Fast Approaches to Designing RNA Molecules

Viann W. Chan
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4
vchan@cs.ubc.ca

Daniel Archambault
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4
archam@cs.ubc.ca

Peter Huang
9845 Cameron Street
Burnaby, B.C. V3J 1M3
peterhuang4@yahoo.com

Barry Po
Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1Z4
po@cs.ubc.ca

Abstract

Can we generate an RNA sequence that will fold up into a given structure? We propose several approaches that can be applied to solving this problem: probabilistic modeling using sequence frequencies, self-loop fold-avoidance, sub-structure energy negation, and graph-based approaches.

1 Introduction

As our collective knowledge of genetics and molecular biology continues to expand, exciting innovations in genetic engineering are quickly becoming possible. The improvement of disease diagnoses, the design of individualized drugs, a better understanding of cellular pathways, and the synthesis of life are only a few examples of the advances that are currently being made. A thorough understanding of the nature of ribonucleic acid (RNA) molecules has, and continues to be, a central component of many such research thrusts. By applying our understanding of the properties of RNA molecular sequences and their natural configurations, RNA molecules continue to provide a significant source of inspiration for advances in the biological sciences.

The symbiosis of biology and computer technology through the area of bioinformatics now makes it feasible for us to examine complex theoretical and physical models of RNA molecules. With computational models of RNA, we may begin to explore the complex relationship between RNA molecular sequences and their physical structures. In particular, we may be interested in the question of how we might generate artificially coherent RNA molecular sequences. The answer to this question may hold the promise for even more scientific advances in the future. As our contribution, we seek to rigorously define and to provide insight into one particular aspect of this question: given a specific secondary structural configuration, is it possible for us to design an appropriate RNA sequence that matches the given structure?

This paper summarizes the results of our efforts and is organized into eight major sections. The first section is made up of this introduction. The second section provides some general background into the nature of RNA molecules and secondary structure. The third section provides a solid foundation for the problem that we attempt to address, citing other similar work that precedes our own research. The fourth section exhaustively describes the algorithmic models that we have chosen to pursue and implement. The fifth section offers a complete statistical examination and evaluation of our models and their performance. The sixth section addresses the fundamental limitations of our approach, describing the effects of the computational assumptions that we made. The seventh section outlines possible future research that might be pursued. Finally, the eighth section provides some concluding thoughts.

2 RNA Sequences and Secondary Structure

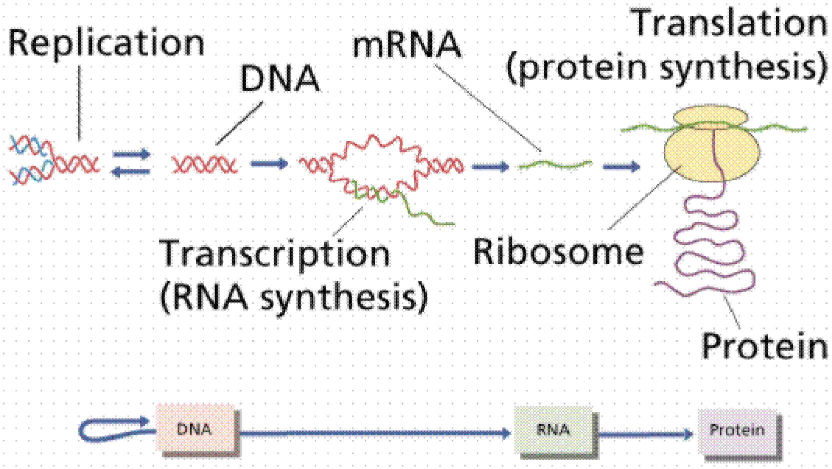

Figure 1: The central dogma of molecular biology.

Figure 1 above outlines the central dogma of molecular biology. Informally, the central dogma states that deoxyribonucleic acid (DNA) molecules, which contain all of the information that is necessary to make proteins, undergo a process of transcription, whereby given DNA molecules are used as templates for the synthesis of RNA molecules. In a subsequent process, RNA molecules undergo a process of translation, whereby given RNA molecules are used to build the amino acid sequences that make up proteins. In this way, RNA serves as an intermediate structure through which strands of DNA are made to correspond to final protein products.

The presence of RNA molecules is critical to the central dogma because RNA can explain a great deal about existing organisms' evolutionary history and organization. RNA molecules are thought to protect DNA molecules, keeping DNA molecules safe from the potentially dangerous reactions inherent in the cytoplasm of eukaryotic cells. It is also believed that genetic information can be amplified simply by having many copies of RNA produced from a single copy of DNA. Furthermore, it is accepted that the regulation of gene expression can be affected by having specific ”controls” at each element of the pathway between DNA and proteins. By introducing RNA into the process of synthesizing proteins from DNA, there are more natural opportunities to control the expression of particular genes under different circumstances.

Chemically speaking, RNA molecules are polymer structures of varying length that consist of a primary sugar-phosphate backbone to which sequences of basic nucleotide units are attached. Each nucleotide unit in an RNA molecule is made up of a base, a ribose sugar, and a phosphate. There are four essential kinds of nucleotide units - adenine (A), uracil (U), cytosine (C), and guanine (G). Since these nucleotide units are connected to the backbone (by forming phosphodiester linkages) via named 5' oxygen and 3' oxygen molecules, the sequences of nucleotides that make up a particular molecule of RNA are frequently denoted as a character string that are ”read” from a designated 5' end of the molecule to a corresponding 3' end.

Particular RNA molecules are often categorized into one of the four basic kinds of RNA that are generated by organic cells. Messenger RNA (mRNA) molecules are duplicate copies of specific genes that act to carry the information stored by a DNA molecule in the nucleus of a cell to the cytoplasm of a cell. Transfer RNA (tRNA) molecules are small RNA structures that have very specific sub-structures that allow them to bind one amino acid onto one end of a given tRNA, and an mRNA molecule onto the other end, thereby acting as an adaptor for carrying specific amino acids to appropriate locations for protein binding. Ribosomal RNA (rRNA) molecules define one of the complementary structural components for ribosomes, which are construction molecules from which other molecules can be built. Finally, small nuclear RNA (snRNA) molecules are those molecules that are involved in processing other kinds of RNA as they travel between the nucleus and cytoplasm of a given cell.

Physically, RNA molecules exist as single-stranded entities with various local hairpin, bulge, and loop sub-structures that are the result of nucleotide complementation between certain pairs of nucleotide bases (U-A and G-C). A ”wobble” base pair (G-U) complementation is also possible under certain circumstances. Figure 2 below outlines some of the more common sub-structures that are possible.

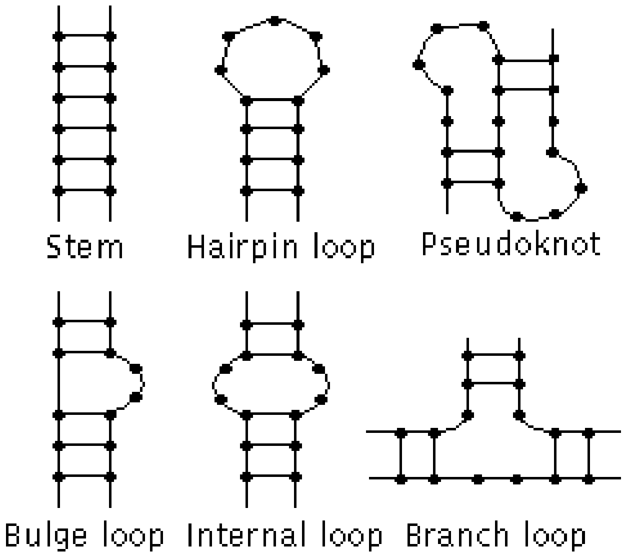

Figure 2: Possible sub-structures in RNA secondary structure.

For a given RNA molecule, the collection of hairpin, bulge, and loop sub-structures that are the result of base pair complementation define the characteristic *secondary structure* of that particular molecule. Because of the nature of nucleotide base pair complementation, hairpins, bulges, and loops do not necessarily occur uniformly throughout a particular RNA molecule. As such, RNA molecules are said to fold into given secondary structures.

In order to appropriately express the physical nature of RNA secondary structure, we must be able to formalize our expression of arbitrary secondary structure geometry. Throughout this paper, we adopt the convention of representing secondary structures as sequential strings that consist of ’(’, ’.’, ’)’ characters. For simplicity, we assume that any secondary structures will be defined from the 5’ to the 3’ of an RNA molecule, unless otherwise stated.

Geometry	Description
(	base pairs with another base downstream
.	base is unpaired
)	base pairs with another base upstream

It is very important to note that only specific sequences of nucleotides permit an RNA molecule to fold into a specific secondary structure. That is, it is not possible for any random RNA sequence to fold into any random secondary structure. Instead, both RNA sequences and secondary structures must coincide such that base pair complementation occurs at the appropriate locations in a specified secondary structure, and such that there is sufficient energy across all base pairings in order to allow given sequences to sustain particular shapes.

Because RNA molecules are not restricted to any particular geometric organization, RNA molecules are seen to form more globally into three-dimensional super-structures that are known as tertiary structures. Figures 3 and 4 below provide examples of tRNA secondary and tertiary structure.

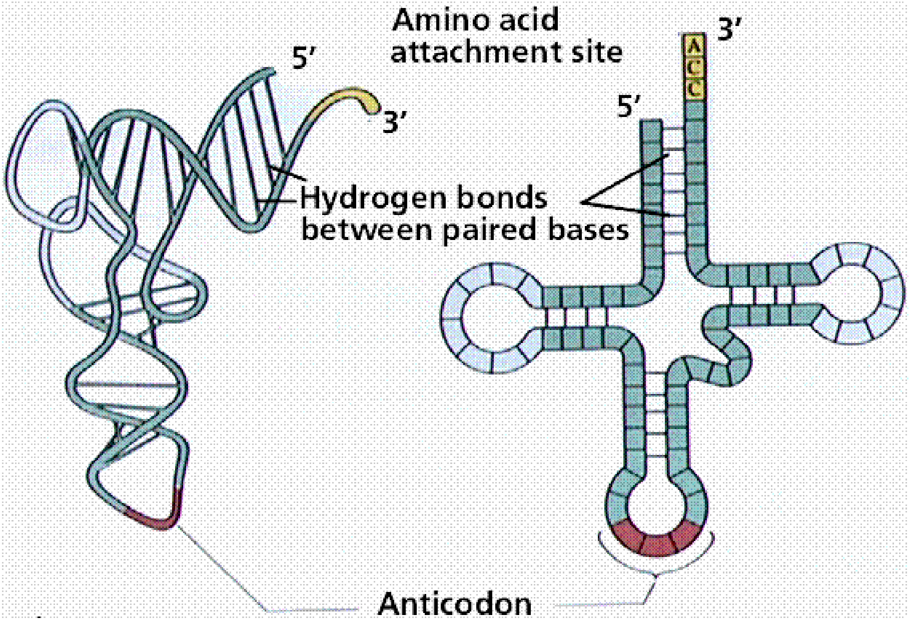
Amino acid attachment site
5′
3′
3′
A
C
C
5′
Hydrogen bonds between paired bases
Anticodon

Figure 3: An example tRNA and its secondary structure.

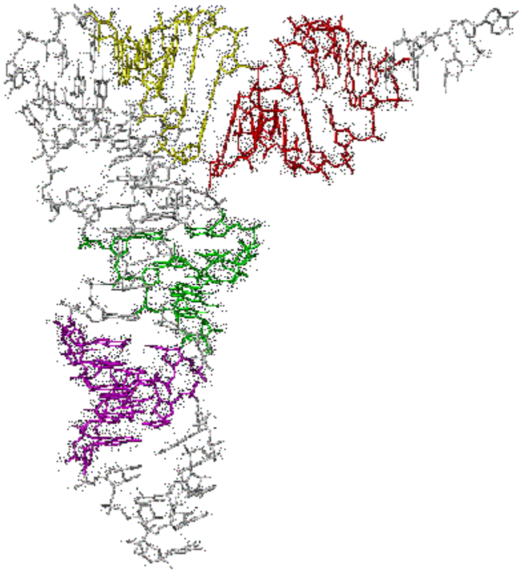

Figure 4: An example tRNA and its tertiary structure.

3 Designing RNA Sequences from Secondary Structure

A traditional and well-known problem in the field of bioinformatics is the question of how we might go about predicting the secondary structure of an RNA molecule, given that we have an RNA sequence as input. This particular problem is well-studied, and a plethora of software tools and theoretical models for RNA secondary structure prediction are readily available [6][11]. For example, a widely known and well-respected software tool for RNA secondary structure prediction is the Vienna RNA folding package, which provides tools for the generation, conversion, prediction, and visualization of RNA molecules [5]. In this light, it seems strange that there is comparatively little research on the inverse problem: given an RNA secondary structure as input, how might we go about designing a possible RNA sequence that folds into the input secondary structure?

In relation to this inverse problem, Flamm, Hofacker, Maurer-Stroh, Stadler and Zehl [3] have examined the issue of designing general RNA sequences that are stable in multiple secondary structure configurations. In their work, they view the generation of artificial RNA sequences as a formal combinatorial problem for which a set of heuristics can be developed. Their heuristics are based on the formulation of RNA structures as combinatorial sets that define ”RNA switches” that model particular RNA-related processes. By maintaining the stability of individual switches, they claim that it becomes possible to generate appropriate conformations of RNA secondary structure to nucleotide sequences. Thus, through the use of heuristics that manipulate the switch formulations in a manner that seeks to maximize stability, they believe that it is possible to design stable RNA sequences with reasonably good accuracy.

Along similar lines, Friede [4] has studied the more specific problem of creating a computational model for artificially designing sequences that match the secondary structures of tRNA molecules. In his methodology, he applies techniques that attempt to minimize the values of sets of chemical thermodynamic equations in an attempt to create RNA sequences that maximize the amount of free energy in a conformation of a derived tRNA sequence and a given tRNA secondary structure. His work is of particular interest because he claims that many of the properties that are assumed in his work on tRNA sequence design can be readily applied to the design of RNA molecules in general. Furthermore, his work is inherently unique because it includes ”wet lab” results in the form of actually synthesizing artificially generated tRNA molecules through in vitro transcription techniques.

In light of such recent work, we felt that our own examination of the RNA sequence design problem should be chosen to complement existing work. Taking cues from [3], we wanted to build a collection of heuristics that could be applied generally to as many RNA secondary structure configurations as possible. Similarly, taking cues from [4], we wanted to ensure that our heuristics were based on physically acceptable models of RNA that took into account the considerations of chemical processes, thermodynamics, and physical stability as much as possible. Because of the importance of operational efficiency, we also believed that the examination of fast techniques was warranted. Thus, we rigorously define the problem that we set out to solve as the development and comparison of fast heuristics for designing RNA sequences.

4 Algorithms

In the following sub-sections, we describe the collection of approaches and implementations that we investigated.

4.1 Probabilistic Model Using Sequence Frequencies

Our first approach was based on using the frequencies of nucleotide base occurences in existing RNA sequences in order to design RNA sequences with an appropriate proportion of nucleotide bases. By identifying particular sub-structures in the input secondary structure, we hoped to be able to construct plausible RNA sequences.

RNA sequences were partitioned into loop regions ”.” and stem regions ”(”.

Let base $b \in \{A, U, C, G\}$. Let base pair $bp \in \{AU, UA, CG, GC\}$. Let N be any base.

Let base $b_i \in \{A, U, C, G\}$ be the base at position i.

Let $g_i \in$ { (, .,) } be the geometry character pattern at position i.

Let B_i be a random variable. $B_i = b$ is the event of generating base b at position i of the sequence.

Let BP_{ij} be a random variable. $BP_{ij} = bp$ is the event of generating a base pair bp at position i of the sequence (where we assign the first base of the pair to position i and assign the second base of the pair to position j).

Let $count_{pattern,region}$ be the number of $pattern$s found in the given $region$, where the $pattern$ is a string of bases (”A”, ”U”, ”C”, ”G”), and $region$ is a string of geometry characters (”(”,”.”,”)”).

All structures were parsed using a stack, and sequences were generated in $O(n)$ time.

The following sub-sub-sections describe how we obtained the probabilities that we used for assigning nucleotide bases to specific positions in a particular sequence.

4.1.1 Random Base Probabilities

$$P_{loop}(B_i = b) = \frac{1}{4} \tag{1}$$

$$P_{stem}(BP_{ij} = bp) = \frac{1}{4} \tag{2}$$

4.1.2 Knudsen Base Probabilities

Knudsen and Hein [7] obtained single base probabilities from a larger empirical data set. We incorporated their probabilities in our experiments.

Comparison of Knudsen versus Single Base Frequency (percentage)

Stem	Knudsen	Single	Loop	Knudsen	Single	Overall	Knudsen	Single
AU/UA	35.6	33	A	36.4	35	A	26.8	25
GC/CG	53.4	67	C	15.1	17	C	21.4	24
UG/GU	9.8	-	G	21.2	23	G	26.7	29.5
other	1.2	-	U	27.3	25	U	25.1	21.5

Though we had a smaller training data set size (60) than the Knudsen sample size (305), our base frequencies are comparable, with the exception of an over representation of G's and slight under representation of U's.

4.1.3 Single Base Probabilities

Some additional base probabilities were obtained in the same manner as the Knudsen paper, except that these probabilities were drawn from our own data set.

$$P_{loop}(B_i = b) = \frac{count_{b,'.'}}{\sum_{b_j=A,U,C,G} count_{b_j,'.'}} \tag{3}$$

$$P_{stem}(BP_{ij} = bp) = \frac{count_{b,'('}}{\sum_{b_k=A,U,C,G} count_{b_k,'('}} \tag{4}$$

4.1.4 Base Conditional (Double) Probabilities

In double probabilities, the current base was generated by looking at the previous base and associated geometry character.

$$P_{loop}(B_i = b|b_{i-1}, g_{i-1}, g_i =' .') = \frac{count_{b_{i-1}b_i,g_{i-1}g_i}}{\sum_{b_j=A,U,C,G} count_{b_j,g_{i-1}g_i}} \tag{5}$$

$$P_{stem}(BP_{ij} = bp|b_{i-1}, g_{i-1}, g_i =' (') = \frac{count_{b_{i-1}b_i,g_{i-1}g_i}}{\sum_{b_k=A,U,C,G} count_{b_{i-1}b_k,g_{i-1}g_i}} \tag{6}$$

4.2 Base Conditional (Simplified Triple) Probabilities

In simplified triple probabilities, the current base was generated by looking at the previous two geometry characters.

$$P_{loop}(B_i = b|g_{i-2}, g_{i-1}, g_i =' .') = \frac{count_{NNb,g_{i-2}g_{i-1}g_i}}{\sum_{b_j=A,U,C,G} count_{NNb_j,g_{i-2}g_{i-1}g_i}} \tag{7}$$

$$P_{stem}(BP_{ij} = bp|g_{i-2}, g_{i-1}, g_i =' (') = \frac{count_{b,g_{i-2}g_{i-1}g_i}}{\sum_{b_k=A,U,C,G} count_{b_k,g_{i-2}g_{i-1}g_i}} \tag{8}$$

4.3 Base Conditional (Triple) Probabilities

In triple probabilities, the current base was generated by looking at the previous two geometry characters and associated bases and the current geometry character.

$$P_{loop}(B_i = b | g_{i-2}, g_{i-1}, g_i =' .', b_{i-2}, b_{i-1}) = \frac{count_{b_{i-2}b_{i-1}b_i, g_{i-2}g_{i-1}g_i}}{\sum_{b_j=A,U,C,G} count_{b_{i-2}b_{i-1}b_j, g_{i-2}g_{i-1}g_i}} \tag{9}$$

$$P_{stem}(BP_{ij} = bp | g_{i-2}, g_{i-1}, g_i =' (', b_{i-2}, b_{i-1}) = \frac{count_{b_{i-2}b_{i-1}b_i, g_{i-2}g_{i-1}g_i}}{\sum_{b_k=A,U,C,G} count_{b_{i-2}b_{i-1}b_k, g_{i-2}g_{i-1}g_i}} \tag{10}$$

4.4 Self-Loop Fold-Avoidance

Some RNA structures have long loop sequences. In randomly generated sequences, we observed that sub-sequences within a loop will fold with another sub-sequence within the loop.

We start with any sequence generated from the previous sub-section. This heuristic essentially examines each loop in the input secondary structure, searching for the reverse complement of each triplet in the loop. If a reverse complement is found, the loop sequence associated with the match is modified and the search is repeated from the beginning.

Suppose there are L loops, where loop i is of length l_i. This approach takes time $\Omega(L * l_i^2)$. Its expected running time is $Pr(\text{reverse-complement is in } i)^{-1} * L * l_i^2$. Since most loops are relatively short, this approach runs quickly in practice.

4.5 Sub-Structure Energy Negation

The goal of this approach is not only to drive an overall RNA molecule toward negative energy, but to drive every sub-structure contained within that molecule toward negative energy in the fastest way possible. Although it is not the case that sub-structures must be negative in order to fold into a particular secondary structure, negative energy ensures that a given sub-structure can fold into the assigned sequence. This is similar to the work of Seeman[9], who parses large molecules into smaller sub-structures in order to create novel secondary structures.

In this approach, we parse the geometry string into sub-structures by finding the outermost pair of balanced brackets and parsing our way inwards. For example:

..((((.))).(((..(((..)))))))

would have the following sub-structures,

..**(**((.))).(((..(((..))))))**)** (1)

..(**(**((.))**)**.**(**((..(((..)))))**)**) (2)

..((((.))).(((..**(**((..))**)**)))) (3)

Each sub-structure in this example is denoted in boldface brackets. We can define the concept of a sub-structure level by noting what brackets are read at a particular stage of the inward parsing. It is important to note that sub-structures can contain other sub-structures (1, 3) and many independent sub-structures can exist on the same level (2).

Once the parse of the geometric string is obtained, the algorithm proceeds to use a probability heuristic in order to assign a preliminary sequence to the secondary structure. Then, this approach searches for the deepest sub-structure level by iteratively parsing inwards. Optimizations are performed on each sub-structure independently at each level in order to heuristically drive the preliminary sequence toward the desired secondary structure.

For a given sequence, we define the following optimizations:

1. Define a flank region as the set of unpaired bases that are immediately adjacent to a stem. Then, flank regions of each sub-structure are checked so that they do not pair with parts of a stem in order to prevent sliding.
2. For loops larger than six bases, we perform a self-folding loop check. We use the fact that it has been empirically observed that stems of size larger than three have a tendency to fold. The goal of this optimization is to ensure that there is no stem larger than three bases within a loop sequence. We accomplish this by executing the self-folding loop check described in section 4.4.
3. For each sub-structure that is larger than forty bases, we take loop regions and slide them against associated stem regions in order to ensure that subsequences of at least length three do not pair with other parts of the sequence. If such pairings occur, bases are mutated as in (2) in order to disrupt the pairing.
4. If the flank regions are of length at least five, we perform the same check as in (2) in order to ensure that they do not pair with other regions of the sequence.

After the optimization process is complete, we use the Vienna Package [5] in order to ensure that each sub-structure maintains negative energy. If all sub-structures are of negative energy at a particular level, we proceed outward to the next deepest level and repeat the optimization process. If the energy of any sub-structure is positive, we change more of the loop bases to As and A-U to C-G pairs in the stems of that sub-structure. If we ever run into a situation where we cannot achieve negative energy by any of our optimization techniques, this approach simply gives up on the generated sequence and returns its findings.

4.6 Graph-Based Approaches

Given a structure, we can examine its sub-structure and try to find sequences that will fold into the sub-structure. We would like to combine these sequences to produce a longer sequence that will fold into the original structure.

An alternative perspective is to consider the following. Suppose we are given a simple structure and several sequences that fold into that structure. We would like develop a more complicated structure by merging the sequences.

For example, we can try to extend

(((...))).....(((...))) (i)

into

(((...))).....(((...))).....(((...))) (ii)

by deriving a set of rules that combine known sequences that fold into (i) to a longer sequence that folds into (ii).

We can think of this approach as building up a chain.

In this approach we made the following simplifying assumptions: loop regions only contain ’A’s, all stems are of length 3, every stem must begin and end with either a ’C’ or a ’G’, no pseudoknots, and no GU/UG pairs are allowed.

We can create a directed graph of possible sequences. Each node represents a sequence that corresponds to a stem. Each edge represents an ordered pair of stems that fold into (i). We can represent the graph using an adjacency matrix. Eg. If GACAAAGUCAAAAACCCAAAGGG folds into (((...))).....(((...))), then (GACAAAGUC, CCCAAAGGG) form a directed edge.

5 Results

We collected 120 RNA sequences from an online small-RNA database [10], and the NCBI [8]. Secondary structures for these RNA sequences were obtained by forward folding (i.e. predicting the secondary structure of) these sequences with the Vienna RNA package [5].

From our set of 120 RNA sequences, 60 sequences were randomly selected as a training set from which we obtained estimates for the frequency of individual nucleotide bases.

We tested the probabilistic model, self-loop fold-avoidance, and sub-structure energy negation approaches in two experiments. We chose 5 moderate length strands (around 200bp) for experiment 1, and 5 short strands (less than 80bp) for experiment 2. We ran 50 trials per strand. We have included more information about these strands in the Appendix, section 9.1.

A scoring scheme was devised in order to provide a clear metric of the precision and accuracy afforded by each individual approach.

We also performed a small case study of our graph-based approach.

Experiment 1: 50 trials per strand

Strand	Length	Source	Structure
1	189	S. mytilis	(((((((((((.........(((.....(((.((((....................)))).))) ......(((((((...((....))...))))))).)))))))))))))).((((.(..... ......((((...))))(((...(((((...)))))...)))...).))))............
2	203	P. caudatum	(.((((..((((((.....((((..(..(....)..)..))))..)))).............. ..........((((...((..(....((....))......)..))....)))).))..))))).. (((..((((.((((.(((((((................)))))))......))))))))..)))..........
3	201	P. multimicron.	((((((((..(((....(((.((((......))))((((..........)))).)))..... .......(((..((.((((.((((((.....))))))....)))).))..))).....)))..))))))))..(((((((((((.(((((((((.............))))))))).....))))))))))).........
4	205	P. primaurelia	(((((((....(((.....((((((.(((....))).))))))....)))........... ....((..(((.(((((((((.....))))))...))).)))..)).......... ..)))))))..((((((((..((((((((((...............))))))....)))). .)))))))).........
5	205	P. tetraurelia	(((((((....(((.....((((...(((....)))...))))....)))........... ....((..(((.(((((((((.....))))))...))).)))..)).......... ...)))))))..((((((((..((((((((((...............))))))....))))..)))))))).........

Experiment 2: 50 trials per strand

Strand	Length	Source	Structure
1	64	D. radiodurans	((((........)))).((((((.....)))))).....(((.(..((.....))...).))).
2	35	E. coli	.....((((.(....).))))..(((.....))).
3	77	E. coli	(((.(((...............((...(((((.......)))))...)).((((.......)))).))).)))....
4	78	X. laevis	.(((.(((..(((...(((((.......(((((.......)))))........)))))..))).....))).)))..
5	71	K. lactis	..(((((...((.....)).......(((((.(((........))))))))....)))))...........

5.1 Scoring

We computed scores in the following manner. We obtained the structure of the original sequence and the structure of each generated sequence with the RNA folder from the Vienna RNA package [5]. Using dynamic programming, we aligned the structure of the generated sequence with the structure of the original sequence. Matches were scored with +1, gaps were scored with -1, and mismatches were scored with 0.

This is a reasonable scoring scheme since we are trying to measure structure similiarity. It accounts for the length of matches in loop and stem regions. It gives a high score to similar structures and low scores to dissimilar structures. For example, if our original structure contained a hairpin (with 4 free bases), and our generated structure contained a hairpin (with 5 free bases), both structures still form a hairpin, and would result in a high similarity score.

Another possible scoring metric might be to count the number of positions where the base pairs with its intended complement. However, unlike our chosen method, this scheme may inadvertently give low scores to similar structures.

5.2 Probabilistic Model Using Base Frequencies

We obtained the score percentage for each sequence by dividing the alignment score with the total sequence length. As such, a scoring value of 1 indicates a perfect match.

From the histograms, we can see that modeling the base frequences increases the percentage of correct alignment between the generated structure and the original structure. The Knudsen probability model slightly out-performed the rest since it produced some perfect scoring strands. The conditional probability models performed slightly worse then the Knudsen and Single probability models, since their average score is slightly lower. This is expected since conditional probability models need proportionally more data to perform equally well to the single and Knudsen probability models.

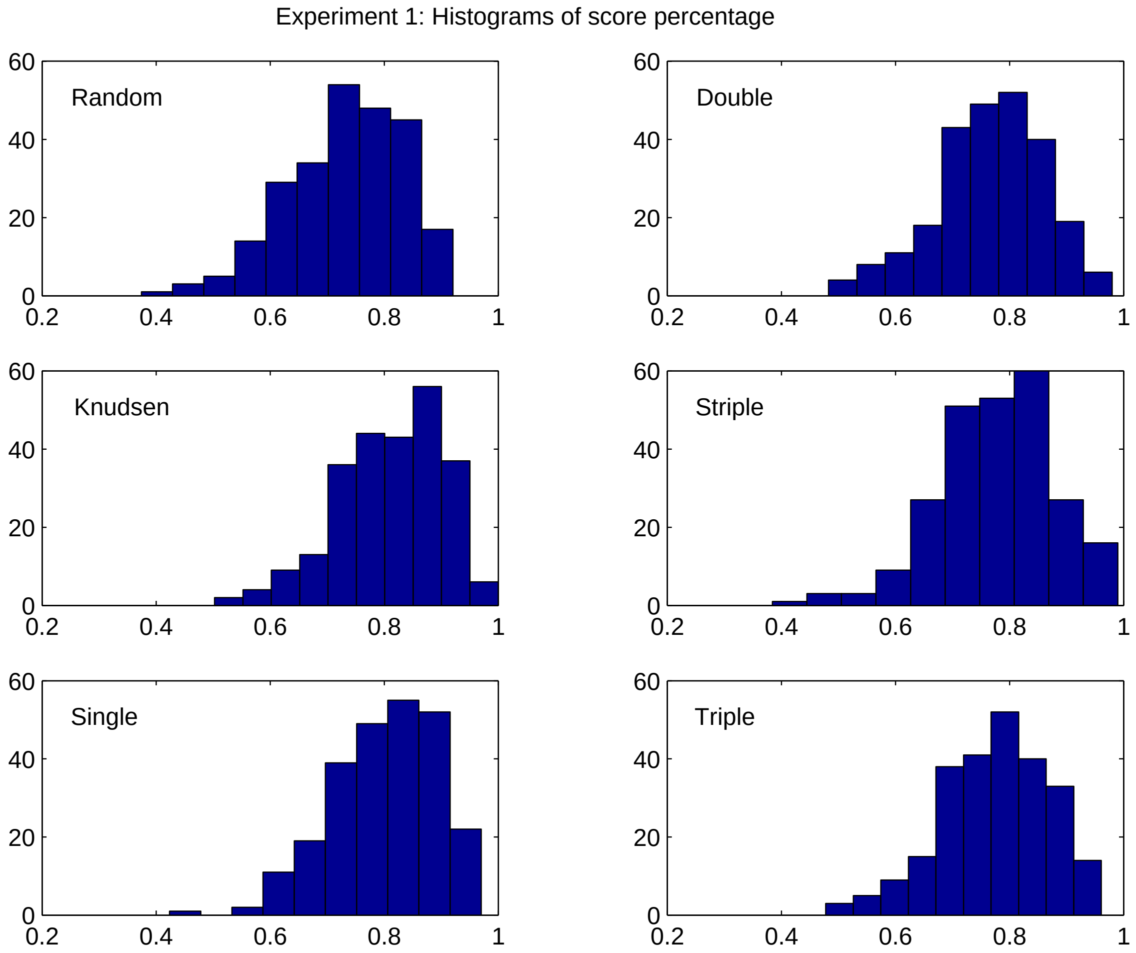

Figure 5: This shows the distribution of correct scores under each probability distribution from experiment 1.

Our heuristic performs well with short sequences. Each probability model created sequences that would fold into the correct structure. The single probablility model predicted more correct sequences than each of the other probability models.

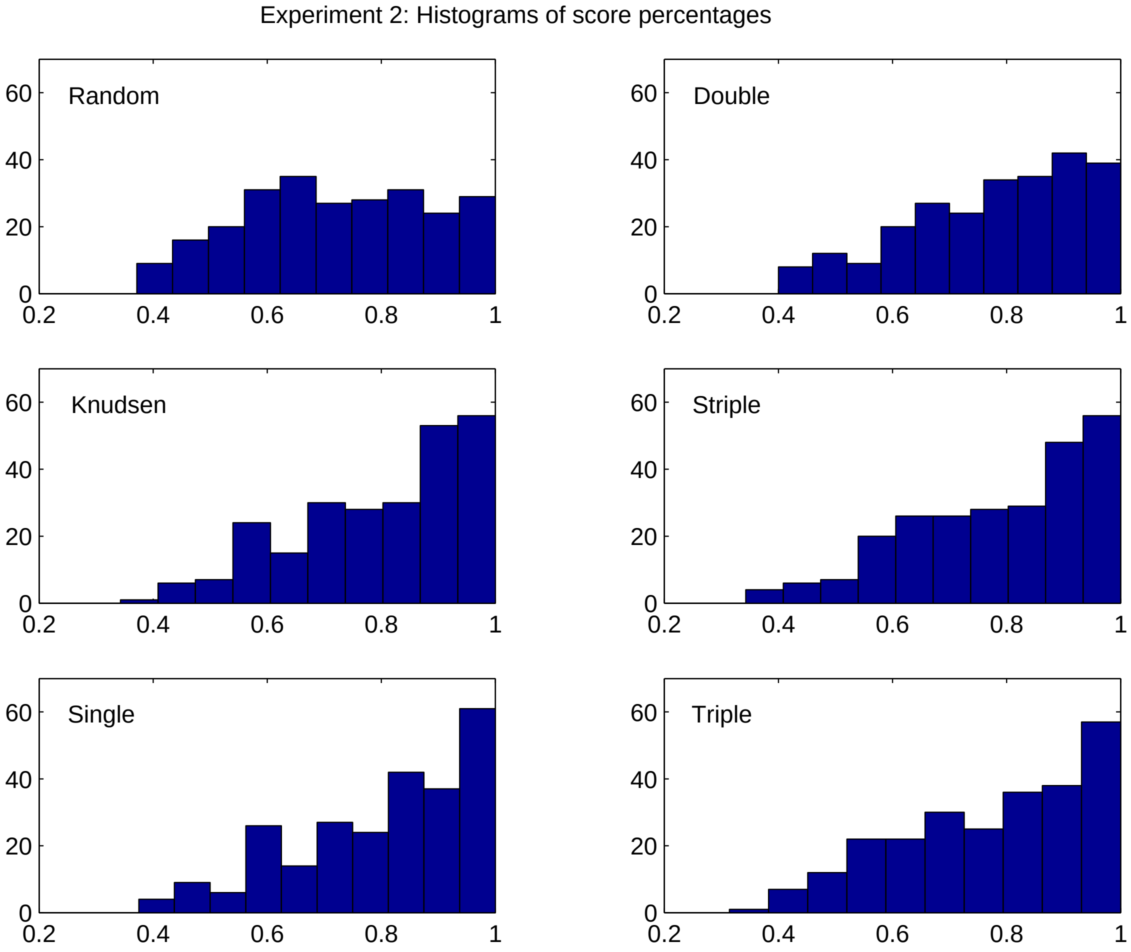

Figure 6: This shows the distribution of correct scores under each probability distribution from experiment 2.

We have included histograms based on strand by strand results in the Appendix, section 9.2.

5.3 Self-Loop Fold-Avoidance

In the following two scatterplots, each point $p_i = (x_i, y_i)$, represents the ratio of no-self-loop-check score versus self-loop-check score. eg. x_i is the score percentage of our sequence before performing a self-loop check, and y_i is the score percentage of our sequence after performing a self-loop check.

For each experiment there were 5 strands, 6 probabilistic models, and 50 trials. There are 1500 points in each of our plots.

In shorter strands, we expected that there would be more short loops, meaning that generally, the heuristic would not make as many changes to these sequences. This is verified by examining the scatter plot of experiment two. We observe more points along the line y = x, indicating no changes in scoring values.

It is not immediately obvious whether or not this heuristic will help to improve scores since most of the scores appear to be concentrated close to the diagonal. Sometimes applying the self-loop check improves the score and sometimes it becomes worse.

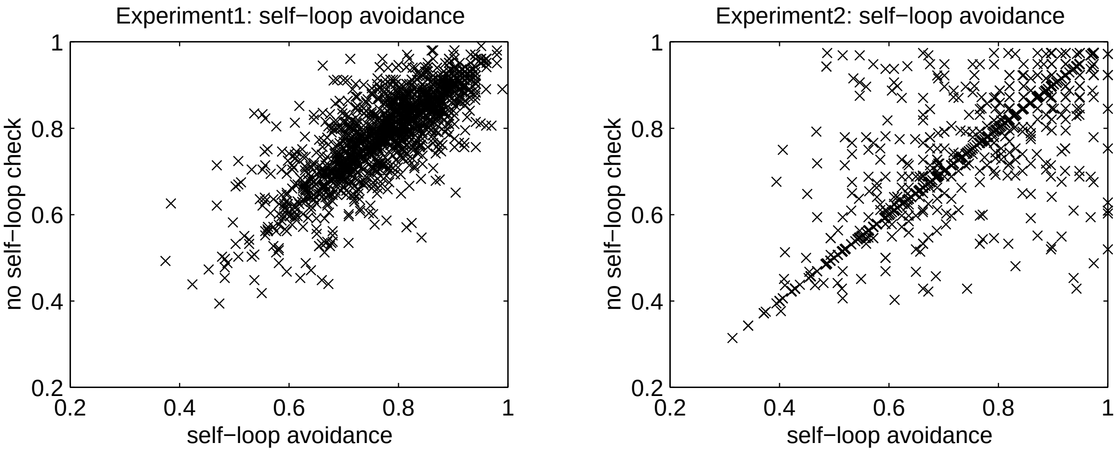

Figure 7: Plot of correct scores with self-loop check vs. no self-loop check.

5.4 Graph-Based Approach

There are 16 unique stem sequences for one hairpin (under our simplifying assumption). This gives us $16 * 16 = 256$ possible sequences that potentially fold into structure our substructure containing two hairpins. We used the RNAfolder from the Vienna package[5] to fold the 256 sequences into the structure listed in the following table (structures listed from 5’ to 3’):

	Structure	Correctly folds into structure
1	(((...))).....(((...)))	59
2	(((....))).....(((...)))	90
3	(((...))).....(((....)))	155
4	(((....))).....(((....)))	233

For our case study, we tried to extend structure 1 containing two hairpins into (((...))).....(((...))).....(((...))) (a structure containing three hairpins). We will refer to the hairpins in order from 5’ to 3’. (eg. hairpin 1 is the left-most hairpin closest to the 5’ end of the strand.)

Attempt 1: we created a graph from the 59 sequences the folded into structure 1. We explored our graph (finding all possible paths through our graph) and predicted that 285 sequences would fold into our target structure.

Suppose hairpin 1 has sequence $< x >$, hairpin 2 has sequence $< y >$ and hairpin 3 has sequence $< z >$. When we examined the incorrect sequences and found that the sequence $< x >< y >< x >$ (corresponding to hairpin 1, hairpin 2, and hairpin 3 respectively,) did not occur. In order to account for this, we created rule one: do not generate $< x >< y >< x >$ sequences.

In a similar examination of the missing sequences, we found that it did not matter what the second hairpin was (as long as it has degree > 0 in our graph). As such, we devised rule two: add additional edges to the graph. We incorporated this by creating nodes (corresponding to a sequence for a hairpin) for each hairpin, and adding an edge to the nodes corresponding to the second hairpin from each of the nodes corresponding to the first hairpin.

Attempt 2: we created a graph from our 59 sequences and two additional rules. Then we explored the graph to predict sequences that fold into our target structure.

The following table show the results of our two attempts. We compared our predicted results to the 748 sequences that actually folded into the target structure.

	Number of sequences generated	Correct	Incorrect	Missing
Attempt 1	285	255	30	493
Attempt 2	531	524	7	224
total	748			

’Correct’ - number of generated sequences that correctly folded into target; ’Incorrect’ - number of generated sequences that did not fold into target; and ’Missing’ - number of sequences from the set of 256 we did not generate, but was found to fold into target. The total was obtained by folding all 4096 sequences and counting the number of structures that matched our target.

We also tried to extend structure 4 to (((....))).....(((....))).....(((....))) with the two rules and obtained the following results:

	Number of sequences generated	Correct	Incorrect	Missing
Test 2	3270	3270	0	204
total	3474			

Thus, we came to realize that this approach might be generalized to accurately find varying tRNA structures. Since hairpins with free bases of lengths 3, 4, and 5 have special properties, we found that we might be able to apply brute force techniques in order to develop similar graph structures for other inputs. Using these structures, we could conceivably create long chains of hairpins. For example, there are 27 structures that contain two hairpin sub-structures, where each hairpin has 3, 4, or 5 free bases and where each has a fixed number of bases between the two hairpins. It would be feasible to create and store a graph for each of these structures. It would be interesting to see if this approach can be generalized to other simple structures.

6 Limitations

Our approaches distinctly rely on heuristics that attempt to drive the overall energy of an RNA strand toward some minimum value while simultaneously biasing the RNA strand toward a desired secondary structure. Since our approaches are based entirely on heuristics, their largest limitation is that there is no guarantee that our approaches will find a final sequence with minimal energy. As such, there is also no guarantee that resulting RNA sequences will exactly fold into the desired secondary structures that are passed as input to our models. Nevertheless, our algorithmic models are, at the very least, likely to generate sequences that fold into a secondary structure with features that are similar to that of the desired structure.

We have also discovered that our heuristics have a greater tendency to malfunction on longer secondary structure geometries. In particular, there are issues with sliding in stem sub-structures, and pairing within loop sub-structures that are especially evident in the application of our algorithms toward longer strands of RNA. It is to be expected that the designed sequences for longer strands of RNA will be more inaccurate than the designed sequences for shorter RNA molecular sequences.

Finally, our approaches do not strictly handle all arbitrary secondary structures because they are incapable of handling pseudoknot sub-structures. This is a commonly made simplification in RNA folding problems because no suitable models exist for minimizing the energy of secondary structures that contain pseudoknots. It is for this reason that pseudoknots are frequently considered to be part of tertiary, as opposed to secondary, structure in models of RNA molecules.

7 Future Work

Similar to probabilistic search techniques such as simulated annealing, genetic algorithms are often used to obtain solid approximations of optimal solutions for computationally expensive problems. Classic genetic algorithms accomplish this by starting out with a ”population” of initial solutions, evaluating each solution in the population against a ”fitness” function, and assigning probability values for each solution based on the determined fitness of each potential solution. Solutions that are deemed to be fit, or more desirable, have a greater probability to carry some aspects of themselves toward the development of new, and conceivably better, solutions in future iterations of a genetic algorithm. In this way, new solutions are iteratively generated by the application of so-called genetic operators, which permit the recombination and permutation of an existing population of solutions. Given such potential, we believe that we might be able to improve the performance of our heuristic models by using genetic algorithms alongside our devised approaches.

In order to apply a genetic algorithm to our approaches, we would need to be able to parse and identify the individual sub-structures that are present in a given RNA secondary structure. Once these sub-structures are identified, we would be able to apply our own algorithmic models to generate plausible sequences for each sub-structure. In order to test the plausibility of a conformation between our designed RNA sequence and a particular secondary sub-structure, we might be able to make use of currently existing folding algorithms that take RNA sequences and predict secondary structure. In a subsequent step, multiple concatenations of designed sub-sequences would permit us to create an initial population of sequences that should fold into the overall desired secondary structure. Furthermore, by applying well-known genetic operators such as crossover and mutation, we would be able to affect the evolution of a population of designed sequences over multiple iterations. In the end, those designed sequences with the highest fitness, or the most stable energy configurations, would make up the final resulting outputs for our augmented approach.

It might also be interesting to see how an implementation of a heuristic for ”expanding RNA” might affect the performance of our existing approaches. The process of ”expanding RNA” begins with a single RNA sequence that completely pairs with itself, thereby permitting our approach to start with an RNA sequence that is guaranteed to have minimum energy. In subsequent steps, our approach would proceed to locate the unpaired bases in the desired secondary structure geometry, mutating the initially paired bases in our initial RNA sequence in order to ”expand” the strand into the appropriate secondary structure. A final step in this heuristic would permit the insertion or deletion of bases in order to generate a final, coherent RNA sequence.

We believe that the expansion of an RNA sequence in this fashion has the potential for quickly creating RNA molecules with minimal energy states. Since the initial sequence prescribed by the heuristic is already at a minimal energy state, this heuristic would presumably also create a molecule with similarly minimal energy state; the only notable source of energy state modification would come from the insertion or deletion of bases, which might globally perturb the energy configuration of a designed RNA molecule. As such, it should become relatively clear that the largest difficulty to overcome in the implementation of this heuristic would be determining the appropriate criteria for inserting or deleting nucleotides from the final secondary structure without greatly disrupting global energy state, or inducing structural sliding, which could potentially destroy the stability of a predicted conformation.

8 Conclusion

Although the problem of designing RNA sequences from secondary structures is not particularly well-studied, we have clearly demonstrated that feasible solutions exist. We have presented a selection of several different heuristics, including probabilistic modeling, self-loop fold avoidance, sub-structure energy negation, and graph-based approaches. Each of these techniques might be used in order to tractably design RNA sequences for input secondary structure geometries. From an evaluation of these techniques, our initial evidence suggests that prediction models that are built around basic probabilistic techniques may make a good starting point for exploration, although more complex models are necessary in order to obtain the best predictions that are possible.

Our exploratory approach to designing heuristics has further yielded some very interesting insights into formally modeling the relationship between RNA molecular sequences and their secondary structures. We believe that with further research and experimentation, even better solutions may be possible in the future. In this light, we hope that our own results may provide some inspiration for future work in this area.

9 Appendix

9.1 Strand Information

Experiment 1

Strand	Source	Accession	GI	RNA type
1	Stylonychia mytils	U10570	726453	RNA subunit of telomerase
2	Paramecium caudatum	U45437	1245049	telomerase RNA
3	Paramecium multimicronucleatum	U45436	1245050	telomerase RNA
4	Paramecium primaurelia	U45434	1245051	telomerase RNA
5	Paramecium tetraurelia	U45433	1245053	telomerase RNA

Experiment 2

Strand	Source	Accession	GI	RNA type
1	Deinococcus radiodurans	AE002087	6460405	tRNA-Gly-4
2	Escherichia coli	V00336	42763	23S-rRNA
3	Escherichia coli	V00336	42763	tRNA-Asp
4	Xenopus laevis	L15434	295540	scRNA
5	Kluyveromyces lactis	U31465	968979	tRNA-Ile

9.2 Histograms of individual strands

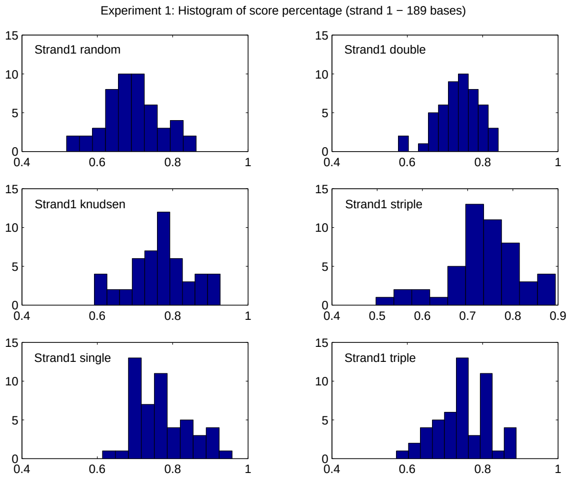

Figure 8: Experiment 1: This shows the distribution of correct scores for strand 1 under each probability distribution.

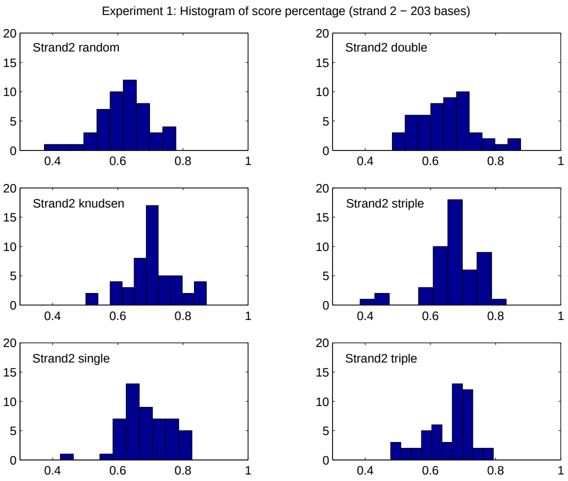

Figure 9: Experiment 1: This shows the distribution of correct scores for strand 2 under each probability distribution.

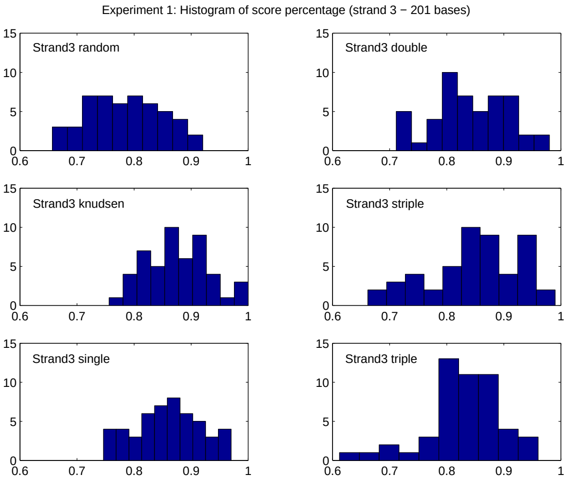

Figure 10: Experiment 1: This shows the distribution of correct scores for strand 3 under each probability distribution.

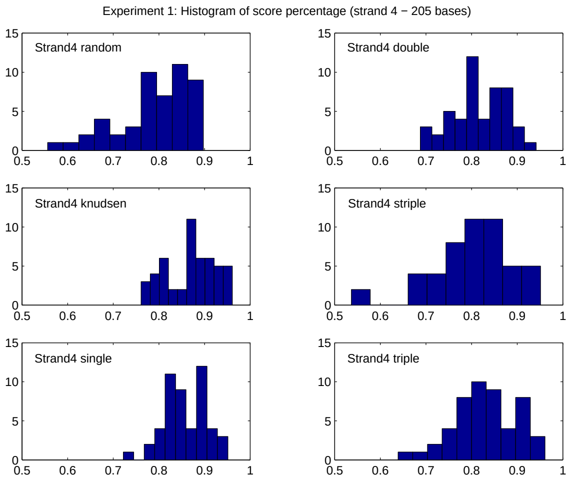

Figure 11: Experiment 1: This shows the distribution of correct scores for strand 4 under each probability distribution.

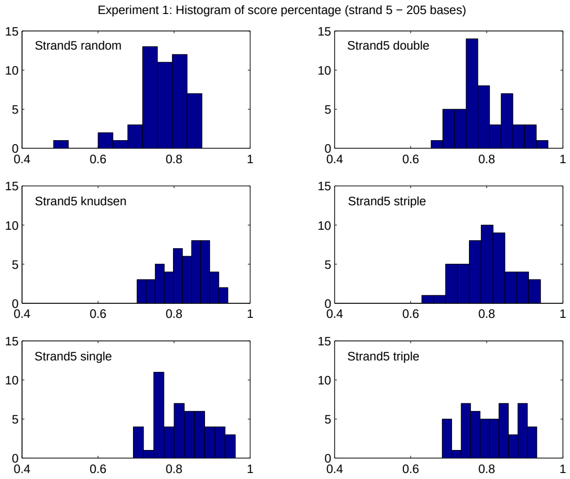

Figure 12: Experiment 1: This shows the distribution of correct scores for strand 5 under each probability distribution.

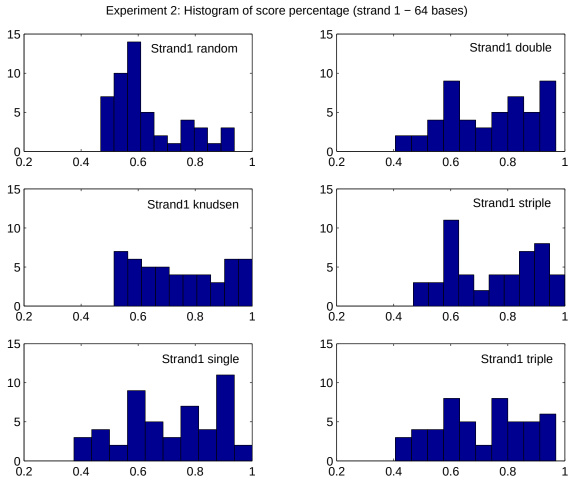

Figure 13: Experiment 2: This shows the distribution of correct scores for strand 1 under each probability distribution.

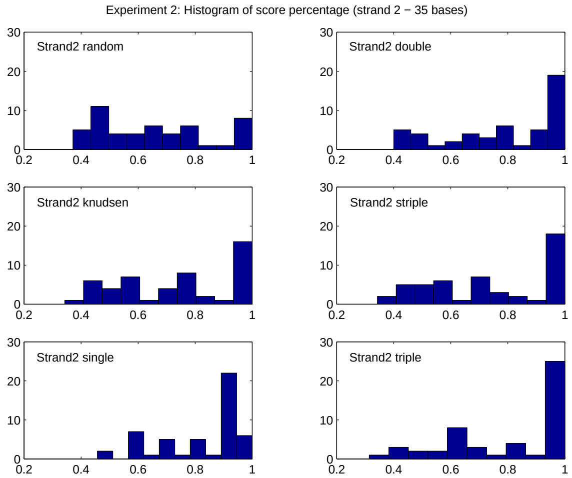

Figure 14: Experiment 2: This shows the distribution of correct scores for strand 2 under each probability distribution.

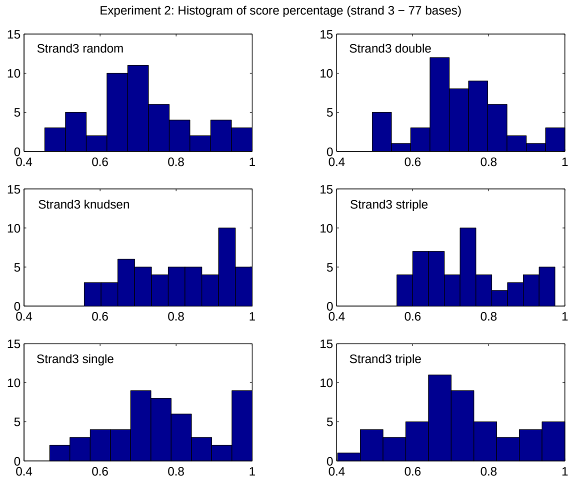

Figure 15: Experiment 2: This shows the distribution of correct scores for strand 3 under each probability distribution.

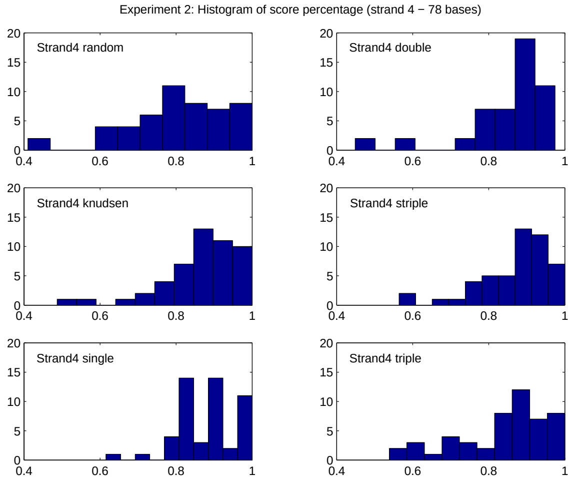

Figure 16: Experiment 2: This shows the distribution of correct scores for strand 4 under each probability distribution.

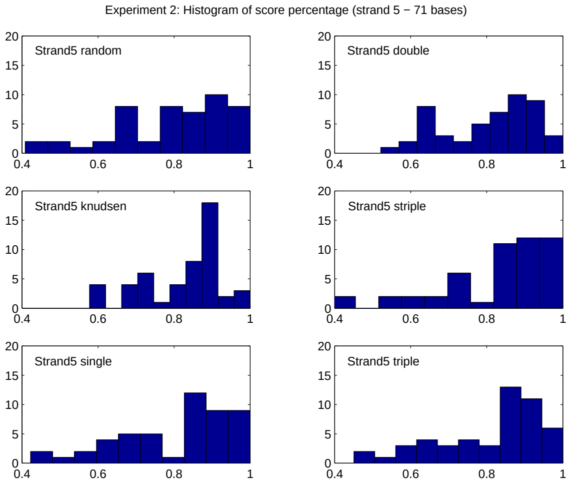

Figure 17: Experiment 2: This shows the distribution of correct scores for strand 5 under each probability distribution.

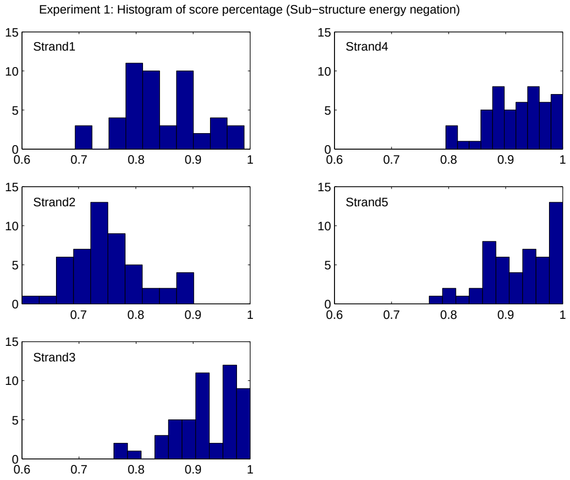

Figure 18: Experiment 1: This shows the distribution of correct scores for each strand under substructure energy negation.

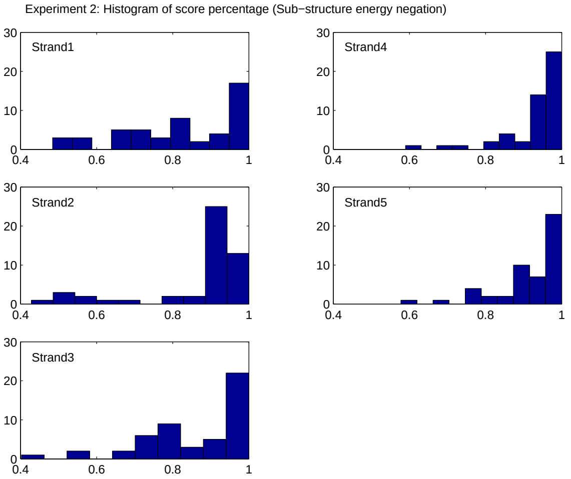

Figure 19: Experiment 2: This shows the distribution of correct scores for each strand under substructure energy negation.

References

[1] Berman, H. M. (1996). Foundations in Molecular Biophysics. In Nucelic Acid Database Project. Available: http://ndbserver.rutgers.edu/NDB/archives/NAintro/ (March 31, 2002).

[2] Brenner, S., Crick, F. H., et al. (1961). General nature of the genetic code for proteins. In Nature, Vol. 192, pp. 1227-1232.

[3] Flamm, C., Hofacker, I., Maurer-Stroh, S., Stadler, P. F., & Zehl, M. (2001). Design of Multi-Stable RNA Molecules. In RNA, Vol. 7, pp. 254-265. Available: http://www.tbi.univie.ac.at/papers/Abstracts/00-05-027abs.html (March 31, 2002).

[4] Friede, D. (2001). Design of Artificial tRNAs. Ph.D. Dissertation. Available: http://www.tbi.univie.ac.at/papers/PhD_theses.html (March 31, 2002).

[5] Hofacker, I. (2002). Vienna RNA Package: RNA Secondary Structure Prediction & Comparison. Available: http://www.tbi.univie.ac.at/ ivo/RNA/ (March 31, 2002).

[6] Hofacker, I.L., Fontana W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., & Schuster, P. (1994). Fast Folding and Comparison of RNA Secondary Structures. In Monash Chem, Vol. 125, pp. 167-188. Available: http://www.tbi.univie.ac.at/papers/Abstracts/93-07-044abs.html (March 31, 2002).

[7] Knudsen, B., & Hein, J. (1999). RNA Secondary Structure Prediction using Stochastic Context-Free Grammars and Evolutionary History. In Bioinformatics, 15(6), pp. 446-454.

[8] National Center for Biotechnology Information (NCBI) (1988). Nucleotide database. Available: http://www.ncbi.nlm.nih.gov/index.html

[9] Seeman, N.C. (1990). De novo design of sequences for nucleic acid structural engineering. In Journal of Biomolecular Structure and Dynamics, Vol. 8, Issue 3, 1990, pp. 573-581.

[10] Perumal, K., Gu, J., Chen, Y., & Reddy, R. (1999) Small RNA database Available: http://mbcr.bcm.tmc.edu/smallRNA/index.html

[11] Zuker, M., Mathews, D. H., & Turner, D.H. (1999). Algorithms and Thermodynamics for RNA Secondary Structure Prediction: A Practical Guide. In RNA Biochemistry and Biotechnology, pp. 11-43. Dordrecht, NL: NATO ASI Series, Kluwer Academic Publishers.