

2011-01-01

Prediction of Ribonucleic Acid Secondary Structures Using A Heuristic Backtracking Search

Christopher Roman Cuellar

University of Texas at El Paso, ccuellar4@miners.utep.edu

Follow this and additional works at: https://digitalcommons.utep.edu/open_etd



Part of the [Bioinformatics Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Cuellar, Christopher Roman, "Prediction of Ribonucleic Acid Secondary Structures Using A Heuristic Backtracking Search" (2011).
Open Access Theses & Dissertations. 2263.

https://digitalcommons.utep.edu/open_etd/2263

PREDICTION OF RIBONUCLEIC ACID SECONDARY STRUCTURES USING A
HEURISTIC BACKTRACKING SEARCH

CHRISTOPHER R. CUELLAR

Department of Computer Science

APPROVED:

Olac Fuentes, Ph.D., Chair

Martine Ceberio, Ph.D.

Ming-Ying Leung, Ph.D.

Benjamin C. Flores, Ph.D.

Acting Dean of the Graduate School

Copyright ©

by

Christopher R. Cuellar

2011

Dedication

To my mother and father who have given me unfaltering support,
to my brother and sister who have always been there for me,
to my fiancé Neith and Daniel for their love and compassion,

Thank you and my love goes to all of you.

PREDICTION OF RIBONUCLEIC ACID SECONDARY STRUCTURES USING A
HEURISTIC BACKTRACKING SEARCH

by

CHRISTOPHER ROMAN CUELLAR, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

In Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Computer Science

THE UNIVERSITY OF TEXAS AT EL PASO

December 2011

Abstract

Ribonucleic acid (RNA) is essential for all forms of life. RNA is made up of a large chain of nucleotide bases: Guanine (G), Uracil (U), Cytosine (C), and Adenine (A). An RNA strand can fold on itself to allow G-C, A-U, and G-U bases to form hydrogen bonds, this is known as a secondary structure. Knowing the secondary structure of an RNA chain is very important because it will allow researchers to better understand its specific functions. RNA will create secondary structures that tend to minimize their free energy. RNA secondary structure prediction is the attempt to predict physical folding of RNA given its linear strand.

A common approach to RNA secondary structure prediction is dynamic programming. Dynamic programming is based on the assumption that a given problem can be solved optimally by recursively solving its subproblems optimally. Dynamic programming approaches for secondary structure prediction have running times of $O(n^3)$, where n is the length of the RNA sequence. There are two main problems with the dynamic programming approach to RNA secondary structure. First, for very long chains, computing a prediction can take a substantial amount of time. Second, some foldings contain secondary structures that violate the assumption of optimal substructure.

In this thesis, I propose an approach to RNA secondary structure prediction that attempts to overcome the limitations of dynamic programming. The approach is based on depth-first search in combination with a set of heuristics. I use a preprocessing stage, first proposed by Weise for his genetic algorithms, to find palindromic sequences, which are helical regions of RNA pairings. Then I search for a subset of structures that are mutually compatible and minimize the free energy using depth-first search. This search is further sped by applying a set of heuristics that take into consideration palindrome length and likely compatibility with other potential structures. A couple of advantages of this depth-first search approach are that it does not rely on optimal substructures and is easily parallelizable. Experiments show that the proposed methodology is promising because of these advantages and the results that were produced being competitive with those of MFOLD, a well-established secondary structure prediction algorithm.

Table of Contents

Abstract.....	v
Table of Contents.....	vi
List of Tables	viii
List of Figures.....	ix
Introduction.....	1
1.1 RNA folding characteristics	1
1.2 RNA secondary structure prediction background.....	2
1.3 Goals.....	4
Related work.....	5
2.1 Description of generic algorithms	5
2.2 Description of related approaches	7
Heuristic techniques for secondary structure prediction	12
3.1 Determination of preprocessing technique	12
3.2 Backtracking searches	16
3.3 Energy calculation	22
Experimental results	24
4.1 RNA sequence database	24
4.2 Experimental Method	25
4.3 Heuristic Scoring	27

4.4 Discussion.....	29
Conclusion and future work.....	31
5.1 Future Work.....	32
Bibliography	34
Complete results	35
A.1 Search Results for Heuristic A with Subpalindromes	35
A.2 Search Results for Heuristic A without Subpalindromes	39
A.3 Search Results for Heuristic B with Subpalindromes.....	43
A.4 Search Results for Heuristic B without Subpalindromes.....	47
A.5 Search Results for Heuristic C.....	51
Example input, output, and visualization of given heuristics	55
RFam sequence database that was used.....	58
Vita	74

List of Tables

Table 4.1 Resulting averages on differences and energies	28
Table 4.2 Resulting standard deviations and scores	28

List of Figures

Figure 1.1 RNA strand RF0037_A with secondary structure.....	2
Figure 1.2 Examples of various secondary structure formations.	3
Figure 3.1 Possible loops detected by the preprocessing algorithm (8)	13
Figure 3.2 Displaying overlap of palindromic sequences.	14
Figure 3.3 Pseudocode for palindromic sequence finder.....	15
Figure 3.4 Displaying decomposition of parent palindrome.	16
Figure 3.5 Example of the created search tree.....	17
Figure 3.6 Pseudocode for heuristic A	19
Figure 3.7 Pseudocode for heuristic B.....	20
Figure 3.8 Pseudocode for heuristic C.....	22
Figure 4.1 Example of fasta format	25
Figure 4.2 Example of visualized generated secondary structure.	27

Chapter 1

Introduction

Ribonucleic acid serves multiple essential roles within the body. RNA can be used as a carrier of genetic information, a guide to proteins for genetic modification or destruction, storage of genetic information, a template for protein synthesis, and a sensor to determine ionic concentrations and temperature (1). RNA can also exist as an enzyme which is used to unwind RNA helices and reorganize RNA structures for repurposing, to change its secondary structure functionality (2). Determining the structure of RNA will allow insight into the structural use of specific RNA strands.

1.1 RNA folding characteristics

RNA strands are created by the chemical bonding between Guanine (G), Uracil (U), Cytosine (C), and Adenine (A). The common bonds found within an RNA strand is associative and consist of G-C, A-U, and G-U pairings. Linear strands of RNA, considered primary structures, can fold upon themselves to form more complex structures, commonly known as secondary and tertiary structures. Secondary structures are created by the hydrogen bonding within a singular RNA strand. Tertiary structures are characterized by a second folding on top of the secondary structure that was created; in effect, this produces a three dimensional structure of RNA (2). These structures are necessary to stabilize the RNA within the environment.

RNA utilizes two types of chemical bonding to form its secondary structure: G-C and A-U pairs known as Watson-Crick hydrogen bonds and G-U pairs considered non-Watson-Crick bonding which are unstable (2). Hydrogen bonding is generally a weak form of bonding when compared to covalent and ionic bonds; however, if present in great numbers, the bonding between molecules becomes strong. Guanine and Uracil bonding tends to distort the ideal form of the molecule if it were standalone by causing tension on the backbone structure. Guanine and Cytosine pairing form three hydrogen bonds and are the most stable out of the three common forms of base pairs (2).

1.2 RNA secondary structure prediction background

For the purposes of this thesis, I will focus on secondary structure prediction based upon the primary structure. RNA secondary structure prediction is the attempt by a computer to determine the secondary structure of an RNA chain. As seen in Figure 1.1, bases A and U and G and C tend to bond with each other much more frequently than G U bonding. These more favorable bonds will be given more consideration than G-U pairings when the secondary structure is predicted.

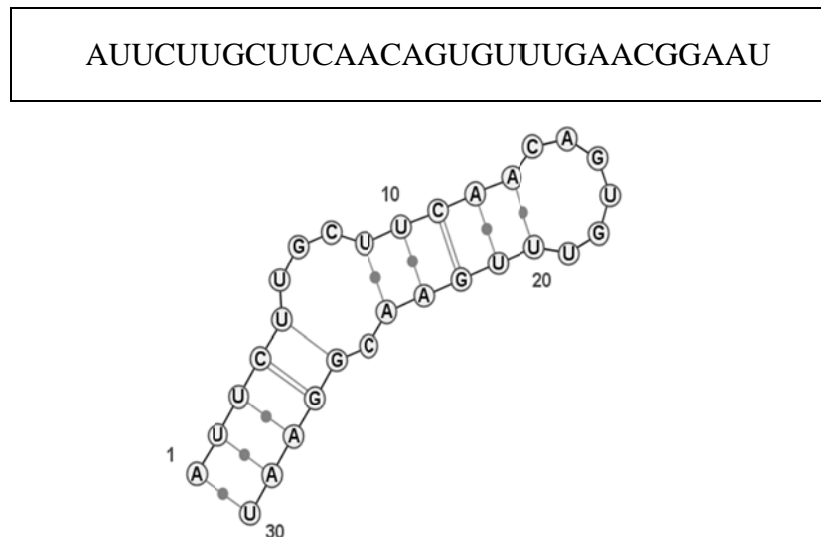


Figure 1.1 RNA strand RF0037_A with secondary structure

Within a secondary structure, there are many sub-structures that are created: the simple helix, the loop, pseudoknots, kissing loop complex, and helical junction (2). The helix is characterized just by general base pairing within the structure as seen in Figure 1.1. A term that will be utilized within this thesis is “palindromic sequence” or simply “palindrome”; these terms refer to a helical section of RNA. The loop has a few variations: a hairpin loop where there are approximately three or four free bases between a helical section, internal loop which is the presence of an equal number of free bases between two helical joins, and bulge which is characterized by an uneven number of free bases between two helical joins (2). A pseudoknots is found where a loop’s unpaired bases become paired with another set

of bases after the helix. The kissing loop complex is when two loop unpaired bases attach to each other (2). A helical junction is when three loops are formed and their helixes all meet in the same spot.

Complications to predicting RNA secondary structures arise when the form of the secondary structure is considered as a part of the body ecosystem rather than as a singular unit. Sections of RNA synthesis is a consideration point wherein certain parts of RNA are synthesized at different times, restricting other folds from occurring (2). Environmental factors play a role in how RNA folds upon itself. RNA is inherently negative due to the oxygen in their backbone structure; however, if there are free positive ions within the environment, it allows the RNA to pair more efficiently (2). Another external factor is caused by enzymes slowing the folding process which provides RNA strands time to form their secondary structures properly; an example is found in e-coli wherein the RNA polymerase can stall during transcription to allow proper folding (2).

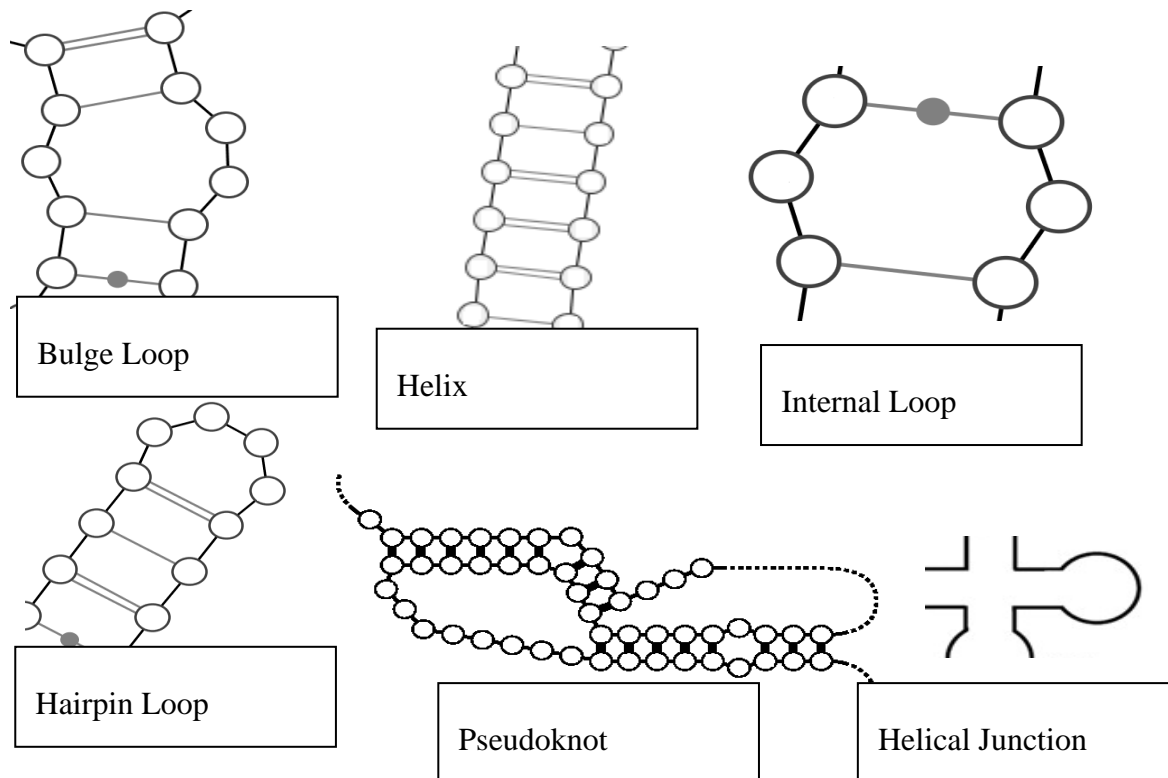


Figure 1.2 Examples of various secondary structure formations.

In the top row from top to left is a bulge, helix, internal loop. On the bottom row is a hairpin loop, pseudoknots, and helical junction.

1.3 Goals

The following goals lay the foundation for this thesis.

1. To determine the possibility of creating a competitive algorithm that will be based on a backtracking search;
2. To create an extendable algorithm that can be parallelized based upon a backtracking search.

The first goal will attempt to address the capability of creating a competitive algorithm that is not based heavily on physics models nor pseudo-random searching such as a genetic algorithm. The second goal address making an algorithm that is able to be parallelized that is based on a backtracking search. By attempting to perform a search upon subsections of RNA, we allow the search tree that is created to be distributed and searched upon in a parallel manner. This will help with running time and predictions as the search tree would be able to be searched more efficiently.

Chapter 2

Related work

Several techniques have been employed in order to predict RNA secondary structures. They either use a base-pair representation where each possible base pair is considered or a palindromic sequence based representation. A palindromic-sequence-based representation is when entire helices are considered rather than each individual base pair. Data-driven secondary structure prediction approaches attempt to have the computer learn common substructures by examining a multitude of secondary structures. Energy minimization algorithms attempt to minimize the free energy of a secondary structure to improve predictions. The prediction techniques that will be covered in this section are a conditional log-linear model (CLLM), a parallelizable genetic algorithm, and a dynamic programming algorithm. The CLLM is a data driven type algorithm, while the genetic algorithm and dynamic programming algorithm are both energy minimization algorithms. In order to fully understand the techniques presented in this section, a basic understanding of what these algorithms are designed to do must be had.

2.1 Description of generic algorithms

I will describe the algorithmic concepts related to the current approaches of secondary structure prediction. I will also clarify terms pertinent to understanding the algorithm descriptions that follow.

2.1.1 Dynamic programming

Dynamic programming is based upon the principle of optimizing a larger problem by first considering smaller optimal subproblems. This relies on the problem to have the property of being able to be split into these smaller subproblems and recursively the subproblems split into even more subproblems. The subproblems results are then stored for later use, at any time the same problem is encountered. In order to develop a dynamic programming algorithm, four generic steps can be utilized: determine the structure of an optimal solution, recursively define the value of an optimal solution,

compute the value of an optimal solution starting with smaller problems, and create an optimal solution (not necessarily a single optimal solution in the even that multiple solutions exist) from the results of subproblems (3).

2.1.2 Genetic Algorithms

Genetic algorithms take their foundation from the biological principles of evolution. It will use nondeterministic means in order to generate a solution; however, this generated solution is not necessarily the best solution. Genetic algorithms are designed to handle problems whose search space is too large to search systematically. Concepts that will be used to define the processes of genetic algorithms are generation, population, parents, and children. A generation is the set of possible solutions after each iteration of the genetic algorithm. The population is the set of results at any given generation. Parents are defined as a couple of prior results that will be used to generate a couple of new results; the new results are called children. In order to use a genetic algorithm, the following must be created: a representation, a crossover technique, mutation, a fitness function, and a selection technique.

A representation is a way of encoding the configuration of a solution to the problem being solved; usually this is a binary string. An example of a representation is if you consider packing and trying to fit everything based upon size and worth into the truck. Then to signify that something was taken, a 1 is used, and to signify something was not taken, a 0 is used. In this manner, you can then classify all objects by an in-order series of 1's and 0's as taken or not taken.

Crossover techniques are used to combine two parent representations to form new children representations containing some traits from both parents. In this way, the search can continue to progress in the same general area in the search space. What this means is that you would not randomly select a solution without basis, it would build upon its parents. Some common crossover techniques include taking half of the representation from the first parent and concatenating it with the other half of the second parents' representation or using concatenating alternating quarters of the representation of both parents.

Mutation is when a representation is altered in a random way, replacing some 0's with 1's and vice-versa. Mutation allows for a representation to occur that would not have otherwise. Mutation

should occur with a small percent chance otherwise the algorithm could fall easily into a local minima or maxima and be unable to escape. By introducing a random factor to the population, it allows the search space to be expanded after becoming too narrow (if the representation that was generated after random altering is a good representation).

A fitness function is a mathematical way to calculate the worth of a representation. A high fitness is good and will characterize a relatively good representation while a low fitness means that the representation is not characteristic of a good result. Fitness functions are unique to the problem and not reusable. A good fitness function is pivotal to a genetic algorithm as the overall generated result is relative to the effectiveness of a fitness function characterizing a good result.

A selection technique is used to determine who the parents are for crossover. A well-defined selection technique is also very important as a bad selection technique can deteriorate diversity rapidly which results in a much narrow solution space. A common selection technique is called tournament selection where the representation with the best fitness is taken as a parent from a subset of the population. This assures that some lower fitness results might be included in as a parent which is necessary to preserve diversity.

2.2 Description of related approaches

For the log-linear model, I will be examining the Contrafold (4) approach. For the genetic algorithm, I will be examining an approach described by Bruce A Shapiro et al (5); and for the dynamic programming algorithm, I will examine MFOLD (6).

2.1 Contrafold

Contrafold utilizes a combination of known patterns which have been embedded in a context-free grammar and patterns learned from known secondary structures. A key similarity between Contrafold and the algorithms within the thesis is that both attempt to generalize the structures by obtaining a set of possible substructures given a certain sequence of RNA. These substructures can be used to compose the overall secondary structure of RNA. A main difference between the algorithms given in this thesis and Contrafold is that Contrafold does not utilize all the complex rules found in strict

thermodynamic models in order to avoid overfitting during the learning process. The best predicted secondary structure is determined by calculating the conditional probability $P(y|x)$ where x is the RNA sequence input and y is the predicted secondary structure.

The calculation of this probability is determined by the learned probabilities for given features present within the CLLM. The overall probability of a given secondary structure is the sum of the probabilities of each parsing within the structure. The learning process is dependent on calculating the conditional maximum likelihood for each parsing based upon the training data.

This technique was designed to be independent of thermodynamic models and rely more on probabilities based upon the consensus secondary structures of RNA sequences. Due to this design, it is significantly different than the backtracking search method proposed within this thesis.

2.2 Parallel Genetic algorithm

This parallel genetic algorithm follows the same design as described in Section 2.1.2. The parallel genetic algorithm is split into two sections, preprocessing and then the actual genetic search. Preprocessing relies on finding “stems” or palindromes within a RNA strand for use later; this is similar to the technique described in Section 3.1. Before discussing the parallel genetic algorithm, I will first describe the Boltzmann filter and the parallel approach. The actual explanation of the genetic algorithm will follow, the description of a couple components prior to the actual algorithm is intended to improve understanding.

A Boltzmann filter determines if a palindrome is accepted into a structure based on certain criteria related to the free energy of the current structure. All palindromes that are considered for inclusion into the generated secondary structure must be geometrically compatible; this means that they must not conflict with any palindrome that is already included. If the inclusion of a palindromic region will result in lowering the energy within the structure, then the palindromic region is included in the newly generated structure without restriction. If the inclusion of a palindromic region will result in a more unstable structure, a structure with higher free energy, then it is included if it is likely to create a new structure. A new structure is defined as a structure that has not been previously generated by the genetic algorithm. A palindrome is considered likely to create a new structure by generating a random

number in the space of $[0,1)$ and comparing it to the Boltzmann transitional probability. If this random number is less than the Boltzmann transitional probability, then it is accepted; otherwise, the palindrome is rejected and not included into the newly formed secondary structure. The Boltzmann transitional probability is a number that was calculated by Contrafold to represent the likelihood of creating a new structure. By utilizing this filter, it will allow for more diversity of the population because it will still allow palindromes that increase instability to become a part of a secondary structure.

The parallel approach taken by this algorithm needs to be clarified so later when I refer to it, it will not be ambiguous. In the smallest instance, there are 9 processors all linked together in grid. Each processor will house a single representation and the list of palindromic regions that were found during preprocessing. Each processor will then apply the genetic algorithm upon the structure that it contains.

The preprocessing technique that this genetic algorithm uses will differ in a couple of ways, the minimum size of a palindromic sequence mainly with respect to the restrictions placed upon it. One invariant is that the palindromic sections are restricted to at least 3 base pairs. The generated palindromes will also include subpalindromes of at most two bases of unzipping on either side, this means that a palindromic sequence of size 9 can create two of size 8, three of size 7, two of size 6, and one of size 5. Upon completion of the search for palindromic sequences, the list is stored in each processor for quicker access.

The selection technique chosen by this algorithm utilizes a fitness function and a rule-based engine. The rule-based engine has a bias towards better fitness although it does not fully exclude others that have lower fitness. The 3x3 neighborhood structures described previously are the only structures considered for selection.

The crossover technique first requires that a mutation occurs. The mutation occurs by making major changes to the overall child structure, two independent child templates are created by randomly including palindromic sequences. These included sequences follow the same rules as initialization. Then the crossover is applied such that palindromic sequences are included from the chosen parents as long as they pass the Boltzmann filter. Finally, the child structure with the lowest free energy is saved as the result of that generation.

The resulting predicted secondary structure relies on the consensus of all involved processors, not just the 3x3 grid. One main problem with this algorithm is that because it relies on consensus, the secondary structures contained within a large number of processors are equivalent. This repetition is a shortcoming because of repeated work that was done to find that secondary structure. The unpredictability of genetic algorithms also does not assure that repeated searches would not be performed. The algorithms provided within this thesis are inherently parallelizable and their format would allow for each processor to calculate a part of the palindromic sequence tree independent of others and will therefore exclude a majority of repeated work that this genetic algorithm relies on.

2.3 MFOLD

MFOLD is a dynamic programming approach whose basis is “divide and conquer”. MFOLD considers the folding of each region as its subproblems. This evaluation of subproblems involves the use of energy calculations provided by EFN2 (7). Due to this constant evaluation, MFOLD becomes heavily reliant on its thermodynamic modeling algorithm.

This approach relies on the splitting of an input RNA sequence into smaller sections of RNA, folding that smaller section and then calculating the Gibbs free energy of that section. MFOLD will then use this smaller sequence folding and continue expanding the region being considered (expanding the subproblem) until the entire RNA sequence is considered. This approach inherently has problems with structures such as pseudoknots because of how a pseudoknot structure exists. Pseudoknot inclusion is where bases within a loop will pair with bases outside of that loop causes the subproblem, that already had its determined structure, to be examined again. This revisiting of the subproblem should not happen in dynamic programming. Once the best solution has been found for a given RNA region, the algorithm should no longer have to examine that region again. A matrix is created to represent the energies of each substructure within the RNA strand and a traceback algorithm will find a path of least energy. Multiple output structures can be generated by forcing the traceback algorithm to start at varying points within the energy matrix.

One problem with this approach is that dynamic programming algorithms are inherently serial. This restricts the utility of MFOLD as computing becomes more decentralized. A problem with the

dynamic programming approach for secondary structure prediction is that constant evaluation of each suboptimal problem can compound error caused by energy calculation. This means that MFOLD has a heavy reliance on the energy calculation techniques and is susceptible to defects in the energy calculation. A direct result of this heavy reliance is also that MFOLD is restricted to be as effective as the underlying energy calculations it uses. A common trait of MFOLD with the algorithms provided within this thesis is the use of EFN2 for energy calculation.

Chapter 3

Heuristic techniques for secondary structure prediction

The approaches that will be described within this section have three main parts: preprocessing the RNA data, performing the search on this data, and calculating the energy (sometimes in tandem with the search) of the results which are generated structures. The success of achieving the stated goals was tested by comparing this algorithm to the results of a well-known algorithm, MFOLD (6), as well as the known structure.

3.1 Determination of preprocessing technique

The problem of RNA folding falls under the class of NP-complete problems, which means that it is impossible to perform a search on every possible base for practical purposes. NP-complete problems are characterized for their inability to be solved using deterministic methods in reasonable amount of time; but, the capability for a result to be verified in a reasonable amount of time. The search space for RNA secondary structure generation is too large to be searched systematically. To attempt to mitigate this, the RNA strand was to be split into manageable chunks that could be interchanged and searched upon. This technique was inspired by the original idea created by Weise and Glen in their preprocessing technique (8). By using preprocessing to extract palindromic sequences from an RNA strand, the search space is reduced and should be able to be directly searched upon using a backtracking search to provide meaningful estimations. Possible palindromes that can be found are illustrated within a secondary structure can be seen in Figure 3.1. This figure was created by using canonical base pairing with a minimum loop size of three and minimum helix size of 3.

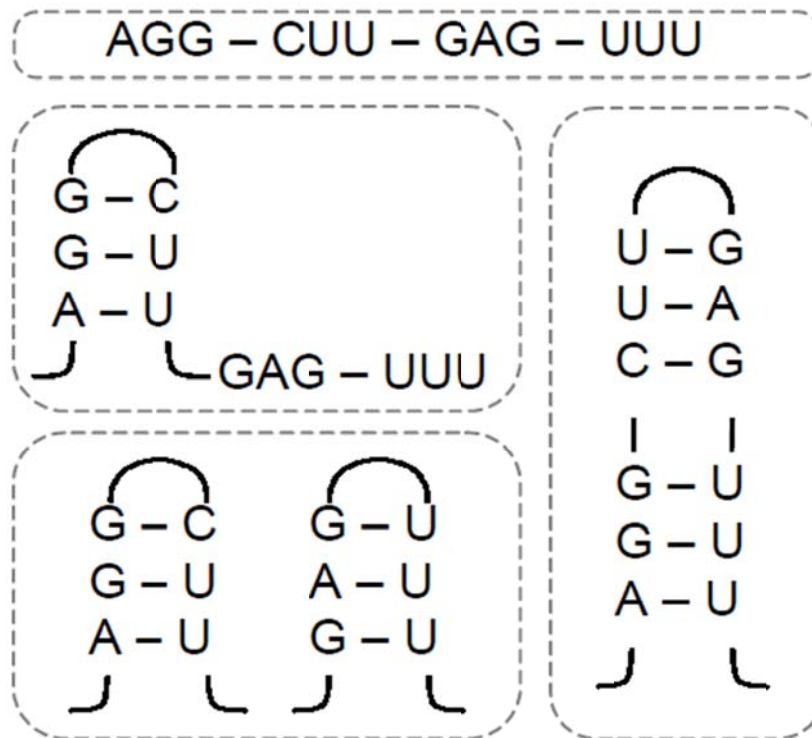


Figure 3.1 Possible loops detected by the preprocessing algorithm (8)

3.1.1 Palindromic sequence divisor

RNA strands consist of the linear connection between the G, A, U, and C bases. Palindromic sequences are found within RNA strands by determining which bases can pair with which other bases within the molecule. Input into the algorithm is the RNA strand, max length of a palindromic sequence, min length of a palindromic sequence, max gap size between base pairs, whether or not to allow overlap when searching for palindromic sequences, and if it should include wobble pairs (GU). Expected run time is $O(n^2)$ as it iterates through the RNA strand n times to search for palindromes.

The search begins at the first base of the RNA strand and then sets the endpoint for the search based upon the max length specified. It will iterate bases backwards until it finds a compatible base. When it finds the base, it will start to grow the palindrome in the start direction. In the case that the palindrome has a mismatched pair, then it will check the gap length and determine if it is acceptable. If not then it will discard the palindrome and continue the backwards search. If an appropriate gap length

is found, then it will continue the expansion until the completed palindrome size is determined. If the palindrome size is greater than minimum length provided, then the palindrome is stored in a list; otherwise, it is discarded.

Gap comparison is a special algorithm in itself; this is due to the fact that bulges and inner loops exist. Within the confines of an acceptable gap size, the possibility of a bulge must be examined as it cannot just be considered as an inner loop where the same numbers of non-matching base pairs exist.

Overlapping palindromic sequences are able to be found or not. Reasons to allow overlapping palindromic sequences are to provide a more robust search, namely perhaps palindromic sequence A is on bases one through seven and palindromic sequence B is on bases five to seventeen, then we'd certainly want palindrome B to be within the list of palindromes because this is likely one of the palindromic sequences in the generated secondary structure. Another prime example of when you would want overlapping palindromes is if palindromic sequence A is over bases 1-15 paired with 60-45, palindromic sequence B is over bases 1-6 paired with 45-39, and palindromic sequence C is over 7-27 paired with 39-19 then palindromic sequence A would inhibit B and C from being contained within the list even though optimally, B and C would be within the RNA sequence. In the event that simply too many palindromes are found, then it is possible to exclude overlapping palindromes in order to have a quicker search. Exclusion of overlapping palindromes will have a significantly reduced effectiveness unless the first palindromes found are the longest as all other conflicting palindromes will be ignored. For the purposes of my experimentation, I always included overlapping palindromes.

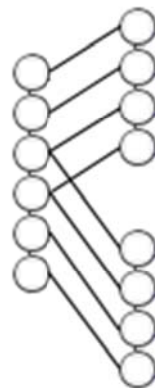


Figure 3.2 Displaying overlap of palindromic sequences.


```

beginning = start of sequence, end=end of sequence
for (current=beginning; current<end; current++)
    set the end of the palindrome to the max possible distance from current
    set beginning of palindrome to current + min length of palindrome
    for (backwards=the end of the palindrome; backwards > beginning of palindrome; backwards--)
        while mismatches is less than desired amount and the pointers do not go past each other
            check if the bases can base pair otherwise increment counted mismatches
            on base pair, increment count of size of palindrome
        if the palindrome ends in a mismatch, remove it and decrement counted size of palindrome
        if the gap between the halves of palindrome is acceptable and it reaches the minimum length and
            if an overlap exists, and overlaps are allowed, add it to the list of palindromes

```

Figure 3.3 Pseudocode for palindromic sequence finder

3.1.2 Subpalindromes

Palindromes provide a subset of the total number of base pairs to search on but in some instances are too restrictive. Overlapping palindromic sequences do not include subset sequences. While this may seem optimal for the search, it also does not provide for optimal subsequences for the overall secondary structure. If we consider the case where palindromic sequence A, over bases 1-5 paired with 40-35, which is a subset of palindromic sequence B, over bases 1-7 paired with 40-33, and palindromic sequence C which is an overlapping palindromic sequence on B. Then palindromic sequence C and A could co-exist and form a much longer chain than B by itself; however, since the subset of palindromic sequences of B are excluded from the search, this cannot occur.

In order to address this problem, the notion of “subpalindromes” is introduced. Subpalindromes are simply a set of subsequences of the original palindrome that was found. The calculation of subpalindromes was a decomposition of the parent palindromic sequence into its various combinations.

Each subpalindrome was tagged with the signature of the parent palindrome in order to keep track of origin.

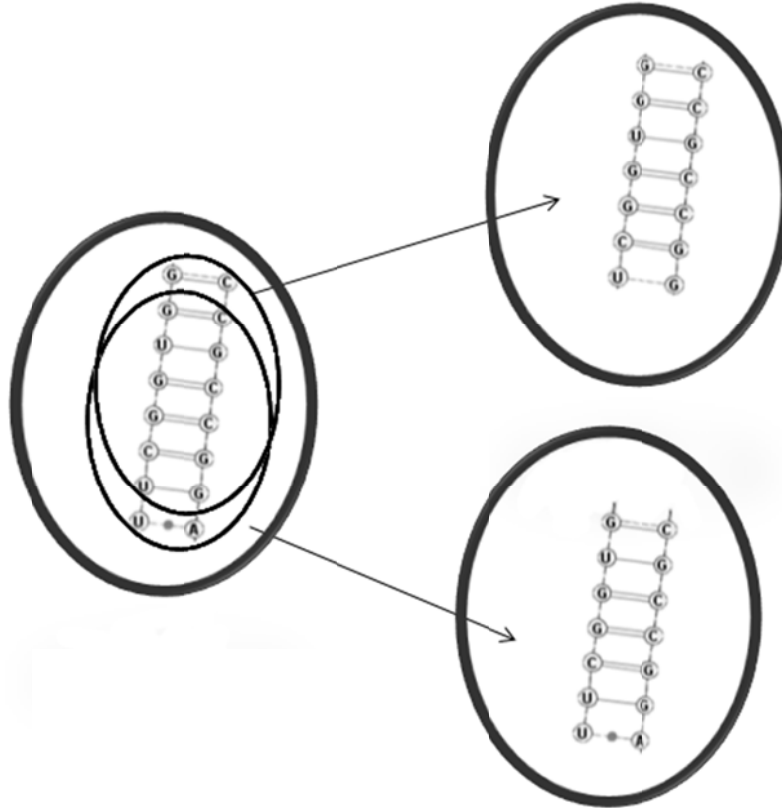


Figure 3.4 Displaying decomposition of parent palindrome.

3.2 Backtracking searches

A backtracking search is a type of search algorithm wherein the search can retrace its steps in the event that it took a wrong path. This allows searches to be flexible and not waste time attempting to recreate the incorrect path. The most common type of backtracking search algorithm is a depth-first search. In general, a depth-first search is where nodes are expanded in a depth manner. In this way, a depth-first search will find a solution after searching the length of a search tree. A main flaw of the depth-first search approach is that the solution found is not assured to be the best. This flaw is countered only by the fact that this search is, on average, faster as the whole search tree is not usually needed to be searched. The other general search technique, which is not a backtracking search, is a breadth-first search which expands all possible nodes/solutions on the tree and can assure the best solution. A

breadth-first search is impossible for this application as it would require expanding an exponential number of nodes.

The search tree that is used for this thesis is a binary search tree where each node represents a palindromic sequence and the two paths leading from this node is the option to either include or exclude the palindromic sequence from being part of the secondary structure. An attempt to perform a search on every single base within a RNA strand would result in an exponential running time. This hinders this approach because any practical RNA sequence would not have its secondary structure generated within a reasonable amount of time. To address this inadequacy, we utilize the preprocessing described in Section 2.1 to create manageable chunks to search on within the RNA strand. These palindromic sequences are sorted by relative values of their containing base pairs: 3 for G-C, 2 for A-U, and 1 for G-U.

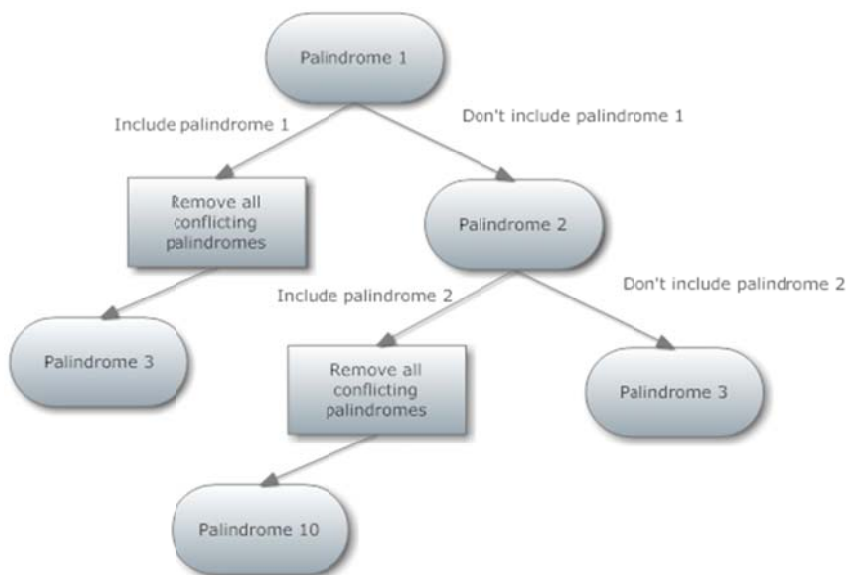


Figure 3.5 Example of the created search tree

3.2.1 First heuristic improvement on the depth-first search approach

A random ordering of palindromic sequences would expend resources on probably undesirable secondary structures being created. To prevent this from occurring, palindromic sequences are ordered by the relative scoring of each base pair that it contains. The score that each base pair gets is relative to the amount of stabilization provided to the secondary structure. G-C pairing, being the strongest,

receives a score of three due to the three hydrogen bonds that are formed between the two bases. A-U pairing, of middle strength, has a score of two because of the two hydrogen bonds formed. G-U wobble pairs receive the lowest at a one because of its mediocre stabilization capability.

After ordering the list of palindromes, the search tree is then created with the highest scoring palindrome at the root node. Arrays are created to maintain information pertaining to compatibility of palindromic sequences as well as to store the data of the search (which palindromes were selected as “taken”). The depth-first search is performed on the generated tree. The compatibility of each remaining palindromic sequence is checked based upon the taken sequence array that was created. Then the taken sequence array is updated after making the decision on whether to include the next palindrome or not. After the compatibility array of palindromic sequences is completely false, signifying that there are no more palindromic sequences that can be included, or the last palindrome has been examined, the completed configuration is added to the list of generated secondary structures.

The number of created secondary structures is bounded in order to create multiple structures while not attempting to search the whole tree. As stated previously, DFS does not guarantee the best solution, this is the reason why a set number of structures are created so they can be compared and the best generated structure selected. Clearly this algorithm has a possibility to ignore the best structure because it exists in a subtree that was not expanded. This heuristic also has the problem of considering palindromes that are not compatible as nodes that contain incompatible palindromes are still created. This heuristic limits the number of generated structures because all structures must be saved in memory. This will consequently diminish performance because less of the search space can be examined.

```
list of palindromes P
```

```
list of structures SL
```

```
function HeuristicA(current structure S, current pal index p) <- Current pal index is the index into the  
palindrome list
```

```
if size of SL < max size
```

```
if P[p] is not null
```

```
    if P[p] does not conflict with S
        add P[p] to S
            HeuristicA(S including P[p],p++)
        HeuristicA(S,p++)
    else
        add S to SL
```

Figure 3.6 Pseudocode for heuristic A

3.2.2 Second heuristic improvement

The second heuristic improvement utilizes the first improvement as a base. It has the same search including the stepwise progression through palindromic sequences as heuristic A. This heuristic also uses the same relative scoring of palindromic sequences that is used by heuristic A. This heuristic uses the same arrays to maintain search/generated configuration information as well as the compatible palindromes list that is updated every iteration. This heuristic is aimed at fixing the deficiencies of the prior heuristic; it will attempt to remedy memory restrictions.

This approach was devised to help with the amount of memory required as the first heuristic stored all structures created. This heuristic will only store a limited number of structures while continuously looking through the tree, this allows for the search to be expanded much more than before. To accomplish this task, an array with a small number of structures is maintained. After a completed path is created within the search tree, the energy of the structure is calculated. The energy is then compared to the energy of the structures within the array and either placed within the small storage array if its corresponding energy is less than that of at least one of the structures within the array or if there is an empty slot. The array is maintained in ascending order by calculated free energy.

The benefit of this heuristic is the capability to expand the search to more than the number that can fit in memory. This allowed for a more robust search as it could search, in some cases, a significant increase in the number of generated secondary structures than its predecessor. This also allows the

search to store the best structure based upon its energy, the benefit of this is that it will not use the rough estimate of the energy based upon the relative scoring.

```
list of palindromes P
list of structures SL
max expanded nodes M
function HeuristicB(current structure S, current pal index p)
  if size of SL < M
    oldS=S
    if P[p] is not null
      if P[p] does not conflict with S
        add P[p] to S
        HeuristicB(S,p++)
    HeuristicB(oldS,p++)
  else
    addStructure(S) // This is the main difference because this method will store only a set k
function addStructure(S)
  add S to SL
  sort SL based on energy from EFN2
  if SL > max size
    remove last structure in SL
```

Figure 3.7 Pseudocode for heuristic B

3.2.3 Third heuristic improvement

This heuristic utilizes the improvements that the second heuristic brought as well as attempting to improve the quality of the search by handling an increased search space with greater efficiency. This

search was optimized for use of subpalindromes in an attempt to make it feasible to search on a subpalindrome containing tree. The first two heuristics were geared towards a search without subpalindromes as it was a brute force search with ordering of the palindromic sequences using a good energy calculation. This heuristic attempts to address the problem of the large expansion of the search tree that is created.

This utilizes the same brute force approach with the palindromic sequences sorting but also will use palindrome signing. To illustrate the idea of palindrome signing, I will consider the case when there is no signing included. Suppose palindrome A contained 7 base pairs and palindrome B and C contained 3 and 4 respectively, B and C are also both subpalindromes of A, if B and C are not overlapping then when the search contains either A taken or both B and C taken, it results in the same path. Also assume there is a palindrome D which is a subpalindrome of A and contains 6 base pairs and palindrome B and E which are of size 3 and are subpalindromes of both A and D, then it results in the same path being searched again. This is a recursive problem when any “parent” palindrome is large enough to be reconstructed by smaller palindromes without reaching the minimum threshold. This complicates the search significantly because it becomes very difficult to know what was previously searched. What this means is, during the preprocessing stage, each subpalindrome was created and their corresponding parent’s signature was stored on the palindromic sequence, at any time only one palindrome with that signature can be used. This relies on the assumption that I would never want to create a loop where I would otherwise have a completely paired set of bases (as energy decreases with the number of base pairs within the generated secondary structure). What this does allow is the capability of a subset of a palindromic sequence to be considered in conjunction with, perhaps, another longer palindromic sequence. This would result in a higher percentage of base pairs.

The search itself is performed similar to the prior two heuristics after taking into consideration palindrome signing and non-compatible pairs. The search will iterate until a certain number of generated secondary structures are examined. Finally, it will return the best structure as determined by the calculated energy of the secondary structure.

```

list of palindromes P

list of structures SL

max expanded nodes M

function HeuristicC(current structure S, current pal index p, current signatures Sig)

  if size of SL < M

    oldS=S

    oldSig=Sig

    if P[p] is not null

      if P[p] does not conflict with S and Sig does not contain P[p].sig

        add P[p] to S

        add P[p].sig to Sig

          HeuristicC(S,p++,Sig)

        HeuristicC(oldS,p++,oldSig)

    else

      addStructure(S)

function addStructure(S)

  add S to SL

  sort SL based on energy from EFN2

  if SL > max size

    remove last structure in SL

```

Figure 3.8 Pseudocode for heuristic C

3.3 Energy calculation

Energy calculation for all searches is reliant on a thermal table with K-nearest neighbors backing (7). The code for the algorithm was based on a journal article in 1999, the implementation was provided by an open source program RNAstructure (9). Energy function 2, as called by RNAstructure, does not

support pseudoknot energy calculation which is slightly problematic when considering that some structures that have pseudoknots would not have pseudoknots in their predicted secondary structure. The database used for testing ignored pseudoknots so the effects of using EFN2 will not be examined; however, if EFN2 was replaced, this program would be able to generate pseudoknots effectively.

Chapter 4

Experimental results

The generated secondary structures were compared to the consensus secondary structure as well as a mainstream dynamic programming approach, MFOLD (10). A consensus secondary structure is created by using the most common pairing of a given set of nucleotide bases; the most common pairing is determined by using covariance models. The consensus secondary structures were generated by RFam (11). Later in the chapter, the results obtained during the experimentation process will be described. These results were used to analyze the differences, successes, and shortcomings of the previously described heuristics in comparison to those of MFOLD and the consensus structures.

To perform all experiments, the S151 Rfam database was used and the energy calculation for both MFOLD structures and the heuristic's generated structures (the heuristics seen in Section 3.2) was EFN2 to assure consistency (see Section 3.3).

4.1 RNA sequence database

The RNA sequences utilized when determining the effectiveness of the heuristics described in Section 3.2 were part of the S-151Rfam structural dataset (see Appendix C). This dataset is freely available by Rfam for their specific secondary structure research. The secondary structures that are contained within this package provide multiple RNA strands paired with their consensus secondary structures. These consensus secondary structures are used to calculate the precision and recall of the generated secondary structures.

simply by searching on plain helical regions. The maximum gap allowed between the end of one half of a palindromic sequence to the start of the other half was set to the length of the RNA sequence, this is to allow for the first bases to pair with the last bases of the RNA strand. A minimum gap was imposed to force hairpin loops not to have a loop less than the size of three. Finally, overlapping pairs were allowed, this was to bring variety and choices to the search algorithms as without overlaps included, it would have been very restrictive on the palindromes that were created.

After preprocessing was completed, each search algorithm would be performed on the palindromic sequences that were generated. Their outputs were stored independent of each other and included a .ct file describing the generated secondary structure and their predicted energies (See Appendix B for examples of generated output). The known structure was then read in for the specific RNA strand and the energy was calculated using the efn2 algorithm provided by RNAStructure. The information for the consensus secondary structure was also output to a flat file in .ct format describing the known structure and the associated calculated energy.

I then used MFOLD Server to calculate the generated structures for the same RNA sequences. This output was saved to a flat file that needed to be parsed into the various associated structures. The flat file contained the generated structure and its predicted energy. Since the energy calculation on MFOLD differs from mine, I would reparse the flat file that was generated and perform the same energy calculation I used for the determination of my generated structures. This disallowed the energy calculation algorithm not to be the defining factor for why either approach is better.

After determining each heuristic's generated structures and the calculated energies for the known, generated, and MFOLD structures, statistical analysis was applied for comparison. Precision and Recall was calculated based upon base pairs of the generated structure in comparison to the known structure for both the heuristics described here and the structures generated by MFOLD. The calculated energies of all three structures were then compared as well in order to determine the effectiveness in calculating a minimum energy structure. Runtime of each heuristic was logged for comparison purposes.

4.3 Heuristic Scoring

Each heuristic had their precision and recalls taken with respect to the consensus structure as well as their generated secondary structure energy compared to the consensus structure energy. Precision is a measure of how accurate the answer is with respect to the known answer and recall is a measure of how much was not included that was correct. In terms of this program, precision would be out of all the base pairs or singular bases included in the predicted structure, how many were correct with respect to the consensus structure; recall would be how many base pairs and singular bases were not included that should have been. The standard deviation was calculated on the precision and recall to determine exactly how effective this algorithm was in producing actual secondary structures. These calculations were performed for every generated secondary structure. An example of a visualized secondary structure can be seen in Figure 4.2; these visualizations were provided by Varna (12).

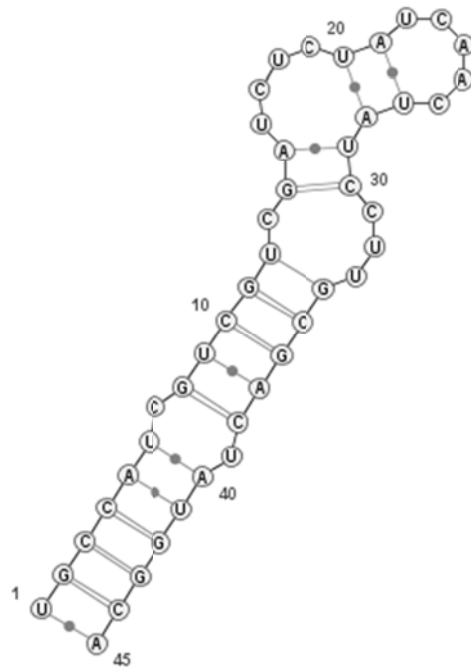


Figure 4.2 Example of visualized generated secondary structure.

Table 4.1 Resulting averages on differences and energies

			Average Precision	Average Recall	Average Energy Difference (Known-Generated)
Heuristic Subpalindromes	A		.4096	.4882	-.6719
Heuristic Subpalindromes	A	No	.4180	.4739	1.186
Heuristic Subpalindromes	B		.4091	.4889	-.6719
Heuristic Subpalindromes	B	No	.4174	.4739	1.186
Heuristic subpalindromes	C		.4002	.4996	-2.057
MFOLD			.5852	.6983	10.3

Table 4.2 Resulting standard deviations and scores

		Win	Tie	Loss	Standard Deviation Precision	Standard Deviation Recall	Standard Deviation Energy (Known- Generated)
Heuristic Subpalindromes	A	45	24	82	.6832	.6420	13.33

Heuristic A No Subpalindromes	52	20	79	.6734	.6427	12.58
Heuristic B Subpalindromes	45	24	82	.6832	.6424	13.33
Heuristic B No Subpalindromes	52	20	79	.6739	.6426	12.58
Heuristic C subpalindromes	34	15	102	.6875	.6380	13.85
MFOLD				.7143	.7216	16.03

Tables 4.1 and 4.2 show the resulting averages and standard deviations from all heuristic algorithms. Wins, losses, and ties are in relation to the MFOLD-calculated energy. Standard deviation is for the difference in precision, difference in recall, and energy differences.

4.4 Discussion

Heuristic A without subpalindromes and Heuristic B without subpalindromes obtained the exact same results. They both achieved the same average energy difference and the same standard deviation for energy difference. This result would make sense given that both of them use the same general approach, with the only difference being that Heuristic A would store up to N structures and stop the search while Heuristic B would store the top three structures and search M structures total. If M equals N, then the exact same search would be performed. In the case when M is larger than N, then they would differ only when a better structure was found later in the tree.

Heuristic A with subpalindromes performed relatively close to Heuristic A without subpalindromes. There were slight variations between the precision, recall, and energy difference of the heuristic with and without palindromes. An interesting difference between the generated search trees is that Heuristic A with subpalindromes, given its limited search capability, should have had a polluted tree with inclusion of subpalindromes. Since the differences were not so great across precision, recall, and

energy difference from the consensus structure, it can be derived that the solutions found when the algorithm included subpalindromes resided relatively closely to the initial solutions found when no subpalindromes were included.

Heuristic B with subpalindromes performed relatively similar to Heuristic B without subpalindromes. There were slight variations between precision, recall, and energy difference. The differences can be accounted for because of the limited search that was performed. Due to the integration with the energy calculation efn2, it was impossible to search the tree as deep as desired because of too much overhead with file I/O. With this limited search, Heuristic B with subpalindromes performed relatively the same as Heuristic A. It can be derived then, that Heuristic B was unable to take advantage of the gained free space in memory and therefore was unable to search much further than the palindromic sequences that did not contain subpalindromes. This also means that the addition of subpalindromes here could not benefit this algorithm due to the time constraints imposed.

Heuristic C inherently uses subpalindromes because it was designed specifically to eliminate repeated searches. Heuristic C has a slightly higher average energy difference and average recall than all the other purposed heuristics. This search did perform worse out of all the purposed heuristics when compared to MFOLD's predicted energy. The reason why Heuristic C performed so poorly was because of more restricted search capabilities. Heuristic C performed an energy calculation for each structure, this was computationally expensive due to the poor integration with efn2. As a result of the poor integration, Heuristic C had a much more restricted search than the prior searches which restricted the number of generated secondary structures. Since the amount of structures generated was lower than that of Heuristic A and B, the comparisons against MFOLD tended to be worse.

The amount of wins and losses were calculated by whoever has the lowest energy between MFOLD and each of the heuristics. Wins were in favor of MFOLD; although Heuristic A and B without subpalindromes achieved a relatively close win to loss ratio with a 17.9% difference. The results against MFOLD would be improved if the energy function integration were improved. This improved integration with efn2 would allow the search tree to be expanded more thoroughly causing more possibilities to encounter other structures.

Chapter 5

Conclusion and future work

This thesis attempted to achieve the following goals:

- To determine the possibility of creating a competitive algorithm that will be based on a backtracking search; and
- To create an extendable algorithm that can be parallelized based upon a backtracking search.

The heuristic backtracking searches explained in this thesis provide a set of competitive algorithms with at least one widely accepted secondary structure predicting algorithm. As a whole, the heuristics presented in Chapter 3 predicted secondary structures that had much higher free energy than that of MFOLD; however, the results are relatively close to those of MFOLD in terms of free energy which is good considering the lack of complexity with these approaches. This being the case the first goal was partially supported.

The algorithms described within this thesis provide an easy creation of a search tree that can be split and separated to various processors. The search tree can be passed to other processors by using prefixes on which palindromes are taken and not taken. This would allow for multiple prefixes to be passed to the processors and then have the search tree expanded for that prefix. With this, the algorithm is easily parallelizable and assures that repeated secondary structures should not occur, which will allow for greater efficiency with this search. With this, the second goal was supported; however, the actual parallelization implementation has yet to be performed.

A couple of possible problems exist with these algorithms. One possible source of error would be a bad target function (optimizing the wrong parameters), this would result in improper sorting of palindromic sequences which would affect the tree being created as possibly more significant palindromes could be searched first. A second source of potential error is a bad search heuristic, if a heuristic allowed repeated expansions then it would waste time and possibly bog down the entire set of results that were able to be created within a given amount of time.

The energy function that was used has some limitations, which could play key factors in the overall generated secondary structure. The energy tables used are not perfect and could continue to be refined as more information is found out about the energy of RNA secondary structures. The energy function cannot calculate secondary structure energy for structures containing pseudoknots. This will cause complications in generating structures with pseudoknots as their calculated energies would be set to the “error” energy which is significantly higher than any energy found in nature.

The search depth also plays a role in the final results. The search itself was shallow in that it explored relatively few structures; this is because of my self-imposed limitation motivated by the integration with the energy function. It currently utilizes I/O to relay the structures and energies between the search and the energy calculation. This I/O time engulfs the overall running time. In effect, improvements to rely on a much more native approach such as java’s native interface would allow running time to be improved which also would allow for more nodes to be explored in a certain amount of time. This problem is exacerbated when subpalindromes are introduced to the search tree as it generates even more possible structures; as a result, subpalindrome searches tended to be worse than their counterparts.

A benefit to this algorithm is the possibility of separating the search tree amongst multiple computers and parallelizing the search. This is a vast difference from the dynamic programming approach, which does not benefit from parallelization.

5.1 Future Work

There are several possible improvements on our depth-first-search-based technique. One would be to better integrate the energy function calculation so that runtime evaluations can be compared with other methods such as MFOLD. Currently the energy calculation integration relies on file I/O and causes the entire runtime to be devoted to that file I/O. An improvement this energy calculation would be to use a much more natural way to transfer information would significantly improve the running time and allow it to be compared to other methods. Another improvement would be to compare the results to those obtained from genetic algorithms. Currently, only the dynamic programming approach (MFOLD) is compared against. To obtain a better grasp on the potential of these algorithms, comparisons to other

secondary structure prediction methods need to be performed. Other potential heuristics should be devised to improve the results of the backtracking search. Possible heuristics include: create a better palindromic sequence evaluator to improve the initial ordering of palindromic sequences, and considering palindrome interactions more when performing the search rather than only using if palindromic sequences are compatible or not. Another possible improvement would be to actually perform the parallelization upon this algorithm. This would allow for much more of the search tree to be expanded which would provide better results than only a single processor performing the search. Another possible improvement would be to take a more greedy approach, rather than generating subpalindromes during preprocessing, create them based upon the palindromic sequences that are in conflict by only taking a subsection of that palindrome. This would allow for better search time by reducing the number of palindromic sequences used to search upon. Finally, we could try to forward predict better paths to search upon based upon the free energy reduction it is speculated to provide. This would theoretically improve the generated secondary structure from this search.

Bibliography

1. **Meister, Gunter.** *RNA Biology*. Weinheim, Germany : Wiley-Vch Verlag & Co. KGaA, Boschstr., 2011.
2. **Elliott, David and Ladomery, Michael.** *Molecular biology of RNA*. New York : Oxford University press inc., 2011.
3. **Cormen, Thomas H, et al., et al.** *Introduction to Algorithms*. Cambridge : The Massachusetts Institute of Technology, 2001.
4. **Do, Chuong, Woods, D A and Batzoglou, S.** CONTRAfold: RNA Secondary Structure Prediction without Energy-Based Models. *Bioinformatics*. 2006, Vol. 22, 14, pp. 90-98.
5. **Shapiro, Bruce A, et al., et al.** The massively parallel genetic algorithm for RNA folding: MIMD implementation and population variation. *Bioinformatics*. 2001, Vol. 17, 2.
6. **Zuker, Michael.** On Finding All Suboptimal Foldings of an RNA Molecule. *Science*. 1989, Vol. 244, 4900.
7. **Wiese, Kay C and Glen, Edward.** A permutation-based genetic algorithm for the RNA folding problem: a critical look at selection strategies, crossover operators, and representation issues. *Biosystems*. 2003, 72.
8. **Mathews, David H., Sabina, Jeffrey and Zuker, Michael.** Expanded Sequence Dependence of Thermodynamic Parameters Provides Improved Prediction of RNA Secondary Structure. *Journal of Molecular Biology*. 1999, Vol. 288, 5, pp. 911-940.
9. **Mathews Lab.** University of Rochester Medical Center. *RNAstructure*. [Online] May 10, 2011. [Cited: August 20, 2011.] <http://rna.urmc.rochester.edu/RNAstructure.html>.
10. **Zuker, Michael.** Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*. 2003, Vol. 31, 13, pp. 3406-3415.
11. **Griffiths-Jones, S, et al., et al.** Rfam: Annotating Non-Coding RNAs in Complete Genomes. *Nucleic Acids Research*. 2005, Vol. 33, pp. 121-141.
12. **Darty, Kévin, Denise, Alain and Ponty, Yann.** VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*. 2009, Vol. 25, 15, pp. 1974-1975.

Appendix A

Complete results

A.1 Search Results for Heuristic A with Subpalindromes

HBT Search Results					
Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00001_A	0.277778	0.3125	-35.6	-23.5	226768
RF00002_B	0	0	-29.3	-13.6	246788
RF00003_B	0	0	-33.8	-22.7	260664
RF00004_A	0.315789	0.4	-33.5	-38.9	269382
RF00005_A	0.461538	0.6	-33.6	-28.7	267015
RF00007_B	0.294118	0.322581	-32.4	-39.9	260580
RF00008_B	0.333333	0.266667	-14.6	-12.6	259454
RF00009_A	0.096774	0.204545	-53.2	-26.8	259115
RF00010_A	0.243902	0.25	-65.2	-72.3	248915
RF00011_B	0.081395	0.077778	-32.8	-50.1	253092
RF00012_B	0.44	0.372881	-48.6	-63.7	282035
RF00013_A	0.230769	0.226415	-59	-68	339512
RF00014_A	0.956522	1	-32.8	-31.7	388665
RF00015_B	0.236842	0.310345	-18.3	-23.7	387080
RF00017_B	0.563218	0.590361	-101.8	-58.5	400025
RF00019_A	0.318182	1	-11	-7.1	398233
RF00020_A	0.487179	0.655172	-32.4	-25	398144
RF00021_B	0.277778	0.27027	-31.6	1356.6	403257
RF00023_B	0.0875	0.089744	-57.3	-85.9	429000
RF00024_A	0.216	0.310345	-159.9	-122.4	400260
RF00025_A	0.208333	0.151515	-10.5	-14.8	370052
RF00026_B	0	0	-23.5	-4.5	364442
RF00028_B	0.036585	0.06	-47.2	-31.2	358152
RF00030_A	0.465909	0.554054	-73.7	-37.6	342547
RF00031_A	0.904762	1	-18.7	-16.6	304857
RF00032_B	1	1	-9.5	-9.5	1030
RF00035_B	0.37931	0.423077	-28.7	-30.8	300812
RF00036_A	0.375	0.324324	-108.9	-137.1	305442
RF00037_A	1	1	-6.8	-6.8	5981
RF00040_B	0.316456	0.297619	-53.2	-79.9	277474
RF00045_B	0.105263	0.125	-45.4	1364.5	261520
RF00048_A	0.3125	0.416667	-9.4	-7.3	235409

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00050_A	0.227273	0.434783	-56.5	-17.7	238406
RF00059_B	0.888889	0.8	-22.5	1372.3	241416
RF00061_B	0.09375	0.076923	-57.8	-83.2	233128
RF00094_A	0.24	0.285714	-29.6	-26.7	208191
RF00100_A	0.25	0.282051	-77.1	-69.8	213795
RF00102_B	0.255814	0.333333	-44.6	-20.2	181671
RF00106_B	0.475	0.558824	-41.7	-30	181317
RF00107_A	0.96	1	-38.2	-36.1	156901
RF00109_A	0	0	-7.9	-1.3	154242
RF00114_B	0.2	0.461538	-13.4	-11.6	169167
RF00140_B	0	0	-18.3	-9.8	203429
RF00161_A	1	0.826087	-21.1	-18.7	157156
RF00162_A	0.4375	0.518519	-28.4	-18.9	155887
RF00163_B	0.714286	0.769231	-10	-18.4	69320
RF00164_B	0.666667	0.666667	-11.8	-9.7	33102
RF00165_A	0	0	-13	-11	171049
RF00167_A	0.785714	1	-31.6	-25.5	190646
RF00168_B	0.581818	0.64	-62.8	-57.8	173119
RF00169_B	0	0	-19.7	-10.7	162031
RF00171_A	0.102041	0.172414	-29.4	-18.6	182331
RF00172_A	0.166667	0.333333	-16.4	-5.8	153459
RF00173_B	0.571429	1	-14.6	-6	155881
RF00175_B	0.357143	0.384615	-20.3	-31.9	181414
RF00176_A	0.416667	0.454545	-17.8	-19.6	167855
RF00177_A	0.348148	0.345588	-155	-152.9	258841
RF00179_B	0.375	0.333333	-2.8	-9.7	300284
RF00180_B	0.4	0.666667	-4.6	-0.6	28985
RF00181_A	0.217391	1	-13.4	-3.2	301397
RF00182_A	0	0	-31.5	-20.8	303662
RF00183_B	0.607143	0.653846	-29.3	-35.9	296738
RF00184_B	1	1	-5.6	-5.6	300461
RF00192_A	1	1	-7.7	-7.7	304247
RF00193_A	0.171429	0.190476	-64.3	-66.3	308942
RF00194_B	0	0	-25.2	-34.4	285053
RF00196_B	1	1	-13	-13	21542
RF00197_A	0.875	0.4375	-1.4	-4.1	273265
RF00198_A	0.533333	0.666667	-28.9	-18.2	278915
RF00199_B	0	0	-15.2	-12.4	262257
RF00207_B	0.916667	0.647059	-4.9	-7.8	253944
RF00209_A	0.056818	0.054945	-69.6	1294.9	269024
RF00210_A	0.272727	0.270492	-97.3	-135.7	243047
RF00214_B	0.791667	0.826087	-28.3	-24	173145

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00215_B	0.777778	1	-16.1	-14.7	190315
RF00216_A	0.197368	0.241935	-82.3	-97.2	196557
RF00220_A	0	0	-6.6	-18.3	155393
RF00225_B	0.157895	0.24	-15.9	1402.6	184928
RF00230_B	0	0	-21.9	-16.5	181373
RF00231_A	0.174603	0.186441	-43.9	-70.2	191353
RF00232_A	0.652174	0.555556	-64.5	1335.3	169289
RF00233_B	0.458333	0.578947	-21.9	-28	158153
RF00234_B	0.268293	0.297297	-32	-39.2	163226
RF00236_A	1	1	-43.6	-43.6	155746
RF00242_A	0.714286	1	-19	-15.6	162451
RF00250_B	0.952381	0.869565	-27.2	-29.3	157942
RF00252_B	0.735849	0.696429	-54.9	-56.1	180069
RF00259_A	0.315789	0.244898	-24.7	-28.4	154864
RF00260_A	1	1	-19.4	-19.4	153114
RF00261_B	0.224138	0.185714	-96	-111.9	177169
RF00264_B	0	0	-31	-45.9	161232
RF00286_A	0.225806	0.318182	-27.3	-20.3	163076
RF00290_A	0.548387	0.566667	-26	-28.4	152549
RF00362_B	0	0	-21.8	-25.8	150738
RF00363_B	0.84	0.875	-28.5	-27	150452
RF00364_A	1	1	-26	-26	169973
RF00365_A	0.916667	1	-31.3	-25.8	175657
RF00366_B	0.961538	1	-28.1	-24.2	171185
RF00367_B	1	1	-22.8	-22.8	157569
RF00373_A	0	0	-18.7	-0.2	155886
RF00374_A	0.5	0.419355	-34.5	-40.2	161330
RF00378_B	0.7	0.84	-39.2	-24.3	155826
RF00381_B	0.666667	1	-24.4	-22.2	166052
RF00382_A	0.642857	0.9	-28.5	-21.1	152050
RF00383_A	0.3	0.5625	-35.2	-16	153406
RF00384_B	0.636364	0.823529	-13.4	-6.8	160231
RF00385_B	1	0.866667	-18.7	-18.6	29023
RF00386_A	0.45	0.310345	-28.7	-29.9	151685
RF00387_A	0	0	-40.4	-53.6	170166
RF00388_B	0.805556	0.852941	-35.5	-35.2	152182
RF00389_B	0.142857	0.131579	-25	-40.5	155519
RF00390_A	0	0	-1.8	-0.2	451
RF00391_A	0.35	0.4375	-40.2	-36.1	162769
RF00433_B	0.354839	0.22449	-11.3	-25.3	154645
RF00434_B	0.30303	0.344828	-26.9	-18.3	152117
RF00435_A	0.4375	0.378378	-44.9	-53.3	170286

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00436_A	0.8	0.8	-14	-11.7	156747
RF00437_B	0	0	-20.8	-19.8	160677
RF00444_B	0.377778	0.607143	-41.1	-41.1	159315
RF00453_A	0.7	0.7	-7.7	-7.5	1211
RF00458_A	0.5	0.534884	-38	-31.8	166390
RF00460_B	0	0	-15.8	-3.7	154318
RF00461_B	0	0	-94.8	-130.6	186203
RF00462_A	0.692308	0.818182	-5.9	-4.8	151040
RF00463_A	0	0	-48.4	-55	157107
RF00465_B	0.3125	0.277778	-17.6	-24.2	150809
RF00466_B	0.48	0.387097	-29.9	-35	155449
RF00467_A	0.555556	1	-29.2	-17.8	161155
RF00480_A	0.555556	1	-23.9	-17.9	88237
RF00481_B	0.466667	0.388889	-39.5	-45.5	156273
RF00483_B	0.076923	0.125	-52.7	-23.8	156811
RF00484_A	0.147059	0.15625	-32	-34.3	159203
RF00485_A	0.419355	0.541667	-30.1	-21	154224
RF00487_B	0.254545	0.264151	-42	-50.5	169406
RF00488_B	0.090909	0.191781	-56.7	-47	220177
RF00489_A	0.52	1	-27.6	-19.9	199102
RF00490_A	0	0	-18.1	-17	211029
RF00491_B	0	0	-1.8	1.5	206886
RF00492_B	0.318182	0.4375	-59.5	-47.3	205043
RF00493_A	0.5	1	-12.7	-7.2	210073
RF00494_A	0.6875	0.6875	-17	-5.6	202375
RF00496_B	0	0	-5.1	-9	119826
RF00497_B	0.9375	1	-42.8	-39	203353
RF00498_A	0.785714	1	-13.7	-12.5	123100
RF00499_A	0.517241	0.535714	-34.5	-40.4	213070
RF00500_B	1	1	-16.9	-16.9	180390
RF00502_B	1	1	-16.9	-16.9	230579
RF00503_A	0.174603	0.15942	-26.5	-47.2	257423
RF00505_A	0	0	-15.9	-10.2	258747
RF00506_B	0.857143	0.769231	-39.1	-47.6	278913
RF00507_B	0.423077	1	-37.5	-20.6	301343

A.2 Search Results for Heuristic A without Subpalindromes

HBT Search Results					
Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00001_A	0.277778	0.3125	-35.6	-23.5	226768
RF00002_B	0	0	-29.3	-13.6	246788
RF00003_B	0	0	-33.8	-22.7	260664
RF00004_A	0.315789	0.4	-33.5	-38.9	269382
RF00005_A	0.461538	0.6	-33.6	-28.7	267015
RF00007_B	0.294118	0.322581	-32.4	-39.9	260580
RF00008_B	0.333333	0.266667	-14.6	-12.6	259454
RF00009_A	0.096774	0.204545	-53.2	-26.8	259115
RF00010_A	0.243902	0.25	-65.2	-72.3	248915
RF00011_B	0.081395	0.077778	-32.8	-50.1	253092
RF00012_B	0.44	0.372881	-48.6	-63.7	282035
RF00013_A	0.230769	0.226415	-59	-68	339512
RF00014_A	0.956522	1	-32.8	-31.7	388665
RF00015_B	0.236842	0.310345	-18.3	-23.7	387080
RF00017_B	0.563218	0.590361	-101.8	-58.5	400025
RF00019_A	0.318182	1	-11	-7.1	398233
RF00020_A	0.487179	0.655172	-32.4	-25	398144
RF00021_B	0.277778	0.27027	-31.6	1356.6	403257
RF00023_B	0.0875	0.089744	-57.3	-85.9	429000
RF00024_A	0.216	0.310345	-159.9	-122.4	400260
RF00025_A	0.208333	0.151515	-10.5	-14.8	370052
RF00026_B	0	0	-23.5	-4.5	364442
RF00028_B	0.036585	0.06	-47.2	-31.2	358152
RF00030_A	0.465909	0.554054	-73.7	-37.6	342547
RF00031_A	0.904762	1	-18.7	-16.6	304857
RF00032_B	1	1	-9.5	-9.5	1030
RF00035_B	0.37931	0.423077	-28.7	-30.8	300812
RF00036_A	0.375	0.324324	-108.9	-137.1	305442
RF00037_A	1	1	-6.8	-6.8	5981
RF00040_B	0.316456	0.297619	-53.2	-79.9	277474
RF00045_B	0.105263	0.125	-45.4	1364.5	261520
RF00048_A	0.3125	0.416667	-9.4	-7.3	235409

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00050_A	0.227273	0.434783	-56.5	-17.7	238406
RF00059_B	0.888889	0.8	-22.5	1372.3	241416
RF00061_B	0.09375	0.076923	-57.8	-83.2	233128
RF00094_A	0.24	0.285714	-29.6	-26.7	208191
RF00100_A	0.25	0.282051	-77.1	-69.8	213795
RF00102_B	0.255814	0.333333	-44.6	-20.2	181671
RF00106_B	0.475	0.558824	-41.7	-30	181317
RF00107_A	0.96	1	-38.2	-36.1	156901
RF00109_A	0	0	-7.9	-1.3	154242
RF00114_B	0.2	0.461538	-13.4	-11.6	169167
RF00140_B	0	0	-18.3	-9.8	203429
RF00161_A	1	0.826087	-21.1	-18.7	157156
RF00162_A	0.4375	0.518519	-28.4	-18.9	155887
RF00163_B	0.714286	0.769231	-10	-18.4	69320
RF00164_B	0.666667	0.666667	-11.8	-9.7	33102
RF00165_A	0	0	-13	-11	171049
RF00167_A	0.785714	1	-31.6	-25.5	190646
RF00168_B	0.581818	0.64	-62.8	-57.8	173119
RF00169_B	0	0	-19.7	-10.7	162031
RF00171_A	0.102041	0.172414	-29.4	-18.6	182331
RF00172_A	0.166667	0.333333	-16.4	-5.8	153459
RF00173_B	0.571429	1	-14.6	-6	155881
RF00175_B	0.357143	0.384615	-20.3	-31.9	181414
RF00176_A	0.416667	0.454545	-17.8	-19.6	167855
RF00177_A	0.348148	0.345588	-155	-152.9	258841
RF00179_B	0.375	0.333333	-2.8	-9.7	300284
RF00180_B	0.4	0.666667	-4.6	-0.6	28985
RF00181_A	0.217391	1	-13.4	-3.2	301397
RF00182_A	0	0	-31.5	-20.8	303662
RF00183_B	0.607143	0.653846	-29.3	-35.9	296738
RF00184_B	1	1	-5.6	-5.6	300461
RF00192_A	1	1	-7.7	-7.7	304247
RF00193_A	0.171429	0.190476	-64.3	-66.3	308942
RF00194_B	0	0	-25.2	-34.4	285053
RF00196_B	1	1	-13	-13	21542
RF00197_A	0.875	0.4375	-1.4	-4.1	273265
RF00198_A	0.533333	0.666667	-28.9	-18.2	278915
RF00199_B	0	0	-15.2	-12.4	262257
RF00207_B	0.916667	0.647059	-4.9	-7.8	253944
RF00209_A	0.056818	0.054945	-69.6	1294.9	269024
RF00210_A	0.272727	0.270492	-97.3	-135.7	243047
RF00214_B	0.791667	0.826087	-28.3	-24	173145

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00215_B	0.777778	1	-16.1	-14.7	190315
RF00216_A	0.197368	0.241935	-82.3	-97.2	196557
RF00220_A	0	0	-6.6	-18.3	155393
RF00225_B	0.157895	0.24	-15.9	1402.6	184928
RF00230_B	0	0	-21.9	-16.5	181373
RF00231_A	0.174603	0.186441	-43.9	-70.2	191353
RF00232_A	0.652174	0.555556	-64.5	1335.3	169289
RF00233_B	0.458333	0.578947	-21.9	-28	158153
RF00234_B	0.268293	0.297297	-32	-39.2	163226
RF00236_A	1	1	-43.6	-43.6	155746
RF00242_A	0.714286	1	-19	-15.6	162451
RF00250_B	0.952381	0.869565	-27.2	-29.3	157942
RF00252_B	0.735849	0.696429	-54.9	-56.1	180069
RF00259_A	0.315789	0.244898	-24.7	-28.4	154864
RF00260_A	1	1	-19.4	-19.4	153114
RF00261_B	0.224138	0.185714	-96	-111.9	177169
RF00264_B	0	0	-31	-45.9	161232
RF00286_A	0.225806	0.318182	-27.3	-20.3	163076
RF00290_A	0.548387	0.566667	-26	-28.4	152549
RF00362_B	0	0	-21.8	-25.8	150738
RF00363_B	0.84	0.875	-28.5	-27	150452
RF00364_A	1	1	-26	-26	169973
RF00365_A	0.916667	1	-31.3	-25.8	175657
RF00366_B	0.961538	1	-28.1	-24.2	171185
RF00367_B	1	1	-22.8	-22.8	157569
RF00373_A	0	0	-18.7	-0.2	155886
RF00374_A	0.5	0.419355	-34.5	-40.2	161330
RF00378_B	0.7	0.84	-39.2	-24.3	155826
RF00381_B	0.666667	1	-24.4	-22.2	166052
RF00382_A	0.642857	0.9	-28.5	-21.1	152050
RF00383_A	0.3	0.5625	-35.2	-16	153406
RF00384_B	0.636364	0.823529	-13.4	-6.8	160231
RF00385_B	1	0.866667	-18.7	-18.6	29023
RF00386_A	0.45	0.310345	-28.7	-29.9	151685
RF00387_A	0	0	-40.4	-53.6	170166
RF00388_B	0.805556	0.852941	-35.5	-35.2	152182
RF00389_B	0.142857	0.131579	-25	-40.5	155519
RF00390_A	0	0	-1.8	-0.2	451
RF00391_A	0.35	0.4375	-40.2	-36.1	162769
RF00433_B	0.354839	0.22449	-11.3	-25.3	154645
RF00434_B	0.30303	0.344828	-26.9	-18.3	152117
RF00435_A	0.4375	0.378378	-44.9	-53.3	170286

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00436_A	0.8	0.8	-14	-11.7	156747
RF00437_B	0	0	-20.8	-19.8	160677
RF00444_B	0.377778	0.607143	-41.1	-41.1	159315
RF00453_A	0.7	0.7	-7.7	-7.5	1211
RF00458_A	0.5	0.534884	-38	-31.8	166390
RF00460_B	0	0	-15.8	-3.7	154318
RF00461_B	0	0	-94.8	-130.6	186203
RF00462_A	0.692308	0.818182	-5.9	-4.8	151040
RF00463_A	0	0	-48.4	-55	157107
RF00465_B	0.3125	0.277778	-17.6	-24.2	150809
RF00466_B	0.48	0.387097	-29.9	-35	155449
RF00467_A	0.555556	1	-29.2	-17.8	161155
RF00480_A	0.555556	1	-23.9	-17.9	88237
RF00481_B	0.466667	0.388889	-39.5	-45.5	156273
RF00483_B	0.076923	0.125	-52.7	-23.8	156811
RF00484_A	0.147059	0.15625	-32	-34.3	159203
RF00485_A	0.419355	0.541667	-30.1	-21	154224
RF00487_B	0.254545	0.264151	-42	-50.5	169406
RF00488_B	0.090909	0.191781	-56.7	-47	220177
RF00489_A	0.52	1	-27.6	-19.9	199102
RF00490_A	0	0	-18.1	-17	211029
RF00491_B	0	0	-1.8	1.5	206886
RF00492_B	0.318182	0.4375	-59.5	-47.3	205043
RF00493_A	0.5	1	-12.7	-7.2	210073
RF00494_A	0.6875	0.6875	-17	-5.6	202375
RF00496_B	0	0	-5.1	-9	119826
RF00497_B	0.9375	1	-42.8	-39	203353
RF00498_A	0.785714	1	-13.7	-12.5	123100
RF00499_A	0.517241	0.535714	-34.5	-40.4	213070
RF00500_B	1	1	-16.9	-16.9	180390
RF00502_B	1	1	-16.9	-16.9	230579
RF00503_A	0.174603	0.15942	-26.5	-47.2	257423
RF00505_A	0	0	-15.9	-10.2	258747
RF00506_B	0.857143	0.769231	-39.1	-47.6	278913
RF00507_B	0.423077	1	-37.5	-20.6	301343

A.3 Search Results for Heuristic B with Subpalindromes

HBT Search Results					
Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00001_A	0.277778	0.30303	-31.9	-23.5	174922
RF00002_B	0	0	-24.8	-13.6	194939
RF00003_B	0.111111	0.142857	-32.2	-22.7	205086
RF00004_A	0.285714	0.428571	-30.9	-38.9	212878
RF00005_A	0.461538	0.6	-33.6	-28.7	208531
RF00007_B	0.277778	0.322581	-28.4	-39.9	223235
RF00008_B	0.833333	0.666667	-12.1	-12.6	225558
RF00009_A	0.073684	0.166667	-70	-26.8	276512
RF00010_A	0.233766	0.227848	-52.4	-72.3	318622
RF00011_B	0.073684	0.074468	-30.1	-50.1	340604
RF00012_B	0.290323	0.305085	-38.8	-63.7	340663
RF00013_A	0.226415	0.226415	-57.5	-68	341053
RF00014_A	0.607143	0.772727	-34	-31.7	328972
RF00015_B	0.275	0.37931	-9	-23.7	340315
RF00017_B	0.479592	0.566265	-100.7	-58.5	358811
RF00019_A	0.318182	1	-10.4	-7.1	340387
RF00020_A	0.487179	0.655172	-32.4	-25	344463
RF00021_B	0.411765	0.378378	-29.4	1356.6	343343
RF00023_B	0.083333	0.092105	-58.2	-85.9	365507
RF00024_A	0.10219	0.170732	-140.5	-122.4	437563
RF00025_A	0.212121	0.225806	-11.3	-14.8	405479
RF00026_B	0	0	-23.2	-4.5	408816
RF00028_B	0.035294	0.061224	-39.8	-31.2	462157
RF00030_A	0.553191	0.702703	-62.9	-37.6	464498
RF00031_A	0.863636	1	-20.1	-16.6	445030
RF00032_B	1	1	-9.5	-9.5	77109
RF00035_B	0.225806	0.269231	-27	-30.8	456738
RF00036_A	0.428571	0.401786	-109.2	-137.1	373579
RF00037_A	0.818182	0.9	-6.9	-6.8	352298
RF00040_B	0.284091	0.297619	-49.6	-79.9	375945
RF00045_B	0.115385	0.12766	-31.4	1364.5	361239
RF00048_A	0.625	0.833333	-14.6	-7.3	344247

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00050_A	0.232558	0.434783	-57.1	-17.7	355622
RF00059_B	0.8	0.8	-21.6	1372.3	358141
RF00061_B	0.085714	0.076923	-50.1	-83.2	374058
RF00094_A	0.392857	0.52381	-30.4	-26.7	355581
RF00100_A	0.236559	0.289474	-76.2	-69.8	366443
RF00102_B	0.340426	0.470588	-53.6	-20.2	361709
RF00106_B	0.736842	0.823529	-46.4	-30	351203
RF00107_A	0.96	1	-38.2	-36.1	342403
RF00109_A	0	0	-6.8	-1.3	340335
RF00114_B	0.2	0.461538	-7	-11.6	349139
RF00140_B	0	0	-16.3	-9.8	351425
RF00161_A	0.722222	0.565217	-16.5	-18.7	334778
RF00162_A	0.258065	0.32	-10.5	-18.9	344497
RF00163_B	0.714286	0.769231	-10	-18.4	347904
RF00164_B	0.666667	0.666667	-11.8	-9.7	2447784
RF00165_A	0.588235	1	-12.3	-11	333155
RF00167_A	0.785714	1	-31.6	-25.5	365039
RF00168_B	0.666667	0.84	-69.6	-57.8	379623
RF00169_B	0	0	-19.7	-10.7	366002
RF00171_A	0.102041	0.16129	-29.2	-18.6	367646
RF00172_A	0.24	0.5	-6.8	-5.8	366520
RF00173_B	0.571429	1	-14.6	-6	359832
RF00175_B	0.357143	0.416667	-14.7	-31.9	367317
RF00176_A	0.434783	0.47619	-9.4	-19.6	364312
RF00177_A	0.367647	0.367647	-155.4	-152.9	438059
RF00179_B	0.333333	0.277778	0.1	-9.7	376875
RF00180_B	0.4	0.666667	-4.6	-0.6	381746
RF00181_A	0.217391	1	-13.4	-3.2	375867
RF00182_A	0	0	-30.2	-20.8	386327
RF00183_B	0.607143	0.653846	-29.5	-35.9	380769
RF00184_B	1	1	-5.6	-5.6	381204
RF00192_A	1	1	-7.7	-7.7	384775
RF00193_A	0.246575	0.276923	-49.3	-66.3	393947
RF00194_B	0	0	-19.1	-34.4	376542
RF00196_B	1	1	-13	-13	381668
RF00197_A	1	0.875	-4.2	-4.1	381147
RF00198_A	0.5	0.666667	-28.3	-18.2	376065
RF00199_B	0	0	-5.2	-12.4	381007
RF00207_B	0.941176	0.941176	-7.8	-7.8	377256
RF00209_A	0.11236	0.111111	-70.6	1294.9	423169
RF00210_A	0.263566	0.280992	-92.8	-135.7	433480
RF00214_B	0.791667	0.826087	-28.3	-24	374820

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00215_B	0.777778	1	-16.1	-14.7	393154
RF00216_A	0.1875	0.25	-82	-97.2	441346
RF00220_A	0	0	-3.5	-18.3	424643
RF00225_B	0.166667	0.24	-13.8	1402.6	431815
RF00230_B	0	0	-33.1	-16.5	435576
RF00231_A	0.319444	0.370968	-61.1	-70.2	439701
RF00232_A	0.875	0.777778	-63.8	1335.3	427032
RF00233_B	0.458333	0.578947	-21.9	-28	425827
RF00234_B	0.243902	0.263158	-25.2	-39.2	428537
RF00236_A	0.583333	0.583333	-31.1	-43.6	429174
RF00242_A	0.516129	0.8	-16.4	-15.6	428524
RF00250_B	0.956522	0.956522	-29.3	-29.3	416897
RF00252_B	0.716981	0.690909	-48.4	-56.1	430506
RF00259_A	0.326531	0.326531	-17.6	-28.4	433303
RF00260_A	1	1	-19.4	-19.4	419044
RF00261_B	0.220339	0.185714	-94.4	-111.9	433868
RF00264_B	0	0	-23.3	-45.9	429933
RF00286_A	0.2	0.318182	-20.3	-20.3	434284
RF00290_A	0.540541	0.666667	-24.2	-28.4	431567
RF00362_B	0.357143	0.217391	-15.9	-25.8	423934
RF00363_B	0.875	0.875	-28.5	-27	423955
RF00364_A	1	1	-26	-26	420305
RF00365_A	0.916667	1	-31.3	-25.8	415587
RF00366_B	0.961538	1	-28.1	-24.2	410666
RF00367_B	1	1	-22.8	-22.8	410822
RF00373_A	0	0	-15.3	-0.2	426287
RF00374_A	0.258065	0.266667	-31.5	-40.2	424064
RF00378_B	0.6	0.84	-39.2	-24.3	425529
RF00381_B	0.666667	1	-24.4	-22.2	418488
RF00382_A	0.642857	0.9	-28.5	-21.1	424386
RF00383_A	0.25	0.642857	-24.5	-16	427187
RF00384_B	0.636364	0.823529	-13.4	-6.8	422968
RF00385_B	1	0.866667	-18.7	-18.6	420687
RF00386_A	0.533333	0.551724	-26	-29.9	420107
RF00387_A	0	0	-39.7	-53.6	429027
RF00388_B	0.805556	0.852941	-35.5	-35.2	428694
RF00389_B	0.138889	0.131579	-29.1	-40.5	427843
RF00390_A	0	0	-1.8	-0.2	46595
RF00391_A	0.25641	0.322581	-39.7	-36.1	429457
RF00433_B	0	0	-2.6	-25.3	432116
RF00434_B	0.285714	0.344828	-26.9	-18.3	429157
RF00435_A	0.4	0.388889	-45.3	-53.3	427524

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00436_A	0	0	-7.5	-11.7	419097
RF00437_B	0	0	-8.8	-19.8	430916
RF00444_B	0.509804	0.928571	-49.8	-41.1	434338
RF00453_A	0.777778	0.7	-8.2	-7.5	418054
RF00458_A	0.489362	0.534884	-34.9	-31.8	431959
RF00460_B	0	0	-12.8	-3.7	429041
RF00461_B	0	0	-92.9	-130.6	453148
RF00462_A	0.545455	0.545455	-1.9	-4.8	433558
RF00463_A	0	0	-42.4	-55	437312
RF00465_B	0	0	-19.3	-24.2	420296
RF00466_B	0.516129	0.5	-28.9	-35	429679
RF00467_A	0.517241	1	-30.9	-17.8	433326
RF00480_A	0.555556	1	-24	-17.9	409056
RF00481_B	0.515152	0.472222	-42.7	-45.5	431208
RF00483_B	0.076923	0.125	-52.7	-23.8	436887
RF00484_A	0.135135	0.16129	-28.2	-34.3	429907
RF00485_A	0.333333	0.545455	-27.4	-21	433405
RF00487_B	0.111111	0.115385	-34.7	-50.5	443718
RF00488_B	0.084848	0.197183	-47.7	-47	440773
RF00489_A	0.481481	1	-29	-19.9	389349
RF00490_A	0	0	-9.9	-17	382037
RF00491_B	0	0	0.8	1.5	382467
RF00492_B	0.297872	0.4375	-52.9	-47.3	392800
RF00493_A	0.5	1	-12.7	-7.2	382375
RF00494_A	0.55	0.6875	-16.1	-5.6	389848
RF00496_B	0	0	-4.1	-9	377696
RF00497_B	0.517241	0.5	-37.6	-39	389208
RF00498_A	0.785714	1	-13.7	-12.5	383678
RF00499_A	0.8	1	-41	-40.4	384447
RF00500_B	1	1	-16.9	-16.9	388315
RF00502_B	0.727273	0.666667	-14.4	-16.9	387497
RF00503_A	0.367816	0.470588	-34.9	-47.2	427159
RF00505_A	0	0	-15.9	-10.2	392100
RF00506_B	0.868421	0.846154	-40.2	-47.6	400208
RF00507_B	0.5	1	-28.3	-20.6	397266

A.4 Search Results for Heuristic B without Subpalindromes

HBT Search Results					
Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00001_A	0.277778	0.3125	-35.6	-23.5	147365
RF00002_B	0	0	-29.3	-13.6	157599
RF00003_B	0	0	-33.8	-22.7	157546
RF00004_A	0.315789	0.4	-33.5	-38.9	160069
RF00005_A	0.461538	0.6	-33.6	-28.7	159306
RF00007_B	0.294118	0.322581	-32.4	-39.9	189874
RF00008_B	0.333333	0.266667	-14.6	-12.6	207576
RF00009_A	0.096774	0.204545	-53.2	-26.8	210674
RF00010_A	0.243902	0.25	-65.2	-72.3	167781
RF00011_B	0.078652	0.075269	-32.8	-50.1	148296
RF00012_B	0.44	0.372881	-48.6	-63.7	144426
RF00013_A	0.230769	0.226415	-59	-68	145208
RF00014_A	0.88	1	-32.8	-31.7	144142
RF00015_B	0.236842	0.310345	-18.3	-23.7	147014
RF00017_B	0.550562	0.590361	-101.8	-58.5	146088
RF00019_A	0.318182	1	-11	-7.1	129945
RF00020_A	0.487179	0.655172	-32.4	-25	135099
RF00021_B	0.277778	0.27027	-31.6	1356.6	141238
RF00023_B	0.0875	0.089744	-57.3	-85.9	148738
RF00024_A	0.216	0.310345	-159.9	-122.4	171746
RF00025_A	0.208333	0.151515	-10.5	-14.8	193279
RF00026_B	0	0	-23.5	-4.5	222503
RF00028_B	0.036585	0.06	-47.2	-31.2	264387
RF00030_A	0.465909	0.554054	-73.7	-37.6	275519
RF00031_A	0.904762	1	-18.7	-16.6	260850
RF00032_B	1	1	-9.5	-9.5	821
RF00035_B	0.37931	0.423077	-28.7	-30.8	273173
RF00036_A	0.375	0.324324	-108.9	-137.1	366899
RF00037_A	1	1	-6.8	-6.8	7658
RF00040_B	0.316456	0.297619	-53.2	-79.9	367942
RF00045_B	0.105263	0.125	-45.4	1364.5	351420
RF00048_A	0.3125	0.416667	-9.4	-7.3	275200

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00050_A	0.227273	0.434783	-56.5	-17.7	285398
RF00059_B	0.888889	0.8	-22.5	1372.3	284011
RF00061_B	0.09375	0.076923	-57.8	-83.2	298505
RF00094_A	0.24	0.285714	-29.6	-26.7	281725
RF00100_A	0.25	0.282051	-77.1	-69.8	300991
RF00102_B	0.255814	0.333333	-44.6	-20.2	309289
RF00106_B	0.475	0.558824	-41.7	-30	381679
RF00107_A	0.96	1	-38.2	-36.1	377765
RF00109_A	0	0	-7.9	-1.3	386236
RF00114_B	0.2	0.461538	-13.4	-11.6	392798
RF00140_B	0	0	-18.3	-9.8	401328
RF00161_A	1	0.826087	-21.1	-18.7	403647
RF00162_A	0.4375	0.518519	-28.4	-18.9	401917
RF00163_B	0.714286	0.769231	-10	-18.4	159916
RF00164_B	0.666667	0.666667	-11.8	-9.7	74919
RF00165_A	0	0	-13	-11	398500
RF00167_A	0.785714	1	-31.6	-25.5	380032
RF00168_B	0.561404	0.64	-62.8	-57.8	391768
RF00169_B	0	0	-19.7	-10.7	386624
RF00171_A	0.102041	0.172414	-29.4	-18.6	404405
RF00172_A	0.166667	0.333333	-16.4	-5.8	395561
RF00173_B	0.571429	1	-14.6	-6	401429
RF00175_B	0.357143	0.384615	-20.3	-31.9	401574
RF00176_A	0.416667	0.454545	-17.8	-19.6	401635
RF00177_A	0.348148	0.345588	-155	-152.9	436474
RF00179_B	0.375	0.333333	-2.8	-9.7	392003
RF00180_B	0.4	0.666667	-4.6	-0.6	37786
RF00181_A	0.217391	1	-13.4	-3.2	377238
RF00182_A	0	0	-31.5	-20.8	389447
RF00183_B	0.607143	0.653846	-29.3	-35.9	379154
RF00184_B	1	1	-5.6	-5.6	395015
RF00192_A	1	1	-7.7	-7.7	384778
RF00193_A	0.171429	0.190476	-64.3	-66.3	398267
RF00194_B	0	0	-25.2	-34.4	389251
RF00196_B	1	1	-13	-13	30058
RF00197_A	0.875	0.4375	-1.4	-4.1	391186
RF00198_A	0.533333	0.666667	-28.9	-18.2	382238
RF00199_B	0	0	-15.2	-12.4	388107
RF00207_B	0.916667	0.647059	-4.9	-7.8	399535
RF00209_A	0.056818	0.054945	-69.6	1294.9	417824
RF00210_A	0.272727	0.270492	-97.3	-135.7	394935
RF00214_B	0.791667	0.826087	-28.3	-24	372069

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00215_B	0.777778	1	-16.1	-14.7	374048
RF00216_A	0.197368	0.241935	-82.3	-97.2	388747
RF00220_A	0	0	-6.6	-18.3	371868
RF00225_B	0.157895	0.24	-15.9	1402.6	388387
RF00230_B	0	0	-21.9	-16.5	389073
RF00231_A	0.174603	0.186441	-43.9	-70.2	389198
RF00232_A	0.652174	0.555556	-64.5	1335.3	377823
RF00233_B	0.458333	0.578947	-21.9	-28	373707
RF00234_B	0.268293	0.297297	-32	-39.2	377301
RF00236_A	1	1	-43.6	-43.6	370823
RF00242_A	0.714286	1	-19	-15.6	366518
RF00250_B	0.952381	0.869565	-27.2	-29.3	365804
RF00252_B	0.735849	0.696429	-54.9	-56.1	373146
RF00259_A	0.315789	0.244898	-24.7	-28.4	371536
RF00260_A	1	1	-19.4	-19.4	358034
RF00261_B	0.224138	0.185714	-96	-111.9	376657
RF00264_B	0	0	-31	-45.9	359252
RF00286_A	0.225806	0.318182	-27.3	-20.3	351404
RF00290_A	0.548387	0.566667	-26	-28.4	349331
RF00362_B	0	0	-21.8	-25.8	303407
RF00363_B	0.875	0.875	-28.5	-27	298218
RF00364_A	1	1	-26	-26	305007
RF00365_A	0.916667	1	-31.3	-25.8	308083
RF00366_B	0.961538	1	-28.1	-24.2	307650
RF00367_B	1	1	-22.8	-22.8	311426
RF00373_A	0	0	-18.7	-0.2	313988
RF00374_A	0.5	0.419355	-34.5	-40.2	309793
RF00378_B	0.7	0.84	-39.2	-24.3	313849
RF00381_B	0.666667	1	-24.4	-22.2	304268
RF00382_A	0.642857	0.9	-28.5	-21.1	305269
RF00383_A	0.3	0.5625	-35.2	-16	315941
RF00384_B	0.636364	0.823529	-13.4	-6.8	332625
RF00385_B	1	0.866667	-18.7	-18.6	59730
RF00386_A	0.45	0.310345	-28.7	-29.9	332806
RF00387_A	0	0	-40.4	-53.6	348677
RF00388_B	0.805556	0.852941	-35.5	-35.2	330825
RF00389_B	0.142857	0.131579	-25	-40.5	334638
RF00390_A	0	0	-1.8	-0.2	924
RF00391_A	0.35	0.4375	-40.2	-36.1	341965
RF00433_B	0.354839	0.22449	-11.3	-25.3	354324
RF00434_B	0.285714	0.344828	-26.9	-18.3	324742
RF00435_A	0.4375	0.378378	-44.9	-53.3	347396

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00436_A	0.8	0.8	-14	-11.7	336224
RF00437_B	0	0	-20.8	-19.8	355810
RF00444_B	0.377778	0.607143	-41.1	-41.1	349603
RF00453_A	0.7	0.7	-7.7	-7.5	3020
RF00458_A	0.5	0.534884	-38	-31.8	357712
RF00460_B	0	0	-15.8	-3.7	359141
RF00461_B	0	0	-94.8	-130.6	254673
RF00462_A	0.692308	0.818182	-5.9	-4.8	255023
RF00463_A	0	0	-48.4	-55	251935
RF00465_B	0.3125	0.277778	-17.6	-24.2	249653
RF00466_B	0.48	0.387097	-29.9	-35	250987
RF00467_A	0.555556	1	-29.2	-17.8	251594
RF00480_A	0.555556	1	-23.9	-17.9	140036
RF00481_B	0.466667	0.388889	-39.5	-45.5	259277
RF00483_B	0.076923	0.125	-52.7	-23.8	251045
RF00484_A	0.147059	0.15625	-32	-34.3	252205
RF00485_A	0.419355	0.541667	-30.1	-21	253091
RF00487_B	0.254545	0.264151	-42	-50.5	256686
RF00488_B	0.090909	0.191781	-56.7	-47	248575
RF00489_A	0.52	1	-27.6	-19.9	210568
RF00490_A	0	0	-18.1	-17	220409
RF00491_B	0	0	-1.8	1.5	209792
RF00492_B	0.318182	0.4375	-59.5	-47.3	212573
RF00493_A	0.5	1	-12.7	-7.2	211930
RF00494_A	0.6875	0.6875	-17	-5.6	211865
RF00496_B	0	0	-5.1	-9	120585
RF00497_B	0.9375	1	-42.8	-39	215958
RF00498_A	0.785714	1	-13.7	-12.5	121024
RF00499_A	0.517241	0.535714	-34.5	-40.4	205833
RF00500_B	1	1	-16.9	-16.9	164238
RF00502_B	1	1	-16.9	-16.9	207271
RF00503_A	0.174603	0.15942	-26.5	-47.2	208470
RF00505_A	0	0	-15.9	-10.2	193419
RF00506_B	0.857143	0.769231	-39.1	-47.6	207210
RF00507_B	0.423077	1	-37.5	-20.6	221823

A.5 Search Results for Heuristic C

HBT Search Results					
Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00001_A	0.277778	0.3125	-31	-23.5	143664
RF00002_B	0	0	-20.8	-13.6	156549
RF00003_B	0	0	-30.2	-22.7	163090
RF00004_A	0.26087	0.428571	-26.7	-38.9	178603
RF00005_A	0.461538	0.6	-33.6	-28.7	183038
RF00007_B	0.243902	0.322581	-24.7	-39.9	198715
RF00008_B	0.625	0.666667	-10	-12.6	203320
RF00009_A	0.070707	0.179487	-71.9	-26.8	262999
RF00010_A	0.222222	0.230769	-49.7	-72.3	257552
RF00011_B	0.073684	0.074468	-29.2	-50.1	308363
RF00012_B	0.290323	0.305085	-38.8	-63.7	291799
RF00013_A	0.206897	0.226415	-53.1	-68	311096
RF00014_A	0.607143	0.772727	-30.4	-31.7	358625
RF00015_B	0.244444	0.37931	-1.4	-23.7	379679
RF00017_B	0.5	0.566265	-100.8	-58.5	399109
RF00019_A	0.333333	1	-9.8	-7.1	338258
RF00020_A	0.475	0.655172	-30.2	-25	338183
RF00021_B	0.378378	0.378378	-27.5	1356.6	341117
RF00023_B	0.079545	0.089744	-57	-85.9	402061
RF00024_A	0.10219	0.17284	-142.1	-122.4	529222
RF00025_A	0.2	0.233333	-5.5	-14.8	408606
RF00026_B	0	0	-19.6	-4.5	401865
RF00028_B	0.034091	0.0625	-34.9	-31.2	472724
RF00030_A	0.483871	0.608108	-67	-37.6	494566
RF00031_A	0.76	1	-18.1	-16.6	408093
RF00032_B	1	1	-9.5	-9.5	11248
RF00035_B	0.212121	0.269231	-20.5	-30.8	424870
RF00036_A	0.416667	0.401786	-109.9	-137.1	479948
RF00037_A	0.909091	1	-6.9	-6.8	405308
RF00040_B	0.287356	0.297619	-52.6	-79.9	495510
RF00045_B	0.090909	0.125	-39.6	1364.5	451781
RF00048_A	0.5	0.833333	-13.2	-7.3	407316

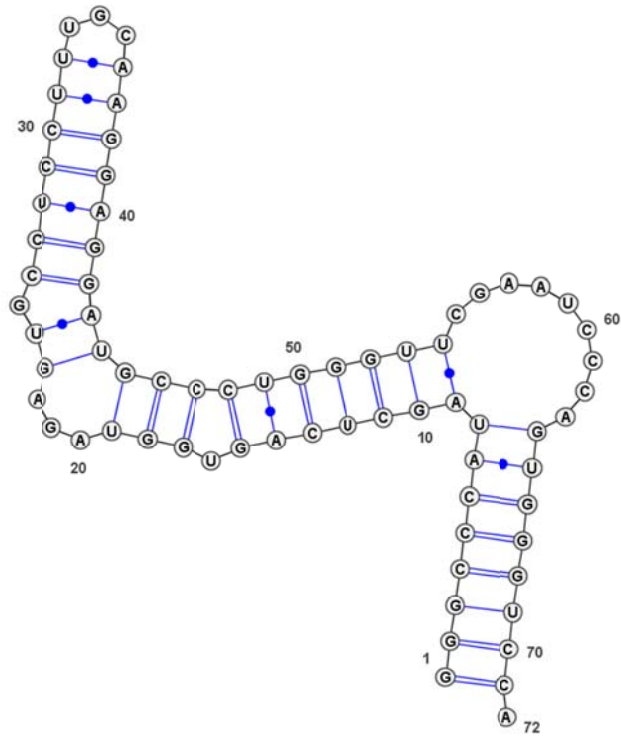
Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00050_A	0.217391	0.434783	-60	-17.7	420798
RF00059_B	0.606061	0.666667	-13	1372.3	414134
RF00061_B	0.075	0.08	-41.8	-83.2	374843
RF00094_A	0.357143	0.47619	-26.6	-26.7	348787
RF00100_A	0.236559	0.289474	-76.2	-69.8	399994
RF00102_B	0.340426	0.484848	-56.2	-20.2	354604
RF00106_B	0.710526	0.794118	-43.2	-30	358185
RF00107_A	0.96	1	-38.2	-36.1	354380
RF00109_A	0	0	-1.1	-1.3	349779
RF00114_B	0.176471	0.461538	-5.8	-11.6	360125
RF00140_B	0	0	-13.4	-9.8	362438
RF00161_A	1	0.956522	-19.5	-18.7	350283
RF00162_A	0.242424	0.307692	-9	-18.9	352620
RF00163_B	0.714286	0.769231	-10	-18.4	337068
RF00164_B	0.666667	0.666667	-11.8	-9.7	343177
RF00165_A	0.5625	0.9	-12.3	-11	344858
RF00167_A	0.6875	1	-28.5	-25.5	353878
RF00168_B	0.666667	0.84	-69.6	-57.8	364648
RF00169_B	0	0	-18.1	-10.7	358623
RF00171_A	0.1	0.16129	-25.2	-18.6	330843
RF00172_A	0.208333	0.416667	-7.4	-5.8	325829
RF00173_B	0.5	1	-12.8	-6	317275
RF00175_B	0.333333	0.384615	-13	-31.9	328277
RF00176_A	0.4	0.47619	-7.7	-19.6	313019
RF00177_A	0.364964	0.367647	-146.7	-152.9	618578
RF00179_B	0.25	0.3125	3	-9.7	375645
RF00180_B	0.4	0.666667	-4.6	-0.6	379876
RF00181_A	0.217391	1	-13.4	-3.2	380717
RF00182_A	0	0	-18.6	-20.8	400408
RF00183_B	0.607143	0.653846	-28.3	-35.9	389905
RF00184_B	0.8	1	-2.7	-5.6	386977
RF00192_A	1	1	-7.7	-7.7	380317
RF00193_A	0.2	0.230769	-50.3	-66.3	442590
RF00194_B	0	0	-20.6	-34.4	381873
RF00196_B	1	1	-13	-13	387388
RF00197_A	1	1	-4.1	-4.1	384395
RF00198_A	0.484848	0.666667	-26.6	-18.2	405449
RF00199_B	0	0	-3.6	-12.4	400494
RF00207_B	0.941176	0.941176	-7.8	-7.8	393706
RF00209_A	0.103093	0.111111	-65.9	1294.9	497221
RF00210_A	0.257576	0.283333	-92.8	-135.7	495271
RF00214_B	0.846154	0.956522	-24.2	-24	366644

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00215_B	0.7	1	-14.9	-14.7	347056
RF00216_A	0.219512	0.3	-81.8	-97.2	441734
RF00220_A	0	0	-7.2	-18.3	381641
RF00225_B	0.166667	0.24	-13.8	1402.6	388487
RF00230_B	0	0	-33.1	-16.5	442958
RF00231_A	0.302632	0.370968	-55	-70.2	424105
RF00232_A	0.777778	0.792453	-57.3	1335.3	401852
RF00233_B	0.458333	0.578947	-21.9	-28	381944
RF00234_B	0.232558	0.27027	-21.4	-39.2	397213
RF00236_A	0.8	1	-38.7	-43.6	384609
RF00242_A	0.516129	0.8	-16.4	-15.6	393587
RF00250_B	0.956522	0.956522	-29.3	-29.3	389991
RF00252_B	0.678571	0.690909	-46.1	-56.1	409860
RF00259_A	0.355556	0.326531	-19.3	-28.4	402688
RF00260_A	1	1	-19.4	-19.4	382280
RF00261_B	0.203125	0.19403	-97.4	-111.9	441336
RF00264_B	0	0	-23	-45.9	410247
RF00286_A	0.205882	0.318182	-23.9	-20.3	406729
RF00290_A	0.404762	0.566667	-12.9	-28.4	409527
RF00362_B	0.277778	0.217391	-12.3	-25.8	386210
RF00363_B	0.84	0.875	-26.3	-27	384276
RF00364_A	1	1	-26	-26	387258
RF00365_A	0.916667	1	-31.3	-25.8	383364
RF00366_B	0.862069	1	-26.4	-24.2	375486
RF00367_B	0.904762	0.863636	-20.3	-22.8	376311
RF00373_A	0	0	-10.9	-0.2	402927
RF00374_A	0.258065	0.266667	-28.6	-40.2	408895
RF00378_B	0.6	0.84	-39.2	-24.3	398820
RF00381_B	0.6	1	-23.6	-22.2	392258
RF00382_A	0.642857	0.9	-28.5	-21.1	387371
RF00383_A	0.257143	0.5625	-23	-16	406961
RF00384_B	0.636364	0.823529	-13.4	-6.8	384505
RF00385_B	1	0.866667	-18.7	-18.6	379637
RF00386_A	0.516129	0.551724	-29.8	-29.9	401082
RF00387_A	0	0	-36.5	-53.6	404008
RF00388_B	0.888889	0.941176	-34.8	-35.2	400617
RF00389_B	0.102564	0.108108	-29.1	-40.5	409018
RF00390_A	1	1	-0.2	-0.2	6774
RF00391_A	0.25641	0.322581	-39.7	-36.1	396623
RF00433_B	0.15625	0.102041	-5.5	-25.3	409585
RF00434_B	0.27027	0.344828	-23.8	-18.3	407415
RF00435_A	0.388889	0.378378	-43.3	-53.3	397986

Structure Name	Precision	Recall	Predicted Energy	Known Energy	Run Time(ms)
RF00436_A	0	0	-5.5	-11.7	382796
RF00437_B	0	0	-4.8	-19.8	368495
RF00444_B	0.509804	0.928571	-49.7	-41.1	385818
RF00453_A	0.777778	0.7	-8.2	-7.5	345040
RF00458_A	0.425926	0.534884	-22.9	-31.8	387903
RF00460_B	0	0	-10.2	-3.7	355353
RF00461_B	0	0	-89	-130.6	417697
RF00462_A	0.5625	0.818182	-4.2	-4.8	355839
RF00463_A	0	0	-45.3	-55	359279
RF00465_B	0	0	-18.3	-24.2	352332
RF00466_B	0.470588	0.5	-28.2	-35	368106
RF00467_A	0.517241	1	-30.9	-17.8	352918
RF00480_A	0.5	0.9	-24.9	-17.9	354445
RF00481_B	0.529412	0.5	-39.6	-45.5	358011
RF00483_B	0.076923	0.125	-52.7	-23.8	379856
RF00484_A	0.121951	0.16129	-22.2	-34.3	369391
RF00485_A	0.333333	0.545455	-27.4	-21	373145
RF00487_B	0.095238	0.111111	-33.3	-50.5	379383
RF00488_B	0.084337	0.202899	-39.6	-47	441969
RF00489_A	0.481481	1	-29.3	-19.9	371209
RF00490_A	0	0	-17.1	-17	359788
RF00491_B	0	0	3.5	1.5	359048
RF00492_B	0.27451	0.4375	-45.3	-47.3	374989
RF00493_A	0.434783	0.909091	-7.8	-7.2	357918
RF00494_A	0.52381	0.6875	-12.1	-5.6	354476
RF00496_B	0	0	-4.1	-9	353129
RF00497_B	0.46875	0.5	-33.7	-39	362396
RF00498_A	0.785714	1	-13.7	-12.5	349528
RF00499_A	0.8	1	-41	-40.4	358074
RF00500_B	0.923077	1	-14.7	-16.9	357112
RF00502_B	0.642857	0.75	-11.8	-16.9	362827
RF00503_A	0.355556	0.463768	-35.2	-47.2	433406
RF00505_A	0	0	-15.9	-10.2	372431
RF00506_B	0.825	0.846154	-35.1	-47.6	377701
RF00507_B	0.37931	1	-35.8	-20.6	375226

30	C	29	31	38	30
31	U	30	32	37	31
32	U	31	33	36	32
33	U	32	34	0	33
34	G	33	35	0	34
35	C	34	36	0	35
36	A	35	37	32	36
37	A	36	38	31	37
38	G	37	39	30	38
39	G	38	40	29	39
40	A	39	41	28	40
41	G	40	42	27	41
42	G	41	43	26	42
43	A	42	44	24	43
44	U	43	45	23	44
45	G	44	46	19	45
46	C	45	47	18	46
47	C	46	48	17	47
48	C	47	49	15	48
49	U	48	50	14	49
50	G	49	51	13	50
51	G	50	52	12	51
52	G	51	53	11	52
53	U	52	54	10	53
54	U	53	55	9	54
55	C	54	56	0	55
56	G	55	57	0	56
57	A	56	58	0	57
58	A	57	59	0	58
59	U	58	60	0	59
60	C	59	61	0	60
61	C	60	62	0	61
62	C	61	63	0	62
63	A	62	64	0	63
64	G	63	65	8	64
65	U	64	66	7	65
66	G	65	67	6	66
67	G	66	68	5	67
68	G	67	69	4	68
69	U	68	70	3	69
70	C	69	71	2	70
71	C	70	72	1	71
72	A	71	73	0	72

Visualization of predicted structure:



GGUCUUUUUACCUGUUCGGUUUAUGGACCGCUAGAGGUGGCUAGUAAUAGCCAUCCAGAGAGAUAAACUGCC
CUCUGUCUUCGACAGAGAACAGAACCCGGCUUA

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

> RF00011_B

GUGCAAUUUUUUGGAUAAUCGCGUAGUUAUUGCAAUACUAUGAGGAAAGUCCAUGCUAGCACC GGCU
GUGAUGCUGGUAGUGUUUGUGCUAGGCGAACAAAUAAGCCUAGGGGAUGUGUUUGACACAUUACGGCGAGU
GAAAAGGCUAAGUCCUUGAUUAGCCAGAGUAGCUCUGAAAGUGCCACAGUGACGAAGUUUUUAUGGAAAC
GUAAAAGUGGAACGCGGUAAACCCCUCAAGCUAGCAACCAAAACUUUGGUAGGGGCAUGGGGAUAGUUGG
AAACGAACAAGCUAUCCUGACUGUUGACAAACAGUAGACAGAUAUUAUCGAAGGAGAUAAUACCUAGU
AUUCUGGAACAAAACAUGGCUUAUAGAA

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

> RF00012_B

AAGACUAUACUUUCAGGGAUCAUUUCUAUAGGUUGUACCUGGUGAAAUGUGCUCGAAAGUGUCUG
AACUCACAAACCACGAGGAAGAGCGUCAGUGUUUCUCCUGAGCGUGAAGUGAGCUCACAGUGCUGCUUCA
UUGUGGCUGCUGUUUGCUAUUGAUGAACGUUCUGCUCCCCUUUAUUAUUGGGGAGAUAGAGGGAGAGAAC
ACAAGCUGAGUGG

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

> RF00013_A

AUUUCUCUGAGAUGUUUGCAAGCGGGCCAGUCCCGAGCCGAUAAUUCUAUACCACAAGAAUGUGG
CGCUCGCGGUUGGUGAGCAUGCUCGGUUCGUCCGAGAAGCCUAAAACUGUGACGACACAUUCACCUUGA
ACCAAGGGUUAAGGGUACAGCCUGCGGCGGCAUCUCGGAGAUUC

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

> RF00014_A

AACGCAUCGGAUUUCCCGGUGUAACGAAUUUUAAGUGCUUCUUGCAUUAAGCAAGUUUGAUCCCG
ACUCCUGCGAGUCGGGAUUU

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

> RF00015_B

AGCUUUGCGCAGUGGCAGUAUCGUAGCCAAUGAGGGUCUAUCCGAGGCGCGAUUAUUGCUAAUUGA
AAACUUUCCCAAUACCCCGCCGUGACGACUUGCAUAUAGUCGGCACUGGCAAUUUUUGACAGUCUCUAC
GGAGA

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

> RF00017_B

GACGGGCUUAGCAACGUGGGCCUGUAACCCAAGUGGGGGCAUGUGGGAAAUGGGACUUUGGGUCA
ACCUAGUGGAUCGGGUCCAGUGUUAGCUGCUUACUGGUCUGCCAUUCCAAGCCGGGAGUUGGGCUGAGUG
ACCUAGGGCGAAGGCUGGGUUGCGCAGCUCCUAGAGUGGAGGGCAAUGCGUGAGGCUGGCUUCACAGAGCAG
CGACUACCUCCGCUUCGCGCAGUGGAAGGAUACGGGCCGUGCUACCUUGGAUCCACCAUGCUUCACUGG
GCUGACUCUAAUAGGACCAUUUCUUU

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

> RF00019_A

GGUUGGUCCGCGAGUAGUGGUGUUACAACUAAUUGAUCACAACCAGUUAACAGAUUUCUUGUUC
CUUCUCCACUCCACUGCU

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

> RF00020_A

AUACUCUGGUUUCUCUUCAGAUUCGUAUAAAUCUUUCGCCUUUUACUAAAAGAUUUCGGUGGAGAGG
AACAAACCACGAGUGUCGUGGAAUUUUUUGAGGCUCGCCUUCGGCGGAGCU
..(((.....((((((((.....((((((((.....)))))))))...)))))).....)))).....((((.....))))))

> RF00021_B
AUUAGCUCGCGUAGGGUACAGAGGUAAGAUGUUCUAUCUUUCAGACCUUUUGUUUCACGUUAUUG
GAUUAGGCUGAUUCAGCCGCCCAGUCACCAUUUGACUGGGGGCGUUUUUA
.....((((.....((((((((.....)))))))))...)))))).....)))).....((((.....)))))).....

> RF00023_B
GGGGCUGAUUCUGGAUUCGACGGGAUUUGCGAAACCCAAGGUGCAUGCCGAGGGGCGGUUGGCCU
CGUAAAAAGCCGCAAAAAUAGUCGCAAACGACGAAAACUACGCUUAGCAGCUUAAUAACCUGCUUAGAG
CCCUCUCUCCCUAGCCUCCGUCUCUAGGACGGGGAUCAAGAGAGGUCAAACCCAAAAGAGAUCGCGUGGAA
GCCUCUGCCUGGGGUUGAAGCGUUA AACUUAUCAGGCUAGUUUGUUAGUGGCGUGUCCGUCGCCAGCUGG
CAAGCGAAUGUAAAGACUGACUAAGCAUGUAGUACCGAGGAUGUAGGAAUUUCGGACGCGGGUUC AACUC
CCGCCAGCUCCACCA
((((.....((((.....((((.....))))))))).....)))).....((((.....)))))).....
))..)))).....)))))).....)))).....)))).....)))).....)))).....)))).....)))).....
(.....)))))).....

> RF00024_A
GGGUUGCGGAGGGUGGGCCUGGGAGGGUGGUGGCCAUUUUUUGUCUAACCCUAACUGAGAAGGG
CGUAGGCGCCGUGCUUUUGCUCUCCCGCGCGCUGUUUUUCUCGUCAGCUUUCAGCGGGCGGAAAAGCCUCGG
CCUGCCGCCUUCACCGUUCAUUCUAGAGCAAACAAAAAUGUCAGCUGCUGGCCCGUUCGCCCUCCCGG
GGACCUGCGGCGGGUCGCCUGCCAGCCCCGAACCCGCGGAGGCGCGGUCGGCCCGGGGCUUCUCCG
GAGGCACCCACUGCCACCGCGAAGAGUUGGGCUCUGUCAGCCGCGGGUCUCUCGGGGGCGAGGGCGAGGUU
CAGGCCUUUCAGGCCGAGGAAGAGGAACGGAGCGAGUCCCCGCGCGCGGCGCGAUUCCUGAGCUGUGGG
ACGUGCACCCAGGACUCGGCUCACACAUGC
.....((((.....((((.....((((.....))))))))).....)))).....)))).....)))).....
.....)))).....)))).....)))).....)))).....)))).....)))).....)))).....)))).....
(.....)))))).....

> RF00025_A
AUACCCGCAAAUUCACUCAAUCUGUAAUAGGUUUGUCAUUCACCCCAAAAUCUAGUGCAAUA
UUACUUUCGCCAAUAGGUAAUAAUAGGUAAGCGGGACAAGACUCGACAUUUGAUACACUAUUUAUCA
UGGAUGUCUUUUUCU
.....((((.....((((.....))))))))).....)))).....)))).....)))).....)))).....

> RF00026_B
UGCAGUUUGCUCGCUAUUAGUUUGGAACAACACUGAGAAGAUUAGCAUGGCCCCUGCGCAAGGA
CGGCAUCUUUCUUGAGAGGUGUGCUGGGCUCGCCAGCUUU
((((.....)))).....

> RF00028_B
AUGACUCUAAAUUGCAAAAUUUACCUUUGGAGGGAAAAGUUAUCAGGCCUGCACCUGAUAGCU
AGUCUUUAAACCAAUAGAUUGCAUCGCUUUAAUAGGCAAGACCGUCAAAUUGCGGGAAAAGGGUCAACAG
CCGUUCAGUACCAAGUCUCAGGGGAAACUUUGAGAUGGCCUUGCAAAGGAUAUGGUAUAAGCUGACGGA
CAGGGUCCUAACCACGCAGCCAAGUCCUAAGUCAACAUUUCGGUGUUGAUUAGGAUGCAGUUCACAGACUA
AAUGUCGGUCGGGGAAGAAUAGGUUUCUUCUAUAAGAUUAUGUCGGACCUCUCCUUAUUGGGAGCU
.....((((.....((((.....))))))))).....)))).....)))).....)))).....)))).....
.....)))).....)))).....)))).....)))).....)))).....)))).....)))).....)))).....

> RF00030_A
AUCGUCACAAAUCGAAGCUACAAAUGGAGUAAAUUUUUUUACUCAGUAAUAUGCUUUGGGU
UGAAAGUCUCCACCAAUUCGUAUGCGGAAAACGUAAUGAGAUUUAAAAUUCUAAAUUGUUAAAUCAA
CUCAUUAAGGAGGAUGCCCUUGGGUAUUCUGCUUCUUGACCUGGUACCUCUAUUGCAGGGUACUGGUGU
UUCUUCGGUACUGGAUUCGGUUUGUAUGGAAUCUAAACCAUAGUUAUGACGAUUGCUCUUUCCCGUGCUG

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

> RF00175_B

GGACUCGGCUUGCUGAGGUGCACACAGCAAGAGGCGAGAGCGGCGACUGGUGAGUACGCCAAAUU
UUGACUAGCAGAGGCUAGAAGGAGAGAGAUGGGUGCGAGAGCGUCAUA
...((((.....)))).....((((.....)))).....((((.....)))).....

> RF00176_A

GACCUAGACACGGUUGAUCUCACCUUCGGGGGGGCUAUAGAGAUCGCUGGAAGCAUUACCGGACAA
CCGGAACAUUGCAGCAAUGCAGCCC
.....((((.....)))).....((((.....)))).....

> RF00177_A

AGGUCACUGCUAUCGGAGUCCGAUUUAGCCAUGCUAGUCCACCGAGCCCUUGCGGCUUGGGGCGGA
UAGCUGAGUAGCACGUGGCUAACCUACCCUGUGGACCGGAAUAACCUUCGGGAAACUGAGGCUAAUAGCCGGA
UAAGAGAGGGAUACUGGAAAGUCCUCUCAGAAAUGCUACGGCGCCACAGGAUGGGGCCGCGGCUAUA
GGUUGUUGGUGAGGUAACGGCUCACCAAGCCAGUGAUCAGUAACAGCCGUGGAAGCGGGAGCUGUCAGUC
GUGUUCUGAGACAAGAACACGGGCCCUACGGGGCGCAGCAGGCACGAAACCCUCACAAUGCACGAGAGUGC
GAUGGGGGGACUCCGAGUGCGAUCUCCACAAGGAAGGUGCGUUUCUGCAGUGUAAGGAGCUGCAGGA
ACAAGGGUGGGCAAGACGGCUGCCAGCCCGCGGUAUAGCCGCACCCCGAGUGGUGACCG
.....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....

> RF00179_B

GUAUGUGAAAGUGUAAAUAAGAAUGUACUUGGAAUGACUAUAAACAUUGAAAGAAGACUGGAA
ACAUAC
((((.....)))).....

> RF00180_B

CUGC UUUGAGGACAAAGAGAAUAAAGACUUCAUGUUC
((((.....)))).....

> RF00181_A

UGGAUCAUUGAUGAACAAAAAAAAAAACAUCUGGGAGUCCUCUGAGACAUCAUGAUGACCACAA
CAUUGGGAGUCUGAGGUCCAC
.....

> RF00182_A

GAGGGUCAGGAUGUCAUCUUCAGCCAAUUCGACAGCCUGAGAGUCAGCUCUAACCAGAGCCCACAA
GGUAAUCUGGGGAGUAAUGAACCCGGUAAUGUCGGUGGUAUUGAUGCUCUGGCAACCUCCACU
.....

> RF00183_B

CUGAGGGUCCCAACCUUGGACCCUUGAGAGUAUCAGGUCUCCACGUGGGAGACAAGAAAUCCUG
UUUAAUUAUAAACAGAGUUCUCC
.....

> RF00184_B

CCAUAUUACGUCUACAUAACCGACGCCUACCCAGUUUCAUAGUAUUUUCUGG
.....

> RF00192_A

AUGGGAAAUUCCCCUCCUAUAACCCCGCUGGUAUCUCCCCUCAGACUGGC
.....

> RF00193_A

> RF00234_B
 AUUAUAGUUAAAAGCGCCCGAACUUGAGUGAUUAAGUUCAUUUAAGUUGACGAGGAUGGGGAGAAU
 CGAAUCUUCGGCGGAUGCCCCACGGUACCGCACUACCGUUAAGCGGUUGGUAAAAGCAGAAAGUGAUUUCUG
 UCACAAAGCCAAUCUGGUGUAAA
((((((((.....))))))))).....((((.....((((.....)))))))))((((.....)))).....((((.....((((((((.....))))))))).....)))))).....

> RF00236_A
 GCUUUCUCCCAGGUUUGGACUGGGGGUAGCCGACGCCUGUGAGUUACCGCUCACGGGGCGUUCA
 ACAUUUUUCAGGUAAU
 ((.....((((.....))))))..)).((((((((.....))))))))).....

> RF00242_A
 AUACAAGAUUAUAAAAACAACUCAGUGUUUUUUCUUGAAUGAUGUCGUUCACAAACUUUGGUC
 AGGGCGUGAGCGACUCCUUUUUAUUUUGUU
((((.....((((.....))))))))).....((((.....)))).....

> RF00250_B
 GGUUUCUCUUGUAGACCAGGUCGAGCCCGGGAGCUCUCUGGCUAGCAGGGGAACC
 ((.....((((.....)))))).....

> RF00252_B
 UGAUGUACCCCAUUAUUUGGGAUGCUAAAGUCAUUUAAUGCUGACCUCACUGAGUGGAUUAAG
 GUCAAGGUAUGAAGUCCUAUUCGCUCCUGAUAGGAACGACUUCAUAUUGCUUAUAUUGUCUAACGCAC
 AUUAUAAAUGCUC AUGCAAACUGCAUGAAUGCCCCUAAGGGGAUGC
((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....

> RF00259_A
 CAUUGUUCUGAUCAUCUGAAGAUCAGCUAUUAGAAGAGAAAGAUCAUUAAGUCCUUUGGACCUG
 AUCAGCUUGAUACAAGAACUACUGAUUUAACUUCUUGGCUUAAUUCUCUCGAAACGAUGAAAUAUAC
 AAGUUAUAUCUUGGCUUUUCAGCUCUGCAUCGU
((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....

> RF00260_A
 UACAGCGGGGAGACAUAUAUCACAGCCUGUCUCGCGCCCGACCCUGCUGG
 ..((((.....)))).....

> RF00261_B
 AAUGCGCCUGCAGCUCGCGCUCGCGCCGAUCCCGAGAGCGUCCGGGCCGCGUGCGCGAGCGAG
 GGAGGGCGCGCGCGCGGGGGGGCGCGCUCGUGAGUGCGGGCCGCGCUCUCGCGCGCGCAUGUGCGUGU
 GUGCUGGCUGCCGGGCGUCCCCGAGCCGCGGGGAGCCGGUCCGCUCAGGUGGCGGGCGGCGUGGAGCGAG
 GUGAGGCUGCGGG
((.....((((.....)))).....))))..)).((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....((((.....)))).....

> RF00264_B
 ACUCUCUCGGCUCUGCAUAGUUGCACUUGGCUUACCCGUGUGACUUCGUAACGGGGAGAGAGAG
 AAAAGAUCUCCUCAGGACCUCGGAUGGGCCUACUGUGGCCUCUCUUCUUGAGGGGUGCAACAGGC
 .((((.....)))).....

> RF00286_A
 CGCAACUUGGGCAGAAGUACUGCUCAGUUGUCACUGGACCUCGCCAGAGUAAACUGCCUUUUGAU
 GACCAGGGCGAAUUGAGUGAAAUCGUCAUGGACAGAUACGGGGCAGACAAU
 .(((.....)))).....

AUGUGUGAUGCGACCGGUCUGUCGCAACGUCGUGCCUGCAGGCUCACAGGUU
.....(((((((.....(((.....)))...(((.....))))))))).....

> RF00384_B
GAAUCUUUCACAAAAGAUUUUAUCCGAUAAUUCUCCUCGGACAAUUCGGAUUUGAAUAC
.(((((((.....))))))...(((.....(((.....)))...)))).....

> RF00385_B
GGAAGAUAGGCAUGUAGCUUAGCACCUACAUGUCUAUCGCC
((.....(((((((.....))))))))).....

> RF00386_A
UAAAAACAGCCUGUGGGUUGCACCCACCCACAGGGCCCACUGGGCGCUGGUACACUGGUUAUUAAGG
UACCCUUGUACGCCUGUUUAUUAU
.(((((((.....))))))((.....))((.....(((.....)))...)))).....

> RF00387_A
CGCUCAGGGGAAUCAGGGCAUCGCCUCCUUUCUGGGAGGACACUCCCUUCUGAUGGUGAAUGGG
AACUCCUUCUCCUGCAGCAGCCUGGCCUGCAGCUGUCCUGGUAGAACAGUGUGGACAUUGCAGAAGCUGU
CACUGCCCCAGAAAGAAAGCACCCACAGGCC
.(((.....(((.....)))))).....(((.....(((.....)))...))).....(((.....(((.....)))))).....
.....)))))).....

> RF00388_B
GUUGUAACUUAUGUUGCAACAAACGAGAACC GAUUAUAGUUAUGGUGUCUCAUGAAUUAUUUA
CUGUCCUCUGGGAUUCGCGUCCCAUGGCAGUCGGUUC
(((((((.....)))))).....((((.....(((.....)))))).....((((.....(((.....)))))).....

> RF00389_B
GAAAACUCACCGCAACGAAACGAAACAAGUCGUUCAGAAACACUAGACCACGAGGGCCCCCUAUA
GUCCUGUUGAAGGUGCGGCAGGCCCGUGCGAUAGGCUAACUGUGGUGUUCGCGCACUCCGUCGAGCGGU
UAAUACGACGCUUACCAAGAC
.....((((.....(((.....))))))((.....(((.....(((.....)))...)))).....)))).....

> RF00390_A
UGAGUUCUCGAUCUCUAAAAUCG
.....((((.....))))

> RF00391_A
CAAAAGUCCUGAACUCUCAACGAAUCCGUAAUUAUUAUUCUGCCCAAGCGGGGAAGGAUGAG
AAGCUUCGACCAAGGUUCGACUUGAGCGCCAGCGAGAGAGCGUUGCCGAGGCAACGACCCGAAGGGCGAA
GCGCGCAGCGCUGAGUAAUCCUUCGCGCACCA
.....(((((((.....)))))).....((((.....(((.....)))))).....)))).....

> RF00433_B
AGUCUUGAAAAAAUUCGUACGGUGUGCGUCGUAACAACAAGCAGCGUCUGAAAAGUUUGUGA
AUUUCCAAUUCUAUACAAAGCAAAGUGAAAAUAUCUGUAUUUUUACCUUUAUUCUGUGAAUAGAACGAAA
AACAUACAUAACAAGAU
.((((.....(((.....)))...)))).....(((.....(((.....)))...)))).....

> RF00434_B
ACACCACUAACACAAGCCGAAUCCUGGGAAACAGGCAGAACUAGGUUCGUAAGCUCGGGUAAUUG
GAAGGUGGGCUGUCAACCCACCACCAUCUACCGCCGCAUUGCCUUGUGUUUGGCC
...(((.....(((.....)))...)))).....

> RF00435_A

GCCGCGACAAGCGGUCCGGGCGCCCUAGGGGCGCCGGCGGAGACGGGCGCCGGAGGUGUCCGACGCC
 UGCUCGUACCCAUCUUGCUCUUGGAGGAUUUGGCUAUGAGGA
 ((((((.....))))(((((.....)))))).....(((((((.....))))))(((((.....)))))).....

> RF00436_A
 UCAUUUGUAAGUCGCUUUGGAUAAAAGCGUCUGCUAAAUGACUAAAUGUAAAUGU
 (((((((.....)))))).....

> RF00437_B
 GUAGACAUUUCACAUCAUUCGCCGGGAUUGCGCAAUGUUGCUUUGAAGUGUUGCAAACAUGCGA
 AUCCUAAACUCGGUUCACAACUUCGUUGGCUUAGUUUCCUGGCUUAUAUCCUGGAAACCCGUCGAC
(((((((.....)))))).....

> RF00444_B
 UUGACUUACAAAUGAGAAUCGUUAUGAUUAUCACGUCGCGGACUGGACGGUAAACUGAAA
 GACCUUGGUUCGGACUUUCAGAUUAUCUCCUCAUCAGGCUAAUCACGGUUUUUGACCCGGCUCUUUGCCGG
 GUCUUUUUUUG
(((((((.....)))))).....

> RF00453_A
 UCGGCCACAAACACACAAUCUACUGUUGGUCGA
 (((((((.....)))))).....

> RF00458_A
 CCAACAUGUGAUCUUGCUUGCGGAGGCAAAAUUUGCACAGUAUAAAUCUGCAAGUAGUGCUAU
 UGUUGGAAUCACCGUACCUAUUUAGGUUUACGCUCCAAGAUCGGUGGAUAGCAGCCCUAUCAAUAUCUAGG
 AGAACUGUGCUAUGUUAGAAGAUUAGGUAGUCUCUAAACAGAACAUAUUUACCUGCUGAACAAAUU
 (((((((.....)))))).....

> RF00460_B
 CCACACAGCAUUUUACCCAGAGUCUGUCCCGAGACAUUGCACCUGGGGCGUGUAAAUUGUGAUUAA
 AGUGAGUU
(((((((.....)))))).....

> RF00461_B
 GCUAGCUCGGGCGGGAGGAGCCGCGAGCCGGAGGAGGGGGAGGAGGAAGAAGAGAAGGAAGAGGA
 GAGGGGGCCGCGAGUGGCGACUCGGGCGCUCGGAAGCCGGGCUAUGGACGGGUGAGGCGGGGUGUGCGCAG
 ACAGUGCUCACGCGCGCGCUCUCCCGAGGCCUGGCCCGGGCCUCGGGCGGGGAGGAAGAGUAGCUCGCC
 GAGGCGCCGAGGAGAGCGGGCCGCCACAGCCCGAGCCGGAGAGGGAGCGCGAGCCGCGCCGGCCCGGU
 CGGGCCUCCGAAACCAUGAACUUUCUGCUG
 .(((.....)))).....

> RF00462_A
 AGUCUCCUUUAACUGAAAUUUUCCUACAAACAGAUUUGACCAGAAGGCA
 .((((.....)))).....

> RF00463_A
 AUUCCACCGGGACCGUGCGGGGCGAGUGCCCUUCUCGGUUGCUGCCGCGAGGAGCCCGCCAGC
 CAGCCAGGGCCGCGAGGCCGAGGCCAGGCCGAGCCAGGAGCCCGCCACCGCAGCUGGC
 ..(((.....)))-(((((((.....)))))).....

> RF00465_B
 AUGGAGUCAGGCCAGCAAAGCUGCCACCGGAUACUGGGCAGACGGUGCUGCCUGCGUC
(((((((.....)))))).....

> RF00466_B
CCAGCCGCUGAGGUCAGGGGCUUGGCCUUGUUGGACGCGCCGUCUUCAGCGCUCCGUACCCAUGUU
GCUUCAAGGAGGAUAUGGCUAUGGCAA
(((((((.....))))))..(((.....)))..(((.....))))..(((.....))))..

> RF00467_A
AUUCCCUAACGAUUGCGAACACCUGAAUGAAGCAGAAGGCUUCAUUUGGUGACCCCGACGUGAUAG
UUAGGGAAU
..(((((((.....)))))).....))))..

> RF00480_A
UUUUUUAGGGAAAAUUUGGCCUCCAACAAGGGAAGGCCAGGAAAUUUUCCU
.....(((((((.....)))))).....

> RF00481_B
UUGGUGGCUCCAUCUAGCCCUAGUCACGGCUAGCUGUGAAAGGUCCGUGAGCCGCAUGACUGCAG
AGAGUGCUGAUACUGGCCUCUCUGCAGAUAUGU
..(((((((.....))))))..(((.....)))..(((.....))))..(((.....))))..

> RF00483_B
ACGACUGGGCAUUGCCCCAGUUUCCCCAAAUUGGGCAUUGUCCCGGGUCUCCAACGGACUG
GGCGUUGCCCCGCUACUGGGGACUGCCCUUGGGGUCUCGCUCACCUUCAGCAG
..(((.....)))..(((.....)))..(((.....)))..(((.....))))..

> RF00484_A
AUAUGACUCUCUAGCACCGGGAGGCGAUGAAUUGGGACGCAGGCGCGACCACAGGGACCACUCCCC
CUACACAGACAUGAGACCAUAGGGGAGCUGUCUGGGUGGCCUCAAGGAUAGGCGCUCCCCGAGGUAACCGG
GCUAUCUGGGUU
.....(((.....))))..(((.....))))..(((.....))))..(((.....))))..

> RF00485_A
GUGGGCUAUGGUGACAUGUACCCUGUGACAAUUGGAGGCAAAAUCGUGGGCUCCUUGUGUGCCAU
CGCUGGUGUGCUGACAAUUGCCCUGCCUGUACCUGUCAUCGUGUCCAAC
(((.....))))..(((.....))))..

> RF00487_B
AAAAACGCUUUUACGAGGUAUCAGCACUUUCUUUCAUUGGGGGGAAGGCGUGAGGAAAGUACCA
AACAGCAGCAGACUUUUAACUUAAACAGACAGGUCUGAGAGCCUGAACUCUCAUUUUCCUUGACUUC
AGCCUCCAAGGAGUCCACCAACUUUGGCGCGCCGGCUUCACUUUCAUUAAGUGAAAGAGAGGUGCCAGA
CAUG
.....(((.....)))..(((.....))))..(((.....))))..(((.....))))..

> RF00488_B
AUACUUAACCUAAGAUUAUCAGAGGAGAUACAAGAAGUCCUACUGAUCAAACAUGCGCUUCCAUAUG
UAGAAGGACGUUAAGCAUUUAUCAUUGAACUUAUAUUGUUCUUAUGAAGUCAUUGAUGCAAACUCCUUGGU
CACACACACAUACGGCGCGGAAGGCGUGUUUGCUGACGUUCCAUCUCCUUGUUUCAUCAUUGGUUAUC
CCUUGAUUCCUUGGGGAUUUUUGGGUUAAACUGAUUUUUGGGGCCUUGUUCUUCUGCCUGGAGAAG
UUUGACACCCAAAUUCAAAUUGGUGUUAAGGGAGCUGGGGCCUUCAAAAGAGAGCUUUGUAGAGGCAU
CUUUUUGACUACUUUCUCUAGCGUGCCAUUUUAGUUUUUGACGGAGAUUCGAAUGAACUUAAGUUUAUG
AUGAAGGUAUGGCUGUUGAGAUUAUUUGGUCGGGAUUGUAGUUUGAAGAUGGUCUUCUUUGAGCAGUCUC
AACUUUGCUCGUUCCCGUUAUGGGAAAAUUUUGGAAGGUCUUGGUAGGAACGGGUGGAUCUUAUAUUU
UUGAUUUUUUU
.....(((((((.....))))))..(((.....))))..(((.....))))..(((.....))))..

> RF00489_A
GCUAGAGAGACAAAAGGGCUUCCGCGACGGCGACGUUGAGGGGGCUCUUUUCUUUUGCGGUUUAC
UCUCCCCGUUCCGU
.....(((((((.....)))))).....)

> RF00490_A
AGUCAAGAGUCUUUGGCACCGUUUUUGGUGAGCAGAUUUCUUUUGCCUAUUGAAGGGUGAAAGAC
AUG
.....(((((((.....)))))).....)

> RF00491_B
CAACAAUUGCAUUCUUUAUGUUUCAGGUUCAGGGGGAGGUGUGGGAGGUUUUU
.....(((((((.....)))))).....)

> RF00492_B
CUUUCGGGUGCCCAUUGCAGCCCCGGAUGAGCCCGCAGUAUUUCCUUAUAUGAUCAGGUCCCAU
UGCGGGCGGCGCCGCUUGCCCGGAGCCUGAGAGGAUUAUGAAAACGUGGCGAGCGAAAUGGGGCCAGGGGA
CCUGGAG
(((.....(((((((.....)))))).....)))).....)

> RF00493_A
GGGCAAUGAUGUAAAGGUUUUACGACUGACCUUUGUAACUAUGAAGGUUUCUACACUUGACCUGA
GCUCA
(((.....(((((((.....)))))).....)))).....)

> RF00494_A
GAUCAAUGAUGAAACUAGCCAAAUCUGAGCAUCAGAAGGCUUCCAGUCUACCUGAUGCAUGAUC
UCUACAGUUCUGAGA
..(.....(((.....(((((((.....)))))).....)))).....)

> RF00496_B
CACUCCUGUAUUCUAUGCUUGUGGGCGUAGAUUUUUCAUAG
.....(((((((.....)))))).....)

> RF00497_B
CUGGGAGAGACCAGAGAUCCUGCUGUCUCUACAGCAUCAUCCAGGCACAGAGCGCCAGAAAUGG
AAUGGUGCUGUUGAAUCAACAGGUUCU
(((.....)))(((.....(((((((.....)))))).....)))).....)

> RF00498_A
UGCCAUCGUCGUCGAUCUCUAUCAACUAUCCUUGCGACUAUGGCA
(((((((.....)))))).....)

> RF00499_A
UCCUAGAGAGCUUGGCCGUCGGGCCUUAUACCCGACUUGCUGAGUUUCUCUAGGAGAGUCCCUUC
CCAGCCUGAGGCGGCGUGUCAUAAAAGCCUCAA
(((((((.....(((((((.....)))))).....)))))).....)

> RF00500_B
GGGUAACCACUAAAUCCCGAAAGGGUGGGCUGUGGUGACCUUCC
(((.....(((((((.....)))))).....)))).....)

> RF00502_B
GGGUAACCACUAAAUCCCGAAAGGGUGGGCUGUGGUGACCUCCAAGCCAA
(((.....(((((((.....)))))).....)))).....)

> RF00503_A

UAACUAGAUCACAGAGAUGUGAUGGAAAAUAGUUGAUGAGUUGUUUAAUUUUUAAAGAAUUUUUAUC
UUAAUUAAAGGAAGGAGUGAUUUCAAUGGCACAAGAUUCAUUUCAACAAUCGGUGACUUAGUAAAAUGGA
UUAUCGACACAGUGAACAAAUCACUAAAAAAUAAAGAUGAAUAAUUAAUUACUUUCAUUGUAAAAUUUGUU
AUCUUGGUAAUAGUACUAAAAGUAUGAGUUAAUAAAGCCGUCCCAACUAAUAACCAUGUAAAAUUAGCAAG
UGAGUAACAUUUGCUAGU

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

> RF00505_A

CUUCCGAUGUAGGCCCGUAUUCUUCGCCUGUACCACGGGUCGGUUUUAGUACAGGCGUUUUCUU

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

> RF00506_B

ACAUCCAUGAAACGCAUUAGCACCACCAUUACCACCACCAUCACCACCACCAUCACCAUUACCAUU
ACCACAGGUAACGGUGCGGGCUGACGCGUACAGGAAACACAGAAAAAAGCCCGCACCUGACAGUGCGGGCU
UUUUUUU

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

> RF00507_B

GAGUCCGGGCUCUAGUGCCGCUCGACUAGAGCCCUGUAAUGGUACAGACAUAGAUUACUGUGUCC
GUGCAUUUGACGU

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

Vita

Christopher Roman Cuellar was born on July 21, 1987 in Fresno, California. He moved to El Paso, Texas when he was five years old due to the amount of family present in the area. He graduated from Silva Health Magnet High School in 2005 and started his college career at the University of Texas at El Paso (UTEP) in Fall 2005. During this time he worked as a peer-facilitator for precalculus at UTEP where he tutored students in multiple disciplines of math, not just precalculus. He became a research assistant for the Interactive Systems Group at UTEP during the Spring and Summer of 2008. Christopher obtained his Bachelors of Science in Computer Science Spring of 2009.

After obtaining his Bachelor's degree, Christopher returned to UTEP for Graduate School to pursue his Masters of Science in Computer Science in the Fall of 2009. In the Spring of 2010, he joined the Vision and Learning Lab. During his time in graduate school, Christopher has worked as a Teaching Assistant for Software Engineering for two years, taken two summer internships, and worked at a ecommerce startup.

Permanent address: 9216 Shaver Dr.

El Paso, Texas 79925

This thesis/dissertation was typed by Christopher Roman Cuellar.