A GRAMMAR BASED METHODOLOGY FOR STRUCTURAL MOTIF FINDING IN ncRNA DATABASE SEARCH

Daniel Quest[⋆], **William Tapprich**[°], **Hesham Ali**[⋆]
[⋆] College of Information Science and Technology, University of Nebraska at Omaha
[°] Department of Biology, University of Nebraska at Omaha
Omaha, NE 68182-0694, USA
E-mail: djquest@unmc.edu

In recent years, sequence database searching has been conducted through local alignment heuristics, pattern-matching, and comparison of short statistically significant patterns. While these approaches have unlocked many clues as to sequence relationships, they are limited in that they do not provide context-sensitive searching capabilities (e.g. considering pseudoknots, protein binding positions, and complementary base pairs). Stochastic grammars (hidden Markov models HMMs and stochastic context-free grammars SCFG) do allow for flexibility in terms of local context, but the context comes at the cost of increased computational complexity. In this paper we introduce a new grammar based method for searching for RNA motifs that exist within a conserved RNA structure. Our method constrains computational complexity by using a chain of topology elements. Through the use of a case study we present the algorithmic approach and benchmark our approach against traditional methods.

1. INTRODUCTION

Functional non-coding RNA (ncRNA) has received great attention in recent years because of their diverse functional activities within the cell. A ncRNA forms a secondary structure that enables other molecules to interact with it and carry out functional activities. In many cases, molecules interact with conserved primary structure patterns or motifs given that the ncRNA is in the correct secondary structure conformation. Because of this, the bioinformatics community has focused considerable energy towards methods that predict ncRNA secondary structure and search for homologous structures within a sequence database (e.g. 1, 2, 3) .

Currently there are many approaches to find a RNA homolog. The first approach is to construct a structure for the sequence, and then use that structure to query a sequence database 1. One option to construct the structure is to use sequence profiles of the RNA as it is conserved through evolution 4. Another approach is to use a package such as Mfold 5 and chemical probing validation experiments to determine the RNA structure. As soon as the structure is determined, one can use pattern-matching software to find structural homologs within the RNA database.

Pattern-matching software packages were first used to find homologous tRNAs 6, 7. Over time, they have evolved to consider multiple different abstractions of the structural patterns. Pattern-matching programs have evolved from regular expression tools to scripting languages capable of considering errors, non-watson-crick base pairs, complementary base pairing, and common structural profiles. Some example programs include RnaBob 8, RNAMOT 9, Palingol 10, and RNAMotif 11. Although these methods are extremely powerful and fast, they require significant user expertise to obtain reliable profiles. In addition, they do not easily allow probabilistic scoring schemes to be integrated into them. This implies that these tools return all hits that are possible given our current understanding of the secondary structure. These tools do not rank profiles based on what is most likely to occur based on the phylogenetic relationships of the ncRNA.

A second approach to finding a ncRNA homolog is to use a stochastic context free grammar (SCFG) 12 to simultaneously align the primary sequence and the secondary structure. SCFGs have an advantage over pattern-matching programs in that they require less manual expertise and tuning to find accurate structural alignments once the global parameters are set. In practice however they are impractical because of running time $O(n^4)$ 1. If pseudoknots are considered 13 time complexity ($O(n^6)$) makes database searches impossible. To circumvent these obstacles Weinberg and Ruzzo proposed a HMM filter that allows for faster ncRNA searches without the loss of accuracy 2. Recently Zhang *et al* proposed an additional filter 3 and a sequence filtering methodology

14 for constructing fast SCFG searchers without the loss of accuracy.

This capability allows us to construct queries over large datasets based on primary and secondary structure instead of primary sequence alone. However, implicit in the assumptions of these filtering techniques is the concept that scoring matrices are homogeneous across all putative alignment regions. In some cases however we have more evidence that require our scoring system to be heterogeneous. For example, when some of the bases have been biologically verified by chemical probing and other bases have not been verified we wish a model with two classes verified and not verified. We then wish to search for RNAs that have a conserved secondary structure subject to the constraint that all verified bases remain functional. In other words, we wish the bases in the motif to remain functional, and so they can not be considered in other base pair interactions in the folding of the molecule.

This problem can be solved with pattern-matching programs, but search results suffer because errors are not scored in a probabilistic way. Consequently, for a short ncRNA, such a program can return a pattern that satisfies all constraints but is not closely related to known ncRNA found in nature. This implies that the number of matches is dictated by the length of the database instead of functional relationships inside the database.

SCFGs can be modified to impose additional constraints through additional grammar rules, however this process is time consuming. More importantly, changing the grammar and the parameters has the effect of also changing the relationships used to construct the filters that allow SCFGs to run in reasonable time.

In this paper, we propose a new approach to search a sequence database for RNA structures that have known functional sites (motifs). Our approach uses the strategy of nested grammars to simultaneously integrate secondary structure, primary structure, and biologically verified constraints. We will show that our method is capable of finding significant substrings or motifs when pattern-matching approaches can not, and that our method can serve as a reasonable second step for imposing constraints on putative hits from a filtered SCFG filter (therefore avoiding the need to construct constraint-aware filters). To illustrate our nested-grammar paradigm, we will first show that a grammar with favorable runtime characteristics can be used as an approximation for a grammar with more complex runtime characteristics. In this way, a heuristic for a complex grammar G can be generated via a simple grammar G'. We also show that G' can provide a solution within τ for G where τ is an arbitrary error threshold. Finally, we illustrate our algorithm for evaluating G and G' via an example and a case study.

2. PROBLEM DESCRIPTION AND METHODOLOGY

Our primary interest in this work is to search large databases for significant signals within a conserved two-dimensional structure. Given that we know some pattern or signal from biologically verified data, we wish to find two-dimensional structural homologs in a database subject to the constraint that the structural homolog must contain this signal. In this paper, we present a robust grammar-based approach for finding non-deterministic RNA structural motifs in a conserved secondary structure. Like the pattern-matching approaches, our approach allows the user to decide the level of flexibility of constraints of the profile to search. Given reasonable constraints, the proposed approach also has a favorable computational complexity. The core idea is to define a primary grammar for running the nucleotide comparisons, and a secondary grammar to model the secondary structure relationships. This idea is similar to the idea independently developed in MilPat using constraint networks 15. In MilPat, a constraint network is used to model secondary structure dependencies. Our approach, on the other hand, uses a secondary grammar to model constraints. The key advantage of using nested grammars is that all constraints can be integrated homogeneously using Bayes rule. The direct impact is that we allow for mismatches and thus our models can be entirely probabilistic in nature.

Our nested grammar based method functions by considering two grammars: G and G'. The structural alignment grammar G represnents the known constraints that exists in the molecule. The sequence alignment grammar G' represents an ordered set of phylogenetic subsequences found in the structure. Elements of G' may be scored with traditional scoring matricies, or additional information from biological experements can be added to score subsequences heterogeneously. We use a pairwise hidden markov model (discussed below) to evaluate all possible alignment positions for subsequences in G' and then combine evidence using the more robust SCFG grammar to select the subsequence alignments with most supporting evidence and construct the grammar to sequence alignment. The next few paragraphs provide a background for our method. In sections 2.1 to 2.5 we detail the key components of the algorithm. The algorithm in its entirety is presented in section 3.

Consider a pattern $P = \{p_1, p_2, \ldots, p_m\}$ that we wish to search for, and a sequence S such that $|P| < |S|$ ($|x|$ denotes the length of x). Our objective is to create a sequence S from P using production rules from the set: insert ($I = {}^{-}_{a}$), delete ($D = {}^{a}_{-}$), match ($M = {}^{a}_{a}$), and mismatch ($X = {}^{b}_{a}$) where a and b are any characters in the terminal alphabet Σ such that $a \neq b$ and $-$ is a space. The production transcript T is an ordered list of production rules that produces S from P 16. T is a generative regular probabilistic grammar (G) that, for each production rule, generates a pair of characters corresponding to P and one corresponding to S such that both P and S are produced (a pairwise hidden Markov model or PHMM). For example, if $P = AAAC$ and $S = TAGCC$ we could construct both P and S with the production transcript $T = \{X, D, M, I, M, I\}$ as shown in figure 1.

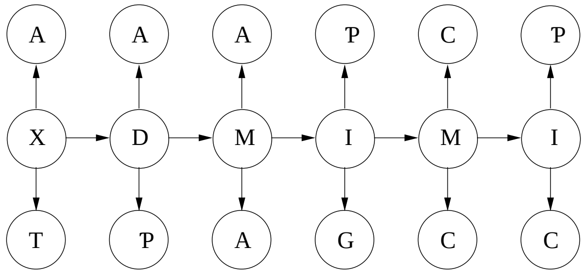


Fig. 1. A PHMM representation of a production transcript

PHMMs have proven to be useful in global sequence alignments between two sequences. In order to cluster gaps or account for palindromic base pairing, the grammar needs to be extended to include both primitive production rules (I, D, M, X) and non-terminal production rules. A non-terminal production rule is a grammar production rule that may produce any other production rule (terminal or non-terminal including itself) from a finite set of options. Non-terminal production rules exist to allow shortcuts in the alignment path to more closely approximate our biological problem. The classic example is one where gaps are clustered together to represent introns in an alignment between genomic DNA and messenger RNA. Such a grammar could be represented as follows:

$$\begin{aligned} G_1: \quad & T \rightarrow {}^{a}_{a}T \;|\; {}^{b}_{a}T \;|\; {}^{a}_{-}T \;| \\ & \qquad {}^{-}_{a}T \;|\; {}^{-}_{a}L \;|\; {}^{a}_{-}R \;|\; \epsilon \\ & L \rightarrow {}^{-}_{a}L \;|\; T \;|\; \epsilon \\ & R \rightarrow {}^{a}_{-}R \;|\; T \;|\; \epsilon \end{aligned}$$

In G_1, production rules L and R represent the gaps in P and S respectively. ϵ represents the final character in P and S. More recently, non-terminal production rules have been used to model structural parameters in RNA folding. Such folding parameters are modeled by production rules that produce two characters simultaneously. The resulting production rules represent a palindromic language. For example, we could extend our non-terminals to include production rules such as ${}^{a}_{b}T^{a'}_{b'}$ where the notation implies a basepairs with a' in P and b basepairs with b' in S. A simple, but effective grammar for RNA structure prediction was proposed by Knudsen and Hein in the PFold package 17:

$$\begin{aligned} G_2: \quad & S \rightarrow LS \;|\; L \\ & L \rightarrow {}^{b}_{a}F^{b'}_{a'} \;|\; {}^{-}_{a} \\ & F \rightarrow {}^{b}_{a}F^{b'}_{a'} \;|\; LS \end{aligned}$$

Additional production rules allow models to more closely represent biological function. They also increase computational complexity, sometimes so much so that realistic models on large data sets can not be computed in reasonable time even on a large cluster. Production rules that allow for non-regularity (i.e. both-sides emission) make database search intractable in practice without filters. To circumvent this problem, statistical techniques are used to infer where non-terminal operations can be applied. Traditionally, this approach has been to find some statistical properties of a dataset given a grammar G and then restrict the search based on those properties. In this work, we wish to show a method for compiling evidence that can restrict the number of non-regular production rules to areas of greatest interest and therefore manage tractability through a multi-level grammar strategy. In other words, given our grammar G that is difficult to compute, we wish to run a grammar G' that approximates G to some threshold τ. Given those approximations, we then wish to bind the search of non-terminal operations that exist in G to regions generated in G' that have the most evidence to support a non-terminal short-cut.

2.1. Transcript Evidence

Each of the production rules has an associated cost. To calculate the cost of a production transcript, we sum the costs of all production rules in that production transcript. Traditionally, alignment imposes few limitations on the costs chosen for each of the production rules. Drastically different alignment summaries can be obtained from different scoring schemes [18]. To combine multiple production transcripts in a logically consistent manner, we use a Bayesian method for scoring production transcripts. Imagine we draw production rules from an urn at random to construct our production transcript. At each draw, we are constrained by the pattern and the sequence of the production rule we choose because P must produce S. If H_i is the hypothesis that production operation exists in the transcript at position i, and X_i is our prior information about all other possible production operations at position i (a position specific scoring matrix), then we can relate our hypotheses by the inversion formula:

$$P(H_i|P,S,X_i) = \frac{P(H_i|X_i)P(S,P|H_i,X_i)}{P(S,P|X_i)} \qquad (1)$$

Axiomatically, if H_i' represents the hypothesis that any production operation other than H_i exists at position i in the transcript, then we can construct an identical equation for $P(H_i'|P,S,X_i)$. If we take the log of the ratio of $P(H_i|P,S,X_i)$ and $P(H_i'|P,S,X_i)$ we can obtain the evidence:

$$e(H_i|P,S,X_i) = e(H_i|X_i) + 10\ log_{10}\frac{P(P,S|H_i,X_i)}{P(P,S|H_i',X_i)} \qquad (2)$$

In equation 2, $e(H_i|X_i)$ represents our prior evidence in production rule H at position i based on our grammar production model. If this to zero, it indicates that we have no evidence supporting or refuting H. The evidence for an entire production transcript T can be calculated as:

$$e(T|P,S,X) = \sum_i e(H_i|P,S,X_i) \qquad (3)$$

The evidence in a production transcript depends only on our prior knowledge stated explicitly in X. Given N represents all possible production operations at position i in the transcript, SM is the substitution matrix based on sampling of production rules from ncRNA, and W represents the current production operation, we have a general formula for evaluating evidence of a production rule versus all other production rules at the same position in the transcript:

$$e(T|P,S,X) = \sum_i e(H_i|P,S,X_i) \qquad (4)$$

At this point, we can integrate our chemical probing data or other biological evidence using the term $e(H_i = W|X_i = SM)$.

2.2. Scoring a Grammar

The objective now is to find a maximum production list amongst all possible production lists.

Definition 2.1. A maximum weighted grammar production list (MWGPL) is an ordered list of all allowable grammar production steps and their associated evidence such that the production list: (1) produces S from P when the list of production rules are taken in order and (2) has maximum total evidence over all paths that produce S from P.

If the MWGPL is known, then we can make a statement about how well the pattern and the sequence correspond to the model. A great deal of evidence is likely to imply that the model, the sequence and the pattern agree and that the sequence has the same characteristics as the pattern.

This leads us to the three key considerations that are the subject of this work: (1) Find a partition of S such that our algorithms can be run efficiently in practice with minimum loss to the quality of a query, (2) Minimize the use of non-terminals but retain the benefits of non-terminal operations, and (3) Integrate relationships in our data into the grammar model.

2.3. Optimization Through Nesting Grammars

To manage tractability of the evaluation polynomial, we would like to be able to partition S recursively as the query collects evidence towards the most likely propositions. To obtain some guarantee about the running time of our partition, we also want to select production operations that are consistent with run-time expectations. To do this, we define the notion of a topology element.

Definition 2.2. A topology element $TE = \{PS_c, R\}$ is a grammar production rule set that contains a collection of patterns $PS_c = \{ps_1, ps_2, \dots\}$ and a set of allowed production rules R. A topology must use the set of production rules R to produce S', a subsequence of S. Each topology element must have one prior associated with all of the production rules in R. In a grammar G a topology element may be used only once. A topology element is also a grammar.

To evaluate the evidence that topology element TE produced S' we use Equation 3 selecting ps_1 from PS_c and selecting S' in S such that evidence is maximized. As each topology element may have more than one string, the production of S from the topology element series is constructed by picking the minimum weighted grammar production list over all topology alternatives in TE. Topology elements are produced through a grammar. The global grammar G contains productions for the topology grammar G'. A heuristic grammar HG for G approximates G by using production rules in G and production rules from a simpler grammar G'.

The topology element paradigm allows us to manage complexity by recursively defining partitions on S. Consequently, it allows us to restrict complexity by bounding non-terminal production rules to regions specified by the partition. A topology element serves as a heuristic to cut vertexes in the production transcript graph so that the graph may be evaluated using divide and conquer (for a description of the relationship between edit transcripts and edit graphs and how grammar production rules can be represented as both a graph and a sequence of rules see 16, 12). The topology element serves as evidence towards G with G'. In practice, we want to evaluate all topology elements that satisfy an evidence threshold τ. If τ is large, the approximation for G will be inaccurate because G' will miss many candidates although the runtime will be favorable. If τ is small, the likelihood that we miss a production transcript lessens, but the computational cost of evaluating HG increases.

A topology element TE is evaluated with grammar G'. Evaluating TE with G' inevitably reduces the correctness of the MWGPL for G because some production rules in G do not exist in G'. There are two approaches to solving this problem: (1) Allow grammar refinement, or (2) assume that higher order relationships can be approximated, given enough alternatives in the data. Grammar refinement is a strategy where we may reexamine the production list for topology element TE produced by G' and substitute production rules that exist in G (but not in G') into the production list for TE. Using a grammar refinement strategy, we can guarantee that the MWGPL for HG has the same evidence as the MWGPL for G. The disadvantage of this approach is that in the worst case we will actually evaluate G. While there are many potential approaches to bound the number of refinements, we choose to save this for future work. Instead, we choose to focus on a data-driven approach. We assume that a relationship found in a higher order grammar production can be discovered if we have enough supporting sequences in our model. As the number of sequences increases, the known alternatives for a topology element approaches the real number of alternatives in the database.

As an example, consider sequence A =`tgtCCCaTATAaGGGata` that we know can be partitioned into 3 consensus regions. TE_1 =`tata`, TE_2 =`ccc`, and TE_3 =`ggg`. TE_2 and TE_3 are related because they complementary base pair. Here is an example topology element based grammar for A:

$$\begin{aligned}
GT_1: \quad A &\rightarrow {}^{a}_{a}\,A \;\mid\; {}^{b}_{a}\,A \;\mid\; {}^{a}_{-}\,A \;\mid \\
&\quad {}^{-}_{a}\,A \;\mid\; TE_1\,B \\
B &\rightarrow {}^{a}_{a}\,B \;\mid\; {}^{b}_{a}\,B \;\mid\; {}^{a}_{-}\,B \;\mid \\
&\quad {}^{-}_{a}\,B \;\mid\; TE_2\,C\,TE_3 \\
C &\rightarrow {}^{a}_{a}\,C \;\mid\; {}^{b}_{a}\,C \;\mid\; {}^{a}_{-}\,C \;\mid\; {}^{-}_{a}\,C \;\mid \epsilon
\end{aligned}$$

In GT_1, insertion of TE_1, TE_2, and TE_3 is done via calls to a simpler grammar GT_1'. Insertion may allow grammar substitution operations that increase evidence based on known topology relationships. For example, in the above grammar TE_2 and TE_3 are known to complementary basepair so regular grammar productions $\{M \rightarrow {}^{c}_{c}M, M \rightarrow {}^{c}_{c}M, M \rightarrow {}^{c}_{c}M\}$ on the MWGPL of TE_2 and the regular grammar productions $\{M \rightarrow {}^{g}_{g}M, M \rightarrow {}^{g}_{g}M, M \rightarrow {}^{g}_{g}M\}$ on the MWGPL of TE_3 can be substituted with palindromic grammar productions representing complementary base pairs in GT_1: $\{M \rightarrow {}^{c}_{c}M^{g}_{g}, M \rightarrow {}^{c}_{c}M^{g}_{g}, M \rightarrow {}^{c}_{c}M^{g}_{g}\}$. Using this framework, we can pursue likely complementary base pairs without being forced to evaluate all possible complementary base pairs.

2.4. Non-terminal Grammar Operations

The goal of this section is to show how one grammar can be used to approximate another grammar. Consider the following grammar, G_3, that produces a local alignment:

$$
\begin{array}{lrl}
G_3: & L \rightarrow & {}^{-}_{a}\,L \;\mid\; A \\
 & A \rightarrow & {}^{a}_{a}\,A \;\mid\; {}^{b}_{a}\,A \;\mid\; {}^{a}_{-}\,A \;\mid\; {}^{-}_{a}\,A \;\mid\; \epsilon \\
 & R \rightarrow & {}^{a}_{-}\,R \;\mid\; {}^{-}_{a}\,R \;\mid\; \epsilon
\end{array}
$$

In this grammar, L and R are production rules that result in no evidence; we wish a local alignment. Production rule A is where G_3 collects evidence. If we use dynamic programming to build all maximal solutions, A requires $O(|P| \times |S|)$ to evaluate the MWGPL. The bottleneck comes from the fact that at each position in the production list, we have three choices: we may advance our position in S but not P, or in P but not S, or in both P and S. Consider a prototype grammar, G_4 that we wish to use to approximate G_3:

$$
\begin{array}{lrl}
G_4: & L \rightarrow & {}^{a}_{-}\,L \;\mid\; {}^{-}_{a}\,L \;\mid\; A \\
 & A \rightarrow & {}^{a}_{a}\,A \;\mid\; {}^{b}_{a}\,A \;\mid\; \epsilon \\
 & R \rightarrow & {}^{a}_{-}\,R \;\mid\; {}^{-}_{a}\,R \;\mid\; \epsilon
\end{array}
$$

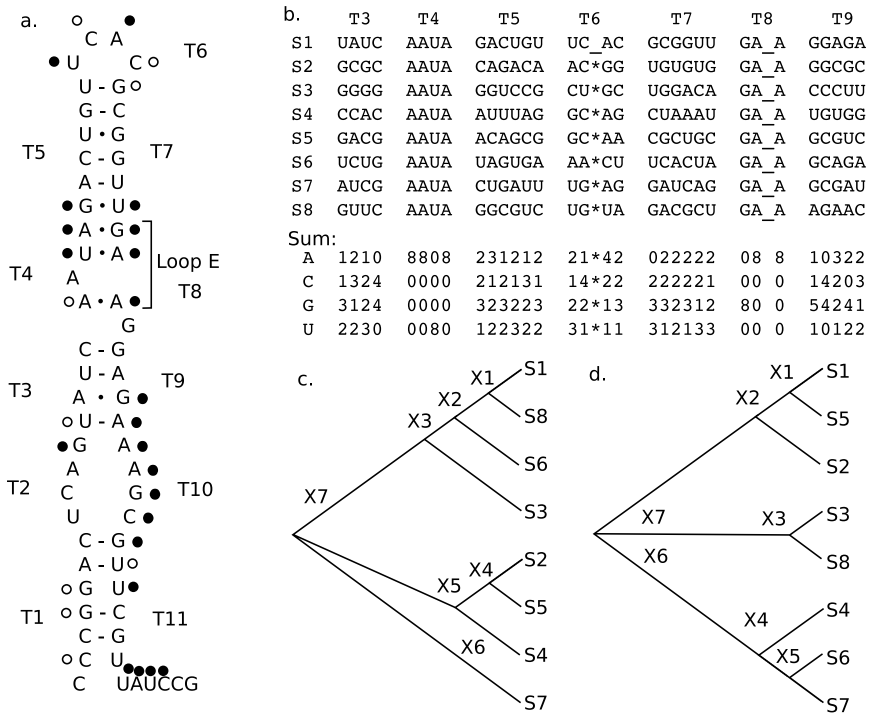
Given an evidence cutoff τ, we can store all possible production transcripts with evidence over τ in $O(|P| + |S|)$. Note that G_4 only need a starting position and an ending position. Given these two positions, the evidence transcript is unique. This grammar can be evaluated in $O(|P| + |S|)$. We can further increase speed by hashing all possible production transcripts resulting in a score of at least τ and index all instances that actually appear in producing S from P. To produce G_3 with the transcript T from G_4 we could use the following grammar G_5:

$$
\begin{array}{lrl}
G_5: & {}^{a}_{a}\,T \rightarrow & {}^{a}_{a}\,T \;\mid\; {}^{-}_{a}\,R \;\mid\; {}^{a}_{-}\,L \;\mid\; \epsilon \\
 & {}^{b}_{a}\,T \rightarrow & {}^{b}_{-}\,L \;\mid\; {}^{-}_{a}\,R \;\mid\; {}^{b}_{a}\,T \;\mid\; \epsilon \\
 & L \rightarrow & {}^{b}_{-}\,L \;\mid\; T \\
 & R \rightarrow & {}^{-}_{a}\,R \;\mid\; T
\end{array}
$$

G_5 functions to stitch elements with large amounts of supporting evidence from G_4 and construct our approximation for G_3. For each grammar alignment in G_4 with evidence over τ, we construct the table for the grammar production list. As an example, consider a list lst of non-overlapping components that produce $S\ lst = (A, B, C \ldots)$ via G_4. To stitch A, B, C together using G_5, we first make all the grammar productions in A, thus constructing the final row of the dynamic programming table from A. The final row in the table is then used as we make grammar rule productions in G_5 until we get to the position in S_i where i is the first position in B. We then make all production rules in B and again use the last row from B to continue making grammar productions using G_5. This process continues until S is produced. If τ is large, then most of the computational time will be spent evaluating G_5 instead of G_4 and the computational time will approach G_3. If τ is small, then computing time is dominated by G_4 and our approximation algorithm will be nearly linear.

2.5. Integrating Relationships in Data into Search

A topology element allows us to produce sequences instead of characters. Assume that we have a regular grammar that produces the hairpin in figure 2a. The grammar is divided into topology elements $T1-T11$. Figure 2b shows several example sequences that all contain the same hairpin. At the base of figure 2b are the totals of the number of bases at each position. $T4$ and $T8$ are highly conserved and easily detected. However, the signal for $T4$ and $T8$ has very little information content. We would like to use the surrounding elements to increase (decrease) the evidence that we have supporting $T4$ and $T8$ as a real site. Simple graphical models such as an HMM will not be able to detect the site $T4 + T8$ because



	T3	T4	T5	T6	T7	T8	T9
S1	UAUC	AAUA	GACUGU	UC_AC	GCGGUU	GA_A	GGAGA
S2	GCGC	AAUA	CAGACA	AC*GG	UGUGUG	GA_A	GGCGC
S3	GGGG	AAUA	GGUCCG	CU*GC	UGGACA	GA_A	CCCUU
S4	CCAC	AAUA	AUUUAG	GC*AG	CUAAAU	GA_A	UGUGG
S5	GACG	AAUA	ACAGCG	GC*AA	CGCUGC	GA_A	GCGUC
S6	UCUG	AAUA	UAGUGA	AA*CU	UCACUA	GA_A	GCAGA
S7	AUCG	AAUA	CUGAUU	UG*AG	GAUCAG	GA_A	GCGAU
S8	GUUC	AAUA	GGCGUC	UG*UA	GACGCU	GA_A	AGAAC
Sum:							
A	1210	8808	231212	21*42	022222	08 8	10322
C	1324	0000	212131	14*22	222221	00 0	14203
G	3124	0000	323223	22*13	332312	80 0	54241
U	2230	0080	122322	31*11	312133	00 0	10122

Fig. 2. a) A hairpin structure containing the loop E motif from coxsackievirus B3 b) An illustrative example of sequences from the characteristic portions of the loop E motif c) A phylogenetic tree constructed from sequences S1-S8 d) A phylogenetic tree constructed from T5 + T7. Note that we chose the partitions because of our chemical probing data

the evidence found in the other sites is lost when you consider only the previous base. On the other hand, SCFGs will catch the base pairing relationships between elements $T3$, $T9$ and $T5$, $T7$ but they must check every possible base pair in the sequences to find the relationship. We would prefer a method that can detect the base pairing, but is not forced to evaluate all possible pair alignments. Our approach is to use the phylogenetic relationships found in the data to add evidence toward base pairing. To construct our prior belief in the sequence relationships, we cluster all of the known sequences by constructing a phylogenetic tree as shown in figure 2c. Then, to represent the complementary base pairs, we concatenate $T5$ and $T7$ and construct a tree. The evidence that a relationship exists between S_i and S_j is the distance between S_i and S_j in the tree shown in figure 2c minus the distance between S_i and S_j in tree shown in figure 2d. This is evaluated in the grammar when the term $e(H_i|X_i)$ of Equation 2 sums evidence for topology elements.

3. ALGORITHM

To score a sequence, first we introduce a global grammar G for evaluating sequences in the database D. For example, to evaluate the structure of the hairpin in Figure 2 we can use the following grammar G:

$$
\begin{aligned}
G: \quad & D \rightarrow \genfrac{}{}{0pt}{}{a}{a} D \;\mid\; \genfrac{}{}{0pt}{}{-}{a} D \;\mid\; \genfrac{}{}{0pt}{}{a}{-} D \\
& E \rightarrow \genfrac{}{}{0pt}{}{a}{a} E \;\mid\; \genfrac{}{}{0pt}{}{-}{a} E \;\mid\; \genfrac{}{}{0pt}{}{a}{-} E \;\mid\; \epsilon \\
& P1 \rightarrow D \;\mid\; T3 \quad P2 \quad T9 \quad E \\
& P2 \rightarrow T4 \quad P3 \quad T8 \\
& P3 \rightarrow T5 \quad T6 \quad T7
\end{aligned}
$$

Our algorithm contains two phases, top down and bottom up. Top down refers to the phase where we traverse productions in G generating instances of G' that will produce S. Bottom up is the procedure where instances of G' are stitched together with operations in G into a production list that produces S. The transcript with the maximum evidence is selected as the approximation for the MWGPL of G. In this example, we assume the grammar for evaluating $T3, T5, T6, T8$ and $T9$ are instances of the regular grammar G_3. While we could choose to evaluate G_3 with an approximation, as was done in the previous section, the overhead in RNA structure matching comes from the palindromic non-terminals. Because the partitions of $S1 - S8$ in our example are such short sequences, evaluating G_3 directly using the Gotoh 19 memory reduction method has better alignments for small sequences. Because the sequences are short, the dynamic programming tables are also short and the overhead is low. $T4$ and $T8$ can be evaluated with $G' = G_4$ because they are absolutely conserved.

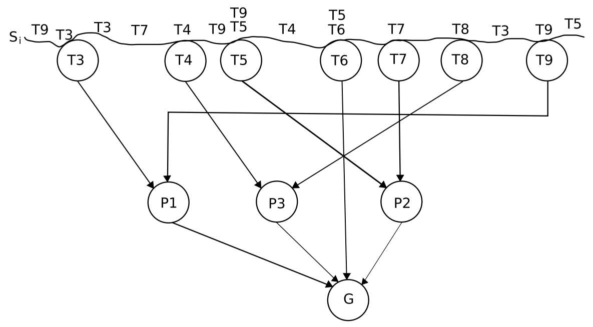


Fig. 3. A grammar for finding the loop E RNA motif.

The top down algorithm for constructing G proceeds forward by first producing $P1$. $P1$ in turn produces $T3$, $P2$, and $T9$. Candidates for the MWGPL for G over $T3$, and $T9$ are computed via G_3. All non-overlapping candidates that satisfy the condition that $T3_k < T9_l$ and that the evidence for $e(T3_k) + e(T9_l) > \tau$ are stored in a table ($table_1$). G then proceeds by producing $P2$ constrained such that $T3_k < T4_m < T8_n < T9_l$ and evidence for the transcript greater than τ. Because we do not know which entries in $table_1$ exist in the MWGPL for G, we must store all potential candidates for $T4$ and $T8$ that exist between $S_{k+|T3|+i}, min(k)$ and $S_l, max(l)$ in $table_2$. Variables k and l represent an index in $0 - |S|$. Note that a potential candidate for a topology element such as $T4$ may not overlap with another candidate for the production list of $T4$, but it may overlap with a candidate from any other topology element (e.g. $T5$). In a similar way, G then produces $T5$, $T6$, and $T7$ and the non-self-overlapping production lists over τ are stored in tables. Once tables are created for all elements of G, we construct the MWGPL approximation for G by stitching candidate topology production lists together if they contain more evidence after being merged. Intuitively, the tables mark candidate positions for G that may exist on the MWGPL. Figure 3 illustrates this basic idea. On S, we have putative positions marked by the forward production operations on the grammar G.

```
score(S,t):
 l = {}
 productionTranscript = 0
 while(productionTranscript.hasMoreWaysToMakeS()):
   productionTranscript += makeProduction(G,S,t,l)
   if (productionTranscript.produces(S) == True):
    l += productionTranscript
    productionTranscript = 0
 forAll i in l:
   GX.computeEvidence(i)
 if(i.isMaxEvidence()) return i
makeProduction(G,S,t,l):
 SelectNextRule = FSA.DP(l)
 if(productionSet.contains(TE):
   evaluateAndMark(TE) forAll TE  > t
 BayesNet[l.index].add(productionRule)
 return productionRule
```

4. RESULTS

Nondetermanistic structural motif finding is one of the most outstanding problems in bioinformatics. The proposed method, advanced grammar alignment search tool (AGAST) can be applied to find motifs in any biological sequences including DNA/RNA/Protein. In this section, we assess the performance of the proposed method in finding loop E motifs in conserved secondary RNA structures.

The loop E motif is a fold that organizes structure in hairpin loops and multi-helix junctions in many RNA molecules. The motif is prevalent in 16S and 23S ribosomal RNA and derives its name from its discovery in loop E of 5S rRNA 20. The loop E motif is particularly significant in RNA structure because it uses a series of non-canonical base pairs to form a characteristic fold. This fold widens the major groove of the RNA helix and presents a cross-strand adenosine stack that serves as a recognition feature for RNA-protein and RNA-RNA interactions 21. The presence of this motif in molecules as diverse as ribosomal RNA, potato tuber spindle viroid, RNase P RNA, and the hairpin ribozyme, proves that the loop E motif is an important feature in RNA structure and function.

Sequence comparison and chemical probing analysis has revealed a consensus pattern for the loop E motif. This pattern consists of a parallel purine-

purine pair (usually AA), a bulged nucleotide, a non-Watson-Crick UA that is absolutely conserved, and a purine-puring pair (AA or AG, but not GA). As a result of the non-canonical pairing, the motif generates a signature pattern of susceptibilities in chemical probing experiments 22. We have identified the sequence pattern and the chemical probing pattern of the loop E motif in the coxsackievirus B3 (CVB3) genomic RNA. The general character of the absolutely conserved properties of this motif were `a.ua.*gaa`. The CVB3 loop E motif in the context of the surrounding RNA is shown in Figure 2.

In this section, we compare the performance of our proposed method, AGAST, with RSEARCH and RNAMotif in finding the loop E motif. We selected RNAMotif because of the pattern-matching tools, it is one of the most flexible and can be customized to our specific problem. We selected RSEARCH because it is guarenteed to give optimal results over other methods because it computes all possible sequence-structure alignments. We did not use MilPat because its current release does not allow errors and thus is more constrained than both of these programs.

The structure of the loop E motif must correspond exactly to the character of the sequence shown figure 2. The character of the motif is a hairpin loop with the loop E sequence immediately flanked by paired regions. The loop E section must be absolutely conserved (with motif `a.ua.*[ga]aa`). Complementary base-pairing flanking the loop E is responsible for maintaining the structure of the loop. The turn at the top of the loop may contain a large secondary structure (instead of a 4 base turn).

To test the sensitivity and specificity of our approach, we collected a set of sequences from coxsackievirus B3 (CVB3) genomic RNA. We partitioned these sequences into two groups, testing and modeling. With the modeling sequences, we constructed a multiple sequence alignment as shown above by overlaying our chemical probing data, phylogenetic conservation and possible folding conformations from mfold. Then, for each sequence in the testing set, we constructed a false positive sequence using a third order Markov chain (to preserve the motif, but destroy complementary base pairing required for secondary structure). For each of the sequences in the dataset we ran RSEARCH, RNAmotif, and AGAST.

In the case of RNAmotif, we designed two pattern-matching queries using the same information that we had in constructing the AGAST query. The first query which we call RNAmotif-intuitive, constrains results such that they form a hairpin around the conserved loop E motif and that complementary base pairs exist in the hairpin both 5' and 3' of the motif. This query is based on our understanding that the motif can only be formed if there is significant stability provided by complementary base pairs both 5' and 3' of the motif. In our second RNAmotif query, which we call RNAmotif-permissive, we give RNAmotif the 5' and 3' regions surrounding the loop E motif. Because this query did not match any sequences in our test database, we gradually increased the error threshold in the regions 5' and 3' of the motif until we obtained matches. RSEARCH was provided only with the sequence from the hairpin, that it uses to make a grammar. Each of these grammars was queried against our database. The results from this experiment are in Table 1.

These results indicate that the traditional methods of SCFGs (RSEARCH) and expertly tuned queries (RNAMotif-permissive) remain the most sensitive methodologies when searching for a double stranded RNA motif in a two-dimensional structure. However, this sensitivity comes at the cost of an increased number of false predictions. In sequences that have no conserved two-dimensional structure (HMM-3 Jumbled sequences), we found the false positive rates to be 1.22 and 1.29 for RSEARCH and RNAMotif-permissive respectively. This is because both programs predicted more sites than there exist sequences in the generated database. The RNAMotif-intuitive query was able to substantially reduce the number of false predictions, but it was far too restrictive, eliminating 85% of true positives. We believe that our approach has significant promise because it was capable of maintaining relatively high sensitivity while increasing specificity to the same level as our intuitive description of the motif. Moreover, upon closer investigation, we realized that the false positives found in our real database where all phylogentically diverse from those instances we had in our training set (all forming in different clades from our training sequences). This indicates that our approach may perform better with a representative sequence from each clade in the phylogenetic tree, but finding such representatives in a new domain remains a challenging problem. Among the

Tool	Time	True Positives	False Positives	Sensitivity	Specificity
RSEARCH	963m53.2s	55	165	0.98	0.56
RNAMotif-intuitive	89.65s	8	6	0.15	0.98
RNAMotif-permissive	15.2s	55	83	0.98	0.78
AGAST	68.36s	50	7	0.91	0.98

other methods, none produce acceptable values for both sensitivity and specificity in our problem domain. On the other hand AGAST had over 90% for both parameters.

Another experiment was conducted to search a larger dataset to find additional unknown instances of the loop E motif. Because ribosomal RNA sequences are known to contain the loop E, we generated a data set of all ribosomal RNA by parsing species specific data files (e.g. gbpri1) from Genbank release 143 for all files with ribosomal RNA. We found 176,371 rRNA records using this method. We shuffled each of the sequences in the database using a Markov chain of order three and ran our algorithm on both ribosomal RNA sequences. Figure 4 shows the distribution of scores for records from the two sets.

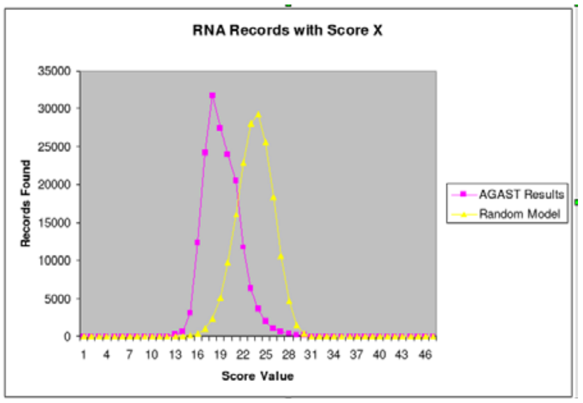


Fig. 4. Grammar scores ($max\{evidence\} - evidence\ found$) versus number of records for finding the loop E RNA motif.

In this experiment, there is a significant difference between sequences generated by the Markov chain that contain the sequence `a.ua.*[ga]aa`, and rRNA sequences. Also, of the top 10 records returned in our search we were able to verify that 9 of the records did contain the loop E motif and the other records are unknown. We are currently working to verify if the final sequence also contain the loop E motif. This demonstrates that our tool can find motif in structure homology even in a large database.

5. CONCLUSIONS

In this paper, we have proposed a grammar based method based on constructing graphical models that relate subsequences instead of forcing the evaluation of individual characters. We have used this method to find the loop E structural motif inside of ncRNA with conserved secondary structure. Our results show that our method produced the best sensitivity/specificity combination among the tested methods for the problem domain. It may also serve as a strong complement to current methods in accelerating ncRNA homology detection because it can be more specific than SCFGs in the case where we have additional information about interior structural motifs. We believe that well structured data relationships can play a key roll in difficult problems such as motif searching. We also believe topology models are very general and could be used in modeling and searching for complex patterns in DNA or proteins. We believe that this work points to the need of more general approaches to automatically generate RNA database queries; especially queries where some possible structures can be eliminated from the SCFG on the basis of biological evidence. Our method would serve well for building filters that can be combined with existing methods such as FastR for increased specificity in selecting structures from the SCFG with conserved structural motifs.

ACKNOWLEDGMENTS

We would like to thank Mohammad Shafiullah for help on scaling the code to large architectures, and Laura A. Quest/Brad Friedman/Mark Pauley for help with the manuscript. This research project was made possible by the NSF grant number EPS-0091900 and the NIH grant number P20 RR16469 from the INBRE Program of the National Center for Research Resources.

References

1. Klein RJ, Eddy SR. RSEARCH: nding homologs of single structured RNA sequences. *BMC Bioinformatics*, September 2003; **vol. 4**.
2. Weinberg Z, Ruzzo WL. Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. *Bioinformatics*, August 2004; **vol. 20 Suppl 1**.
3. Zhang S, Haas B, Eskin, E, and Bafna V. Searching genomes for noncoding RNA using FastR. *IEEE/ACM Trans Comput Biol Bioinform*, 2005; **vol. 2, no. 4** 366–379.
4. Rivas E, Klein RJ, Jones TA, and Eddy SR. Computational identication of noncoding RNAs in E. coli by comparative genomics. *Curr Biol*, September 2001; **vol. 11, no. 17**: 1369–1373.
5. Zuker M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res*, July 2003; **vol 31, no. 13**: 3406–3415.
6. Lowe TM, Eddy SR. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res*, March 1997; **vol. 25, no. 5**: 955–964.
7. Fichant GA, Burks C. Identifying potential tRNA genes in genomic DNA sequences. *J Mol Biol*, August 1991; **vol. 220, no. 3**: 659–671.
8. Eddy SR. RNABob: A program to search for RNA secondary structure motifs in sequence databases. http://selab.wustl.edu/cgi-bin/selab.pl?mode=software
9. Laferrire A, Gautheret D, and Cedergren R. An RNA pattern matching program with enhanced performance and portability. *Comput Appl Biosci*, April 1994; **vol. 10, no. 2**: 211–212.
10. Billoud B, Kontic M, and Viari A. Palingol: a declarative programming language to describe nucleic acids secondary structures and to scan sequence database. *Nucleic Acids Res*, April 1996; **vol. 24, no. 8**: 1395–1403.
11. Macke TJ, Ecker DJ, Gutell RR, Gautheret D, Case DA, Sampath R. Rnamotif, an RNA secondary structure denition and search algorithm. *Nucleic Acids Res*, November 2001; **vol. 29, no. 22**: 4724–4735.
12. Durbin R, Eddy S, Krogh A, Mitchison G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids* Cambridge University Press, London. 1998.
13. Rivas E, and Eddy SR. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J Mol Biol*, February 1999; **vol. 285, no. 5**: 2053–2068.
14. Zhang S, Borovok I, Aharonowitz Y, Sharan R, Bafna V. A sequence-based ltering method for ncRNA identication and its application to searching for riboswitch elements. *Bioinformatics*, July 2006; **vol. 22, no. 14**.
15. Thbault P, de Givry S, Schiex T, Gaspin C. Searching RNA motifs and their intermolecular contacts with constraint networks. *Bioinformatics*, July 2006.
16. Guseld D. *Algorithms on strings trees and sequences.* Cambridge University Press, London 1999.
17. Knudsen B, Hein J. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res*, July 2003; **vol. 31, no. 13** 3423–3428.
18. Dewey CN, Huggins PM, Woods K, Sturmfels B, Pachter L. Parametric alignment of drosophila genomes. *PLoS Computational Biology*, June 2006; **vol. 2, no. 6**: e73+.
19. Gotoh O. An improved algorithm for matching biological sequences. *J Mol Biol*, December 1982; **vol. 162, no. 3**: 705–708.
20. Branch AD, Benenfeld BJ, and Robertson HD. Ultraviolet light-induced crosslinking reveals a unique region of local tertiary structure in potato spindle tuber viroid and HeLa 5S RNA.*PNAS*, October 1985; **vol. 82, no. 19**: 6590–6594.
21. Correll CC, Wool IG, and Munishkin A. The two faces of the Escherichia coli 23 S rRNA sarcin/ricin domain: the structure at 1.11 a resolution. *J Mol Biol*, September 1999; **vol. 292, no. 2**: 275–287.
22. Leontis NB, Westhof E. A common motif organizes the structure of multi-helix loops in 16 s and 23 s ribosomal RNAs. *J Mol Biol*, October 1998; **vol. 283, no. 3**: 571–583.