BIOINFORMATICS ORIGINAL PAPER

Vol. 24 no. 20 2008, pages 2281–2287
doi:10.1093/bioinformatics/btn393

Genome analysis

Fast and accurate search for non-coding RNA pseudoknot structures in genomes

Zhibin Huang[1,†], Yong Wu[1,†], Joseph Robertson[2], Liang Feng[2], Russell L. Malmberg[2,3] and Liming Cai[1,2,*]

[1]Department of Computer Science, [2]Institute of Bioinformatics and [3]Department of Plant Biology, University of Georgia, Athens, GA 30602

Received on April 23, 2008; revised on June 20, 2008; accepted on July 25, 2008

Advance Access publication August 7, 2008

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: Searching genomes for non-coding RNAs (ncRNAs) by their secondary structure has become an important goal for bioinformatics. For pseudoknot-free structures, ncRNA search can be effective based on the covariance model and CYK-type dynamic programming. However, the computational difficulty in aligning an RNA sequence to a pseudoknot has prohibited fast and accurate search of arbitrary RNA structures. Our previous work introduced a graph model for RNA pseudoknots and proposed to solve the structure–sequence alignment by graph optimization. Given *k* candidate regions in the target sequence for each of the *n* stems in the structure, we could compute a best alignment in time $O(k^t n)$ based upon a tree width *t* decomposition of the structure graph. However, to implement this method to programs that can routinely perform fast yet accurate RNA pseudoknot searches, we need novel heuristics to ensure that, without degrading the accuracy, only a small number of stem candidates need to be examined and a tree decomposition of a small tree width can always be found for the structure graph.

Results: The current work builds on the previous one with newly developed preprocessing algorithms to reduce the values for parameters *k* and *t* and to implement the search method into a practical program, called RNATOPS, for RNA pseudoknot search. In particular, we introduce techniques, based on probabilistic profiling and distance penalty functions, which can identify for every stem just a small number *k* (e.g. $k \leq 10$) of plausible regions in the target sequence to which the stem needs to align. We also devised a specialized tree decomposition algorithm that can yield tree decomposition of small tree width *t* (e.g. $t \leq 4$) for almost all RNA structure graphs. Our experiments show that with RNATOPS it is possible to routinely search prokaryotic and eukaryotic genomes for specific RNA structures of medium to large sizes, including pseudoknots, with high sensitivity and high specificity, and in a reasonable amount of time.

Availability: The source code in C++ for RNATOPS is available at www.uga.edu/RNA-Informatics/software/rnatops/

Contact: cai@cs.uga.edu

Supplementary information: The online Supplementary Material contains all illustrative figures and tables referenced by this article.

1 INTRODUCTION

Non-coding RNAs (ncRNAs) have been shown to be involved in many biological processes including gene regulation, chromosome replication and RNA modification (Frank and Pace, 1998; Nguyen *et al*., 2001; Yang *et al*., 2001). Searching genomes using computational methods has become important for annotation of ncRNAs (Griffiths-Jones, 2007; Hofacker, 2006; Lowe and Eddy, 1997; Rivas and Eddy, 2001; Rivas *et al*., 2001; Washietl *et al*., 2005). In general, to annotate an individual genome for a specific family of ncRNAs, a computational tool needs to scan through the genome and align its sequence segments to some structure model for the ncRNA family. Those segments with significant alignment scores are then reported as the results. An algorithm that can perform an accurate sequence–structure alignment is thus the core of such a searching tool.

A few programs (Brown and Wilson, 1996; Klein and Eddy, 2003; Liu *et al*., 2006; Lowe and Eddy,1997) have been developed for genome annotation using the covariance model (CM) introduced by Eddy and Durbin (1994). Based on a CM, the optimal alignment between a sequence and a pseudoknot-free structure can be performed with a dynamic programming algorithm in $O(WN^3)$, where N is the size of the model and W is the length of the sequence. In particular, RSEARCH (Klein and Eddy, 2003) and Infernal (http://infernal.janelia.org/) are two programs that can perform such searches. CM-based methods can achieve high searching accuracy; however, due to the time complexity needed for sequence–structure alignment, a CM-based search may be inefficient on complex or large RNA structures. Further, pseudoknot structures, which contain at least two interweaving stems, cannot be modeled with CMs.

Searches on genomes can be speeded up with filtering methods (Bafna and Zhang, 2004; Lowe and Eddy, 1997; Weinberg and Ruzzo, 2004, 2006; Zhang *et al*., 2005). Sometimes it is possible to efficiently remove genome segments unlikely to contain the desired pattern. For example, in tRNAscan-SE (Lowe and Eddy, 1997), two efficient filters are used to preprocess a genome and remove the part that is unlikely to contain the searched tRNA structure; the remaining part of the genome is then scanned with a CM to identify the tRNA. FastR (Bafna and Zhang, 2004) considers the

[*]To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

structural units of an RNA structure; it evaluates the specificity of each structural unit and construct filters based on the specificity of these structural units. In Weinberg and Ruzzo (2004), an algorithm is developed to safely break the base pairs in an RNA structure and automatically select filters from the resulting Hidden Markov Model (HMM). These approaches have significantly improved the computational efficiency of genome searches.

RNA structures that contain pseudoknots pose special problems. A number of creative approaches (Cai *et al*., 2003; Rivas and Eddy, 1999, 2000; Uemura *et al*., 1999) have been tried to model the crossing stems of pseudoknots; however, the time and space complexities for optimal sequence–structure alignment based on these models are $O(N^4)$ or $O(N^5)$. These models are not practical for efficient searching. Intersecting CMs have been proposed for pseudoknots (Brown and Wilson, 1996), and used to search small genomes (Liu *et al*., 2006), but these have the same efficiency problem. Several heuristic search methods have been developed that can work with RNAs containing pseudoknots; as heuristics, each has some limitations. For example, ERPIN (Gautheret and Lambert, 2001), considers the stem loops contained in a secondary structure. The genome is then scanned to find the possible hit locations for each stem loop. A hit for the overall structure is reported when there exists a combination of hit locations for different stem loops that conform with the overall structure. However, ERPIN does not allow gaps in the alignment and thus may have low sensitivity when the target is a remote homolog of the query structure model.

Our previous work introduced a graph modeling method that can profile the secondary structure of a family of RNAs including pseudoknots (Song *et al*., 2005, 2006). In this method, the topology of an RNA structure is specified with a mixed graph, with non-directed edges denoting stems and directed edges for loops. With this model, we proposed to efficiently solve the structure–sequence alignment problem, including pseudoknots, by exploiting the small tree width (Robertson and Seymour, 1986) demonstrated by the structure graphs of almost all existing RNA pseudoknots. Theoretically, given k (pairs of) regions as candidates for each of the n stems in the structure and given a tree decomposition of tree width t for the structure graph, the alignment can be computed in time $O(k^t n)$. However, to implement the algorithm into computer programs that can routinely perform fast, accurate RNA pseudoknot search, heuristics for the preprocessing steps need to be able to associate results with small values of parameters k and t while maximizing search accuracy.

In this article, we present our current work, built upon the previous one, to develop a practical program, called RNATOPS, for RNA pseudoknot search. In this work, we have introduced new, effective heuristic techniques for generating stem candidates and for tree decomposition of RNA structure graphs. In particular, parameter k can be chosen relatively small (e.g. $k \leq 10$) to ensure both accuracy and efficiency of the search. The alignment algorithm (and thus the search algorithm) runs in time $O(k^t n)$, linear in the number n of stems in the profiled RNA structure. It is scalable with the complexity of the profiled structure because the yielded tree decompositions have small tree width t, $t \leq 4$, for almost all RNA secondary structures (including pseudoknots). In this article, we evaluate RNATOPS with search tests conducted on several medium to large size RNAs (including pseudoknots) and make comparisons with existing RNA structure search programs such as Infernal.

2 APPROACH

We refer the reader to the publications (Song *et al*., 2005, 2006) for detailed discussions of our graph modeling method for RNA structures and on the solution to structure–sequence alignment based on tree decomposition of the structure graph. In this section, we give a brief recap of the necessary notions and techniques relevant to the current article. We then present the new heuristic techniques for stem candidate identification and for tree decomposition designated for RNA structure graphs. These heuristic techniques aim at achieving a fast structure–sequence alignment without degrading the accuracy.

2.1 A graph model for structure search

Our structure model based on a mixed graph specifies the consensus structure of an RNA family as a relation among all involved structural units: stems and loops. In this graph, each vertex defines either base pairing regions of a stem; two vertices representing two complementary regions (forming a stem) are connected with a non-directed edge. Two vertices defining two regions that are physically next to each other (forming a loop) are connected with a directed edge (from $5'$ to $3'$). The individual structural units are stochastically modeled; every stem is associated with a simplified CM and every loop with a profile HMM. The structure graph is capable of modeling RNA structures resulting from multi-body interactions of nucleotides, such as triple helices, as well as pseudoknots. Figure 1 in Supplementary Material shows the structure graph of a typical bacterial tmRNA.

Searching in a target genome consists of sliding a window of appropriate size along the target genome, then testing for a possible alignment of the structural model with the sequence segment within the current window. With the graph model, the structure–sequence alignment is identical to the task of finding the optimal subgraph of a graph G isomorphic to another graph H, where H is the RNA structure graph and G is constructed from the target sequence in a preprocessing step. We proposed two methods to cope with the computational intractability of the subgraph isomorphic problem. One method was to pre-identify in the target sequence top k candidates for every stem in the structure. The other method was to tree decompose the structure graph. Based upon a tree decomposition, a dynamic programming algorithm could solve the subgraph isomorphic (thus the structure–sequence) problem in theoretical time $O(k^t n)$, where n is the number of stems in the structure and t is the tree width of the graph tree decomposition (Song *et al*., 2005, 2006). This article presents new heuristic techniques to support these two methods.

2.2 Model training

Model training involves defining the structure graph, individual CMs and profile HMMs from a set of training RNA sequences given in a *pasta* file. The pasta format (*p*airing plus f*asta*) is a representation we developed for multiple structural alignment and consensus structure of RNA sequences (Fig. 2 in Supplementary Material). It labels stem positions with an upper case letter for one side, the corresponding lower case letter for the other side. The first line of the file denotes the consensus structure using matching (upper and lower case) letters for conserved base pairs and '.'s for unpaired nucleotides or possibly consensus insertions. Representation with pairing letters has the advantage of being able to denote arbitrary RNA structures, including pseudoknots and triple helices. A structure graph is produced from the consensus structure, where one vertex is for one letter, one non-directed edge connects the two vertices of matched letters and one directed edge connects two neighboring letters (from $5'$ to $3'$, Fig. 1 in Supplementary Material).

The rest of the lines in the pasta file are RNA sequences structurally aligned to the consensus structure, possibly containing '-'s for deletions. Individual CMs and profiles HMMs are constructed from the multiple structure alignment as follows. Every stem of base-paired regions (with matching letters) produces one simplified CM that does not contain bifurcation rules or rules for the sequence connecting the two base-paired regions. One profile HMM is generated from every two neighboring base regions. The profile HMM allows possible match, insertion and deletion states in every column of the multiple alignment. The parameters of these stochastic models are computed from the multiple structural alignment using the maximum likelihood method. To avoid over-fitting the models, we incorporate background statistics. In particular, we allow pseudocounts for nucleotides in the match, insertion and deletion states of the profile HMM. For the simplified CM, a 4×4 prior probability matrix P_p for base pairs and a weighting parameter w are introduced so that the probability of a base pair $P(x,y)$ is defined as the weighted sum $wP_t(x,y)+(1-w)P_p(x,y)$, where P_t is the base pair probability matrix obtained from the training data.

2.3 Identifying stem candidates

The sequence segment within the sliding window is preprocessed to identify top k candidates for the CM of every stem. Given a CM modeling some consensus stem, the score of every possible structural motif within the window aligned to the model is computed (Fig. 3 in Supplementary Material). Candidates can be found by a simple dynamic programming algorithm; we describe here four heuristic techniques developed to ensure that the correct motif structure for the CM, if it does exist in the sequence, is highly likely to be among the selected top k candidates for some small value of k.

(1) Regions from which candidates can be selected are constrained according to the statistical distribution of the consensus stem in the sample (training sequences). In particular, we assume a Gaussian distribution for the position of the consensus stem in the RNA structure. The constrained region for the correct motif of the consensus stem is within a certain number (e.g. 3) of the SD of the average position.

(2) For training sequences that demonstrate a large SD for the position of some consensus stem, training sequences are partitioned into clusters, each with a small SD for the stem position. Therefore, more than one (constrained) region may be derived for the correct motif of the consensus stem.

(3) The candidates so identified are then ranked again according to statistical distributions of various length parameters associated with a consensus stem, including the length of the stem, the distance between the two stem arms and the head and tail offsets. The scores of every possible motif candidate c of the CM M are recalculated according to the formula: $S(c,M)=uA(c,M)+(1-u)P(c,M)$, where $A(c,M)$ is the logodds score from the alignment, $P(c,M)$ is the penalty function for the deviations of all lengths list above from their means and u, $0\le u\le1$, is a weighting parameter. In particular, $P(c,M)$ is computed based on the log score $\log(1/cK^2)$, where $K=|l-\mu|/\sigma\ge1$ for the length l deviating from mean μ (with a SD σ) and c is a selected constant.

(4) Finally, since it is possible that several structural motifs, heavily overlapping in their positions, may all have decent alignment scores with respect to a stem model, it suffices to record only one representative for them. Strategies have been used to select representatives and to ensure a low value for k, the number of top candidates.

2.4 Tree decomposition for structure graphs

With our model, almost all ncRNAs have structure graphs of small tree width. However, finding the optimal tree decomposition (one with the smallest tree width) is NP-hard. Available efficient tree decomposition algorithms are for general graphs and usually do not guarantee the optimal tree width. For RNA structure graphs, we develop a linear-time greedy algorithm that can yield tree decomposition of tree width almost always bounded by 4. An earlier version of this algorithm was given in (Song *et al.*, 2005), but it used the idea of minimum fill-in and may produce decompositions of unnecessarily larger tree widths. We present a self-contained version of the algorithm here.

First, the algorithm removes arcs (i.e. non-directed edges) in the structure graph that cross with other arcs. It does this by greedily removing the arc crossing the most other arcs and repeating the step on the remaining graph until there is no crossing arc in the graph (Fig. 4a and b in Supplementary Material). This step actually removes stems involved in pseudoknots in the corresponding RNA structure; a crossing arc-free structure graph corresponds to a pseudoknot-free RNA structure. Such a graph is an outer-planar graph that has tree width 2, whose optimal tree decomposition can be found as follows.

Note that in a structure graph, the vertices are arranged in the direction of from $5'$ to $3'$ (left to right in the figures) based on the directed edge relation. We also add the source s and sink t as the left most and the right most vertices, respectively. We use notation H_b^a to represent the subgraph induced by the set of vertices 'from' vertex a 'to' vertex b (inclusive, from $5'$ to $3'$). Then to decompose the subgraph H_t^s, the algorithm handles the following three major scenarios recursively (and the recursive process terminates when the considered subgraph is empty).

(1) If (s,X) is a directed edge but (x,t) is not, where (X,x) is an arc (Fig. 5a in Supplementary Material), then the root node $\{s,t\}$ has child node $\{s,x,t\}$, which in turn has child node $\{s,X,x\}$ (Fig. 5b in Supplementary Material). Node $\{s,X,x\}$ will be the root for the subtree generated from subgraph H_x^X and node $\{s,x,t\}$ will be the root for the subtree generated from subgraph H_t^x.

(2) If (s,X) and (x,t) both are directed edge, where (X,x) is an arc (Fig. 5c in Supplementary Material), then the root $\{s,t\}$ has child node $\{s,X,t\}$, which in turn has child node $\{X,x,t\}$. Node $\{X,x,t\}$ will be the root for the subtree generated from subgraph H_x^X (Fig. 5d in Supplementary Material).

(3) If (s,X) is a directed edge but (X,x) is not an arc (Fig. 5e in Supplementary Material), then the root $\{s,t\}$ has a child node

{s, X, t}, which in turn will be the root for the subtree generated from subgraph H_t^X (Fig. 5f in Supplementary Material).

The algorithm modifies the resulting tree decomposition as follows. For every removed arc (v, v'), the algorithm identifies two nodes, one containing vertex v and another containing its counterpart v'. For every tree node on the path from the former node to the latter, the algorithm adds v to it (Fig. 6 in Supplementary Material). This gives a tree decomposition for the original structure graph.

3 IMPLEMENTATION

RNATOPS, implemented in language C++, has been compiled and tested on several systems, including Desktop Linux computers, a Linux cluster and a SUN workstation running SunOS 5.1.

4 EVALUATION

To evaluate the search program and the effective of the heuristics, we tested RNATOPS using four types of RNAs of medium to large sizes: bacterial tmRNA, bacterial RNaseP type B RNA, yeast telomerase RNA and bacterial 16S rRNA. We compare both search accuracy and efficiency of RNATOPS with those of Infernal and FastR, two of the best known general-purpose programs for RNA structure search.

4.1 Data preparation and tests conducted

Bacterial tmRNAs (Moore and Sauer, 2007; Nameki *et al*., 1999) have a complex structure containing four pseudoknots; there are 178 molecules in the Rfam (Griffiths-Jones *et al*., 2005) seed alignment with an average length of 364 bases (Fig. 1 in Supplementary Material). The tmRNA sequences have variations in structure with certain stem loops present in some sequences and absent in others. We extracted a subset of 43 tmRNA sequences from the 178 molecules in the alignment, which did not differ from each other in the presence or absence of any stem loops, and for which the entire bacterial genome sequence was available; columns consisting entirely of gaps were then removed from the alignment.

RNaseP, bacterial type B, RNAs have multiple stem loops and one sophisticated pseudoknot (Brown, 1999; Harris *et al*., 2001; Fig. 7 in Supplementary Material). There are 31 sequences of average length 367 in the Rfam seed alignment. We extracted a subset of 10 sequences which did not differ from each other in the presence or absence of any stem loops; the full genome sequence was available for 7 of the 10.

Yeast telomerase RNAs contain a conserved, essential, pseudoknot within a large stem loop (Chen and Greider, 2004). We used an alignment, of length 834, for this region (Dandjinou *et al*., 2004) of six *Saccharomyces* species telomerase RNAs. While the genome of *S.cerevisiae* has been completely sequenced, those of the other *Saccharomyces* species are available in varying degrees of completeness and assembly. We were able to collect four *Saccharomyces* genomes total, three in addition to *S. cerevisiae*, to search.

The bacterial 16S rRNA is a conserved molecule which has been extensively used for phylogenetic studies of bacteria. We obtained an alignment (of 1570 bp) of the 16S rRNA for gammaproteobacteria from the ribosomal database (Cole *et al*., 2007); from this we selected those sequences which contained an identical match in a fully sequenced bacterial genome. Although many gammaproteobacteria genomes have been sequenced, for only 12 was there an exact match between the database sequence and a genomic sequence, which we required to take advantage of the expert alignment from the database. These sequences were used as the training set.

For all the genomic searches, we followed a cross-validation approach in which the RNA found in a genomic sequence was removed from the alignment, and the remaining sequences were used as a training set for a search on that genome.

To search genomes of a considerable length, we identified highly conserved motifs of the RNA molecules, then searched the genomes with these, after which we examined the region around a potential hit for a structural match to the whole molecule. We note that a program that can automatically identify a conserved motif as the optimal filter is currently being developed for RNATOPS.

4.2 Comparison to other search programs

To compare with Infernal (infernal.janelia.org), we downloaded Infernal from its website, compiled it and installed it, and compared its performance on one of the same Linux computers we used for testing of RNATOPS. Both Infernal and RNATOPS use multiple structural alignments for model training and use filters to speed up the search.

We used FastR (Bafna and Zhang, 2004; Zhang *et al*., 2005) through job submission at a website. As such, it is difficult for us to compare the performance of FastR on a server of unknown configuration and numbers of cpus with the performance of RNATOPS. During the times we tested it, our analyses were the only ones listed in the job queue. We estimated the time of the run from the time of submission and the time at which the job finished e-mail was sent. The user can pick from pre-defined profiles for searching. It is unknown to us if these profiles included the tmRNAs for the genomes we tested. Hence these FastR tests may or may not correspond to the training sets we used, in which we left out the RNAs for the genome targeted for searching.

4.3 Search accuracy

4.3.1 Bacterial tmRNAs We searched 43 bacterial genomes with RNATOPS for tmRNAs using a leave-one-out cross-validation approach. Table 1 in Supplementary Material gives a comparison of the results achieved with RNATOPS with those of Infernal. RNATOPS was evaluated with varying parameter k, the number of candidate regions examined for each stem in the structure. Increasing k from 10 to 15 to 25 increased the sensitivity of the whole structure search, but also increased the time taken. For example, at $k = 10$, the bacterial genome searches gained 88% sensitivity and 100% specificity; at $k = 25$, the sensitivity increased to 98%. Infernal had 100% sensitivity and specificity for these searches with comparable times spent.

We observed that the tmRNAs missed by RNATOPS at the low k values generally had one or more portions in the structure, which significantly deviated from the consensus structure. In particular, several stems in these sequences consisted of mainly rare, non-canonical base pairs, which may have been placed in pairing positions during the multiple alignment process.

We also compared the alignments of the tmRNA structures found by Infernal and RNATOPS. Four structures identified by RNATOPS have stem alignments off their correct positions for more than a few nucleotides in their alignments; Infernal identified seven such structures. There are in total nine such stem misalignments in the structures identified by RNATOPS; there were total 17 in those structures identified by Infernal. In addition, because Infernal is based on the pseudonot-free CM, in a structure alignment, regions 'belonging to' a pseudoknot may be mistakenly aligned to pseudoknot-free substructures. In particular, in this set of search tests, there were totally five such incorrect assignments found in the search results of Infernal while the issue was not raised on RNATOPS (Table 6 in Supplementary Material).

We also tried the search with FastR web server, which includes tmRNAs as a profile. We selected one bacterial genome on which RNATOPS successfully found the tmRNA, and one genome on which RNATOPS failed to find the tmRNA, then submitted these to the FastR server. FastR gave the same results as RNATOPS with these two sequences, finding the structure in one sequence and missing it in the other (Table 2 in Supplementary Material), again suggesting there is something unusual about the tmRNA that both programs missed. Several additional bacterial genomes were submitted to the FastR server, but no results were returned.

4.3.2 Bacterial RNaseP (Bact. B) RNAs The bacterial RNaseP (Bact. B) RNA is similar in size to the tmRNAs, but has a more complex pseudoknot structure. Both the RNATOPS and Infernal programs had 100% sensitivity and 100% specificity in finding the RNaseP RNAs in the seven genomes tested (Table 3 in Supplementary Material); RNATOPS identified two structures whose alignments put in total four stems off their correct positions by more than a few nucleotides (Table 6 in Supplementary Material). A comparison with the tmRNA results suggests that the more complex pseudoknot structure in RNaseP (Bact. B) was handled well by RNATOPS, with less than a doubling in time taken for similar sized genomes, while Infernal took about nine times as long.

4.3.3 Saccharomyces telomerase RNAs The conserved core region of Saccharomyces telomerase RNAs is more than twice as long as the bacterial tmRNAs or RNaseP RNAs, and the *Saccharomyces* genomes are 2 to 10 times larger than the bacterial genomes tested. The pseudoknot structure itself is not complex, but it is contained within a stem-loop and some additional stem-loops are present. Both programs found the four *Saccharomyces* fungal telomerase RNAs perfectly in their genomes; RNATOPS took from 5.5 to 6.4 min, while Infernal took from 295 to 654 min for the same searches (Table 4 in Supplementary Material).

4.3.4 Bacterial 16S rRNAs The bacterial 16S rRNAs are the longest molecule we tested with lengths around 1500 bp. The results were similar to the telomerase and RNAseP RNAs, with both RNATOPS and Infernal finding the target with perfect specificity and sensitivity, but with RNATOPS performing the search in an average of 14.1 min as opposed to 88 min for Infernal (Table 5 in Supplementary Material).

4.4 Efficiency

The theoretical time of the search method can be expressed as $O(T_aN)$, where T_a is the time needed for the structure alignment between the structure model and the sequence segment within the window sliding through the genome of N nucleotides. T_a actually consists of two parts: the time for the preprocessing step and the time for the dynamic programming step for the subgraph isomorphism based upon a tree decomposition. The latter takes $O(k^tn)$ time, where t, usually not >4, is the tree width of the tree decomposition and n is the number of stems in the structure. Recall that k is the number of candidates selected for the simplified CM model of a stem during the preprocessing; it is a relatively small parameter that can be used to tune the accuracy of the alignment. The time for the preprocessing step is $O(R^2Mn)$, where M is the maximum size of a CM and R is the maximum length of the sequence regions from which candidates are selected. These regions are fairly restricted by the preprocessing techniques we introduced here (Section 2.3). Our experiments showed that the preprocessing time $O(R^2Mn)$ is roughly the same as the time $O(k^tn)$ needed for the dynamic programming step when k is around 10 and that it is dominated by the latter for larger values of k or t. So the time for searching a whole genome is very much scalable with the size and complexity of the RNA structure searched.

Overall, our results indicate that the RNA graph model plus tree decomposition method incorporated into RNATOPS performed very well in efficiency while maintaining high search accuracy. The advantage of RNATOPS in speed, compared to other programs, increased as the length of the modeled molecule increased. This is because its search time depends on the number of stems, not the number of nucleotides, in the structure. Thus, the efficiency advantage becomes even more significant for RNATOPS to search for the larger yeast telomerase RNA and bacterial 16S rRNA (Tables 4 and 5 in Supplementary Material). Note that RNATOPS search accuracy can be tuned by the user through parameter k, to balance search sensitivity versus running time. The problems that RNATOPS had, where target RNAs were not found, were in stem-loop regions of tmRNAs where individual molecules deviated from the consensus structure; increasing the k value allowed RNATOPS to resolve most of these, at the cost of a slight decrease in speed.

5 DISCUSSION

Heuristic techniques have been presented in this article with the aim to develop a fast and accurate RNA pseudoknot search program based on our previous work in an RNA graph modeling method. Through search tests on the implemented program RNATOPS, we have shown its performance comparable with or better than that of Infernal and FastR in identifying large or complex RNA structures including pseudoknots. We discuss in the following the strengths and weaknesses of RNATOPS.

One apparent advantage of RNATOPS is its ability to detect pseudoknots accurately without compromising computation time. Theoretically, RNATOPS can feasibly consider all combinations of stems for pseudoknot alignment through a non-conventional, tree decomposition-based dynamic programming. Detecting a pseudoknot as a whole structure avoids the difficulty with pseudoknot-free models that the predicted alignment sometimes incorrectly forms pseudoknot-free substructures in a 'pseudoknot territory'.

Another advantage of RNATOPS is its search speed. The theoretical time $O(k^tn)$ for structure–sequence alignment with RNATOPS has been effectively speeded up by the introduced heuristic techniques that can yield small values for k and t. Another important factor contributing to the efficient time is parameter n, the number of stems, not the number of nucleotides in the structure which would otherwise be at least one magnitude larger. As shown in the test results, RNATOPS has essentially broken the inefficiency barrier that might have heldback other pseudoknot detection models, reducing the computation time from hours to minutes.

Nevertheless, since the introduced heuristics produce only k pairs of candidate regions for each individual stem in the structure to align to, for a small k, they may not include the real candidate of the stem and may bring inaccuracy to the search result. In particular, when a stem in the RNA contains non-canonical base pairings, for which candidates may not be accurately identified, it is possible that all pairs of candidates between this stem and another are ‘incompatible’, resulting in a invalid alignment and lower sensitivity. This issue does not exists in the CM–CYK-based programs like Infernal as its stem candidates are found globally instead of locally.

Another issue with the current version of RNATOPS is the computation of the structure–sequence alignment without reusing the data from the previous scanning window frame. In fact, the CM–CYK-based search method can save a factor of $O(M)$ computation time by reusing data between two consecutive window frames (Durbin *et al*., 1998), where M is the CM model length. This issue might have cost RNATOPS some speed in the search tests; however, we believe that it is possible to make technical improvements for RNATOPS in reusing the data between scanning window frames to further speed up the search.

We consider two future developments for RNATOPS. First, the graph model can also easily profile structures caused by nucleotide interactions beyond the binary base pairing. For example, the graph model makes it easy to profile tertiary interactions or triple helices recently found in the telomerase RNA genes of human and yeast genomes (Chen and Greider, 2004; Lin *et al*., 2004; Shefer *et al*., 2007; Theimer *et al*., 2005). Although one of the two stems involved in such a triple helix is actually formed by two base pairing regions that are arranged in the same direction ($5'$ to $3'$), our approach will allow the stem to be modeled with an individual CM the same way as modeling a regular stem, without the need of additional, new techniques.

Second, the current implementation of program does not allow the search for an instance of ncRNA in the target genome that differs in structure significantly from those in the training set; nor can the current program consider alternative or optional substructures in RNAs. One solution to this will be to develop probabilistic profiling of variable substructures that may occur in the structure model. In particular, our modeling method makes it possible to characterize and implement the structure of an RNA family with a graph model that contains probabilistic edges to specify variable substructures. This will bear similarity to earlier methods by Holmes (2004) and Rivas (2005) but with the ability to include pseudoknots.

ACKNOWLEDGEMENTS

The authors are grateful to the anonymous referees for their constructive comments that have greatly improved this manuscript. We also thank the authors of Infernal, RSEARCH and FastR who have made their software packages publicly available to use.

Funding: National Institutes of Health (BISTI grant No: R01GM072080-01A1).

Conflict of Interest: none declared.

REFERENCES

Bafna,V. and Zhang,S. (2004) FastR: fast database search tool for non-coding RNA. In *Proceedings of the 3rd IEEE Computational Systems Bioinformatics Conference*. Imperial College Press, London, pp. 52–61.

Brown,J.W. (1999) The Ribonuclease P database. *Nucleic Acids Res.*, **27**, 314.

Brown,M and Wilson,C. (1996) RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. In Hunter,L. and Klein,T. (eds) *Proceedings of Pacific Symposium on Biocomputing*. World Scientific Publishing Co, Singapore.

Cai,L. *et al*. (2003) Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. *Bioinformatics*, **19** (Suppl. 1), i66–i73.

Chen,L. and Greider,C.W. (2004) An emerging consensus for telomerase RNA structure. *Proc. Natl Acad. Sci. USA*, **101**, 14683–14684.

Cole,J.R. *et al*. (2007) The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data. *Nucleic Acids Res.*, **35**, D169–D172.

Dandjinou,A.T. *et al*. (2004) A phylogenetically based secondary structure for the yeast telomerase RNA. *Curr. Biol.*, **14**, 1148–1158.

Durbin,R. *et al*. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Eddy,S.R. and Durbin,R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res.*, **22**, 2079–2088.

Frank,D.N. and Pace,N.R. (1998) Ribonuclease P: unity and diversity in a tRNA processing ribozyme. *Annu. Rev. Biochem.*, **67**, 153–180.

Gautheret,D. and Lambert,A. (2001) Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. *J. Mol. Biol.*, **313**, 1003–1011.

Griffiths-Jones,S. (2007) Annotating noncoding RNA genes. *Annu. Rev. Genomics Hum. Genet.*, **8**, 279–298.

Griffiths-Jones,S. *et al*. (2003) Rfam: an RNA family database. *Nucleic Acids Res.*, **31**, 439–441.

Griffiths-Jones,S. *et al*. (2005) Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res.*, **33**, D121–D124.

Harris,J.K. *et al*. (2001) New insight into RNase P RNA structure from comparative analysis of the archaeal RNA. *RNA*, **7**, 220–232.

Hofacker,I.L. (2006) RNAs everywhere: geonom-wide annotation of structured RNAs. *Genome Inform.*, **17**, 281–282.

Holmes,I. (2004) A probabilistic model for the evolution of RNA structure. *BMC Bioinformatics*, **5**, 166.

Infernal: inference of RNA alignments. (2008) http://infernal.janelia.org/ (last accessed date June 30, 2008).

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

Lin,J. *et al*. (2004) A universal telomerase RNA core structure including structured motifs required for binding the telomerase reverse transcriptase protein. *Proc. Natl Acad. Sci. USA*, **101**, 14713–14718.

Liu,C. *et al*. (2006) Efficient annotation of non-coding RNA structures including pseudoknots via automated filters, In *Proceedings of Life Science Society Computational Systems Biology Conference (CSB 2006)*. Imperial College Press, London, pp. 99–110.

Lowe,T.M. and Eddy,S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.*, **25**, 955–964.

Moore,S.D. and Sauer,R.T. (2007) The tmRNA system for translational surveillance and ribosome rescue. *Annu. Rev. Biochem.*, **76**, 101–124.

Nameki,N. *et al*. (1999) Functional and structural analysis of a pseudoknot upstream of the tag-encoded sequence in E. coli tmRNA. *J. Mol. Biol.*, **286**, 733–744.

Nguyen,V.T. *et al*. (2001) 7SK small nuclear RNA binds to and inhibits the activity of CDK9/cyclin T complexes. *Nature*, **414**, 322–325.

Rivas,E. (2005) Evolutionary models for insertions and deletions in a probabilistic modeling framework. *BMC Bioinformatics*, **6**, 63.

Rivas,E. and Eddy,S.R. (1999) A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, **285**, 2053–2068.

Rivas,E. and Eddy,S.R. (2000) The language of RNA: a formal grammar that includes pseudoknots. *Bioinformatics*, **16**, 334–340.

Rivas,E. and Eddy,S.R. (2001) Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, **2**, 8.

Rivas,E. *et al*. (2001) Computational identification of noncoding RNAs in E. coli by comparative genomics. *Curr. Biol.*, **11**, 1369–1373.

Robertson,N. and Seymour,P.D. (1986) Graph minors II. Algorithmic aspects of tree-width. *J. Algorithms*, **7**, 309–322.

Shefer,K. *et al*. (2007) A triple helix within a pseudoknot is a conserved and essential element of telomerase RNA *Mol. Cell Biol.*, **27**, 2130–2143.

Song,Y. *et al*. (2005) Tree decomposition based fast searching for RNA structures with and without pseudoknots. *Proc. IEEE Comput. Syst. Bioinform. Conf.*, IEEE Computer Society Press. 223–234.

Song,Y. *et al*. (2006) Efficient parameterized algorithms for biopolymer structure-sequence alignment. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **3**, 423–431.

Theimer,C.A. *et al*. (2005) Structure of the human telomerase RNA pseudoknot reveals conserved tertiary interactions essential for function. *Mol. Cell*, **17**, 671–682.

Uemura,Y. *et al*. (1999) Tree adjoining grammars for RNA structure prediction. *Theor. Comput. Sci.*, **210**, 277–303.

Washietl,S. *et al*. (2005) Fast and reliable prediction of noncoding RNAs. *Proc. Natl Acad. Sci. USA*, **102**, 2454–2459.

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), I334–I341.

Weinberg,Z. and Ruzzo,W.L. (2006) Sequence-based heuristics for faster annotation of non-coding RNA families. *Bioinformatics*, **22**, 35–39.

Yang,Z. *et al*. (2001) The 7SK small nuclear RNA inhibits the CDK9/cyclin T1 kinase to control transcription. *Nature*, **414**, 317–322.

Zhang,S. *et al*. (2005) Searching genomes for noncoding RNA using FastR. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **2**, 366–379.