Algorithms for Searching RNA Motifs in Genomic Sequences

Jingping Liu, Bin Ma, and Kaizhong Zhang
Dept. of Computer Science, University of Western Ontario,
London, Ont. N6A 5B7 Canada
jliu36, bma, kzhang@csd.uwo.ca

Abstract

With the progress of genome projects, a vast amount of nucleotide sequence data is now available, which makes it possible to study the species-specific gene expression systems involving transcription and translation for a wide range of organisms. Biological experiments indicate that functional RNAs have characteristic RNA structural motifs represented by specific combinations of base pairings and conserved nucleotides in the loop regions. The searching for those well-ordered RNA structures and their homologues in genomic sequences is very helpful for the understanding of RNA-based gene regulation. In this study, we consider the following problem: Given an RNA sequence with a known secondary structure, efficiently determine candidate segments in a given genomic sequence that can potentially form RNA secondary structures similar to the given RNA secondary structure. We search all potential stem-loops similar to ones of the given RNA secondary structure first, then based on located stem-loops, we propose an efficient bottom-up algorithm to detect potential homologous structural RNAs in the complete genome.

Keywords: RNA stem-loop/stem, RNA tree representation, pattern recognition, homologous structural search.

1 Introduction

The discovery of microRNAs (miRNAs) suggests that there is a large class of small non-coding RNAs (ncRNAs). Such ncRNAs are involved in the specific recognition of cellular nucleic acid targets through complementary base pairing to control cell growth and differentiation[10]. Accumulating evidence indicates that ncRNAs can play critical roles in a wide range of cellular processes, such as chromosomal silencing, transcriptional regulation, and developmental control. Due to the importance of ncRNAs, Knowledge discovery of them in genomic sequence databases by computational approaches is highly desirable [3].

Over the last decade, computational search approaches for distinct RNA structural motifs, such as Palingol, Patsearch, tRNAscan-SE, and tRNAscan [2, 5, 9], have made a great progress. However, Palingol and Patsearch use a motif descriptor or an indirect quantitative scoring system, it is difficult and not sufficient to characterize exactly the structural features. On the other hand, tRNAscan-SE is limited to the prediction of tRNA genes. Bafna et al. [1] presented FastR for searching non-coding RNA, which is faster than RSEARCH [7] but in some applications the ratio of false negative of filters may high. Zhang et al. [8] proposed a novel algorithm, HomoStRscan, to search for homologous structural RNAs by scanning a genomic sequence. HomoStRscan differs from other currently used approaches in considering each base and base pair in the query RNA and applying gap penalty and stacking pair bonus. In general, this approach is suitable for detecting any RNA gene with an established secondary structure in genome sequences.

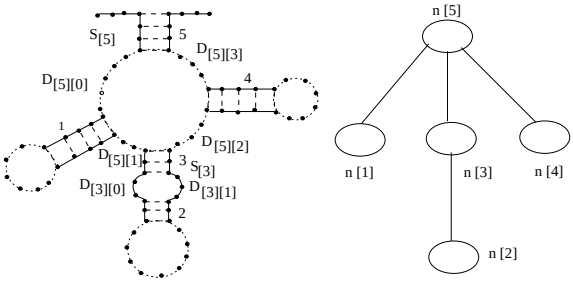
In this paper, we describe an efficient and sensitive bottom-up approach for searching homologs in a given genomic sequence. It intentionally functions as a filer to improve HomoStRscan [8] to be more time efficient. For a tRNA searching in the *Helicobacter Pylori 26695* genome ($\sim$1.67M bases), the bottom-up tool located around two thousand candidate segments for tRNAs, and took approximately 30s to run under an Intel/Linux PC with 2.8GHz, 1 Gb memory. Moreover, all true tRNAs of the genome listed in the NCBI database are in those candidate segments.

2 Tree Representation of RNA Secondary Structure

Let R_s represent the set of base pairs of an RNA secondary structure for an RNA primary sequence. In general, an RNA secondary structure contains base pairs and unpaired bases. For $k > 0$, a *stem* (stack) of RNA secondary structure is composed of contiguous base pairs $(i, j), (i+1, j-1), \ldots, (i+k, j-k)$, where base $i-1$ does not pair with base $j+1$ and base $i+k+1$ dose not pair with base $j-k-1$. The number of base pairs in a stem is called *stem size*, denoted by S. For $p < q$, a *loop-segment* of RNA secondary structure consists of all contiguous unpaired bases $p, p+1, \ldots, q-1, q$, where $p-1$ and $q+1$ are not unpaired bases. In a

loop-segment, the number of unpaired bases is called *loop-segment size*, denoted by D. Basically the most fundamental and interesting feature in an RNA secondary structure is the stem-loop motif [4, 11]. In this study, the *stem-loop*, denoted by S_l, is composed of a stem surmounted by a loop-segment. Notice that this definition is more restrictive than the general definition of stem-loop. The surmounted loop-segment in the stem-loop is also referred to as *hairpin loop*. The number of un-paired bases in a hairpin loop is called *hairpin seize*, denoted by H.

In [14], Zhang presented a tree representation of RNA secondary structure. In this tree representation, all internal nodes are base pairs and leaf nodes are unpaired bases. The definitions of child, sibling, and leaf follow naturally. The order of base pair (i, j)'s children is the order they appear in the RNA primary structure. We modify the tree representation in [14] to form a tree representing RNA secondary structure R_s. In our tree representation, each internal node is a stem, and each leaf is a stem-loop. However, the parent-child relationship is similar to the one in [14]. An example of our tree representation is shown in Figure 1.



(a) RNA Secondary Structure (b) Tree Representation

Figure 1: RNA secondary structure and the tree.

In an RNA secondary structure, we utilize stem size S as a parameter to measure a stem, while the distance between two consecutive stems is measured by the parameter, loop-segment size D. Let R_t be our tree representing a given RNA secondary structure R_s. The nodes of R_t are numbered from 1 to $|R_t|$ according to the standard *postorder* (left-to-right) method. For an internal node $n[i]$ with $d[i]$ degree, let $D_{[i][0]min}$, $D_{[i][0]max}$, $\cdots$, $D_{[i][d[i]]min}$, $D_{[i][d[i]]max}$ be the corresponding min and/or max values of the distances between two consecutive nodes. That is, $D_{[i][0]min}$ and $D_{[i][0]max}$ are the min and/or max values of the distance between $n[i]$ and its first child node. $D_{[i][1]min}$ and $D_{[i][1]max}$ are the min and/or max values of the distance between the first and second child nodes of $n[i]$, and so on. $D_{[i][d[i]]min}$ and $D_{[i][d[i]]max}$ are the min and/or max values of the distance between $n[i]$ and its last child node. In R_t, $D_{[i][0]min}$ and $D_{[i][0]max}$ are stored in the first child of $n[i]$, $D_{[i][1]min}$ and $D_{[i][1]min}$ are stored in the second child of $n[i]$, and so forth. $D_{[i][d[i]]min}$ and $D_{[i][d[i]]max}$ will be contained in $n[i]$. Moreover, if $n[i]$ is the first child of its parent, the min and/or max values of the distance between $n[i]$ and its parent will also be stored in $n[i]$, or if the $n[i]$ has left sibling(s), the min and/or max values of the distance between $n[i]$ and its first left sibling will also be stored in $n[i]$. The min and/or max stem sizes of the stem represented by $n[i]$ will be stored in $n[i]$.

3. Algorithms

Based on our tree representation of RNA secondary structure, we introduce an efficient and sensitive bottom-up approach for determining whether some segments in a given DNA sequence can potentially form RNA secondary structures similar to the given RNA secondary structure.

In terms of structural parameters (i.e., stem sizes and loop-segment sizes) of the given RNA secondary structure R_s, we can set up the corresponding min and/or max values of stem and loop-segment sizes. The default min and/or max values of each size are half of the size, and one and half of the size, respectively. For a certain application, those values can be easily modified by users. The main stages of the bottom-up approach are as follows. (1) Search for each potential stem-loop, $S_{l[1]}$, $\ldots$, $S_{l[i]}$, $\ldots$, satisfying the corresponding min and/or max values of loop-segment (i.e. hairpin loop) and stem sizes, in the given genomic sequence R. (2) Based on our tree representation of RNA secondary structure, and corresponding min and/or max values of stem sizes and loop-segment sizes of the given RNA secondary structure R_s, we design a new efficient bottom-up approach to compute more and more complicated substructures until they form RNA secondary structures similar to the given one R_s. (3) If applicable, we utilize additional RNA biological features, such as the terminal tag CCA with the $3'$ end of tRNA, to further improve the false positive ratio (i.e., to reduce the number of candidates). Finally, we output all candidate segments.

3.1 Stem-Loop Searching Algorithm

Suppose that in an RNA secondary structure, the base pairing is limited to the Watson-Crick base pairs, and the Wobble base pair. We now propose a searching algorithm, SL-SEARCH, for finding a *single* potential stem-loop in an given genomic sequence R of length n by looking for base pairs. The SL-SEARCH starts at two specified positions i and j ($1 \leq i < j \leq n$), in sequence R, then extends i to left and j to right for locating contiguous base pairs character by character until no further complement base matching. During the complement base matching, we compute the stem size S. Obviously, the hairpin loop size $H = j - i - 1$. Since SL-SEARCH looks for base pairs character by character, the time complexity can be bound by $O(S)$. Notice

that the algorithm SL-SEARCH is central to locate RNA motifs in genomes. There is an interesting way to improve SL-SEARCH using suffix tree data structures [6, 13] and the constant-time lowest common ancestor (LCA) query technique [6, 12]. After being improved, the SL-SEARCH can achieve the desirable time complexity: constant-time.

For searching *all* of potential stem-loops in the given genomic sequence R of length n, we iteratively call SL-SEARCH to scan the entire R. In other words, $j = 1, 2, \cdots, n$ and $i = j - H - 1$. Notice that when scanning R, we may locate "invalid" potential stem-loops. If the stem of one stem-loop is totally overlapped by the stem of another stem-loop, we call such a stem-loop *invalid*. Invalid potential stem-loops can be easily deleted.

Lemma 1 *Let H_{min} and H_{max} be the min and max values of hairpin size H, respectively. The total number of potential stem-loops in a given genomic sequence of length n can be bounded by $O(n)$, assuming H_{min} and H_{max} are constants.*

3.2 The Bottom-Up Approach

Based essentially on located stem-loops and our tree representation of RNA secondary structure, the bottom-up computing starts at leaf nodes representing stem-loops until the root. The key ideas of the bottom-up algorithm, MOTIF-SEARCH, are as follows. (1) For internal node $n[i]$ with $d[i]$ degree (i.e., $d[i]$ substructures, such as stem-loops, see Figure 1), we first integrate the first substructure and the second substructure, such that the distance D between them satisfying $D_{[i][1]min} \leq D \leq D_{[i][1]max}$, to form an integrated substructure, then integrate the integrated substructure and the third substructure to form one more complex substructure, and so on until the $d[i]$-th proper substructure. All of such substructures containing $d[i]$ consecutive substructures will be saved in an array. In the array, each element has the start and end positions, SP and EP, of the integrated substructure. (2) After finding such integrated substructures containing $d[i]$ consecutive substructures, we now call SL-SEARCH to extend each integrated substructure to contain the stem represented by $n[i]$, such that the distances and stem size S satisfying the corresponding min and/or max values. Those potential substructures containing extended stems are saved in array $L_{[i]}$. In the array, each element has the start and end positions, SP and EP, of the more complicated substructure. (3) Repeat steps 1 and 2 until the root in the tree.

Theorem 1 *Given an RNA secondary structure R_s and a genomic sequence R of length n, let N be the number of nodes in the tree representing R_s, S_{max} be the max value of stem size, D_{max} be the max distance between two potential consecutive substructures, and r_{max} be the max number of potential substructures starting at some position i (i-th base) in R, the time and space complexities of MOTIF-SEARCH can be bound by $O(nNr_{max}D_{max} + nNS_{max})$ and $O(nN)$, respectively.*

4 Results and Discussion

We reported here the applications of our algorithms and the corresponding software "mSearch" in locating tRNAs and 5S rRNAs in several bacterial genome databases. In order to distinguish the difference of structural feature of the query RNA encoded in either the positive stranded sequence (PSS) or reverse complementary sequence (RCS), we use the same target sequence data, and two structural and/or parameter profiles of the given RNA secondary structure R_s.

Applications for Searching tRNAs: The given -tRNA 5322 - 4247 in Helicobacter Pylori 26695 is encoded in the $5' \leftarrow 3'$ sequence. Based essentially on the given tRNA secondary structure and the common features of tRNA, such as the variable loop containing $3 - 21$ bases, we design two parameter profiles (i.e., the min and/or max values of distances) to search tRNAs similar to the given RNA secondary structure R_s in the same target sequence data. One of the parameter profiles is for searching -tRNAs (see Table 1), and the other one is for searching +tRNAs (See Table 2). The parameter profiles can be modified by users.

Acceptor stem size S	$S \geq 6$
AC base pairs in Acceptor stem	≤ 3
Hairpin size H in stem-loops	$3 \leq H \leq 12$
Stem size S in stem-loops	$3 \leq S \leq 5$
AC base pairs in stem-loops	≤ 1
D_{apt} between Acceptor-TψC stems	$0 \leq D_{apt} \leq 2$
D_{ta} between TψC-Anticodon stems	$0 \leq D_{ta} \leq 21$
D_{ad} between Anticodon-D stems	$0 \leq D_{ad} \leq 6$
D_{dap} between D-Acceptor stems	$0 \leq D_{dap} \leq 4$

Table 1: -tRNA Parameter Profile

Acceptor stem size S	$S \geq 6$
GU base pairs in Acceptor stem	≤ 3
Hairpin size H in stem-loops	$3 \leq H \leq 12$
Stem size S in stem-loops	$3 \leq S \leq 5$
GU base pairs in stem-loops	≤ 1
D_{apd} between Acceptor-D stems	$0 \leq D_{apd} \leq 4$
D_{da} between D-Anticodon stems	$0 \leq D_{da} \leq 6$
D_{at} between Anticodon-TψC stems	$0 \leq D_{at} \leq 21$
D_{tap} between TψC-Acceptor stems	$0 \leq D_{tap} \leq 2$

Table 2: +tRNA Parameter Profile

For the complete genome of *Mycoplasma Genitalium (MG)*, 36 tRNAs were annotated in the NCBI database of Bacteria Complete Genomes. Among them, 16 tRNAs (+tRNAs) were encoded in the PSS, while 20 tRNAs (-tRNAs) were encoded in the RCS. Based essentially on the two default parameter profiles(see Tables 1 and 2), our "mSearch" located 992 candidate segments for tRNA in *MG* genome. In another experiment, using the same two default parameter profiles, we found 1640 candidate segments for tRNAs in the complete genome of *Helicobacter pylori 26695 (HP)*. For *HP* genome, 36 tRNAs were annotated in the NCBI database of Bacteria Complete Genomes. Among them, 15 tRNAs (+tRNAs) were encoded in the PSS, while 21 tRNAs (-tRNAs) were encoded in the RCS. In the two applications, we found all of tRNAs listed in NCBI databases (i.e. The false negative is zero).

Applications for Searching 5S rRNAs: The given - 5S rRNA in Bacillus Subtilis genome is encoded in the $5' \leftarrow 3'$ sequence. Similarly, we use one target sequence data and design two default parameter profiles to search homologous 5S rRNAs in genomes. Due to lack of space, we will not state the parameter profiles for searching 5S rRNAs in detail.

Since there exists a relatively small number of 5S rRNA genes in bacteria genomes, we only choose some subsequences of genomes, which contain relatively more 5S rRNAs, to evaluate the performance of our algorithms. For the subsequence (bases 1,900,000 - 2,300,000) of *Staphylococcus aureus subsp.aureus MW2 (MW2)* genome, three -5S rRNAs were annotated in the NCBI bacteria database. Based on the default -5S rRNA parameter profile, the "mSearch" located 1986 candidate segments for -5S rRNAs in the subsequence of *MW2*. In another experiment, the program located 1349 candidate segments for +5S rRNAs in the subsequence (bases 3,900,000 - 4,300,000) of *Escherichia Coli K12 (EK12)* using the default +5S rRNA parameter profile. For the subsequence of *EK12*, four +5S rRNAs were annotated in the NCBI bacteria database. In the two applications, we found all true 5S rRNAs of the two subsequences listed in the NCBI databases (i.e. The false negative is zero). Due to the existing of non-canonical base pairs, the ratio of false positive is not good enough. In ongoing work, we plan to add some constrains to reduce the number of candidates.

5 Conclusion

We have presented an efficient bottom-up approach, which takes as input an RNA sequence with a known secondary structure and a target genomic sequence, for detecting potential homologous structural RNAs in a sequence. In general, our approach can be used to search for any RNA segments with a known structure. The computational experiments on several complete genomes and subsequences indicate that the approach can efficiently filter out the regions that may not form such structural homologs, and locate all true tRNAs and 5S rRNAs annotated in NCBI databases. We expect to improve the ratio of false positive with additional structural features.

References

[1] V. Bafna, and S. Zhang, "FastR: Fast database search tool for non-coding RNA", *IEEE Computational Systems Bioinformatics Conference, CSB*, pp52-61, 2004.

[2] B. Billoud, M. Knotic, and A. Viari Palingol, "A Declarative programming Language to Describe Nucleic Acids' Secondary Structures and to Scan sequence Database",*Nucleic Acids RES. 24*, pp1395-1403, 1996.

[3] S. R. Eddy, "Computational genomics of noncoding RNA genes", *Cell 109*, pp137-140, 2002.

[4] N. El-Mabrouk, and M. Raffinot, "Approximate Matching of secondary Structures", *ACM*, pp156-164 2002.

[5] G. Grillo, F. Licciulli, S. Liuni, E. Sbisa, and G. Pesole, "Patsearch: a Program for the Detection of Patterns and Structural Motifs in Nucleotide sequences", *Nucleic Acids RES. 31*, pp3608-3612, 2003.

[6] D. Gusfield, "Algorithms on Strings, Trees, and Sequences, Computer Science and Computational Biology". *Cambridge University Press*, 1997.

[7] R. Klein, and S. Eddy, "Rsearch: Finding homologs of single structured rna sequences", *BMC Bioinformatics*, 4(1):44, 2003.

[8] S. Le, J. V. Maizel, and K. Zhang, "An Algorithm for Detecting Homologues of Known Structured RNAs in genomes", *IEEE Computational Systems Bioinformatics Conference, CSB*, pp300-310, 2004.

[9] T. M. Lowe, and S. R. Eddy, "tRNAscan-SE: a Program for Improved Detection of Transfer RNA Genes in Genomic Sequence", *Nucleic Acids RES. 25*, pp995-964, 1997.

[10] M. Matzke, A. J. M. Matzke, and J. M. Kooter, "RNA: Guiding Gene Silencing", *Science 293*, pp1080-1083, 2001.

[11] G. Mauri, and G. Pavesi, "Pattern Discovery in RNA Secondary Structure Using Affix Trees", *CPM*, pp278-294, 2003.

[12] B. Schieber, and U. Vishkin, "On Finding Lowest Common Ancestors: Simplifications and Parallelization" *SIAMJ. Comput., 17*, pp1253-62, 1988.

[13] E. Ukkonen, "On-Line Construction of Suffix Trees", *Algorithmica, 14(3)*, pp249-260, 1995.

[14] K. Zhang, "Computing Similarity between RNA Structures," *Proceeding of IEEE International Joint Symposia on Intelligence and Systems*, pp126-132, 1998.