A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences

Yongmei Ji, Xing Xu and Gary D. Stormo*

Department of Genetics, Washington University, School of Medicine, St. Louis, MO 63110, USA

Received on September 9, 2003; revised on December 30, 2003; accepted on December 31, 2003
Advance Access publication February 12, 2004

ABSTRACT

Motivation: RNA structure motifs contained in mRNAs have been found to play important roles in regulating gene expression. However, identification of novel RNA regulatory motifs using computational methods has not been widely explored. Effective tools for predicting novel RNA regulatory motifs based on genomic sequences are needed.

Results: We present a new method for predicting common RNA secondary structure motifs in a set of functionally or evolutionarily related RNA sequences. This method is based on comparison of stems (palindromic helices) between sequences and is implemented by applying graph-theoretical approaches. It first finds all possible stable stems in each sequence and compares stems pairwise between sequences by some defined features to find stems conserved across any two sequences. Then by applying a maximum clique finding algorithm, it finds all significant stems conserved across at least k sequences. Finally, it assembles in topological order all possible compatible conserved stems shared by at least k sequences and reports a number of the best assembled stem sets as the best candidate common structure motifs. This method does not require prior structural alignment of the sequences and is able to detect pseudoknot structures.

We have tested this approach on some RNA sequences with known secondary structures, in which it is capable of detecting the real structures completely or partially correctly and outperforms other existing programs for similar purposes.

Availability: The algorithm has been implemented in C++ in a program called comRNA, which is available at http://ural.wustl.edu/softwares.html

Contact: stormo@genetics.wustl.edu

INTRODUCTION

A large number of mRNA structure motifs have been found to play important roles in the transcriptional and post-transcriptional regulation of gene expression, including transcription termination, mRNA localization and stability, mRNA alternative splicing and translation efficiency (Pesole et al., 2001; Stormo and Ji, 2001; Mandal et al., 2003). With the accumulation of sequence data accompanying genome sequencing projects, numerous novel mRNA structure motifs potentially involved in gene regulation are to be identified. Although some computational efforts have been made to identify new RNA motifs similar to known ones (Laferriere et al., 1994; Billoud et al., 1996; Pesole et al., 2000; Macke et al., 2001), the prediction of novel RNA regulatory motifs is much more difficult. Effective automatic tools for finding novel RNA structure motifs are needed.

Current RNA secondary structure prediction methods mainly employ free energy minimization for a single sequence, as exemplified by the dynamic programming algorithms (Zuker, 1994; Rivas and Eddy, 1999; Dirks and Pierce, 2003), and phylogenetic comparisons for common structures shared by multiple related RNAs. It has been widely accepted that the latter is more reliable.

A number of RNA secondary structure prediction methods based on phylogenetic comparisons have been developed. Sankoff designed a dynamic programming algorithm that can simultaneously fold, align and build the phylogenetic tree on multiple RNAs, but it is computationally intractable for more than a couple of sequences (Sankoff, 1985; Mathews and Turner, 1999). Sankoff and Pierce, 2003, and phylogenetic comparisons for common structures shared by multiple related RNAs. It has been widely accepted that the latter is more reliable.

A graph-theoretical approach, maximum weighted matching, has been introduced to predict common RNA secondary structures including pseudoknots...
Fig. 1. Overall scheme for searching common RNA secondary structure motifs in \( n \) (four in this example) sequences. (a) Find in each sequence all possible stable stems. (b) Construct an \( n \)-partite undirected weighted graph of stems. Stems are compared pairwise between sequences and those stem pairs whose similarity scores are higher than a certain threshold are connected. (c) Find in the graph all maximum cliques of size \( \geq k \), each maximum clique representing a potential conserved stem shared by the \( \geq k \) sequences. Lines of the same colour form a maximum clique. (d) The conserved stems obtained from (c) are illustrated in the primary sequences. (e) Construct a directed graph of conserved stem blocks. Stem block \( b_1 \) is connected to \( b_2 \) if \( b_1 \) is before, compatible and forms a consistent structure pattern with \( b_2 \) in a critical number of sequences. (f) Find in the graph all maximum paths of stem blocks, in which a downstream stem block should be connected by all upstream stem blocks. Each path corresponds to a common structure shared by at least \( k \) sequences. A number (two here) of highest-scoring common structures are reported.

(Cary and Stormo, 1995; Tabaska et al., 1998). It also requires reliable structural alignment.

Recently, a few new approaches based on stem comparisons have been introduced. One approach presents binary representations of the stem pattern in a structure and simplifies the common RNA structure prediction problem to searching for common binary structure patterns (Bouthinon and Soldano, 1999). This method is fast in dealing with short patterns, but it does not consider sequence similarities at all. Another approach applies a genetic algorithm to recursively evolve and incorporate stems from different sequences to form best common structures (Chen et al., 2000; Hu, 2002). This method is computationally expensive on large data sets. A third approach predicts common RNA structures on two sequences by using dynamic programming to get the best alignment of stems (Perriquet et al., 2003).

In this paper, we present a new graph-theoretical approach to automatic detection of common RNA secondary structure motifs in a group of functionally or evolutionarily related RNA sequences. It finds sets of stable stems conserved across multiple sequences and assembles compatible conserved stems to form consensus secondary structure motifs. Our method does not require the presence of global sequence similarities (but can take advantage of it), does not require prior structural alignment and is able to detect pseudoknot structures.

METHODS

When talking of a RNA secondary structure, we mean the RNA structure formed by stems and loops, where stems represent palindromic helices and loops, single-stranded regions. From this perspective, the common RNA secondary structure prediction problem can be reduced to the problem of finding sets of conserved stems and loops across multiple sequences. Thus, our method, instead of taking the whole individual sequences or single columns of aligned nucleotides as the basic analytic subjects as most of the other methods do, works with stems.

The overall scheme of our method is illustrated in Figure 1. It solves the problem in three major steps:

- Find all possible stable stems in each sequence and compare them pairwise between sequences.
● Find all potential conserved stems shared by subsets of sequences.
● Assemble compatible of conserved stems to construct consensus secondary structure profiles and report a number of the best sets following structure refinement.

Finding all possible stable stems in each sequence
We define a stem as a palindromic helix within one sequence comprising consecutive Watson–Crick (A · U or G · C) or wobble (G · U) base pairs, with a minimum of \( L \) base pairs. \( L \) is a user-specifiable parameter, 4 by default. No internal loop or bulge is allowed in a stem until the final stage of structure refinement. In a primary sequence, a stem is represented by its pairing partners: the 5′ half-stem and 3′ half-stem.

For each sequence under study, we can list all possible stems within it through a branch and bound procedure, as does the program dotPlot (Tinoco et al., 1971). These stems may overlap or be independent of each other, and various combinations of them can form different RNA structures. To reduce the search space, we only focus on stable stems. We evaluate the stability of a stem by its stacking energy according to Turner’s energy parameters (Freier et al., 1986; Walter et al., 1994). A user-definable cutoff \( E \) (\( E < 0 \), default \(-5 \) kcal/mol) is set so that only stems with stacking energy lower than \( E \) are considered stable and saved for further analysis.

Comparing stems across sequences
To uncover the conserved stems, we first compare each pair of sequences and their stems. To take advantage of the information given by sequence conservation, we align two sequences globally using the Needleman–Wunsch (1970) dynamic programming algorithm and find in the alignment the highly conserved regions, which are defined as regions of 10 nt or longer giving at least 80% sequence identity. These conserved regions are used as anchor sequences for stem comparison, as Perriquet et al. (2003) did in their research. As illustrated in Figure 2, we require that two stems from two sequences can be compared only if their corresponding 5′ or 3′ half-stems are from the same context, i.e. the same anchor or non-anchor regions, and their counterparts in the anchor regions should not shift beyond 10 nt. We do not give shifting constraints on half-stems located in the non-anchor regions, which allows certain flexibility for stem conservation across sequences. However, if the similarity between two sequences is low and no anchor regions can be found, then the whole sequence is regarded as a non-anchor region and all stems between the two sequences are compared with each other.

We define a function, \( S(i_x, j_y) \), to measure the similarity between two stems \( i_x \) and \( j_y \) from sequences \( x \) and \( y \), respectively. The similarity between two stems is measured by their similarities in five features: helix length, helix sequence, loop sequence closed by the stem, stem stability and relative positions of the stem start and end sites in the whole sequence.

Finding conserved stems across multiple sequences
We want to find in the sequences all putative conserved stems shared by subsets of sequences that are potentially
part of the common structures. We set a significance level, \( p \) \((0 < p \leq 1)\), which is the minimum percentage of the \( n \) sequences that share a common structure. \( p \) is user-specifiable and is set to 0.5 by default, based on a general assumption that at least half the input sequences should share a common structure. We want to find conserved stems occurring in at least \( k \) \((k = \lceil p \times n \rceil, 1 \leq k \leq n)\) sequences. We achieve this by a graph theoretical approach, as illustrated in Figure 1.

First, we construct an \( n \)-partite undirected weighted connectivity graph to represent the stems and their relations. Each node represents a stem, and the graph is partitioned into \( n \) parts, each part comprising the population of stems from one sequence. Only nodes from different parts can be connected. Those potential equivalent stems from different sequences with a similarity score exceeding the threshold \( S \) are connected and edge-weighted by the similarity score.

To find out which stems are conserved across multiple sequences, we inspect the stems on the basis of multi-way comparisons. If \( k \) stems from \( k \) different sequences are all potential equivalent stems of one another, we will be confident in inferring that they are probably instances of a conserved stem. In graph theory, the all-way connections are represented as a clique, a complete sub-graph in which every node is connected to every other node. The size of a clique is defined as the number of nodes in it. A clique is maximum if it is not contained in any larger clique. Thus, our task of finding all potential conserved stems shared by at least \( k \) sequences becomes finding all maximum cliques of size \( \geq k \) in the \( n \)-partite graph.

Finding maximum cliques is an NP-hard problem, and has been studied extensively, with some optimization and heuristic algorithms having been developed (Johnson and Trick, 1996; Bonze et al., 1999). For the case of finding all \( \geq k \)-size maximum cliques in an \( n \)-partite graph, we are not aware of any effective optimization or heuristic algorithm feasible for our problem, although applications of this technique to solving different problems have been found (Pevzner and Sze, 2000; Grünert et al., 2002). Therefore, we decide to use an enumerative algorithm based on depth-first search (Cormen et al., 1989). It takes as input the \( n \)-partite graph presented in the form of an adjacency list and outputs an array of unique \( \geq k \)-size maximum cliques. Briefly, it starts a clique with each stem of each sequence and expands the clique by incrementally including from a new sequence a stem that is connected to all existing stems in the clique until all sequences have been searched. At every step of clique expansion, it checks if the maximum possible size of the clique being found can reach \( k \). If not, it will skip this clique and start a new one. Each maximum clique associates with a score, which is the average edge weight multiplied by the clique size. This score measures the plausibility of the corresponding conserved stems. The larger the clique size is and the more similar its stems are to each other, the more plausible that the stems are instances of a conserved stem. The pseudocode is given below.

Algorithm for finding all maximum cliques of size \( \geq k \) in an \( n \)-partite weighted graph.

Input: An \( n \)-partite undirected weighted graph \( G \) presented in the form of an adjacency list. Each connected stem associates with a similarity score.

Output: An array of \( \geq k \)-size maximum cliques \( C[ ] \), each clique associating with a conservation score.

\[
\text{FindMaxiumCliques}(G, k) \}
\]

\[
\text{foreach (seq_x in G) \{}
\]

\[
\text{originalClique} = \text{stem}_i;
\]

\[
\text{connectedStemGroups} = \text{GetConnectedStemGroups(stem}_i);\]

\[
\text{ExpandClique (originalClique, 0, connectedStemGroups, k);}
\]

\[
\}
\]

\[
\text{ExpandClique (currentClique, currentScore, connectedStemGroups, k) \{}
\]

\[
\text{foreach (stem_i in seq_x) \{}
\]

\[
\text{newClique} = \text{stem}_i;
\]

\[
\text{newScore} = \text{currentScore} + \text{stem}_i.GetScore();
\]

\[
\text{newClique = GetCommonGroups(connectedStemGroups, connectedStemGroups);}
\]

\[
\text{foreach (stem_j in newConnectedStemGroups) \{}
\]

\[
\text{newClique = stem}_i;\]

\[
\text{newScore} = \text{currentScore} + \text{stem}_i.GetScore();
\]

\[
\text{expandedClique (newClique, newScore, connectedStemGroups, k);}
\]

\[
\}
\]

The worst-case time complexity for this algorithm is \( O(m^n) \), where \( m \) is the maximum number of stems examined in one sequence and \( n \) is the number of total sequences. This complexity makes the algorithm seem impractical, but in practical applications the average run-time is much less than the worst case because the input stem graphs are often sparse due to the introduction of the anchor-region constraints.
and the user-specifiable cutoff parameters, including $L, E, S$ and $p$. Our tests on medium size biological sequences have shown that the algorithm can give reasonably good results in acceptable time (see Results section).

We list the cliques obtained in the order of decreasing score. To reduce the search space for later steps, we scan the list and remove ‘redundant’ cliques. If two cliques share more than 70% of their member stems, the lower-scored clique is regarded as a redundant clique and removed from the list.

**Assembling compatible conserved stems**

Each obtained maximum clique corresponds to a set of similar stems from different sequences, with size ranging from $k$ to $n$. We use the term ‘stem block’ to represent the set of stems corresponding to a maximum clique. The stem blocks may be independent of each other or overlap in different ways in various subsets of sequences, with many possibilities for arranging them to form consensus structures.

Our objective is to find the best assemblies of the stem blocks, which should represent the best candidates for consensus structures. To achieve that, we apply a graph-theoretical technique similar to a topological sort, which has been used to solve similar problems, such as assembling protein blocks for database searching (Henikoff and Henikoff, 1991). The idea is to assemble into a structure as many compatible stem blocks as possible in the order of the stem start sites from 5′ to 3′ of the sequences and to evaluate the plausibility of a structure by the significance of its member stem blocks. What is complex about RNA block assembling is that we need to consider not only the sequential order but also the consistency of structure patterns formed in different sequences.

We first construct a directed graph in which each obtained maximum clique, i.e. stem block, is a node. All stem blocks are compared with each other. Between any two stem blocks, we examine each sequence carrying stems from both blocks for their order and arrangement. As Bouthinon and Soldano (1999) did, we use the following representation method to describe the relationship between two stems. We define that, within one sequence, stem $s_1$ is before stem $s_2$ if the helix start site of $s_1$ precedes that of $s_2$ and they are compatible if they do not overlap considerably in their helix regions. Considering the relative positions of the starts and ends of the stems, the arrangement of two compatible stems can be only one of the three patterns: juxtaposed, embedded or overlapped (pseudoknotted), as shown in Figure 3.

A recursive depth-first-search-like algorithm is then applied to find in the directed graph all maximum paths of stem blocks that occur in at least $k$ common sequences, where each path stands for a unique common structure. To assemble the stem blocks in the order from 5′ to 3′ of the sequences and to make sure the blocks form consistent structure patterns with each other across at least $k$ sequences, it requires that a new stem block cannot be included into a path unless it is connected by all existing stem blocks in the path. Because different edges may include different subsets of sequences, a path terminates when it fails to include all passed conserved stems in at least $k$ sequences. Each path associates with a score, which is the sum of the scores of all stem blocks in the path. This score evaluates the plausibility of the corresponding structure contributed by the number of sequences involved, degree of conservation and number of stem blocks. The paths found are ranked by their plausibility scores and saved in this order.

This algorithm has a worst-case time complexity of $O(m^n)$, $m$ being the number of stem blocks in the graph. However, because the introduced constraint parameters, $c$ and $k$, help keep the density of the graph low, the average time complexity is much less than the worst-case. As we will show in the results, the run-time is acceptable on data sets of medium size and moderate length.

A faster approach to assembling stem blocks is to apply a greedy algorithm, which starts with the highest-scoring stem block and recursively includes the next highest-scoring block compatible with all existing stem blocks in the structure until no more compatible blocks can be added. It has a time complexity of $O(m)$, $m$ being the total number of stem blocks in the graph. This approach is fast, but it only gives one structure and does not guarantee its optimality. In the case that a stem block is incorrectly picked, succeeding blocks may be wrongly included too, which may result in a wrong structure. Nevertheless, our tests on a few data sets have shown that this greedy assembly method can sometimes give good
results. Therefore, it has been implemented in our program as a choice for the user.

**Refining and reporting a number of highest scoring common structures**

Having obtained a list of all potential common structure profiles ranked by their plausibility scores, we scan through the list and choose a number \( n \) (default 10) of the highest-scoring ones as the best candidate common structures to report. To avoid reporting redundant structures, we require that a newly chosen structure must differ from all previously chosen structures by at least one stem block, where a stem block is considered different if at least 50% of its stems are not included in the previous structures. Because of the cutoff constraints and heuristic procedures applied in previous steps, the common structures obtained are probably incomplete, with some conserved stems possibly missed in some sequences. To make up for this, for each common structure to be reported, we define its structure pattern and run the RNA secondary structure searching program RNASAT (Laferriere *et al.*, 1994) to search for those possibly missed stems and include successfully found new stems into the structure. Finally, each structure to be reported is refined, where every stem in every sequence is refolded, within up to 10 nt from the stem edges in both directions or until reaching an edge of another stem, allowing internal loops of size up to 4 and bulges of size up to 2. The refolding is done by applying a dynamic programming algorithm that gives a stem with minimum free energy using Turner’s energy parameters (Freier *et al.*, 1986). The \( n \) refined consensus structures are finally reported as the best candidate common structures.

**RESULTS**

We have implemented the algorithm in a C++ program called comRNA. We tested it on several sets of RNA sequences with known structures, examined the statistical significance of its predictions and compared its performance with those of other existing RNA secondary structure prediction programs. We also explored the effect of parameter settings on prediction performance. All the following tests were performed on a Linux machine with a Pentium III 800 MHz CPU and 1 GB RAM, unless otherwise mentioned. In all tests, the parameters \( L, E \) and \( p \) were set as the default values, \( L = 4, E = -5, p = 0.5 \) and \( S \) was determined separately, as discussed below. The results are summarized in Table 1.

**Bacterial \( \alpha \) operon mRNA leader sequences**

The first data set consists of 10 orthologous bacterial \( \alpha \) operon leader sequences, each containing 94 nt immediately upstream and 40 nt downstream of the first translational start codon of the operon. These sequences are retrieved from 10 bacterial genomes downloaded from NCBI. In *Escherichia coli*, a pseudoknot structure involved in \( \alpha \) operon translational regulation has been experimentally identified in this region of the \( \alpha \) operon mRNA sequences (Spedding and Draper, 1993). The 10 sequences are highly conserved in the coding region but are somewhat divergent in the upstream region. Based on CLUSTALW alignment (same for the other data sets), most of the pairwise sequence identities are in the range 60–100%. Figure 4 shows the best predicted structure of comRNA under the parameter \( S = 0.56 \). It contains five stems, represented by letters \( a, b, c, d, e \) and \( f \), three of which \((b, d, e)\) correspond to the real stems in the published \( E. coli \) pseudoknot structure. It did not predict the fourth real stem but instead occupied the region with an extended stem of \( b \) and an extra stem \( f \). This prediction took 3 s CPU time.

**Bacterial S15 mRNA leader sequences**

The second data set contains orthologous ribosomal protein S15 mRNA leader sequences retrieved from 11 bacterial genomes downloaded from NCBI. The sequences range in size from 172 to 204 nt. Each sequence contains the first 37 nt downstream from the S15 gene start codon and \( \sim 150 \) nt upstream sequences. Most of the pairwise sequence identities are in the range 30–50%. It has been reported that in *E. coli* alternative structures form in this region, one being a stem–loop structure and the other a pseudoknot structure, in response to the absence or presence of excess ribosomal protein S15 in the cell, which signals S15 translation autoregulation (Benard *et al.*, 1996; Serganov *et al.*, 2002). Both structures have three stems, in which two \((a \text{ and } b)\) are common and the third \((c \text{ or } d)\) is different and conflicting. As shown in Figure 5, with the parameter set as \( S = 0.56 \), the first and ninth structures predicted by comRNA are very similar to the published pseudoknot and stem–loop structure. The prediction was done in 16 s.

**Viral 3′-UTR sequences**

We also tested comRNA on a set of 18 viral 3′-UTR sequences that has been used to test GPRM (Hu, 2002). They were retrieved from the PseudoBase (van Batenburg *et al.*, 2001) at http://wwwbio.leidenuniv.nl/~Batenburg/PKB.html, and their accession numbers were PKB183-189 and PKB194-204. Seven of them are from soil-borne rye mosaic viruses, and the rest are from soil-borne wheat mosaic viruses. The 18 sequences range in length from 23 to 34 nt, and most of the pairwise sequence identities are in the range 15–35%. Each sequence contains a pseudoknot structure formed by two stems. With \( S = 0.38 \), comRNA predicted the pseudoknot structures almost completely correctly in all the sequences. The sequences and prediction results can be found at http://ural.wustl.edu/softwares.html. This prediction took 4 s.

**Bacterial G-box sequences**

Another data set we tested contains 15 bacterial G-box sequences in the 5′-UTR of the xpt-phuX operon, each with length of \( \sim 80 \) nt. Most of the pairwise sequence identities are
Table 1. Comparison of prediction performance between comRNA and other programs

<table>
<thead>
<tr>
<th>Program</th>
<th>10 α operon mRNA leader sequences, length 134 nt, sequence identity 60–100%</th>
<th>11 S15 mRNA leader sequences, length 172–204 nt, sequence identity 30–50%</th>
<th>18 viral 3′-UTR sequences, length 23–34 nt, sequence identity 15–35%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC first/best/avg prediction Correct/total stems in first structure</td>
<td>CC for two structures first/best/avg prediction Correct/total stems in first structure</td>
<td>CC first/best/avg prediction Correct/total stems in first structure Correct/total stems in best structure Run-time</td>
</tr>
<tr>
<td></td>
<td>Run-time                   Correct/total stems in best structure</td>
<td>Run-time</td>
<td>Correct/total stems in best structure</td>
</tr>
<tr>
<td>COMRNA</td>
<td>0.65/0.65/0.25 3 (a, b, c)/5 3 (a, b, c)/5 3 s</td>
<td>0.95/0.95/0.45 0.69/0.91/0.34 0/5</td>
<td>3 (a, b, d)/3 2 (a, b)/3 3 (a, b, c)/3 0/5</td>
</tr>
<tr>
<td>RNA</td>
<td>0.50/0.69/0.35 3 (a, b, c)/5 3 (a, b, c, d)/5 9 s</td>
<td>0.00/0.00/0.00 0.00/0.00/0.00</td>
<td>0/0 0/0 0/0 0/0</td>
</tr>
<tr>
<td>RNAGA</td>
<td>0.35/0.37/0.32 1 (a)/3 1 (a)/3 4.3 h</td>
<td>0.0/0/0 0/0/0</td>
<td>0/0 0/0 0/0 0/0</td>
</tr>
<tr>
<td>FOLDALIGN</td>
<td>0.30/0.30/0.30 1 (c)/1 1 (c)/1 18 h</td>
<td>0.0/0/0 0/0/0</td>
<td>0/1 0/1 0/1 0/1</td>
</tr>
<tr>
<td>COVE</td>
<td>0/0/0 0/24 0/24 80 s</td>
<td>0.02/0.02/0.02 0.02/0.02/0.02</td>
<td>0/50 0/50 0/50 0/50</td>
</tr>
<tr>
<td>CARNAC</td>
<td>−0.45/0.44 – 1 (a)/1 0.2 s</td>
<td>−0.67/0.38 −0.65/0.39</td>
<td>2 (a, b)/3 2 (a, b)/3 2 (a, b)/3 2 (a, b)/3</td>
</tr>
<tr>
<td>DYNALIGN</td>
<td>−0.37/0.32 – 1 (a)/3 57 h</td>
<td>−0.64/0.45 −0.88/0.57</td>
<td>2 (a,b │ a,d)/6 2 (a,b │ a,d)/6 2 (a,b │ a,d)/6 2 (a,b │ a,d)/6</td>
</tr>
<tr>
<td>MFOLD</td>
<td>0.35/0.35/0.35 1 (a)/4 1 (a)/4 16 s</td>
<td>0.54/0.57/0.49 0.76/0.78/0.64</td>
<td>2 (a,b)/6 3 (a,b,c)/6 2 (a,b)/6 3 (a,b,c)/6</td>
</tr>
<tr>
<td>PKNOTS</td>
<td>0.34/0.34/0.34 1 (a)/6 1 (a)/6 71 h</td>
<td>0.0/0/0 0/0/0</td>
<td>2 (a,b)/6 3 (a,b,c)/6 0/6 0/6</td>
</tr>
</tbody>
</table>

**Parameters used for each program.**
- **COMRNA**—see Results section.
- **RNA**—percentage of sequences sharing common structures, 80%; evaluated the best of all reported structures in the top five patterns, and the predicted structure with the lowest free energy was regarded as the ‘first’ prediction. **RNAGA**—stem population, 100; maximum iterations, 100; evaluated the structures with top 10 scores, and the predicted structure with the highest conservation score was regarded as the ‘first’ prediction. **DYNALIGN**—max separation, 10; gap, 3. **MFOLD**—predicted optimal and suboptimal structures within 10% of the minimum energy. **FOLDALIGN, COVE, CARNAC and PKNOTS**—used default parameters. For **CARNAC and DYNALIGN**, there is no ‘first prediction’ because they predict structures on all pairs of sequences.

**The run-time is the CPU time a program spent on running on all sequences at once (for COMRNA, RNA, RNAGA, FOLDALIGN and COVE), on all pairs of sequences (for CARNAC and DYNALIGN) or on all individual sequences (for MFOLD and PKNOTS) in a data set.**

**Program RNA did not report structures of short patterns on the viral data set.**

Downloaded from https://academic.oup.com/bioinformatics/article-abstract/20/10/1591/237294 by guest on 10 June 2018
Fig. 4. Prediction on bacterial α operon leader sequences with \( S = 0.56 \). The five predicted stems are represented by the letters \( a, (b, c), d, e \) and \( f \).

Fig. 5. Prediction on bacterial S15 mRNA leader sequences with \( S = 0.56 \). (a) The highest-scoring structure is similar to the published pseudoknot structure. (b) The ninth highest-scoring structure is similar to the published stem–loop structure. The bold lines represent sequences and the boxes represent stems. Two boxes marked with the same letter (e.g. \( a \) and \( a' \)) in a sequence represent the 5' and 3' halves of a stem. The lengths of the bold lines and boxes are in proportion to the real sequence and stem length. Internal loops and bulges may exist but are not displayed in the boxes.

in the range 45–75%. The G-box structure was recently discovered to be highly conserved across bacterial species and involved in guanine recognition (Mandal et al., 2003). It is composed of a three-stem junction, and base pairing between the two stem-loop sequences possibly permits formation of a pseudoknot structure. With \( S = 0.5 \), the first structure reported by comRNA is very similar to the recently proposed structure, as shown in Figure 6. The prediction was completed in 4 s.

Statistical significance of the predicted structures
comRNA reports structures with the highest plausibility scores. To measure how statistically significant these
Common RNA secondary structure motif prediction

Fig. 6. Prediction on bacterial G-box sequences at $S = 0.5$. The four predicted stems are represented by the letters, $a, b, c$ and $d$. Stems of the proposed structure are marked in different shades.

Fig. 7. Comparison between the structure score distribution on the real sequences and 300 randomly shuffled sequences.

structures are compared with random structures, we compared the structure score distribution between real and randomly shuffled sequences. For each data set, we aligned the sequences using CLUSTALW (Thompson et al., 1994), and reassembled the aligned columns in a random order to form a random data set, which maintained the length and sequence similarity as in the real sequences. Three hundred such random sequence sets were generated and run by comRNA, and their structure scores were collected and their distribution was compared with that of the real sequence set. Figure 7 shows the percentage of structures that have scores higher than a certain value, for both the real and random sequence sets. We can see that the structure score distributions on real and shuffled sequences are well distinguished, and high-scoring structures have a much higher probability of occurring in the real sequences as opposed to shuffled sequences, which suggests that the high-scoring structures reported by comRNA are statistically significant. We should mention that because we have 300 more examples of random sequences than the real sequences, sometimes the structures from random sequences can have scores higher than the highest score from the real sequences, as seen on the $\alpha$ operon data set.

Parameter setting
Since heuristic procedures are applied in the algorithm, the settings for the major threshold parameters ($L, E, S$ and $p$) are critical to the prediction performance. We have found that the default settings for $L (e=4)$, $E (e=-5)$ and $p (e=0.5)$ generally work well for most of the cases, and the only critical parameter we need to worry about is $S$, the stem similarity threshold.

We examined how the prediction accuracy, search space and run-time change with different parameter settings for $S$, as shown in Figure 8. We found that, generally, the prediction accuracy increased with decreasing $S$ (Fig. 8a). $S$ needed to be lower than a critical value for the majority of the information of the real stems to be included in the graph. Beyond the critical value, the specificity may decrease slightly because more false positive stem connections can be included in the graph. On the other hand, the search space and run-time increased exponentially with a lowering of $S$ beyond a critical value and can be unaffordable with current computer power (Fig. 8b and 8c). However, for the four data sets we tested, the program gave very good predictions before the run-time became unaffordable.

Therefore, to get a reasonably good prediction in acceptable time, we estimate the value for $S$ based on the size and
density of the graph so that the program can finish with the best predictions in the desired run-time (default 1 min). This is done automatically by the program by default. The user can choose to run the program longer or set a specific value for S as desired.

**Comparison with other programs**

We compared the performance of comRNA with that of other existing RNA secondary structure prediction programs, including RNA (Bouthinon and Soldano, 1999), RNAGA (Chen et al., 2000), FOLDALIGN (Gorodkin et al., 1997), COVE (Eddy and Durbin, 1994), CARNAC (Perriguet et al., 2003), DYNALIGN (Mathews and Turner, 2002), MFOLD (Zuker, 1994) and PKNOTS (Rivas and Eddy, 1999). RNA, RNAGA, FOLDALIGN and COVE predict common RNA secondary structures on multiple sequences and were run once on each data set. CARNAC and DYNALIGN were run on all pairs of sequences in a data set. MFOLD and PKNOTS were run on individual sequences.

We ran these programs on the first three data sets with the known structures described above and evaluated their prediction accuracy and speed. For the α operon and S15 sequence sets, since we only know the real structure in *E. coli*, we compared the prediction accuracy of the programs only based on their predicted structures in the *E. coli* sequence. For the viral data set, their predictions on all sequences were considered.

We evaluated the prediction accuracy on two levels: the stem level and the nucleotide level. At the stem level, we counted the total and correctly predicted number of stems in each program’s prediction. A predicted stem is considered correct if it covers at least half of the real stem in the known structure. At the nucleotide level, we compared the base pairs at the exact positions between the predicted and known structures, and quantified their agreement using the Matthews correlation coefficient derivative for RNA secondary structure, which was presented and used by Gorodkin et al. (2001). The correlation coefficient (CC) is approximated as

\[
CC \approx \sqrt{\frac{TP}{TP + FN} \cdot \frac{TP}{TP + FP}}
\]

where TP, FP and FN are the number of true positives (base pairs that are predicted and exist in the known structure), false positives (base pairs that are predicted but do not exist in the known structure) and false negatives (base pairs that are not predicted but exist in the known structure), respectively. CC can be viewed as the geometric mean of sensitivity and specificity. 0 ≤ CC ≤ 1. A higher CC means better prediction accuracy.

The prediction performance of the nine programs is shown in Table 1. We can see that comRNA gave the best overall prediction performance on all the three data sets, with a higher CC and a higher ratio of correct/total predicted stems than those of the other programs. ComRNA also ran fast compared with other programs. We should point out that all the data sets we tested have pseudoknot structures, but the other programs except RNA and PKNOTS were not designed to predict pseudoknots and thus could not possibly predict the real structures at 100% accuracy.

**DISCUSSION**

We present a new graph-theoretical approach for predicting common RNA secondary structure motifs in multiple sequences by finding and assembling conserved stems. As shown in the results, in the absence of any expert knowledge or structural alignment, our program was able to find...
simple or complicated conserved structures completely or partially correctly in a medium number (<20) of RNA sequences of moderate length (<300 nt) in acceptable time on current computers. By adopting a procedure that takes potential RNA stems as comparison units, it is able to retain both speed and sensitivity on complicated RNA structures. A similar topology-based pattern matching algorithm has been designed for protein structure comparison (Gilbert et al., 1999, 2001).

An important advantage of this method is its ability to detect pseudoknot structures because it does not have restrictions on how the stems are arranged. Moreover, our scheme of comparing stems allows detection of common structure motifs in both conserved and diverged RNA sequences, without requiring a global sequence similarity or knowing the structural alignment in advance. In addition, it is intrinsically able to distinguish sequences by the presence or absence of one or more conserved stems in them, which makes it possible to find common RNA structure motifs shared only by subsets of sequences.

The algorithm can take advantage of the sequence similarity information by introducing alignment anchor constraints, which contributes to reducing the search space and runtime of the algorithm, especially on sequences sharing high sequence similarity. Comparison of the run-time between predictions with and without anchor constraints is illustrated in Figure 8c.

Another advantage of our method lies in the fact that it is able to report a number of the best candidate common structures. It does so partly because the predicted optimal structure according to our structure significance measurement is not necessarily the real structure. More importantly, reporting multiple candidate structures is helpful in determining alternative RNA structures possibly formed in the sequences, which has been found to be the case for many RNA regulatory motifs, such as the previously shown ribosomal protein S15 mRNA leader regulatory motifs (Benard et al., 1996; Serganov et al., 2002).

One concern about this approach is that the worst-case time complexity of the maximum clique/path finding algorithm is non-polynomial, which makes it seem impractical to work on large numbers (>20) of long (>300 nt) sequences. Our experience with the test sets shows that the practical average time complexity is much less than the worst-case because the stem and clique graph size and density are generally much smaller than they appeared to be, due to the anchor-region constraints and heuristic procedures involving various cutoff parameters during graph construction. For large data sets, to reduce the run-time without badly affecting the sensitivity, we can try to set stringent cutoffs in the beginning to make the original graph smaller and sparser, find the strongest conserved stems and then search the remaining regions with less stringent cutoffs for weaker conserved and compatible stems.

ACKNOWLEDGEMENTS

We thank Dr Sean Eddy and Dr Elena Rivas for helpful comments and discussions. This work is supported by grant HG00249 from NHGRI at NIH.

REFERENCES


1601
Y. Ji et al.


