Acknowledgments

This thesis was carried out under the direct supervision of **Professor Danny Barash**. The initial idea for the research originated in a question I asked Prof. Barash in a class he taught. Prof. Barash embraced the idea and guided me toward a coherent goal with specific achievable targets along the way. His insightful notes and advice introduced me to numerous solutions to similar problems in our field.

Secondly, **Professor Michal Shapira** and her lab team. From the dedicated guidance in experimental work to helping me understand the mindset necessary for research in the field of biology. As a computer scientist, I learned through their questions and work the biological aspect I should pursue in my own research. I am also thankful for the fact they needed my help in topics unrelated to my thesis. Through the work on RNA-seq and mass spectrometry statistical analysis, I was exposed to a whole world of bioinformatics applications and gathered important experience for my future.

I would also like to thank the Kreitman School of Advanced Graduate Studies for the financial support through the Negev scholarship. Also, the department of Computer Science for an amazing administrative staff and the opportunity to teach.

Finally, my family, Maayan and little Lillymei. Their patience, support and understanding in times of stress and deadlines and their presence was an invaluable contribution to this thesis.

Abstract

Riboswitches are RNA genetic control elements that provide a mechanism for gene regulation. They were initially found in bacteria and observed to control both translation initiation and transcription termination [1; 2]. Riboswitches do not require the intervention of proteins and are thus considered to represent an ancient regulatory system in evolutionary terms. The Rfam database [3-6] classifies 39 Riboswitch families; only one family has been found in eukaryotes. This single known class of eukaryotic riboswitch, namely the TPP riboswitch class, has been found in bacteria, archaea, fungi and plants but not in animals [7]. These few examples of eukaryotic riboswitches were identified using sequence-based bioinformatics search methods such as a combination of BLAST [8] and pattern matching techniques that incorporate base-pairing considerations. None of these approaches perform energy minimization structure predictions. Therefore, there is a clear motivation to develop new bioinformatics methods, in addition to the ongoing advances in covariance models, that will sample the sequence search space more flexibly using structural guidance while retaining the computational efficiency of sequence-based methods.

In this dissertation I present an efficient and flexible pattern-matching tool for RNA based on secondary structure and sequence, as well as a novel pipeline for minimum free energy structure-based discovery of non-coding RNAs (ncRNAs). The pipeline is also extended to generate RNA families around a single design sequence. Whereas pattern matching is still a good solution for practitioners attempting to search for specific ncRNA, its use is still difficult for large genomic datasets.

The pipeline is based on a synthetic biology approach. It uses an inverse RNA solver we developed, RNAfbinv 2.0, to generate new sequences based on target structure and additional constraints. Those sequences are synthetic candidates of the target ncRNA in question. They can be used for sequence-based search methods such as BLAST and statistical models on large genomic databases. The matched sequences are then filtered for loss of information and nearby annotations that might show biological significance. Since single sequence matches can be found randomly in any large genomic database, this filtration process is sometimes insufficient. Therefore, we developed a process in which

we create new RNA families around the designed sequence and initial matches. This allows us to expand a small group of matches and reduce sequence constriction while preserving general structural information.

I selected the purine riboswitch as a target for our study; this family has never been found in eukaryotes. The structures of known prokaryote sequences belonging to this family are predicted accurately by free energy minimization methods. We were able to detect most of the known prokaryote purine riboswitches as well as multiple eukaryote candidates using our initial pipeline. When extending specific candidates to families we were able to reconstruct the prokaryote purine riboswitch family from minimal sequence information as well as building novel eukaryote candidate families. The eukaryote candidates were then filtered based on covariation of structure and phylogeny. These candidates, if tested experimentally, may be the first purine riboswitches in eukaryotes. I believe that the pipeline we suggested can be used to search for the occurrences of other new, more complex, structurally dependent ncRNAs.

Table of Contents

Acknowledgments 1

Abstract 2

List of Figures 7

List of Tables 12

1. Introduction 13

1.1. Definitions for Computational RNA Biology 18

1.1.1. Sequence 18

1.1.2. Secondary Structure 18

1.1.3. Minimum Free Energy Secondary Structure 20

1.2. Traditional search methods for Riboswitches 21

1.2.1. Pattern-Based Search Methods 21

1.2.2. Profile Hidden Markov Models 22

1.2.3. Covariance Models 24

1.3. The Purine Riboswitch 26

2. RNAPattMatch – Simple and Efficient Pattern-Based Search 29

2.1. Preface 29

2.2. Introduction 29

2.3. The Affix array data structure 31

2.3.1. Suffix Arrays 31

2.3.2. Affix Arrays 32

2.4. Implementation 35

2.4.1. Input, Output and Data Structures 35

2.4.2. Motif Breakdown and Merge 36

2.4.3. Search Tasks 37

2.4.4. Web server 38
2.5. Conclusions 41
3. incaRNAfbinv 2.0 – Fragment Based Design with Motif Specific Control 43
3.1. Preface 43
3.2. Introduction 43
3.3. Implementation 45
3.3.1. incaRNAtion 45
3.3.2. RNAfbinv 2.0 46
3.3.3. Web Server 49
3.4. Conclusions 51
4. A New Energy Minimization Structure-Based Search Method for Riboswitches 52
4.1. Preface 52
4.2. Introduction 52
4.3. Design to Search Pipeline 53
4.3.1. Seed Generation 54
4.3.2. Candidate Design 56
4.3.3. Database Search 58
4.3.4. Result Filtering 59
4.4. Results and Discussion 60
4.4.1. In-line probing 60
4.4.2. Prokaryote Riboswitches 64
4.4.3. Eukaryote Riboswitches 67
4.5. Conclusions 70
5. Eukaryote Riboswitch Candidates Supported by Covariance Models 72
5.1. Preface 72

5.2. Introduction 72
5.3. Iterative Covariance Model Expansion 74
5.4. Results and Discussion 76
5.4.1. The Rfam Purine Riboswitch Model 76
5.4.2. Reconstructing the Rfam purine riboswitch family 77
5.4.3. Eukaryote transketolase candidate 79
5.4.4. Eukaryote myosin IX candidate 81
5.4.5. Unannotated mammalian candidate 83
5.5. Conclusions 85
6. Thesis Conclusions 87
7. References 91
תקציר 106

List of Figures

Figure 1-1: Different representations of an RNA sequence. Stem motifs are marked in blue, Multi-loop in green, internal loop in yellow, Hairpin-loop in red and Bulge in light blue. A. The graphical representation of the RNA sequence and structure. B. Text representation of an RNA sequence, secondary structure and Shapiro motif-based representation. 20

Figure 1-2: Profile HMM Construction and Search. Taken from the HMMER manual [97]. A. An example of multiple sequence alignment. B. The states generated by column 2. We can see that position 2 has four U's four C's meaning that the emission probability is 0.5 for each. C. The final pHMM computed for sequence alignment. D. An example of a path generated by matching the sequence GUGAUUCHUGC..... 24

Figure 1-3: Covariance model based on consensus structure. Taken from Infernal [52]. A. The consensus structure matching the MSA. B. The CM guide tree containing states based on the structure. Note that indexes 4-14 match the left subgraph whereas 15-27 match the right. C. Zoom of the MATX states 6 to 8. Since 12 is a bulge to the right we see a MATR state. All states include insert and delete substates to allow for flexibility. A single side can be deleted in the MATP nodes by selecting the match left or match right node. 26

Figure 1-4: Purine riboswitch model. Taken from Rfam family RF00167, Purine Riboswitch. A. The consensus sequence and structure for purine riboswitch. The high sequence conservation is visible. B. Molecular model of the interaction between ligand and surrounding aptamer. Position 74 affects the binding molecule. 28

Figure 2-1: Affix array data structures. Taken from Structator [63]. Matching the sequence "AUAGCUGCUGCUGCA". This includes suffix array, reverse prefix array and matching the longest common prefix and affix links arrays. Aflk is the affix links array. We can see that the index 4 in the suffix array matches index 4 in the reverse prefix array because it ends with CUGC which is common for this interval. 33

Figure 2-2: Unidirectional vs Bidirectional search. Taken from Structator [63]. In a unidirectional search the 'N' IUPAC code can fit any of the four nucleic acids increasing the number of intervals exponentially. With a bidirectional search we start from the

hairpin and when arriving at the stem, the comparison alternates across it. The selection in level 5 reduces the selection for level 6 for legal base pairing. 34
Figure 2-3: RNAPattMatch input query example. Above an illustration of the input. '.' Marks gaps results that can be used up to all given points. Below the query sequence and structure input. 36
Figure 2-4: RNAPattMatch input screen. Example inputs for a guanine-binding riboswitch aptamer on the full genome of *T. tengcongensis* with default base pairing rules. 39
Figure 2-5: RNAPattMatch output screen. Results for the Guanine-binding riboswitch aptamer on the full genome of *T. tengcongensis* with default base pairing rules. Note that many results will appear on the same index but with different structures based on the gaps used. 40
Figure 2-6: RNAPattMatch comparison screen. Results for the Guanine-binding riboswitch aptamer. Folding the sequence using the Vienna RNA package induces a structure with more base pairs as many of the base pairs were matched to gaps that are considered unbound by default. 41
Figure 3-1: Tree construction. Starting from an RNA sequence, secondary structure is calculated using RNAfold [67]. Shapiro structure is then generated for the structure. In the last phase a tree is constructed as a combination of all the prior data. Unlike design sequence, target secondary structure is taken from input. The only motifs that might have more than a single child are Multi-loops and External regions. 47
Figure 3-2: Illustration of the differences between incaRNAfbinv 1.0 and incaRNAfbinv 2.0 for the FMN riboswitch aptamer. A. Target sequence and structure. Critical nucleic acid bases indicated in black. B. Typical design output of incaRNAfbinv 1.0. Sequence constraints were satisfied but not always properly in the correct structural context (red color). C. Typical design output of incaRNAfbinv 2.0. Sequence constraints are not only satisfied, but also shifted to match their original structural context. 49
Figure 3-3: IncaRNAfbinv 2.0 web server design screen. Input corresponds to a guanine-binding riboswitch (available as an example). The image on the right is generated using VARNA [126]. The bases marked in green belong to a motif selected from the list for preservation. The image will update automatically upon selection and when a balanced structure is inserted. 50

Figure 3-4: IncaRNAfbinv 2.0 web server result screen. Output corresponds to a guanine-binding riboswitch. We can see the sequences are matched to the exact sequence constraints and structure. Longer targets may result in higher scores. Design scores above 100 imply change in motif while scores above 1000 imply missing sequence constraint. 50
Figure 4-1: Design to Search Pipeline overview. Note that in some phases multiple products are generated and passed onto the next phase. 54
Figure 4-2: Schematic illustration of a potential well with a minimum in the known aptamer sequence. It is possible to escape the minimum by performing nucleotide mutations to the initial sequence until there are no BLAST hits when inserting the mutated sequence as input. If the mutations disrupt the known aptamer structure, the use of RNAfbinv will restore the known structure while generating designed sequences as output. Subsequently, designed sequences in the borderline of the potential well that do show BLAST hits should then be carefully examined in their hits. It is expected that most of the hits observed will be from known bacteria, but a few unknown bacteria and exceptional eukaryotic organisms will also show up. 55
Figure 4-3: Variety and design scores for different seed types. Based on generation and design of purine riboswitch aptamer (n=400). On the left we can see that random and incaRNAtion seeds were more diverse compared with mutated *xpt* seeds. The effect continued to the design results that were based on those seeds. On the right, mean design scores. The incaRNAtion seed were slightly better. For more complex design problems the advantage of incaRNAtion increases. 56
Figure 4-4: RNAfbinv 2.0 target for Purine Riboswitch aptamer design. Generated using VARNA[126]. Note that only four nucleic acids were targeted based on direct binding interaction. 58
Figure 4-5: Comparison of in-line probing assay results between Breaker's lab taken from [143] on the left and my gel on the right. Although the resolution obtained in my experiments does not match that of the Breaker lab, we clearly observe binding reactivity for 1 μM of Guanine and Hypoxthanthine. NR – no reaction; T1 – G ladder; -OH – single base ladder; ─ – in-line reaction without ligand; G – Guanine; H – Hypoxthantine; X- Xthanthine; A – Adenine. 64

Figure 4-6: Tracing the search process from mutated *xpt* seed through design to matched BLAST results. Tracing highlighted run to find multiple known aptamers. Top pane shows the output screen for the multiple random mutation phase resulting in a seed sequence. Middle pane shows the output of multiple RNAfbinv runs. Bottom pane shows the output of nucleotide BLAST for designed sequence from run 7 (highlighted). Top BLAST matches include the *xpt* aptamer from *B. subtilis* yet further down we note multiple known riboswitches from different genomes (highlighted)..... 65

Figure 4-7: *Pelosinus* sp. UFO1 riboswitch candidate in-line probing results. The assay was performed with guanine and 2'-deoxyguanosine using a concentration of 1 nM and 1 μM of the ligand. Although the resolution is low, we do observe band reduction around the multi-loop and the two hairpin loops only in the 1 μM guanine lane. 67

Figure 4-8: Select eukaryote candidates. A. *Conticribra weissflogii* (microalgae) and B. *Cymbopleura sp. TN-2014* (microalgae), C. *Yarrowia lipolytica CLIB122* (fungus), D. *Callorhinchus milii* (Australian ghostshark), E. *Latimeria chalumnae* (coelacanth), F. *Haplochromis burtoni* (African cichlid fish). Putative ligand-binding nts are marked in green. 69

Figure 4-9: *A. oryzae* candidate predicted minimum energy structure compared with the *M. florum* 2'-deoxyguanosine riboswitch [75]. The four nucleic acids that directly bind to guanine appear in the correct structural context (marked by roman numerals). However, the predicted minimum energy structure did not match the structure that appeared in my in-line probing assay. 70

Figure 5-1: Conceptualization of the sequence expansion method. Our proposed method samples the sequence space by matching sequences that share the general shape of the purine riboswitch. The dotted circle with A represents sequences matched in bacteria identified by the Rfam model. The circles with B show single sequence matches using our energy minimization method. The dotted circles with C are single sequences expanded to contain multiple similar sequences with structure-preserving covariation. 74

Figure 5-2: Overview of model expansion pipeline. Generating new covariance models based on search results as long as new sequences are matched. The RNAfbinv 2.0 alignment function allows us to expand to include results with higher sequence

variability if they maintain key sequence-structure features. Once the model is more relaxed, new results can be found. 75
Figure 5-3: Consensus sequence-structure of the purine riboswitch family from Rfam [3-6]. Left, the consensus using seeds sequences (133) chosen by Rfam; Right the consensus using the full list of sequences (2,702). The model generated from the full alignment shows only a slight relaxation compared with the model generated from the seed. 77
Figure 5-4: Comparison between Rfam purine riboswitch family and our generated family. The design target shows the minimal sequence constraints used in the design process (red circles). The designed sequence includes the relevant nucleic acids within their structural context. Comparing the two families shows high conservation in the stems as we would expect. 78
Figure 5-5: Transketolase candidate model compared with Rfam model. Structural preservation of sequences can be found mostly in the left stem, bottom stem and left side of the multi-loop. Compatible mutations were found but not covarying mutations. 79
Figure 5-6: Phylogenetic distribution of the transketolase candidate family. Green, fungi from the Ascomycota phylum; Yellow, other fungi; Red, not fungi. 81
Figure 5-7: Phylogenetic distribution of the myosin candidate family. Green, mammals; Brown, reptiles; Blue, birds. 82
Figure 5-8: Myosin candidate model compared with Rfam model. Heavy structural preservation of the bottom and right stems include compatible G-C/U mutations and a single covarying mutation. 83
Figure 5-9: Phylogenetic distribution of the mammalian candidate family (green). Red, a single species of nematode. 84
Figure 5-10: Mammalian candidate model compared with Rfam model. A G-U rich sequence with C present in the Watson-Crick binding position. The high covariation in the stems was rare in the reviewed results. 85

List of Tables

Table 2-1: Run time for RNAPattMatch. Queries are available on the web server as examples. Only the target *T. tengcongensis* MB4 is available as an example. A. Matches for a guanine riboswitch were not found in the eukaryote organisms reported in the table. B. Running times were taken from the RNAPattMatch web server for non-cached targets and do not include file upload time. C. Difficulty is indicated by the specificity of the query and the amount of hairpin loops to merge. In this case, the G-C rich hairpin has no specific nucleic acid in the unbound section. 38

Table 4-1: Primers used for aptamer transcription. Red, short tail for T7 RNA polymerase; Blue, T7 RNA polymerase promotor; Green, sequence matched by the search method as the aptamer candidate. The sections marked in bold text mark the overlapping region between forward and reverse annealing primers. 62

Table 4-2: List of buffers and reaction solutions used in the assay. reagents marked with a star were added just before loading the gel. pH values were measured in room temperature (~23°C). 63

1. Introduction

Genetic control of fundamental processes such as transcription, splicing and translation is a complex process in many cases mediated by proteins that monitor the environment and bind selectively to targets. Surprisingly, cis-acting RNA genetic control elements have been discovered that are capable of directly sensing small ligands thereby playing a regulatory role by switching their conformational states without the participation of proteins. These RNA elements are called riboswitches and they are encoded as part of the gene they regulate [9]. Riboswitches can be conceptually divided into two distinct parts: an aptamer and an expression platform. The aptamer contains a binding pocket that directly interacts with a small molecule. This binding forces the expression platform to undergo structural changes that in turn affect gene expression. In most riboswitch families the aptamer and expression platform partially overlap but there are exceptions [7]. The SAM-III riboswitch in bacteria includes the Shine-Dalgarno sequence it regulates within the aptamer itself [10]. A single riboswitch may also consist of multiple aptamers thus constructing a Boolean gate [11].

The aptamers were shown to bind many different ligand classes from small Magnesium [12] ions to larger enzyme cofactors such as Adenosylcobalamin [13]. Ligands can also be found in the form of nucleotides derivatives, amino acids and even sugar in the form of glucosamine-6-phosphate [14]. The genes regulated by the riboswitch typically encode proteins related to *de novo* synthesis, salvage or transport of the bound ligand or a related derivative [7]. Riboswitches engage with a variety of different gene regulation mechanisms. In bacteria, riboswitches were shown to both disable and enable translation by controlling the formation of a terminator loop in the mRNA. They were also shown to both inhibit and activate translation by enforcing access to the Shine-Dalgarno sequence, thus controlling ribosomal binding. RNA degradation is also affected by riboswitches. In gram-positive bacteria, the glamS (encoding glucosamine-6-phosphate synthase) riboswitch induces cleavage of the 5'UTR exposing the 5'-OH that accelerates degradation by RNase J. In eukaryotes, riboswitches were shown to control alternative splicing. The expression platform structural conformation controls access to different sets of splicing sites. Some induce premature termination by introducing a stop codon [7].

Although the first experimental validations were published in 2002 [1; 2; 15; 16], conserved sequence patterns in the 5' UTRs of bacteria were identified several years earlier using comparative analysis of the upstream regions of several genes expected to be co-regulated. These studies contributed to the description of the *RFN* element [17], the S-box [18] and the THI-box [19].

One of the first riboswitches to be discovered, the 'TPP-riboswitch', provides an RNA control mechanism for both transcription termination and translation initiation during thiamine biosynthesis in bacteria. It is an RNA element that responds to concentration changes of thiamine pyrophosphate (TPP) with a conformational rearrangement that affects transcription termination in *Bacillus subtilis* and translation initiation in *Escherichia coli*. Other discovered riboswitches respond to molecules that change conformation upon binding like flavin mononucleotide (FMN), S-adenosylmethionine (SAM) [20; 21], coenzyme B12, lysine, guanine, adenine, and later some more peculiar riboswitches including *glmS* [14], glutamine-, glycine-, and the cyclic di-GMP riboswitches [7; 22]. The structural basis and biochemical properties of several of these riboswitches have been elucidated at high resolution (e.g., [23-25] for the glycine riboswitch, [26; 27] for the *glmS* riboswitch, and [28-32] for the cyclic di-GMP riboswitch). All the above riboswitches were identified solely in prokaryotes with the exception of the 'TPP-riboswitch' that is the only riboswitch class discovered in fungi and plants. Nevertheless, they are so widespread within the different phyla of prokaryotes that they are considered an ancient mechanism, remnants of an RNA world. However, to date riboswitches have not been discovered in animals. The identification of a wider spectrum of riboswitches with a more significant representation among eukaryotes relative to what is known at present [22; 33] remains a challenging task.

To achieve this goal, improved computational search methods for riboswitch discovery must be developed. From the bioinformatics standpoint, a substantial amount of information can be inferred about the riboswitch mechanism by examining its structure in addition to its sequence. The conserved sequence and structure of the aptamer domain can identify riboswitches in analogy to a fingerprint, a fact that can be utilized by structure-based bioinformatics search methods that also include sequence considerations.

Noteworthy bioinformatics riboswitch search methods include an early and simple search program, SequenceSniffer [34] used to identify new riboswitches in bacteria as well as most of the multiple examples of eukaryotic riboswitches known to date. Subsequent work on a genomic scale by Barrick et al. [35; 36] triggered identification of multiple additional bacterial riboswitches. Weinberg and Ruzzo helped discover new classes of riboswitches through a covariance models (CM) approach implemented in CMfinder [37; 38], and applied biologically in [39-43]. The insertion of known riboswitches into the Rfam database [3-6] was also instrumental in advancing the field. Gelfand and coworkers [44] have continued advancing the comparative analysis approach in microbes, with more recent findings using comparative genomics in metagenomes [40; 43; 45]. Other simpler sequence-based methods have also been developed [46; 47], the more sophisticated of them employs sequence-based filters for detecting new riboswitches [48]. Another approach is that of RSEARCH [49], which by stochastic context-free grammar, can also be used to search for new riboswitches. Hidden Markov Models (HMMs) are frequently used to search for new riboswitches such as in pHMM [50] and the advanced Infernal, which also incorporates covariance models [51-53]. Other approaches include Boltzmann probability of RNA structural neighbors for riboswitch detection as in RNAbor [54]. Genome-wide measurement of RNA secondary structure by high-throughput sequencing [55] can also be useful for riboswitch discovery. The topic of searching for new riboswitches in genomes is reviewed in [56]. New ways to detect riboswitches based on their 3D structural modules are also being developed [57].

Here, we present two tools that enhance existing methods for riboswitch detection. In addition we present A New Energy Minimization Structure-Based Search Method for Riboswitches supported by Covariance modeling of similar candidates.

The first tool RNAPattMatch is a software and web service for flexible RNA sequence-structure pattern matching and the second is an RNA design tool called IncaRNAfbinv 2.0 we improved for the design of functional RNAs. The detection method uses a new approach based on synthetic biology wherein we design riboswitch candidate sequences and then use the efficiency of sequence-based search to scan large genomic

databases. The method is also extended to support single results with a structurally modeled family.

Several programs that implement the RNA pattern-matching approach have been developed over the years; these are not available as web servers. They range in the sophistication of their content from simple and specific programs, like SequenceSniffer [34; 58] and RNA-PATTERN [59] for identifying riboswitches, to more complicated and general programs like RNAmot [60] from the Eddy/Rivas lab, PatSearch [61], RNAmotif [62] and Structator [63] These general-purpose programs are rather sophisticated for the practitioner. The RNAPattMatch web server I present here aims to provide a user-friendly server that maintains the most important ingredients of a general-purpose program and yet is simple, practical and efficient for users of different backgrounds. From the methodology standpoint, our approach is closest to that of RNA Structator. In analogy with RNA Structator [63], we utilize the highly efficient index data structure, called affix arrays, that is suitable for sequence-structure patterns. The affix array data structure is equivalent to the affix tree with respect to its algorithmic functionality for pattern matching but with smaller memory requirements and improved performance [64]. However, we extend the capabilities of Structator by allowing more flexible variable gaps (providing an upper boundary to the gap length permitted at any position in the sequence) as part of the pattern definition, as is also becoming available in *de novo* motif search tools beyond RNAs [65], and by analyzing the results using energy-minimization methods. In addition we provide a user-friendly web server.

The second tool we present is an improved version of RNAfbinv [66]. IncaRNAfbinv 2.0 is a coarse-grained inverse RNA solver. The inverse RNA folding problem (or RNA design) is designed to generate RNA sequences that fit a given RNA structure. The first program to tackle the problem is called RNAinverse, and was put forth as part of the Vienna RNA package [67]. It receives a dot bracket RNA secondary structure and sequence constraints as input and outputs sequences that fit that structure. Unlike RNAinverse, RNAfbinv attempts to match the general shape of the dot bracket secondary structure thus generating flexible structures. The new version was optimized along with our novel search method. A key feature of a riboswitch is the ligand binding interactions:

with IncaRNAfbinv 2.0 sequence constraints are bound to the motif to which they are aligned and not to a static index as in the previous version. This allows us to keep the structural flexibility while maintaining control on key nucleotide locations. RNAfbinv 2.0 was also integrated with incaRNAtion [68] which increases the variability of the designed sequences. incaRNAtion generates initial seed sequences that are sampled globally from a pseudo Boltzmann distribution and that have high affinity to the target structure thus generating varied and thermodynamically stable sequences.

Subsequently I present a novel design method that incorporates energy minimization techniques for riboswitch searching while also considering conservation in sequence. Our approach, developed independently, follows the same philosophy as that of [69] who presented a method to identify IRES-like structural domains but did not address riboswitches. For riboswitches, we use folding prediction algorithms such as Mfold/UNAFold [70-72] or the Vienna RNA package [67; 73; 74] with the most updated energy rules [75]. Reference [76] provides a preliminary focused review about using folding prediction methods in the context of riboswitches. Energy minimization was used to identify a potential purine riboswitch in *Arabidopsis thaliana* that exhibited some basic properties shown in [77-79] and was tested biochemically using in-line probing [80]. A putative SAM riboswitch in vertebrates was reported in [81] based on comparative analysis and was tested experimentally.

From the computational perspective, the main drawbacks of existing structure-based methods are the repeated use of folding predictions within a moving window with a fixed window size throughout a sizeable data set. This approach leads to high complexity, extended computational time and gives limited accuracy of the folding predictions performed on the data set (only the accuracy of the folding prediction performed on the query can be checked in advance). To resolve these drawbacks, we used our improved RNA fragment-based design tool, IncaRNAfbinv 2.0, to design sequences that contain sequence-structure similarity and may be riboswitch candidates. These candidates are searched on large genomic databases using efficient sequence-based search methods and filtered. To improve the filtering process, we improved the method further by generating

and expanding RNA families for results gathered from the same designed sequence and modeling them using covariance models.

1.1. Definitions for Computational RNA Biology

In this section we define basic aspects of computational RNA biology that are used throughout this text. We will focus on the basic representation of RNA sequence, secondary structure and minimum energy folding.

1.1.1. Sequence

An RNA sequence is represented as a word over an alphabet: $w \in \{A, C, G, U\}^+$. Each of the four letters indicates a specific nucleic acid: Adenine, Cytosine, Guanine and Uracil, respectively. The order of the letters is equivalent to the connection order of the bases from the 5' carbon atom of the ribose to the 3' carbon atom of the next ribose nucleic acid via a phosphate group. To describe a pattern matching multiple RNA sequences we use the IUPAC code alphabet [82]. The IUPAC alphabet contains codes that match one or more nucleic acid such that each combination of nucleic acids can be represented. For examples the *'N'* code represents either A, C, G or U. Thus, an RNA pattern is represented as a word over the IUPAC code alphabet, $w \in \{A, C, G, U, R, Y, S, W, K, M, B, D, H, V, N, ., -\}^+$, with similar ordering rules as for the RNA sequence representation. The '.' and '-' symbols define a gap of unspecified length. Here we will refer to the latter as an RNA sequence even though it represents a pattern of multiple sequences.

1.1.2. Secondary Structure

RNA nucleic acids can interact within one another. The most common interaction, referred to as canonical or Watson-Crick base pairing, is between Adenine and Uracil or Guanine and Cytosine. This interaction is based on intermolecular hydrogen bonds. Another common interaction occurs between Guanine and Uracil, this interaction is called wobble base pairing. Whereas these three interactions are most common, other interactions are also possible. Here I will focus on canonical and wobbling base pairs alone since they determine the global shape of a sequence [83]. RNAs usually appear as a single strand and contain a flexible backbone which allows the strand to fold upon itself. The base pairing interactions support the fold and force it into a somewhat stable structure.

The goal of an RNA secondary structure is to define a two-dimensional structural representation of the RNA sequence which indicates its general shape. The secondary structure of a given RNA sequence is defined as a set of pairs $S \in \{(i_1, j_1), \dots, (i_m, j_m)\}$. A pair $(i, j) \in S$ specifies that the nucleic acid in position *i* is connected to the nucleic acid in position *j* via base pairing such that: $i < j, \nexists (i, j') \in S \; or \; (i', j) \in S$ for $i \neq i' and \; j \neq j'$. This means that each base is paired only once. Moreover, $if \; (i_l, j_l), (i_k, j_k) \in S \; s.t. \; i_l < i_k$ means that either $i_l < j_l < i_k < j_k \; or \; i_l < j_l < j_k < j_l$, thus the base pairs to not cross each other. Such crossings are referred to as a pseudoknot and while they might have a significance in biological function, they are rare [84] and introduce excessive computational complexity.

RNA secondary structure can sometimes be too specific since it represents the shape at single base pair resolution. However, it can be generalized as a list of motifs (Figure 1-1). A stem is a bounded motif which includes two strands of consecutive nucleic acids that are paired to each other. The unbound motifs include a multi-loop, hairpin-loop, interior-loop and bulge. These motifs are comprised of unpaired nucleic acids. We use a motif-based representation of RNA called a Shapiro representation [85]. The Shapiro representation is defined as a list of motifs marked in a tree-like form. Each motif is surrounded by round brackets and contains a letter representing the motif, the number of nucleic acids in the motif and a list of child motifs.

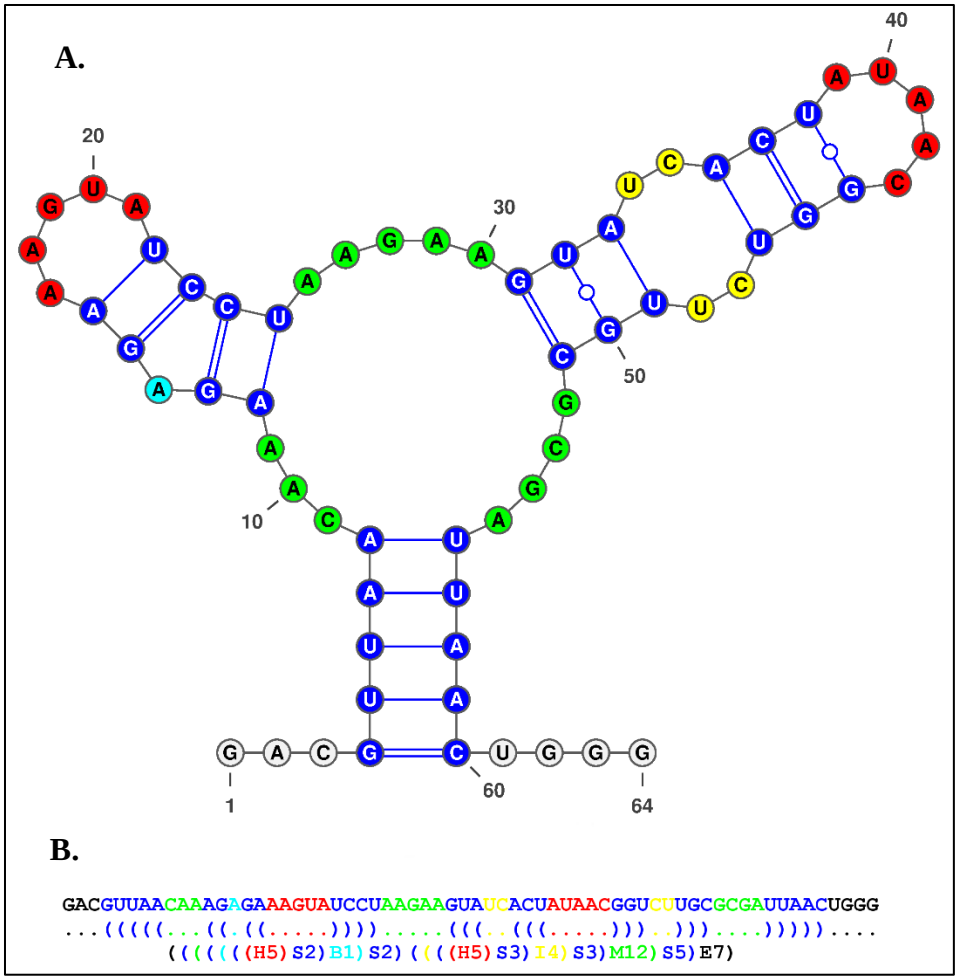

Figure 1-1: Different representations of an RNA sequence. Stem motifs are marked in blue, Multi-loop in green, internal loop in yellow, Hairpin-loop in red and Bulge in light blue. **A.** The graphical representation of the RNA sequence and structure. **B.** Text representation of an RNA sequence, secondary structure and Shapiro motif-based representation.

1.1.3. **Minimum Free Energy Secondary Structure**

The RNA energy model is defined by thermodynamic rules based on the chemical interaction of RNA molecules. RNA strands tends to lower the entropy and to fold into a minimal free energy state [86]. RNA structure prediction was initially solved by maximizing base pairing using the Nussinov algorithm [87]. The issue is that not all base pairs are equivalent in free energy gain from the interaction. Thus a new algorithm was developed by M. Zuker in 1981[70]. It generates the minimum energy structure by using free energy values experimentally measured from small RNA sequences [88].

Subsequently Turner & Matthews published a more comprehensive database of energy rules called Nearest Neighbor Parameters Database [89] which was updated in 2004.

It is important to note that whereas the minimum free energy structure is the optimal composition, it does not always occur in nature. Sub-optimal structures may contain energy barriers that are stable enough to stop further changes. Looking at all sub-optimal structures is not a viable approach as even the number of structures to which no base pairs can be added, and therefore have lowest energy, grows exponentially [90]. Furthermore, chaperones and other RNA binding proteins may intervene. Therefore in this text we will assume no external intervention.

1.2. Traditional search methods for Riboswitches

The traditional methods that are mostly sequence-based are still useful when analyzing a newly sequenced genome (e.g., as in [91]). These include methods like SequenceSniffer that was used earlier for the detection of additional prokaryotic riboswitches [35; 58] and for the initial detection of eukaryotic riboswitches [34], as well as the state-of-the-art Infernal [52] that supports riboswitch searches using Covariance Models and Hidden Markov Models [53]. Those methods will be discussed below.

Development of efficient new structure-based methods is important for further advancement and for *de novo* discovery of structured ncRNA in general [92]. Here, we focus on RNA secondary structure when referring to structure-based methods, but this can also be expanded to RNA tertiary structures as in RMDetect, developed by Cruz and Westhof [57].

1.2.1. Pattern-Based Search Methods

Initial work on riboswitch identification by Barrick and Breaker led to an internal program called SequenceSniffer [34] (unpublished algorithm). This program used a simple input that included both sequence and structure constraints as well as finite gaps. For example in [34] a pattern “CTGAGA[200]ACYTGA[5] <<<GNTNNNNC>>>[5]CGNRGGRA” was used to find multiple instances of potential TPP riboswitches in bacteria and eukaryotes. For this simple approach the user, aka the biologist, must iteratively refine the search pattern. Therefore, it is crucial to create a simple pattern language which maintains the

ability for flexible constraint. Note that the pattern above extends the IUPAC code alphabet to allow for finite gaps and structural information.

Many pattern matching approaches have been developed and used in the search for riboswitches. RNAbob [93], developed by Sean Eddy, uses a nondeterministic finite state machine with node rewriting rules. Other approaches include the use of filters to reduce the amount of search space. The Denison riboswitch detector [94] uses the heaviest path problem solver. The nodes are created by an initial local alignment filter which matches small fragments of the entire pattern. This tool is quite fast; however, the complexity of the pattern language might be an issue for some users.

Another major approach involves indexing large DNA databases into Affix arrays. Structator [63] is a pattern-matching tool based on Affix arrays. Once the affix arrays are created, the search time depends solely on the length of the pattern thus allowing for fast multiple searches. Affix arrays will be discussed in the Methods section as it was the approach chosen for our tool – RNAPattMatch.

1.2.2. **Profile Hidden Markov Models**

Profile analysis has long been a useful tool for identification of new sequences that fit a sequence domain. The purpose of a profile is to describe the consensus of a multiple sequence alignment. Generating manual profiles, such as a sequence pattern, can be a difficutl task requiring many biological assumptions. A profile HMM (pHMM) is the process of building such a model from an alignment of multiple sequences by turning it into a position-specific scoring system [95]. Profile-hmm are currently used in both the Rfam [3-6] and Pfam [96] databases to describe families of ncRNA and proteins, respectively. The HAMMER tool suite [97] is the most common tool for pHMM in bioinformatics use.

We define a model as $M = (S, TP, EP)$ where:

- $S = \{Begin, End, Insert_i, Delete_i, Match_i\}$ is a group of states such that i is an index from the sequence alignment.
- $TP: S \times S \in [0,1]$ the Transition Probability from two given states.

- $EP: S \in [0,1]$ The Emission Probability for a state S is the distribution of values in the state. $EP(s) = 1\ for\ s \neq Match_i$, otherwise $EP(s)$ is the distribution of each nucleic acid in the alignment.
- Note that in a HMM future states do not depend on earlier events thus: $TP(s_i, s_j) = 0\ \forall s_i, s_j \in S\ s.t. j < i$ and $TP(s, Begin) = 0\ \forall s \in S$

Once the sequences are aligned and redundant sequences are removed, a model can be built. For each column in the alignment, three states are generated: Match state, Deletion State and an Insertion state. Between states for two consecutive columns, a transfer probability is generated. For match and deletion states, the transition probability is based on the number of rows where a nucleic acid is not aligned. A deletion state might still have some minimal transition probability. For insertion states, the number is decided such that a larger probability means a more flexible model. A transition probability is also defined from each insertion state to allow for multiple consecutive insertions. As for emission probabilities, these are only relevant to Match states. Each nucleic acid (A, C, G and U) has a specific probability based on the percentage of sequences that has that nucleic acid in the given index (See figure 1-2).

The model is then used to score given sequences. The algorithm finds the path $\{(Begin, S_1), (S_1, S_2), \dots, (S_n, End)\}$ such that the following value is maximal:

$$\prod_{s \in path} EP(S) \cdot \prod_{(S_i, S_j) \epsilon path} TP(S_i, S_j)$$

For simplicity, a sum of log values is calculated. If the maximal value for the given sequence is higher than that of a precalculated threshold, that sequence is considered as fitting the model. The threshold is usually calculated by trying to find the model on a randomly generated sequence database. As we see, the pHMM does not consider any secondary structure information.

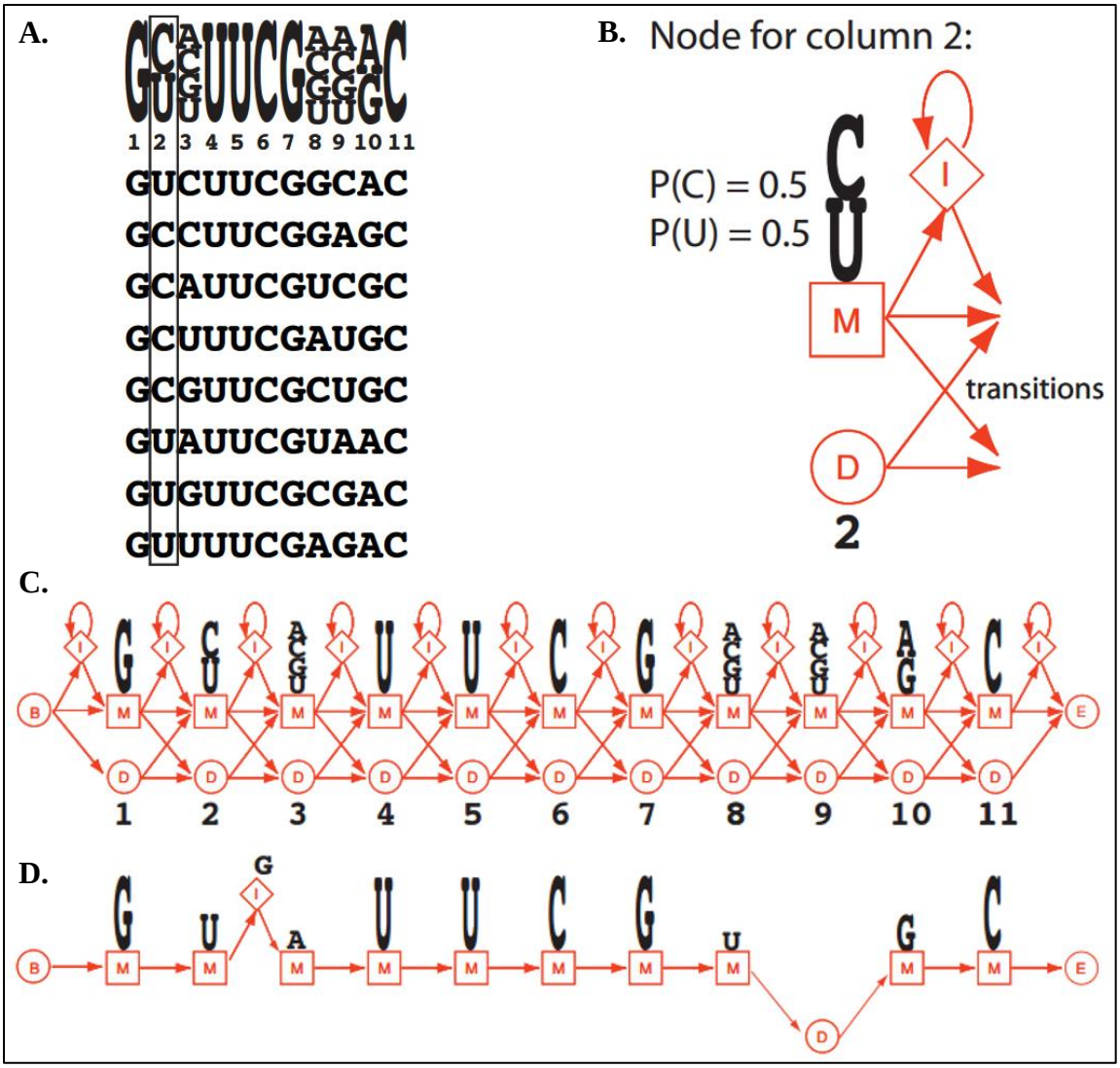

Figure 1-2: Profile HMM Construction and Search. Taken from the HMMER manual [97]. **A.** An example of multiple sequence alignment. **B.** The states generated by column 2. We can see that position 2 has four U's four C's meaning that the emission probability is 0.5 for each. **C.** The final pHMM computed for sequence alignment. **D.** An example of a path generated by matching the sequence GUGAUUCHUGC

1.2.3. **Covariance Models**

Covariance models (CM) can be described as an extension of pHMM that combines sequence and structure information. Like pHMM, CMs require a multiple sequence alignment (MSA) for the construction process. In addition, it requires a single consensus structure that matches the MSA. CMs are widely used for RNA families in Rfam [3-6].

The Infernal tool suite [51-53] can be used to generate and search via CMs. Newly curated genomes are tested using Infernal against all the RNA families in Rfam regularly making it the most used tool for ncRNA profiling.

The CM is defined with States, Transition Probabilities and Emission Probabilities similar to pHMM. There are five types of states, each can contain multiple substates within the graph. The graph generated from these state is called the guide tree. Root is the state equivalent to Begin in pHMM. Unlike pHMM, the CM graph has multiple End nodes. Each time the structure splits to a stem-loop motif, a special node call BIF (Bifurcation) is added to the graph. From the BIF node two subgraphs are generated, one describing the stem-loop motif and the other continuing to describe the rest of the sequence. These subgraphs start from the BEGL and BEGR nodes, respectively. Each of these ends with its own End node thus the CM search requires finding a tree starting from Root up to all End nodes instead of a path from Begin to End as in pHMM.

The actual matching node is called a MATX node. It has three forms: MATL/R which matches a nucleic acid on the left/right side, respectively for unbound nucleic acids, and MATP which matches the nucleic acids on both sides of a base pair. Those nodes contain a few subnodes corresponding to matching, deleting or inserting a nucleic acid. The internal structure of these nodes corresponds to their type. All contain Delete (D) and Insert left/right (IL/IR) subnodes. MATP contains both match left (ML), match right (MR) and match pair (MP) nodes whereas MATL/R contains the corresponding matching node

(left or right). Each matching subnode contains the emission probabilities based on the MSA. Figure 1-3 shows an example of a CM model graph.

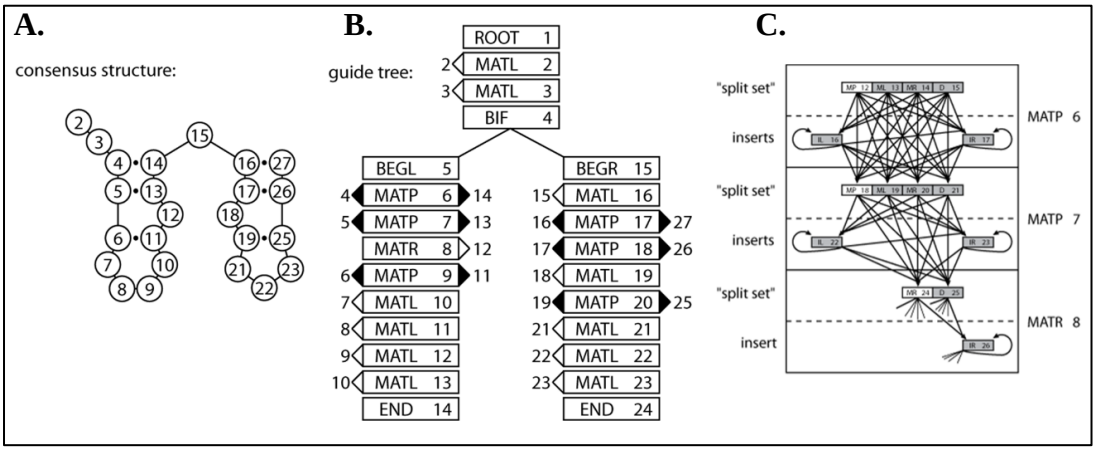

Figure 1-3: Covariance model based on consensus structure. Taken from Infernal [52]. **A.** The consensus structure matching the MSA. **B.** The CM guide tree containing states based on the structure. Note that indexes 4-14 match the left subgraph whereas 15-27 match the right. **C.** Zoom of the MATX states 6 to 8. Since 12 is a bulge to the right we see a MATR state. All states include insert and delete substates to allow for flexibility. A single side can be deleted in the MATP nodes by selecting the match left or match right node.

Whereas the CM introduces important structural information, tools like Infernal rely heavily on pHMM as a pre-filter. An Infernal scan for Rfam uses the entire array of pHMM filters. Running Infernal without pHMM filters is possible, but runtimes can be very long even for small databases. This means that covariance models that include heavy sequence conservation can be run far more efficiently.

1.3. The Purine Riboswitch

Half the known riboswitch aptamers interact with a molecule that contains a purine component, i.e., Guanine and Adenine, a group of riboswitches that bind nucleotide derivatives such as, 2'-deoxyguanine, cyclic-di-GMP and preQ1, and members of another group that bind coenzymes [98]. We focus on identifying the purine riboswitch (Rfam ID RF00167), that binds Guanine, Adenine and 2'-deoxyguanosine. Purine riboswitches have been found in the 5'UTR of prokaryotic mRNAs [58]. The binding of the purine ligand causes a conformational change that can affect either transcription or translation. It can

block or reveal an expression platform for a downstream gene and can also form a translation-termination stem-loop.

The binding pocket is in a multi-loop where it interacts with 4 specific nucleic acids, the most interesting of which is the position 74 nucleic acid where a Watson-Crick base pairing is formed with the ligand. For adenine-sensing riboswitches, position 74 contains a Uracil; for guanine-sensing riboswitches it contains Cytosine [99]. Looking at the conserved RNA sequence and structure from Rfam, we see high sequence conservation (Figure 1-4). This can be attributed to the methods used to search for the purine riboswitch. If we attempt to search using high sequence similarity, we will increase the sequence conservation within the model.

We used our method to attempt to discover novel purine riboswitches in eukaryotes. Many purine riboswitches sequences fold accurately under MFE prediction while that structure is also highly valued within the ensemble. The interactions between purine ligand and aptamer are well defined and researched. Since our methods focus on secondary structure and require only 4 nucleic acids to be conserved for interaction, this allows us to search sequences that would likely be filtered out by the pHMM model on known purine riboswitches.

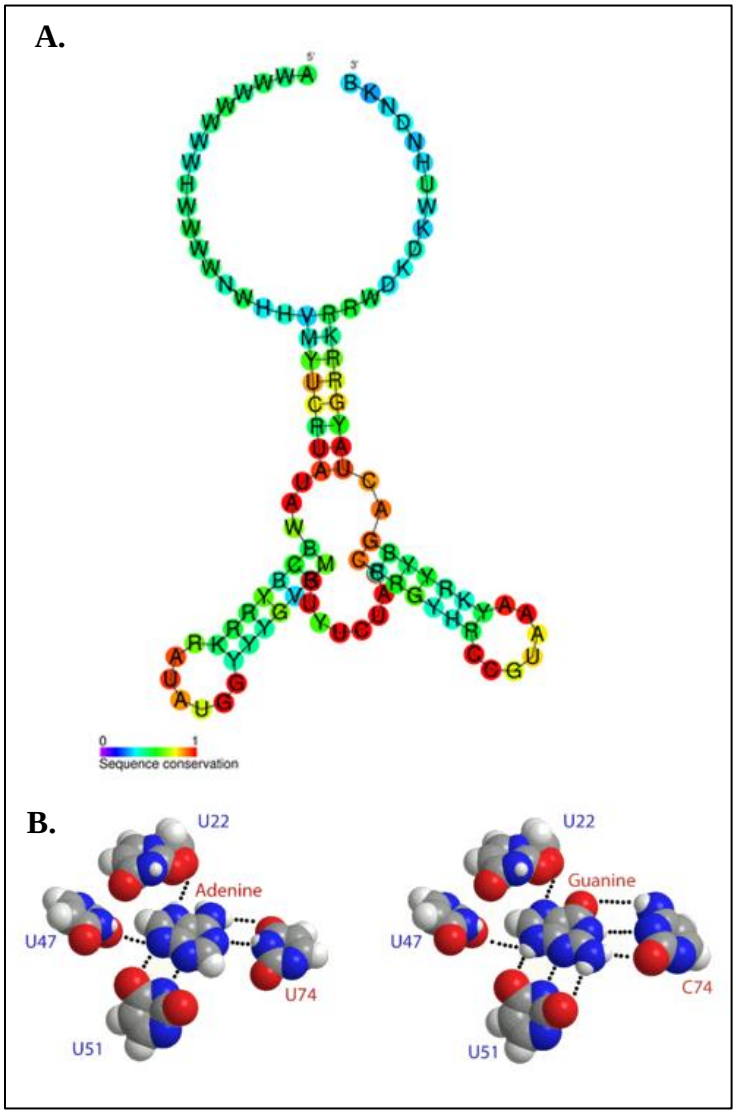

Figure 1-4: Purine riboswitch model. Taken from Rfam family RF00167, Purine Riboswitch. **A.** The consensus sequence and structure for purine riboswitch. The high sequence conservation is visible. **B.** Molecular model of the interaction between ligand and surrounding aptamer. Position 74 affects the binding molecule.

2. RNAPattMatch – Simple and Efficient Pattern-Based Search

2.1. Preface

I developed RNAPattMatch as a tool and web server that allows for RNA pattern matching that includes sequence and secondary structure considerations together with flexible gaps. The tool enables a user with no computational background to search for well-structured riboswitches and other ncRNAs. It is based on a data structure called affix array that enhances the speed of the search based on secondary structure information.

This chapter is based on a published paper: *Drory Retwitzer, M., Polishchuk, M., Churkin, E., Kifer, I., Yakhini, Z., and Barash, D. (2015). RNAPattMatch: a web server for RNA sequence/structure motif detection based on pattern matching with flexible gaps. Nucleic Acids Res 43, 507-12.*

2.2. Introduction

The search for homology of biological molecules is considered an important endeavor in the field of bioinformatics. Searching genomes for peculiar RNAs such as ribozymes and riboswitches [56], as well as other examples of RNA molecules that may possess catalytic activity [100] or ncRNAs that can function as regulators of disease [101], is undoubtedly an indispensable computational task. As mentioned in the Introduction to Chapter 1, sequence homology is often insufficient for comprehensive detection of new riboswitches, and RNA homology search methods that integrate information about the secondary structure of functional RNAs should have a higher specificity and sensitivity. For example, a simple pattern-matching approach that utilizes RNA sequence-structure patterns led to discovery of the first eukaryotic riboswitches in fungi and plants [34]. Finding additional eukaryotic riboswitches in animals in addition to fungi and plants is still one of the main open questions in riboswitch research that extends beyond those riboswitches discovered in prokaryotes [7; 22]. Another example of the use of sequence-structure patterns is in the search for tRNAs [102; 103].

The search methods can vary in sophistication, some may use experimental evidence for the secondary structure and others may also combine secondary structure predictions by energy minimization [71; 73], as was attempted for riboswitch identification in [76; 94]. We note that there are also more elaborate methods to search for riboswitches and other non-coding RNAs in genomes [37; 52; 92] but pattern matching remains a simple and useful approach.

The Denison Riboswitch Detector (DRD) [94] follows other riboswitch detectors that utilize pattern matching [46; 47; 104]. All these riboswitch detectors are available as web servers. However, apart from DRD, all other riboswitch detectors contain known riboswitches and do not allow a flexible definition file of a new riboswitch to be inserted by the user. Even in DRD, inserting a new sequence-structure pattern for which an experimentally derived structure is available is a complicated task for a less experienced practitioner. The definition file expects some parameters such as the minimum number of identities in a global alignment of Vienna strings (strings composed of dots and brackets for RNA secondary structure representation [73]) that are not straightforward to derive for simple sequence-structure patterns as available in [34; 58; 59]. Moreover, DRD considers mismatches, a strategy that was not considered in the simple, yet biologically significant pattern search programs used in [34; 58; 59]. Mismatches will ultimately result in more solutions, however, some of these are spurious. Our aim is to change the sequence-structure pattern to give more flexibilty and to give more solutions that are biologically meaningful. In our opinion, it is more important to consider variable gaps than mismatches and to opt for a user-friendly web server in which the practitioner can insert simple sequence-structure patterns without additional complications.

To this end we developed RNAPattMatch to address the need for a sequence-structure pattern-matching web server, which is flexible in terms of the patterns it allows to search for, and which is user friendly and simple. As elaborated in the next paragraph, the ability of RNAPattMatch to search multiple sequences in a single run enhances its biological significance, since it provides a convincing substitute for in-house programs such as SequenceSniffer [34] and RNA-PATTERN [59], which are not trivial to develop in typical biology labs. Not all experimental labs have the resources to write such in-house

programs and our web server is geared toward providing a convenient and efficient answer to the needs of biology labs that would like to perform a pattern search. The user can upload FASTA files of up to 100 MB that consist of different sequences. For example, RNAPattMatch can scan around a hundred bacterial genome sequences as performed in [59] by uploading one or more target files in FASTA format (most bacterial genomes are under 10 MB, e.g., the complete genome of *T. tengcongensis* MB4 is 2.7 MB) and then searching the multiple sequences in a single or small number of runs. In another example, the user can receive a FASTA file containing BLAST [8] results as performed in [34] and search for the same pattern using RNAPattMatch in a user-friendly manner.

To summarize and put more generally, one distinctive feature of RNAPattMatch is the fact that it accepts FASTA files as target sequences for the search. This allows for flexible usage in several contexts such as: search pattern in multiple genomes (as done in [59]), in transcriptomics data actually measured in metagenome samples, in lists of differentially expressed transcripts or others determined by measurement to have some other molecular property such as binding an RNA-binding protein (RBP) [94]. These capabilities also represent future research directions that will efficiently utilize the computational power of RNAPattMatch. The server was developed along with a standalone C++ that is available at http://www.cs.bgu.ac.il/rnapattmatch alongside the web server. The source code for the application is also available under the GNU public license at https://github.com/matandro/RNApattmatch-client.

2.3. The Affix array data structure

2.3.1. Suffix Arrays

Historically, many data structures were developed to solve pattern matching in text. Suffix trees are indexed data structures that allow for fast pattern matching. Each path from the root to a leaf in the tree outlines a suffix of the target sequence while each branch is a diversion between suffixes with different letters at the position of the height of that vertex. Searches based on this method compare each letter of the query while progressing down the tree, such that once all the query has been compared, the entire subtree of that given vertex are occurrences of the query.

Since suffix trees are a memory-consuming data structure, most modern tools use a more compact version called suffix array or enhanced suffix array [105]. Enhanced suffix arrays introduce a few additions that allow for faster pattern matching. The longest common prefix is computed for each suffix compared with the preceding suffix in the array, and the child table gives us the ability to rapidly find an interval of suffixes with longer common prefixes within a given interval, like walking down a suffix tree. Suffix arrays can be efficiently constructed on a finite alphabet with the DC3 algorithm in a time of $O(n)$Error! Bookmark not defined..

This data structure can perform fast pattern matching in a unidirectional search which progresses along a given query from start to end. The search starts from the interval $[0, N]$ of the suffix array, where N is the length of the string and thus the size of the suffix array. Each time we progress over the query sequence, we compare the next character in the query. If we find a subinterval (that may be the entire interval, depends on LCP) that matches the character, we progress to that subinterval. If we reach the end of the query, the subinterval has the indexes of all the matching results. If we cannot match the entire query, then there are no matches. When matching IUPAC codes, the query may contain wild-card characters that match multiple different suffixes. In such a case we must continue the search separately on multiple subintervals, each matching a relevant character.

2.3.2. **Affix Arrays**

RNA sequence-structure pattern matching introduces additional information not used in a search based on suffix arrays. The unidirectional search runs over the entire query string from start to end and ignores the additional information regarding base-pairing until it reaches the closing base. For this specific reason the affix tree, or its more compact version the affix array, were introduced. The affix array uses a suffix array in conjunction with a reverse prefix array, which can be described as the suffix array of the reverse target string and an additional set of links between them. Those affix links connect an interval from the suffix array to an interval in the reverse prefix array, such that the longest common prefix representing the suffix interval is equal to the end of the longest common suffix of the interval in the reverse prefix array. Similar links are connected in the reverse direction (Figure 2-1).

This new data structure allows us to perform a bidirectional search. In such a search we can use structural information on the RNA to speed up the process. Since sequence patterns may include many wild-card IUPAC codes, they may match multiple intervals in the target sequence. Any time the structure introduces a base pair, matching a nucleic acid on one side of the base pair may clue us regarding the nucleic acid on the other side. For example, matching a G forces the other base to be a C or a U. The search can use this information to reduce the number of intervals that are still relevant by looking only at those that have C or U in the opposite strand.

i	$suf_F[i]$	$lcp_F[i]$	$aflk_F[i]$	$S^F_{suf_F[i]}$	$(S^R_{suf_R[i]})^{-1}$	$aflk_R[i]$	$lcp_R[i]$	$suf_R[i]$	i
0	2	0	0	AGCUGCUGCUGCA	AUAGCUGCUGCUGCA	0	0	0	0
1	0	1		AUAGCUGCUGCUGCA	AUA		1	12	1
2	14	1		A	A		1	14	2
3	13	0	3	CA	AUAGC	7	0	10	3
4	10	1	4	CUGCA	AUAGCUGC	8	2	7	4
5	7	4	5	CUGCUGCA	AUAGCUGCUGC	9	5	4	5
6	4	7		CUGCUGCUGCA	AUAGCUGCUGCUGC		8	1	6
7	12	0	3	GCA	AUAG	7	0	11	7
8	9	2	4	GCUGCA	AUAGCUG	8	1	8	8
9	6	5	5	GCUGCUGCA	AUAGCUGCUG	9	4	5	9
10	3	8		GCUGCUGCUGCA	AUAGCUGCUGCUG		7	2	10
11	1	0	11	UAGCUGCUGCUGCA	AU	11	0	13	11
12	11	1	4	UGCA	AUAGCU	8	1	9	12
13	8	3	5	UGCUGCA	AUAGCUGCU	9	3	6	13
14	5	6		UGCUGCUGCA	AUAGCUGCUGCU		6	3	14
15	15	0					0	15	15

Figure 2-1: Affix array data structures. Taken from Structator [63]. Matching the sequence "AUAGCUGCUGCUGCA". This includes suffix array, reverse prefix array and matching the longest common prefix and affix links arrays. Aflk is the affix links array. We can see that the index 4 in the suffix array matches index 4 in the reverse prefix array because it ends with CUGC which is common for this interval.

Whereas the unidirectional search kept progressing over intervals in the suffix tree, the bidirectional search skips from suffix array to reverse prefix array via the affix links every time a base pair is encountered. The links ensure that the end of all the suffixes in the reverse array contain the information matched prior to the skip. With that, we can match the second base in the pair as the first letter in the reverse array and thus filter out many intervals. Figure 2-2 shows an example of the reduced number of intervals to test on a simple hairpin. Affix arrays can handle single stem-loop motifs efficiently. They do not work if there is more than one. Thus, for a full search the query sequence must be

deconstructed into multiple stem-loop motifs. Each motif must be searched separately via the affix arrays. Once these are found the resulting intervals must be matched to construct the full query results.

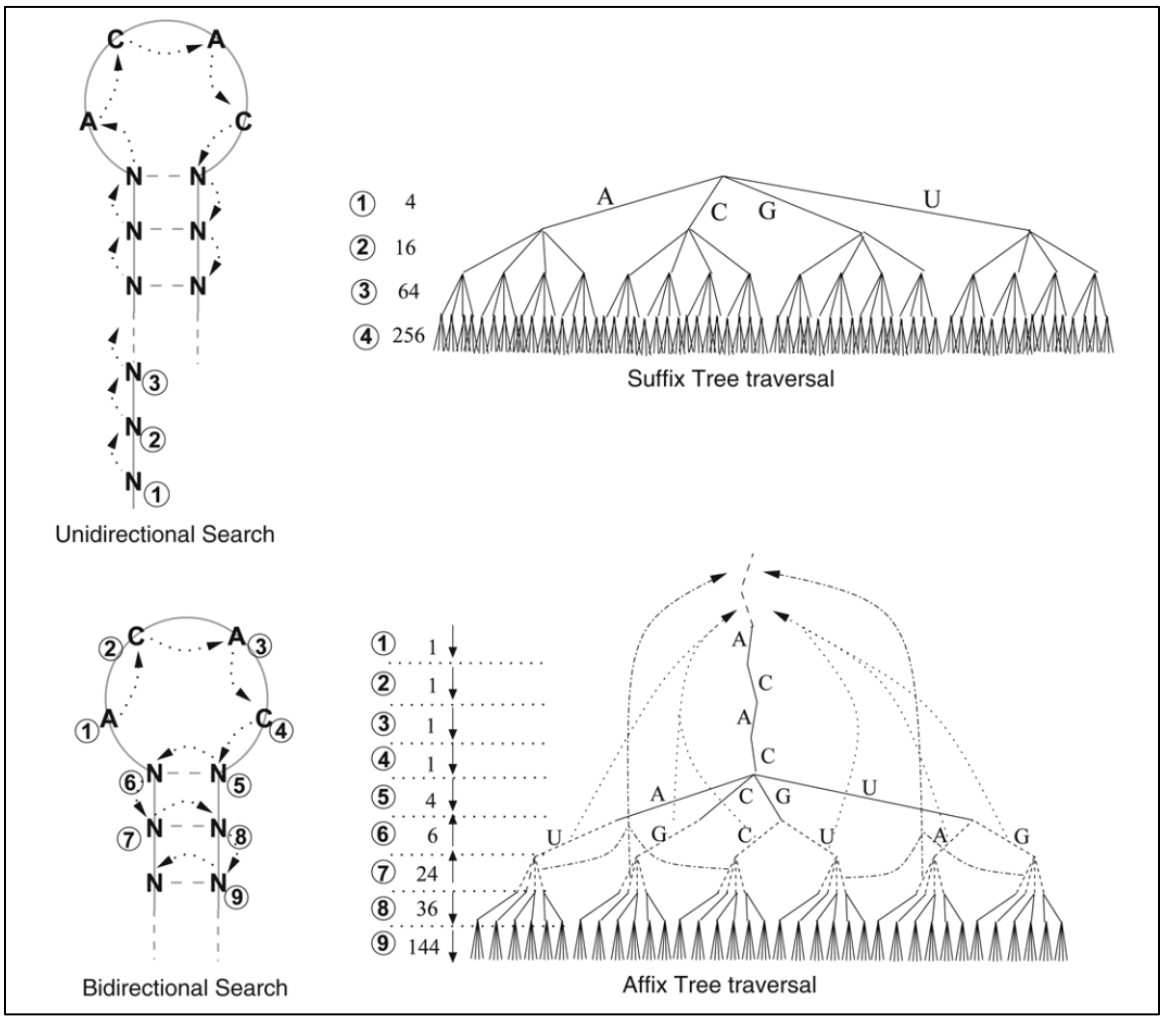

Figure 2-2: Unidirectional vs Bidirectional search. Taken from Structator [63]. In a unidirectional search the 'N' IUPAC code can fit any of the four nucleic acids increasing the number of intervals exponentially. With a bidirectional search we start from the hairpin and when arriving at the stem, the comparison alternates across it. The selection in level 5 reduces the selection for level 6 for legal base pairing.

2.4. Implementation

2.4.1. Input, Output and Data Structures

The input for the software includes a query RNA sequence and structure and a target FASTA file. The sequence is defined as a string of IUPAC RNA codes (excluding the '.' Or '-' symbols) and gap marks. The gap marks are numbers surrounded by square brackets that indicate a maximum gap at the given position. The structure is in dot-bracket format which includes '.' Symbol for unbound nucleic acids and round brackets for bounded ones. It must also include the gaps in the same format and index as in the sequence. Note that the structure must have balanced brackets. The FASTA file contains the sequences in which the pattern will be searched. The web server has a limit of 100 MB FASTA files, however, the application has a 2 GB limit based on the use of signed integers. The output is a list of sequences with the FASTA header, start index and an array with the number of gaps used in each square bracket (Figure 2-3). The application also accepts the number of threads, custom base pairing rules, result limit and an option to save or load the data structure from prior runs.

Suffix array and reverse prefix array were built using the DC3 algorithm on each of the sequences in the FASTA file. The longest common prefix was calculated for each of them. Affix links are calculated on demand because of the high computational cost. The application supports caching of data structures for future use. Moreover, the web server allows the user to perform multiple searches on a submitted FASTA file. The search itself is done on each of those FASTA targets separately.

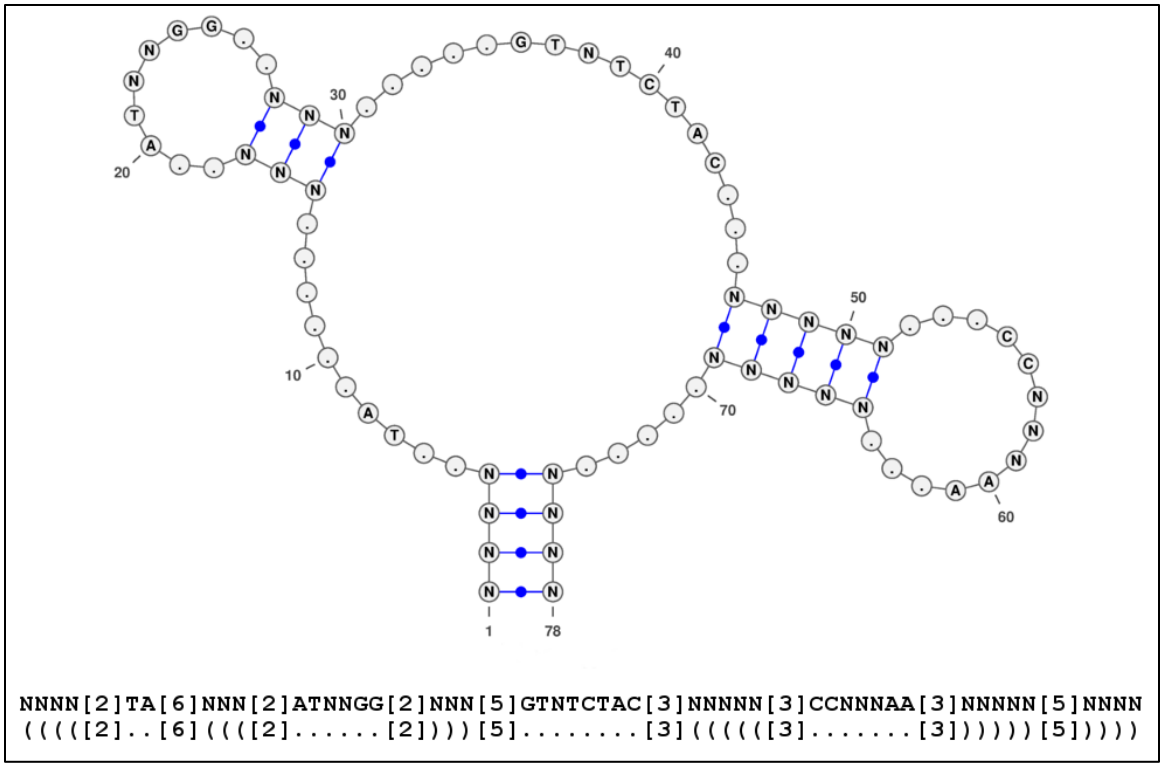

Figure 2-3: RNAPattMatch input query example. Above an illustration of the input. '.' Marks gaps results that can be used up to all given points. Below the query sequence and structure input.

2.4.2. **Motif Breakdown and Merge**

The query sequence and structure are broken into stem-loop motifs. Each of those motifs must be searched on the FASTA database individually. To improve performance, we calculate a score for each motif based on its specificity. The score approximates the increase factor for the search tree (Similar to Figure 2-2). For example an 'N' IUPAC code can match 4 nucleic acids which increases the cost by a factor of 4; irrespective of whether it is found in an unbound section or in a base pair as the complementary strand does not increase the cost at all (excluding G-U base pairs).

Starting from the motif with the lowest score, which we assume is more specific and less abundant in the database, we run an affix search. The affix search returns a list of results, each defined by start index and list of used gaps. From there we go to either the next or the previous motif based on the lowest score. When we have the results for that second motif, we sort both lists by indexes. Since we know how the sequence between the motifs is represented, we merge results that have the correct distance between them. This

reduces the number of duos that need to be tested. If a duo is matched according to the sequence and structure restrictions in the query, it is then merged and considered as a single result. Once all the motifs have been merged, we are left with results that match the entire query.

2.4.3. **Search Tasks**

We define a bi-directional search task as a single search instance on a single stem-loop motif. The search task contains the following parameters: query forward index, query reverse index, direction, gap usage, target index and suffix/affix array interval. The interval represents all the matches for the query section that was tested. Since an affix search starts from the hairpin of the step-loop we need to recall both forward query location and reverse. A search task performs a comparison for the query, if a single match is possible (matching A, C, G or U) it continues, otherwise it generates multiple new search tasks. Each task is separated with a different part of the suffix/affix interval. This is done since the interval contains all the sequences that are equal up to the longest common prefix which is everything that was matched from the current direction. But once we are required to continue the search for different values where the interval is no longer viable, the single interval much be split into subintervals in which the next nucleic acid is different. This can be visualized in Figure 2-2. This is the same as done when a gap is encountered. A gap is like an ‘N’ character where the query index does not progress, but the target index does. Since we support a limited number of gaps at each defined position, we need to generate an additional task for any number of gaps up to the defined limit. Adding gaps increases the computational complexity and run time of the pattern. If the query location is the initial tested side of a base pair, we change direction and use the affix link to replace our current interval with the complementary one in the reverse suffix array. This may still generate a new search task if the initial side was matched to G, since the opposite side can match with both C and U.

Since each search task is defined to a specific part of the query and to a specific interval, the tasks are independent. Therefore, we use a dispatcher pattern. Increasing the number of threads available for the algorithm increases performance by a factor of the number of threads. Each thread takes a single search task from a work queue and runs it

until a result is matched or the interval no longer matches the query. Once this occurs, the thread can move on to the next search task. When the list is empty and no search tasks are currently running, the affix search for the specific stem-loop motif is done.

Query	***Thermoanaerobacter tengcongensis MB4 (2.7 mb)***	***Saccharomyces cerevisiae sacCer3 (12mb)***	***Human chromosome 16 hg38, GRCh38 (89mb)***
Guanine-binding riboswitch aptamer[A]	2(s)[B]	6(s)	127(s)
Hairpin with G-C stem[C]	7(s)	29(s)	149(s)

Table 2-1: Run time for RNAPattMatch. Queries are available on the web server as examples. Only the target *T. tengcongensis* MB4 is available as an example. **A.** Matches for a guanine riboswitch were not found in the eukaryote organisms reported in the table. **B.** Running times were taken from the RNAPattMatch web server for non-cached targets and do not include file upload time. **C.** Difficulty is indicated by the specificity of the query and the amount of hairpin loops to merge. In this case, the G-C rich hairpin has no specific nucleic acid in the unbound section.

2.4.4. **Web server**

The RNAPattMatch web server (http://www.cs.bgu.ac.il/rnapattmatch) runs on a Unix IBM x3550 M4 server with Quad Intel(R) Xeon(R) CPU E5-2620 2.00 GHz processors containing six logical cores and 15 MB L3 cache each. Memory size is 64 GB to allow for the extensive memory needed for the affix arrays data structure. Every search task runs on up to four cores depending on the load. The server runs up to 10 simultaneous search tasks while the rest wait in a queue.

The input screen (Figure 2-4) accepts a query sequence and structure, base pairing rules and a FASTA target. The query sequence and structure are as described in 2.4.1. Base pairing rules can be set to Watson-Crick with or without dangling G-U pairing or a custom base pairing matrix. The target file can be uploaded or given by GenBank accession number. If uploaded, the file size is limited to 100 MB. The user can add his/her email

address and set a name for the query. The results can be searched by the given name or by ID and a notification email is sent when the search is done.

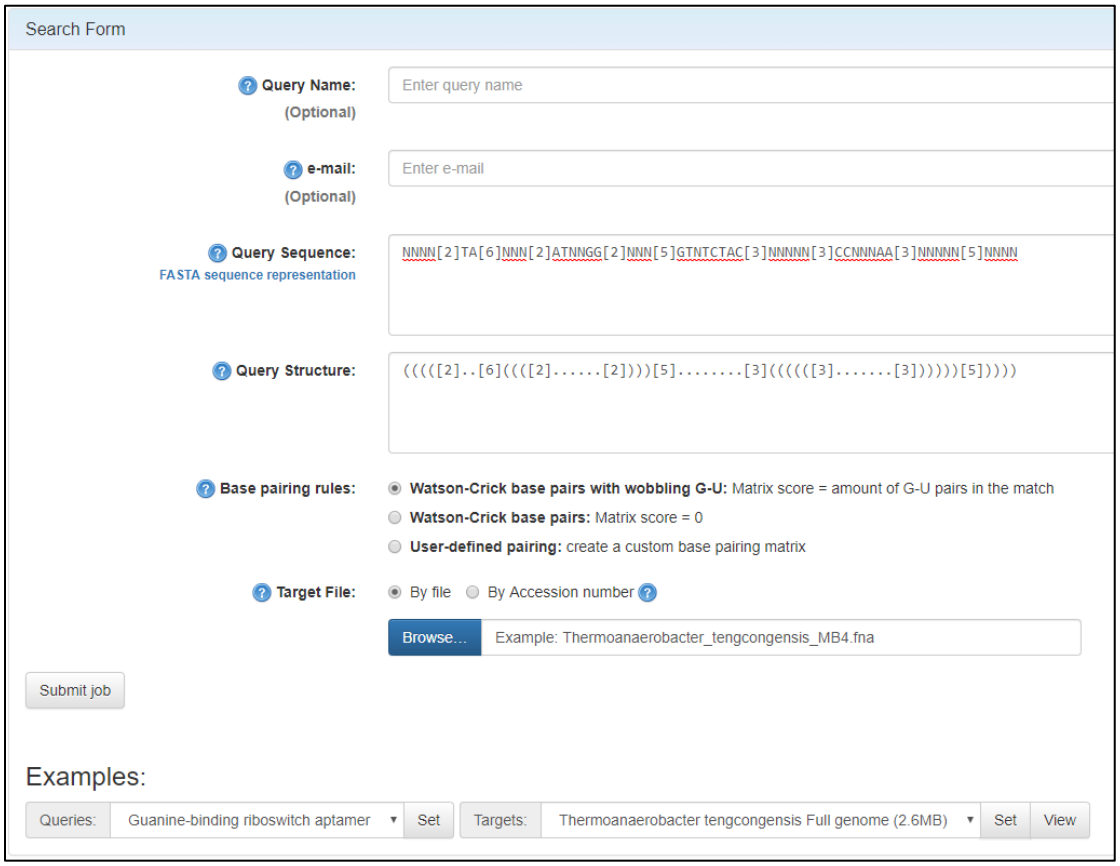

Figure 2-4: RNAPattMatch input screen. Example inputs for a guanine-binding riboswitch aptamer on the full genome of *T. tengcongensis* with default base pairing rules.

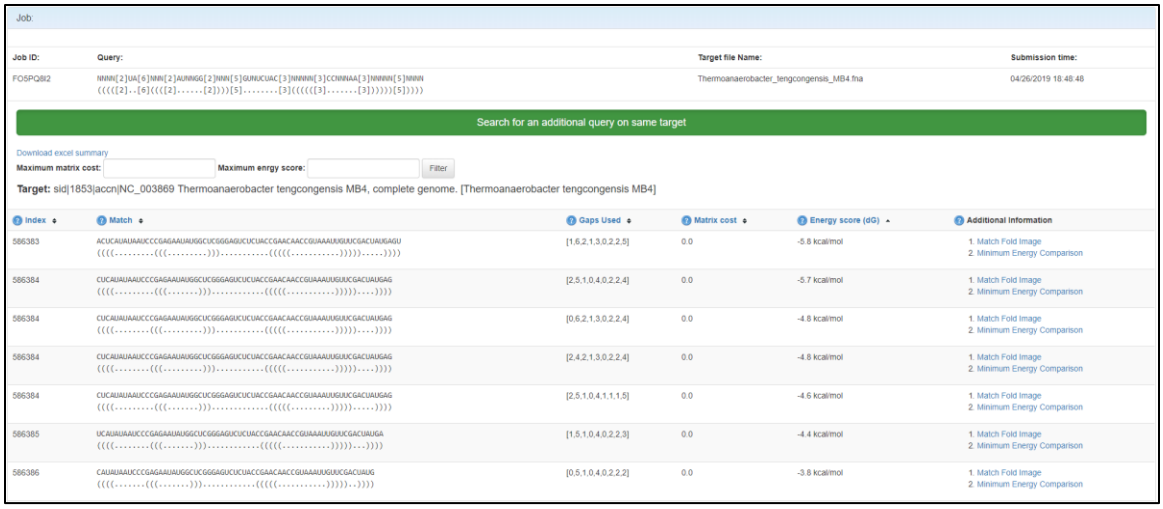

Figure 2-5: RNAPattMatch output screen. Results for the Guanine-binding riboswitch aptamer on the full genome of *T. tengcongensis* with default base pairing rules. Note that many results will appear on the same index but with different structures based on the gaps used.

The results are kept for one week from submission. They can be downloaded in excel format or viewed on site. The output screen presents all the results including an alignment of the sequence and structure based on the gaps used (Figure 2-5). Additionally, a free energy score is presented, calculated by the Vienna RNA package [74] on the resulting sequence structure combination. The last item is a comparison screen between the aligned sequence structure to a minimum free energy structure calculated by the Vienna RNA package (Figure 2-6). The images are generated using SIR graph from Mfold [71]. The screen also presents the Shapiro structure for motif-based comparison.

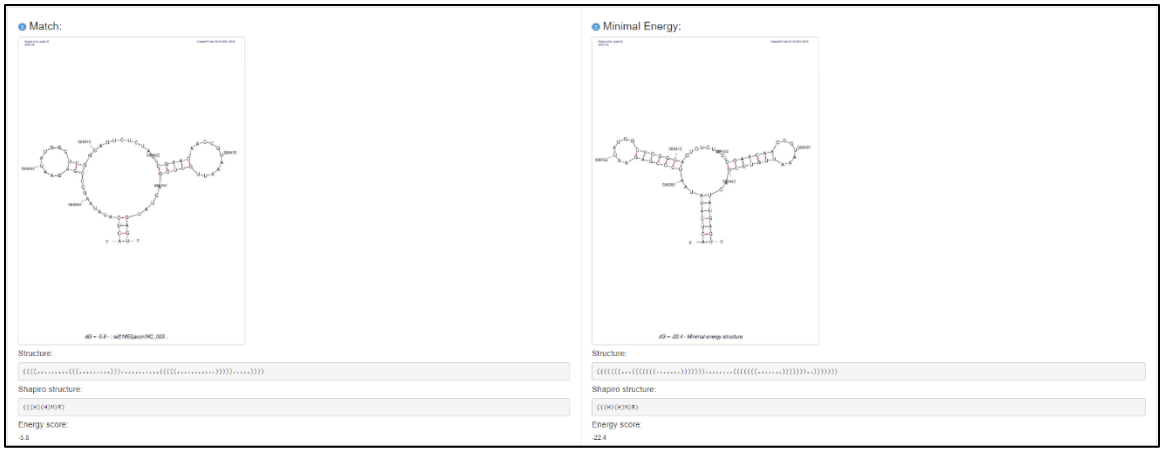

Figure 2-6: RNAPattMatch comparison screen. Results for the Guanine-binding riboswitch aptamer. Folding the sequence using the Vienna RNA package induces a structure with more base pairs as many of the base pairs were matched to gaps that are considered unbound by default.

2.5. Conclusions

Homology searches for RNAs in genomes are limited if based on sequence considerations alone and finding sequence-structure patterns offers an improvement in cases where the pattern is known in advance. Indeed, such patterns are available in practice, e.g., [34; 59], and there is a clear biological motivation for their search. Several programs have been developed over the years to find sequence-structure patterns, but none are available as a user-friendly web server that can accommodate practitioners of various backgrounds. Here we present a new web server called RNAPattMatch that fulfills this need. It is based on the methodology implemented in the program RNA Structator that is not available as a web server [63] and offers a significant extension to Structator, by addressing variable gaps and providing a comprehensive analysis of results with RNA folding prediction by energy minimization including secondary structure drawings.

The RNAPattMatch web server was developed with the goal of making the efficient method of using affix arrays and a dynamic programming search algorithm available for the entire biological community. The web server is user-friendly and accessible to practitioners, both in terms of ease of use and simplification of the output. We believe that it will serve experimental groups for improving their capability to perform RNA sequence-structure pattern searches.

When used in a search for novel purine riboswitches, the pattern should be optimized using multiple search cycles. Initial patterns should include gaps in multiple locations for complete flexibility which was shown to increase run times. Since the search was performed on large-scale genomic databases, the approach is inefficient. RNAPattMatch is still a superior tool in the realm of sequence-structure based pattern matching in both efficiency and flexibility. It can be used as a replacement for SequenceSniffer [34] in search pipelines for ncRNAs.

3. incaRNAfbinv 2.0 – Fragment Based Design with Motif Specific Control

3.1. Preface

We developed a new web server, IncaRNAfbinv 2.0, as an RNA design tool targeted for flexible RNA sequence design. The design process comprises two parts: Seed preparation using incaRNAtion [68] and specific design based on RNAfbinv 2.0. The features presented in RNAfbinv 2.0 were introduced to support the search process presented in Chapter 4 below. IncaRNAfbinv 2.0 is described in the following 2 publications:

1) *Drory Retwitzer, M., Reinharz, V., Ponty, Y., Waldispühl, J., and Barash, D. (2016). incaRNAfbinv: a web server for the fragment-based design of RNA sequences. Nucleic Acids Res 44, 308-14.*
2) *Drory Retwitzer, M., Reinharz V., Alexander C., Ponty Yann., Waldispühl J., and Barash D. (submitted). IncaRNAfbinv 2.0 – A web server and software with motif control for fragment-based design of RNAs. Bioinformatics, Application note.*

3.2. Introduction

The design of RNAs with desired traits is a promising endeavor that can be viewed as part of growing efforts in synthetic biology [106], as well as other applications. For example, it can be used to enhance the search for RNAs such as ribozymes and riboswitches in sequenced genomes [56], as well as other non-coding RNAs that may act as regulators of disease [101] or participate in catalysis [100]. For riboswitches [7; 22], aside from the classical problem of computationally designing transcription regulators and validating them experimentally [107; 108] to complement pure experimental designs [109-111], the inverse RNA folding problem that was initially formulated and addressed in [67] can be used as a pre-processing step before BLAST [8] for riboswitch identification [112]. This recent use was also worked out for IRES-like structural subdomain identification in [69]. .It has the potential to advance the field described in [92] for conserved RNAs in general

Thus, computational RNA design is of increasing biological importance. Since the first program for solving the inverse RNA folding problem (or RNA design) called RNAinverse was put forth in [67], several other programs have been developed. The approach to solve the inverse RNA problem by stochastic optimization relies on the solution of the direct problem using software available in RNA folding prediction tools, e.g. the RNAfold [73] or Mfold [71], by performing energy minimization with thermodynamic parameters [75]. Initially, a seed sequence is chosen, after which a local search strategy is used to mutate the seed which is then repeatedly applied for RNA folding prediction by energy minimization. Finally, in the vicinity of the seed sequence, a designed sequence is found with the desired folding properties according to the objective function in the optimization problem formulation.

In recent years, several programs for RNA design have been developed with the goal of offering additional features with respect to the original RNAinverse, most of which are general in purpose for solving similar problems [113-121] and a few are more specialized for nanostructure design [122] and for fixed-backbone 3D design [123]. Recently, an extension to the problem was incrementally developed [66; 124; 125] that allows designing sequences that fold into a prescribed shape, leaving some flexibility in the secondary structure of RNA motifs that do not necessarily possess a known functional role. This extension, which offers the user a fragment selection, is called ‘fragment-based’ design because it is based on a user-selected secondary structure motif (the fragment) that possesses a functional role and is therefore inserted as a ‘fragment-based’ constraint to the design problem. The shape of the RNA can be represented as a tree-graph [85] that groups together a family of RNA secondary structures, all belonging to the same coarse-grained graphical representation.

The aforementioned extension led to a unique inverse RNA folding program called RNAfbinv [66] that is more general in scope than any existing program in its shape-based approach. In that regard a shape-based approach can generate a larger number of designed sequences that match the same design target. Sequence constraints was available as a feature in RNAfbinv, however, it was disconnected from the fragment design as it was compared via specific indices. This allowed for designs where critical nucleic acid bases

were matched in the correct index but not the correct fragment. RNAfbinv 2.0 fixes this issue and allows for even more control on the final design while still targeting a coarse-grained structure.

In parallel, we presented a web server that addresses the possibility to control nucleotide distribution in RNA design problems in a unique way by a weighted sampling approach [68]. This approach is of general importance for the future of inverse RNA folding because instead of a random start for performing a local search, we carefully pick the initial sequence for performing the iterative procedure of solving the inverse problem using global considerations in a guided manner in the search space. The program for the weighted sampling approach called incaRNAation has so far only been exemplified in [68] for RNA design. The IncaRNAfbinv 2.0 web server we describe here merges RNAfbinv 2.0 with incaRNAtion [68]. It offers sequence design solutions that are not available in either the most recently published programs for RNA design, namely antaRNA [121] and RNAiFold [113], or similar programs that have been devised subsequent to the seminal program, RNAinverse from the Vienna RNA package [67]. It should be noted that IncaRNAfbinv 2.0 relies on other programs in addition to RNAinverse from the Vienna RNA package such as RNAfold that solves a direct problem in each iteration. The web server is available at https://www.cs.bgu.ac.il/incaRNAfbinv/. RNAfbinv 2.0 is available as a python 3 package named *rnafbinv* on pypi (Python Package Index).

3.3. Implementation

3.3.1. incaRNAtion

IncaRNAtion [68] addresses RNA design in a complementary way. Rather than preventing the formation of alternative secondary structures (negative design principle), it stochastically produces sequences with high affinity toward the target structure S^*, as measured by its free-energy (positive design principle). To that purpose, a pseudo-Boltzmann distribution is postulated on the set of sequences compatible with S^*, where the probability of emitting an RNA ω for a given pseudo-temperature T is proportional to $e^{\frac{-E_\omega(S^*)}{kT}}$, where $E_\omega(S^*)$ is the free-energy of S^* upon an RNA sequence ω and k is the suitably-dimensioned Boltzmann constant. A linear-time dynamic programming algorithm

is then used to generate sequences at random taken without change from the pseudo-Boltzmann distribution, resulting in candidate designs whose affinity toward S^* ranges from extreme to reasonable, depending on the value of T. Further terms can be incorporated into the free-energy function, and combined with a provably efficient rejection step, to control the equation GC-content of the output sequences.

Preliminary analyses [68] revealed that incaRNAtion produces sequences that are more diverse than those obtained using competing algorithms. Furthermore, we show that sequences designed by incaRNAtion could be used as seeds for algorithms that implement negative design principles, increasing the diversity of their final output, while usually retaining the general properties (high-affinity, prescribed equation GC-content) enforced by incaRNAtion in its initial generation.

3.3.2. **RNAfbinv 2.0**

RNAfbinv 2.0 applies a 4-step lookahead simulated annealing approach like the initial RNAfbinv program [66]. It is available as a python 3 package named *rnafbinv* on pypi, a GUI wrapper is also available at https://github.com/matandro/RNAsfbinv/. The designed sequence is based on a given secondary structure and includes sequence constraints which will be referred to as target sequence and structure. Additional optional parameters are possible such as a starting seed sequence, motifs to be preserved, length flexibility, target free energy in kcal/mol, target mutational robustness and number of simulated annealing iterations.

The application starts with a given seed that is sent to RNAinverse [67]. In each iteration, the sequence is mutated and scored. The score is used to decide whether the mutation is eliminated or kept in the next iteration where smaller is better. The 4-step lookahead means that a sequence may be mutated four times before it is eliminated, and the prior sequence is used. A sequence with a higher score may still be used for the next iteration, the lower the round, the higher the chance for exploration of higher scored sequences. Score calculation is done by comparing the target sequence structure to the current designed sequence.

The designed sequence is folded using RNAfold [67]. The structure is then reconstructed to tree form, where each node represents a single motif. The nodes also conserve sequence information relevant to the motif (Figure 3-1). We define two types of nodes: a bounded motif (Stem) or an unbound motif (Loop, Bulge or Exterior region). Two motifs are comparable if both are of the same type. If two nodes are matched, a score is calculated based on sequence alignment between the sequences attached to the node. Sequence alignment is done per sequence segment by order. For example, an internal loop has two unconnected sequences while a bulge motif has only one. Matching the two nodes will only align one of the sequences of the internal loop while the other will be calculated as if removed. By default, each deletion of an 'N' nucleotide from the target motif results in a small penalty whereas deletion of any other IUPAC symbols incurs a very large penalty. Removing a node includes a medium penalty plus the removal of the attached sequences.

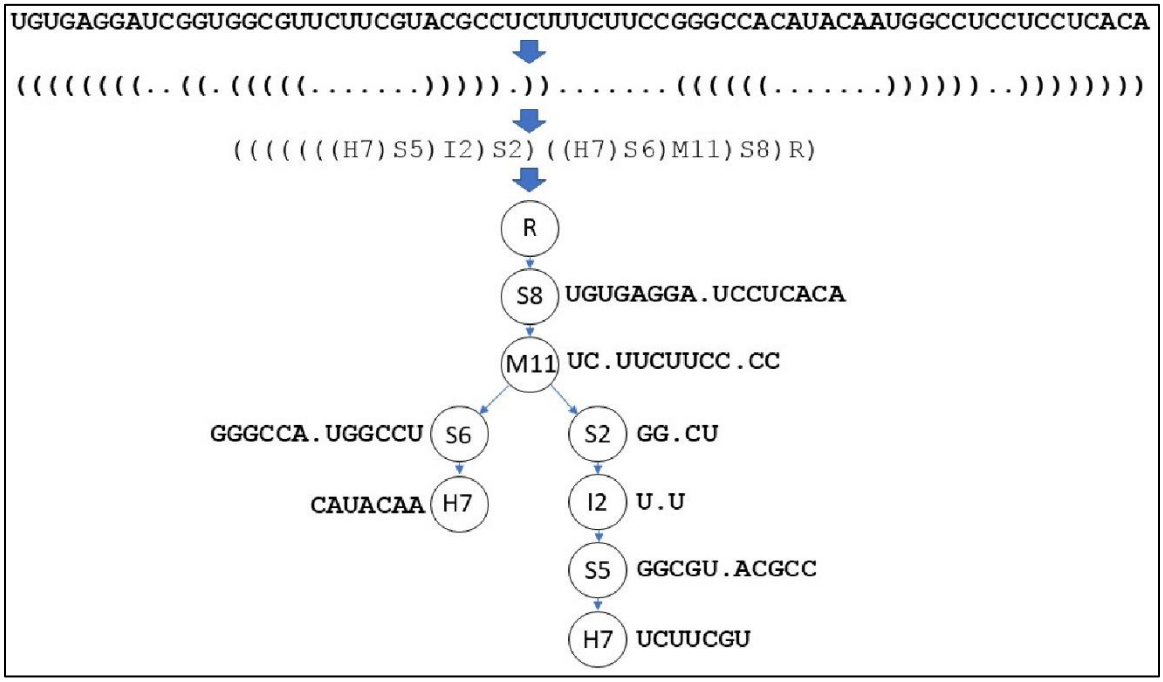

Figure 3-1: Tree construction. Starting from an RNA sequence, secondary structure is calculated using RNAfold [67]. Shapiro structure is then generated for the structure. In the last phase a tree is constructed as a combination of all the prior data. Unlike design sequence, target secondary structure is taken from input. The only motifs that might have more than a single child are Multi-loops and External regions.

The final score is a combination of multiple values as such:

$$Score = \left|neutrality_{target} - neutrality_{design}\right| \cdot 100 \mid \left|dG_{target} - dG_{design}\right| + AlignmentScore(target, design)$$

$AlignmentScore$ is the best alignment of both trees, and is implemented as a dynamic programming algorithm that minimizes the following equation:

$$AlignmentScore(target, design) =$$

$$\min \begin{cases} (*)AlignSequences(target.seq, design.seq) + ChildCombination(target, design) \\ \min\limits_{child \in target.children} AlignmentScore(child, design) + AlignSequences(target.seq, "") + 100 \\ \min\limits_{child \in design.children} AlignmentScore(target, child) + AlignSequences("", design.seq) + 100 \end{cases}$$

$(*)\ if\ comperable$

$ChildCombination$ finds the best alignment between the children of target and design trees. If a motif has multiple children, a minimal alignment is calculated as a combination of $AlignmentScore$s of the different children while maintaining order. Removing a target motif marked as preserved will result in a penalty of 1000 instead of 100. It is important to note that mutational robustness (marked as neutrality) greatly increases run times. Neutrality is calculated by summing the base pair distance between the design sequence and all single point mutation sequences and dividing it with the maximum sum distances.

RNAfbinv 2.0's new score is truly motif-based and gives the user full control of each motif while allowing for fragment-based design flexibility. The prior version RNAfbinv had rigid constraints besides comparing motif symbols based on Shapiro structure. This allows us to design sequences of varying length and attach sequence constraints to the motif itself (Figure 3-2).

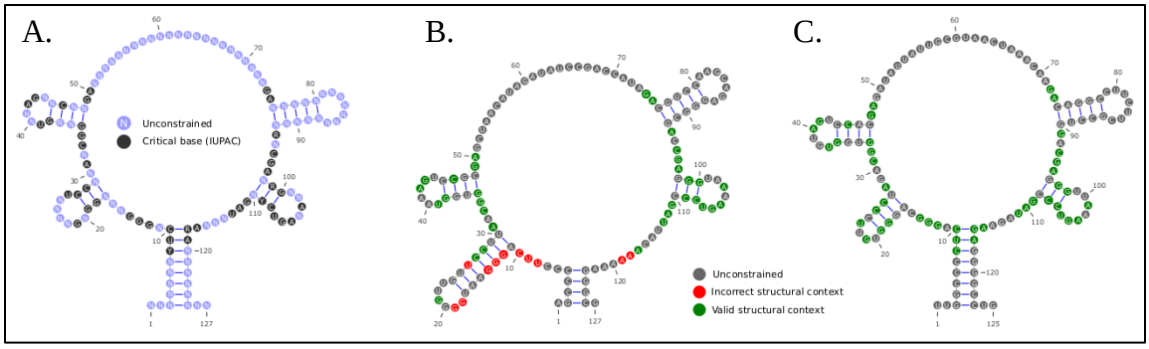

Figure 3-2: Illustration of the differences between incaRNAfbinv 1.0 and incaRNAfbinv 2.0 for the FMN riboswitch aptamer. A. Target sequence and structure. Critical nucleic acid bases indicated in black. **B**. Typical design output of incaRNAfbinv 1.0. Sequence constraints were satisfied but not always properly in the correct structural context (red color). **C**. Typical design output of incaRNAfbinv 2.0. Sequence constraints are not only satisfied, but also shifted to match their original structural context.

3.3.3. **Web Server**

The web server is available at https://www.cs.bgu.ac.il/incaRNAfbinv/. It runs on a Unix Lenovo system x3650 M5 server with Dual Intel(R) Xeon(R) CPU E5-2620 v3 2.4 GHz processors containing six logical cores and 15 MB L3 cache each. The backend is written in Java EE and run on Tomcat 8. It dispatches design tasks responsible for running incaRNAtion [68] and RNAfbinv 2.0. Each design task runs on up to four cores depending on load. The server runs up to ten simultaneous design tasks while the rest wait in a queue. The frontend is designed using the Bootstrap css framework. Web pages are generated using JSP and JSTL. They utilize JavaScript, Jquery, JSON and ajax.

The user must input a target structure and may add optional sequence constraints. GC-content can be selected for incaRNAtion seed preparation. Additional optional parameters from RNAfbinv 2.0 are available; some are shown only when the Advanced Options checkbox is activated (Figure 3-3). Whenever a legal structure is inserted, an image of that structure appears showing a list of motifs. Selecting motifs will tag them for preservation. Note that the use of incaRNAtion is optional, the user may choose a manual or random seed.

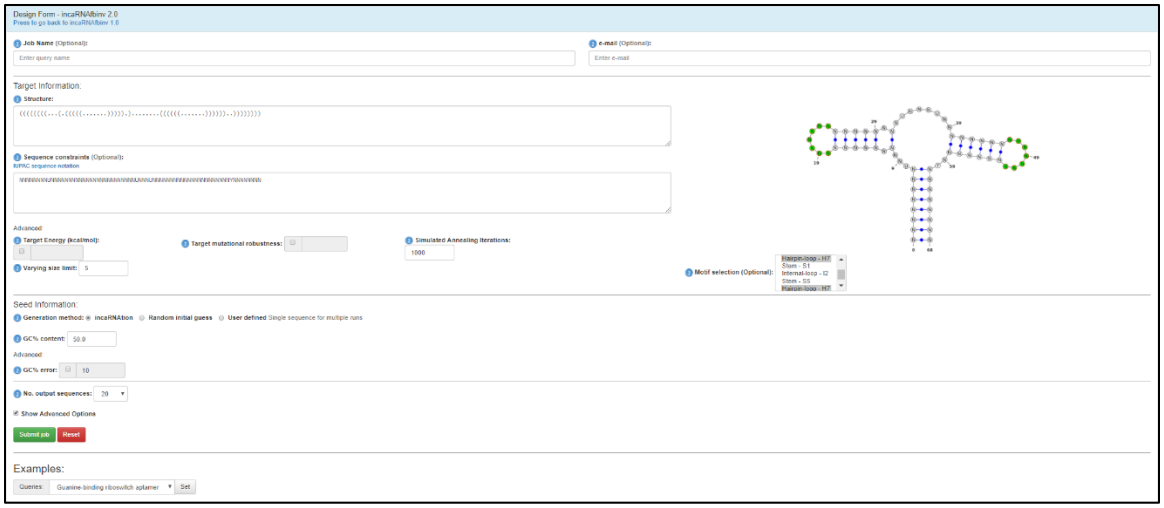

Figure 3-3: IncaRNAfbinv 2.0 web server design screen. Input corresponds to a guanine-binding riboswitch (available as an example). The image on the right is generated using VARNA [126]. The bases marked in green belong to a motif selected from the list for preservation. The image will update automatically upon selection and when a balanced structure is inserted.

Upon completion, a table with all designed sequences will appear. Each sequence is shown with its minimal energy structure and Shapiro structure. Base pair edit distance, Shapiro distance and RNAfbinv 2.0 score are calculated between the design and target. Additionally, GC-content of the target sequence is shown (Figure 3-4). Selecting the fold image will generate an image of the design sequence structure. Constrained nucleotides that are in the correct motif are marked with green filling.

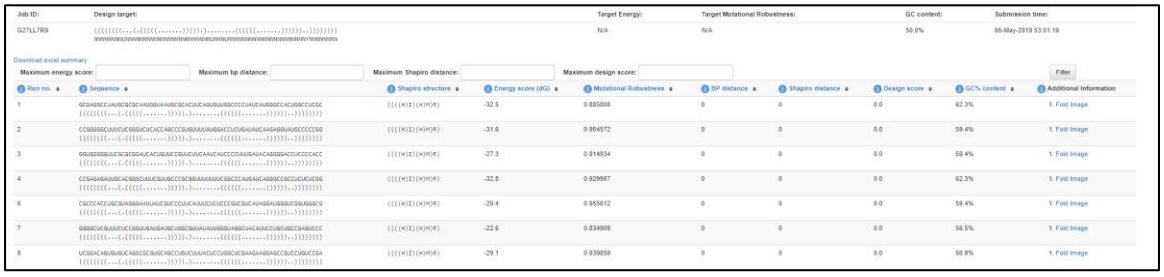

Figure 3-4: IncaRNAfbinv 2.0 web server result screen. Output corresponds to a guanine-binding riboswitch. We can see the sequences are matched to the exact sequence constraints and structure. Longer targets may result in higher scores. Design scores above 100 imply change in motif while scores above 1000 imply missing sequence constraint.

3.4. Conclusions

When solving the inverse RNA folding problem, it is important to be able to address biological constraints in the forms of structural constraints, as well as physical observables and sequence constraints. Recently developed new programs such as antaRNA [121] and RNAiFold [113] attempt to address these constraints but are limited in their scope: they cannot handle fragment-based constraints like the ones handled in RNAfbinv 2.0, nor can they handle GC-content in the same structured and efficient way as does incaRNAtion [68]. These types of constraints can substantially improve targeted design of RNA sequences in cases where a biological-driven constraint is known in advance. The uniqueness of the fragment-based design approach combined with the weighted sampling approach that traverses the search space in a guided manner merits a user-friendly web server that can accommodate practitioners of various backgrounds. We therefore present a new web server called incaRNAfbinv that fulfills this need. It is based on the methodologies implemented in the programs RNAfbinv 2.0 and incaRNAtion, neither of which is available as a web server. It offers a significant extension to programs for RNA design that do not consider the aforementioned advanced constraints and are limited to strictly obeying the RNA secondary structure of the input as in the original and well-accustomed formulation of RNAinverse [67]. Even small deviations from it can produce a designed sequence that can much better accommodate the biological constraint imposed based on prior knowledge.

The IncaRNAfbinv 2.0 web server was developed with the goal of making the unique methods of fragment-based design with RNAfbinv and targeted weighted sampling with incaRNAtion available for the entire biological community. The web server is user-friendly and accessible to practitioners, both in terms of ease of use and simplification of the output. RNAfbinv 2.0 itself was forged to add the control required by the search method described in chapter 4. The improvement helped design flexible yet coherent sequences for the purine riboswitch.

4. A New Energy Minimization Structure-Based Search Method for Riboswitches

4.1. Preface

Here we describe a novel method for riboswitch detection based on energy minimization of RNA secondary structure. The method is a "design to search" pipeline wherein sequences are designed to match the purine riboswitch aptamer using IncaRNAfbinv 2.0. These sequences are considered potential riboswitch aptamer candidates and can be searched against large genomic databases. The pipeline is an improved version of the method described in the following paper: *Drory Retwitzer, M., Kifer, I., Sengupta, S., Yakhini, Z., and Barash, D. (2015). An Efficient Minimum Free Energy Structure-Based Search Method for Riboswitch Identification Based on Inverse RNA Folding. PLoS One 10, 0134262.*

4.2. Introduction

Whereas some of the search methods for riboswitches include a certain degree of structural consideration, we believe that to find additional examples of eukaryotic riboswitches it is important to put a considerable emphasis on structure (without neglecting sequence conservation). This is because searching for a known consensus query pattern from bacteria in a eukaryotic database with sequence-based methods is from the evolutionary standpoint expected to be less fruitful than searching the same prokaryotic consensus in a database of prokaryotic organism. Even Infernal [51-53], which incorporates secondary structure information into the search model, uses sequence-based filters heavily and these may remove many of the candidates. Structure conservation should play a more dominant role in the delicate balance between sequence and structure when patterns from more distant organisms are aligned. A well-known structure identifier program called RNAMotif [62] is available but it is a descriptor language that cannot be easily modified for the purpose of riboswitch searches.

Motivated to address this challenge, we developed a method for eukaryotic riboswitch discovery that is highly structure based [76; 80; 127; 128]. This method utilizes

consecutive energy minimization predictions with a sliding window that each time compares the predicted structure of the query with the predicted structure of the sequence data inside the window segment. Because energy minimization predictions are computationally expensive, the method cannot be used on a genome scale. Another deficiency of this approach is that whereas folding prediction of the riboswitch aptamer (the query sequence) can be checked by comparing its structure to a biological experiment, scanning the sequenced data does not permit the folding prediction within each window segment to be checked for its accuracy. This limits its use to special cases of small sequenced data, such as when focusing on a certain metabolic pathway and genes that are associated with it. In such a case, the method can be employed with high resolution to look only at genes that play a role in that specific metabolic cycle (e.g., in purine metabolism, as demonstrated in [76]). We note that Infernal [51-53] can run with reduced sequence sensitivity partly by removing the pHMM filters that run prior to the actual covariance model. Running the software without the filters increases runtimes severely which also limits the availability of this method solely to small datasets.

The method described in this chapter was devised to remedy the shortcomings in the efficiency of existing energy-minimization structure-based search methods discussed above, also outlined and exemplified in [76]. Importantly, whreas all other methods scan the target sequence data using a moving window, repeatedly performing expensive energy minimization predictions, this step is completely circumvented in our method, allowing for an extensive genome-wide scan rather than being restricted to a single gene or pathway (e.g., purine metabolism as in [76]).

4.3. Design to Search Pipeline

The search method can be described as a pipeline that is separated into four sections: Seed preparation, Candidate design, Database search and Result filtering (Figure 4-1). The approach was influenced by synthetic biology in which sequences are designed to perform specific tasks: in our case the task of a ncRNA family, or more specifically purine riboswitch aptamer. The first two sections, seed preparation and candidate design, are responsible for designing multiple synthetic sequences that may act as purine riboswitch candidates while the last two sections, database search and results filtering, are responsible

for the identification of such candidates in nature. In each section multiple approaches were tested and will be discussed.

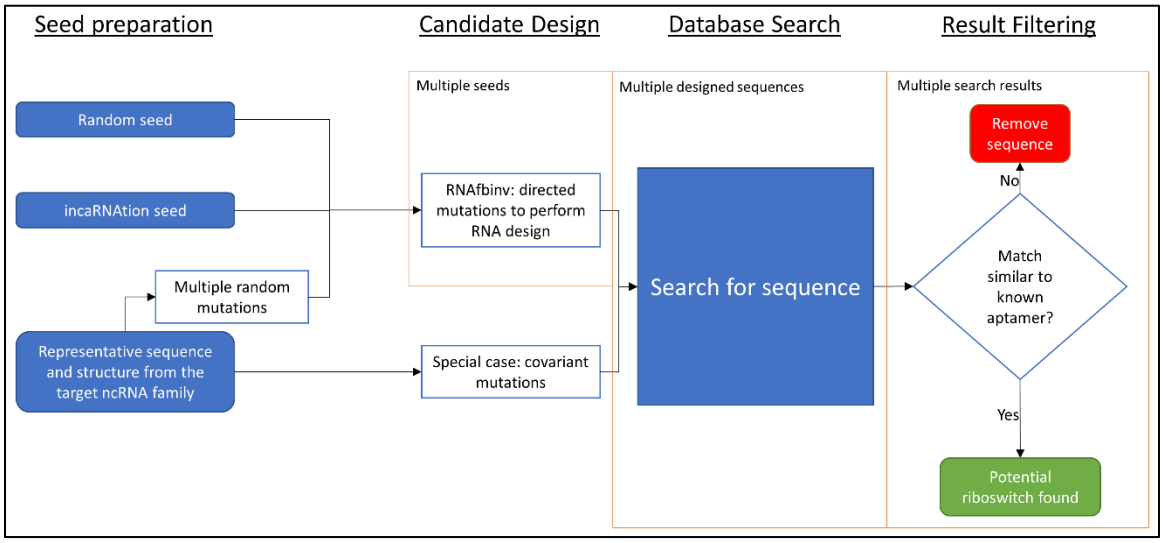

Figure 4-1: Design to Search Pipeline overview. Note that in some phases multiple products are generated and passed onto the next phase.

The design to search approach allows us to enjoy the efficiency of sequence-based searches while exploring sequences that are further away than those accepted using sequence similarity methods. By searching for sequences designed to preserve general shape with minimal sequence preservation we can explore a large sequence space. Our goal is to design many varied sequences so that the search process can explore as much of the sequence space as possible.

4.3.1. **Seed Generation**

Sequence design can be heavily influenced by the starting seed used in the process. As discussed in chapter 3, IncaRNAfbinv 2.0 uses an interactive approach starting from a seed sequence. Using the design on different seeds helps force varied design solutions.

Our initial attempts used a sequence from the riboswitch responsible for regulation of the *xpt-pbuX* operons in *B. subtilis* by terminating transcription in the presence of hypoxanthine or guanine [7]. Starting from the *xpt* riboswitch aptamer sequence we performed multiple single point mutations, after each mutation BLAST [8] was used to search the XPT sequence on the newly mutated sequence. For this step we chose to use the

option "Somewhat similar sequences (blastn)" that is available in the standard nucleotide BLAST utility at NCBI. Once the sequence was no longer discoverable, it was used as a seed for design (Figure 4-2).

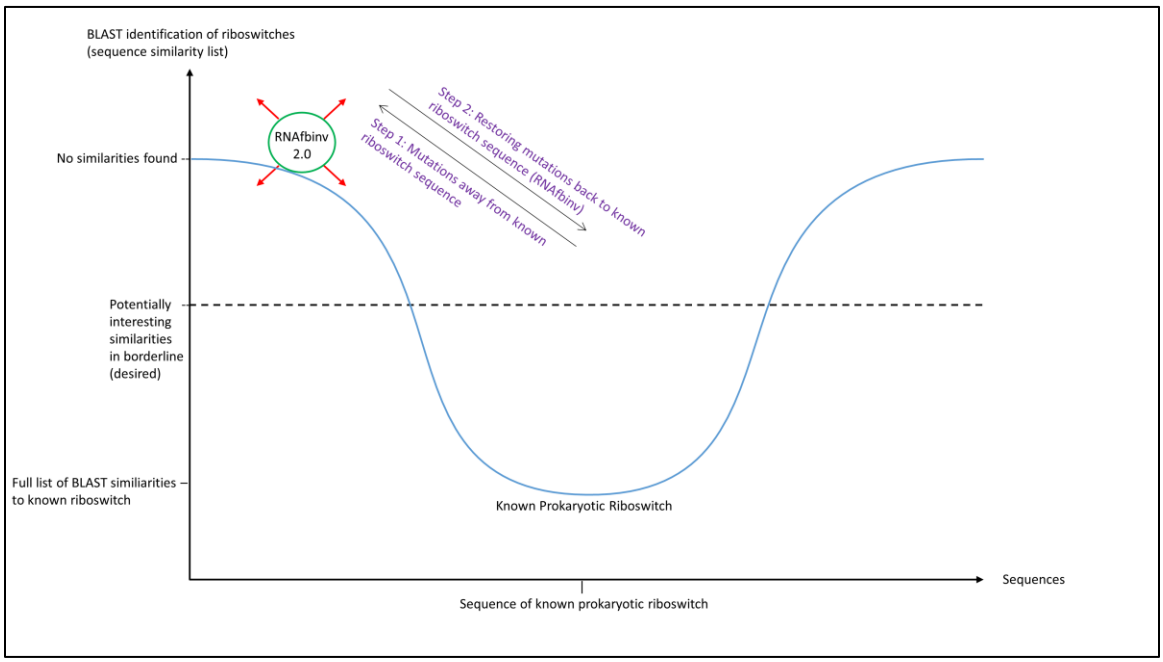

Figure 4-2: Schematic illustration of a potential well with a minimum in the known aptamer sequence. It is possible to escape the minimum by performing nucleotide mutations to the initial sequence until there are no BLAST hits when inserting the mutated sequence as input. If the mutations disrupt the known aptamer structure, the use of RNAfbinv will restore the known structure while generating designed sequences as output. Subsequently, designed sequences in the borderline of the potential well that do show BLAST hits should then be carefully examined in their hits. It is expected that most of the hits observed will be from known bacteria, but a few unknown bacteria and exceptional eukaryotic organisms will also show up.

The seeds created by the above method generated multiple results (discussed below) but we required a method that generates a larger variety of seed sequences. Fully randomized seed sequences are uniformly distributed. The sequence was chosen due to the high accuracy of structure prediction. Running the design step on many such sequences resulted in very different structures compared with the target. This is a result of the difficulty of random walk design starting from unstructured RNA sequences or heavily structured sequences that are stable but differ from the target. To resolve the issue, we

generated seed sequences using incaRNAtion [68]. IncaRNAtion uses a global sampling approach with weighted sampling techniques. It generates multiple sequences that fold near a given target secondary structure that are thermodynamically stable and well distributed. IncaRNAtion have been shown to improve the ability of RNAinverse [67] and RNAfbinv [129] to generate diverse sequences under different GC-content.

Figure 4-3 shows a comparison of the three seed methods. The sequence variability factor was calculated as the mean nucleic acid mismatch of any single sequence to all the other sequences in the group. We can see that random and incaRNAtion seeds and designed sequences have a higher variability factor as discussed above. It is interesting to note that the average design scores of mutated *xpt* seeds was lower than the random seeds. This can be attributed to specific sequences similar to the *xpt* aptamer that are in a local minimum that is difficult for the design program to escape.

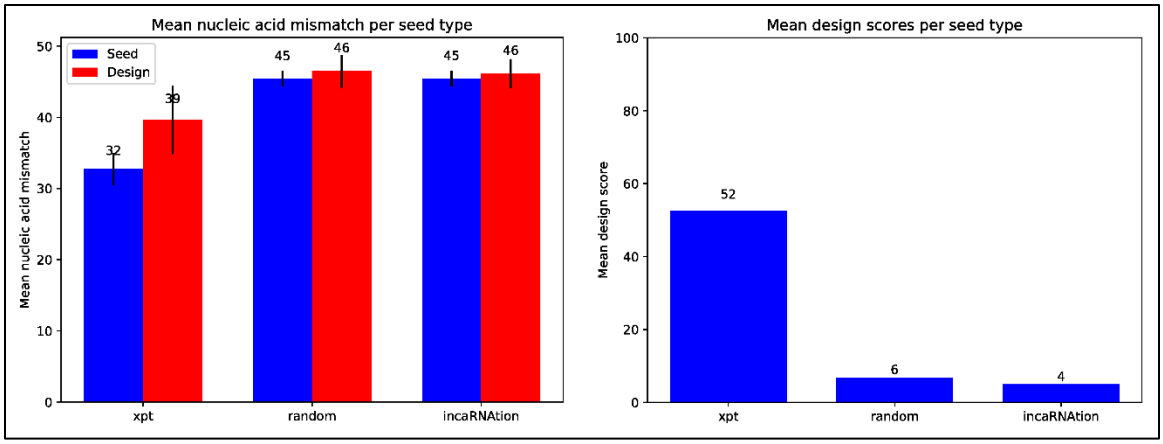

Figure 4-3: Variety and design scores for different seed types. Based on generation and design of purine riboswitch aptamer (n=400). On the left we can see that random and incaRNAtion seeds were more diverse compared with mutated *xpt* seeds. The effect continued to the design results that were based on those seeds. On the right, mean design scores. The incaRNAtion seed were slightly better. For more complex design problems the advantage of incaRNAtion increases.

4.3.2. **Candidate Design**

One option in sequence design can be performed using chosen covariant mutations (compensatory mutations performed on a base-pair, e.g. A-U to C-G, which are shape-preserving in line with the RNAfbinv approach). If an RNA inverse folding solver other

than RNAfbinv is used that strictly preserves the secondary structure, both random nucleotide mutations that are performed to escape the potential well in Fig 4-2 and the directed nucleotide mutations that are performed by the RNA inverse folding solver to restore the initial aptamer structure should be made to preserve structure. In this case, there is no distinction between random mutations to reach outside the well and directed mutations for structure restoration to get back inside, since all the mutations are covariant mutations that preserve structure. This special case which only covers a small portion of the sequence search space is depicted at the bottom side of Fig 4-1. It is fast but it is not capable of achieving the vast majority of possible designed sequence results. While there is evolutionary strength to such a method, it does not consider structural energy minimization and generates strict structures. The process includes two phases of mutations: inside bulges and loops, random single point mutations are introduces, whereas in stems, a mutation is introduced for both nucleotides to preserve two bases that can pair.

Another option is to design based on energy minimization. Using RNAfbinv 2.0, we take seed sequences and design them into the target shape of the ncRNA in question with additional constraints that define it. The RNAfbinv 2.0 design process optimizes a sequence to a general secondary structure shape based on minimum energy prediction generated by Vienna RNAfold [67]. It also allows the preservation of motif-specific constraints such as sequence patterns as discussed in chapter 3. Figure 4-4 shows the constraints used for the purine riboswitch design. Since the constraints are within a structural context, we can generate sequences of multiple lengths with different size distribution per motif as seen in nature. The design process output is a variety of different sequences that should act as synthetic purine riboswitch aptamer. Theoretically, if the folding prediction is correct, these sequences should act as purine binding aptamers. However, since folding prediction is inaccurate, finding similar sequences in nature is strong support for the activity of the designed sequences.

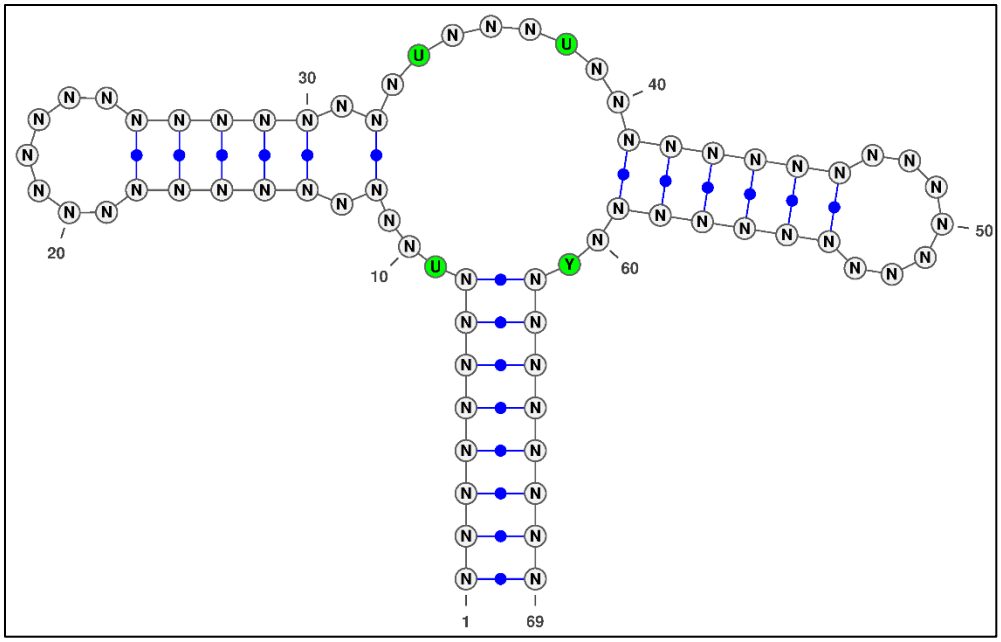

Figure 4-4: RNAfbinv 2.0 target for Purine Riboswitch aptamer design. Generated using VARNA[126]**.** Note that only four nucleic acids were targeted based on direct binding interaction.

4.3.3. **Database Search**

Our method targets searching over large genomic datasets like NCBI's nucleotide collection (nt) database. In order to work with such data sets, we require a fast and efficient method. Since we generate sequences, we can use highly effective sequence search methods such as BLAST [8]. We assume that as in the bacterial world, a single eukaryotic riboswitch sequence will lead to many other similar sequences proving such methods effective. Whereas BLAST is effective for sequence search it has no structural considerations. BLAST encourages results that match specific sections of a designed sequence with high accuracy. Whereas high sequence conservation can indicate highly preserved active motifs in proteins, for the problem of matching a riboswitch aptamer we require the existence of the general structure. This leads to many false positives generated by the search process.

To overcome this issue while still preserving speed, we use Infernal [52]. To use Infernal, we must generate a covariance model for the sequence we wish to search. We use CMbuild from the Infernal package to generate a single sequence model. To generate such a model, we require a multiple sequence alignment (MSA) with a single representative

structure. For our purpose, the MSA includes only the designed candidate sequence and the attached structure is the minimum energy structure predicted by RNAfold. Covarience models also require a calibration step that is responsible for assessing the E-value threshold required for a hit to be significant. This step is done using CMcalibrate on the default randomized database size of 1.6 MB.

Since the model is based on a single sequence, it has very strict sequence constraints (no variability). cmsearch allows us to search large databases very efficiently using this model. The initial pHMM filters rule out a large part of the database thanks to the highly constrained sequence in the model. Once the filter is done, the covarience model is applied to check for the attached structural information. This phase can rule out many false positives that only include sequence resemblance. Construction and calibration of such models can be time consuming, however, they are only done once per designed sequence and the calibrated model can be used on multiple genomic databases.

The search was performed on the nt database which is over 160 GB in size.

4.3.4. **Result Filtering**

The database search may result in multiple matching sequences for each designed candidate. Some of the matched sequences may lose the most fundamental constraints that are targeted by our design process. BLAST results may lose structural similarities even within the general shape of the purine riboswitch aptamer. Moreover, the ligand binding nucleic acids may be lost or moved away from their structural context. Even when structural considerations are applied, minimum energy prediction is not considered in both methods for the newly matched sequences. Even if all constraints hold, on a large genomic database some of the results may be random occurrences. Additional filtering steps are required to remove false positives.

Each search result is folded using RNAfold and then compared with the design target using the RNAfbinv 2.0 target function. The resulting score eliminates sequences that do not contain critical sequence constraints within their structural context. Since the structure is generated by RNAfold this also adds minimum energy considerations. Moreover, nearby genomic annotations can show a connection between a purine associated

molecule or process and a gene product which can increase confidence in the search result. Tagging multiple similar results for the same design sequence in multiple genomes can also hint at an evolutionary connection.

4.4. Results and Discussion

The results discussed in this section were matched by our described pipeline with either seed type or search method. Initial matched sequences were tested using an in-line probing assay I performed in Prof. Michal Shapira's laboratory at Ben Gurion University of the Negev.

4.4.1. In-line probing

In-line probing is a well established experimental assay used to establish the structure of RNA molecules as well their binding affinity to an external ligand. The protocol we applied is based on that used in R. Breaker's lab [130] to identify and test many of the initial riboswitches. The assay relies on the fact that an RNA strand at room temperature and in an RNase free environment tends to degrade more prevalently in unbound regions. The phosphodiester bonds in the RNA backbone undergo nonenzymatic cleavage through a nucleophilic attack of the 2' oxygen at the phosphorous center. An in-line conformation of the 2'oxygen, the phosphorous center and the 5' oxygen leads to displacement of the 5' oxygen effectively cleaving the connection. Bound regions of an RNA strand can sample less conformations because of the geometry restrictions enforced by a bound nucleic acid. This means that the speed of cleavage is greater in unbound regions.

The assay begins with the preparation of the desired RNA sequence and the attachment of a radioactive phosphate isotope (^{32}P) to the 5'. The labeled RNA is then divided into groups that are incubated in different solutions. Each solution is analyzed by denaturing polyacrylamide gel electrophoresis (PAGE) and visualized with a Phosphoimager. A highly basic solution (pH 9.0) is used to generate a single base resolution ladder and RNase T1 is used to create a ladder marking the location of G nucleic acids. A basic solution (pH ~8.3) is used to allow for slow degradation of the RNA based on in-line reactions with and without the target ligand. Comparing the lanes allows for the identification of secondary structure and binding affinity compared with the input

sequence. Before I started testing our new sequences, we calibrated the experiment on the well-known *xpt* riboswitch apatamer from *B. subtilis*. I also attempted the in-line probing assay using fluorescent labels based on [131] but because of low resolution images we returned to using the R. Breaker method [130].

Note: All the tasks below were performed using DEPC-treated DDW, single-use tips and tubes were RNase free and glass and metal tools were heat-sterilized.

RNA transcription and labeling

Forward and reverse DNA annealing primers were designed for each tested sequence (Table 4.1). The primers included a T7 promotor and the sequence of the tested aptamer candidate with a slight extension at both the 5' and 3' ends that appear not to affect transcribed RNA secondary structure under energy minimization prediction. The primers included an 18- to 22-base overlap for annealing purposes. Additional PCR primers were ordered for amplification of DNA strands. PCR was performed with all four primers using ThermoFisher Phusion DNA polymerase with an initial ligation step and 30 amplification cycles. Products were checked on a 2% agarose gel and recovered using a Zymoclean™ Gel DNA recovery kit

Target	**Primer (F/R)**	**Primer sequence (F/R) 5'→3'**
Xpt	Annealing (F)	ATAATTAATACGACTCACTATAGGGAATATAATAGGAA CACTCATATAAT**CGCGTGGATATGGCACGC**
Xpt	Annealing (R)	GTTCCATTGCTCACCCATAGTCGGACATTTACGGTGCC CGGTAGAAACTT**GCGTGCCATATCCACGCG**
Xpt	PCR (F)	ATAATTAATACGACTCACTATAGGG
Xpt	PCR (R)	GTTCCATTGCTCACCC
Pelosinus sp.	Annealing (F)	ATAATTAATACGACTCACTATAGATGTTACTACATT CTCGTATA**TTTTTGGGAATATGGCCCAAAA**
Pelosinus sp.	Annealing (R)	ATGCATTTTCACCCGTAGACGGCAATTCATGGTTGC CTGTAGAAAC**TTTTGGGCCATATTCCCAAAAA**
Pelosinus sp.	PCR (F)	ATAATTAATACGACTCACTATAGATG
Pelosinus sp.	PCR (R)	ATGCATTTTCACCCG
A. oryzae	Annealing (F)	ATAATTAATACGACTCACTATAGACAACGAGAGGCAA

		AACAACAT**CAACTGGCGCCATAGAAAGGTG**
A. oryzae	Annealing (R)	GTTGAAGAACTACATGAGGGATTCTTGATACTCCCGA ATGTTCTTA**CACCTTTCTATGGCGCCAGTTG**
A. oryzae	PCR (F)	ATAATTAATACGACTCACTATAGACA
A. oryzae	PCR (R)	GTTGAAGAACTACATGAGGG

Table 4-1: Primers used for aptamer transcription. Red, short tail for T7 RNA polymerase; Blue, T7 RNA polymerase promotor; Green, sequence matched by the search method as the aptamer candidate. The sections marked in bold text mark the overlapping region between forward and reverse annealing primers.

1 µg of DNA was transcribed each time for two hours using AmpliCap-Max T7 RNA polymerase with addition of NTPs and ScriptGuard RNase Inhibitor to prevent RNase contamination. RNA transcription products were purified using an RNA Clean & Concentrator kit (Zymo Research). 10 pmol of purified RNA products were de-phosphorylated with Calf Intestinal Alkaline Phosphatase (CIP) (New England BioLabs) and then phosphorylated with [γ-^{32}P]-ATP using T4 Polynucleotide Kinase (ibid). Labeled product was diluted into batches of 300,000 cpm/µl. To remove labeling reagents products were cleaned using the above RNA Clean & Concentrator kit. The entire process was usually done for each experimental batch, but in some cases the labeled products were preserved at -80° C for a few days prior to analysis.

In-line probing assay

Table 4.2 lists solutions for each of the reactions of the assay. The In-line probing reactions with and without ligand (Guanine, Adenine, Hypoxanthine and 2'-Deoxyguanosine) were incubated for 72 hrs. The single base pair resolution ladder was incubated for 5 to 15 mins at 90°C whereas the G ladder based on T1 RNase was incubated for 5 to 15 mins at 55°C. An additional solution with only RNA and loading buffer was added to reduce background noise of untreated RNA. The 40 cm long 0.75 mm thick 8% Acrylamide gel made with a 40% Acrylamide\Bis-Acrylamide 29:1 solution was preheated by running it at 45 W for 30-60 mins. RNA reactions were loaded and run at 45 W for ~3 hrs. The gel was removed from its casing, dried and left overnight in a Phosphoimager cassette in the -80°C freezer.

Reagent	Materials
Loading buffer (x2)	1.5 mM EDTA, 10 M Urea
Na_2CO_3 buffer (x10)	0.5 M Na_2CO_3 (pH 9.0), 10 mM EDTA
In-line reaction buffer (x2)	100 mM Tris-HCL (pH 8.3), 40 mM $MgCl_2$, 200 mM KCl
Sodium citrate buffer (x10)	Sodium citrate 0.25M (pH 5.0)
NR – Untreated RNA	10 µl loading buffer x2, 9 µl DEPC DDW, 1 µl labeled RNA
T1 – G ladder	7 µl loading buffer x2, 1 µl Sodium citrate buffer x10, 1 µl labeled RNA, 1 µl RNase T1, 3 µl loading buffer x2*, 7 µl DEPC DDW*
-OH – partial alkaline digestion	1 µl Na_2CO_3 buffer x10, 8 µl DEPC DDW, 1 µl labeled RNA, 10 µl loading buffer x2*
In-line reaction	5 µl in-line reaction buffer x2, 3 µl DEPC DDW, 1 µl ligand solution (concentration depends on experiment), 1 µl labeled RNA,10 µl loading buffer x2*

Table 4-2: List of buffers and reaction solutions used in the assay. reagents marked with a star were added just before loading the gel. pH values were measured in room temperature (~23°C).

xpt aptamer calibration results

We ran the assay on the *xpt* riboswitch aptamer with two ligands (Guanine and Hypoxanthine) at two different concentrations (1 nM and 1 µM). The X-ray film was removed from the Phosphoimager cassette and scanned. The results are not quite as well resolved as those of the Breaker lab, however, they do show the ligand reaction as designed (Figures 4-5).

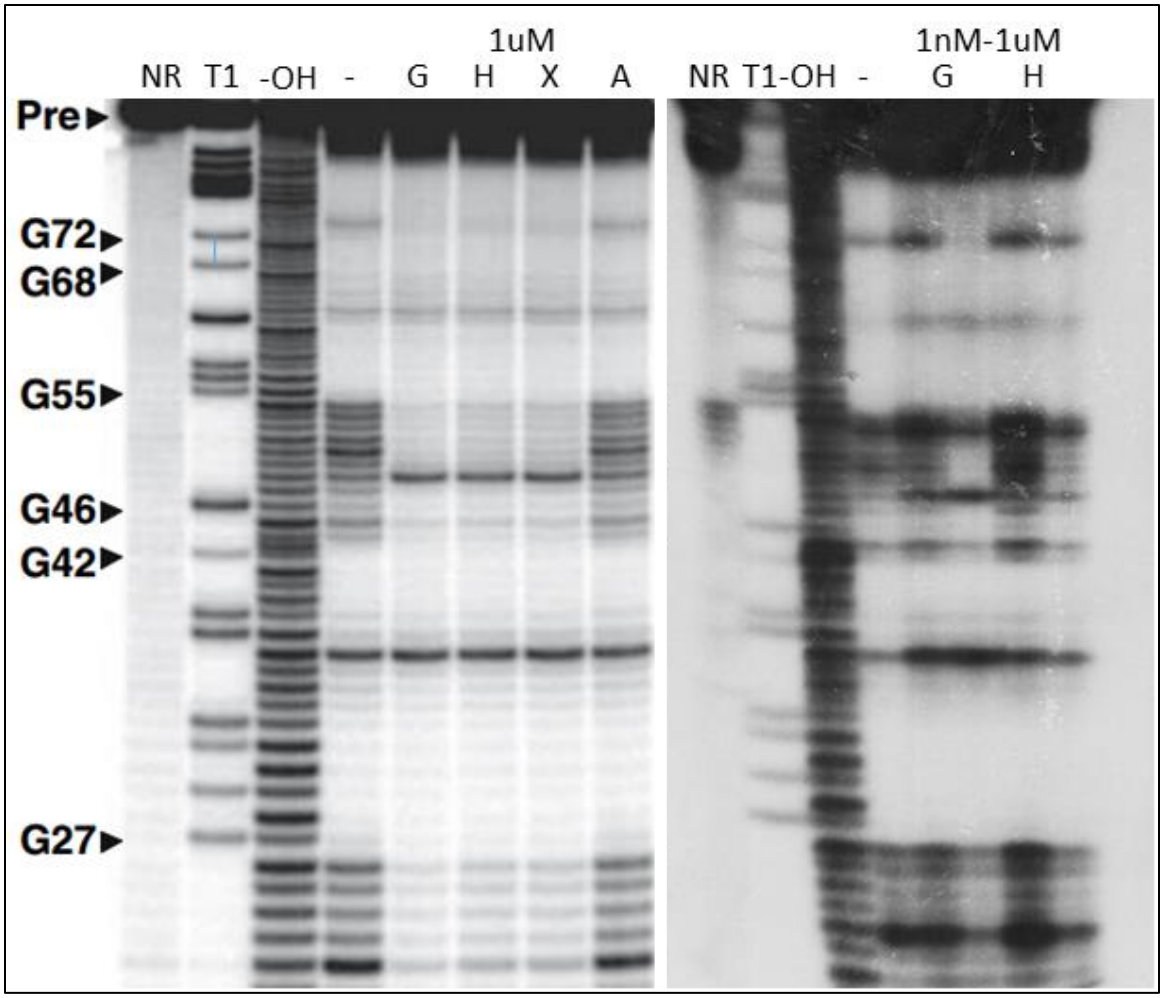

Figure 4-5: Comparison of in-line probing assay results between Breaker's lab taken from [143] on the left and my gel on the right. Although the resolution obtained in my experiments does not match that of the Breaker lab, we clearly observe binding reactivity for 1 μM of Guanine and Hypoxthanthine. NR – no reaction; T1 – G ladder; -OH – single base ladder; ─ – in-line reaction without ligand; G – Guanine; H – Hypoxthantine; X- Xthanthine; A – Adenine.

4.4.2. **Prokaryote Riboswitches**

Our purpose in identifying existing prokaryote riboswitches was to establish that the method works. Our method is designed to enhance the capabilities of existing methods and to identify new occurrences of the purine riboswitch. Therefore in the first instance it is important to demonstrate that we can trace back many of the known riboswitches. Tracing the outputs from seed generation to BLAST results starting from a mutated *xpt* sequence identifies multiple known prokaryote riboswitches (Figures 4-6). The high sequence

similarity of prokaryotic purine riboswitch aptamers means that designing one seed sequence close to the original sequence space results in the matching of many of those sequences.

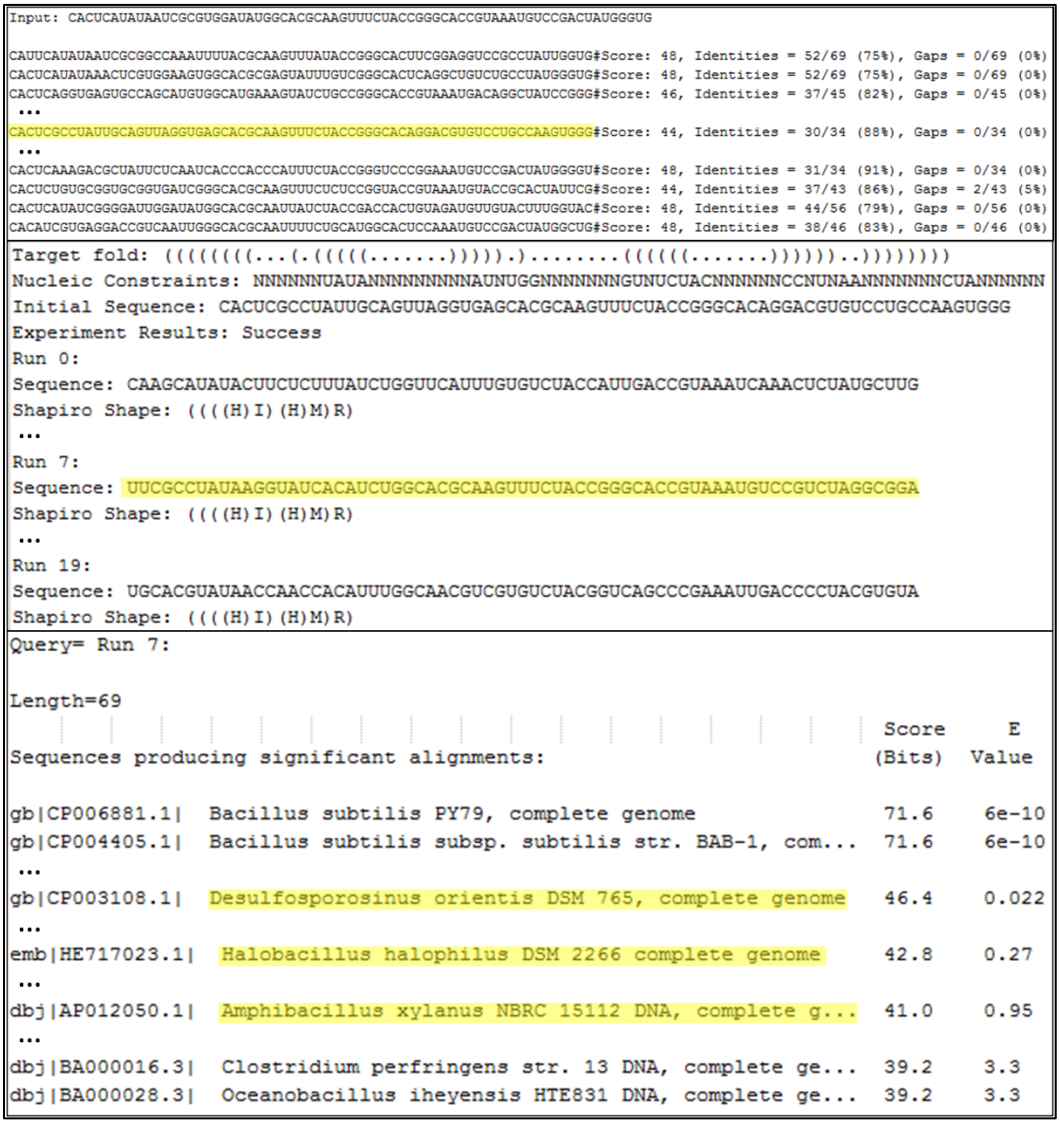

Figure 4-6: Tracing the search process from mutated *xpt* seed through design to matched BLAST results. Tracing highlighted run to find multiple known aptamers. Top pane shows the output screen for the multiple random mutation phase resulting in a seed sequence. Middle pane shows the output of multiple RNAfbinv runs. Bottom pane shows the output of nucleotide BLAST for designed sequence from run 7 (highlighted). Top BLAST matches include the *xpt* aptamer from *B. subtilis* yet further down we note multiple known riboswitches from different genomes (highlighted).

Search results for design batches generated from incaRNAtion seeds also match many existing prokaryote riboswitches. In general, we see that around 15% of search results are identified by Infernal using the purine riboswitch covariance model available in Rfam. Even though many of the bacterial purine riboswitch sequences are very similar, the single sequence covariance models generated by our search methods are much more restrictive and can only detect a small subset of those detected by the Rfam model. Designing multiple sequences and constructing multiple single sequence covariance models allows us to identify a large number of existing riboswitch aptamer.

In addition to detecting riboswitches that exist in the Rfam database we identified a novel prokaryote candidate (not included in Rfam at the PLOS ONE publication date). This candidate appeared in *Pelosinus* sp. strain UFO1. We could not identify this candidate using the special case of covariance mutations as it does not align accurately with the original *xpt* aptamer. The candidate appeared in the 5' UTR of a gene annotated as a Xanthine phosphoribosyltransferase similar to the annotation found in the *xpt* riboswitch. While performing in-line probing on the sequence we did discover binding to 1 μM of Guanine (Figures 4-7). Due to the ability of existing methods to detect the sequence and the fact that it appears in a prokaryote we made no further attempts to improve the resolution.

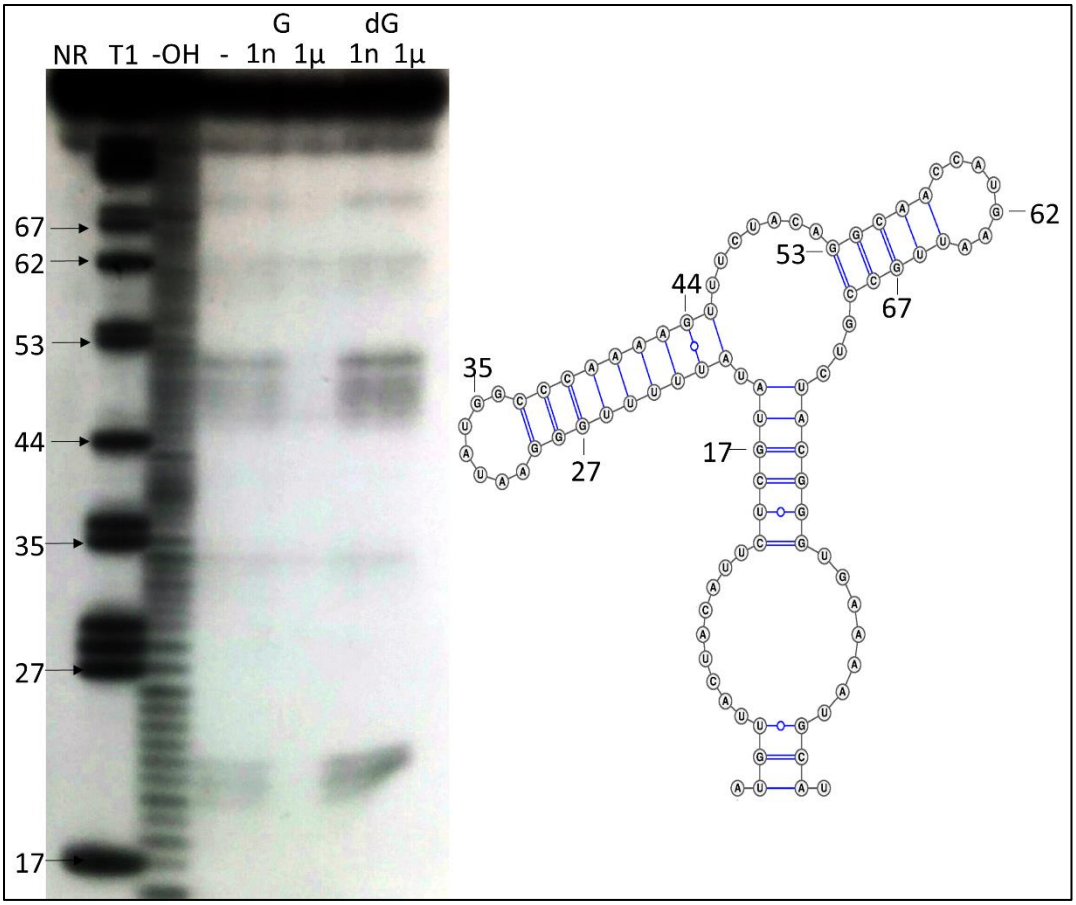

Figure 4-7: *Pelosinus* sp. UFO1 riboswitch candidate in-line probing results. The assay was performed with guanine and 2'-deoxyguanosine using a concentration of 1 nM and 1 µM of the ligand. Although the resolution is low, we do observe band reduction around the multi-loop and the two hairpin loops only in the 1 µM guanine lane.

4.4.3. **Eukaryote Riboswitches**

The main purpose of our method is to identify new eukaryotic candidates for the purine riboswitch aptamer. We identified multiple sequences that matched the sequence-structure model we defined and we present a selected few that were reinforced by additional indications such as evolutionary information or genomic annotations.

Candidates were matched in a variety of taxa (Figures 4-8). A candidate from *Callorhinchus milii* (Australian ghostshark) was found 146 bases after the end of the marked coding sequence (presumed to be the 3'UTR) of a Niban-like protein. It contains a U in the Watson-Crick binding positions indicating adenine binding and includes the ability

for a CC-GG pseudo-knot between the hairpin loops. A candidate from the coelacanth, *Latimeria chalumnae*, was found 14 bases before the start of a coding region (presumed to be the 5'UTR) of a lamin tail domain-containing protein. It contains a C in the Watson-Crick binding positions indicating guanine binding and includes the ability for a CUU-GAA pseudo-knot between the hairpin loops. Another candidate in the African cichlid fish, *Haplochromis burtoni*, was found 623 bases before the start of the coding region (presumed to be the 5'UTR) of an actin binding protein. It contains a U in the Watson-Crick binding position indicating adenine binding and includes the ability for an AA-UU or an AG-UC pseudo-knot between the hairpin loops. Another result was found near the ATP binding protein *YALI0A18590p* from the fungus *Yarrowia lipolytica CLIB122*. It includes a U in the Watson-Crick binding position possibly targeting adenine with the ability for an AAA-UUU pseudo-knot between the hairpin loops. Two results were matched in the microalgae *Cymbopleura sp. TN-2014* and *Conticribra weissflogii* near or within a photosystem I P700 apoprotein gene. Both candidates contain a U at the Watson-Crick binding position and include some possibility for a pseudo-knot.

These novel candidates were not tested using in-line probing. Each has its merits and supporting evidence and may function as a purine sensing aptamer. It is possible that there are single occurrences of purine sensing aptamers in nature, but it is more likely that there exists an evolutionary connection between multiple species. These single match results do not include such information.

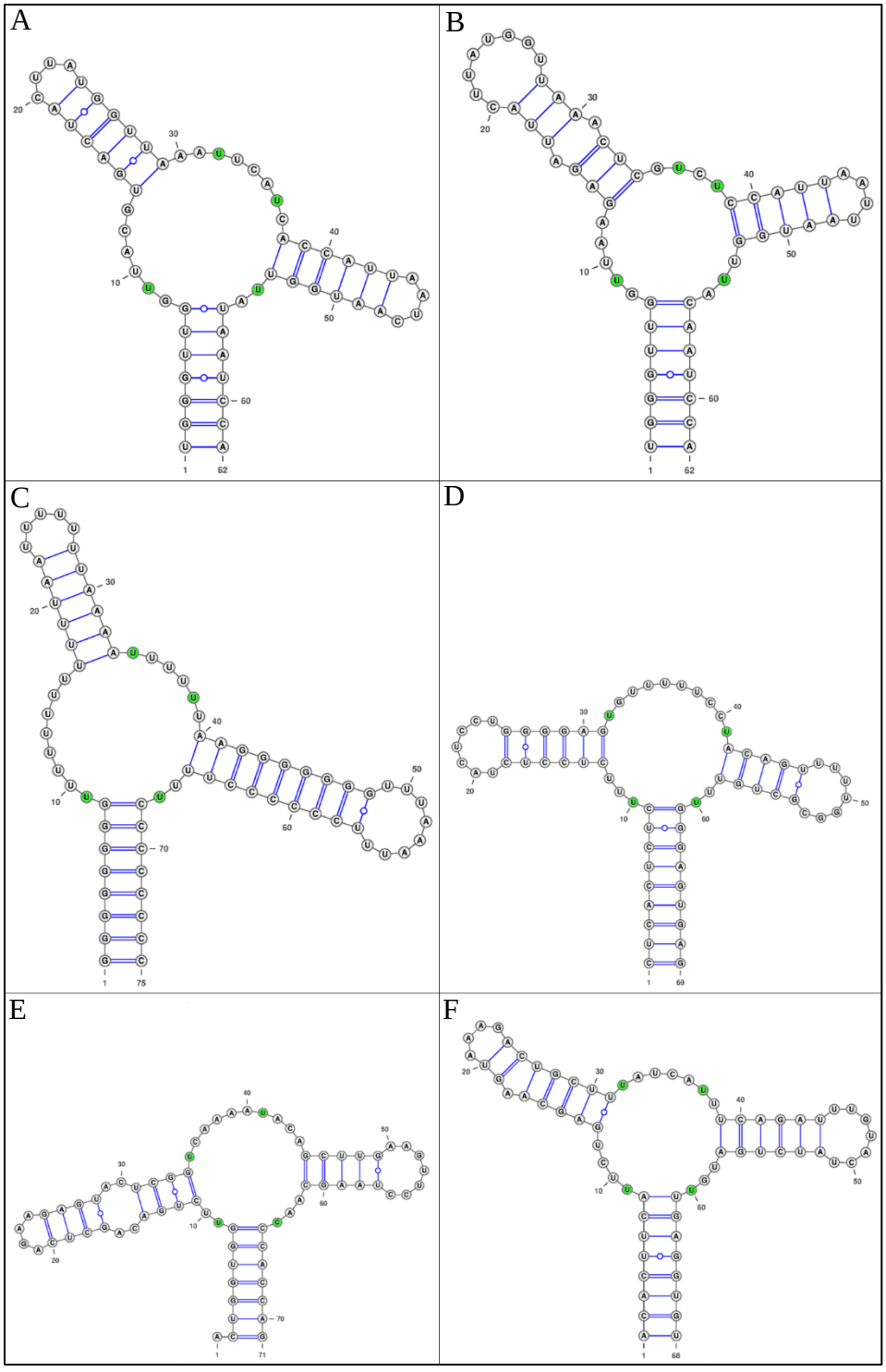

Figure 4-8: Select eukaryote candidates. A. *Conticribra weissflogii* (microalgae) and **B.** *Cymbopleura sp. TN-2014* (microalgae), **C.** *Yarrowia lipolytica CLIB122* (fungus), **D.** *Callorhinchus milii* (Australian ghostshark), **E.** *Latimeria chalumnae* (coelacanth), **F.** *Haplochromis burtoni* (African cichlid fish). Putative ligand-binding nts are marked in green.

Another dominant candidate was matched in *A. oryzae*, a fungus that includes a validated TPP riboswitch [132]. It was matched using a pipeline starting from a mutated *xpt* seed sequence that matched the general shape and ligand binding nucleic acids within their structural context. We observed a clear similarity between the multi-loops of our candidate and the *Mesoplasma florum* 2'-deoxyguanosine riboswitch (Figures 4-9). The sequence was matched to the 3'UTR of a gene marked as encoding a hypothetical protein that resembles a synaptic vesicle transporter. There is no direct link between synaptic vesicle transport and guanine production or transport, however, guanine derivatives have been shown to modulate glutamate transporter activity by decreasing glutamate uptake into rat brain synaptic vesicles [133]. Even though we found multiple positive computational indications, the in-line probing assay showed that the actual secondary structure did not match the predicted minimum energy structure.

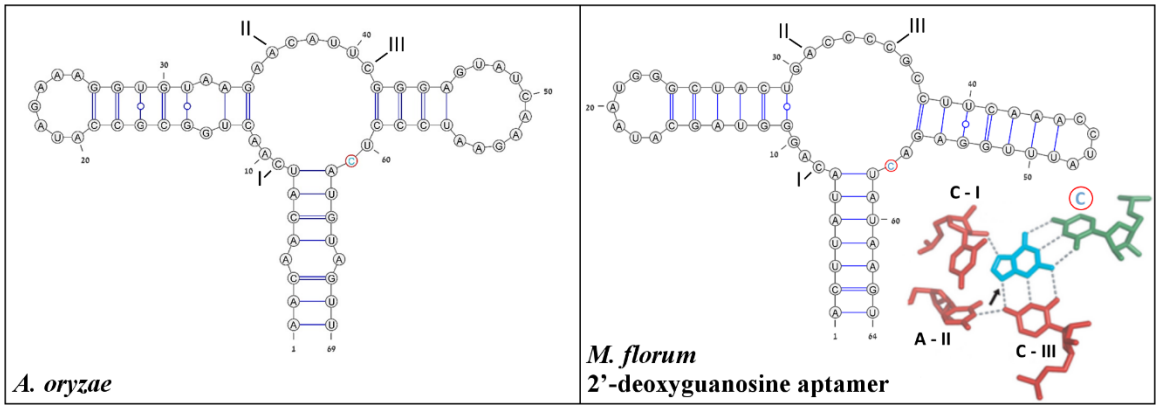

Figure 4-9: *A. oryzae* candidate predicted minimum energy structure compared with the *M. florum* 2'-deoxyguanosine riboswitch [75]. The four nucleic acids that directly bind to guanine appear in the correct structural context (marked by roman numerals). However, the predicted minimum energy structure did not match the structure that appeared in my in-line probing assay.

4.5. Conclusions

Finding new occurrences of existing riboswitch families requires the relaxation of current models. Using covariance models, we can easily detect candidates with high sequence similarity even in large genome databases. However, when relaxing such models to reduce sequence similarity we decrease the performance by multiple factors, making them unsuitable for large genome databases. Moreover, many false positives are obtained when

looking at structure alone since covariance models do not test the energy efficiency of the matched structures.

In contrast, the energy minimization method we present here allows us to relax the current models by designing multiple sequences that match our target purine riboswitch aptamer. This method has the following advantages:

1. The designed sequences preserve nucleic acids only at critical locations where the ligand interacts with the aptamer while maintaining its general shape.
2. By searching for the designed sequences we reduce the search problem to a sequence-based search making it efficient to run on large genome databases.
3. We focus on specific sequences that are predicted to fold to the desired structure and are thus thermodynamically stable to reduce false positives.

Our method was able to detect many of the existing riboswitches that are currently available in the Rfam database while identifying new candidates in both eukaryotes and prokaryotes. Testing our new prokaryote candidate shows a working riboswitch aptamer which binds to 1 μM of guanine. However, the in-line probing assay on our *A. oryzae* candidate was unsuccessful. The sequence did not fold into the predicted minimum energy structure. This shows that even a compatible predicted structure and an interesting annotation downstream is not enough as a filtering step. We require better support to ensure that the matches are not random. Given a large enough sequence database the probability increases that any sequence can be found. By finding a very similar sequence in multiple organisms which includes variation in regions that are less critical for ligand interaction and structure stabilization we can increase the confidence of having identified a genuine candidate riboswitch. This solution will be presented in the next chapter.

5. Eukaryote Riboswitch Candidates Supported by Covariance Models

5.1. Preface

Chapter 4 describes a pipeline for the identification of a structure dependent ncRNA. It focuses on high throughput scans that return single sequence results. The main limitation of this approach is that on a large enough genome database any sequence may exist. This leads to many possible false positives. In this chapter I present a method to address these problems by expanding single sequence results into RNA families. In each RNA family structural covariations and phylogenetic relations that indicate an evolutionary connection are highlighted. This RNA family search engine is particularly useful for genomes in which gene annotations are sparse or non-existing. Moreover, it can also be used to filter out false positives in the form of single randomly matched sequences.

The described pipeline as well as the resulting families have not been published or tested experimentally. Supplementary data includes matched locations, designed sequences and covariance model files. The data is available via a public git repository at: https://github.com/matandro/RiboSearch under the **supplementary** folder. The folder also includes a reference to the supplementary data in the form of a README file.

5.2. Introduction

The search for new occurrences of existing riboswitch families is dominated by covariance model scans using the Infernal package [51-53]. The models in use are taken from the Rfam database [3-6]. Rfam models are based on a group of sequences, called the model seed, chosen to represent the covariation of a given ncRNA family. Although the model seed attempts to display the variation in these sequences it is still based on previously identified sequences. In some cases, such as the purine riboswitch, where all occurrences belong to bacterial candidates, sequence conservation is very high and this leads to a restricted search. Rfam RNA family models were expanded in methods that tried to relax the model, but these do not solve the problem of high sequence conservation.

Model expansion is a process done on newly found RNA families and proteins [134]. A single sequence search for each of the seed sequences can be used to try and match new sequences. The resulting matches can then be aligned back to the original covariance model given enough computational evidence. Another method is to use an unfiltered covariance model search. A standard covariance search done by Infernal cmsearch application includes an initial filtering phase. In that phase multiple sequence based pHMM scans are used to reduce the input target sequence into areas of interest. Removing these filters can be very time consuming for a single average eukaryote genome. Even an unfiltered covariance model includes sequence information but it has a reduced effect on the significance of hits.

To resolve these issues, we developed a novel method based on energy minimization that attempts to find sequences that match the shape of the target structure with minimal sequence conservation based on key nucleic acids such as those that interact directly with the ligand. Although the method detects many new candidates, the problem of eliminating false positives remains difficult. The filtering process we developed takes into consideration reinforcing factors such as existing genomic annotation. However, these considerations present multiple issues: many genomes include only limited annotations in respect to amount or labeling. Even if annotations do exist, since we do not know which processes may be affected by eukaryote riboswitches, it is hard to distinguish relevant from irrelevant annotations. Moreover, as illustrated by the *Aspergillus oryzae* candidate (chapter 4), even when genomic annotations exist and the organism contains the only riboswitch family identified in eukaryotes, minimum energy predictions may still be wrong, resulting in false positives.

Therefore, to support our matched candidates, we expand single sequence results into a covariance model that represents similar sequences that preserve structure and key nucleic acids. By expanding the model iteratively, the model grows beyond the initial search results while maintaining the heavy sequence conservation that allows this search to run efficiently. Once a model is generated, the resulting sequences can be tested as a group to recover similar annotations and evolutionary connections. This allows us to cover a wider area of the sequence space and to identify clusters of similar sequences that may

behave as does the target ncRNA (figure 5-1). Unlike existing model expansion methods [134], the method we present here focuses on structure and biological function and not solely on alignments.

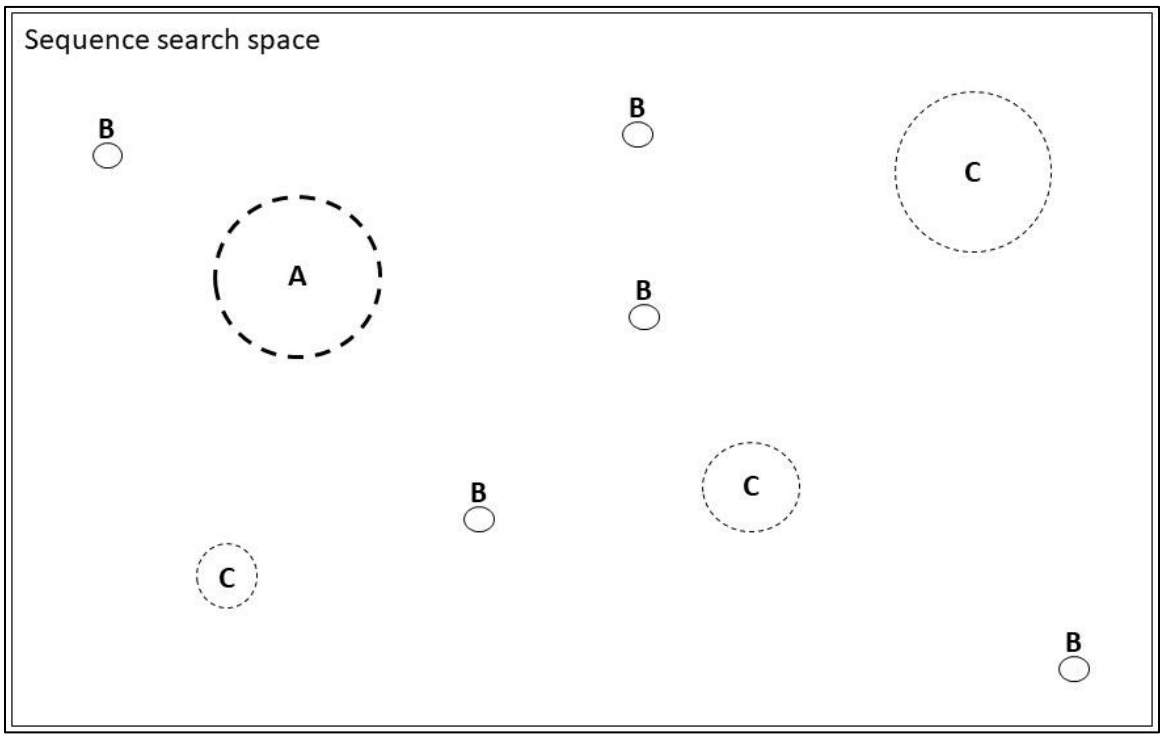

Figure 5-1: Conceptualization of the sequence expansion method. Our proposed method samples the sequence space by matching sequences that share the general shape of the purine riboswitch. The dotted circle with A represents sequences matched in bacteria identified by the Rfam model. The circles with B show single sequence matches using our energy minimization method. The dotted circles with C are single sequences expanded to contain multiple similar sequences with structure-preserving covariation.

5.3. Iterative Covariance Model Expansion

Model expansion is done iteratively to increase the variability of existing results and generalize the starting model while preserving key features (figure 5-2). The initial model is based on the designed ncRNA sequence and a predicted minimum energy structure as the model's consensus structure (see section 4.3.3). The single sequence model is searched against large genomic databases using Infernal cmsearch. Each matching sequence is aligned to the consensus secondary structure in the model. The sequences and their aligned

structures can then be compared with the target sequence-structure using the RNAfbinv 2.0 alignment function (chapter 3) to ensure that the secondary structure shape has not been lost and that the sequence constraints are still present within their structural context.

Using the RNAfbinv 2.0 alignment function as a filter allows us to ignore the E-value threshold set by the model calibration that may be high because of lower sequence similarity. Sequences with an alignment score lower than a set threshold are considered relevant to the new model and are kept in the alignment. This alignment is then sent to cmbuild to generate a new covariance model. The new model is an expansion of the model we started with and can be used in the next iteration. The new model should allow for higher sequence variability while maintaining the initial secondary structure.

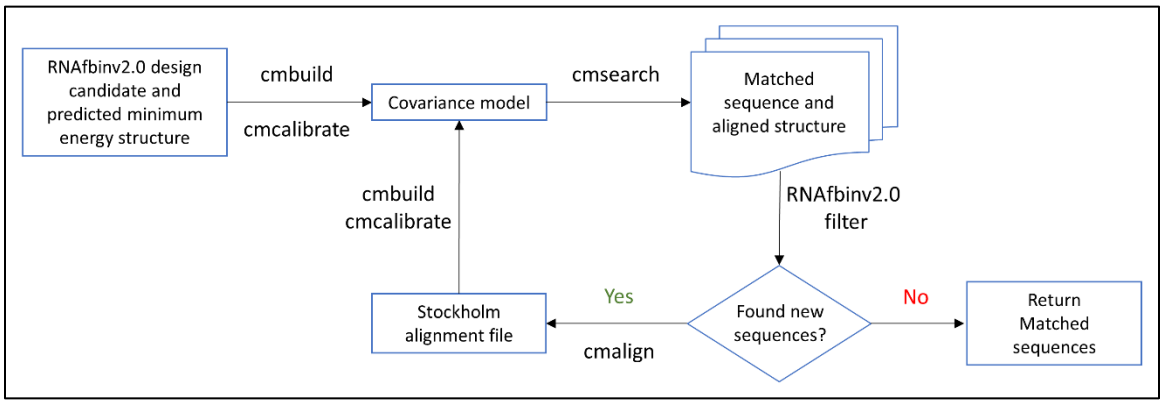

Figure 5-2: Overview of model expansion pipeline. Generating new covariance models based on search results as long as new sequences are matched. The RNAfbinv 2.0 alignment function allows us to expand to include results with higher sequence variability if they maintain key sequence-structure features. Once the model is more relaxed, new results can be found.

The expansion is slow since the starting model is based on a single sequence. Accepting results with higher E-value means that some will have somewhat lower sequence similarity. The newly generated models are still very restricted in respect to sequence. This preserves the efficiency of profile-hmm filters used by cmsearch, making the search process fast. Since we generate multiple models, we can cover more of the sequence search space with small RNA families.

The covariance in the models is somewhat forced since unlike models based on pure sequence alignment this model is based on a sequence alignment generated from a

previous covariance model. This is an artifact of the high throughput search approach. Unlike comparative genomic methods that select sequences based on an annotation and compare them, in out method sequences are compared after matching a covariance model using a search.

5.4. Results and Discussion

Analysis of the results is focused on covariation of the sequence-structure in the model visualized by the R2R [135] as well as characteristics of the sequences used to assemble the model. Our target search is the purine riboswitch aptamer. Below we compare the sequence-structure model of each resulting RNA family to the purine riboswitch model found in Rfam.

5.4.1. The Rfam Purine Riboswitch Model

The purine riboswitch model available in Rfam shows high sequence conservation in the multi-loop and on the top part of the hairpin loops (figure 5-3). The hairpins are known to create a pseudo-knot starting from the GG-CC pairs and continuing with an interaction of 4 to 5 bases. The multi-loop itself was shown to contain internal pseudo-knots that help to create a binding pocket. The Rfam database contains 2,703 sequences that match the purine riboswitch, and even when creating a model from all of them we still observe high sequence conservation. Even highly conserved nucleic acids, such as the AU at the bottom left of the multi-loop, can be seen to be changed in some working purine riboswitches. The *M. florum* 2'-deoxyguanosine sensing riboswitch contains an AC at that same position [99].

Rfam version 13.0 scans the standard (STD) and whole genome shotgun (WGS) sequence sets obtained from the European nucleotide archive (ENA) which comprises over 2.2 TB of data. ENA also annotates a non-redundant database based on sequence and taxonomic spread of species [6]. Earlier versions had over 10,000 purine riboswitch sequences.

A comparative genomic analysis of purine riboswitches shows that most occurrences appear near a purine transporter, salvage pathway or *de novo* synthesis of purine derivatives [136].

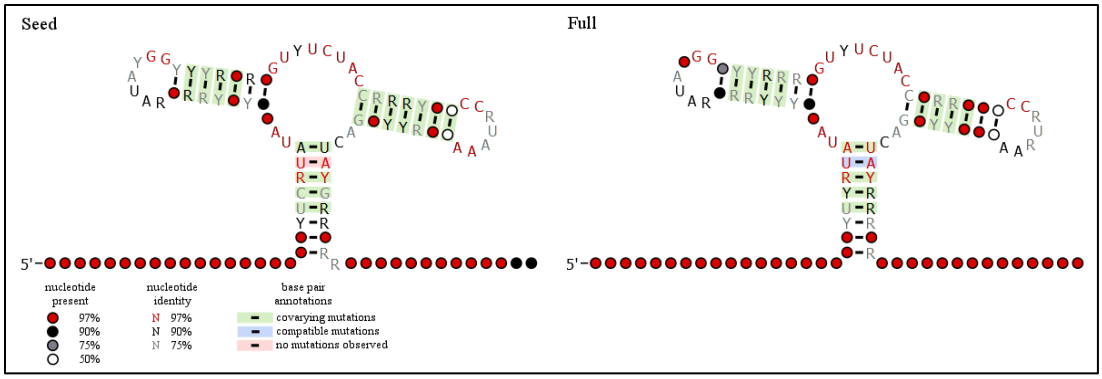

Figure 5-3: Consensus sequence-structure of the purine riboswitch family from Rfam [3-6]. Left, the consensus using seeds sequences (133) chosen by Rfam; Right the consensus using the full list of sequences (2,702). The model generated from the full alignment shows only a slight relaxation compared with the model generated from the seed.

5.4.2. **Reconstructing the Rfam purine riboswitch family**

To date, purine riboswitches have only been detected in prokaryotes. Our aim is to identify novel purine riboswitch candidates in eukaryotes. However, an important step in the validation of the search method is to reconstruct the existing purine riboswitch family as it is defined in the Rfam database. Our design process started with a seed generated by incaRNAtion [68] and then redesigned using RNAfbinv 2.0 (chapter 3). The design target included minimal sequence constraints in the form of the four ligand-binding nucleic acids within the multi-loop. We subsequently constructed a covariance model based on the designed sequence and a predicted minimal energy structure. Within three search and expansion iterations we reached a target model that did not recover any further sequences. The resulting model is almost a perfect match to the Rfam model (Figure 5-4).

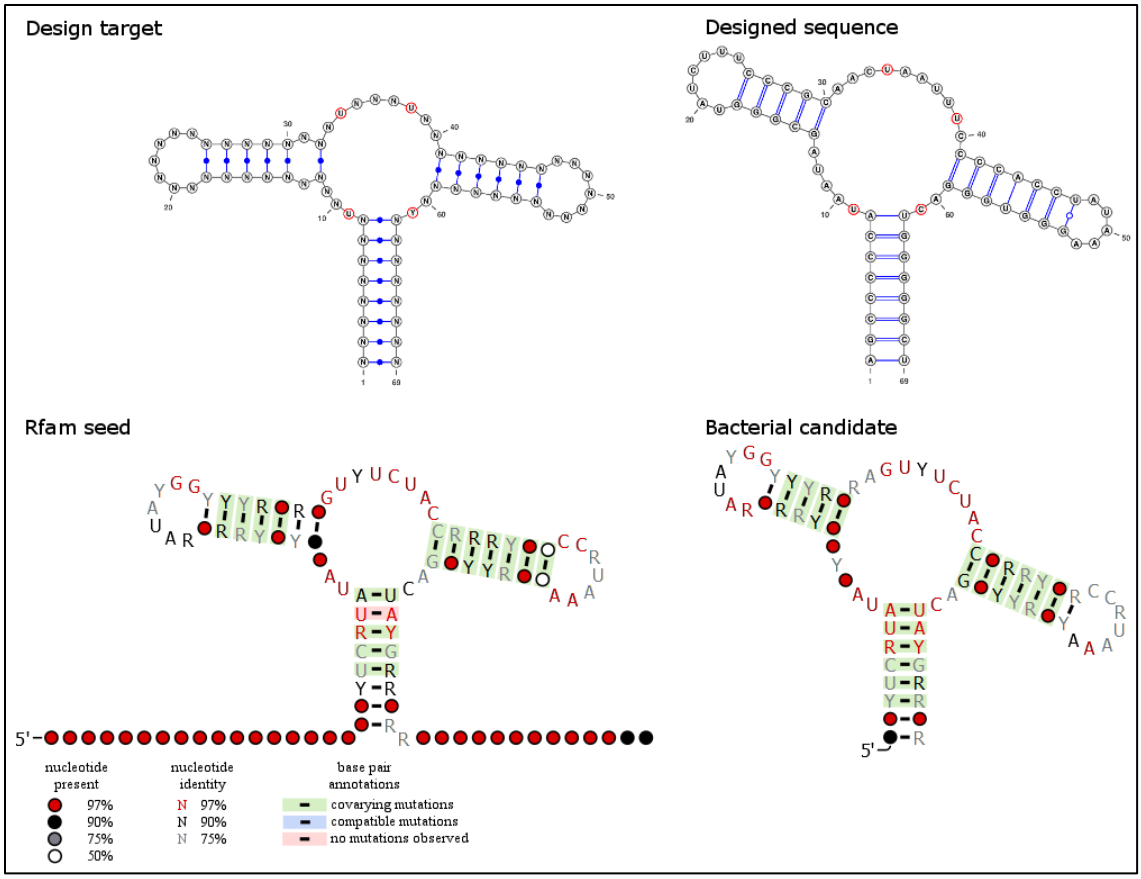

Figure 5-4: Comparison between Rfam purine riboswitch family and our generated family. The design target shows the minimal sequence constraints used in the design process (red circles). The designed sequence includes the relevant nucleic acids within their structural context. Comparing the two families shows high conservation in the stems as we would expect.

Our model is based on 3,576 sequences, all of which were matched using the existing Rfam covariance model. The sequences had a low E-value (largest at 5.10 E-10) and a score above the trusted cutoff defined for the Rfam model (37.0). Out of the 2,703 sequences in the full Rfam listing, 466 overlap with our sequences. The disparity in number is attributed to the larger database scanned and the non-redundant nature of the Rfam database as discussed above. Our model can also be used to find all 2,703 sequences in the Rfam database with a low E-value.

Comparing the models shows preservation of all but one of the covariance mutations in the stems. Core sequence is mostly preserved, mainly in the multi-loop. The

CC on the right hairpin appears in only 75% of the sequences in our model. This might be an indication of mismatched candidates or a weaker but still viable $_{pseudo}$-loop connection.

5.4.3. **Eukaryote transketolase candidate**

Transketolase is an enzyme that interacts with Ribose-5-phosphate (R5P) as a substrate such that Ribose-5-phosphate + Ribulose-5-phosphate are transformed into Heptulose-7-phosphate + Glyceraldehyde-3-phosphate [137]. R5P is also the substrate for the Ribose-phosphate pyrophosphokinase enzyme that transforms it into Phosphoribosyl pyrophosphate (PRPP). PRPP is known to start the de-novo purine biosynthesis pathway [138]. We hypothesize that when the organism lacks purines, it may need to regulate the amount of transketolase produced so that R5P can be used to generate PRPP and thus increase purine biosynthesis.

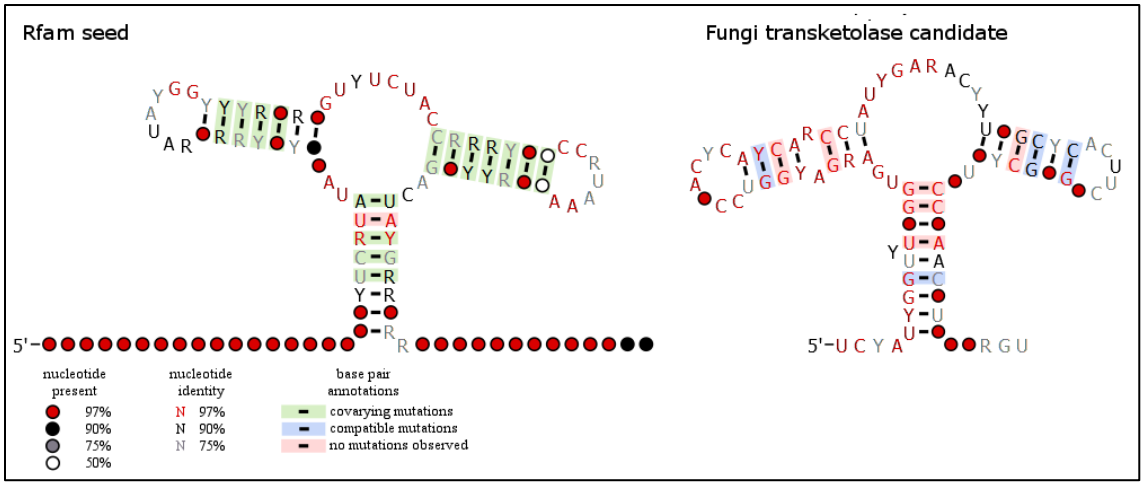

Figure 5-5: Transketolase candidate model compared with Rfam model. Structural preservation of sequences can be found mostly in the left stem, bottom stem and left side of the multi-loop. Compatible mutations were found but not covarying mutations.

All the 59 sequences used in the model contain a U at the Watson-Crick base paired position that targets the binding to Adenine. The alignment (figure 5-5) shows U only in 75%-90% of the sequences, but when U is not at that position it appears in the red circle below it. The stems include conserved sequences as well as some compatible G-C/U.

Whereas transketolase is present in a wide variety of organisms from bacteria to mammals, our candidate was matched in fungi, specifically the Ascomycota division

(figure 5-6). The fungi annotations are computational based on an InterProScan [139] over full genome assembly or single treatment transcripts. In most fungi, the matches appear between 560 to 640 bases before the end of the coding sequence for an annotation marked as “partial start, partial stop”. In the fungus *Lichtheimia ramose* (LK023324.1) which includes annotations of multiple exons, the sequence is similarly located within a coding sequence found 80 bases into the exon 10.

The single riboswitch identified in eukaryotes, the TPP riboswitch, has been shown to affect alternative splicing (Introduction) and we therefore looked whether our candidate riboswitch is encoded in the vicinity of a splice junction. Splicing prediction was performed using the NetAspGene 1.0 Server [140] on a subgroup of sequences that include GU at the end of the model and were missing additional exon annotations. The server is based on a machine learning model that was trained on known splicing sites in four Aspergillus genomes. In *Saccharomyces cerevisiae* (X73532.1, NM_001178465.3) a splicing donor site is predicted with high confidence at this position with a possible downstream acceptor site that includes a premature stop codon. Similar results were identified in *Ascoidea rubescens* (XM_020194220.1), *Candida orthopsilosis* (XM_003866255.1), *Candida tropicalis* (XM_002548829.1) and other subspecies with lower confidence probabilities. Thus in all likelihood our candidate riboswitch is also encoded near a splice junction sequence.

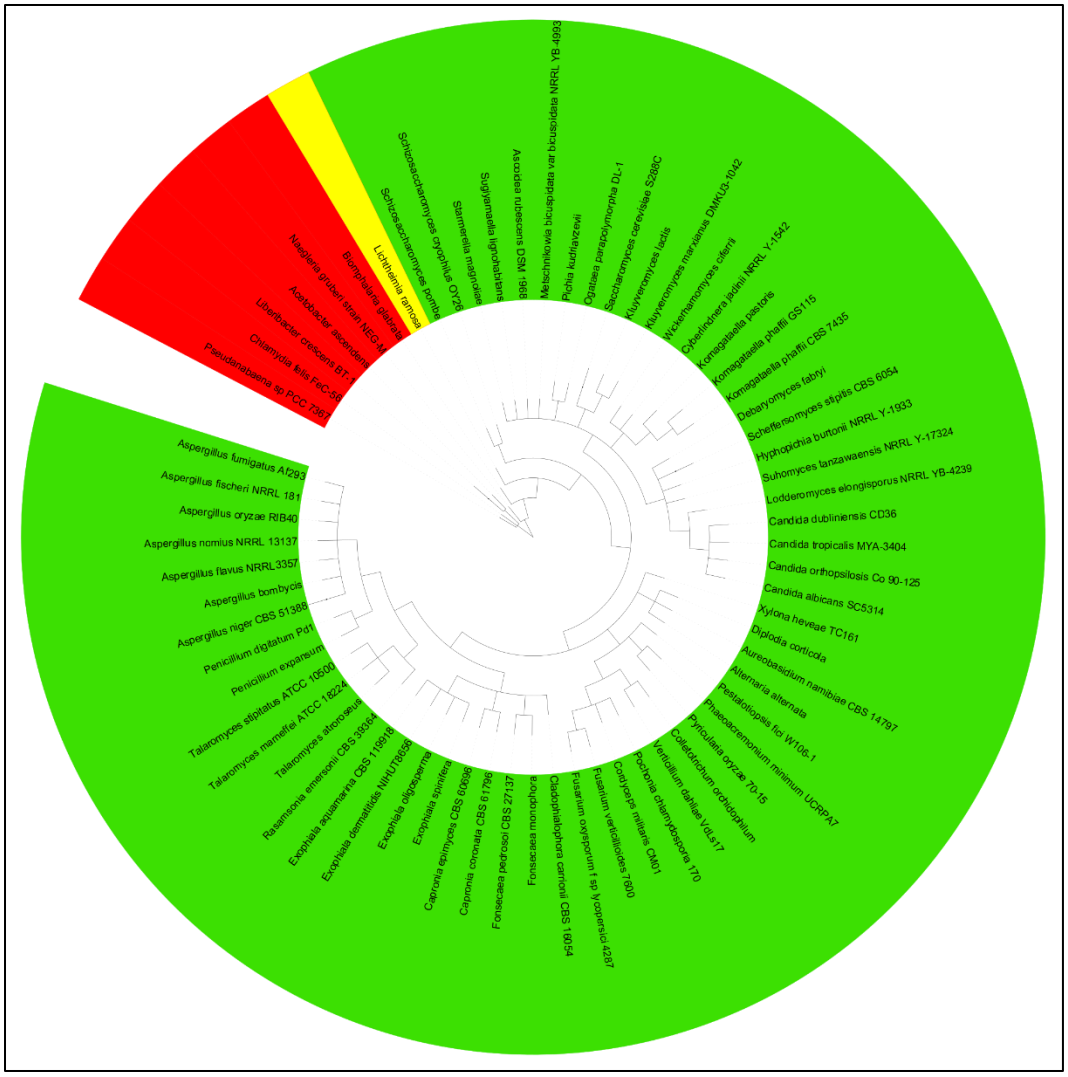

Figure 5-6: Phylogenetic distribution of the transketolase candidate family. Green, fungi from the Ascomycota phylum; Yellow, other fungi; Red, not fungi.

5.4.4. **Eukaryote myosin IX candidate**

Myosin is an ATP-dependent single-headed processive motor protein. The myosin IX variety also includes a RhoGAP domain present in GTPase-activating proteins. Myosin IX appeared early in the metazoan radiation and is present in multiple species from worms and fish to mammals [141].

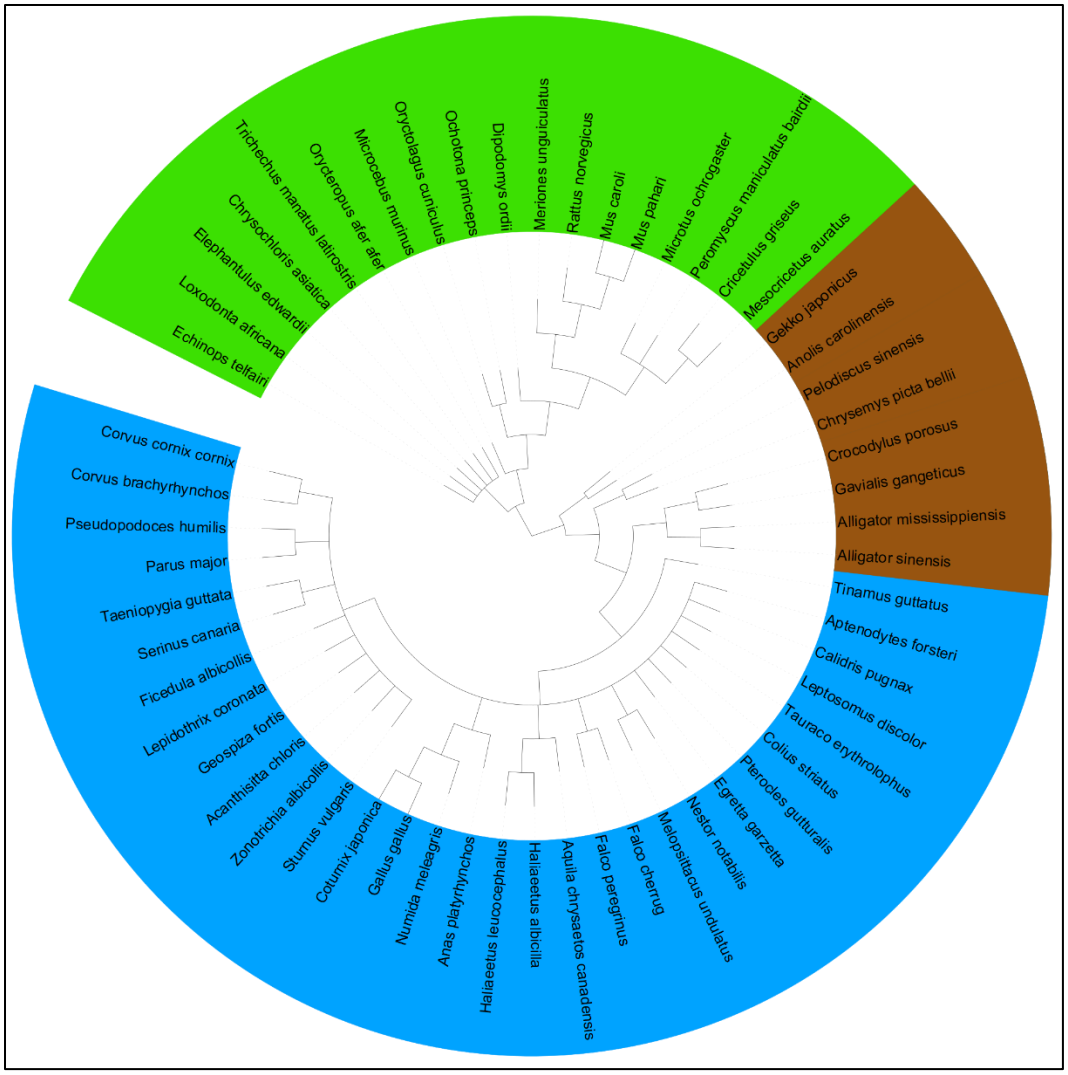

Figure 5-7: Phylogenetic distribution of the myosin candidate family. Green, mammals; Brown, reptiles; Blue, birds.

Our candidate family appears in mammals, reptiles and birds (figure 5-7). In *Rattus norvegicus* (NM_001271066.1, NM_001271067.1 and NM_012984.3), the candidate aptamer ends eight bases before the start of an exon predicted by transcript alignments. Other matches, such as in *Gallus gallus* and *Anolis carolinesis*, were found in the marked mRNA transcript variant but were not observed in others. This puts the sequence 200-1000 bases from the end of the spliced transcript.

The structure is very well preserved in the stems with mostly highly preserved bases and some compatible mutations with only a single covarying mutation (figure 5-8). Out of

the 252 matched sequences, 249 (98%) contain a U in the Watson-Crick binding position whereas only 99 (40%) contain a C. This indicates a possible binding affinity to adenine but also the possibility of guanine sensing. A well-preserved AG-UC pseudo-knot is also possible between the hairpin loops.

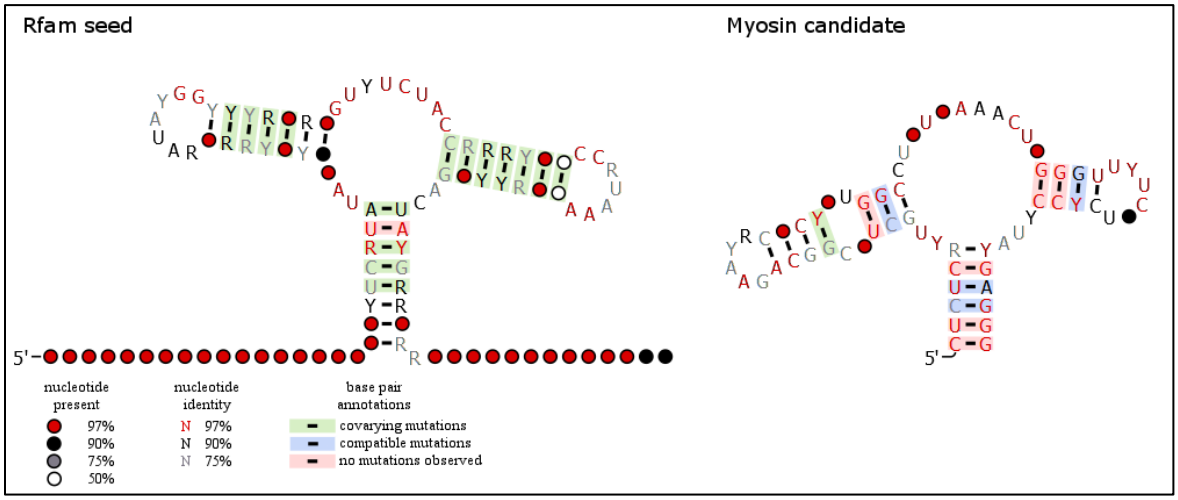

Figure 5-8: Myosin candidate model compared with Rfam model. Heavy structural preservation of the bottom and right stems include compatible G-C/U mutations and a single covarying mutation.

5.4.5. **Unannotated mammalian candidate**

A eukaryote candidate matched mainly in mammals. Unlike previous candidates, this candidate family is not annotated in most species. Out of the 925 sequences in the family, 878 are marked as genomic DNA, 13 as transcribed RNA and the rest are hypothetical or domain-containing (GTP binding domain, zinc finger domain, *etc*). Some matches in *Homo sapiens* seem to be located near or within regions of repetitive DNA.

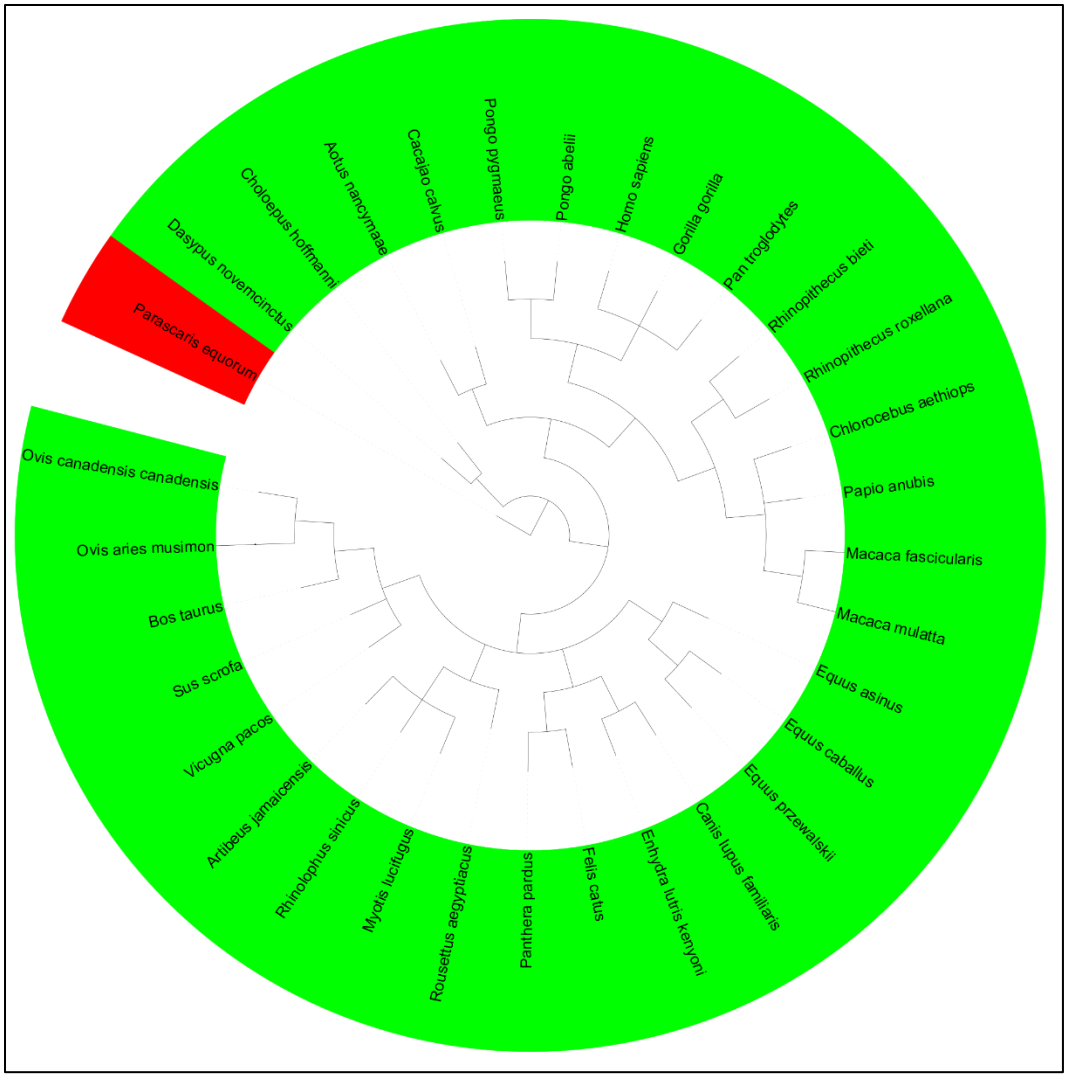

Figure 5-9: Phylogenetic distribution of the mammalian candidate family (green). Red, a single species of nematode.

The structure includes high covarying mutations in the stems indicative of a highly conserved structure (figure 5-10). An AAA to UUU pseudo-knot is viable between the two hairpin loops. The Watson-Crick base pairing position includes a C in 98% of the sequences indicating a guanine-binding aptamer. The purpose of the aptamer candidate is unknown and has almost no viable annotations nearby. However, the prokaryote purine riboswitch family was almost the only RNA family to appear in our searches with this amount of covarying mutations.

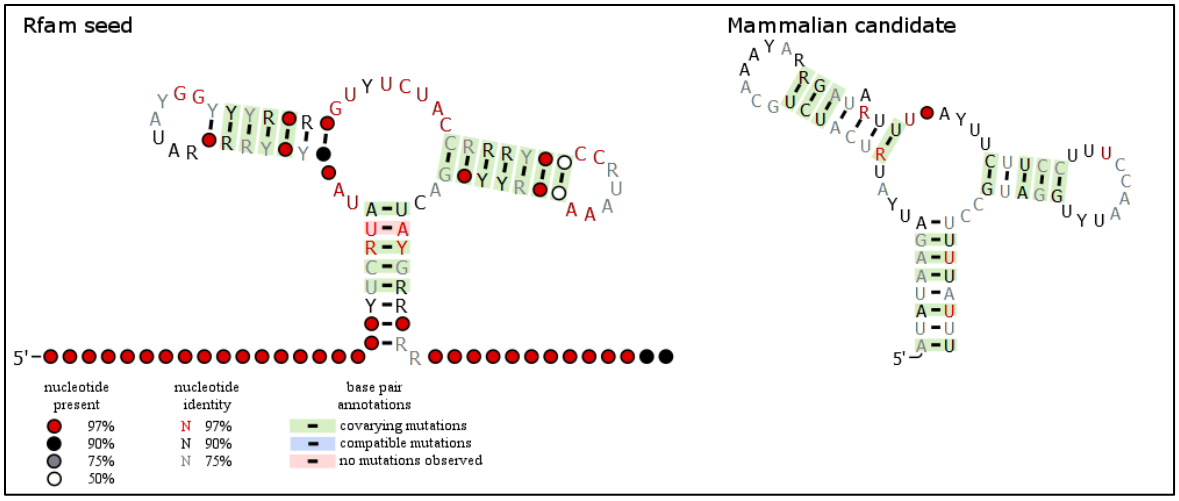

Figure 5-10: Mammalian candidate model compared with Rfam model. A G-U rich sequence with C present in the Watson-Crick binding position. The high covariation in the stems was rare in the reviewed results.

5.5. Conclusions

In this chapter we used an iterative method to generate covariance models for search results based on our designed candidate. The designed sequence is used to guide the search toward a thermodynamically stable structure. The model is built around it and generates context to the matched results. Preservation of structure is visible and phylogenetic distribution suggests an evolutionary connection.

Using a single designed sequence we were able to detect the prokaryote purine riboswitch family. The designed sequence started from highly limited sequence constraints and general shape structure. With only four nucleic acids that were kept as target, we designed a sequence that when expanded, generated a model almost identical to that of the Rfam purine riboswitch. As we conceptualize in figure 5-1, we were able to use the guidance of structure energy minimization based design to find a sequence in the prokaryote riboswitch space and expand it to cover the entire known family. The purine riboswitch model shows a high covariation in the stems that was detected by the expansion method. This result validates our method by discovering the original target element family.

To detect novel eukaryote candidates we reviewed candidate model families. None of the presented candidates were tested via in-line probing or any other experimental

validation and are considered computational predictions only. We first note that after removing redundant designed sequences and viewing the resulting families, it was rare to see structural conservation that did not come hand-in-hand with full sequence conservation. Results with little to no variation were filtered. Secondly, since the method searches for covariance, it will not show sequences that do not contain said covariance. In such cases, if a group of sequences from a multiple phylogeny is detected via the search, we might miss another group of sequences that contain the same backbone but with no covariation. These families can be misleading.

The eukaryote candidates were shown to preserve the target structure as well as the ligand binding nucleic acids. Each group had a dominant nucleic acid at the Watson-Crick base pairing position targeting either an adenine or guanine ligand. In two cases, indirect connections appear between the nearby annotations and the relevant ligand. Prediction of nearby intron-exon junctions in our candidate riboswitch is relevant to existing mechanisms found to be affected by the TPP riboswitch. However, we cannot exclude the possibility that novel riboswitches in eukaryotes may affect gene translation in new undiscovered ways that are not related to alternative splicing. An example could be by forcing ribosomal stalling via a strongly connected RNA structure which has been shown to cause frameshifts or to undergo endonucleolytic cleavage [142; 143]. Such a mechanism would require the riboswitch to be located inside an exon and not in an intron or UTR as previously observed. Therefore, our searches have included such regions as seen in some of the resultant families.

6. Thesis Conclusions

The search for novel eukaryote riboswitches is a very exciting and challenging task and is being attempted by many computational and experimental biologists. However, to date the only riboswitch family identified in eukaryotes remains that of the TPP riboswitch. Discovering a riboswitch family with a known unique ligand is a task for comparative genomics or experimental ligand screening. However, to discover new occurrences of existing riboswitch classes demands using wide-scale genomic scans. In this thesis I tackled the task of finding new occurrences of the purine riboswitch using a combination of bioinformatics tools and approaches. My work led to identification of several putative eukaryotic purine riboswitches which will hopefully be chosen for experimental validation.

I started by implementing a new pattern-matching tool, RNAPattMatch, that expands on the syntax used by R. Breaker in identifying the eukaryote TPP riboswitch. This new tool, described in Chapter 2, allows for RNA patterns which include sequences with full IUPAC alphabet support, a parallel structure and variable gaps. I implemented this tool using the very efficient affix array data structure. Affix arrays allow for a faster search than the simpler suffix arrays, they apply structure information directly by alternating the direction of the search every time a base pair is compared in the query. Since affix array can only be used for stem-loops searches, I break the query into stem-loops and score each one based on its specificity. The search then starts from the most specific stem and continues to the nearby most specific result until the entire query has been compared. This tool is simple and allows practitioners and computational biologist alike to run ncRNA searches on large genome databases. The simplicity of pattern building with the added power of the expanded syntax enable any user to iterate and optimize the pattern to suit their intended target ncRNA. We have implemented the tool as a standalone C++ package and as a web server which enables us to support practitioners with different computer skill levels. Pattern matching search methods are valid for high throughput scans of genomes if the patterns are well defined. To find new occurrences of riboswitches in eukaryote one would have to reduce the currently used constraints in both the sequence and structure space. This would mean an iterative process of pattern optimization that can be very time consuming. Another issue is that many patterns that can be found when examining a

sequence are not thermodynamically stable and thus could not maintain the secondary structure described by the pattern. For that reason, we chose to incorporate energy minimization of RNA structures.

To support such a concept, I improved on our in-house design tool, RNAfbinv. The old version allowed users to design sequences that match a target shape using energy minimization secondary structure prediction but added index-based constraints that ignored structural context. In a riboswitch aptamer, structural context is critical. Whereas sizes of specific motifs can change slightly to maintain a working binding pocket, key nucleic acids are required to perform the actual ligand binding within the binding pocket. In the purine aptamer the binding pocket binds to four nucleic acids in specific section areas of the multi-loop. With our new version, RNAfbinv 2.0, the design sequence is still folded using energy minimization based prediction. However, unlike the prior version, the sequence-structure information is combined into a tree in which every node contains the motif type, the sequence relative to that motif and additional identifiers that define it. A dynamic programming objective function then compares the designed sequence tree to a target tree. The comparison can now consider the sequence content of each of the motifs thus generating sequences that include constraints in the correct structural context.

The aforementioned tool is a core component of our novel energy minimization search method described in chapter 4. In biology, the term “Structure determines function” is key. Interactions between proteins, RNA, DNA and their activity are heavily structure dependent. Guided by this assumption we aimed to find a way to explore the sequence space in a more relaxed manner then has been done before. To reduce the search space under relaxed sequence constraints we used sequences that are predicted to fold into the desired shape. Since we wanted low search complexity to allow for full-scale genome search, we reduced the problem to sequence-based searches. We accomplished this by defining a “design-to-search” pipeline. In the design phase, we designed sequences using RNAfbinv 2.0 that are predicted to fold into the target shape and that contain additional sequence constraints deemed critical based on biological knowledge of the target purine riboswitch aptamer. The produced sequences were then used in efficient sequence-based searches on the NCBI 160 GB nt database.

This search method allowed us to find multiple candidates that appear to behave like a purine riboswitch aptamer. However, in a large enough database, any sequence can be matched. For example, a promising candidate was detected in *Aspergillus oryzae*, however an in-line probing assay on this candidate riboswitch showed that the predicted secondary structure did not match. Our goal is to create stronger support for any candidate riboswitch before time and money are wasted on experimental assays. Strong support for such a candidate can be evolutionary support in the form of a group or finding a specific sequence that appears near a relevant annotation in multiple closely related organisms. We therefore generated RNA families around a design sequence by iteratively expanding the initial search results. By choosing a high E-value threshold we were able to accept sequences that are slightly further away and that increase sequence variability. To avoid loss of structural information, we then aligned the results back to the covariance model and used the RNAfbinv 2.0 comparison function on the sequence and covariance alignment. Sequences that passed the filter were pushed back to the covariance model relaxing it.

Using the "design-to-search" pipeline I found multiple candidate families that may serve as purine riboswitch aptamers. The candidates appear near/in gene annotations that do not directly interact with purine metabolism but appear to be dependent on a purine moiety or to be in indirect competition with an enzyme related to purine metabolism. Moreover, the restriction mechanism may be different from those seen in prokaryotes. The TPP riboswitch was shown to affect alternative splicing [7], a process that does not exist in prokaryotes. New mechanisms such as ribosomal stalling may affect the efficiency of mRNA translation [142; 143] due to strongly connected motifs after ligand binding. This kind of mechanism could be present in exons and not only in intron regions or UTRs.

The candidate riboswitches presented have not been tested by in-line probing. I hope that in the future, some of those sequences will be tested by a qualified lab and that the first purine riboswitch in eukaryotes will be validated. There is a possibility that purine riboswitches in general or the specific class of purine riboswitch detected in prokaryotes may not be present in eukaryotes. Gene expression in eukaryotes is a complex process with multiple regulatory mechanisms that do not exist in bacteria, e.g., miRNA. Eukaryote translation initiation includes cap-binding protein complexes that are nonexistent in

bacteria who lack a cap binding mechanism. Novel riboswitch families may still be found in eukaryotes by use of sophisticated ligands that are prevalent in eukaryote metabolism. Such families will probably be detected using classic comparative genomics and experimental ligand screenings.

The minimum energy structure based search method can be used for any structure-dependent ncRNA. It can be used to scan very large genome databases while focusing on RNA secondary structure. This allows for contextless searches unlike comparative genomics approaches. The reconstruction of the Rfam purine riboswitch family shows that the method can produce a comprehensive family using minimal constraints on a high throughput scan of very large genome databases.

Future work on ncRNA scans can apply the pipeline as is. Given a sequence-structure description of any ncRNA family, the pipeline can identify RNA families with structure preserving covarying mutations. Reducing sequence constraints will produce results with higher sequence variation then existing methods while still being thermodynamically stable. The pipeline can also be improved as outlined below:

RNAfbinv 2.0 can be improved to support pseudo-knot constraints in the post alignment phase of the comparison function. This can be done using simple sequence alignment for Watson-Crick pairings between the different motifs with the existing infrastructure. Support for complex tertiary connections will be more difficult although not impossible using pattern definitions of multi-motif interactions. Ligand binding simulation can be added to the filtering phase. This requires tertiary structure prediction that is even less accurate then secondary structure prediction. There is an option to predict tertiary structure using a guide in the form of an experimentally validated structure of the target ncRNA. The problem with such methods is that as the sequences and length distribution of the different motifs are changed, it becomes a difficult and inaccurate task. Given such a prediction is made, if the user has prior knowledge of the binding pocket, a docking software such as DOCK 6 [144] and ligandRNA [145] can be used to score the binding affinity.

7. References

1. Nahvi, A., Sudarsan, N., Ebert, M.S., Zou, X., Brown, K.L., and Breaker, R.R. (2002). Genetic control by a metabolite binding mRNA. Chem Biol *9*, 1043.
2. Mironov, A.S., Gusarov, I., Rafikov, R., Lopez, L.E., Shatalin, K., Kreneva, R.A., Perumov, D.A., and Nudler, E. (2002). Sensing small molecules by nascent RNA: a mechanism to control transcription in bacteria. Cell *111*, 747-756.
3. Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A., and Eddy, S.R. (2003). Rfam: an RNA family database. Nucleic Acids Res *31*, 439-441.
4. Gardner, P.P., Daub, J., Tate, J.G., Nawrocki, E.P., Kolbe, D.L., Lindgreen, S., Wilkinson, A.C., Finn, R.D., Griffiths-Jones, S., Eddy, S.R., *et al.* (2009). Rfam: updates to the RNA families database. Nucleic Acids Res *37*, 136-140.
5. Nawrocki, E.P., Burge, S.W., Bateman, A., Daub, J., Eberhardt, R.Y., Eddy, S.R., Floden, E.W., Gardner, P.P., Jones, T.A., Tate, J., *et al.* (2015). Rfam 12.0: updates to the RNA families database. Nucleic Acids Res *43*, 130-137.
6. Kalvari, I., Argasinska, J., Quinones-Olvera, N., Nawrocki, E.P., Rivas, E., Eddy, S.R., Bateman, A., Finn, R.D., and Petrov, A.I. (2018). Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. Nucleic Acids Res *46*, 335-342.
7. Serganov, A., and Nudler, E. (2013). A decade of riboswitches. Cell *152*, 17-24.
8. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. (1990). Basic local alignment search tool. J Mol Biol *215*, 403-410.
9. Sherwood, A.V., and Henkin, T.M. (2016). Riboswitch-Mediated Gene Regulation: Novel RNA Architectures Dictate Gene Expression Responses. Annu Rev Microbiol *70*, 361-374.
10. Lu, C., Smith, A.M., Fuchs, R.T., Ding, F., Rajashankar, K., Henkin, T.M., and Ke, A. (2008). Crystal structures of the SAM-III/S(MK) riboswitch reveal the SAM-dependent translation inhibition mechanism. Nat Struct Mol Biol *15*, 1076-1083.
11. Sherlock, M.E., Sudarsan, N., Stav, S., and Breaker, R.R. (2018). Tandem riboswitches form a natural Boolean logic gate to control purine metabolism in bacteria. Elife *7*.

12. Dann, C.E., 3rd, Wakeman, C.A., Sieling, C.L., Baker, S.C., Irnov, I., and Winkler, W.C. (2007). Structure and mechanism of a metal-sensing regulatory RNA. Cell *130*, 878-892.
13. Nou, X., and Kadner, R.J. (2000). Adenosylcobalamin inhibits ribosome binding to btuB RNA. Proc Natl Acad Sci U S A *97*, 7190-7195.
14. Winkler, W.C., Nahvi, A., Roth, A., Collins, J.A., and Breaker, R.R. (2004). Control of gene expression by a natural metabolite-responsive ribozyme. Nature *428*, 281-286.
15. Winkler, W., Nahvi, A., and Breaker, R.R. (2002). Thiamine derivatives bind messenger RNAs directly to regulate bacterial gene expression. Nature *419*, 952-956.
16. Winkler, W.C., Nahvi, A., Sudarsan, N., Barrick, J.E., and Breaker, R.R. (2003). An mRNA structure that controls gene expression by binding S-adenosylmethionine. Nat Struct Biol *10*, 701-707.
17. Gelfand, M.S., Mironov, A.A., Jomantas, J., Kozlov, Y.I., and Perumov, D.A. (1999). A conserved RNA structure element involved in the regulation of bacterial riboflavin synthesis genes. Trends Genet *15*, 439-442.
18. Grundy, F.J., and Henkin, T.M. (1998). The S box regulon: a new global transcription termination control system for methionine and cysteine biosynthesis genes in gram-positive bacteria. Mol Microbiol *30*, 737-749.
19. Miranda-Rios, J., Navarro, M., and Soberon, M. (2001). A conserved RNA structure (thi box) is involved in regulation of thiamin biosynthetic gene expression in bacteria. Proc Natl Acad Sci U S A *98*, 9736-9741.
20. Epshtein, V., Mironov, A.S., and Nudler, E. (2003). The riboswitch-mediated control of sulfur metabolism in bacteria. Proc Natl Acad Sci U S A *100*, 5052-5056.
21. Winkler, W.C., Cohen-Chalamish, S., and Breaker, R.R. (2002). An mRNA structure that controls gene expression by binding FMN. Proc Natl Acad Sci U S A *99*, 15908-15913.
22. Breaker, R.R. (2011). Prospects for riboswitch discovery and analysis. Mol Cell *43*, 867-879.

23. Kwon, M., and Strobel, S.A. (2008). Chemical basis of glycine riboswitch cooperativity. RNA *14*, 25-34.
24. Erion, T.V., and Strobel, S.A. (2011). Identification of a tertiary interaction important for cooperative ligand binding by the glycine riboswitch. RNA *17*, 74-84.
25. Butler, E.B., Xiong, Y., Wang, J., and Strobel, S.A. (2011). Structural basis of cooperative ligand binding by the glycine riboswitch. Chem Biol *18*, 293-298.
26. Cochrane, J.C., Lipchock, S.V., and Strobel, S.A. (2007). Structural investigation of the GlmS ribozyme bound to Its catalytic cofactor. Chem Biol *14*, 97-105.
27. Davis, J.H., Dunican, B.F., and Strobel, S.A. (2011). glmS Riboswitch binding to the glucosamine-6-phosphate alpha-anomer shifts the pKa toward neutrality. Biochemistry *50*, 7236-7242.
28. Shanahan, C.A., Gaffney, B.L., Jones, R.A., and Strobel, S.A. (2011). Differential analogue binding by two classes of c-di-GMP riboswitches. J Am Chem Soc *133*, 15578-15592.
29. Smith, K.D., Lipchock, S.V., Ames, T.D., Wang, J., Breaker, R.R., and Strobel, S.A. (2009). Structural basis of ligand binding by a c-di-GMP riboswitch. Nat Struct Mol Biol *16*, 1218-1223.
30. Smith, K.D., Lipchock, S.V., Livingston, A.L., Shanahan, C.A., and Strobel, S.A. (2010). Structural and biochemical determinants of ligand binding by the c-di-GMP riboswitch. Biochemistry *49*, 7351-7359.
31. Smith, K.D., and Strobel, S.A. (2011). Interactions of the c-di-GMP riboswitch with its second messenger ligand. Biochem Soc Trans *39*, 647-651.
32. Smith, K.D., Lipchock, S.V., and Strobel, S.A. (2012). Structural and biochemical characterization of linear dinucleotide analogues bound to the c-di-GMP-I aptamer. Biochemistry *51*, 425-432.
33. Barrick, J.E., and Breaker, R.R. (2007). The distributions, mechanisms, and structures of metabolite-binding riboswitches. Genome Biol *8*, 239.
34. Sudarsan, N., Barrick, J.E., and Breaker, R.R. (2003). Metabolite-binding RNA domains are present in the genes of eukaryotes. RNA *9*, 644-647.

35. Barrick, J.E., Corbino, K.A., Winkler, W.C., Nahvi, A., Mandal, M., Collins, J., Lee, M., Roth, A., Sudarsan, N., Jona, I., *et al.* (2004). New RNA motifs suggest an expanded scope for riboswitches in bacterial genetic control. Proc Natl Acad Sci U S A *101*, 6421-6426.

36. Barrick, J.E. (2009). Predicting riboswitch regulation on a genomic scale. In Methods Mol Biol (Humana Press), pp. 1-13.

37. Weinberg, Z., and Ruzzo, W.L. (2004). Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. Bioinformatics *20 Suppl 1*, 334-341.

38. Yao, Z., Weinberg, Z., and Ruzzo, W.L. (2006). CMfinder--a covariance model based RNA motif finding algorithm. Bioinformatics *22*, 445-452.

39. Mandal, M., Lee, M., Barrick, J.E., Weinberg, Z., Emilsson, G.M., Ruzzo, W.L., and Breaker, R.R. (2004). A glycine-dependent riboswitch that uses cooperative binding to control gene expression. Science *306*, 275-279.

40. Weinberg, Z., Wang, J.X., Bogue, J., Yang, J., Corbino, K., Moy, R.H., and Breaker, R.R. (2010). Comparative genomics reveals 104 candidate structured RNAs from bacteria, archaea, and their metagenomes. Genome Biol *11*, 31.

41. Weinberg, Z., Barrick, J.E., Yao, Z., Roth, A., Kim, J.N., Gore, J., Wang, J.X., Lee, E.R., Block, K.F., Sudarsan, N., *et al.* (2007). Identification of 22 candidate structured RNAs in bacteria using the CMfinder comparative genomics pipeline. Nucleic Acids Res *35*, 4809-4819.

42. Weinberg, Z., Regulski, E.E., Hammond, M.C., Barrick, J.E., Yao, Z., Ruzzo, W.L., and Breaker, R.R. (2008). The aptamer core of SAM-IV riboswitches mimics the ligand-binding site of SAM-I riboswitches. RNA *14*, 822-828.

43. Weinberg, Z., Perreault, J., Meyer, M.M., and Breaker, R.R. (2009). Exceptional structured noncoding RNAs revealed by bacterial metagenome analysis. Nature *462*, 656-659.

44. Kazanov, M.D., Vitreschak, A.G., and Gelfand, M.S. (2007). Abundance and functional diversity of riboswitches in microbial communities. BMC Genomics *8*, 347.

45. Meyer, M.M., Ames, T.D., Smith, D.P., Weinberg, Z., Schwalbach, M.S., Giovannoni, S.J., and Breaker, R.R. (2009). Identification of candidate structured RNAs in the marine organism 'Candidatus Pelagibacter ubique'. BMC Genomics *10*, 268.

46. Bengert, P., and Dandekar, T. (2004). Riboswitch finder--a tool for identification of riboswitch RNAs. Nucleic Acids Res *32*, 154-159.

47. Abreu-Goodger, C., and Merino, E. (2005). RibEx: a web server for locating riboswitches and other conserved bacterial regulatory elements. Nucleic Acids Res *33*, 690-692.

48. Zhang, S., Borovok, I., Aharonowitz, Y., Sharan, R., and Bafna, V. (2006). A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements. Bioinformatics *22*, 557-565.

49. Klein, R.J., and Eddy, S.R. (2003). RSEARCH: finding homologs of single structured RNA sequences. BMC Bioinformatics *4*, 44.

50. Singh, P., Bandyopadhyay, P., Bhattacharya, S., Krishnamachari, A., and Sengupta, S. (2009). Riboswitch detection using profile hidden Markov models. BMC Bioinformatics *10*, 325.

51. Eddy, S.R. (2002). A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. BMC Bioinformatics *3*, 18.

52. Nawrocki, E.P., and Eddy, S.R. (2013). Infernal 1.1: 100-fold faster RNA homology searches. Bioinformatics *29*, 2933-2935.

53. Nawrocki, E.P. (2013). Annotating functional RNAs in genomes using Infernal. In Methods Mol Biol (Humana Press), pp. 163-197.

54. Freyhult, E., Moulton, V., and Clote, P. (2007). Boltzmann probability of RNA structural neighbors and riboswitch detection. Bioinformatics *23*, 2054-2062.

55. Kertesz, M., Wan, Y., Mazor, E., Rinn, J.L., Nutter, R.C., Chang, H.Y., and Segal, E. (2010). Genome-wide measurement of RNA secondary structure in yeast. Nature *467*, 103-107.

56. Hammann, C., and Westhof, E. (2007). Searching genomes for ribozymes and riboswitches. Genome Biol *8*, 210.

57. Cruz, J.A., and Westhof, E. (2011). Sequence-based identification of 3D structural modules in RNA with RMDetect. Nat Methods *8*, 513-521.
58. Mandal, M., Boese, B., Barrick, J.E., Winkler, W.C., and Breaker, R.R. (2003). Riboswitches control fundamental biochemical pathways in Bacillus subtilis and other bacteria. Cell *113*, 577-586.
59. Vitreschak, A.G., Rodionov, D.A., Mironov, A.A., and Gelfand, M.S. (2003). Regulation of the vitamin B12 metabolism and transport in bacteria by a conserved RNA structural element. RNA *9*, 1084-1097.
60. Laferriere, A., Gautheret, D., and Cedergren, R. (1994). An RNA pattern matching program with enhanced performance and portability. Comput Appl Biosci *10*, 211-212.
61. Grillo, G., Licciulli, F., Liuni, S., Sbisa, E., and Pesole, G. (2003). PatSearch: A program for the detection of patterns and structural motifs in nucleotide sequences. Nucleic Acids Res *31*, 3608-3612.
62. Macke, T.J., Ecker, D.J., Gutell, R.R., Gautheret, D., Case, D.A., and Sampath, R. (2001). RNAMotif, an RNA secondary structure definition and search algorithm. Nucleic Acids Res *29*, 4724-4735.
63. Meyer, F., Kurtz, S., Backofen, R., Will, S., and Beckstette, M. (2011). Structator: fast index-based search for RNA sequence-structure patterns. BMC Bioinformatics *12*, 214.
64. Strothmann, D. (2007). The affix array data structure and its applications to RNA secondary structure analysis. Theoretical Computer Science *389*, 278-294.
65. Leibovich, L., and Yakhini, Z. (2012). Efficient motif search in ranked lists and applications to variable gap motifs. Nucleic Acids Res *40*, 5832-5847.
66. Weinbrand, L., Avihoo, A., and Barash, D. (2013). RNAfbinv: an interactive Java application for fragment-based design of RNA sequences. Bioinformatics *29*, 2938-2940.
67. Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., and Schuster, P. (1994). Fast Folding and Comparison of Rna Secondary Structures. Monatsh Chem *125*, 167-188.

68. Reinharz, V., Ponty, Y., and Waldispuhl, J. (2013). A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. Bioinformatics *29*, 308-315.
69. Dotu, I., Lozano, G., Clote, P., and Martinez-Salas, E. (2013). Using RNA inverse folding to identify IRES-like structural subdomains. RNA Biol *10*, 1842-1852.
70. Zuker, M., and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res *9*, 133-148.
71. Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. Nucleic Acids Res *31*, 3406-3415.
72. Markham, N.R., and Zuker, M. (2008). UNAFold: software for nucleic acid folding and hybridization. In Bioinformatics (Humana Press), pp. 3-31.
73. Hofacker, I.L. (2003). Vienna RNA secondary structure server. Nucleic Acids Res *31*, 3429-3431.
74. Lorenz, R., Bernhart, S.H., Honer Zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P.F., and Hofacker, I.L. (2011). ViennaRNA Package 2.0. Algorithms Mol Biol *6*, 26.
75. Mathews, D.H., Disney, M.D., Childs, J.L., Schroeder, S.J., Zuker, M., and Turner, D.H. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc Natl Acad Sci U S A *101*, 7287-7292.
76. Barash, D., and Gabdank, I. (2010). Energy minimization methods applied to riboswitches: a perspective and challenges. RNA Biol *7*, 90-97.
77. Batey, R.T., Gilbert, S.D., and Montange, R.K. (2004). Structure of a natural guanine-responsive riboswitch complexed with the metabolite hypoxanthine. Nature *432*, 411-415.
78. Holbrook, S.R. (2005). RNA structure: the long and the short of it. Curr Opin Struct Biol *15*, 302-308.
79. Lescoute, A., and Westhof, E. (2005). Riboswitch structures: purine ligands replace tertiary contacts. Chem Biol *12*, 10-13.

80. Cohen, A., Bocobza, S., Veksler, I., Gabdank, I., Barash, D., Aharoni, A., Shapira, M., and Kedem, K. (2008). Computational identification of three-way junctions in folded RNAs: a case study in Arabidopsis. In Silico Biol *8*, 105-120.
81. Parker, B.J., Moltke, I., Roth, A., Washietl, S., Wen, J., Kellis, M., Breaker, R., and Pedersen, J.S. (2011). New families of human regulatory RNA structures identified by comparative analysis of vertebrate genomes. Genome Res *21*, 1929-1943.
82. Cornish-Bowden, A. (1985). Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. Nucleic Acids Res *13*, 3021-3030.
83. Tinoco, I., Jr., and Bustamante, C. (1999). How RNA folds. J Mol Biol *293*, 271-281.
84. Bon, M., Vernizzi, G., Orland, H., and Zee, A. (2008). Topological classification of RNA structures. J Mol Biol *379*, 900-911.
85. Shapiro, B.A. (1988). An algorithm for comparing multiple RNA secondary structures. Comput Appl Biosci *4*, 387-393.
86. McNaught, A.D. (1997). Compendium of chemical terminology, Vol 1669 (Blackwell Science Oxford).
87. Nussinov, R., Pieczenik, G., Griggs, J.R., and Kleitman, D.J. (1978). Algorithms for Loop Matchings. Siam Journal on Applied Mathematics *35*, 68-82.
88. Tinoco, I., Jr., Borer, P.N., Dengler, B., Levin, M.D., Uhlenbeck, O.C., Crothers, D.M., and Bralla, J. (1973). Improved estimation of secondary structure in ribonucleic acids. Nat New Biol *246*, 40-41.
89. Turner, D.H., and Mathews, D.H. (2010). NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Res *38*, 280-282.
90. Clote, P., Kranakis, E., Krizanc, D., and Salvy, B. (2009). Asymptotics of canonical and saturated RNA secondary structures. J Bioinform Comput Biol *7*, 869-893.
91. Cruz, J.A., and Westhof, E. (2011). Identification and annotation of noncoding RNAs in Saccharomycotina. C R Biol *334*, 671-678.
92. Ruzzo, W.L., and Gorodkin, J. (2013). De novo discovery of structured ncRNA motifs in genomic sequences. In Methods Mol Biol (Humana Press), pp. 303-318.

93. Riccitelli, N.J., and Luptak, A. (2010). Computational discovery of folded RNA domains in genomes and in vitro selected libraries. Methods *52*, 133-140.
94. Havill, J.T., Bhatiya, C., Johnson, S.M., Sheets, J.D., and Thompson, J.S. (2014). A new approach for detecting riboswitches in DNA sequences. Bioinformatics *30*, 3012-3019.
95. Eddy, S.R. (1998). Profile hidden Markov models. Bioinformatics *14*, 755-763.
96. El-Gebali, S., Mistry, J., Bateman, A., Eddy, S.R., Luciani, A., Potter, S.C., Qureshi, M., Richardson, L.J., Salazar, G.A., Smart, A., *et al.* (2019). The Pfam protein families database in 2019. Nucleic Acids Res *47*, 427-432.
97. Eddy, S.R. (2001). HMMER: Profile Hidden Markov Models for biological sequence analysis.
98. Batey, R.T. (2012). Structure and mechanism of purine-binding riboswitches. Q Rev Biophys *45*, 345-381.
99. Kim, J.N., and Breaker, R.R. (2008). Purine sensing by riboswitches. Biol Cell *100*, 1-11.
100. Strobel, S.A., and Cochrane, J.C. (2007). RNA catalysis: ribozymes, ribosomes, and riboswitches. Curr Opin Chem Biol *11*, 636-643.
101. Taft, R.J., Pang, K.C., Mercer, T.R., Dinger, M., and Mattick, J.S. (2010). Non-coding RNAs: regulators of disease. J Pathol *220*, 126-139.
102. Gautheret, D., Major, F., and Cedergren, R. (1990). Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for tRNA. Comput Appl Biosci *6*, 325-331.
103. Tsui, V., Macke, T., and Case, D.A. (2003). A novel method for finding tRNA genes. RNA *9*, 507-517.
104. Chang, T.H., Huang, H.D., Wu, L.C., Yeh, C.T., Liu, B.J., and Horng, J.T. (2009). Computational identification of riboswitches based on RNA conserved functional sequences and conformations. RNA *15*, 1426-1430.
105. Abouelhoda, M.I., Kurtz, S., and Ohlebusch, E. (2004). Replacing suffix trees with enhanced suffix arrays. Journal of Discrete Algorithms *2*, 53-86.
106. Isaacs, F.J., Dwyer, D.J., and Collins, J.J. (2006). RNA synthetic biology. Nat Biotechnol *24*, 545-554.

107. Findeiß, S., Wachsmuth, M., Mörl, M., and F.Stadler, P. (2015). Design of transcription regulating riboswitches. In Methods Enzymol (Elsevier), pp. 1-22.
108. Wachsmuth, M., Domin, G., Lorenz, R., Serfling, R., Findeiss, S., Stadler, P.F., and Morl, M. (2015). Design criteria for synthetic riboswitches acting on transcription. RNA Biol *12*, 221-231.
109. Soukup, G.A., and Breaker, R.R. (1999). Nucleic acid molecular switches. Trends Biotechnol *17*, 469-476.
110. Chang, A.L., Wolf, J.J., and Smolke, C.D. (2012). Synthetic RNA switches as a tool for temporal and spatial control over gene expression. Curr Opin Biotechnol *23*, 679-688.
111. Berens, C., and Suess, B. (2015). Riboswitch engineering - making the all-important second and third steps. Curr Opin Biotechnol *31*, 10-15.
112. Drory Retwitzer, M., Kifer, I., Sengupta, S., Yakhini, Z., and Barash, D. (2015). An Efficient Minimum Free Energy Structure-Based Search Method for Riboswitch Identification Based on Inverse RNA Folding. PLoS One *10*, 0134262.
113. Garcia-Martin, J.A., Clote, P., and Dotu, I. (2013). RNAiFold: a web server for RNA inverse folding and molecular design. Nucleic Acids Res *41*, 465-470.
114. Busch, A., and Backofen, R. (2006). INFO-RNA--a fast approach to inverse RNA folding. Bioinformatics *22*, 1823-1831.
115. Aguirre-Hernandez, R., Hoos, H.H., and Condon, A. (2007). Computational RNA secondary structure design: empirical complexity and improved methods. BMC Bioinformatics *8*, 34.
116. Zadeh, J.N., Wolfe, B.R., and Pierce, N.A. (2011). Nucleic acid sequence design via efficient ensemble defect optimization. J Comput Chem *32*, 439-452.
117. Lyngso, R.B., Anderson, J.W., Sizikova, E., Badugu, A., Hyland, T., and Hein, J. (2012). Frnakenstein: multiple target inverse RNA folding. BMC Bioinformatics *13*, 260.
118. Cohen, B., and Skiena, S. (2003). Natural selection and algorithmic design of mRNA. J Comput Biol *10*, 419-432.
119. Taneda, A. (2012). Multi-objective genetic algorithm for pseudoknotted RNA sequence design. Front Genet *3*, 36.

120. Esmaili-Taheri, A., and Ganjtabesh, M. (2015). ERD: a fast and reliable tool for RNA design including constraints. BMC Bioinformatics *16*, 20.
121. Kleinkauf, R., Mann, M., and Backofen, R. (2015). antaRNA: ant colony-based RNA sequence design. Bioinformatics *31*, 3114-3121.
122. Bindewald, E., Afonin, K., Jaeger, L., and Shapiro, B.A. (2011). Multistrand RNA secondary structure prediction and nanostructure design including pseudoknots. ACS Nano *5*, 9542-9551.
123. Yesselman, J.D., and Das, R. (2015). RNA-Redesign: a web server for fixed-backbone 3D design of RNA. Nucleic Acids Res *43*, 498-501.
124. Dromi, N., Avihoo, A., and Barash, D. (2008). Reconstruction of natural RNA sequences from RNA shape, thermodynamic stability, mutational robustness, and linguistic complexity by evolutionary computation. J Biomol Struct Dyn *26*, 147-162.
125. Avihoo, A., Churkin, A., and Barash, D. (2011). RNAexinv: An extended inverse RNA folding from shape and physical attributes to sequences. BMC Bioinformatics *12*, 319.
126. Darty, K., Denise, A., and Ponty, Y. (2009). VARNA: Interactive drawing and editing of the RNA secondary structure. Bioinformatics *25*, 1974-1975.
127. Bergig, O., Barash, D., Nudler, E., and Kedem, K. (2004). STR2: a structure to string approach for locating G-box riboswitch shapes in pre-selected genes. In Silico Biol *4*, 593-604.
128. Veksler-Lublinsky, I., Ziv-Ukelson, M., Barash, D., and Kedem, K. (2007). A structure-based flexible search method for motifs in RNA. J Comput Biol *14*, 908-926.
129. Drory Retwitzer, M., Reinharz, V., Ponty, Y., Waldispuhl, J., and Barash, D. (2016). incaRNAfbinv: a web server for the fragment-based design of RNA sequences. Nucleic Acids Res *44*, 308-314.
130. Regulski, E.E., and Breaker, R.R. (2008). In-line probing analysis of riboswitches. In Post-Transcriptional Gene Regulation (Humana Press), pp. 53-67.

131. Strauss, B., Nierth, A., Singer, M., and Jaschke, A. (2012). Direct structural analysis of modified RNA by fluorescent in-line probing. Nucleic Acids Res *40*, 861-870.

132. Kubodera, T., Watanabe, M., Yoshiuchi, K., Yamashita, N., Nishimura, A., Nakai, S., Gomi, K., and Hanamoto, H. (2003). Thiamine-regulated gene expression of Aspergillus oryzae thiA requires splicing of the intron containing a riboswitch-like domain in the 5 '-UTR. FEBS Lett *555*, 516-520.

133. Tasca, C.I., Santos, T.G., Tavares, R.G., Battastini, A.M.O., Rocha, J.B.T., and Souza, D.O. (2004). Guanine derivatives modulate L-glutamate uptake into rat brain synaptic vesicles. Neurochem Int *44*, 423-431.

134. Barquist, L., Burge, S.W., and Gardner, P.P. (2016). Studying RNA Homology and Conservation with Infernal: From Single Sequences to RNA Families. Curr Protoc Bioinformatics *54*, 12.13.11-12.13.25.

135. Weinberg, Z., and Breaker, R.R. (2011). R2R--software to speed the depiction of aesthetic consensus RNA secondary structures. BMC Bioinformatics *12*, 3.

136. Singh, P., and Sengupta, S. (2012). Phylogenetic analysis and comparative genomics of purine riboswitch distribution in prokaryotes. Evol Bioinform Online *8*, 589-609.

137. De La Haba, G., Leder, I.G., and Racker, E. (1955). Crystalline transketolase from bakers' yeast: isolation and properties. J Biol Chem *214*, 409-426.

138. Zalkin, H., and Dixon, J.E. (1992). De novo purine nucleotide biosynthesis. In Prog Nucleic Acid Res Mol Biol (Elsevier), pp. 259-287.

139. Quevillon, E., Silventoinen, V., Pillai, S., Harte, N., Mulder, N., Apweiler, R., and Lopez, R. (2005). InterProScan: protein domains identifier. Nucleic Acids Res *33*, 116-120.

140. Wang, K., Ussery, D.W., and Brunak, S. (2009). Analysis and prediction of gene splice sites in four Aspergillus genomes. Fungal Genet Biol *46 Suppl 1*, 14-18.

141. Thompson, R.F., and Langford, G.M. (2002). Myosin superfamily evolutionary history. Anat Rec *268*, 276-289.

142. Joazeiro, C.A.P. (2017). Ribosomal Stalling During Translation: Providing Substrates for Ribosome-Associated Protein Quality Control. Annu Rev Cell Dev Biol *33*, 343-368.

143. Yu, C.H., Teulade-Fichou, M.P., and Olsthoorn, R.C. (2014). Stimulation of ribosomal frameshifting by RNA G-quadruplex structures. Nucleic Acids Res *42*, 1887-1892.

144. Lang, P.T., Brozell, S.R., Mukherjee, S., Pettersen, E.F., Meng, E.C., Thomas, V., Rizzo, R.C., Case, D.A., James, T.L., and Kuntz, I.D. (2009). DOCK 6: combining techniques to model RNA-small molecule complexes. RNA *15*, 1219-1230.

145. Philips, A., Milanowska, K., Lach, G., and Bujnicki, J.M. (2013). LigandRNA: computational predictor of RNA-ligand interactions. RNA *19*, 1605-1616.

146. Drory Retwitzer, M., Polishchuk, M., Churkin, E., Kifer, I., Yakhini, Z., and Barash, D. (2015). RNAPattMatch: a web server for RNA sequence/structure motif detection based on pattern matching with flexible gaps. Nucleic Acids Res *43*, 507-512.

בחרנו במשפחת מתגי רנ"א המגיבים לפורינים כמטרת החיפוש. זו משפחה שנכון להיום, לא נמצאה באיקריוטים. המשפחה הזו נבחרה מכיוון שתחזיות הקיפול השניוני המבוססות על מזעור אנרגיה חופשית מדוייקות עבור חלק נכבד מהרצפים שנמצאו. באמצעות שיטת החיפוש הראשונית הצלחנו לזהות מספר רב של המתגים הקיימים בפרוקריוטים ומספר רצפים חדשים באיקריוטים. כאשר הרחבנו את השיטה לזיהוי משפחות הצלחנו לשחזר את המשפחה הידועה בפרוקריוטים באמצעות מבנה ושימור רצפי מינימלי המסמל רק ארבע בסיסים הנקשרים ישירות לפורין. בנוסף, הצגנו משפחות חדשות שמצאנו באיקריוטים. המשפחות החדשות לא נבדקו בניסוי מעבדה, אך אם יעברו בהצלחה, יהיו מתגי הרנ"א מבוססי פורין הראשונים באיקריוטים. אני מאמין שניתן באמצעות השיטה שפיתחנו למצוא רצפים חדשים של משפחות רנ"א לא מקודד נוספות.

תקציר

מתגי רנ"א הינם יסודות שליטה גנטיים המספקים לתא מנגנון בקרת גנים. הם התגלו לראשונה בחיידקים שם נמצא שהם שולטים בין היתר על תהליכי סיום השיעתוק ותחילת התרגום של הגן עליו הם מפקחים [1; 2]. מתגי רנ"א עובדים ללא התערבות חיצונים של חלבונים ולכן נחשבים למנגנון בקרה עתיק במונחים אבולוציונים. מסד הנתונים Rfam [3-6] מסווג 39 משפחות שונות של מתגי רנ"א, רק אחת מהן נמצאה באיקריוטים. ממשפחת מתגי ה-TPP נמצאה בחיידקים, ארכאונים, פטריות וצמחים אך לא בבעלי חיים [7]. מספר הדוגמאות המצומצמות שנמצאו ביצורים איקריוטים התגלו על ידי שימוש בשיטות חיפוש מחשוביות מבוססות רצף, כגון BLAST [8], בשילוב עם שיטות התאמת תבניות המשלבות מידע על זיווג הבסיסים במבנה השניוני. כיום, שיטת החיפוש הנפוצה ביותר הינה שיטה סטטיסטית המייצרת מודל לשימור רצפי ומבני באמצעות שונות משותפת של רצפים קיימים. השיטות המוזכרות אינם מתחשבות במזעור האנרגיה החופשית של הרצף. בהינתן זאת, במקביל להתקדמות בשיטות מבוססות שונות משותפת, ישנה מוטיבציה ברורה לפיתוח שיטות חיפוש ביואינפורמטיות חדשות שיוכלו לדגום את מרחב החיפוש בצורה גמישה תוך כדי שימוש בהנחיה מבוססת מבנה שניוני ללא התפשרות על היעילות של שיטות מבוססות הרצף.

בתזה זו אני מציג כלי התאמת תבניות גמיש מבוסס רצף-מבנה שניוני עבור רנ"א הממומש על ידי מבנה נתונים יעיל ושיטת חיפוש מקורית המתבססת על מבנים שניוניים ממוזערי אנרגיה עבור רצפי רנ"א לא מקודדים. בנוסף, נרחיב את שיטת החיפוש על מנת לייצר משפחות רנ"א סביב תוצאות יחידות. שיטות התאמת תבניות הינן כלי חיפוש טוב בעבור ביולוג מומחה, הן פחות יעילות לשימוש על מסדי נתונים גנומיים גדולים כאשר התבנית גמישה.

שיטת החיפוש מתבססת על עיקרון מעולם הביולוגיה הסינטטית. השיטה הינה תהליך בו אנו מתכננים רצפי רנ"א על בסיס משפחת הרנ"א אותה אנו מחפשים ואז משתמשים בשיטות מבוססות רצף על מנת לחפש ביעילות על מסדי נתונים גנומיים גדולים. התכנון מבוסס על כלי הפותר את בעיית היפוך הרנ"א שפיתחנו, בשם RNAfbinv 2.0, המייצר רצפי רנ"א על בסיס מבנה שניוני, הגבלות רצפיות ופרמטרים נוספים. הרצפים אותם תכננו מייצגים מועמדים סינטטים של הרנ"א הלא מקודד אותו הגדרנו כמטרה. רצפים אלה היוו בסיס לחיפוש רצפי מבוסס כלים מוכחים כגון BLAST ומודלים של שונות משותפת על מסדי נתונים גנומיים גדולים. תוצאות החיפוש מסוננות על בסיס איבוד מידע רצפי ומבני ואנוטציות המוסיפות גיבוי גנטי. מאחר ועבור מסד נתונים גדול מספיק כמעט כל רצף עלול להימצא גם שיטת הסינון הנ"ל אינה מספיקה. לכן פיתחנו תהליך איטרטיבי המייצר משפחה של רצפים סביב תוצאות החיפוש מהרצף המתוכנן. בהינתן זיהוי של משפחה, ניתן להתבונן על קשרים אבולוציונים, אנוטציות משותפות ומוטציות שומרות מבנה.