O. C. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[18] D. H. Turner, N. Sugimoto, and S. M. Freier. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17:167–192, 1988.

[19] F. H. D. van Batenburg, A. P. Gultyaev, C. W. A. Pleij, and J. Ng. Pseudobase: a database with RNA pseudoknots. *Nucleic Acids Research*, 28:201–204, 2000.

[20] M. S. Waterman. Secondary structure of single-stranded nucleic acids. In *Studies in foundations and combinatorics: Advances in Mathematics Supplemental Studies*, volume 1, pages 167–212. Academic Press, New York, 1978.

[21] M. S. Waterman and T. F. Smith. RNA secondary structure: a complete mathematical analysis. *Mathematical Biosciences*, 42:257–266, 1978.

[22] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

[23] H. Yan, T. H. LaBean, L. Feng, and J. H. Reif. Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices. *Proceedings of the National Academy of Sciences of the United States of America*, 100(14):8103–8108, 2003.

[24] M. Zuker. Calculating nucleic acid secondary structure. *Current Opinion in Structural Biology*, 10:303–310, 2000.

[25] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–147, 1981.

Chapter 4

Design of Nucleic Acid Secondary Structures

The work presented here is heavily based on:

R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce, *Paradigms for computational nucleic acid design.*

Nucleic Acids Research, 2004. **32**(4): p. 1392-1403.

4.1 Abstract

The design of DNA and RNA sequences is critical for many endeavors, from DNA nanotechnology, to PCR-based applications, to DNA hybridization arrays. Results in the literature rely on a wide variety of design criteria adapted to the particular requirements of each application. Using an extensively-studied thermodynamic model, we perform a detailed study of several criteria for designing sequences intended to adopt a target secondary structure. We conclude that superior design methods should explicitly implement both a positive design paradigm (optimize affinity for the target structure) and a negative design paradigm (optimize specificity for the target structure). The commonly used approaches of sequence symmetry minimization and minimum free energy satisfaction primarily implement negative design and can be strengthened by introducing a positive design component. Surprisingly, our findings hold for a wide range of secondary structures and are robust to modest perturbation of the thermodynamic parameters used for evaluating sequence quality, suggesting the feasibility and ongoing utility of a unified approach to nucleic acid design as parameter sets are further refined. Finally, we observe that designing for thermodynamic stability does not determine folding kinetics, emphasizing the opportunity for extending design criteria to target kinetic features of the energy landscape.

4.2 Introduction

Understanding how to design molecular structures is an essential step in allowing technology to interface with biology and in developing systems with increasing functional density. Nucleic acids hold great promise as a design medium for the construction of nanoscale devices with novel mechanical or chemical function [36, 37]. Efforts are currently underway in many laboratories to use DNA and RNA molecules for applications in patterning [45], assembly [21, 7, 25], transport, switching [41, 47, 46], circuitry [42], DNA computing [4], and DNA chips [40, 6]. Computational sequence selection algorithms [36, 38, 18, 28, 13, 5, 2, 20, 11] are likely to play an increasing role in exploring this new design space.

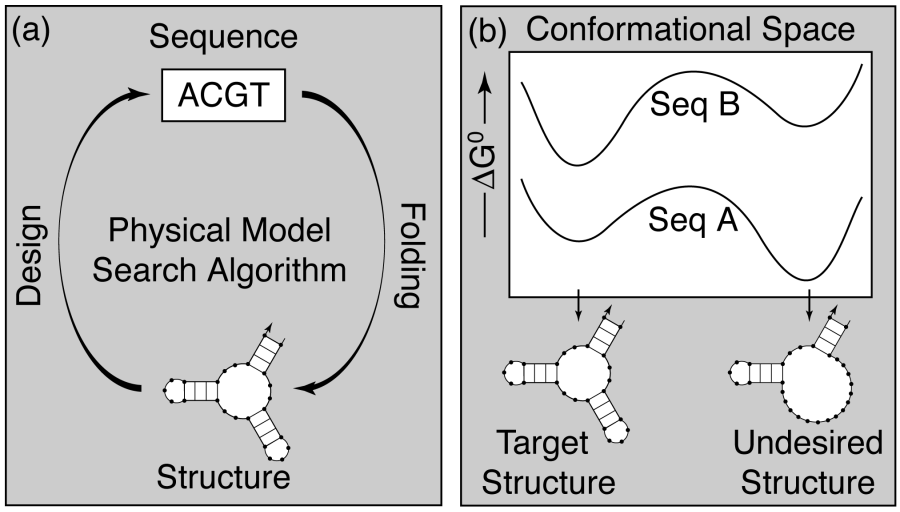


Figure 4.1. (a) Feedback loop for evaluating nucleic acid sequence designs and methodologies. (b) Positive and negative design paradigms. Two sequences are evaluated using an empirical potential on both the desired target structure and an undesired structure. Using a positive design paradigm, sequence A would be selected since it exhibits a stronger affinity than sequence B for the target structure (i.e. lower ΔG°). Using a negative design paradigm, sequence B would be selected since it exhibits specificity for the target structure while sequence A exhibits specificity for the undesired structure. To provide a common basis for comparison, $\Delta G^\circ = 0$ for a strand with no base pairs.

A fundamental design problem consists of selecting the sequence of a nucleic acid strand that will adopt a target secondary structure. As depicted in figure 4.1a, this is the inverse of the more famous folding problem of determining the structure (and folding mechanism) for a given sequence. To attempt the rational design of novel nucleic acid structures, we require both an approximate empirical physical model and a search algorithm for selecting promising sequences based on this model. Experimental feedback on the quality of the design and the performance of the design algorithm can then be obtained by folding the molecule *in vitro*. Alternatively, if this feedback loop can be closed computationally by folding the molecule *in silico*, the quality of sequence designs could be rapidly assessed and improved before attempting laboratory validation.

In designing nucleic acid sequences, we consider the two principal paradigms illustrated in figure 4.1b. *Positive design* methods attempt to select for a desired outcome by optimizing sequence affinity for the target structure. *Negative design* methods attempt to select against unwanted outcomes by optimizing sequence specificity for the target structure. A successful design must exhibit both high affinity and high specificity [38], so useful design algorithms must satisfy the objectives of

both paradigms, even if they explicitly implement only one.

For some applications, it may be desirable to supplement these thermodynamic design considerations with additional kinetic requirements. For example, in designing molecular machines [47], selecting sequences that fold or assemble quickly may be crucial, since naturally occuring RNA sequences have been observed to have persistent metastable states [3] and theoretical models suggest that random sequences have highly frustrated energy landscapes with folding times that grow exponentially with sequence length [17]. Alternatively, it may be important to design interactions with intentionally frustrated folding kinetics in order to control fuel delivery during the work cycle [43].

The present study uses efficient partition function algorithms and stochastic kinetics simulations to examine the thermodynamic and kinetic properties of sequences designed using seven methods that capture aspects of the positive and negative design paradigms. Although several of these design criteria have been widely used, we are not aware of any previous attempt to assess their relative performance. Evaluated based on thermodynamic considerations, we consistently observe that sequence selection methods that implement both positive and negative design paradigms outperform methods that implement either paradigm alone. This trend appears to be robust to changes in both the target secondary structure and the parameters in the physical model, and to the choice of either RNA or DNA as the design material. The trend does not hold when the design criteria are judged based on kinetic considerations, as favorable thermodynamic properties do not ensure fast folding.

4.3 Physical Model

The secondary structure of a nucleic acid strand is a list of base pairs; the precise form is described in sections 1, 2.2 and 2.3.1. A coarse-grained energy landscape may be defined over the finite number of all possible secondary structures, where the properties of each secondary structure represent an ensemble average over the three-dimensional atomic structures consistent with that base-pairing graph.

The folding kinetics of a sequence can be addressed by simulating the trajectory through secondary structure space as a continuous-time Markov process [14, 12]. Changes in secondary structure

are described in terms of elementary steps corresponding to the breaking or formation of a single base pair. For each elementary step, the ratio of the forward and backward rates is defined to be consistent with the equilibrium probabilities of the two end states [12, 48]. However, *ad hoc* arguments are required to set the magnitude of the rates. There is some evidence that qualitative properties of kinetic simulations are insensitive to the specific rate model [12].

4.4 Thermodynamic and Kinetic Evaluation Metrics

The partition function over secondary structure space provides an ideal conceptual framework for evaluating the affinity and specificity of a sequence for the target structure. If $\Delta G^\circ(s)$ is the free energy of a sequence in secondary structure s, then the probability of sampling s at thermodynamic equilibrium is given by

$$p(s) = \tfrac{1}{Q} e^{-\Delta G^\circ(s)/RT},$$

where the partition function

$$Q = \sum_{s \in \Omega} e^{-\Delta G^\circ(s)/RT},$$

is a weighted sum over the set of all secondary structures Ω, R is the universal gas constant and T is the temperature. If the probability $p(s^*)$ of folding to the target graph s^* is close to unity, then within the context of the approximate physical model, the sequence achieves both high affinity and high specificity for the target structure.

The probability $p(s^*)$ represents a very stringent design evaluation criterion since it measures the probability that every nucleotide exactly matches the target graph. For some applications (e.g. those involving large DNA molecules where some "breathing" is unavoidable), it is acceptable to use sequences that adopt an ensemble of secondary structures similar to the target graph. In such cases, requiring $p(s^*)$ to be close to unity is a sufficient but not necessary condition for identifying satisfactory sequence designs.

A more lenient design criterion may be obtained by using a modified form of the partition function algorithm to compute the matrix of base-pair probabilities [30] with entries $P_{i,j} \in [0, 1]$ corresponding

to the probability of forming base pair $i{\cdot}j$. By comparing the entries of P to the structure matrix S^* with entries $S^*_{i,j} \in \{0,1\}$ describing the target secondary structure s^*, we may compute the average number of incorrect nucleotides $n(s^*)$ over the equilibrium ensemble of secondary structures Ω.

The derivation of $n(s^*)$ for a strand of length N proceeds as follows. Each secondary structure $s \in \Omega$ is defined by a symmetric $N \times N$ structure matrix S with entries $S_{i,j} = 1$ if s contains base pair $i \cdot j$ and $S_{i,j} = 0$ otherwise. We augment the matrix S by adding an additional column with entries $S_{i,N+1} = 1$ if base i is unpaired and $S_{i,N+1} = 0$ otherwise. Hence, every row sum is one. Using the same convention, the augmented structure matrix corresponding to the target structure s^* is denoted S^*. Given a sequence, if the probability of sampling structure s is $p(s)$, then the average number of incorrect nucleotides may be expressed

$$n(s^*) = N - \sum_{s \in \Omega} \Bigg[p(s) \sum_{\substack{1 \le i \le N \\ 1 \le j \le N+1}} S_{i,j}\, S^*_{i,j} \Bigg].$$

This may be rearranged to give

$$n(s^*) = N - \sum_{\substack{1 \le i \le N \\ 1 \le j \le N+1}} \Bigg[\sum_{s \in \Omega} p(s)\, S_{i,j} \Bigg] S^*_{i,j},$$

where the quantity in brackets is just the matrix of pair probabilities P with entries $P_{i,j}$ equal to the probability of forming base pair $i{\cdot}j$. The extra column has entries $P_{i,N+1}$ equal to the probability that base i is unpaired. Again, each row sum is one. Hence the average number of incorrect nucleotides[1]

[1] The formula for $n(s^*)$ is a special case of a general metric, $d(p, p')$, between two ensembles of secondary structures, p and p', that measures the average number of differing nucleotides when one secondary structure is chosen from each ensemble. By a derivation similar to the one above,

$$d(p, p') = N - \sum_{\substack{1 \le i \le N \\ 1 \le j \le N+1}} P_{i,j} P'_{i,j}.$$

The metric of Morgan and Higgs [17] is equivalent to $d(p, p)$ and $n(s^*) = d(p, s^*)$, where we abuse notation to indicate that the probability distribution is concentrated entirely on the target structure s^*.

may be expressed

$$n(s^*) = N - \sum_{\substack{1 \le i \le N \\ 1 \le j \le N+1}} P_{i,j} S^*_{i,j}.$$

This metric has the advantage that sequences that adopt secondary structures similar to s^* (e.g. due to breathing) are now identified as promising candidates. However, even if $n(s^*) << N$, it is possible that the consistent omission or addition of certain base pairs (e.g. a hairpin stem) may cause dramatic changes to the geometric structure. The requirement that $n(s^*) << N$ is necessary but not sufficient to ensure that $p(s^*)$ is close to unity. On the other hand, $n(s^*) \approx 0$ is both necessary and sufficient to ensure $p(s^*) \approx 1$.

We measure folding efficiency as the median time, $t(s^*)$, to achieve the target structure starting from a random coil initial condition (no secondary structure). This metric is distinct from fast folding time when the target structure is not the minimum free energy structure. Thus, $t(s^*)$ being small is neither necessary nor sufficient to imply $p(s^*)$ is near unity. In this chapter, we consider ideal sequences to be those with $p(s^*) \approx 1$, $n(s^*) \approx 0$ and $t(s^*)$ small.

4.5 Design Criteria

We evaluate the following sequence design criteria:

1. Random. Sequences are selected to satisfy the complementarity requirements of the base-pairing graph but are otherwise random. This is a primitive approach to both positive and negative design; compatibility with the target graph implies some affinity for the structure and incompatibility with many other graphs. At least this mild level of positive and negative design is implicit in each of the design methods that follow.

2. Energy Minimization. Sequences are selected that attain a low energy on the target structure using the standard energy model. This approach implements explicit positive design.

3. Minimum Free Energy (MFE) Satisfaction. Sequences are selected to ensure that the target structure is the lowest energy structure [18, 2]. Note that a sequence with the correct minimum

energy structure may nonetheless have a low probability of adopting the target fold. This approach implements explicit negative design.

4. Sequence Symmetry Minimization (SSM). Sequences are selected to prohibit repeated subsequences of a specified *word* length [36]. For subsequences that are not base-paired to consecutive bases in the target graph (e.g. single stranded or branched regions), the complementary words are also prohibited from appearing in the design. This is a heuristic approach to negative design, attempting to ensure specificity for the target structure by guaranteeing mismatches within any subsequence of the word length that hybridizes incorrectly.

5. Energy Minimization and SSM. Sequences are selected that attain a low energy on the target graph subject to the constraint that SSM is satisfied [38]. This approach explicitly addresses both paradigms, combining rigorous positive design and heuristic negative design.

6. Probability. Sequences are selected to maximize the probability [18, 13, 11] of sampling the target structure $p(s^*)$. Positive and negative design are simultaneously addressed in a single rigorous approach.

7. Average Incorrect Nucleotides. Sequences are selected to minimize the average number of incorrect nucleotides $n(s^*)$. Positive and negative design are simultaneously addressed in a single rigorous approach.

In each case, a design method is obtained by employing a heuristic search procedure to optimize one of the design criteria. It is these design criteria that are the focus of the present work. Examining a set of sequences obtained by independent search processes provides a characterization of typical performance. For the the random method, as well as MFE satisfaction and SSM, any sequence that satisfies the criterion is a global minimum. For the probability and average incorrect nucleotide methods, the global optimum is not necessarily attained, but there is an absolute standard of success (i.e. $p(s^*) \approx 1$ or $n(s^*) \approx 0$) that is frequently achieved. For methods involving energy minimization, there is no mathematical guarantee that the selected sequences are near the global minimum.[2]

[2]For energy minimization, the global minimum energy is achieved by at least one sequence for all structures we consider. For energy minimization plus SSM, we verified this property only for small structures, (e.g. the one in figure 4.1a), where the global minimum was determined using a branch and bound algorithm [15] (see Materials and Methods).

Table 4.1. Sequence statistics for RNA multiloop designs of figure 4.2

Design Method	$p(s^*)$	$n(s^*)$	CG content	Entropy
Random	0.00	7.22	0.50	1.00
Energy Minimization	0.00	32.46	0.91	0.35
MFE Satisfaction	0.16	4.14	0.50	0.99
SSM	0.08	4.89	0.50	0.99
Energy Minimization & SSM	0.87	0.28	0.66	0.68
Probability	0.97	0.06	0.69	0.40
Average Incorrect Nucleotides	0.97	0.06	0.68	0.39

Implementation details for all design methods are provided in Materials and Methods.

4.6 Results

We now compare the perfomance of these seven design methods. All designed sequences are at local minimima in the sense that no mutation of one base pair or of one unpaired base results in a better sequence based on the given design criterion.

RNA Multiloop Design. Each method was used to perform 100 independent sequence designs for a 4-stem RNA multiloop comprising 71 nucleotides. Histograms of $p(s^*)$ and $n(s^*)$ are shown in figures 4.2a and 4.2b, with median values recorded in table 4.1. For random sequences, approximately 95% of the designs have $p(s^*) < 0.1$ and the median value of $n(s^*)$ is 7.2. Energy minimization performs worse than random while MFE satisfaction and SSM perform somewhat better. There is a dramatic improvement in sequence quality using a combination of energy minimization and SSM. Directly optimizing either $p(s^*)$ or $n(s^*)$ leads to sequences with excellent thermodynamic properties.

To provide an alternative view of average design performance, figure 4.2c depicts the base-pairing probabilities $P_{i,j}$ for the median sequence based on $p(s^*)$. Energy minimization completely fails to capture the connectivity of the target structure. The other methods demonstrate the correct basic structure with varying propensities for extending or adding helices.

Model Robustness. It is inevitable that new parameter sets will continue to be developed for the loop-based potential functions used for these studies [34, 29]. For our design methods to be useful, the quality of a sequence must be robust to perturbations in the approximate physical model; sequences that behave well using many different parameter sets are more likely to perform well in

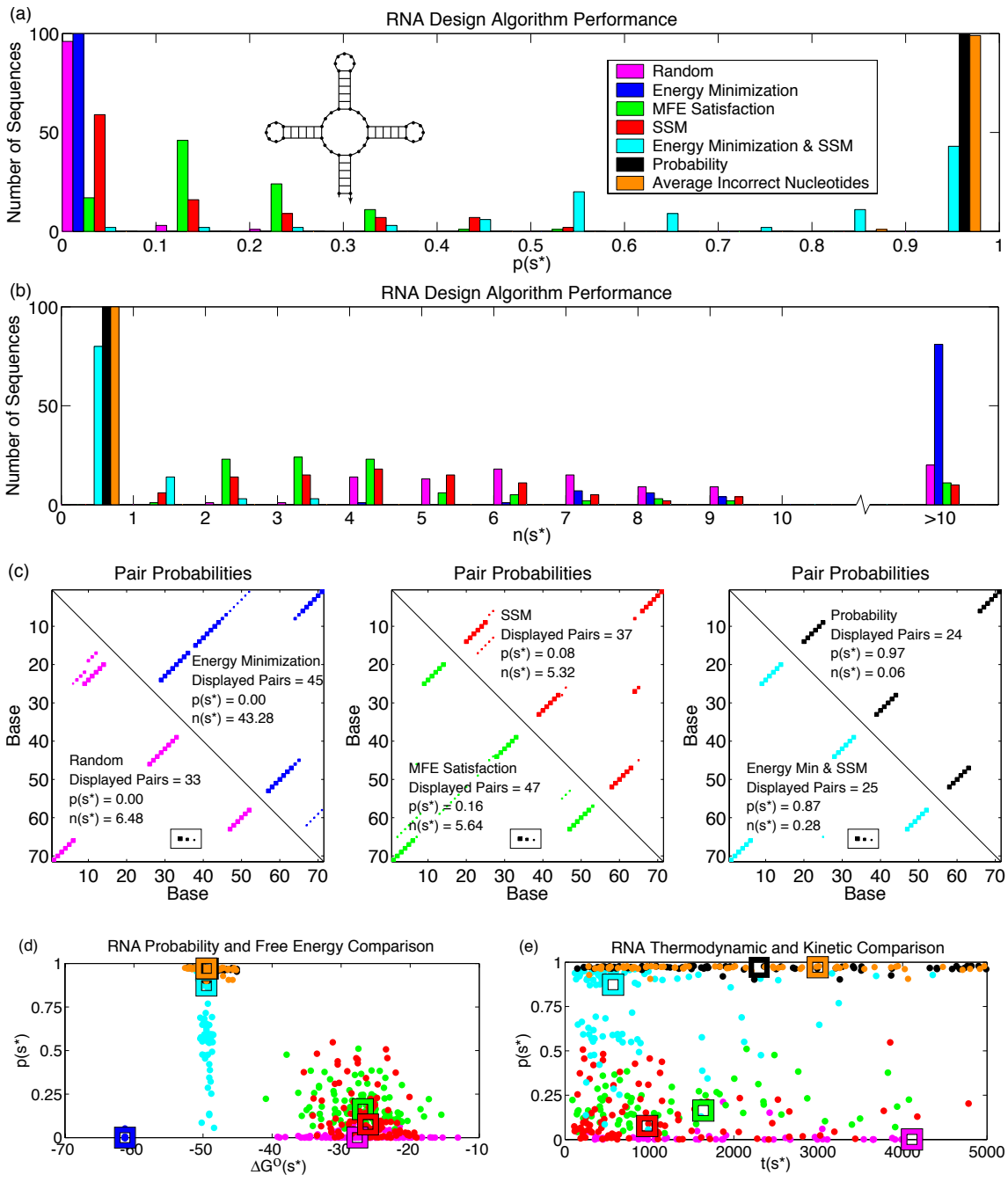


Figure 4.2. RNA multiloop: (a) Histograms for 100 sequence designs based on probability of sampling the target graph, $p(s^*)$. The color legend applies to all plots. (b) Histograms for the same 100 sequence designs based on average number of incorrect nucleotides, $n(s^*)$. (c) Base-pairing probabilities $P_{i,j}$ for the median sequence based on $p(s^*)$. Square sizes correspond to $P_{i,j} \geq \{0.5, 0.05, 0.005\}$, respectively. The target structure is identical to that obtained by optimizing probability (black) or the average number of incorrect nucleotides (not shown). (d) $p(s^*)$ versus free energy, $\Delta G^\circ(s^*)$. Each dot corresponds to one of 100 sequences designed using each method. Each bold square corresponds to the median over the 100 sequences designed using each method. (e) $p(s^*)$ versus median folding time, $t(s^*)$, over 1000 kinetic trajectories starting from random coil initial conditions. Dots and squares interpreted as for part (d).

the laboratory. To examine this issue, we consider 1000 randomized potential functions for RNA where every parameter[3] is independently adjusted by an amount uniformly distributed on ±10%, ±20% or ±50%.

For each design method, the top-ranked sequence based on $p(s^*)$ is reexamined using these modified potentials. The new probabilities are shown in figure 4.3, with the original probabilities depicted as dashed lines. For perturbations distributed uniformly on ±10%, these probability distributions are peaked near the original probabilities, with the sharpest peaks occuring for the best original sequences. The studies with perturbations distributed uniformly on ±20% and ±50% demonstrate that the best sequences are surprisingly robust even to large perturbations.

Sequence Composition. The contrasting behavior of sequences designed by different methods is partly attributable to the variation in sequence composition as summarized by Table 4.1 in terms of fraction of CG nucleotides and average Shannon entropy per position.[4] As expected, the random and SSM designs have a CG content of 50% and an average sequence entropy of approximately one, meaning that each base is equally likely at each position. Similar trends are observed for MFE satisfaction, emphasizing that it is a negative design approach, in that it does not attempt to optimize affinity for the target structure by increasing the CG content. By contrast, energy minimization leads to 91% CG content with a dramatic drop in the sequence entropy at each position. The combined approach of energy minimization and SSM increases the average sequence entropy and reduces the CG content to about 65%. By comparison, designs based on direct optimization of $p(s^*)$ or $n(s^*)$ have similar CG contents, but much lower average sequence entropies, suggesting greater uniformity across independent sequence designs in the placement of C and G bases throughout the strand.

The differing design objectives of methods that implement positive and negative design paradigms are amply illustrated by the plot of probability versus free energy in figure 4.2d. Here, the methods of SSM and MFE satisfaction produce sequences with ΔG° values comparable to those for the random

[3]There are 10,692 and 12,198 nonzero parameters for the RNA [29] and DNA [34] models, respectively.

[4]For 100 sequences designed by a given method, the information entropy at position i is defined by

$$\sigma_i = -\sum_{\eta=A,C,G,U} f_i(\eta) \log_4 f_i(\eta)$$

where $f_i(\eta)$ is the fraction of base η at position i, and σ_i varies between zero (all nucleotides are identical) and one (equal number of each nucleotide). The average entropy per position over a sequence of length N is then $\sum_{i=1,N} \sigma_i/N$.

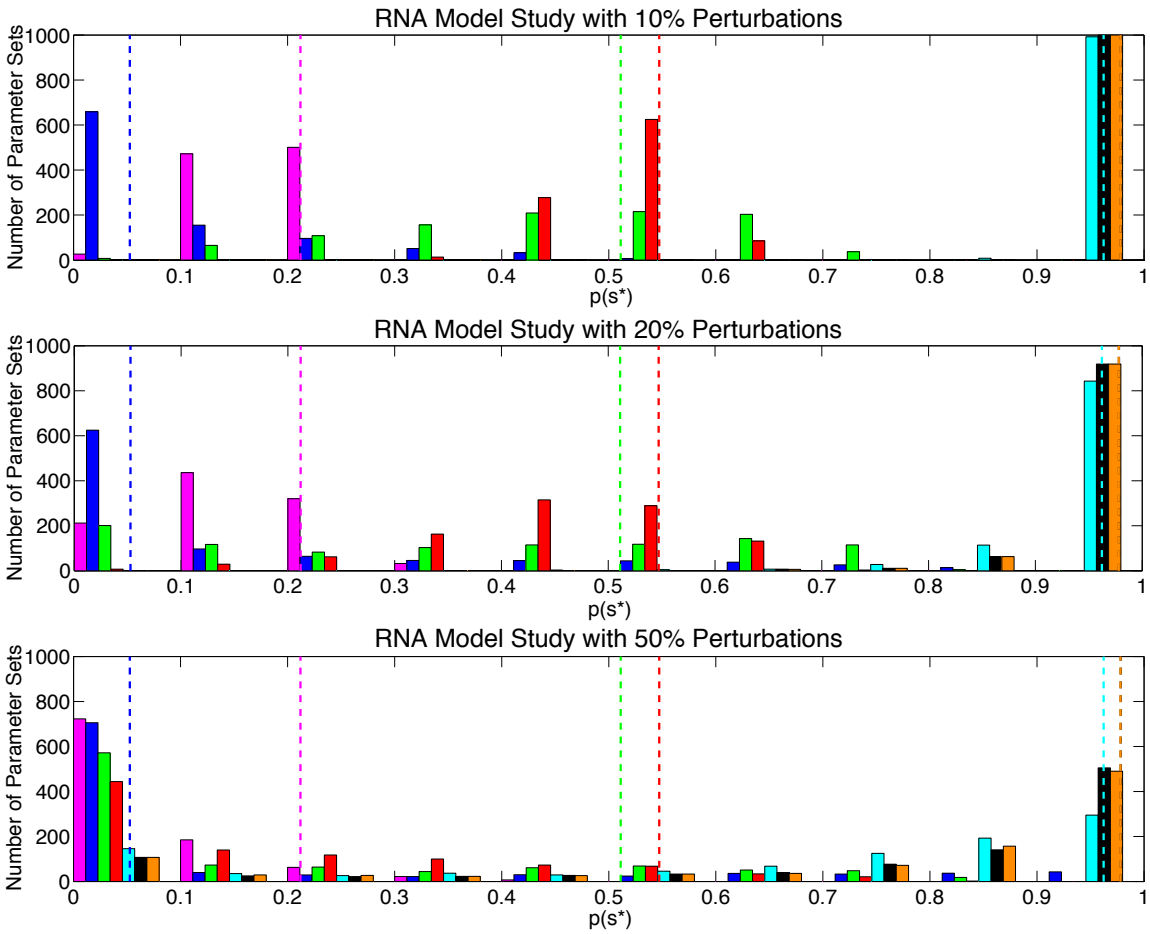


Figure 4.3. RNA model perturbation study. For the multiloop designs of figure 4.2, the top-ranked sequence for each method based on $p(s^*)$ is reexamined using 1000 randomized potential functions where every parameter is independently adjusted by an amount uniformly distributed on $\pm 10\%$, $\pm 20\%$, or $\pm 50\%$.The original probabilities are depicted as dashed lines.

method. Energy minimization naturally produces the lowest ΔG° values, while the methods that combine positive and negative design sacrifice some level of affinity to achieve greater specificity and hence higher $p(s^*)$ values.

Kinetics. We estimate $t(s^*)$ as the median folding time over 1000 stochastic simulation runs as plotted against $p(s^*)$ in figure 4.2e. Each simulation was terminated after 10^4 dimensionless time units had elapsed.

Sequences designed by energy minimization had very low probabilities and failed to fold during the time frame of the simulations. Random sequences also had very low probabilities but did succeed in folding. On average, the negative design approaches of MFE satisfaction and SSM

yielded sequences with improved probabilities and folding times relative to random sequences. The combined approach of energy minimization and SSM yielded significantly higher probabilities with folding times that are comparable with SSM. Sequences designed by direct optimization of $p(s^*)$ or $n(s^*)$ yielded the highest probabilities but somewhat slower folding times. Figure 4.2e illustrates two distinct classes of slow folding sequences: sequences with low $p(s^*)$ have energy landscapes in which s^* is not a prominent local minimum, while sequences with high $p(s^*)$ have s^* as the global minimum, but often have highly frustrated energy landscapes, possibly due to high CG content. Each of the three methods that implement both positive and negative design paradigms produces a number of sequences that appear excellent based on both equilibrium and kinetic properties. However, in general, the depth of the global minimum in the energy landscape does not determine the kinetic accessibility of that conformation [12].

Other RNA Structures. The multiloop structure considered in figure 4.2 had stems of length $\alpha = 6$ and single stranded multiloop regions of length $\beta = 2$. In figure 4.4, the design conclusions are generalized to a related family of multiloop structures with $\alpha \in \{4, 6, 8\}, \beta \in \{0, 2, 4\}$. Results for a larger RNA multiloop with 122 nucleotides and a small RNA pseudoknot with 30 nucleotides are shown in figures 4.5 and 4.6. We have also examined design performance for open structures, hairpins, and three-stem multiloop structures (not shown). In all of these cases, the same trends are observed in the relative performance of the different design methods.

DNA Design. For each of the non-pseudoknotted cases, analogous data is provided for DNA in figures E.1-E.4 and Table E.1. Similar trends are observed in the relative performance of the different design methods. Based on equilibrium properties, the most noticeable differences compared to the RNA designs are: (a) the best methods no longer consistently produce sequences with $p(s^*) > 0.90$, (b) structures with helices of length $\alpha = 4$ are difficult to stabilize. Comparing equilibrium and kinetic properties, higher probabilities are achievable with RNA, and faster folding times are typical for DNA.

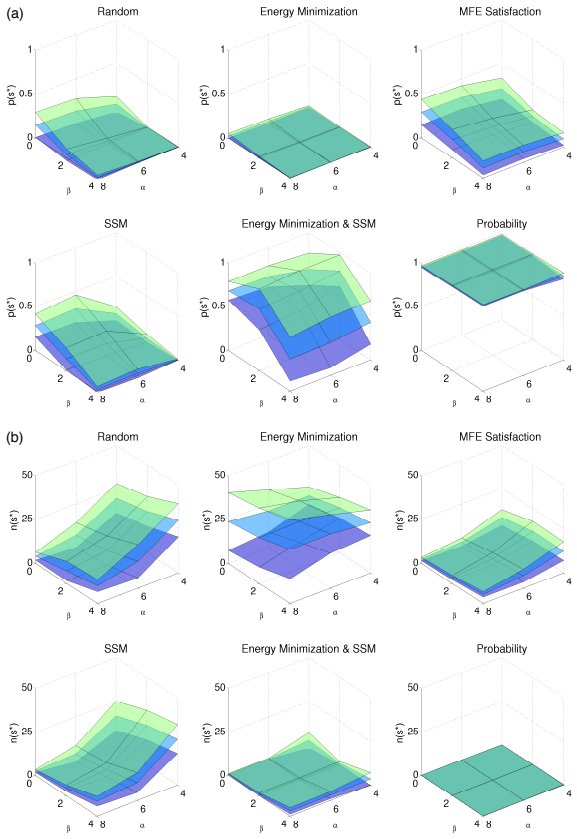


Figure 4.4. RNA Multiloop Variations: Design performance based on (a) $p(s^*)$ and (b) $n(s^*)$ with stem $\alpha = (4, 6, 8)$ and single stranded multiloop regions $\beta = (0, 2, 4)$. Surfaces show the mean values plus and minus one standard deviation for 100 independently designed sequences. The results for optimizing average incorrect nucleotides (not shown) are nearly indistinguishable from those obtained by optimizing probability.

4.7 Discussion

Relative Merits of Design Criteria. Based on thermodynamic considerations, our results support classifying design criteria according to the extent to which they implement positive (affinity) and negative (specificity) design paradigms. The design methods that implement both paradigms (energy minimization plus SSM, probability, average incorrect nucleotides) significantly outperform other methods. In general, the worst performance was observed for methods that implemented neither paradigm (random) or positive design alone (energy minimization), with somewhat better performance observed for negative design methods (MFE satisfaction, SSM). It is perhaps surprising that MFE satisfaction, which performs negative design using the full thermodynamic energy model, performs so similarly to SSM, which neglects the model. Methods based on SSM are widely

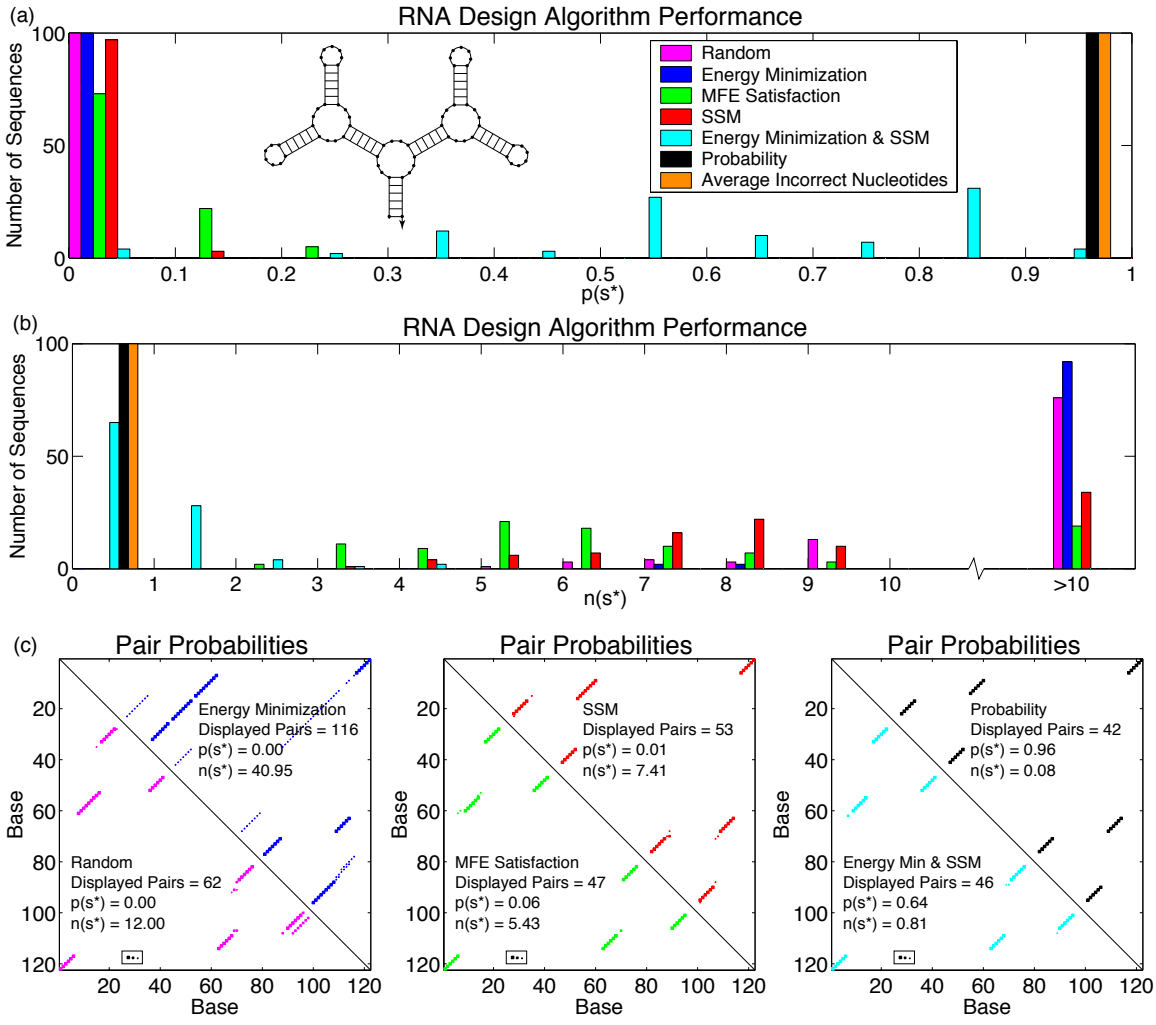


Figure 4.5. Large RNA multiloop. See captions for figures 4.2abc.

used—our results suggest that they could be improved by incorporating a positive design component. For many structures, high probabilities (within the context of an approximate physical model) are obtained by directly optimizing $p(s^*)$ or $n(s^*)$.

Optimization of equilibrium properties leads to sequences with widely differing folding times. One simple design approach is to filter sequences to identify fast or slow folders as desired. Alternatively, new sequence selection algorithms could be developed that explicitly take into account the structure of the energy landscape to optimize the kinetic accessibility of the global minimum energy secondary structure. Furthermore, the observed decoupling of thermodynamic and kinetic properties suggests that there are sufficient degrees of freedom in sequence space to allow the design of more complex

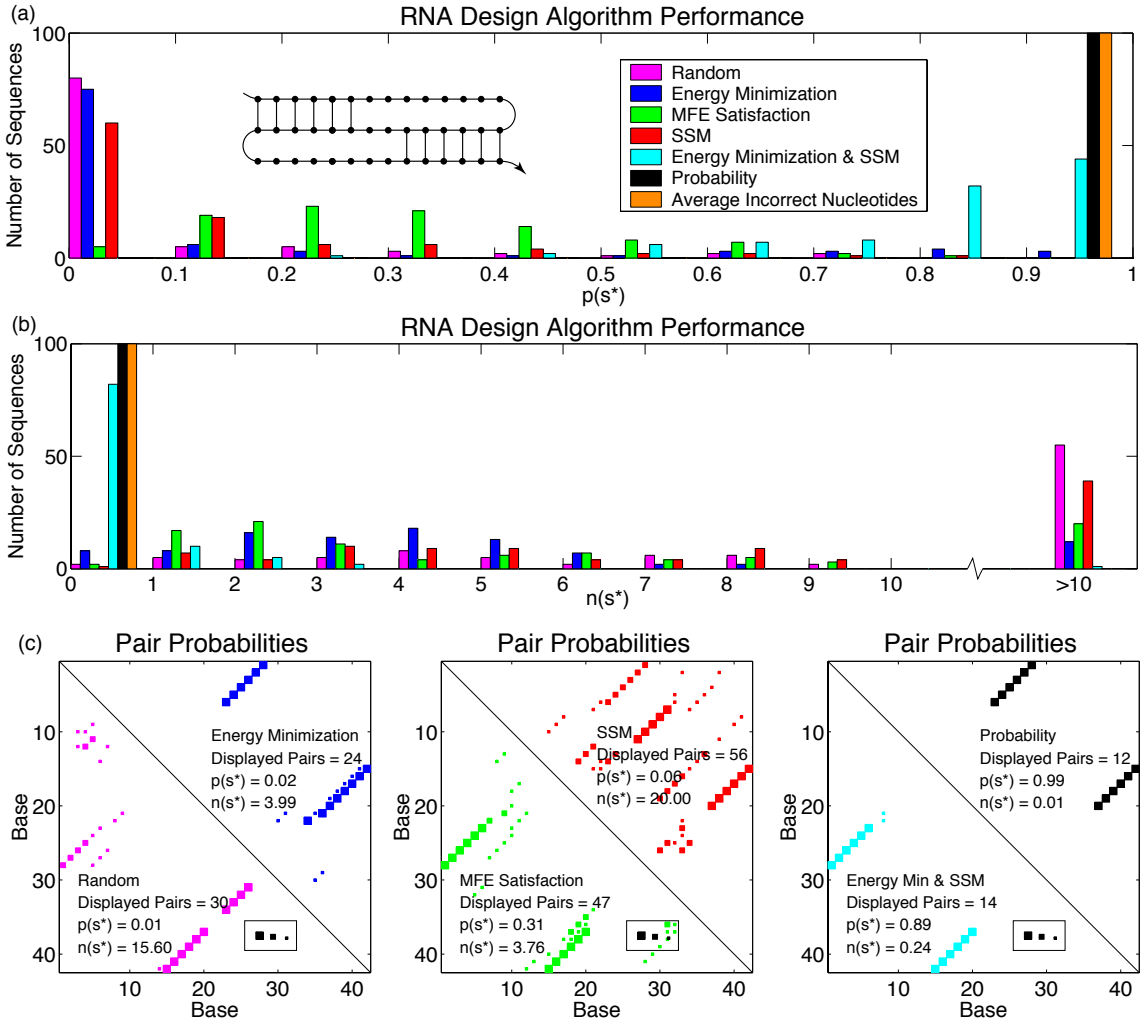


Figure 4.6. RNA Pseudoknot. See captions for figures 4.2abc.

features of the energy landscape (e.g. metastable states [12]).

Robustness of Claims. The consistency in the relative merits of these design methods suggests a level of generality that goes beyond the structures investigated here. It appears that it is not necessary to classify target structures according to the demands that they place on positive or negative design, as methods that implement both paradigms are generally preferred. Furthermore, we observe the same relative performance rankings for RNA and DNA despite systematically different thermodynamic parameters for the two materials. Evaluations of sequence quality for either material appear robust to perturbations in the parameter sets. This suggests that the relative merits of the design criteria are not likely to change as the empirical models are improved.

The validity of our thermodynamic metrics is linked to the validity of the underlying empirical models, which continue to be refined and evaluated by experimental studies [35, 29]. Further improvement of these models for both thermodynamic and kinetic predictive capability will directly benefit rational design methods. Historically, some parameters have experienced adjustments significantly larger than 10% as the model was refined [35]. It seems likely that parameters that have undergone extensive study (e.g. base-pair stacking) will experience relatively small changes in the future, while other parameters that have not received the same degree of scrutiny (e.g. coaxial stacking or dangling ends) may change more dramatically. These adjustments could alter the design conclusions for some target structures.

The partition function may retain utility for design in certain cases where the energy model is known to be incorrect. For example, pseudoknot energy models do not fully consider geometric constraints, such as steric hindrance. Nevertheless, it is reasonable to believe that the unknown energy correction terms are non-negative; that is, structures violating geometric constraints are in fact less likely than predicted. In this case, for design targets that are geometrically unstrained (so that the missing energy term is small), the predicted $p(s^*)$ will be strictly lower than if the energy correction terms had been included (since all undesired structures have non-negative correction terms). Consequently, high-ranking sequences based on existing models are likely to be successful in practice.

Algorithmic Considerations. Each design method consists of a criterion score and a heuristic for optimizing that score. Evaluating the score for a single sequence of length N is an $O(N)$ operation for random, energy minimization, SSM, and energy minimization plus SSM methods. When used in an adaptive walk, each incremental change to the score can be evaluated in constant time. For designs based on MFE satisfaction, probability, and average incorrect nucleotides, each score evaluation is an $O(N^3)$ operation if pseudoknots are excluded and an $O(N^5)$ operation if a class of pseudoknots is included. The cost of these latter methods motivates further investigation of optimization techniques for these scores, including filtering the designs of less expensive methods [38] and assembling larger structures hierarchically [18, 2].

The random and SSM methods apply to single or multi-stranded structures with or without pseudoknots. The other five methods require extensions to the standard empirical potential functions to handle multiple strands or pseudoknots.[5] The dynamic programming algorithms that underly three of the methods also require generalization to handle problems with multiple strands [28, 2, 26].

Additional Design Constraints. Each design method considered here has been simplified to reflect the essence of the approach so as to admit easy description, comparison and replication. In practice, there are many additional considerations that might be used to modify these approaches to satisfy various additional constraints. For example, the designer may wish to limit the CG content, to use a three-letter alphabet [31, 4], to prohibit consecutive stretches of a single base, to fix the melting temperature, or to impose various other rules of thumb that have been garnered from years of lab experience. The intended function of the design may also impose additional requirements, such as the inclusion of subsequences of biological or biochemical relevance (e.g. promoters, restriction sites, genomic targets, ribozymes and deoxyribozymes). Frequently, the intention is to design a set of strands that interact to form one of several allowed secondary structures (e.g. a DNA beacon switches from a hairpin to a helix in the presence of a target ligand). In DNA computing, it is often necessary to design a combinatorial library of strands, each of which is devoid of secondary structure [5]. These problems naturally lead to multi-objective optimizations, where we expect positive and negative design paradigms to continue to play a critical role.

Comparison with Protein Design. It is informative to compare rational nucleic acid design efforts to those in the related area of rational protein design. Proteins provide a rich design space with a much greater demonstrated range of natural function than RNA and DNA. Hence, they represent a fertile medium for the design of new medical and industrial products. While fold affinity and specificity remain fundamental design objectives for proteins, it is not clear to what degree the explicit implementation of both positive and negative design paradigms remains critical. It is possible that the biochemical properties of the twenty amino acids are sufficiently different from those of the four nucleotides that there is a change in the degree to which positive and negative

[5]This complication can be avoided for the two methods involving energy minimization if only stacking energies are considered and other loop terms (which are largely independent of sequence) are neglected.

design methods yield collateral specificity and affinity, respectively.

Computational models for protein thermodynamics currently require three-dimensional fold information. To stabilize a given target fold, rational design efforts have focused on positive design for fold affinity: identify the sequence with the lowest energy on the target fold [10, 8, 23]. Explicit negative design for fold specificity is problematic since it is challenging to describe the space of unwanted three-dimensional folds. However, small ensembles of unwanted structures have been used to explicitly design for fold specificity [16, 19]. Arguments based on the random energy model suggest that implicit negative design may be achieved by fixing the sequence composition prior to optimization [39, 22, 27]. Recently, a novel protein fold was designed from scratch [24] by alternately optimizing the sequence on a fixed backbone and the backbone for a fixed sequence. The former step represents positive design via energy minimization. The latter step was implemented by searching nearby structure space and redefining the target to be the minimum energy structure – a local form of negative design. Conceptually, it is unclear to what extent this local structural optimization implements global negative design.[6]

There is currently no physical abstraction (akin to nucleic acid secondary structure) that facilitates the prediction of protein structure from protein sequence. Hence, the feedback loop in figure 4.1a must be closed either by experimental structural characterization methods or by computationally solving the protein structure prediction problem [32]. This feedback can be used both to improve particular sequence designs and to improve the physical model on which the design process is based. To avoid the risk of introducing artifacts into the physical model [44], significant effort has been invested in developing exact search methods to find the globally optimal sequence based on fold affinity, though approximate search methods have also proved useful in practice [9, 23].

These limitations would similarly apply to nucleic acid design based on three-dimensional atomic coordinates. However, by designing at the level of secondary structure, it is possible to explicitly address both positive and negative design paradigms and to use partition function algorithms to evaluate design quality computationally. The probability of sampling the target graph $p(s^*)$ has a

[6]The related hypothetical approach of performing global structure prediction and adjusting the target to be the minimum free energy structure would correspond to explicit negative design (identical to MFE satisfaction except that specificity is achieved by adjusting the structure instead of the sequence).

maximum value of unity. Hence, it is no longer necessary to perform exact global sequence searches in order to be sure that the sequence accurately reflects the properties of the physical model – it is enough to check that $p(s^*)$ is near unity or that $n(s^*)$ is sufficiently small.

Implications for Design of Nanodevices. For many design applications, nucleic acids represent an attractive building material. Consider for example, an attempt to design a mechanical device that performs work by moving through a series of conformations. It would be cumbersome to parameterize the protein design problem for mechanical devices in terms of atomic coordinates. It also seems unlikely that it would be possible to conditionally stabilize a sequence of non-natural folds using positive design methods that do not explicitly treat fold specificity. However, DNA devices with moving parts and complex conditional conformational changes have already been designed (using *ad hoc* methods) and experimentally demonstrated [47, 43, 42]. We expect that nucleic acid secondary structure will provide a productive framework for formulating the design problem for functional multi-state machines in a way that simultaneously addresses positive and negative design requirements. Ultimately, the objective of rational nucleic acid design efforts is to develop a 'molecular compiler' that takes as input a conceptual design for a device and produces as output, a list of nucleic acid sequences that can be expected to assemble into the desired structures and function robustly.

4.8 Materials and Methods

4.8.1 Design Implementation Details

The parameter sets for RNA and DNA are taken from Mfold3.1 [29] with RNA pseudoknot parameters provided by reference [11]. There are currently no pseudoknot parameters for DNA. Dangle energies were treated as the d2 option in the Vienna package [18]. After each sequence search is performed with any of the methods described below, we check to see whether the sequence is quenched, in the sense that no mutation of a single base pair or of a single unpaired base improves the design according to the design metric. If the sequence is not quenched, we run a further adaptive

walk, checking every 1000 steps to see if the sequence is quenched and terminating the search when quenching is achieved.

Random. 100 random sequences are independently generated that satisfy the target graph base-pairing requirements.

Energy Minimization. 100 independent simulated annealing runs with different random initial sequences are used to identify 100 sequences with a low free energy on the target graph according to the standard loop-based energy model. Each search uses an exponentially decreasing temperature profile over 10^6 steps, where each step corresponds to a point mutation that is accepted if $\exp(-\Delta G^\circ/RT) \geq \rho$, where ΔG° is the change in energy and $\rho \in [0,1]$ is a uniformly distributed random number.

MFE Satisfaction. An adaptive walk of 1000 steps is used to identify a sequence for which the target structure is the lowest energy structure. Each step consists of a random point mutation that is accepted if the new minimum energy structure calculated using dynamic programming methods [49, 18, 33, 1, 11] does not increase the number of mismatches with the target graph [18, 2]. The 100 sequences used for the study are obtained from 100 independent searches starting from different random initial sequences. In each case, the target is the minimum energy structure.

Sequence Symmetry Minization (SSM). 100 sequences are independently selected that are compatible with the target graph and satisfy SSM [36] with word length 4. For the Large Multiloop structure, the word length was increased to 5 to provide a larger vocabulary.

Energy Minimization and SSM. A penalty term is added to the standard energy model to bias the simulated annealing search against sequences that violate SSM. The top-ranked sequences from each of 100 independent searches are free of SSM violations for the cases presented.

Probability. An adaptive walk of 1000 steps is used to search for the sequence with the highest probability of sampling the target structure based on dynamic programing calculations of the partition function [30, 11]. Each step consists of a random point mutation that is rejected if the probability decreases and accepted otherwise [18, 13, 11]. The study uses the top-ranked sequence from each of 100 independent searches starting from different random initial sequences.

Average Number of Incorrect Nucleotides. Independent adaptive walks based on $n(s^*)$ are used to obtain 100 sequences in a manner analogous to the direct optimization of probability described above.

4.8.2 Global Energy Minimization

For methods involving energy minimization, there is no mathematical guarantee that the selected sequences are near a global minimum. For small problems, the performance of heuristic search methods may be assessed by comparison to the global minimum energy obtained using an exact exponential-time branch-and-bound algorithm developed for protein design [15]. If a protein is modeled as a rigid backbone with side-chains represented by discrete *rotamers*, the protein design problem may be formulated as follows [10, 8]: Given p disjoint sets of rotamers R_i (one set for each position i) and a potential function $E(\cdot, \cdot)$ that returns the energy between a pair of rotamers at different positions, choose the rotamer $r_i \in R_i$ at each position that minimizes the sum of the pairwise interactions energies between all positions

$$E_{\text{total}} = \sum_i \sum_{j,j<i} E(r_i, r_j).$$

Methods developed for protein design may be applied to nucleic acid design if the nearest-neighbor empirical potentials [34, 29] are cast as a sum of pairwise terms. For the method based on energy minimization, this is accomplished by constructing overlapping compound 'rotamers' from nearest-neighbor bases and defining infinite energies for neighboring rotamer pairs with inconsistent overlaps. For energy minimization plus SSM, the scope of each rotamer is increased to the SSM word length and infinite energies are assigned to rotamer pairs that violate SSM. Hence, energy minimization results in 3^4 rotamers per position, while energy minimization plus SSM results in W^4 rotamers per position, where W is the SSM word length.

4.8.3 Kinetic Simulation Software

Simulations are performed using Kinfold [12] with Kawasaki rate definitions based on parameter sets provided by the authors.

Bibliography

[1] T. Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104:45–62, 2000.

[2] M. Andronescu, R. Aguirre-Hernandez, A. Condon, and H. H. Hoos. RNAsoft: a suite of RNA secondary structure prediction and design software tools. *Nucleic Acids Research*, 31:3416–3422, 2003.

[3] C. K. Biebricher and R. Luce. In vitro recombination and terminal elongation of RNA by qβ replicase. *EMBO Journal*, 11:5129–5135, 1992.

[4] R. S. Braich, N. Chelyapov, C. Johnson, P. W. K. Rothemund, and L. Adleman. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, 296:499–502, 2002.

[5] A. Brenneman and A. Condon. Strand design for biomolecular computation. *Theoretical Computer Science*, 287:39–58, 2002.

[6] S. Brenner, M. Johnson, J. Bridgham, G. Golda, D. H. Lloyd, D. Johnson, S. Luo, S. McCurdy, M. Foy, M. Ewan, R. Roth, D. George, S. Eletr, G. Albrecht, E. Vermaas, S. R. Williams, K. Moon, T. Burcham, M. Pallas, R. B. DuBridge, J. Kirchner, K. Fearon, J. Mao, and K. Corcoran. Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nature Biotechnology*, 18:630–634, 2000.

[7] J. Chen and N. C. Seeman. The synthesis from DNAs of a molecule with the connectivity of a cube. *Nature*, 350:631–633, 1991.

[8] B. I. Dahiyat and S. L. Mayo. De novo protein design: fully automated sequence selection. *Science*, 278(5335):82–87, 1997.

[9] J. R. Desjarlais and N. D. Clarke. Computer search algorithms in protein modification and design. *Current Opinion in Structural Biology*, 8(4):471–475, 1998.

[10] J. R. Desjarlais and T. M. Handel. De novo design of the hydrophobic cores of proteins. *Protein Science*, 4:2006–2018, 1995.

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

[12] C. Flamm, W. Fontana, I. L. Hofacker, and P. Schuster. RNA folding at elementary step resolution. *RNA*, 6:325–338, 2000.

[13] C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7:254–265, 2001.

[14] D. T. Gillespie. General method for numerically simulating stochastic time evolution of coupled chemical-reactions. *Journal of Computational Physics*, 22(4):403–434, 1976.

[15] D. B. Gordon and S. L. Mayo. Branch-and-terminate: a combinatorial optimization algorithm for protein design. *Structure*, 7:1089–1098, 1999.

[16] J. J. Havranek and P. B. Harbury. Automated design of specificity in molecular recognition. *Nature Structural Biology*, 10(1):45–52, 2003.

[17] P. G. Higgs and S. R. Morgan. Barrier heights between ground states in a model of RNA secondary structure. *Journal of Physics A: Mathematical and General*, 31:3153–3170, 1998.

[18] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Chemical Monthly*, 125:167–188, 1994.

[19] W. Jin, O. Kambara, H. Sasakawa, A. Tamura, and S. Takada. De novo design of foldable proteins with smooth folding funnel: automated negative design and experimental validation. *Structure*, 11:581–590, 2003.

[20] L. Kaderali, A. Deshpande, J. P. Nolan, and P. S. White. Primer-design for multiplexed genotyping. *Nucleic Acids Research*, 31:1796–1802, 2003.

[21] R. K. Kallenbach, R.-I. Ma, and N. C. Seeman. An immobile nucleic acid junction constructed from oligonucleotides. *Nature*, 305:829–831, 1983.

[22] P. Koehl and M. Levitt. De novo protein design. i. in search of stability and specificity. *Journal of Molecular Biology*, 293:1161–1181, 1999.

[23] C. M. Kraemer-Pecore, A. M. Wollacott, and J. R. Desjarlais. Computational protein design. *Current Opinion in Chemical Biology*, 5:690–695, 2001.

[24] B. Kuhlman, G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard, and D. Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302:1364–1368, 2003.

[25] T. H. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J. H. Reif, and N. C. Seeman. Construction, analysis, ligation and self-assembly of DNA triple crossover complexes. *Journal of the American Chemical Society*, 122:1848–1869, 2000.

[26] N. R. Markham. *Algorithms for nucleic acid folding, hybridization, and melting prediction.* Master's thesis, Rensselaer Polytechnic Institute, 2003.

[27] S. A. Marshall and S. L. Mayo. Achieving stability and conformational specificity in designed proteins via binary patterning. *Journal of Molecular Biology*, 305:619–631, 2001.

[28] D. H. Mathews, M. E. Burkard, S. M. Freier, J. R. Wyatt, and D. H. Turner. Predicting oligonucleotide affinity to nucleic acid targets. *RNA*, 5:1458–1469, 1999.

[29] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940, 1999.

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

[31] K. U. Mir. A restricted genetic alphabet for DNA computing. In L. F. Landeweber and E. B. Baum, editors, *DNA Based Computers II*, volume 44 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 243–246. American Mathematical Society, 1996.

[32] J. Moult, F. Krzysztof, A. Zemla, and T. Hubbard. Critical assessment of methods of protein structure prediction (CASP)-round v. *Proteins*, 53:334–339, 2003.

[33] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285:2053–2068, 1999.

[34] J. Santalucia Jr. Improved nearest-neighbor parameters for predicting DNA duplex stability. *Biochemistry*, 35:3555–3562, 1996.

[35] J. Santalucia Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences of the United States of America*, 95(4):1460–1465, 1998.

[36] N. C. Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Bioogy.*, 99:237–247, 1982.

[37] N. C. Seeman. DNA engineering and its application to nanotechnology. *Trends in Biotechnology*, 17:437–443, 1999.

[38] N. C. Seeman and R. K. Kallenbach. Design of immobile nucleic acid junctions. *Biophysical Journal*, 44:201–209, 1983.

[39] E. I. Shakhnovich and A. M. Gutin. Engineering of stable and fast-folding sequences of model proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 90:7195–7199, 1993.

[40] D. D. Shoemaker, D. A. Lashkari, D. Morris, M. Mittman, and R. W. Davis. Quantitative phenotypic analysis of yeast deletion mutants using a highly parallel molecular bar-coding strategy. *Nature Genetics*, 16:450–456, 1996.

[41] G. A. Soukup and R. R. Breaker. Engineering precision RNA molecular switches. *Proceedings of the National Academy of Sciences of the United States of America*, 96:3584–3589, 1999.

[42] M. N. Stojanovic and D. Stefanovic. A deoxyribozyme-based molecular automaton. *Nature Biotechnology*, 21:1069–1074, 2003.

[43] A. J. Turberfield, J. C. Mitchell, B. Yurke, A. P. Mills, Jr., M. I. Blakey, and F. C. Simmel. DNA fuel for free-running nanomachines. *Physical Review Letters*, 90(11):118102, 2003.

[44] C. A. Voigt, D. B. Gordon, and S. L. Mayo. Trading accuracy for speed: a quantitative comparison of search algorithms in protein sequence design. *Journal of Molecular Biology*, 299(3):789–803, 2000.

[45] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

[46] H. Yan, X. Zhang, Z. Shen, and N. C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415(6867):62–5, 2002.

[47] B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, 2000.

[48] W. Zhang and S.-J. Chen. RNA hairpin-folding kinetics. *Proceedings of the National Academy of Sciences of the United States of America*, 99(4):1931–1936, 2002.

[49] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–147, 1981.

Chapter 5

Construction of a DNA-Based Biosensor

The work presented here is heavily based on:

R. M. Dirks, and N. A. Pierce, *Triggered amplification by hybridization chain reaction.* Proceedings of the National Academy of Sciences of the United States of America, 2004. **101**(43): p. 15275-15278. Reprinted with copyright permissions.

5.1 Abstract

Given the previously developed tools for the analysis and design of nucleic acid secondary structures, the challenge is to parlay these techniques into useful DNA systems. To this end, we must first conceive of a system of DNA structures predicted to have experimentally interesting behavior, then design sequences to instantiate this framework. Along these lines, we introduce the concept of hybridization chain reaction (HCR), in which stable DNA monomers assemble only upon exposure to a target DNA fragment. In the simplest version of this process, two stable species of DNA hairpins coexist in solution until the introduction of initiator strands triggers a cascade of hybridization events that yields nicked double helices analogous to alternating copolymers. The average molecular weight of the HCR products varies inversely with initiator concentration. Amplification of more diverse recognition events can be achieved by coupling HCR to aptamer triggers. This functionality allows DNA to act as an amplifying transducer for biosensing applications.

5.2 Introduction

Biosensors require both a recognition component for detection and a transduction component for read-out. In gene chips, recognition is performed by single-stranded DNAs that screen for complementary nucleic acid fragments and transduction is typically performed by optical or electrochemical means [25, 6]. Nucleic acid aptamers obtained by *in vitro* selection methods [8, 34] generalize this recognition capability to a wide range of target analytes [11, 16] and are amenable to optical transduction approaches [14, 22]. Here, we demonstrate that DNA can also play the transduction role via a new amplification approach termed hybridization chain reaction (HCR). This class of mechanisms suggests the possibility of constructing biosensors solely from unmodified single-stranded DNA.

Single-stranded DNA is a versatile construction material [27] that can be programmed [26, 28, 12, 10, 2, 5] to self-assemble into complex structures [26, 4, 18, 37, 15, 38, 19, 20, 30] driven by the free energy of base pair formation. Synthetic DNA machines can be powered by strand displacement interactions initiated by the sequential introduction of auxiliary DNA fuel strands [40, 39, 17, 33, 1,

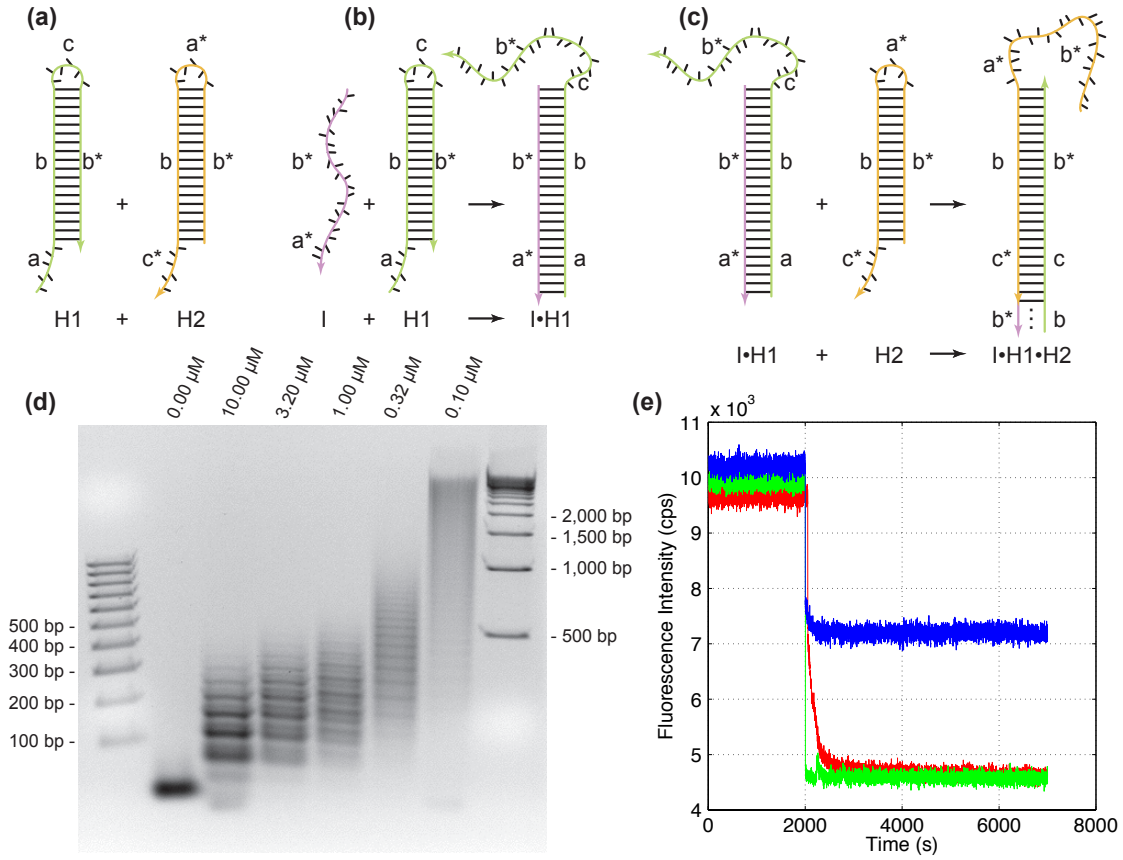


Figure 5.1. Basic HCR system. (a)-(c) Secondary structure schematic of HCR function. Letters marked with * are complementary to the corresponding unmarked letter. (a) Hairpins H1 and H2 are stable in the absence of initiator I. (b) I nucleates at the sticky end of H1 and undergoes an unbiased strand displacement interaction to open the hairpin. (c) The newly exposed sticky end of H1 nucleates at the sticky end of H2 and opens the hairpin to expose a sticky end on H2 that is identical in sequence to I. Hence, each copy of I can propagate a chain reaction of hybridization events between alternating H1 and H2 hairpins to form a nicked double-helix, amplifying the signal of initiator binding. (d) Effect of initiator concentration on HCR amplification. Six different concentrations of initiator (0.00, 10.00, 3.20, 1.00, 0.32, 0.10 μM) in a 1 μM mixture of H1 and H2 (Lanes 2-7). DNA markers with 100-bp and 500-bp increments, respectively (Lanes 1 and 8). (e) HCR kinetics. The hairpin monomers do not hybridize prior to triggering by initiator ((H1^{2AP} + 1.2x H2) + 0.5x I, red). The same quenched baseline is achieved without HCR by adding excess initiator to H1^{2AP} in the absence of H2 (H1^{2AP} + 4.0x I, green). Addition of insufficient initiator to H1^{2AP} provides only partial quenching (H1^{2AP} + 0.5x I, blue).

31, 29, 32]. Typically, various DNA strands begin to associate as soon as they are mixed together. Catalytic fuel delivery provides a conceptual approach to powering autonomous DNA machines by storing potential energy in loops that are difficult to access kinetically except in the presence of a catalyst strand [35]. In the system described here, monomer DNA building blocks are mixed together but do not hybridize on an experimental time scale. Exposure of an initiator strand triggers a chain reaction of hybridization events similar to living chain polymerization but without covalent bond formation. This system introduces the concept of triggered self-assembly of DNA nanostructures.

5.3 Methods

5.3.1 System Specifications

DNA sequences were designed using a combination of criteria [5]: sequence symmetry minimization [26], probability of adopting the target secondary structure at equilibrium [12], average number of incorrect nucleotides at equilibrium relative to the target structure [5] and hybridization kinetics [9]. The sequences for the basic HCR system of figure 1 and the aptamer HCR system of figure 2 are shown in table 1. The aptamer system required new sequence designs to ensure compatibility with the fixed sequence of the aptamer. For the kinetic studies of figure 1c, $H1^{2AP}$ is used in place of H1 with the third base (A) replaced with 2-aminopurine [23].

DNA was synthesized and purified by Integrated DNA Technologies. For the basic HCR system of figure 1, concentrated DNA stock solutions were prepared in buffer that was later diluted to reaction conditions: 50 mM Na_2HPO_4, 0.5 M NaCl, pH 6.8. For the ATP aptamer HCR system of figure 2, concentrated stock solutions were later diluted to reaction conditions: 5 mM $MgCl_2$, 0.3 M NaCl, 20 mM Tris, pH 7.6.

5.3.2 Native Gel Electrophoresis

Samples were heated to 95° C for 2 minutes and then allowed to cool to room temperature for 1 hour prior to use. The 1% agarose gels of Figs 1d and 2b contained 0.5 μg ethidium bromide per

mL of gel volume and were prepared using 1x SB buffer [3]. Agarose gels were run at 150V for 60 minutes and visualized under UV light. The native polyacrylamide gel of figure 2c was a 10% precast gel made with 1x TBE buffer. The gel was run at 150V for 40 minutes in 1x TBE, stained 30 minutes in a solution containing 0.5 μg ethidium bromide per mL, and then viewed under UV light. For the reactions of figure 1d, stock solutions of I, H2 and H1 were diluted in reaction buffer to 3x their final concentrations (see legend) and 9 μL of each species were combined in that order (27 μL reaction volume). For figure 2b, DNA species were combined to yield 1 μM concentrations in 27 μL of reaction buffer, with additions made in the order: buffer and/or (I or I^{ATP}), H1 and then H2 (note that I and I^{ATP} interact with H2 rather than H1). In this case,1 μL of 40 mM ATP, 40 mM GTP or water was added to each reaction, as appropriate, for a total reaction volume of 28 μL. Reactions were incubated at room temperature for 24 hours before running 24 μL of each product on a gel. Reactions for the polyacrylamide gel of figure 2c were performed at half volume and the entire reaction volumes were loaded on the gel.

5.3.3 Fluorescence Kinetics.

Fluorescence data were obtained using a fluorometer from Photon Technologies International, with the temperature controller set to 22° C. Excitation and emission wavelengths [23] were 303nm and 365nm, respectively, with 4 nm bandwidths. Stock solutions of 0.40 μM $H1^{2AP}$ and 0.48 μM H2 were prepared in reaction buffer, heated to 90° C for 90 seconds and then allowed to cool to room temperature for 1 hour prior to use. For each experiment, 250 μL of $H1^{2AP}$ were added to either 250 μL of H2 or 250μL of reaction buffer. These 0.20 μM $H1^{2AP}$ solutions were allowed to sit at room temperature for at least 24 hours before taking fluorescence measurements. The initial signal was obtained after rapidly pipetting the sample in the cuvette to obtain a stable fluorescence baseline. After acquiring at least 2000 seconds of this baseline, runs were paused for approximately one minute to add 20 μL of initiator (either 20μM or 2.5μM) and allow mixing by rapid pipetting. The final reaction volume was 520 μL for all experiments.

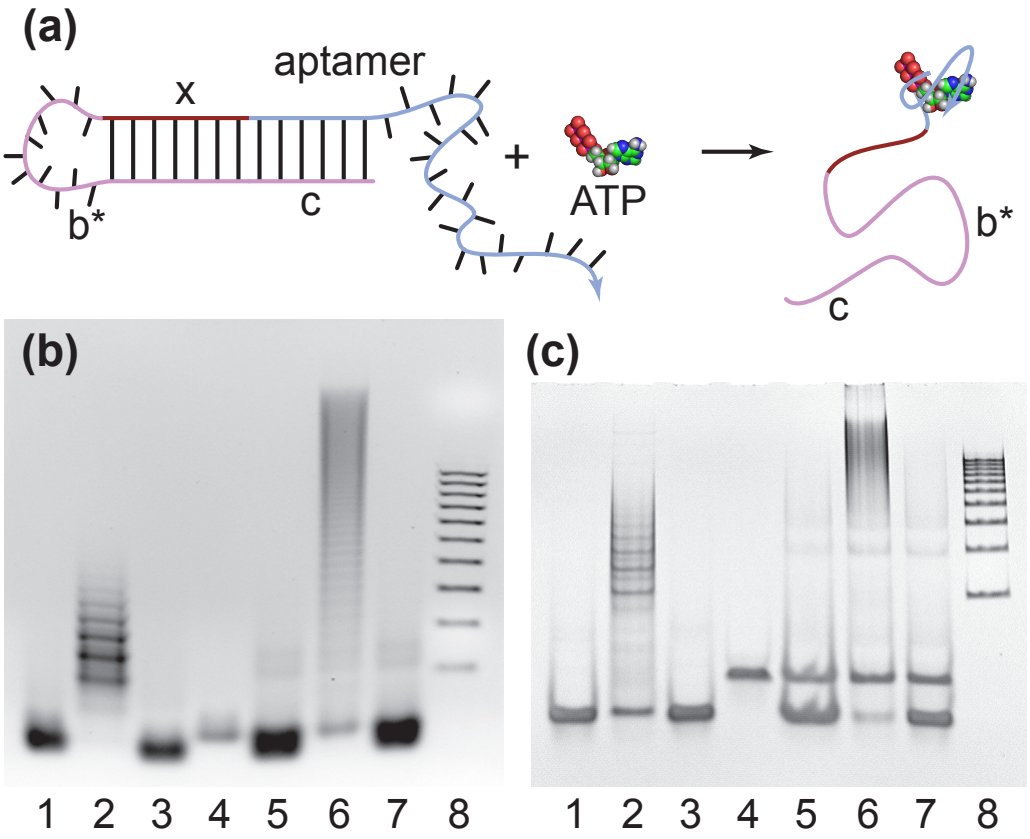


Figure 5.2. Aptamer HCR system. (a) Aptamer trigger mechanism. Binding of the DNA aptamer (blue) to ATP [13] exposes a sticky end [22] (magenta) that triggers the HCR mechanism of figure 1 by opening hairpin H2. The region x (red) is introduced to help stabilize the trigger in the absence of analyte [22]. The region b* includes both the hairpin loop and the portion of the stem complementary to x. b-c, ATP detection via HCR. (b) Agarose and (c) acrylamide gels demonstrate amplification of ATP recognition, with the former providing better resolution of HCR products and the latter providing better resolution of unreacted species. Reactions are performed with 1.4mM ATP and GTP and all DNA species at 1μM. The hairpins do not polymerize when mixed (H1 + H2, Lane 1). Addition of simple initiator triggers HCR (H1 + H2 + I, Lane 2). Hairpins with ATP (H1 + H2 + ATP, Lane 3). Aptamer initiator with ATP (I^{ATP} + ATP, Lane 4). Weak spurious HCR in the absence of ATP (H1 + H2 + I^{ATP} , Lane 5) or the presence of GTP (H1 + H2 + I^{ATP} + GTP, Lane 7). Strong HCR amplification of ATP recognition (H1 + H2 + I^{ATP} + ATP, Lane 6). DNA ladder (100-1000 bp in 100 bp increments, Lane 8).

Table 5.1. HCR systems

System	Strand	Sequence*
Basic	H1	5′-TTAACCCACGCCGAATCCTAGACTCAAAGTAGTCTAGGATTCGGCGTG-3′
	H2	5′-AGTCTAGGATTCGGCGTGGGTTAACACGCCGAATCCTAGACTACTTTG-3′
	I	5′-AGTCTAGGATTCGGCGTGGGTTAA-3′
Aptamer†	H1	5′-CATCTCGGTTTGGCTTTCTTGTTACCCAGGTAACAAGAAAGCCAAACC-3′
	H2	5′-TAACAAGAAAGCCAAACCGAGATGGGTTTGGCTTTCTTGTTACCTGGG-3′
	I^{ATP}	5′-CCCAGGTAACAAGAAAGCCAAACCTCTTGTT*ACCTGGGGGAGTATTGCGGAGGAAGGT*-3′
	I	5′-CCCAGGTAACAAGAAAGCCAAACC-3′

* In the hairpin sequences, loops are underlined and sticky ends are overlined. †Aptamer nucleotides [22] are italicized.

5.4 Results

The simplest HCR mechanism employs two hairpin species (H1 and H2 in figure 1abc). The key to this system is the storage of potential energy in short loops protected by long stems. This situation contrasts with that for molecular beacons [36], where short stems protect long loops to allow the target nucleotide to bind in the loop and open the beacon. In the present HCR system, each hairpin is caught in a kinetic trap, preventing the system from rapidly equilibrating. Introduction of an initiator strand (I) triggers a chain reaction of alternating kinetic escapes by the two hairpin species corresponding to 'polymerization' into a nicked double helix. Amplification of the initiator recognition event continues until the supply of H1 or H2 is exhausted. The average molecular weight of the resulting polymers is inversely related to the initiator concentration (figure 1d), suggesting the potential for quantitative sensing. This inverse relationship follows from the fixed supply of monomer hairpins, but the phenomenon can be observed after 10 minutes, when the supply has not yet been exhausted.

The kinetics of HCR can be explored using fluorescence quenching. 2-aminopurine (2AP), an analog of adenine, fluoresces when single-stranded, but is significantly quenched when in a stacked double-helical conformation [23]. Monomer usage can be monitored as polymerization occurs by replacing H1 with the labelled hairpin $H1^{2AP}$ (obtained by substituting 2AP for an A in the sticky end of H1). Monitoring 2AP fluorescence is preferable to using standard end-labeled strands because the local environment of quenched 2AP will be the same regardless of whether I or H2 performs the quenching. In contrast, dyes tethered to the ends of strands may have different fluorescent properties based on their position (terminal or internal) in the HCR polymer.

The hairpin monomers $H1^{2AP}$ and H2 do not hybridize in the absence of initiator (figure 1e). Addition of I to the hairpin mixture at a lower concentration leads to fluorescence quenching via HCR. The same quenched baseline is achieved without HCR by combining $H1^{2AP}$ with excess I. In this case, each I molecule causes one fluorescent signalling event by binding to $H1^{2AP}$; with H2 present, HCR performs fluorescent amplification, allowing each I molecule to alter the fluorescence of multiple hairpins. Addition of insufficient initiator to $H1^{2AP}$ (at the same concentration as for the first experiment) provides only partial quenching, demonstrating that HCR, and not I alone, is responsible for exhausting the supply of $H1^{2AP}$ monomer. The variation in initial fluorescence intensities is approximately 10% across the three experiments. The off/on kinetic behavior of this HCR system suggests that these hairpin constructs may also be useful for delivering fuel to autonomous DNA machines.

More diverse biosensors can be created by triggering HCR with molecular recognition events based on DNA or RNA aptamers [8, 34]. Figure 2a depicts a scheme for HCR amplification of ATP detection using an aptamer construct [13, 22] that exposes an initiator strand upon binding ATP. Figs 2b-c demonstrate successful detection of ATP, as well as specificity in differentiating ATP from GTP.

5.5 Discussion

These data suggest that HCR can be described by a coarse-grained, structure/function relationship in which secondary structure (as opposed to tertiary atomic coordinates) is sufficient to determine the qualitative interactions of components in the system. This property represents a significant advantage in using nucleic acids as a construction material [27]. The basic HCR system of figure 1 and the aptamer HCR system of figure 2 have identical hairpin secondary structures, with stems of length 18 and loops of length 6 (sticky ends match loop lengths); both sets of hairpins are stable as monomers when mixed, but undergo HCR in response to triggering. The sequence identity between analogous hairpins in these systems is only 30%, corresponding to 9 of 30 independent positions (25% identity would be expected for random sequences).

A series of experiments on hairpins with different combinations of loop and stem lengths gave widely different results (data not shown). For example, systems with stems of length 18 and loops of length 8 or with stems of length 10 and loops of length 6 were not stable at room temperature (monomers polymerized overnight in the absence of initiator). If the loops were too small (e.g. of length 4 with stems of length 10), polymerization in the presence of initiator occurred very slowly, if at all. While sequence and tertiary structure are central to aptamer function [13], secondary structure appears to be the appropriate level of description to represent and design the core functionality of HCR. Further experimental studies of this structure/function relationship are warranted.

The concept of HCR is not limited to polymerization of monomer hairpins. The HCR mechanism depicted in figure 1 represents linear growth in response to initiator. It is possible to envision more complicated sets of monomers that would undergo triggered self-assembly to create branched structures corresponding to quadratic growth or even dendritic systems exhibiting exponential growth. Nonlinear amplifiers of this type are currently under investigation.

Here, HCR detection was performed by gel electrophoresis or fluorescence quenching. Using inexpensive synthesized DNA components and standard gel electrophoresis equipment, HCR schemes could potentially serve as economical and easily adopted amplifiers for molecular detection. Alternatively, it may be possible to develop a nano-gold based colorimetric assay [7, 21] for HCR that could be used for read-out in non-laboratory settings.

HCR amplification could be applied to a wide range of sensing applications by developing a general approach to aptamer triggering. Furthermore, the trigger need not be based on molecular recognition; any physical process that exposes initiator strand will suffice. It may sometimes be useful to employ HCR for both amplification and capture, using the resulting DNA polymers to remove the analyte from solution. For some applications in which detection is the primary objective of amplification, HCR may have the potential to become an attractive protein-free, room temperature alternative to polymerase chain reaction (PCR) [24].

Bibliography

[1] P. Alberti and J.-L. Mergny. DNA duplex-quadruplex exhange as the basis for a nanomolecular machine. *Proceedings of the National Academy of Sciences of the United States of America*, 100:1569–1573, 2003.

[2] M. Andronescu, A. P. Fejes, F. Hutter, H. H. Hoos, and A. Condon. A new algorithm for RNA secondary structure design. *Journal of Molecular Biology*, 336(3):607–624, 2004.

[3] J. R. Brody and S. E. Kern. Sodium boric acid: a tris-free, cooler conductive medium for DNA electrophoresis. *BioTechniques*, 36(2):214–215, 2004.

[4] J. Chen and N. C. Seeman. The synthesis from DNAs of a molecule with the connectivity of a cube. *Nature*, 350:631–633, 1991.

[5] R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–1403, 2004.

[6] T. G. Drummond, M. G. Hill, and J. K. Barton. Electrochemical DNA sensors. *Nature Biotechnology*, 21(10):1192–1199, 2003.

[7] R. Elghanian, J. J. Storhoff, R. C. Mucic, R. L. Letsinger, and C. A. Mirkin. Selective colorimetric detection of polynucleotides based on the distance-dependent optical properties of gold nanoparticles. *Science*, 277(5329):1078–1081, 1997.

[8] A. D. Ellington and J. W. Szostak. In vitro selection of RNA molecules that bind specific ligands. *Nature*, 346:818–822, 1990.

[9] C. Flamm, W. Fontana, I. L. Hofacker, and P. Schuster. RNA folding at elementary step resolution. *RNA*, 6:325–338, 2000.

[10] C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7:254–265, 2001.

[11] T. Hermann and D. J. Patel. Adaptive recognition by nucleic acid aptamers. *Science*, 287:820–825, 2000.

[12] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Chemical Monthly*, 125:167–188, 1994.

[13] D. E. Huizenga and J. W. Szostak. ADNA aptamer that binds adenosine and ATP. *Biochemistry*, 34:656–665, 1995.

[14] S. Jhaveri, M. Rajendran, and A. D. Ellington. In vitro selection of signaling aptamers. *Nature Biotechnology*, 18:1293–1297, 2000.

[15] T. H. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J. H. Reif, and N. C. Seeman. Construction, analysis, ligation and self-assembly of DNA triple crossover complexes. *Journal of the American Chemical Society*, 122:1848–1869, 2000.

[16] J. F. Lee, J. R. Hesselberth, L. A. Meyers, and A. D. Ellington. Aptamer database. *Nucleic Acids Research*, 32:D95–D100, 2004.

[17] J. J. Li and W. Tan. A single DNA molecule nanomotor. *Nano Letters*, 2(4):315–318, 2002.

[18] X. Li, X. Yang, J. Qi, and N. C. Seeman. Antiparallel DNA double crossover molecules as components for nanoconstruction. *Journal of the American Chemical Society*, 118:6131–6140, 1996.

[19] D. Liu, S. H. Park, J. H. Reif, and T. H. LaBean. DNA nanotubes self-assembled from triple-crossover tiles as templates for conductive nanowires. *Proceedings of the National Academy of Sciences of the United States of America*, 101(3):717–722, 2004.

[20] D. Liu, M. Wang, Z. Deng, R. Walulu, and C. Mao. Tensegrity: Construction of rigid DNA triangles with flexible four-arm DNA junctions. *Journal of the American Chemical Society*, 126:2324–2325, 2004.

[21] J. W. Liu and Y. Lu. A colorimetric lead biosensor using DNAzyme-directed assembly of gold nanoparticles. *Journal of the American Chemical Society*, 125(22):6642–6643, 2003.

[22] R. Nutiu and Y. Li. Structure-switching signaling aptamers. *Journal of the American Chemical Society*, 125:4771–4778, 2003.

[23] E. L. Rachofsky, R. Osman, and J. B. A. Ross. Probing structure and dynamics of DNA with 2-aminopurine: effects of local environment on fluorescence. *Biochemistry*, 40:946–956, 2001.

[24] R. K. Saiki, D. H. Gelfand, S. Stoffel, S. J. Scharf, R. Higuchi, G. T. Horn, K. B. Mullis, and H. A. Erlich. Primer-directed enzymatic amplification of DNA with a thermostable DNA-polymerase. *Science*, 239(4839):487–491, 1988.

[25] F. W. Scheller, U. Wollenberger, A. Warsinke, and F. Lisdat. Research and development in biosensors. *Current Opinion in Biotechnology*, 12:35–40, 2001.

[26] N. C. Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Bioogy.*, 99:237–247, 1982.

[27] N. C. Seeman. DNA in a material world. *Nature*, 421:427–431, 2003.

[28] N. C. Seeman and R. K. Kallenbach. Design of immobile nucleic acid junctions. *Biophysical Journal*, 44:201–209, 1983.

[29] W. B. Sherman and N. C. Seeman. A precisely controlled DNA biped walking device. *Nano Letters*, 4(7):1203–1207, 2004.

[30] W. M. Shih, J. D. Quispe, and G. F. Joyce. A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature*, 427:618–621, 2004.

[31] J.-S. Shin and N. A. Pierce. Rewritable memory by controllable nanopatterning of DNA. *Nano Letters*, 4(5):905–909, 2004.

[32] J.-S. Shin and N. A. Pierce. A synthetic DNA walker for molecular transport. *Journal of the American Chemical Society*, 126:10834–10835, 2004.

[33] F. C. Simmel and B. Yurke. A DNA-based molecular device switchable between three distinct mechanical states. *Applied Physics Letters*, 80(5):883–885, 2002.

[34] C. Tuerk and L. Gold. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage t4 DNA polymerase. *Science*, 249(4968):505–510, 1990.

[35] A. J. Turberfield, J. C. Mitchell, B. Yurke, A. P. Mills, Jr., M. I. Blakey, and F. C. Simmel. DNA fuel for free-running nanomachines. *Physical Review Letters*, 90(11):118102, 2003.

[36] S. Tyagi and F. R. Kramer. Molecular beacons: probes that fluoresce upon hybridization. *Nature Biotechnology*, 14(3):303–308, 1996.

[37] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

[38] H. Yan, T. H. LaBean, L. Feng, and J. H. Reif. Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices. *Proceedings of the National Academy of Sciences of the United States of America*, 100(14):8103–8108, 2003.

[39] H. Yan, X. Zhang, Z. Shen, and N. C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415(6867):62–5, 2002.

[40] B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, 2000.

Appendices

The following appendices contain supplementary information pertaining to Robert Dirks's thesis. Appendix A contains a general overview of nucleic acid structure and function. Appendices B and C contain additional figures and pseudocode for the chapters on analysis. Appendix D discusses the thermodynamics of multi-strand complexes, and Appendix E gives more examples of nucleic acid design. Appendix F describes more sophisticated versions of HCR.

Appendix A

Background on Nucleic Acids

A.1 Composition and Structure of Nucleic Acids

Nucleic acids are versatile biopolymers essential to all life. Composed of directed, negatively charged sugar-phosphate backbones and side groups called bases, nucleic acids come in two major forms: deoxyribonucleic acids (DNA) and ribonucleic acids (RNA). For naturally occurring DNA, the predominant four bases are adenine (A), cytosine (C), guanine (G) and thymine (T), while RNA replaces thymine with uracil (U). Strands of DNA and RNA are typically described by their primary structure, a sequence of bases, listed in the 5' to 3' direction (Figure A.1a).

An important feature of nucleic acids is the ability to form stabilizing contacts (see Figure A.1b) between complementary bases (C pairs with G, and A with T or U) when arranged in an anti-parallel direction. If several of these Watson-Crick base pairs are aligned consecutively, the hydrogen bonding between complementary bases, along with the stacking interactions between the adjacent aromatic rings can contribute about -1.0 to -3.5 kcal/mol per stack[1], depending on the sequence. These base pairs lead to higher order configurations known as secondary structure, and are the main reason why nucleic acids are so useful for both biological and synthetic purposes.

When two complementary strands of DNA hybridize, base pairing and stacking can be maximized by adopting a B-DNA structure (Figure A.1c), a right-handed double helix with a diameter of about 20Å, and a height of 34Å per turn (roughly 10 base pairs). These double helices tend to be rigid on length scales of roughly 50 nm (150 base pairs), but are flexible over longer distances. To give

[1] At 37° C and 1M NaCl

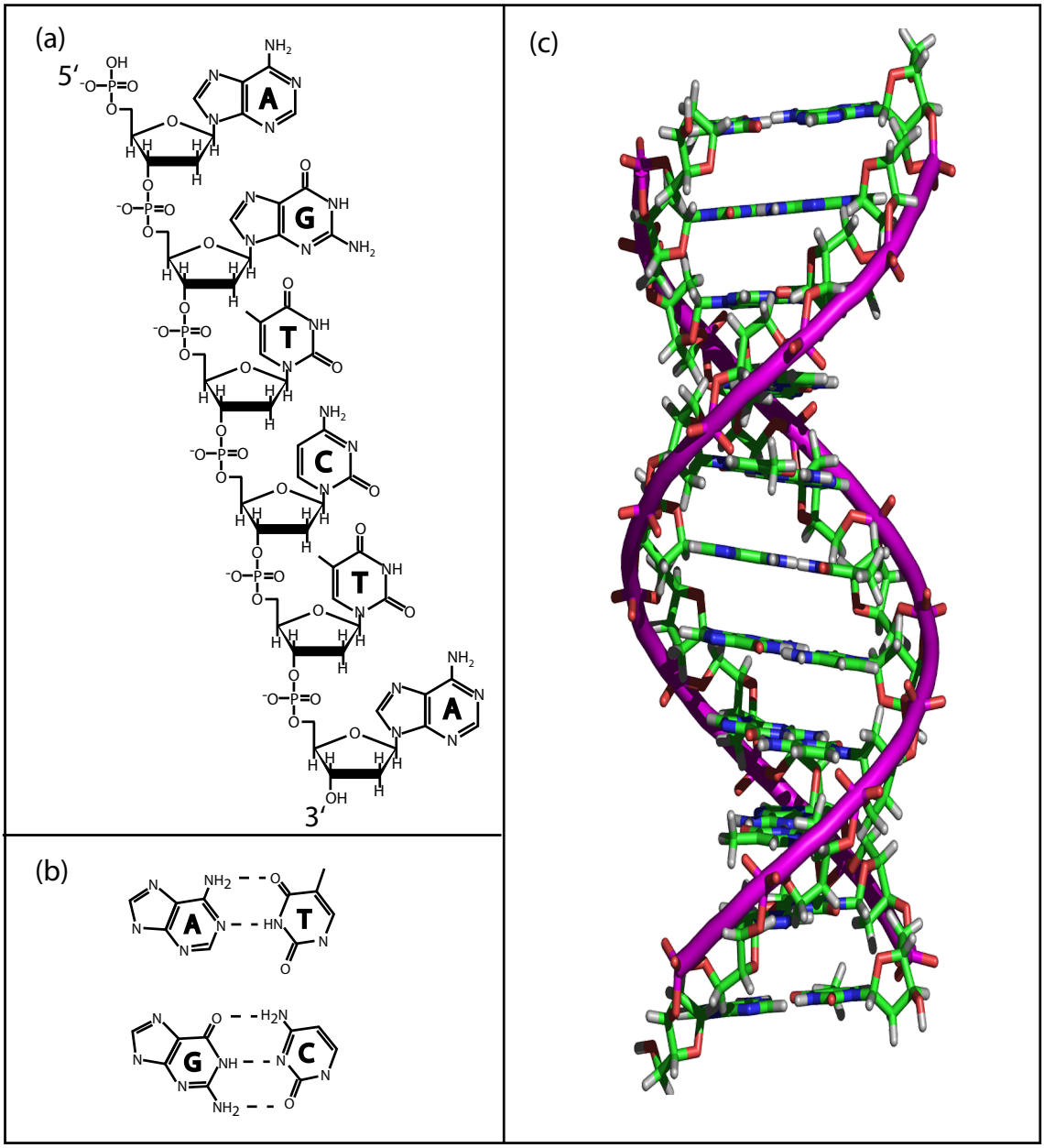


Figure A.1. **(a)** A single strand of DNA showing the sugar phosphate backbone and attached bases. This DNA molecule can be described by its sequence, `AGTCTA`. **(b)** The canonical Watson-Crick base pairs. The dotted lines indicate hydrogen bonds. **(c)** B-DNA. Two complementary strands of DNA can form a double helix, with the two purple tubes added to accentuate the sugar-phosphate backbone. The bases are perpendicular to the helical axis and stabilize the structure thorugh hydrogen bonding and base stacking.

a sense of typical size, humans have roughly three billion base pairs of DNA in the genome, with the average DNA molecule roughly 4 to 5 cm in length [24]. In contrast, synthetic DNA structures typically employ much shorter fragments, on the order of 100 nucleotides per strand.

RNA can also form a double helix (called A-RNA), but the structure is shorter ($\sim$27Å) and fatter ($\sim$26Å) and the base pairs are significantly tilted with respect to the helical axis [24]. A single strand of RNA or DNA can also fold back on itself to form intramolecular double helices, resulting in hairpins or other more complicated patterns called secondary structures (see Figure 1.1). The stability of base pairing interactions are dependent on cations such as Na^+ and Mg^{2+} to stabilize and shield the negatively charged backbone.

A.2 Nucleic Acids in Biology and Engineering

Traditionally, DNA and RNA are regarded solely with respect to their biological roles as the carriers and transmitters of genetic information. According to the central dogma of molecular biology [24], double stranded DNA stores genetic sequences which can be easily replicated with the help of proteins. DNA can also be transcribed to generate single stranded messenger RNAs, which in turn are translated by ribosomes to create proteins. Unlike DNA, RNA primarily exists in single-stranded form, making self-complementary interactions much more significant for both the structure and function of the molecule *in vivo*. For example, hairpins in messenger RNA play an important role in transcription termination, and the cloverleaf structures of transfer RNA are essential in the translation of amino acids [24]. RNAs have also been found to play a role in the regulation of gene expression, through mechanisms such as riboregulator domains [9] and RNA interference [7]. Even more striking are RNAs with catalytic activity. These include the hammerhead ribozyme [6], a nucleic acid with self-cleaving abilities and the RNA component of the ribosome, a key catalytic domain in protein synthesis [25, 2].

Nucleic acids have many properties that make them well-suited for engineering applications. Current techniques in chemical synthesis allow for the inexpensive production and purification of arbitrary sequences of single stranded DNA, up to lengths of about 100 bases. In addition, fluo-

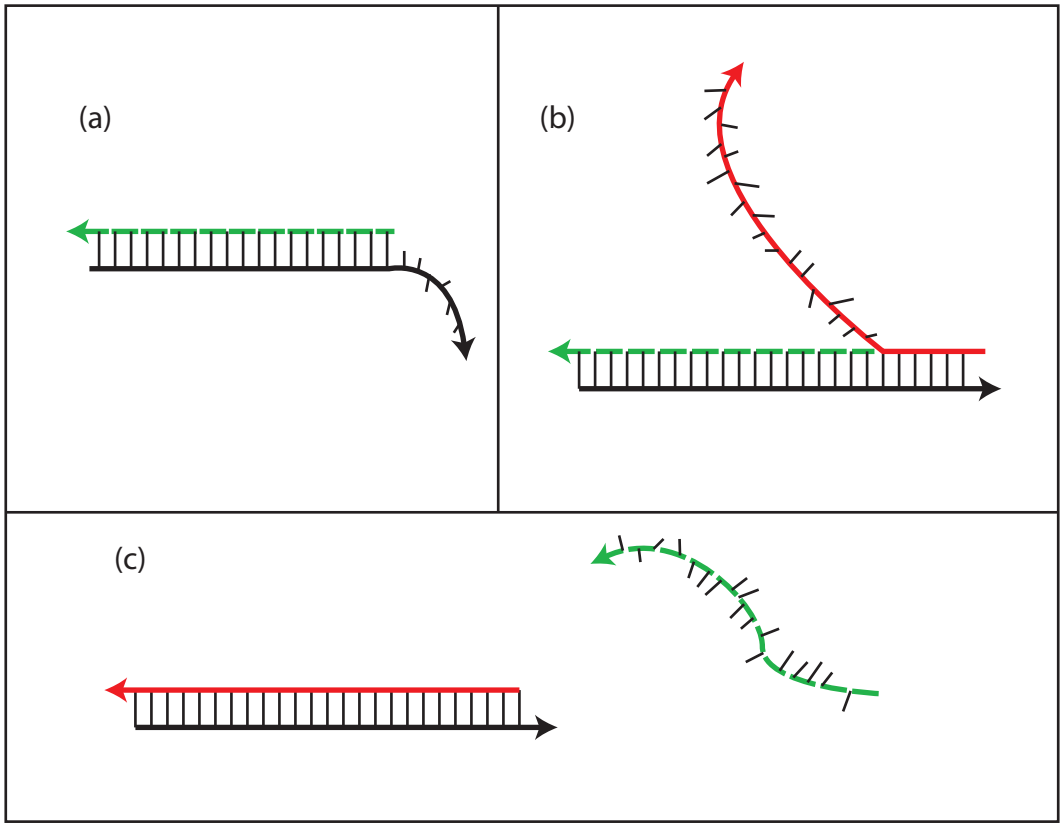


Figure A.2. **(a)** The bottom strand of this double helix has a six base overhang. **(b)** A second strand, perfectly complementary to the bottom strand, can bind at the six base overhang, called a toehold or sticky end. **(c)** A subsequent competition occurs between the dotted strand and the new strand, analogous to a random walk in two dimensions. Eventually the dotted strand is displaced as it has no toehold. In all schematics of this type, arrowheads indicate the 3' end of a nucleic acid.

rophores, linkers and modified bases can be readily incorporated into the syntheses of these molecules. "Simple" base pairing rules allow for the rational design of sequences that will stick together in a specified manner, allowing for the construction of a vast array of shapes including DNA tiles, tubes and polyhedra [17, 4, 12, 26, 8, 27, 13, 14, 16, 19] on a nanometer to micrometer length scale. Dynamic structures, such as molecular tweezers, switches and walkers [29, 28, 11, 22, 1, 20, 18, 21] can also be made of DNA by taking advantage of hybridization and strand displacement (Figure A.2). RNA can also be used for these engineering applications, although the enhanced stability[2] of DNA over RNA with respect to degradation makes RNA a less favorable choice.

In addition, a wealth of naturally occurring enzymes that manipulate nucleic acids in sequence-

[2]This stability is primarily due to the lack of a 2' hydroxyl group in the backbone of DNA

specific and structurally dependent ways have been isolated and are commercially available. These allow for such techniques as ^{32}P radiolabelling, selective cleavage, *in vitro* transcription and polymerase chain reaction (PCR). A process called SELEX [23, 5] selects and amplifies nucleic acids from random combinatorial libraries, allowing for the discovery of aptamers (RNA or DNA that bind specific substrates [10]) and (deoxy)ribozymes (nucleic acids with catalytic activities [3]). Combining these enzymatic methods with developments in systems design and hybridization techniques, the emerging field of nucleic acid engineering has already started to produce useful nucleic acid devices [15] and should continue to flourish as understanding increases and technologies mature.

Bibliography

[1] P. Alberti and J.-L. Mergny. DNA duplex-quadruplex exhange as the basis for a nanomolecular machine. *Proceedings of the National Academy of Sciences of the United States of America*, 100:1569–1573, 2003.

[2] N. Ban, P. Nissen, J. Hansen, P. B. Moore, and T. A. Steitz. The complete atomic structure of the large ribosomal subunit at 2.4 angstrom resolution. *Science*, 289(5481):905–920, 2000.

[3] D. P. Bartel and P. J. Unrau. Constructing an rna world. *Trends in Biochemical Sciences*, 24(12):M9–M13, 1999.

[4] J. Chen and N. C. Seeman. The synthesis from DNAs of a molecule with the connectivity of a cube. *Nature*, 350:631–633, 1991.

[5] A. D. Ellington and J. W. Szostak. In vitro selection of RNA molecules that bind specific ligands. *Nature*, 346:818–822, 1990.

[6] M. J. Fedor and O. C. Uhlenbeck. Substrate sequence effects on "hammerhead" RNA catalytic efficiency. *Proceedings of the National Academy of Sciences of the United States of America*, 87:1668–1672, 1990.

[7] T. Gura. A silence that speaks volumes. *Nature*, 404(6780):804–808, 2000.

[8] T. H. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, J. H. Reif, and N. C. Seeman. Construction, analysis, ligation and self-assembly of DNA triple crossover complexes. *Journal of the American Chemical Society*, 122:1848–1869, 2000.

[9] R. A. Lease, M. E. Cusick, and M. Belfort. Riboregulation in escherichia coli: DsrA RNA acts by RNA: RNA interactions at multiple loci. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, 95(21):12456–12461, 1998.

[10] J. F. Lee, J. R. Hesselberth, L. A. Meyers, and A. D. Ellington. Aptamer database. *Nucleic Acids Research*, 32:D95–D100, 2004.

[11] J. J. Li and W. Tan. A single DNA molecule nanomotor. *Nano Letters*, 2(4):315–318, 2002.

[12] X. Li, X. Yang, J. Qi, and N. C. Seeman. Antiparallel DNA double crossover molecules as components for nanoconstruction. *Journal of the American Chemical Society*, 118:6131–6140, 1996.

[13] D. Liu, S. H. Park, J. H. Reif, and T. H. LaBean. DNA nanotubes self-assembled from triple-crossover tiles as templates for conductive nanowires. *Proceedings of the National Academy of Sciences of the United States of America*, 101(3):717–722, 2004.

[14] D. Liu, M. Wang, Z. Deng, R. Walulu, and C. Mao. Tensegrity: Construction of rigid DNA triangles with flexible four-arm DNA junctions. *Journal of the American Chemical Society*, 126:2324–2325, 2004.

[15] J. W. Liu and Y. Lu. Accelerated color change of gold nanoparticles assembled by dnazymes for simple and fast colorimetric Pb^{2+} detection. *Journal of the American Chemical Society*, 126(39):12298–12305, 2004.

[16] P. W. K. Rothemund, A. Ekani-Nkodo, N. Papadakis, A. Kumar, D. K. Fygenson, and E. Winfree. Design and characterization of programmable DNA nanotubes. *Journal Of The American Chemical Society*, 126(50):16344–16352, 2004.

[17] N. C. Seeman. Nucleic acid junctions and lattices. *Journal of Theoretical Bioogy.*, 99:237–247, 1982.

[18] W. B. Sherman and N. C. Seeman. A precisely controlled DNA biped walking device. *Nano Letters*, 4(7):1203–1207, 2004.

[19] W. M. Shih, J. D. Quispe, and G. F. Joyce. A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature*, 427:618–621, 2004.

[20] J.-S. Shin and N. A. Pierce. Rewritable memory by controllable nanopatterning of DNA. *Nano Letters*, 4(5):905–909, 2004.

[21] J.-S. Shin and N. A. Pierce. A synthetic DNA walker for molecular transport. *Journal of the American Chemical Society*, 126:10834–10835, 2004.

[22] F. C. Simmel and B. Yurke. A DNA-based molecular device switchable between three distinct mechanical states. *Applied Physics Letters*, 80(5):883–885, 2002.

[23] C. Tuerk and L. Gold. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage t4 DNA polymerase. *Science*, 249(4968):505–510, 1990.

[24] D. Voet, J. G. Voet, and C. W. Pratt. *Fundamentals of Biochemistry*. Wiley and Sons, New York, 1 edition, 1999.

[25] B. T. Wimberly, D. E. Brodersen, W. M. Clemons, R. J. Morgan-Warren, A. P. Carter, C. Vonrhein, T. Hartsch, and V. Ramakrishnan. Structure of the 30s ribosomal subunit. *Nature*, 407(6802):327–339, 2000.

[26] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.

[27] H. Yan, T. H. LaBean, L. Feng, and J. H. Reif. Directed nucleation assembly of DNA tile complexes for barcode-patterned lattices. *Proceedings of the National Academy of Sciences of the United States of America*, 100(14):8103–8108, 2003.

[28] H. Yan, X. Zhang, Z. Shen, and N. C. Seeman. A robust DNA mechanical device controlled by hybridization topology. *Nature*, 415(6867):62–5, 2002.

[29] B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, and J. L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406:605–608, 2000.

Appendix B

Supplementary Material for the Analysis of Nucleic Acids

The work presented here is heavily based on the supplementary material provided with:

R. M. Dirks, and N. A. Pierce, *A partition function algorithm for nucleic acid secondary structure including pseudoknots.* Journal of Computational Chemistry, 2003. **24**(13): p. 1664-1677. Reprinted with copyright permissions.

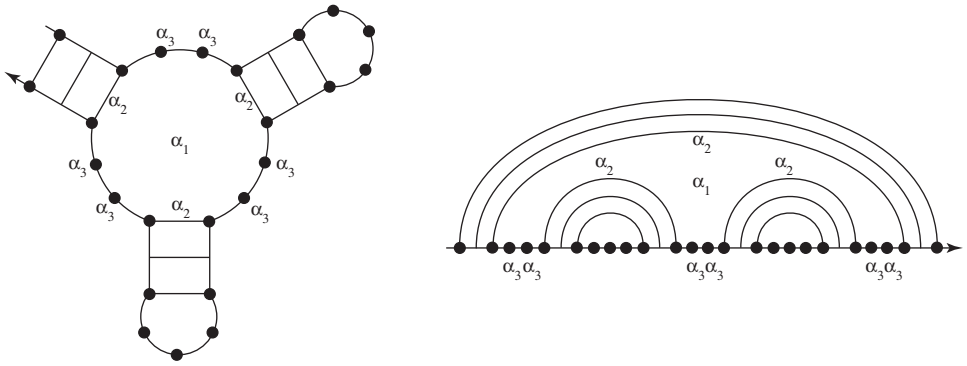


Figure B.1. Multiloop energy expression. For this example there are three base pairs defining the multiloop ($B = 3$) and six unpaired bases in the multiloop ($U = 6$) so $G^{multi} = \alpha_1 + 3\alpha_2 + 6\alpha_3$.

```
function fastiloops(i, j, l, Q^b, Q^x, Q^x2)
//Q^x recursion for O(N^3) internal loop contributions to Q^b
if (l ≥ 15) //smallest subsequence not added to Q^b as special case
  L1 = 4 //explicitly add in terms for L1 = 4, L2 ≥ 4
  d = i + L1 + 1
  for L2 = 4, j−d−5
    s = L1 + L2
    e = j − L2 − 1
    G^partial = γ1(s) + γ2(|L1 − L2|) + γ3(e, d, e+1, d−1)
    Q^x_{i,s} += exp{−G^partial/RT} Q^b_{d,e}
  L2 = 4 //explicitly add in terms for L1 ≥ 5, L2 = 4
  e = j−L2−1
  for L1 = 5, e−i−5
    s = L1 + L2
    d = i + L1 + 1
    G^partial = γ1(s) + γ2(|L1 − L2|) + γ3(e, d, e+1, d−1)
    Q^x_{i,s} += exp{−G^partial/RT} Q^b_{d,e}
//Next convert Q^x into interior loop energies
for s = 8, l−7
  if (sequence permits i·j base pair)
    Q^b_{i,j} += Q^x_{i,s} exp{−γ3(i, j, i+1, j−1)/RT}
//Extend loops from s to s+2 for future calculation
//of Q^b_{i−1,j+1} with subsequence length l+2
if (i ≠ 1 & j ≠ N)
  for s = 8, l−7
    Q^x2_{i−1,s+2} = Q^x_{i,s} exp{−[γ1(s+2) − γ1(s)]/RT}
//Add small inextensible interior loop terms to Q^b as special cases
for L1 = 0, 3
  d = i + L1 + 1
  for L2 = 0, min(3, j−d−5)
    e = j − L2 − 1
    Q^b_{i,j} += exp{−G^internal_{i,d,e,j}/RT} Q^b_{d,e}
//Add bulge loops and large asymmetric loops as special cases
for L1 = 0, 3 //Cases L1 = 0, 1, 2, 3, L2 ≥ 4
  d = i + L1 + 1
  forL2 = 4, j−d−5
    e = j − L2 − 1
    Q^b_{i,j} += exp(−G^internal_{i,d,e,j}/RT) Q^b_{d,e}
for L2 = 0, 3//Cases L1 ≥ 4, L2 = 0, 1, 2, 3
  e = j−L2−1
  for L1 = 4, e−i−5
    d = i + L1 + 1
    Q^b_{i,j} += exp{−G^internal_{i,d,e,j}/RT} Q^b_{d,e}
```

Figure B.2. Pseudocode for computing interior loop contributions to Q^b in $O(N^3)$ as an alternative to the $O(N^4)$ interior loop recursion of Figure 2.6. 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. A schematic representation of "fastiloops" is provided in Figure B.3. The smallest "possible extensible loop" is the case $L_1 = L_2 = 4$ with size $s = 8$. Therefore, the smallest subsequence for which Q^x can be employed is $l = 15$ (adding the four bases for i, d, e, j and a minimum hairpin of three bases between $d{\cdot}e$). For a given i and j, $Q^x_{i,s}$ already contains the contributions to $Q^b_{i,j}$ for all extensible loops of size s except for the two cases when either $L_1 = 4$ or $L_2 = 4$ (which cannot be obtained by extending smaller loops that use a different energy expression). Enriching $Q^x_{i,s}$ with these two new possible extensible loops, we then convert $Q^x_{i,s}$ into contributions to $Q^b_{i,j}$ by introducing the term for closing these loops with pair $i{\cdot}j$. $Q^x_{i,s}$ is then extended to provide future values of $Q^x_{i-1,s+2}$. All other interior loop contributions (cases with either $L_1 \leq 3$ or $L_2 \leq 3$) are then added directly to $Q^b_{i,j}$ using the special energy expressions of the standard model implied by $G^{\text{internal}}_{i,d,e,j}$. Note that the subsequence length l is fixed inside each call to the function "fastiloops". Hence, specifying i implies $j = i+l-1$. For subsequences of length l, we use $Q^x_{i,s}$ (j implied) to compute $Q^x_{i-1,s+2}$ ($j+1$ implied) which will later be used to compute contributions to $Q^b_{i-1,j+1}$ for subsequences of length $l+2$. Thus, for a given value of l, the values of $Q^x_{i,s}$ need only be stored for all legal values of i and s until l has been incremented 3 times, at which point it can be discarded. This is accomplished by using $Q^{x1}_{i,s}$ and $Q^{x2}_{i,s}$ to store future contributions for subsequences of length $l+1$ and $l+2$.

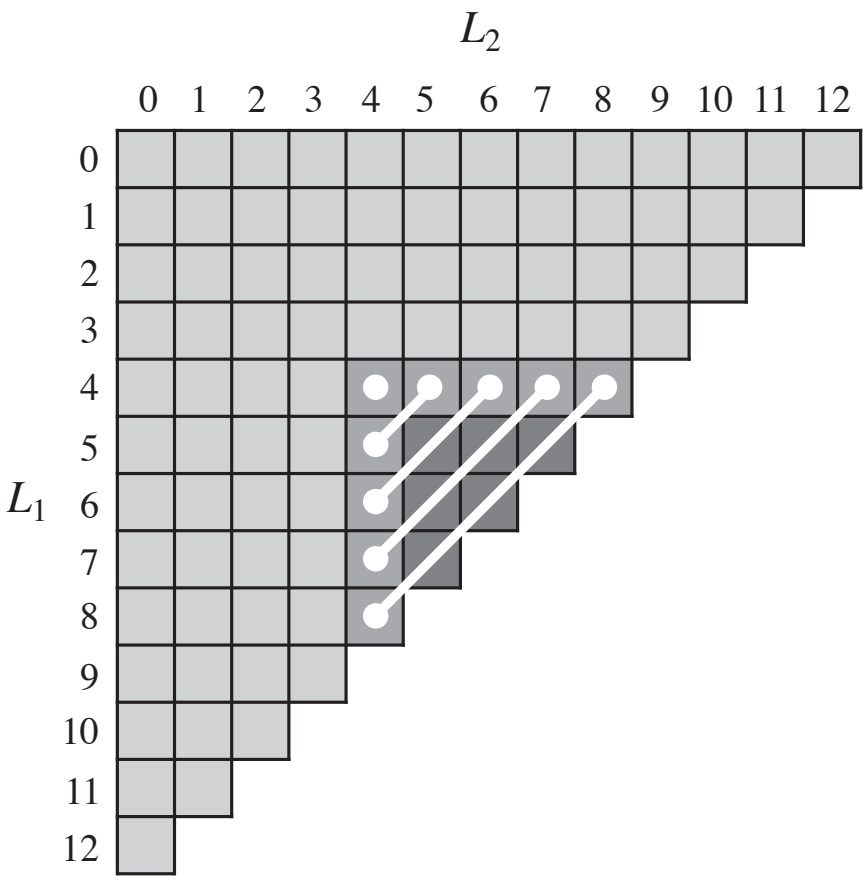


Figure B.3. Schematic for interpreting the pseudocode of Figure B.2, which computes the interior loop contributions to the partition function in $O(N^3)$. For a given i (with j implied by the current subsequence length l), the grid illustrates the method of computing contributions to $Q^b_{i,j}$ for all interior loops with sides of length L_1 and L_2. For the depicted case, the maximum interior loop size is $s = L_1 + L_2 = 12$. Adding seven bases to account for the closing bases i, d, e, j and the smallest allowed hairpin of three bases between d·e, the subsequence under consideration is therefore of length $l = 19$. The $O(N)$ pale gray cases with either $L_1 \leq 3$ or $L_2 \leq 3$ use special energy functions and are added explicitly to $Q^b_{i,j}$. The medium gray and dark gray cases are the possible extensible loop contributions that are computed using $Q^x_{i,s}$. Each diagonal line spans the terms that will be stored in $Q^x_{i,s}$ for a particular value of s. The medium gray terms are the new possible extensible loops with either $L_1 = 4$ or $L_2 = 4$. The dark gray terms were previously incorporated into to $Q^x_{i,s}$ by extending smaller possible extensible loops.

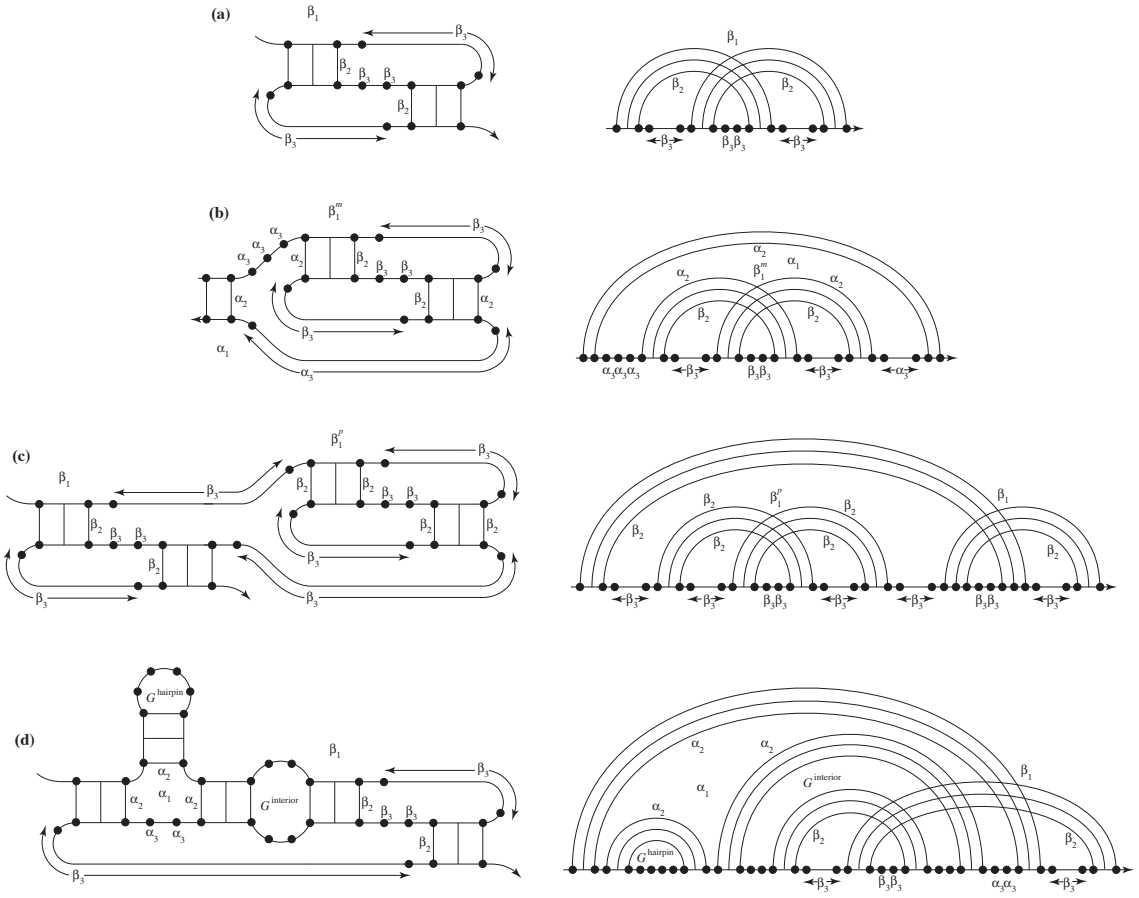


Figure B.4. Illustration of the pseudoknot energy expression. (a) An external pseudoknot. Bases external to the pseudoknot have no associated energy. The penalty for forming an external pseudoknot is β_1. Base pairs that border the pseudoknot interior receive penalty β_2 while unpaired bases on the pseudoknot interior receive penalty β_3. The energies associated with the stacked base pairs are described using the standard model. (b) A pseudoknot inside a multiloop. The penalty for forming a pseudoknot inside a multiloop is β_1^m. The treatment of β_2 and β_3 inside the pseudoknot remains the same. In addition, the standard penalty for formation of a multiloop is α_1, the two pseudoknot base pairs that border the multiloop are given the standard multiloop penalty α_2, and unpaired bases that are inside the multiloop are given penalty α_3. (c) A pseudoknot within a pseudoknot. The energy treatment for the exterior pseudoknot remains the same. The penalty for forming a pseudoknot inside a pseudoknot is β_1^p. Base pairs from the interior pseudoknot that border the exterior pseudoknot receive penalty β_2. Otherwise, the energetic treatment of the interior pseudoknot is the same as for an exterior pseudoknot. The partition function recursions allow an arbitrary number of levels of pseudoknots within pseudoknots. (d) Pseudoknot with a hairpin and an interior loop inside a spanning region of the pseudoknot. The multiloop that forms at the base of the hairpin is treated using the standard multiloop potential. In general, the spanning region of a pseudoknot may contain interior loops, hairpins, multiloops or additional pseudoknots.

```
Initialize (Q, Q^b, Q^m, Q^p, Q^z) //O(N^2) space
Initialize (Q^g) //O(N^4) space
Set all values to 0 except Q_{i,i-1} = Q^z_{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 //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)
    for d = i + 2, j-6 //set inner pair d·e
      for e = d+4, j - 2
        for c = i + 1, d-1 //recursion on middle pair c·f
          for f = e+1, j - 1
            Q^g_{i,d,e,j} += exp(-G^interior_{i,c,f,j}/RT) Q^g_{c,d,e,f}
    for d = i + 6, j-5 //set inner pair d·e
      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 //set inner pair d·e
      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 //set inner pair d·e
      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 //set inner pair d·e
      for e = d+4, j - 2
        for c = i + 6, d-1 //recursion on middle pair c·f
          for f = e+1, j - 1
            G^recursion = α1+2α2+α3(j-f-1)
            Q^g_{i,d,e,j} += Q^m_{i+1,c-1} Q^g_{c,d,e,f} exp{-G^recursion/RT}
    for d = i + 2, j-11 //set inner pair d·e
      for e = d+4, j - 7
        for c = i + 1, d-1 //recursion on middle pair c·f
          for f = e+1, j - 6
            G^recursion = α1+2α2+α3(c-i-1)
            Q^g_{i,d,e,j} += exp{-G^recursion/RT} Q^g_{c,d,e,f} Q^m_{f+1,j-1}
    for d = i + 7, j-11 //set inner pair d·e
      for e = d+4, j - 7
        for c = i + 6, d-1 //recursion on middle pair c·f
          for f = e+1, j - 6
            Q^g_{i,d,e,j} += Q^m_{i+1,c-1} exp{-[α1+2α2]/RT} Q^g_{c,d,e,f} Q^m_{f+1,j-1}
    //Q^p recursion
    for a = i + 1, j-7 //place points from left to right
      for b = a+1, j-6
        for c = b + 1, j-5
          for d = max(c+1, a+4), j - 3
            for e = d+1, j - 2
              for f = max(e+1, c+4), j - 1
                Q^p_{i,j} += Q^z_{a+1,b-1} Q^z_{c+1,d-1} Q^z_{e+1,f-1}
                           · Q^g_{i,a,d,e} Q^g_{b,c,f,j} exp{-2β2/RT}
    //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}
```

Figure B.5. Pseudocode implementation of an $O(N^8)$ 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 2.10-2.14. In this pseudocode, care has been taken to define programming loop bounds so as to consider only valid secondary structures. The standard model requires three or more unpaired bases in a hairpin, so a b-curve must satisfy $j - i \geq 4$. Looking at Figure 2.13, imposing this requirement on all base pairs implies that a p-curve must satisfy $j - i \geq 8$. In the interior of a pseudoknot, the steric constraints on hairpins sometimes lead to two conflicting requirements that are incorporated using a "max" function to define the bounds for d and f. The disallowed structures have infinite energies according to the physical model so it is computationally efficient to exclude them from consideration.

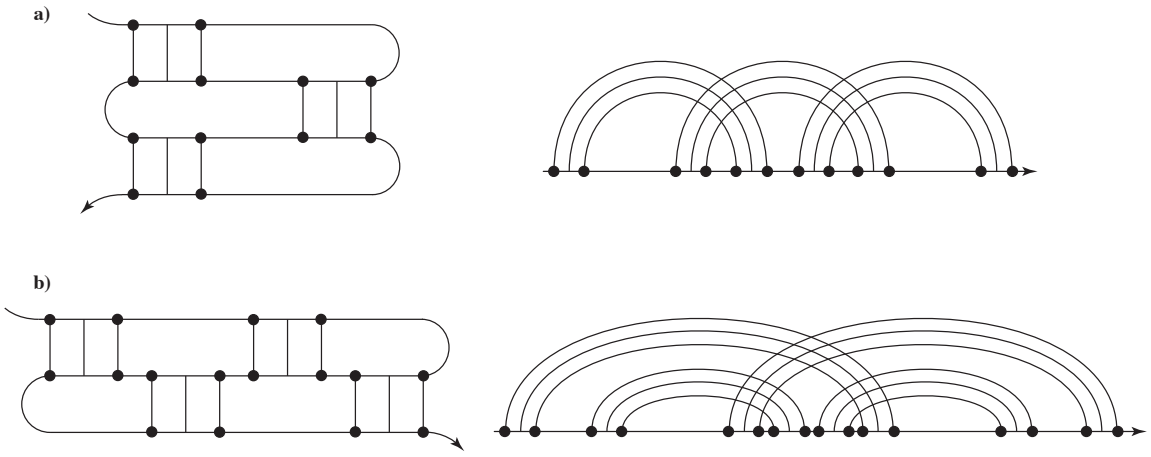


Figure B.6. Examples of pseudoknots that are excluded from the partition function recursions. Neither structure can be decomposed into two spanning regions as required by Figure 2.13. The structure prediction recursions of Rivas and Eddy [2] include both structures while the the structure prediction recursions of Akutsu [1] include the latter.

```
function fastiloops(i, j, l, Q^g, Q^x, Q^x2)
//Q^x recursion for O(N^5) internal loop contributions to Q^g
if (l ≥ 17) //smallest subsequence not added to Q^g as special case
  for d = i+6, j−10
    for e = d+4, j−6
      L1 = 4 //explicitly add in terms for L1 = 4, L2 ≥ 4
      c = i + L1 + 1
      for L2 = 4, j−e−2
        s = L1 + L2
        f = j − L2 − 1
        G^partial = γ1(s) + γ2(|L1 − L2|) + γ3(f, c, f+1, c−1)
        Q^x_{i,d,e,s} += exp{−G^partial/RT} Q^g_{c,d,e,f}
      if ( d ≥ i + 7)
        L2 = 4 //explicitly add in terms for L1 ≥ 5, L2 = 4
        f = j − L2 − 1
        for L1 = 5, d−i−2
          s = L1 + L2
          c = i + L1 + 1
          G^partial = γ1(s) + γ2(|L1 − L2|) + γ3(f, c, f+1, c−1)
          Q^x_{i,d,e,L1+L2} += exp{−G^partial/RT} Q^g_{c,d,e,f}
for d = i+1, j−5
  for e = d+4, j−1
    //Convert Q^x into interior loop energies
    if (l ≥ 17 & sequence permits i·j base pair)
      for size = 8, l−9
        Q^g_{i,d,e,j} += Q^x_{i,d,e,s} exp{−γ3(i, j, i + 1, j − 1)/RT}
    //Extend loops for future use
    if (i ≠ 1 & j ≠ N)
      for s = 8, l−9
        Q^x2_{i−1,d,e,s+2} = Q^x_{i,d,e,s} exp{−[γ1(s+2) − γ1(s)]/RT}
    //Add small inextensible interior loops to Q^g as special cases
    for L1 = 0, min(3, d−i−2)
      c = i + L1 + 1
      for L2 = 0, min(3, j−e−2)
        f = j − L2 − 1
        Q^g_{i,d,e,j} += exp{−G^interior_{i,c,f,j}/RT} Q^g_{c,d,e,f}
    //Add bulge loops and large asymmetric loops as special cases
    for L1 = 0, min(3, d−i −2) //Cases L1 = 0, 1, 2, 3, L2 ≥ 4
      c = i + L1 + 1
      for L2 = 4, j−e−2
        f = j − L2 − 1
        Q^g_{i,d,e,j} += exp{−G^interior_{i,c,f,j}/RT} Q^g_{c,d,e,f}
    for L2 = 0, min(3, j−e−2) //Cases L1 ≥ 4, L2 = 0, 1, 2, 3
      f = j − L2 − 1
      for L1 = 4, d−i−2
        c = i + L1 + 1
        Q^g_{i,d,e,j} += exp{−G^interior_{i,c,f,j}/RT} Q^g_{c,d,e,f}
```

Figure B.7. Pseudocode for computing interior loop contributions to Q^g in $O(N^5)$ as an alternative to the $O(N^6)$ interior loop recursion of Figure 2.17. 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 smallest "possible extensible loop" is the case $L_1 = L_2 = 4$ with size $s = 8$. Therefore, the smallest subsequence for which Q^x can be employed is $l=17$ (adding the four closing bases i, c, f, j, the additional spanning pair $d \cdot e$, and a minimum hairpin loop of three bases). For given values of i, d and e, $Q^x_{i,d,e,s}$ already contains the contributions to $Q^g_{i,d,e,j}$ for all extensible loops of size s except for the two cases when either $L_1 = 4$ or $L_2 = 4$ (which cannot be obtained by extending smaller loops that use a different energy expression). Enriching $Q^x_{i,d,e,s}$ with these two new possible extensible loops, we then convert $Q^x_{i,d,e,s}$ into contributions to $Q^g_{i,d,e,j}$ by introducing the term for closing these loops with pair $i \cdot j$. $Q^x_{i,d,e,s}$ is then extended to provide future values of $Q^x_{i-1,d,e,s+2}$. All other interior loop contributions (cases with either $L_1 \leq 3$ or $L_2 \leq 3$) are then added directly to $Q^b_{i,j}$ using the special energy expressions of the standard model implied by $G^{\text{internal}}_{i,d,e,j}$. Note that the subsequence length l is fixed inside each call to the function "fastiloops". Hence, specifying i implies $j = i+l-1$. For subsequences of length l, we use $Q^x_{i,d,e,s}$ (j implied) to compute $Q^x_{i-1,d,e,s+2}$ ($j+1$ implied) which will later be used to compute contributions to $Q^g_{i-1,d,e,j+1}$ for subsequences of length $l+2$. Thus, for a given value of l, the values of $Q^x_{i,d,e,s}$ need only be stored for all legal values of i, d, e, and s until l has been incremented 3 times, at which point it can be discarded. This is accomplished by using $Q^{x1}_{i,d,e,s}$ and $Q^{x2}_{i,d,e,s}$ to store future contributions for subsequences of length $l+1$ and $l+2$.

Bibliography

[1] T. Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Applied Mathematics*, 104:45–62, 2000.

[2] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285:2053–2068, 1999.

Appendix C

Reducing Computational Complexity

The algorithms presented in the main text provide an inefficient treatment of interior loops. By exploiting the form of the interior loop potential function, the computational complexity of the partition function algorithms excluding and including pseudoknots can be reduced by a factor of N, where N is the sequence length [2, 1]. A detailed description of the "fastiloops" treatment is provided in [1] and the corresponding Supplementary Material. The "fastiloops" modification detracts from the simplicity of the presentation because the necessary recursions do not conform to the same structure as the other terms in the algorithm. Here, we describe the extension of this approach to recursion probability algorithms.

In the unpseudoknotted case, pseudocode for an $O(N^3)$ partition function algorithm is provided in figure 11 of [1], which employs the "fastiloops" function of Supplementary Material figure S2. To this point, we have assumed that all Q-type values are accessible at the end of the partition function calculation. For the "fastiloops" methods, the values Q^x, Q^{x1} and Q^{x2} are computed on the fly and discarded to save memory. Hence, for the recursion probability algorithm, it is necessary to recompute the Q^x-type terms at the same time that the corresponding P^x-type terms are calculated. An $O(N^3)$ recursion probability algorithm that excludes pseudoknots is described in figure A1, which references the function "fastiloopsN3" of figure A2. If pseudoknots are included, the computational complexity of the recursion probability algorithm in figure 3.4 is reduced to $O(N^5)$ using "fastiloopsN5" described in figure A3. A few aspects of the "fastiloopsN3" and "fastiloopsN5"

Compute Q, Q^b, Q^m, Q^s, Q^{ms} using $O(N^3)$ partition function algorithm
Initialize ($Q^x, Q^{x1}, Q^{x2}, P, P^b, P^m, P^s, P^{ms}, P^x, P^{x1}, P^{x2}$) // $O(N^2)$ space
Set all Q^x-type and P-type values to 0
$P_{1,N} = 1$ //probability of "recursing" to the entire strand is 1
for $l = N, 1$ //decrement subsequence length
    Initialize $Q^x = Q^{x1}$, $Q^{x1} = Q^{x2}$, $Q^{x2} = 0$
    Initialize $P^x = P^{x1}$, $P^{x1} = P^{x2}$, $P^{x2} = 0$
    for $i = 1, N-l+1$
        $j = i+l-1$
        //P, P^m **recursions**
        for $d = i, j-4$
            $\Delta p = P_{i,j} Q_{i,d-1} Q^s_{d,j} / Q_{i,j}$
            $P_{i,d-1} \mathrel{+}= \Delta p$
            $P^s_{d,j} \mathrel{+}= \Delta p$
            $P^{ms}_{d,j} \mathrel{+}= P^m_{i,j} \exp\{-\alpha_3[d-i]/RT\}\, Q^{ms}_{d,j} / Q^m_{i,j}$
            $\Delta p = P^m_{i,j} Q^m_{i,d-1} Q^{ms}_{d,j} / Q^m_{i,j}$
            $P^m_{i,d-1} \mathrel{+}= \Delta p$
            $P^{ms}_{d,j} \mathrel{+}= \Delta p$
        //P^s, P^{ms} **recursions**
        for $d = i+4, j$ // loop over all possible rightmost pairs $i \cdot d$
            $P^b_{i,d} \mathrel{+}= P^s_{i,j} Q^b_{i,d} / Q^s_{i,j}$
            $P^b_{i,d} \mathrel{+}= P^{ms}_{i,j} Q^b_{i,d} \exp\{-[\alpha_2 + \alpha_3(j-d)]/RT\} / Q^{ms}_{i,j}$
        // P^b **recursion**
        // Compute internal loop contributions to P^b in $O(N^3)$
        call fastiloopsN3($i, j, l, Q^b, Q^x, Q^{x2}, P^b, P^x, P^{x2}$)
        for $d = i+6, j-5$
            $\Delta p = P^b_{i,j}\, Q^m_{i+1,d-1}\, Q^{ms}_{d,j-1} \exp\{-[\alpha_1 + \alpha_2]/RT\} / Q^b_{i,j}$
            $P^m_{i+1,d-1} \mathrel{+}= \Delta p$
            $P^{ms}_{d,j-1} \mathrel{+}= \Delta p$

Figure C.1. $O(N^3)$ recursion probability algorithm that excludes pseudoknots. The algorithm proceeds from longer subsequences to shorter ones, so in contrast to the analogous partition function algorithm (see figure 11 of [1]), Q^{x1} and Q^{x2} refer to subsequences whose lengths are shorter (by 1 and 2, respectively) than the current subsequence of length l.

routines deserve mention. It is advisable to review the relevant sections of chapter 2 and appendix B before proceeding.

An interior loop with closing pair $i \cdot j$ and interior pair $d \cdot e$ has energy $G^{\text{interior}}_{i,d,e,j}$, sides of lengths

$$L_1 \equiv (d - i - 1), \quad L_2 \equiv (j - e - 1),$$

and size $L_1 + L_2$. Loops with both $L_1 \geq 4$ and $L_2 \geq 4$ are termed "extensible" and their contributions to the partition function algorithm are calculated using Q^x. Furthermore, Q^x also contains information about "possible extensible loops" for which the definitions of L_1, L_2 are the same but i and j are not required to base-pair.

The partition function algorithm examines subsequences of length $l = j-i+1$, starting with $l=1$

and ending with $l=N$. Q^x is efficiently calculated using the extension identity (see equation 15 of [1])

$$\begin{aligned} Q^x_{i-1,s+2} &= \Gamma(s+2)\big|_{\substack{L_1=4 \\ L_1+L_2=s+2}} + \Gamma(s+2)\big|_{\substack{L_2=4 \\ L_1+L_2=s+2}} \\ &+ \left[Q^x_{i,s} \exp\{-[\gamma_1(s+2)-\gamma_1(s)]/RT\} \right] \end{aligned} \tag{C.1}$$

which relates $Q^x_{i,s}$ (for subsequences of length l) to $Q^x_{i-1,s+2}$ (for subsequences of length $l+2$). The first line "seeds" Q^x with cases at an extension border ($L_1 = 4$ or $L_2 = 4$) for subsequent extension to longer subsequences. For conciseness, we have introduced the definition

$$\Gamma(s) \equiv \exp\Big\{ -\Big[\gamma_1(s) + \gamma_2(|L_1-L_2|) + \gamma_3(e,d,e+1,d-1)\Big]/RT \Big\}\, Q^b_{d,e},$$

where d and e are defined implicitly in terms of L_1 and L_2. For implementation purposes, the second line of (C.1) is calculated during the l, i loop and temporarily stored in $Q^{x2}_{i-1,s+2}$. The first line of (C.1) is added to this contribution in the $l+2$, $i-1$ loop. As a result of this two step procedure, we adopt the convention that L_1 and L_2 are defined with respect to the loop index in which they are calculated (i.e. l, i for the second line and $l+2$, $i-1$ for the first line). This convention facilitates the comparison of the extension identity with pseudocode.

The recursion probability algorithm examines subsequences of length l starting with $l=N$ and ending with $l=1$. To recompute Q^x in this context, we use the contraction identity

$$\begin{aligned} Q^x_{i+1,s-2} &= \sum_{\substack{i=1 \\ L_1\geq 4,\ L_2\geq 4 \\ L_1+L_2=s-2}} \Gamma(s-2) + \sum_{\substack{j=N \\ L_1\geq 4,\ L_2\geq 4 \\ L_1+L_2=s-2}} \Gamma(s-2) \\ &+ \left[\Big(Q^x_{i,s} - \Gamma(s)\big|_{\substack{L_1=4 \\ L_1+L_2=s}} - \Gamma(s)\big|_{\substack{L_2=4 \\ L_1+L_2=s}} \Big) \exp\Big\{ -[\gamma_1(s-2)-\gamma_1(s)]/RT \Big\} \right] \end{aligned} \tag{C.2}$$

which relates $Q^x_{i,s}$ (for subsequences of length l) to $Q^x_{i+1,s-2}$ (for subsequences of length $l-2$). The first line "seeds" Q^x with cases that are both extensible ($L_1 \geq 4$ and $L_2 \geq 4$) and at an end of the strand ($i = 1$ or $j = N$). For implementation purposes, the second line of (C.2) is calculated during the l, i loop and temporarily stored in $Q^{x2}_{i+1,s-2}$. The first line of (C.2) is added to this contribution in the $l-2$, $i+1$ loop. We retain the convention that L_1 and L_2 are defined with respect to the loop index in which they are calculated (i.e. l, i for the second line and $l-2$, $i+1$ for the first line).

Derivation of the algorithm to compute P^x requires careful consideration. The quantities Q^x and Q^{x2} contain incomplete partition function information for "possible extensible loops", but they do not represent subsequence partition functions in the manner of other Q-type matrices. In a normal recursion relation, Q-type matrices on the right hand side are subsequence partition functions describing a local structural motif that contributes to the larger subsequence partition function on the left hand side. $Q^x_{i,s}$ contains information about possible extensible loops that may not actually exist (if i and j are not complementary). The extension identity (C.1) passes this potentially useful information on to $Q^{x2}_{i-1,s+2}$. Consider, for example, a chain of Q^x values related by the extension identity in a case where no complementary $i \cdot j$ base pair is encountered while incrementing l by 2 until an end of the strand is reached. In this scenario, the values of Q^x computed in this chain should not contribute to the corresponding recursion probabilities P^x because the values of Q^x are not identified with any secondary structure in the equilibrium ensemble. Hence, the calculation of P^x requires information about which Q^x quantities ultimately contribute to secondary structures in the ensemble. As a result, the extension identity (C.1) cannot simply be transformed using the standard recursion probability approach which assumes that both sides of the equation represent subsequence partition functions that are assured of contributing to the equilibrium ensemble. This realization suggests computing $P^x_{i,s}$ by adding the probabilities of all internal loops that rely on $Q^x_{i,s}$ to incorporate information in the partition function.

To calculate $P^x_{i,s}$ (for a fixed l), note that $Q^x_{i,s}$ will be invoked for all interior loops (i', d, e, j')

function fastiloopsN3($i, j, l, Q^b, Q^x, Q^{x2}, P^b, P^x, P^{x2}$)
//Add small non-contractible interior loop terms to P^b as special cases
for $L_1 = 0, 3$
  $d = i + L_1 + 1$
  for $L_2 = 0, \min(3, j-d-5)$
    $e = j - L_2 - 1$
    $P^b_{d,e}$ += $P^b_{i,j} \exp\{-G^{\text{internal}}_{i,d,e,j}/RT\}\, Q^b_{d,e}/Q^b_{i,j}$
//Add bulge loops and large asymmetric loops as special cases
for $L_1 = 0, 3$ //Cases $L_1 = 0, 1, 2, 3,\ L_2 \geq 4$
  $d = i + L_1 + 1$
  for $L_2 = 4, j-d-5$
    $e = j - L_2 - 1$
    $P^b_{d,e}$ += $P^b_{i,j} \exp\{-G^{\text{internal}}_{i,d,e,j}/RT\}\, Q^b_{d,e}/Q^b_{i,j}$
for $L_2 = 0, 3$//Cases $L_1 \geq 4,\ L_2 = 0, 1, 2, 3$
  $e = j-L_2-1$
  for $L_1 = 4, e-i-5$
    $d = i + L_1 + 1$
    $P^b_{d,e}$ += $P^b_{i,j} \exp\{-G^{\text{internal}}_{i,d,e,j}/RT\}\, Q^b_{d,e}/Q^b_{i,j}$
// Seed Q^x with contractible cases
// Also add cases that are at an end with $L_1 \geq 4, L_2 \geq 4$
if ($i = 1$ or $j = N$) and $l \geq 15$
  for $d = i + 5, j - 9$
    $L_1 = d - i - 1$
    for $e = d + 4, j - 5$
      $L_2 = j - e - 1$
      $s = L_1 + L_2$
      $G^{\text{partial}} = \gamma_1(s) + \gamma_2(|L_1 - L_2|) + \gamma_3(e, d, e+1, d-1)$
      $Q^x_{i,s}$ += $\exp\{-G^{\text{partial}}/RT\}\, Q^b_{d,e}$
//Use Q^x to finish calculation of P^x
if (sequence permits $i \cdot j$ base pair)
  for $s = 8, l - 7$
    $P^x_{i,s}$ += $P^b_{i,j} Q^x_{i,s} \exp\{-\gamma_3(i, j, i+1, j-1)/RT\}/Q^b_{i,j}$
//Calculate P^b contribution using Q^x and P^x
if ($l \geq 15$) // smallest subsequence not added to P^b as special case
  $L_1 = 4$ // explicitly add in terms for $L_1 = 4, L_2 \geq 4$
  $d = i + L_1 + 1$
  for $L_2 = 4, j - d - 5$
    $e = j - L_2 - 1$
    $s = L_1 + L_2$
    $G^{\text{partial}} = \gamma_1(s) + \gamma_2(|L_1 - L_2|) + \gamma_3(e, d, e+1, d-1)$
    $\Delta p = P^x_{i,s} \exp\{-G^{\text{partial}}/RT\}\, Q^b_{d,e}/Q^x_{i,s}$
    $P^b_{d,e}$ += Δp
    $P^x_{i,s}$ −= Δp // Remove border cases
    $Q^x_{i,s}$ −= $\exp\{-G^{\text{partial}}/RT\}\, Q^b_{d,e}$ // Remove border cases
  } (∗)
  $L_2 = 4$ // explicitly add in terms for $L_1 \geq 5, L_2 = 4$
  $e = j - L_2 - 1$
  for $L_1 = 5, e - i - 5$
    $d = i + L_1 + 1$
    Insert (∗)
// Store partial values for Q^{x2} and P^{x2}
for $s = 10, l - 7$
  $Q^{x2}_{i+1,s-2} = Q^x_{i,s} \exp\{-[\gamma_1(s-2) - \gamma_1(s)]/RT\}$
  $P^{x2}_{i+1,s-2} = P^x_{i,s}$

Figure C.2. Pseudocode for computing interior loop contributions to P^b in $O(N^3)$ as an alternative to the $O(N^4)$ interior loop recursion of figure 3.3

with interior pair $d \cdot e$ and closing pair $i' \cdot j'$ such that

$$i - i' = j' - j \geq 0, \quad L_1 \geq 4, \quad L_2 \geq 4, \quad L_1 + L_2 = s, \tag{C.3}$$

where L_1, L_2 and s are defined with respect to i and j. Hence, a particular loop (i', d, e, j') is identified with a set of $Q^x_{i,s}$ terms that are related by the extension identity (C.1). Alternatively, a particular $Q^x_{i,s}$ term is identified with all of the interior loops (i', d, e, j') to which it ultimately contributes via the extension identity. Consequently, from the notion of recursion probabilities introduced earlier, $P^x_{i,s}$ (for a fixed l) should be the sum of the probabilities of all interior loops (i', d, e, j') that satisfy the properties (C.3). For the case where $i-1 \leq N-j$ (the case $i-1 > N-j$ yields analogous results), it follows that

$$P^x_{i,s} = \sum_{i'=1}^{i} \sum_{\substack{L_1 \geq 4,\, L_2 \geq 4 \\ L_1 + L_2 = s}} p(i', d, e, j'), \tag{C.4}$$

where $p(i', d, e, j')$ is the probability of the (i', d, e, j') interior loop in the equilibrium ensemble of secondary structures. Since $P^{x2}_{i+1,s-2}$ is defined similarly, with l and s decremented by 2, it follows that

$$P^{x2}_{i+1,s-2} = \sum_{i'=1}^{i+1} \sum_{\substack{L_1 \geq 5,\, L_2 \geq 5 \\ L_1 + L_2 = s}} p(i', d, e, j'), \tag{C.5}$$

where L_1 and L_2 are temporarily defined with respect to i and j to retain the size constraint $L_1 + L_2 = s$. Comparing (C.4) and (C.5), we then identify the relationship

$$\begin{aligned} P^{x2}_{i+1,s-2} \;=\; & \sum_{\substack{L_1 \geq 5,\, L_2 \geq 5 \\ L_1 + L_2 = s}} p(i', d, e, j')|_{i'=i+1, j'=j-1} \\ + \; & \left[P^x_{i,s} - \sum_{i'=1}^{i} p(i', d, e, j')|_{\substack{L_1 = 4,\, L_2 \geq 4 \\ L_1 + L_2 = s}} \; - \; \sum_{i'=1}^{i} p(i', d, e, j')|_{\substack{L_1 \geq 5,\, L_2 = 4 \\ L_1 + L_2 = s}} \right], \end{aligned}$$

where L_1 and L_2 continue to be defined with respect to i and j. Finally, we shift the indices in the first line so that L_1 and L_2 are defined with respect to $i+1$ and $j-1$

$$\begin{aligned} P^{x2}_{i+1,s-2} &= \sum_{\substack{L_1 \geq 4, L_2 \geq 4 \\ L_1+L_2=s-2}} p(i',d,e,j')|_{i'=i+1,j'=j-1} \\ &+ \left[P^x_{i,s} - \sum_{i'=1}^{i} p(i',d,e,j')|_{\substack{L_1=4, L_2 \geq 4 \\ L_1+L_2=s}} - \sum_{i'=1}^{i} p(i',d,e,j')|_{\substack{L_1 \geq 5, L_2=4 \\ L_1+L_2=s}} \right]. \end{aligned} \tag{C.6}$$

This identity relates $P^x_{i,s}$ (for subsequences of length l) to $P^x_{i+1,s-2}$ (for subsequences of length $l-2$). For implementation purposes, the second line is calculated during the l, i loop and temporarily stored in $Q^{x2}_{i+1,s-2}$. Each of the sums of form $\sum_{i'=1}^{i}$ operates on a single term that is a subset of the terms in the definition of $P^x_{i,s}$ (C.4). Hence, the sums of form $\sum_{i'=1}^{i}$ in (C.6) may be evaluated implicitly as $P^x_{i,s}$ times a quotient with $Q^x_{i,s}$ in the denominator and the corresponding subset of $Q^x_{i,s}$ in the numerator. The first line is added to this contribution in the $l-2$, $i+1$ loop. There, the summation corresponds to exactly those loops treated by $Q^x_{i+1,s-2}$ in the case where $i+1$ and $j-1$ base pair. As usual, L_1 and L_2 are defined with respect to the loop index in which they are calculated (i.e. l, i for the second line and $l-2$, $i+1$ for the first line).

function fastiloopsN5($i, j, l, Q^g, Q^x, Q^{x2}, P^g, P^x, P^{x2}$)
for $d = i+1, j-5$
  for $e = d+4, j-1$
    //Add small non-contractible interior loop terms to P^g as special cases
    for $L_1 = 0, \min(3, d-i-2)$
      $c = i + L_1 + 1$
      for $L_2 = 0, \min(3, j-e-2)$
        $f = j - L_2 - 1$
        $P^g_{c,d,e,f}$ += $P^g_{i,d,e,j} \exp\{-G^{\text{internal}}_{i,c,f,j}/RT\}\, Q^g_{c,d,e,f}/Q^g_{i,d,e,j}$
    //Add bulge loops and large asymmetric loops as special cases
    for $L_1 = 0, \min(3, d-i-2)$ //Cases $L_1 = 0, 1, 2, 3,\ L_2 \geq 4$
      $c = i + L_1 + 1$
      for $L_2 = 4, j-e-2$
        $f = j - L_2 - 1$
        $P^g_{c,d,e,f}$ += $P^g_{i,d,e,j} \exp\{-G^{\text{internal}}_{i,c,f,j}/RT\}\, Q^g_{c,d,e,f}/Q^g_{i,d,e,j}$
    for $L_2 = 0, \min(3, j-e-2)$//Cases $L_1 \geq 4,\ L_2 = 0, 1, 2, 3$
      $f = j - L_2 - 1$
      for $L_1 = 4, d-i-2$
        $c = i + L_1 + 1$
        $P^g_{c,d,e,f}$ += $P^g_{i,d,e,j} \exp\{-G^{\text{internal}}_{i,c,f,j}/RT\}\, Q^g_{c,d,e,f}/Q^g_{i,d,e,j}$
// Seed Q^x with contractible cases
// Also add cases that are at an end with $L_1 \geq 4, L_2 \geq 4$
if ($i = 1$ or $j = N$) and $l \geq 17$
  for $d = i+6, j-10$
    for $e = d+4, j-6$
      for $c = i+5, d-1$
        $L_1 = c - i - 1$
        for $f = e+1, j-5$
          $L_2 = j - f - 1$
          $s = L_1 + L_2$
          $G^{\text{partial}} = \gamma_1(s) + \gamma_2(|L_1 - L_2|) + \gamma_3(f, c, f+1, c-1)$
          $Q^x_{i,d,e,s}$ += $\exp\{-G^{\text{partial}}/RT\}\, Q^g_{c,d,e,f}$
// Use Q^x to finish calculation of P^x
if (sequence permits $i \cdot j$ base pair)
  for $d = i+1, j-5$
    for $e = d+4, j-1$
      for $s = 8, l-9$
        $P^x_{i,d,e,s}$ += $P^g_{i,d,e,j} Q^x_{i,d,e,s}\ \exp\{-\gamma_3(i, j, i+1, j-1)/RT\}/Q^g_{i,d,e,j}$
// Calculate P^g contribution using Q^x and P^x
if ($l \geq 17$)
  for $d = i+6, j-10$
    for $e = d+4, j-6$
      $L_1 = 4$ // explicitly add in terms for $L_1 = 4, L_2 \geq 4$
      $c = i + L_1 + 1$
      for $L_2 = 4, j-e-2$
        $f = j - L_2 - 1$
        $s = L_1 + L_2$
        $G^{\text{partial}} = \gamma_1(s) + \gamma_2(|L_1 - L_2|) + \gamma_3(f, c, f+1, c-1)$
        $\Delta p = P^x_{i,d,e,s} \exp\{-G^{\text{partial}}/RT\}\, Q^g_{c,d,e,f}/Q^x_{i,d,e,s}$
        $P^g_{c,d,e,f}$ += Δp
        $P^x_{i,d,e,s}$ −= Δp // Remove border cases
        $Q^x_{i,d,e,s}$ −= $\exp\{-G^{\text{partial}}/RT\}\, Q^g_{c,d,e,f}$ // Remove border cases
      } (∗)
      $L_2 = 4$ // explicitly add in terms for $L_1 \geq 5, L_2 = 4$
      $f = j - L_2 - 1$
      for $L_1 = 5, d-i-2$
        $c = i + L_1 + 1$
        Insert (∗)
      // Store partial values for Q^{x2} and P^{x2}
      for $s = 10, l-9$
        $Q^{x2}_{i+1,d,e,s-2} = Q^x_{i,d,e,s}\ \exp\{-[\gamma_1(s-2) - \gamma_1(s)]/RT\}$
        $P^{x2}_{i+1,d,e,s-2} = P^x_{i,d,e,s}$

Figure C.3. Pseudocode for computing interior loop contributions to P^g in $O(N^5)$ as an alternative to the $O(N^6)$ interior loop recursion of figure 3.4.

Bibliography

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

[2] R. B. Lyngso, M. Zuker, and C. N. S. Pedersen. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, 15(6):440–445, 1999.

Appendix D

Thermodynamic Analysis of Multi-Stranded Nucleic Acid Systems

This chapter contains unpublished work that is in preparation, and should not be published online until after journal publication. The reference for this paper is:

R. M. Dirks, J. S. Bois, J. Schaeffer, E. Winfree, and N. A. Pierce. Thermodynamic analysis of multi-stranded nucleic acid systems. *In prep.*, 2005.

D.1 Introduction

Single stranded nucleic acids can adopt many different secondary structures depending on their sequence. Using an experimentally parameterized, secondary structure based energy model [9, 8, 5], minimum free energy structures, partition functions and base pairing probabilities [12, 11, 7, 15, 14, 10, 13, 6] can be determined in polynomial time (with respect to sequence length) using dynamic programming (see Chapters 2 and 3). However, the rigorous generalizations of these algorithms to multiple, potentially interacting strands have yet to be fully described, although a recent paper begins to answer some of these questions [1]. In this chapter, methods are provided that explore the thermodynamics of multiple strands, excluding the possibility of pseudoknots.

D.2 Definitions

Before details of the multi-stranded algorithms can be described, extensions to the standard energy model must be made to allow for multi-stranded secondary structures. First, some definitions and lemmas:

Definition 1 *For k strands, an* **ordering** *is a concatenation of the sequences according to some permutation, p. The first base (5' end) of the first strand in the ordering is assigned the number 1, and the last base (3' end) of the last strand is assigned N, where N is the sum of the lengths of all k strands.*

Definition 2 *For an ordering of k strands, the last base of each strand, excluding the final strand in the ordering, is defined as a* **nick** *point. An ordering of k strands will always have $k-1$ nicks.*

Definition 3 *A* **cyclic permutation** *of k strands is one that preserves the relative order of the strands when the strands are considered on a circle. For a given ordering, there are exactly k cyclic permutations if all the strands are distinct. These permutations can be generated by the process of taking the last strand and moving it to the front. The set of cyclic permutations for a given ordering forms an equivalence class.*

Definition 4 *A* **circular permutation** *of k strands is an equivalence class of cyclic permutations. There are exactly $(k-1)!$ circular permutations for k distinct strands.*

Definition 5 *For a sequence of length N, a* **secondary structure** *is a list of ordered pairs, B, such that $\forall(i,j) \in B, 1 \leq i < j \leq N$, and $\forall((i_1,j_1),(i_2,j_2)) \in B \times B$, $i_1 \neq i_2$ or j_2, $j_1 \neq i_2$ or j_2. For a multi-stranded system, a secondary structure refers to the absolute base pairing relationships, so the list B will vary with the particular ordering that is being considered.*

Definition 6 *A secondary structure is considered* **pseudoknotted** *if $\exists((i_1,j_1),(i_2,j_2)) \in B \times B$, such that $i_1 < i_2 < j_1 < j_2$. A multi-stranded secondary structure, S, is pseudoknotted if* every *possible ordering has a pseudoknotted B. A secondary structure that is not pseudoknotted is called* **non-pseudoknotted**. *If the DNA backbone is drawn as a straight line with base pairs represented as arcs above the line, then pseudoknots correspond to the intersection of two or more arcs.*

Definition 7 *Any non-pseudoknotted secondary structure can be completely decomposed into* **loops**. *A loop is created by starting at any base in the secondary structure and following the phosphate backbone in a 5' to 3' direction. Whenever a base pair is encountered, trace along the base pair rather than the backbone, and continue in the 5' to 3' direction until a closed loop is formed. In the case that a nick is encountered, simply go past it. In the case of the end of the strand being reached, continue at base 1. This process can be repeated until every segment of phosphate backbone has been assigned to exactly one loop.*

Definition 8 *A non-pseudoknotted, multi-stranded secondary structure, S, is* **connected** *if no loop contains more than one nick, where the 3' end of the last strand is considered a nick for this purpose. Alternatively, if each strand were reduced to a vertex with edges drawn between vertices when there exists at least one base pair between the corresponding strands, then this graph is connected if and only if S is connected. This second definition also works for pseudoknotted structures. A collection of k connected strands is defined as a* **complex** *of size k.*

Lemma 1 (Concatenation Lemma) *A connected, non-pseudoknotted secondary structure has exactly one circular permutation that allows the structure. Any strand permutation not in this*

cyclic class must contain a pseudoknot for this secondary structure.

The proof of the above lemma is in section D.8.

D.3 Energy Model

The energy of a connected, non-pseudoknotted secondary structure can be obtained by chosing one of the non-pseudoknotted orderings of the k strands and summing the energies of the loops as if it were a single strand, with the exception that all loops containing a single nick are treated as exterior loops. If a loop contains two nicks (counting the end of the last strand as a nick for this purpose), then the energy becomes infinite as this disconnects the structure.

Lemma 2 *For a given sequence with a connected, non-pseudoknotted secondary structure, all cyclic permutations of the strands have the same energy as computed by the multi-stranded energy model described above.*

The proof of the above lemma follows from the fact that loop types and loop energies are preserved by cyclic permutations. Cyclic permutations correspond to rotating the entire complex in space, and hence have no effect on a loop based energy model. If the energy model were to be modified in the future to include pseudoknots for example care must be taken to ensure that calculated energies remain invariant under cyclic permutations. □

As shown above, the calculated energy for a given secondary structure is independent of the particular non-pseudoknotted permutation used to represent it as a single strand. This is crucial in making the energy of a connected, non-pseudoknotted secondary structure a well-defined value.

The nearest neighbor energy model described above is based solely on the local energies of loops and consequently ignores the global structure and geometry of a secondary structure. In particular, the total number of different states available to a molecule, and hence its free energy, is affected by any rotational symmetries. For a linear, single-stranded DNA or RNA, the unique locations of the 5' and 3' ends, as well as the asymmetry of individual nucleotides, preclude any such symmetries. In contrast, complexes of multiple strands may have symmetries if sequences can be rotated onto

identical copies of themselves. The typical example of multi-strand symmetry is the perfectly self-complementary homodimer, a complex with a two-fold rotational axis. When a rotational symmetry of order Y exists, the needed correction to the entropy is $R\ln(Y)$, where R is the gas constant [2]. The free energy of the complex thus needs to be adjusted by $RT\ln(Y)$.

D.4 Symmetry

When multiple strands are concatenated to form a nicked, "single-stranded" secondary structure, it has already been shown that all cyclic permutations of a given ordering are equivalent with respect to the nearest neighbor energy model. Since it is arbitrary which strand is first, a symmetry is possible whenever a non-trivial cyclic permutation maps a strand ordering onto itself. With four copies of two strands, A and B, the strand permutation $ABABABAB$ can potentially have four-fold or two-fold symmetry. If the base pairing is such that every AB segment (or $ABAB$ subunit) behaves in an identical manner, then there must be a corresponding physical symmetry, as each AB unit (or $ABAB$ unit) would be indistinguishable from the others. If the secondary structures of the repeating subunits are not identical, then the complex would not be symmetric. As an illustration, the $AAAA$ tetramer in Figure D.1 is shown with secondary structures that imply four-fold, two-fold or no rotational symmetry.

D.5 Partition Functions

D.5.1 Distinct Strands

For the case where there are L distinct strands, no rotational symmetries are possible. To calculate the partition function for a specific permutation, a variation of the standard, single-stranded partition function algorithm (see Chapter 2 for a review) can be implemented to compute the Boltzmann-weighted average of all connected, non-pseudoknotted secondary structures.

The pseudocode in figure D.2, along with the recursion diagrams in Figure D.3, show how to determine the partition function for a specific ordering of the strands in time $O(N^4)$ where N is

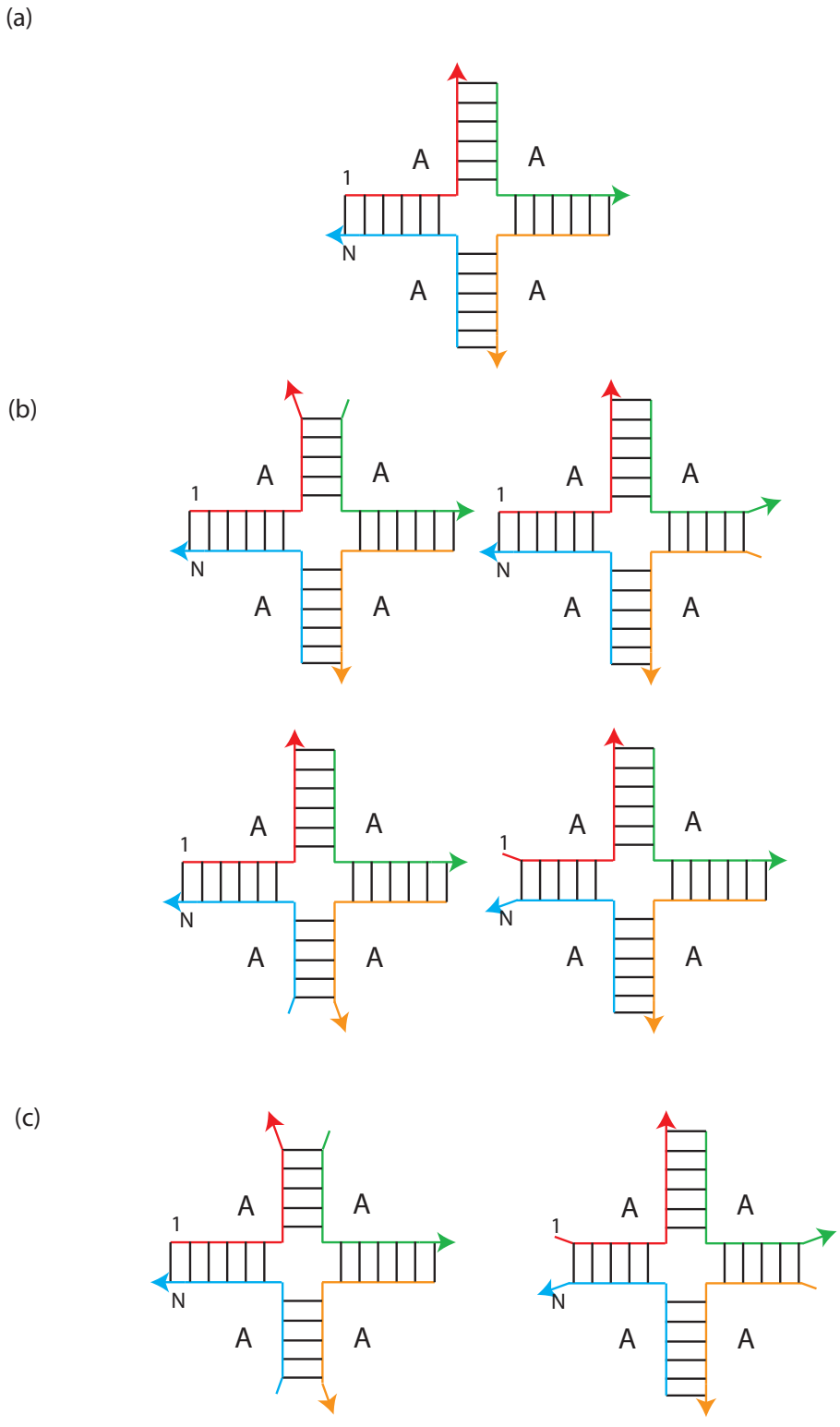


Figure D.1. **(a)** This secondary structure has a symmetry factor of 4. **(b)** These four secondary structures are rotationally equivalent, but do not have rotational symmetry . **(c)** These two rotationally equivalent secondary structures have a symmetry factor of 2.

the total sequence length of the permutation. The time complexity can be reduced to $O(N^3)$ using standard methods [4]. In the pseudocode, the following arrays have importance as partial partition functions. $\mathbf{Q_{i,j}}$ represents all possible structures on the $[i, j]$ interval that contain zero exterior nicks, where an exterior nick is a strand breakpoint that is not contained within a base pair. This includes nicks assigned to base j, but excludes nicks assigned to base $i - 1$.

$\mathbf{Q^b_{i,j}}$ represents all possible structures in the interval $[i, j]$ containing the pair $i \cdot j$.

$\mathbf{Q^m_{i,j}}$ represents all possible interiors of a multiloop that contain at least 1 base pair. $Q^m_{i,j}$ excludes all structures with an exterior nick, and in addition, is always zero when $i - 1$ is nicked.

$\mathbf{Q^1_{i,j}}$ is identical to $\mathbf{Q_{i,j}}$, but with the extra condition that there is at least one base pair in the interval. In other words, $\mathbf{Q^1_{i,j}}$ is simply $\mathbf{Q_{i,j}}$ minus the empty case.

$\mathbf{Q^{1r}_{i,j}}$ is identical in nature to $\mathbf{Q^1_{i,j}}$ but only describes the cases where j is a nick point i.e. the end of a strand. Since a nick at j must be an exterior nick, it follows that whenever $\mathbf{Q^{1r}_{i,j}}$ is non-zero, then $\mathbf{Q_{i,j}}$ must be zero. Similarly, if $Q^{1r}_{i,j}$ is zero, then $\mathbf{Q_{i,j}}$ must be non-zero.

In addition, the definition of some energy terms are given below.

$\mathbf{G^{\mathrm{HairpinMS}}_{i,j}}$ is the standard hairpin loop energy, but with a check for nicks. With no nicks, it is identical to the standard hairpin treatment. With two or more nicks, the energy is infinite. With exactly one nick, the energy is that of an exterior loop. In the case of a nicked hairpin, the restriction that $j - i \leq 4$ no longer applies.

$\mathbf{G^{\mathrm{InteriorMS}}_{i,d,e,j}}$ is analagous to $\mathbf{G^{\mathrm{HairpinMS}}_{i,j}}$, but for interior loops. If there are no nicks in [i, d-1] and no nicks in [e, j-1], then the standard interior loop term is used. With one total nick, the exterior loop energy is used, and with two or more nicks, the energy is infinite.

If we repeat this algorithm for one ordering from each circular permutation, the concatenation lemma proves that each secondary structure will be evaluated exactly once. Summing the values for these permutations will give us the partition function for the complex. Again, since every strand was assumed to be distinct, there are no global symmetry corrections needed.

D.5.2 Repeated Strands

Next, consider a complex of $n_1+n_2+n_3+...+n_k = L$ strands, where sequence i is repeated n_i times. As with the previous case, we can take one linear strand ordering for each circular permutation and compute the partition function. The Cauchy-Frobenius lemma tells us that the total number of circular permutations, and hence the number of partition function evaluations required, will be

$$\sum_{l=1}^{L} \frac{1}{l} \sum_{m=1}^{l} k^{gcd(m,l)}.$$

However, depending on the circular permutation, some secondary structures may have a global symmetry. Figure D.1 shows secondary structures of the $AAAA$ complex with either four-fold, two-fold or no rotational symmetry. Consequently, some secondary structures should have their free energy increased by $RTln(4)$, some by $RTln(2)$, and some should not be altered. Unfortunately, the dynamic program used to calculate the partition function can only use local information and consequently cannot distinguish between symmetric and asymmetric structures.

Before resolving the issue of symmetry, a second source of error worth addressing is over-counting. For an accurate partition function, it is imperative to count each possible secondary structure exactly once. This is ensured in the case of distinct strands by the concatenation theorem. However, for the case of repeated strands, over-counting may still be a factor. This is because two different secondary structures of the same linear permutation may in fact be physically indistinguishable. Figure D.1 highlights this possibility. Consequently, the partition function contribution of each secondary structure needs to be divided by the number of equivalent representations. As with the case of global symmetries, the dynamic program used to calculate the partition function is unable to distinguish which structures are correctly handled and which are not.

Although independently intractable with current methods, the problems of global symmetry and over-counting are related in such a way as to make the simultaneous correction of both possible. The following symmetry theorem illustrates this approach.

Theorem 1 *Consider a permutation, p, of L elements (strands), some of which may be identical. Let m be the smallest integer such that the first m strands can be repeated n = L/m times to produce the entire permutation. For example, $m = 2, n = 4$ for the pattern ABABABAB, whereas $m = 4, n = 2$ for ABBAABBA, and $m = 8, n = 1$ for ABABBABA. Let Ω_p be the set of all possible connected secondary structures without pseudoknots for permutation p. If the multi-stranded partition function algorithm is applied to p and yields q(p), then the corrected partition function, $q'(p)$, which properly accounts for both the reduced entropy caused by rotational symmetries and the over-counting of indistinguishable secondary structures, is precisely $q'(p) = q(p)/n$.*

The proof of the the above uses the orbit-stabilizer theorem from abstract algebra. Let G be the group of n cyclic permutations of p, acting on the set Ω_p. This is best visualized by equally spacing the L strands on the perimeter of a circle. By doing this, the elements of G correspond to physical rotations that map each strand to a strand of the same type, i.e. $A \rightarrow A, B \rightarrow B$, etc. Since m is minimal, it can be readily shown that all rotations that preserve strand identities must be in G.

Let $s \in \Omega_p$ be an arbitrary secondary structure. The stabilizer of s, denoted G_s, is the set of rotations that map s onto itself. In other words, $g \in G_s$ if and only if base pairs map to base pairs and unpaired bases map to unpaired bases. Each of the rotations $g \cdot s$ is described by the exact same secondary structure, so over-counting is not an issue with the partition function algorithm. However, $|G_s|$ corresponds to the degree of physical symmetry in the circular representation of the secondary structure. Due to the indistinguishability of strands of the same type, this symmetry in the secondary structure must correspond to a three-dimensional rotational symmetry of the same degree. Since the nearest-neighbor energy model used in the partition function calculations only applies to local energies and not global properties, the entropy of this structure must be adjusted by $\Delta S^o = -R\ ln(|G_s|)$.

The orbit of s in G, denoted $G(s)$, is the set of all algorithmically distinguishable secondary

structures $\{g \cdot s, g \in G\}$. Each element in $G(s)$ is considered a different structure and contributes exactly once to the partition function. However, even those these secondary structures have different lists of base pairs, they are actually indistinguishable because it was arbitrary which of the L strands we picked to be the first strand in p. Consequently, the partition function algorithm counts all $|G(s)|$ versions of s as distinct instead of as a single structure, so there is an over-counting by a factor of $|G(s)|$ for every $s \in \Omega_p$.

The orbit-stabilizer theorem gives a useful relationship between the sizes of the orbit and stabilizer of s. In particular, $|G_s||G(s)| = |G| = n, \forall s \in \Omega_p$.

The partition function algorithm for p calculates

$$q(p) = \sum_{s \in \Omega_p} e^{-F^o_{NN}(s)/RT}$$

where $F^o_{NN}(s)$ is the standard free energy of s as described by the nearest neighbor energy function. The corrected partition function should then be

$$\begin{aligned} q'(p) &= \sum_{s \in \Omega_p} \frac{1}{|G(s)|} e^{(-F^o_{NN}(s) - RT\, ln(|G_S|))/RT} \\ &= \sum_{s \in \Omega_p} \frac{1}{|G(s)|} e^{-F^o_{NN}(s)/RT} e^{-ln(|G_S|)} \\ &= \sum_{s \in \Omega_p} \frac{1}{|G(s)||G_s|} e^{-F^o_{NN}(s)/RT} \\ &= \frac{1}{n} \sum_{s \in \Omega_p} e^{-F^o_{NN}(s)/RT} \\ &= \frac{1}{n} q(p). \qquad \square \end{aligned}$$

Consequently, the partition function generated by the standard algorithm need only be divided by n in order to correct for both global symmetries and over-counting. Summing over each circular permutation, and dividing each by the appropriate n will give the partition function for a complex with repeated strands.

D.6 Equilibrium Concentrations

Now that the proper partition function can be calculated for any complex of k types of strands, regardless of strand repetition, it becomes possible to calculate the equilibrium concentration of important species, given initial concentrations. Although complexes of arbitrarily large size are possible, considering all of these is computationally intractable. As an approximation, assume that the largest complex is of size L. Define the set, $C_{k,L}$, to be all possible complexes of L or fewer strands. Assuming that their are at least L copies of each of the k strands, it follows that

$$|C_{k,L}| = \begin{pmatrix} L+k \\ \\ k \end{pmatrix} - 1.$$

To solve for the equilibrium concentrations of these complexes, the partition function for each of these complexes needs to be calculated as described in the previous sections. Any permutation independent corrections to the free energy, such as a penalty for the number strands in a complex, can also be incorporated at this time. If the total concentrations of the k different strand types are specified, a system of equations can be solved for the distribution of complexes that minimizes the free energy. By the second law of thermodynamics, the minimum solution is equivalent to the equilibrium distribution of complexes. The precise method for setting up and solving the equilibrium equations is described elsewhere [3].

D.7 Pair Probabilities

After calculating a single-stranded partition function, an additional algorithm can be used to calculate the probability that any two bases are paired. For the multi-stranded case where there is no periodic subunit in the strand permutation, these same values can be calculated using the general method described in Chapter 3. Other useful properties such as $n(s^*)$, the average number of bases that deviate from secondary structure s^*, can also be determined directly (see Chapters 3 and 4). However, when a strand permutation has a periodic symmetry of order $n > 1$, these

algorithms require modification. In particular, every pair $i \cdot j$ belongs to a family of n, physically indistinguishable pairs. For example, when three identical strands of length ten are considered, pairs $\{i \cdot j, i + 10 \cdot j + 10, i + 20 \cdot j + 20\} (\text{mod } 30)$ are equivalent. Consequently, the notion of pair probabilities should be broadened to pair-family probabilities. Unfortunately, since more than one pair from a family could occur in the same secondary structure, an accurate determination of pair-family probabilities would require joint knowledge of multiple pairs simultaneously, something that is not possible with the standard, dynamic programming algorithm.

Rather than solve for joint-probabilities, a similar, yet more tractable value is the expected number of pairs from a given family that is present in a random, Boltzmann-weighted secondary structure. This value can range from 0 to n. By the linearity of expectation, we can separately determine the average occurrence of each pair in the family and add them together to get the final answer. Since a pair-probability is equivalent to the expected number of times a specific pair will occur (assuming we can distinguish all positions), these expected values can be determined using the standard pair-probability algorithm. When $n = 1$, the expected value of any pair $i \cdot j$ is simply the pair-probability, $P_{i,j}$. For $n > 1$, the symmetry theorem states that the partition function contribution of every secondary structure visited by the algorithm should be scaled by a factor of $1/n$. If we temporarily take the algorithm's perspective that all the positions are distinguishable, then it is clear the standard pair-probability algorithm will give the correct expected values for each pair, since the probability of every structure is unaffected by this constant scaling[1]. Since all positions are not distinguishable, however, the physically meaningful value of interest is the sum of the expected values for all pairs in a family. By symmetry, this is equivalent to n times the calculated expected value of any one member of the pair-family.

The calculation of $n(s^*)$ is straightforward once the expected value of every pair-family is known. Define l_m to be the length of the smallest periodic subunit for a strand permutation with total length N. For a secondary structure s^*, define a target matrix S^* such that $S^*_{i,j}$ is the number of pairs in the target structure from the $i \cdot j$ class. Augment this matrix with an $N+1^{st}$ column such that every

[1] By the same argument, equilibrium sampling is also valid if physically equivalent structures are pooled together.

row adds to the symmetry factor, n. The value of $S^*_{i,N+1}$ represents the unpaired state for the n positions equivalent to i. The average number of bases in the incorrect state can then be calculated by

$$n(s^*) \equiv \frac{1}{2} \sum_{1 \le i \le l_m} \sum_{1 \le j \le N+1} |S^*_{i,j} - E_{i,j}|$$

where $E_{i,j}$ is the expected value of pair-family $i \cdot j$, as described above. The one-half is required because every base is penalized for both failing to be in the correct state, as well as being in the incorrect state. It can be shown that this is equivalent to the definition of $n(s^*)$ given in Chapter 4 for the case of $n = 1$. To see this, recall that when $n = 1$, $E_{i,j} = P_{i,j}$, $\sum_{1 \le j \le N+1} P_{i,j} = 1$ and $\forall i, S^*_{i,j} = 1$ for exactly one value of j, call it j'_i, and is zero otherwise. It then follows that

$$\begin{aligned}
n(s^*) &= \frac{1}{2} \sum_{1 \le i \le N} \sum_{1 \le j \le N+1} |S^*_{i,j} - E_{i,j}| \\
&= \frac{1}{2} \sum_{1 \le i \le N} \sum_{1 \le j \le N+1} |S^*_{i,j} - P_{i,j}| \\
&= \frac{1}{2} \sum_{1 \le i \le N} \left[1 - P_{i,j'_i} + \sum_{j \ne j'_i} P_{i,j} \right] \\
&= \frac{1}{2} \sum_{1 \le i \le N} (1 - P_{i,j'_i}) + (1 - P_{i,j'_i}) \\
&= \sum_{1 \le i \le N} 1 - P_{i,j'_i} \\
&= N - \sum_{1 \le i \le N} P_{i,j'_i} \\
&= N - \sum_{1 \le i \le N} \sum_{1 \le j \le N+1} P_{i,j} S^*_{i,j}.
\end{aligned}$$

The value of $n(s^*)$ for a complex can also be determined by properly averaging the values of $n(s^*)$ from each circular permutation, based on their corrected partition functions.

D.8 Proof of Lemma

Lemma 1 states:

A connected, non-pseudoknotted secondary structure has exactly one circular permutation that allows the structure. Any strand permutation not in this cyclic class must contain a pseudoknot for this secondary structure.

To prove this, we start with a few observations:

Observation 1 For any secondary structure, if base pairs are removed and lemma 1 can be shown to be true for this reduced structure, then the lemma must also hold for the original secondary structure.

Observation 2 In attempting to prove lemma 1, all pairs within the same strand can be removed. In addition, all helices, including bulges and interior loops, can be reduced to a single base pair. If lemma 1 holds for all such reduced structures, then the lemma must be true for all structures.

Observation 3 Since cyclic permutations of a structure can neither create nor remove pseudoknots, any structure can be cyclically permuted without affecting the veracity of lemma 1.

Observation 4 If two strands are identical, then this will reduce the number of possible circular permutations. Since the lemma states there is at most 1 circular permutation that allows the structure, reducing the available permutations cannot invalidate the lemma. Therefore, for the sake of the proof, we can assume the strands are distinct.

The proof of lemma 1 will proceed by induction on the number of strands L in a complex. If L is one or two, then there is only a single circular permutation and the lemma is true by default.

Assume the lemma is true for all positive integers $l < L$. We will show that it must be true for L. Consider an arbitrary connected, non-pseudknotted secondary structure on L strands. Remove all intra-strand base pairs and reduce all helices (and interior loops and bulges) to a single base pair. Imagine each strand as a vertex of a graph, with the base pairs as edges connecting the nodes. Reduce this graph to a tree by systematically removing an arbitrary edge (base pair) from every cycle. These reductions will yield a minimal, connected, non-pseudoknotted secondary structure.

Since this reduced secondary structure is equivalent to a tree, there must be at least two leaves (strands with exactly one base pair). Perform a cyclic permutation to move one of the leaves to the first position. Since removing a leaf from a tree cannot disconnect the graph, removing the first strand and its base pair must result in a connected, non-pseudoknotted structure on $L-1$ strands. By the inductive hypothesis, there is at most one circular permutation that is not pseudoknotted. Since the current strand ordering is connected and non-pseudoknotted, this must be the only circular permutation possible. All of the $L-1$ cyclic permutations are legal and equivalent, so there is no need to consider these cases separately.

Number the strands from left to right as 1 to $L-1$, with position i, base i_n being the strand and base to which the removed leaf was base paired. Consider the possible locations where the removed leaf and its base pair can be added back to the graph. Clearly, inserting it before position 1 is valid because this simply recreates the original, non-pseduoknotted permutation. The strand could also be added after $L-1$ because this is cyclically equivalent to inserting it at the first position. If we can show that all other locations would cause a pseudoknot, then this will prove that there is exactly one circular permutation that allows the original secondary structure.

Since adding the strand before position 1 is legal, there can be no base pair directly connecting strands 1 through $i-1$ to any base to the right of i_n. If there were such a pair, this would have intersected the removed base pair, causing a pseudoknot in the original structure. Consequently, if the removed strand were added between positions j and $j+1$, with $1 \leq j \leq i-1$, then strands 1 to j must be disconnected from strands $j+1$ and greater. This is because strands 1 to j cannot pair to the added strand (the added strand is a leaf whose pair is already used), nor to any base between the added strand and i_n (this would form a pseudoknot with the added base pair), nor to any base right of i_n (see the argument above). By the inductive hypothesis the secondary structure must be connected. Thus, no such insertion between positions 1 and $i-1$ is possible.

In addition, since adding the strand after position $L-1$ is also legal, a symmetric argument shows that insertion between j and $j+1$ with $i \leq j \leq L-2$ is also impossible. Therefore, the only legal insertion points are at the beginning and ends of the strand ordering, verifying the lemma for

size L. By induction, this lemma must hold for all L. $\square$

```
function multiFunc( s_1, s_2, ..., s_L) //this function will return the partition function for L strands concatenated
//in the order they are input. The total length of |s_1| + |s_2| + ... + |s_L| = N.
Initialize (Q, Q^b, Q^m, Q^1, Q^1r) // 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^HairpinMS_{i,j}/RT}
    for d = i+1, j-2 // loop over all possible rightmost pairs d·e
      for e = d+1, j-1
        Q^b_{i,j} += exp{-G^InteriorMS_{i,d,e,j}/RT} Q^b_{d,e}
        //Next treat nicked multiloops
        nicksEJ1 = # of nicks >= e and <= j - 1
        //Case 1: Single nick is to the right of d·e pair
        if (nicksEJ1 == 1) and (i not nicked)
          Q^b_{i,j} += Q^1_{i+1,d-1} Q^b_{d,e}
        //Case 2: Single nick is in [i, d - 1]
        if nicksEJ1 == 0 and (i is nick)
          //Consider i as the single nick "in" the multiloop
          //Any other nicks in this interval must therefore be enclosed in a pair
          Q^b_{i,j} += Q^1_{i+1,d-1} Q^b_{d,e}
        if nicksEJ1 == 0 and (i not a nick)
          //If i not nicked, then any other nick in [i, d - 1] can be the single nick
          foreach k ∈ [i + 1, d - 1]|k is a nick
            leftN0 = 1 //number of empty structures in [i + 1, k]
            rightN0 = 1 //number of empty structures in [k + 1, d - 1]
            if ∃ nick in [i, k] then leftN0 = 0
            if ∃ nick in [k + 1, d - 1] then rightN0 = 0
            //Case 2a: No pairs left of k, at least 1 to the right of k
            Q^b_{i,j} += leftN0 · Q^1_{k+1,d-1} Q^b_{d,e}
            //Case 2b: At least 1 pair left of k, no pairs to the right of k
            Q^b_{i,j} += Q^1r_{i+1,k} · rightN0 · Q^b_{d,e}
            //Case 2c: At least 1 pair on both sides of k
            Q^b_{i,j} += Q^1r_{i+1,k} Q^1_{k+1,d-1} Q^b_{d,e}
          //Consider regular multiloop
          Q^b_{i,j} += Q^m_{i+1,d-1} Q^b_{d,e} exp{-[α_1 + 2α_2 + α_3(j-e-1)]/RT}
    // Q, Q^m, Q^1, Q^1r recursions
    //Empty recursion
    Q_{i,j} = 1
    if ∃ nick on [i, j] then Q_{i,j} = 0
    for d = i, j-1 // loop over all possible rightmost pairs d·e
      for e = d+1, j
        if no nicks on [e, j - 1] and j is nick
          Q^1r_{i,j} += Q_{i,d-1} Q^b_{d,e}
        if no nicks on [e, j]
          Q_{i,j} += Q_{i,d-1} Q^b_{d,e}
          Q^1_{i,j} += Q_{i,d-1} Q^b_{d,e}
          //single pair in Q^m
          if no nicks on (i - 1, d - 1)
            Q^m_{i,j} += exp{-[α_2 + α_3(d-i) + α_3(j-e)]/RT} Q^b_{d,e}
          //more than 1 pair in Q^m
          Q^m_{i,j} += Q^m_{i,d-1} Q^b_{d,e} exp{-[α_2 + α_3(j - e)]/RT}
//Partition function for this ordering of strands is Q_{1,N}
return Q_{1,N}
```

Figure D.2. Pseudocode for the partition function evaluation of a single permutation of L strands.

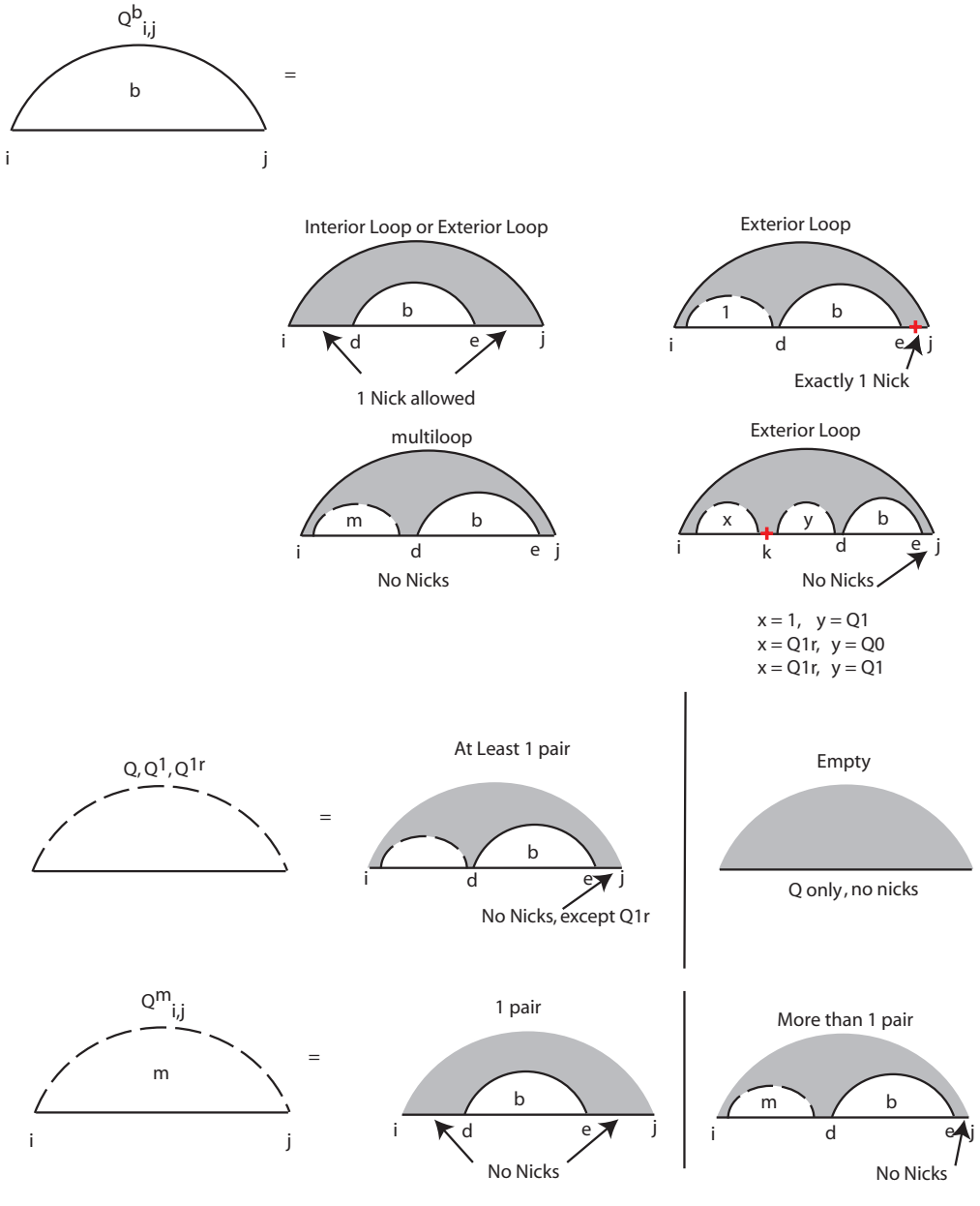


Figure D.3. Recursion diagrams for the $O(N^4)$ partition function calculation for a connected permutation.

Bibliography

[1] M. Andronescu, Z. C. Zhang, and A. Condon. Secondary structure prediction of interacting RNA molecules. *Journal Of Molecular Biology*, 345(5):987–1001, 2005.

[2] V. A. Bloomfield, D. M. Crothers, and I. Tinoco Jr. *Nucleic Acids: Structures, Properties, and Functions*. University Science Books, Sausalito, CA, 2000.

[3] R. M. Dirks, J. S. Bois, J. Schaeffer, E. Winfree, and N. A. Pierce. Thermodynamic analysis of multi-stranded nucleic acid systems. *In prep.*, 2005.

[4] R. B. Lyngso, M. Zuker, and C. N. S. Pedersen. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, 15(6):440–445, 1999.

[5] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940, 1999.

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

[7] R. Nussinov, J. R. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algorithms for loop matchings. *SIAM Journal of Applied Mathematics*, 35:68–82, 1978.

[8] J. Santalucia Jr. Improved nearest-neighbor parameters for predicting DNA duplex stability. *Biochemistry*, 35:3555–3562, 1996.

[9] I. Tinoco Jr., O. C. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[10] D. H. Turner, N. Sugimoto, and S. M. Freier. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17:167–192, 1988.

[11] M. S. Waterman. Secondary structure of single-stranded nucleic acids. In *Studies in foundations and combinatorics: Advances in Mathematics Supplemental Studies*, volume 1, pages 167–212. Academic Press, New York, 1978.

[12] M. S. Waterman and T. F. Smith. RNA secondary structure: a complete mathematical analysis. *Mathematical Biosciences*, 42:257–266, 1978.

[13] M. Zuker. Calculating nucleic acid secondary structure. *Current Opinion in Structural Biology*, 10:303–310, 2000.

[14] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, 1984.

[15] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–147, 1981.

Appendix E

Extra Figures for DNA Design

The work presented here is heavily based on the supplementary material for:

R. M. Dirks, M. Lin, E. Winfree, and N. A. Pierce, *Paradigms for computational nucleic acid design.* Nucleic Acids Research, 2004. **32**(4): p. 1392-1403.

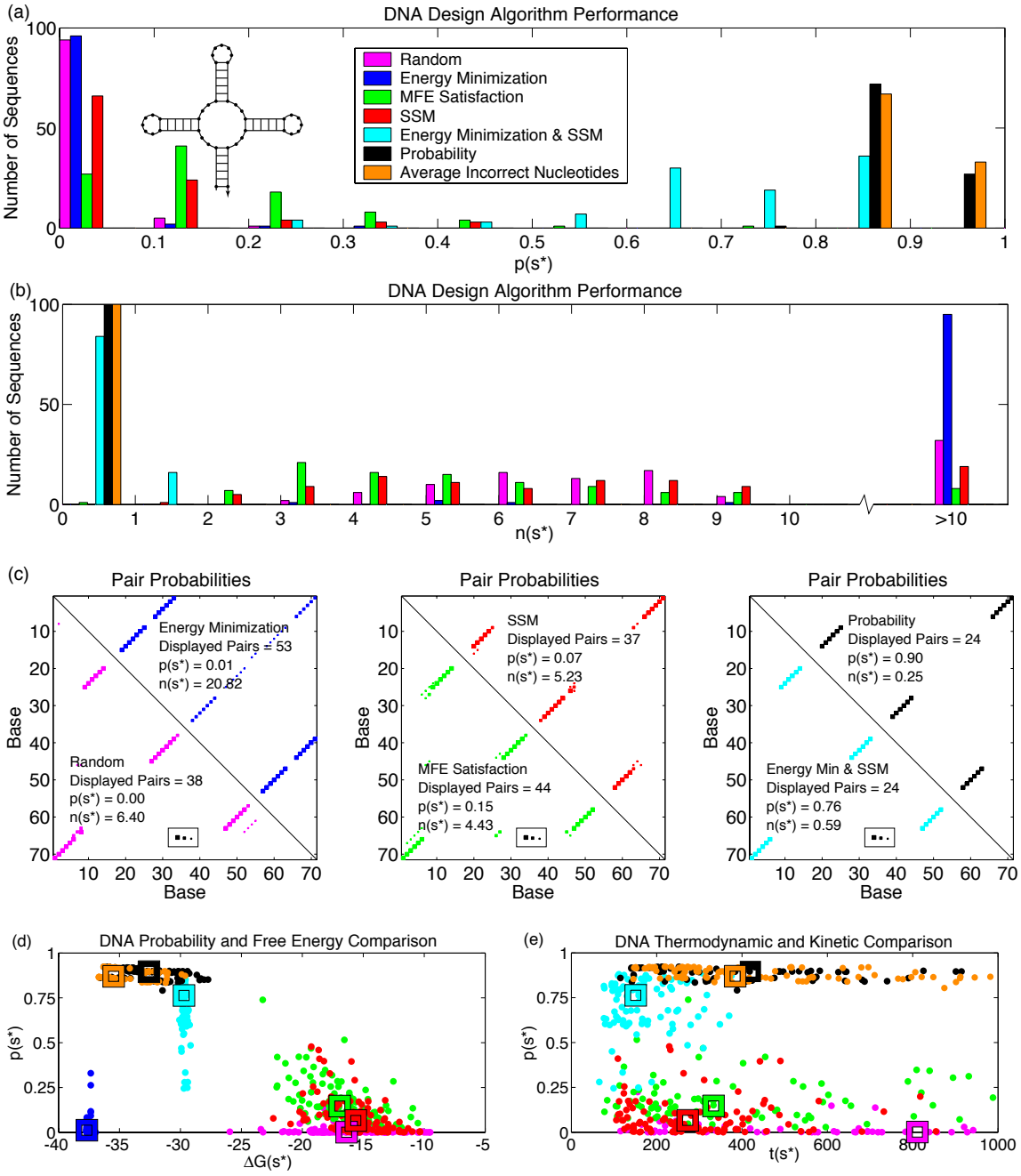


Figure E.1. DNA multiloop: See caption for Figure 4.2.

Table E.1. Sequence designs for DNA multiloop designs of Figure E.1.

Design Method	$p(s^*)$	$n(s^*)$	CG content	Entropy
Random	0.00	8.27	0.50	1.00
Energy Minimization	0.01	23.67	0.78	0.32
MFE Satisfaction	0.15	5.31	0.50	0.98
SSM	0.07	7.06	0.50	0.99
Energy Min & SSM	0.76	0.58	0.65	0.72
Probability	0.90	0.34	0.63	0.48
Average Incorrect Nucleotides	0.87	0.28	0.68	0.46

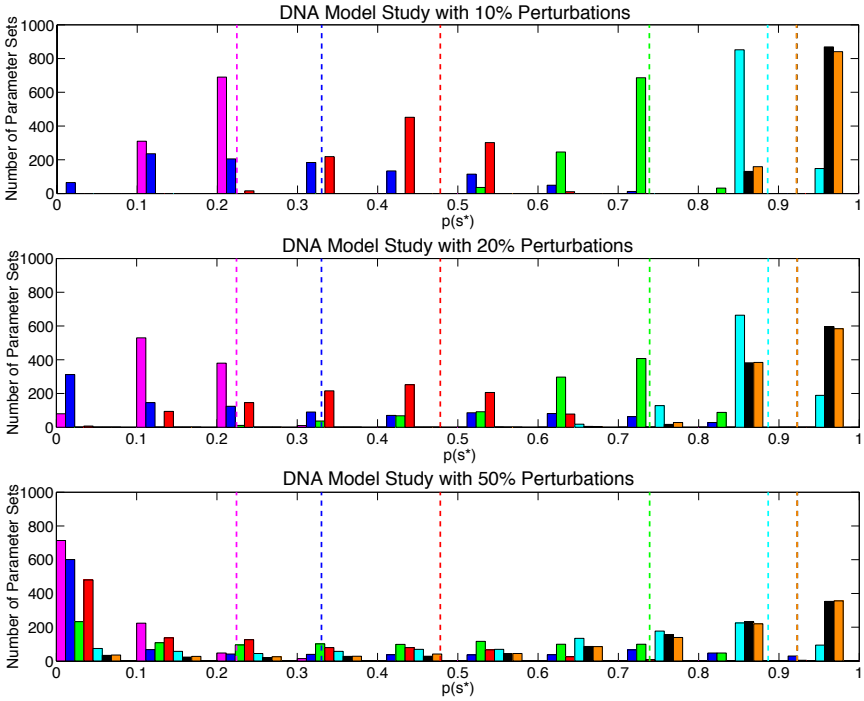


Figure E.2. DNA model perturbation study. See caption for Figure 4.3.

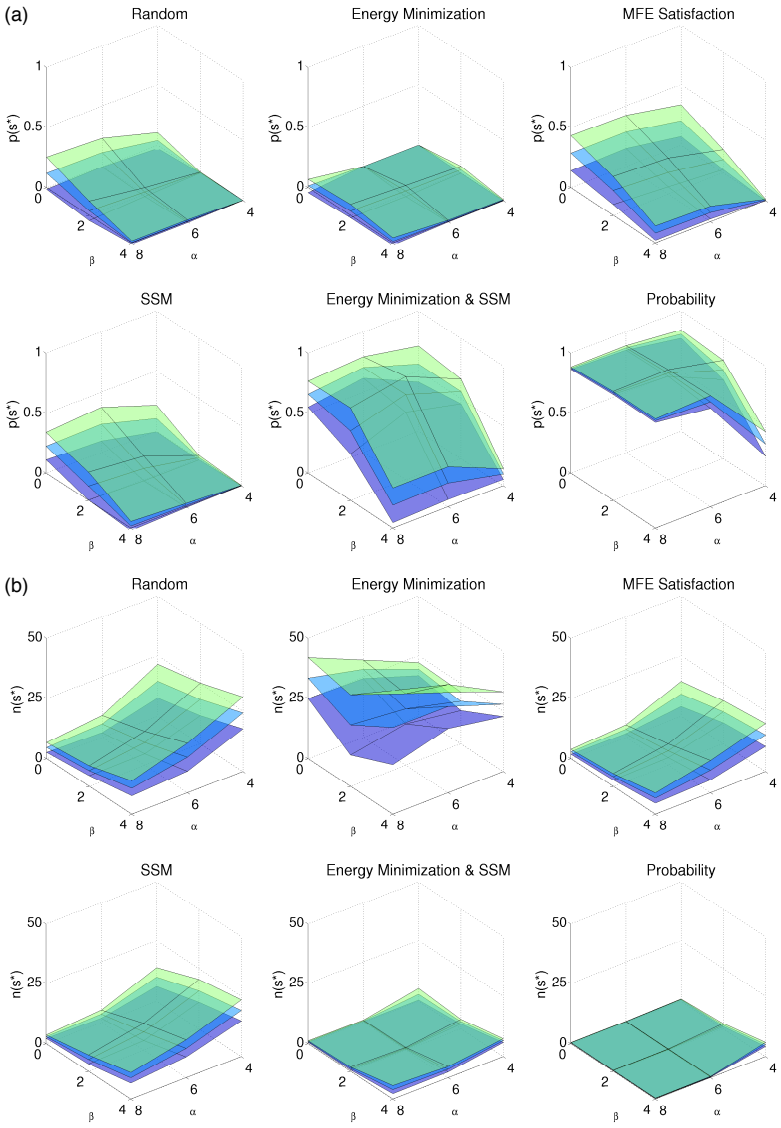


Figure E.3. DNA Multiloop Variations: See caption for Figure 4.4.

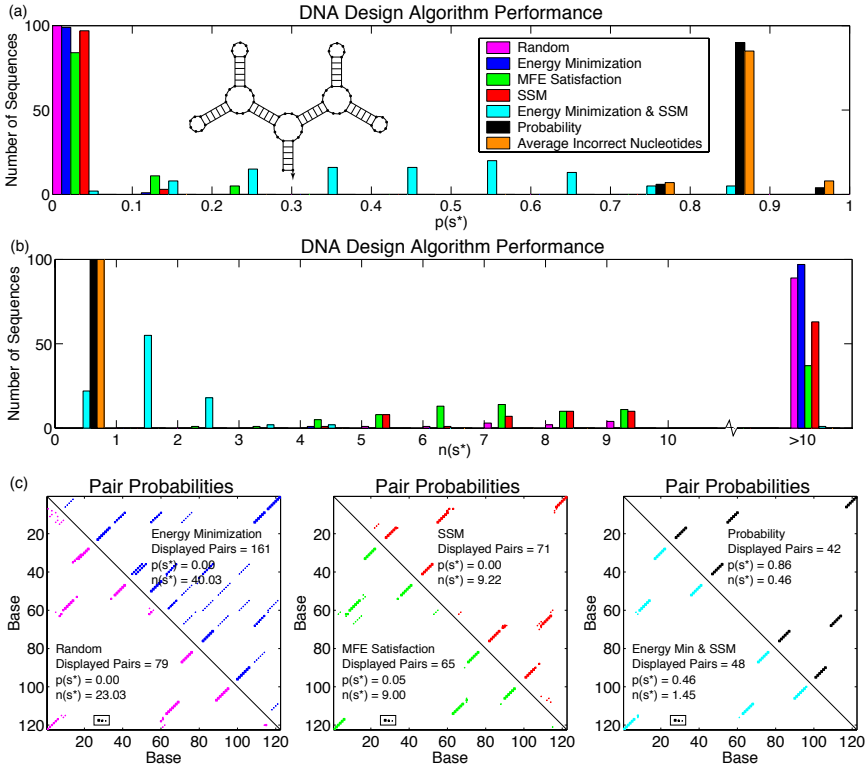


Figure E.4. Large DNA multiloop: See captions for Figures 4.2abc.

Appendix F

HCR Variations

The following variations to HCR are currently being investigated and may appear in future publications.

The original version of HCR exhibits linear growth in response to an initiator. With slightly more sophisticated systems, higher order growth can be achieved. This, in turn, could lead to more sensitive biosensors. To generate quadratic growth, molecules such as Q1 and Q2 (Figure F.1) could be used in conjunction with H1 and H2. Here, an initiator Qi binds to Q1 and a three-way branch migration exposes a sticky end comprising f-e*-b*. This single-stranded region then binds to Q2, exposing e*-d*-b*-c. The e*-d* can act as an initiator for another copy of Q1, thereby creating a linear system with periodic, single-stranded stretches of b*-c. These periodic regions can then act as an initiators for the H1-H2 system, resulting in structures similar to branched polymers. Figure F.1 shows the Q1-Q2 main chain (drawn as a blue and red dashed line) with H2-H1 polymers (orange-green lines) branching off at each Q2.

To achieve pseudo-exponential growth, the same Q1 and Q2 molecules can be used with E1 and E2 rather than H1 and H2. Like Q1 and Q2, E1 and E2 form a linear chain in the presence of an initiator, in this case c-b*. This initiator sequence is identical to the single-stranded stretches produced by Q1-Q2. Furthermore, E1-E2 itself generates periodic stretches of e*-d*, an initiator for the Q1-Q2 system. Consequently, if Q1, Q2, E1, and E2 were mixed together and an initiator were added (either c*-b* or e*-d*), each branch of the original polymer would itself be a branched polymer. The result of this would be exponential growth, similar to a dendrimer. However, since three-dimensional space can only grow cubicly, exponential growth could not be sustained, and the rate of growth would eventually slow to at most cubic.

An alternative way to increase the sensitivity of HCR is to increase the loop length. However, if the loops are too big, the hairpins tend to form large complexes in the absence of initiator. One possible way to avoid this is to break the loops into two parts, and have two sticky ends on the hairpins (see Figure F.2). This also has the advantage of "locking" each hairpin in the open state when the initiator is bound, making it more difficult for the hairpins to close and knock the initiator off. Due to the geometry, the two sticky ends of H1 will have difficulty binding simultaneously to the hairpin loop of H2, making the species less susceptible to direct strand invasion than traditional large loops.

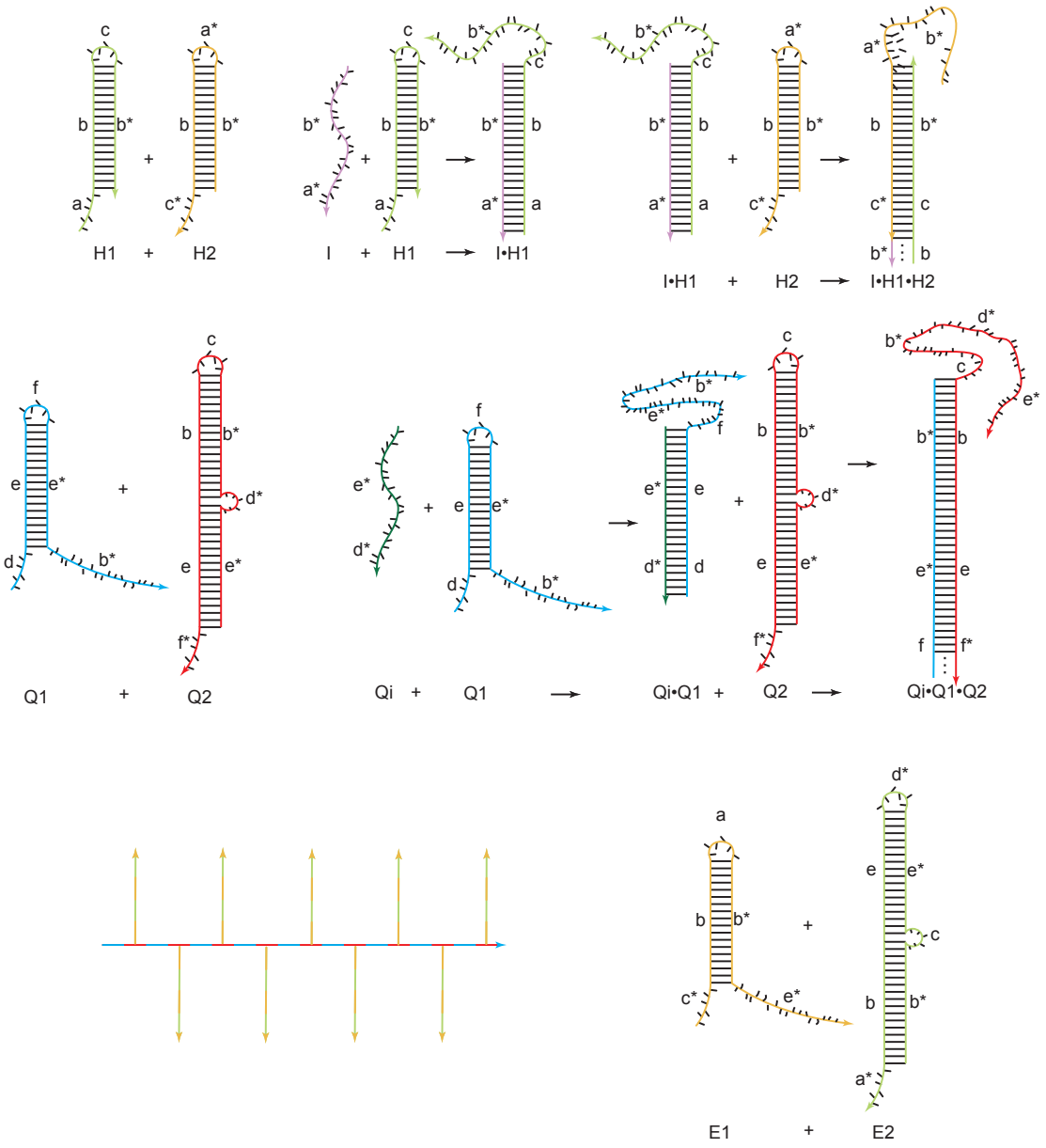


Figure F.1. Schematics for quadratic and pseudo-exponential HCR. Descriptions of the systems are in the text.

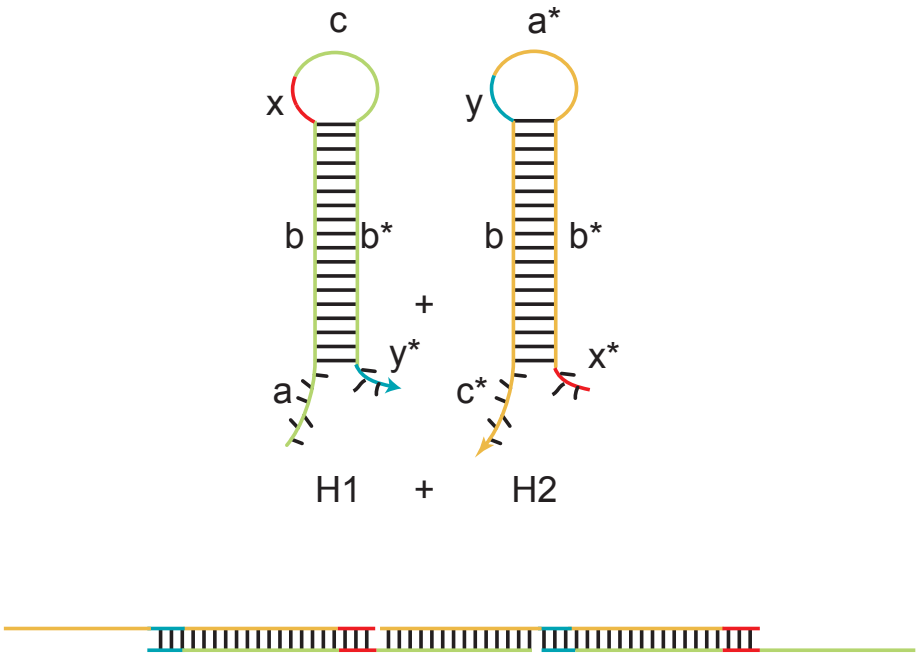


Figure F.2. An initiator for this system would consist of x*-b*-a* and would bind to the sticky end of H1, displace the stem, and invade part of the hairpin. The new sticky end, c-b*-y*, could then do the same to H2. The bottom image shows how two H1 and two H2 strands could interact.