An algorithm for detecting homologues of known structured RNAs in genomes

Shu-Yun Le
Laboratory of Experimental and Computational Biology
NCI Center for Cancer Research
National Cancer Institute, NIH
Bldg. 469, Room 151, Frederick
Maryland 21702, USA
shuyun@ncifcrf.gov

Jacob V. Maizel, Jr.
Laboratory of Experimental and Computational Biology
NCI Center for Cancer Research
National Cancer Institute, NIH
Bldg. 469, Room 151, Frederick
Maryland 21702, USA
jmaizel@ncifcrf.gov

Kaizhong Zhang
Department of Computer Science
University of Western Ontario
London, ON N6A 5B7, Canada
School of Computing Science
Simon Fraser University
Burnaby BC V5A 1S6, Canada
kzhang@csd.uwo.ca

Abstract

Distinct RNA structures are frequently involved in a wide-range of functions in various biological mechanisms. The three dimensional RNA structures solved by X-ray crystallography and various well-established RNA phylogenetic structures indicate that functional RNAs have characteristic RNA structural motifs represented by specific combinations of base pairings and conserved nucleotides in the loop region. Discovery of well-ordered RNA structures and their homologues in genome-wide searches will enhance our ability to detect the RNA structural motifs and help us to highlight their association with functional and regulatory RNA elements. We present here a novel computer algorithm, HomoStRscan, that takes a single RNA sequence with its secondary structure to search for homologous RNAs in complete genomes. This novel algorithm completely differs from other currently used search algorithms of homologous structures or structural motifs. For an arbitrary segment (or window) given in the target sequence, that has similar size to the query sequence, HomoStRscan finds the most similar structure to the input query structure and computes the maximal similarity score (MSS) between the two structures. The homologous RNA structures are then statistically inferred from the MSS distribution computed in the target genome. The method provides a flexible, robust and fine search tool for any homologous structural RNAs.

Keywords: RNA structure similarity; pattern recognition; homologous structural RNA search

1 Introduction

RNA is a conformationally polymorphic macromolecule and can be described by its nucleotide sequence and structural constraints manifested in its secondary and tertiary structure. Though single-stranded regions exist in most RNAs, the RNA molecules often fold back on themselves between complementary segments to form various structures guided mainly by Watson-Crick rules [1]. Recent advances in RNA studies indicate that RNA performs a wide range of functions in various biological processes. Included among these functions are catalysis and regulation of gene expression mediated by self-splicing ribozymes [1], small microRNAs (miRNAs) [2, 3], and other diversified RNA regulatory elements in the control of post-transcriptional [4-6] and translation [7-11]. Most of the functional RNA elements involve the specific structure motifs in the higher order structures rather than simple, linear sequence motifs [1]. It is the well-ordered structures formed uniquely in functional RNAs that play a crucial role in the biological mechanisms [12]. It is evident that a complete understanding of the function of RNA molecules requires knowledge of their 3-dimensional (3-D) structures. The determination of RNA 3-D structure is a limiting step in the study of RNA structure-function relationships because it is very difficult to crystallize and/or get nuclear magnetic resonance spectrum data for large RNA molecules. Thus, the establishment of homologous structures inferred across the diversified RNAs performing the same function is a very important step towards our understanding of the unrevealed property of structure-function for the new discovered RNA regulatory elements.

It is also true that most of the genomic sequences listed in databases contain annotated information of some structured RNAs, such as tRNA, 16S and 23S ribosomal RNAs (rRNAs), but there is little information about other well-established functional RNA elements and non-coding RNAs (ncRNAs). This information is necessary to understand the rapidly expanding databases of genome sequences. Over the last decade, computational search methods for distinct RNA structural motifs have made a great progress. A number of tools such as RNAMOT, Palingol, PatSearch, PatScan, RNAMotif, and ERPIN have been developed and have practical applications to the search for RNA structural motifs of iron-responsive elements (IRE), signal recognition particle (SRP) RNAs, and selenocysteine insertion (SECIS) elements [13-18]. Since these algorithms use a motif descriptor or an indirect quantitative scoring system, it is difficult and/or not sufficient to characterize exactly the structural feature in the well-ordered base-pairing stacking region or in the complicated domain including various non-canonical base-pairings. Although, the computational search tools for specific kinds of ncRNA, such as tRNAscan-SE, tRNAscan, tRNA CM, and EufindtRNA were quite efficient and successful, they are limited to the prediction of tRNA genes [19-22], and are not adapted to find other ncRNAs and RNA structural motifs.

Here, we describe a novel algorithm, HomoStRscan, to search for homologous structural RNAs by scanning a genomic sequence. HomoStRscan takes account of information of both the primary sequence and the secondary structural constraints of the query RNA in detail. The structural constraints are represented by the positions and types of each base pair in the structure. The algorithm provides two scoring matrices of base and base pairs that include 4 types of bases and 16 types of base-pairings as well as insertion and deletion of either base or base-pairings. Non-canonical base pairs, such as A:G, G:G, A:A found in the 5S rRNA structure, are allowable. The input scoring matrices of base and base pairs used in the comparison are controlled by users and can be changed easily based on specific searches. The algorithm 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 the new approach, HomoStRscan looks for the most similar structure to match the query structure in an arbitrary segment given in the target sequence. Variation in the size of the segment around the query sequence length is permitted. Simultaneously, the maximal similarity score (MSS) between the query RNA and the computed matching structure is calculated [23, 24]. The homologous RNAs are predicted by robust statistical inference from the MSS distribution that is computed by moving the window along the target sequence.

2 RNA structural alignment and RNA pattern searching

The primary structure of an RNA molecule is a sequence of nucleotides over the four-letter alphabet $\sum = \{A, C, G, U\}$. The secondary structure of an RNA is a set of base pairs between A and U, C and G, and G and U (canonical base pairs). These bonds have been assumed to be non-crossing in secondary structure. In this section, we first consider the structural alignment between an RNA secondary structure and an RNA sequence. We then extend this to RNA secondary structure pattern searching.

2.1 Structural alignment between an RNA secondary structure and an RNA sequence and RNA structure searching

We now consider the problem of structurally aligning an RNA secondary structure with an RNA sequence. Given an RNA R_1 with primary and secondary structures and an RNA R_2 with only primary sequence, the goal of the structural alignment is to identify a secondary structure for R_2 such that the structural similarity between R_1 and this identified structure is maximized among all the possible secondary structures of R_2.

The structural similarity between RNA structures is based on edit operations, i.e. insertion, deletion and substitution, on unpaired bases and base pairs [24, 27, 28]. In addition, we also allow a base pair to be aligned with non-canonical base pairs and even two unpaired bases to enhance the quality of the alignments [29].

We assume that there is a score function associated with the edit operations. Let $\sum = \{A, C, G, U\}$, a score function, $\gamma()$, for unpaired bases is defined on $\sum \cup \{-\}$ and a score function, $\delta()$, for base pairs is defined on $\sum \times \sum \cup \{(-,-)\}$.

Given an RNA secondary structures R_1 and an RNA sequence R_2, a structural alignment between R_1 and R_2 is represented by (R'_1, R'_2) satisfying the following conditions.

1) R'_1 is R_1 with some new symbols $-$ inserted and R'_2 is R_2 with some new symbols $-$ inserted such that $|R'_1| = |R'_2|$.

2) If $(r'_1[i], r'_1[j])$ is a base pair in R'_1, then either $r'_2[i]$ and $r'_2[j])$ are two bases in R'_2 or $r'_2[i] = r'_2[j] = -$.

¿From this definition, if $r'_1[i]$ is an unpaired base in R'_1, then $r'_2[i]$ is either a base in R'_2 or a $-$. In addition, when

$(r'_1[i], r'_1[j])$ is a base pair in R'_1 and both $r'_2[i]$ and $r'_2[j]$ are bases in R'_2, there is no requirement that $r'_2[i]$ and $r'_2[j]$ are complementary since we allow a base pair to be aligned with a non-canonical base pair and with two unpaired bases. The score function should be designed in such a way that the alignment between two base pairs will have a high positive score, the alignment between a base pair and a non-canonical base pair will have a low score, and the alignment between a base pair and two unpaired bases will have a low or even a negative score (see Table 2).

The similarity score for a given (R'_1, R'_2) is the summation of the individual scores of the edit operations involved plus gap penalty. For any base pair in R'_1, if it is aligned with $(-, -)$, then there is a base pair deletion cost, if it is aligned with two bases in R'_2, then there is a score of aligning two base pairs or aligning a base pair to two unpaired bases depending on the two bases involved. For any unpaired base in R'_1, if it is aligned with $-$, then there is an unpaired base deletion cost, if it is aligned with a base in R'_2, then there is a score of aligning two bases. For any base in R'_2 which is aligned with $-$, there is an unpaired base insertion cost.

A gap in an alignment (R'_1, R'_2) is a consecutive subsequence of $-$ in either R'_1 or R'_2 with maximal length. More formally $[i \cdots j]$ is a gap in (R'_1, R'_2) if either $r'_1[k] = -$ for $i \le k \le j$, $r'_1[i-1] \ne -$, and $r'_1[j+1] \ne -$, or $r'_2[k] = -$ for $i \le k \le j$, $r'_2[i-1] \ne -$, and $r'_2[j+1] \ne -$. For each gap in an alignment, in addition to the insertion/deletion costs, we will assign a constant, gap, as the gap initiation cost. This means that longer gaps are preferred since for a longer gap the additional cost distributed to each base is relatively small. This kind of affine gap penalty has long been used in sequence alignment [30] and structural alignment [31]. In biological alignment, if possible, longer gaps are preferred since it is difficult to delete the first element, but after that to continue deleting is mush easier. We also add a bonus, $stacking$, for each stacking pair of base pairs in R'_1 aligning with a stacking pair of base pairs in R'_2.

The previous algorithms for this problem were either too time consuming and with non-optimal solutions [32, 33] or designed with an alignment scoring scheme which is too simple to be useful in practice [34,27].

Given an RNA secondary structures R_1 and an RNA sequence R_2, the structural alignment problem is to determine a structural alignment with MSS. We will call this alignment as optimal alignment. The set of bases in R_2 that are aligned with base pairs in R_1 in the optimal structural alignment identifies a secondary structure of R_2. The above problem definition can be considered as global structural alignment problem. We can extend it to the RNA pattern searching problem. Given an RNA secondary structures R_1 and an RNA sequence R_2, the RNA structure pattern searching problem is to determine a substring $R_2[K, L]$ such that the score of the optimal structural alignment between R_1 and $R_2[K, L]$ is the largest among all the possible substring of R_2.

2.2 Tree representation of RNA secondary structure

We will use a tree (or a forest) to represent an RNA primary and secondary structures [27]. Suppose that S is the set of base pairs of an RNA secondary structure R. For $(i, j) \in S$, h is accessible from (i, j) if $i < h < j$ and there is no pair $(k, l) \in S$ such that $i < k < h < l < j$. $(k, l) \in S$ is accessible from $(i, j) \in S$ if both k and l are accessible from (i, j). This accessibility defines parent-child relationship. When h is accessible from (i, j), we define h as a child of (i, j). When (k, l) is accessible from (i, j), we define (k, l) as a child of (i, j). The order of (i, j)'s children is the order they appear in the primary sequence. This defines a tree structure of the RNA structure. In this representation, all internal nodes are base pairs and all leave nodes are unpaired bases.

Let T be the tree representing an RNA primary and secondary structures S. The nodes of T are numbered from 1 to $|T|$ according to the postorder. In the following, let $t[i]$ be a node of T with degree d_i and children $i_1, i_2, ..., i_{d_i}$. We use $T[i]$ to represent the subtree rooted at node $t[i]$ and $b[i]$ to represent the base information of node $t[i]$. If $t[i]$ is an internal node, then $b[i]$ is a base pair. If $t[i]$ is a leaf node, then $b[i]$ is an unpaired base. For any s, $1 \le s \le d_i$, let $F[i_1, i_s]$ represent the forest consisting of the subtrees $T[i_1], ..., T[i_s]$.

2.2.1 Notations

For an RNA structure S of length n represented by a tree T and an RNA sequence R of length m, we will consider the optimal alignment between some subtrees and subforests of T and all substrings of R. For any node $t[i]$, we use $A(F[i_1, i_s]; [k, l])$ to denote the score of the optimal alignment between $F[i_1, i_s]$ and subsequence $R[k, l]$, $D(F[i_1, i_s]; [k, l])$ to denote the score of the optimal alignment between $F[i_1, i_s]$ and subsequence $R[k, l]$ such that the alignment ends with a deletion, and $I(F[i_1, i_s]; [k, l])$ to denote the score of the optimal alignment between $F[i_1, i_s]$ and subsequence $R[k, l]$ such that the alignment ends with an insertion.

For subtree $T[i]$, $A(T[i]; [k, l])$, $D(T[i]; [k, l])$, and $I(T[i]; [k, l])$ are similarly defined. Let $t[i]$ be an internal node and $b[i] = (b_1, b_2)$, we also define the following two special alignment scores. We use $M_p(T[i]; [k, l])$ to represent the optimal alignment between $T[i]$ and substring $R[k, l]$ such that the alignment begins with the alignment of b_1 and $r[k]$ and ends with the alignment of b_2 and $r[l]$.

We use $D_p(T[i];[k,l])$ to represent the optimal alignment between $T[i]$ and substring $R[k,l]$ such that the alignment begins with the alignment of b_1 with $-$ and ends with the alignment of b_2 with $-$.

2.3 Properties of the maximum score

Recall that $\gamma(.,.)$ is the score function for unpaired bases and $\delta(.,.)$ is the score function for base pairs. In the following, $\gamma(i,-)$ has two different meanings. If $t[i]$ is a leaf node in T then we use $\gamma(i,-)$ to represent the cost of deleting the unpaired base $b[i]$. If $t[i]$ is an internal node in T then we use $\gamma(i,-)$ to represent the cost of deleting the subtree $T[i]$.

Lemma 1. *For any internal node $t[i]$ and $1 \le k \le m$,*

$$\begin{aligned} A(F[i_1,i_0];[k,k-1]) &= 0; \\ D(F[i_1,i_0];[k,k-1]) &= gap \\ I(F[i_1,i_0];[k,k-1]) &= gap \end{aligned}$$

Proof. Trivial. □

Lemma 2. *For any internal node $t[i]$ with $1 \le s \le d_i$ and $1 \le k \le m$,*

$$\begin{aligned} D(F[i_1,i_s];[k,k-1]) &= D(F[i_1,i_{s-1}];[k,k-1]) \\ &\quad +\gamma(i_s,-) \\ A(F[i_1,i_s];[k,k-1]) &= D(F[i_1,i_s];[k,k-1]) \\ I(F[i_1,i_s];[k,k-1]) &= D(F[i_1,i_s];[k,k-1]) + gap \end{aligned}$$

Proof. Trivial. □

Lemma 3. *For any internal node $t[i]$ and $1 \le k < l \le m$,*

$$\begin{aligned} I(F[i_1,i_0];[k,l]) &= I(F[i_1,i_0];[k,l-1]) + \gamma(-,t) \\ A(F[i_1,i_0];[k,l]) &= I(F[i_1,i_0];[k,l]) \\ D(F[i_1,i_0];[k,l]) &= I(F[i_1,i_0];[k,l]) + gap \end{aligned}$$

Proof. Trivial. □

Lemma 4. *For any internal node $t[i]$ with $1 \le s \le d_i$ and $1 \le k < l \le m$,*
$I(F[d_i,d_s];[k,l]) =$

$$\max \begin{cases} I(F[d_i,d_s];[k,l-1]) + \gamma(-,l) \\ A(F[d_i,d_s];[k,l-1]) + \gamma(-,l) + gap \end{cases}$$

Proof. Trivial. □

Lemma 5. *For any internal node $t[i]$ with $1 \le s \le d_i$ and $1 \le k < l \le m$, if $t[i_s]$ is an unpaired base, then*
$D(F[d_i,d_s];[k,l]) =$

$$\max \begin{cases} D(F[d_i,d_{s-1}];[k,l]) + \gamma(i_s,-) \\ A(F[d_i,d_{s-1}];[k,l]) + \gamma(i_s,-) + gap \end{cases}$$

$A(F[i_1,i_s];[k,l]) =$

$$\max \begin{cases} I(F[i_1,i_s];[k,l]) \\ D(F[i_1,i_s];[k,l]) \\ A(F[i_1,i_{s-1};[k,l-1]) + \gamma(i_s,l) \end{cases}$$

Proof. Trivial. □

Lemma 6. *For any internal node $t[i]$ with $1 \le s \le d_i$ and $1 \le k < l \le m$, if $t[i_s]$ is a base pair, then*
$D(F[i_1,i_s];[k,l]) =$

$$\max_{t=k}^{l+1} \begin{cases} D(F[i_1,i_{s-1}];[k,t-1]) + D_p(T[i_s],[t,l]) - gap \\ A(F[i_1,i_{s-1}];[k,t-1]) + D_p(T[i_s],[t,l]) \end{cases}$$

$A(F[i_1,i_s];[k,l]) = \max$

$$\begin{cases} I(F[i_1,i_s];[k,l]) \\ D(F[i_1,i_s];[k,l]) \\ \max_{t=k}^{t=l-1} A(F[d_1,d_{s-1}];[k,t-1]) + M_p(T[i_s],[t,l]) \end{cases}$$

Proof. Since $D(F[i_1,i_s];[k,l])$ ends with a deletion and $t[i_s]$ is a base pair, we have to consider the case where subtree $T[i_s]$ aligned with $R[t,l]$ starting and ending with deletion of base pair $t[i_s]$ and subforest $F[i_1,i_{s-1}]$ aligned with $R[k,t-1]$. Since we do not know t, we iterate all possibilities. The proof for $A(F[i_1,i_s];[k,l])$ is similar. □

Lemma 7. *For any internal node $t[i]$, if $d_i > 1$ then,*
$M_p(T[i];[k,l]) =$

$$\delta(b(i),(k,l)) + A(F[i_1,i_{d_i}],[k+1,l-1])$$

Otherwise
$M_p(T[i];[k,l]) =$

$$\delta(b(i),(k,l)) + \max \begin{cases} A(F[i_1,i_{d_i}],[k+1,l-1]) \\ M_p(T[i_1];[k+1,l-1]) \\ \quad +stacking \end{cases}$$

Proof. The first case is clear. For the second case, if $t[i]$ is aligned with k and l and $t[i_1]$ is aligned with $k+1$ and $l-1$, then we need to add the stacking pair bonus. □

In order to compute $D_p(T[i];[k,l])$, we need to compute $A_d(F[i_1,i_s];[k,l])$, $D_d(F[i_1,i_s];[k,l])$, and $I_d(F[i_1,i_s];[k,l])$ which are slightly different from $A(F[i_1,i_s];[k,l])$, $D(F[i_1,i_s];[k,l])$, and $I(F[i_1,i_s];[k,l])$. $A_d(F[i_1,i_s];[k,l])$ represents the optimal alignment score assuming that it is appended to an alignment ending with a deletion. $D_d(F[i_1,i_s];[k,l])$ and $I_d(F[i_1,i_s];[k,l])$ are similarly defined.

The formulas for $A_d(F[i_1,i_s];[k,l])$, $D_d(F[i_1,i_s];[k,l])$, and $I_d(F[i_1,i_s];[k,l])$ are exactly the same as those for $A(F[i_1,i_s];[k,l])$, $D(F[i_1,i_s];[k,l])$, and $I(F[i_1,i_s];[k,l])$ except the initialization which is list below.

Lemma 8. *For any internal node $t[i]$ and $1 \leq k \leq m$,*

$$\begin{array}{lcl} A_d(F[i_1, i_0]; [k, k-1]) & = & 0; \\ D_d(F[i_1, i_0]; [k, k-1]) & = & 0 \\ I_d(F[i_1, i_0]; [k, k-1]) & = & gap \end{array}$$

Proof. Trivial. □

Lemma 9. *For any internal node $t[i]$,*

$$D_p(T[i]; [k, l]) =$$

$$\delta(b(i), (-,-)) + 2gap + \max \left\{ \begin{array}{l} A_d(F[i_1, i_{d_i}]; [k, l]) \\ D_d(F[i_1, i_{d_i}]; [k, l]) - gap \end{array} \right.$$

Proof. If the optimal alignment uses $A_d(F[i_1, i_{d_i}]; [k, l])$, then we have to add two gap penalties. If the optimal alignment uses $D_d(F[i_1, i_{d_i}]; [k, l])$, then we will only add one gap penalty. □

2.4 Algorithm and complexity

2.4.1 Algorithm

The above lemmas give us a bottom up algorithm to determine the optimal structural alignment between the given RNA secondary structure with a substring of the RNA sequence R. The algorithm is given in Figure 1. Once $A([T[|T|], [K, L]) = \max_{1 \leq k < l \leq m} A([T[|T|], [k, l])$ is determined, a trace-back can be performed to produce the optimal alignment between T and $R[K, L]$. The set of bases in $R[K, L]$ which are aligned with base pairs in T in the optimal alignment forms the secondary structure of $R[K, L]$.

2.4.2 Complexity

Recall that d_i is the number of children of node $t[i]$. Let dp_i be the number of children of node $t[i]$ which are internal nodes. For each internal node $t[i]$, there are d_i forests of the form $F[i_1, i_s]$. The algorithm considers each forest and each interval of R. By lemma 6 only for dp_i forests we need to spend $O(m)$ time. All the other forests need $O(1)$ time. Therefore the time complexity is $O(\sum_{d_i>0}(d_i m^2 + dp_i m^3))$ = $O(m^2 \sum_{d_i>0} d_i + m^3 \sum_{d_i>0} dp_i)$ = $O(|T|m^2 + bp\, m^3)$ = $O(nm^2 + bp\, m^3)$ where bp is the number of base pairs in T. If for each internal node $t[i]$, we keep $M_p(T[i]; [k, l]$ and $D_p(T[i]; [k, l])$, then the space complexity is $O(bp\, m^2)$.

2.5 Improvements

The above time and space complexities can be further improved. In the following, let $del(k, l) = \sum_{i=k}^{l} \gamma(-, i)$.

Lemma 10. *For any internal node $t[i]$ and $1 \leq k < l \leq m$,* $A(T[i]; [k, l]) =$

$$\max \left\{ \begin{array}{l} \max_{k<s\leq l}\{M_p(T[i]; [s, l]) + del(k, s-1)\} + gap \\ \max_{k\leq t<l}\{M_p(T[i]; [k, t]) + del(t+1, l)\} + gap \\ \max_{k<s\leq t<l}\{M_p(T[i]; [s, t]) \\ \qquad +del(k, s-1) + del(t+1, l)\} + 2gap \\ M_p(T[i]; [k, l]) \\ D_p(T[i]; [k, l]) \\ \gamma(i, -) + del(k, l) + 2gap \end{array} \right.$$

Proof. Trivial. □

Lemma 11. *Let $t[i]$ be an internal node of T and $1 \leq k < l \leq m$, then given $M_p(T[i]; [k, l])$ and $D_p(T[i]; [k, l])$, $A(T[i]; [k, l])$ can be computed in $O(m^2)$ time.*

Proof. From lemma 10, it is clear that if $\max_{k<s\leq l}\{M_p(T[i]; [s, l]) + del(k, s-1)\}$, $\max_{k\leq t<l}\{M_p(T[i]; [k, t]) + del(t+1, l)\}$, and $\max_{k<s\leq t<l}\{M_p(T[i]; [s, t]) + del(k, s-1) + del(t+1, l)\}$ can be computed in $O(m^2)$ time, then $A(T[i]; [k, l])$ for $1 \leq k < l \leq m$ can be computed in $O(m^2)$ time.

Let $lg(k, l) = \max_{k<s\leq l}\{M_p(T[i]; [s, l]) + del(k, s-1)\}$, it is easy to see that $lg(k, l) = \gamma(-, k) + \max\{lg(k+1, l), M_p(T[i]; [k+1, l])\}$. This means that $lg(k, l)$ for $1 \leq k < l \leq m$ can be computed in $O(m^2)$ time. Similarly let $rg(k, l) = \max_{k\leq t<l}\{M_p(T[i]; [k, t]) + del(t+1, l)\}$, then $rg(k, l)$ can be computed in $O(m^2)$ time.

Let $lrg(k, l) = \max_{k<s\leq t<l}\{M_p(T[i]; [s, t]) + del(k, s-1) + del(t+1, l)\}$, then $lrg(k, l) = \delta(k) + \max\{lrg(k+1, l), rg(k+1, l)\}$. Therefore $lrg(k, l)$ for $1 \leq k < l \leq m$ can be computed in $O(m^2)$ time. □

Lemma 12. *If $t[i]$ is an internal node and $d_i = 1$, then* $D_p(T[i]; [k.l]) =$

$$\delta(b[i], (-,-)) + \max \left\{ \begin{array}{l} A(T[i_1]; [k, l]) + 2gap \\ D_p(T[i_1]; [k, l]) \end{array} \right.$$

Proof. If the optimal alignment of $D_p(T[i]; [k.l])$ uses $D_p(T[i_1]; [k, l])$, then no gap penalty will be added. Otherwise, we have to add two gap penalties. □

Lemma 13. *Let $t[i]$ is an internal node of T, $d_i = 1$ and $1 \leq k < l \leq m$, then given $M_p(T[i_1]; [k, l])$ and $D_p(T[i_1]; [k, l])$, $M_p(T[i]; [k, l])$ and $D_p(T[i]; [k, l])$ can be computed in $O(m^2)$ time.*

Proof. Immediate from lemma 7, 11, and 12. □

```
begin

  for i := 1 to |T|
    if t[i] is an internal node
      for k = 1 to n_2
        for l = k to n_2
          for s := 1 to d_i
            Compute A(F[i_1, i_s], [k, l]), D(F[i_1, i_s], [k, l]), and I(F[i_1, i_s], [k, l])

      for k = 1 to n_2
        for l = k to n_2
          for s := 1 to d_i
            Compute A_d(F[i_1, i_s], [k, l]), D_d(F[i_1, i_s], [k, l]), and I_d(F[i_1, i_s], [k, l])

      for k = 1 to n_2
        for l = k to n_2
          compute M_p([T[i], [k, l]) and D_p([T[i], [k, l])

  for k = 1 to n_2
    for l = k to n_2
      compute A([T[|T|], [k, l])

end
```

Figure 1: Algorithm: RNA Structural Pattern Searching

Theorem 1. *Given an RNA structure S of length n and an RNA sequence of length m, let hl be the number of hairpin loops in S and ml be the number of multiple loops in S, the optimal structural alignment score between S and a substring of R can be computed in $O(nm^2 + hl\ m^3)$ time and $O(m^2 \log(ml))$ space.*

Proof. (sketch) We first consider the time complexity. From lemma 13, we know that if we have a stacking pair of base pairs $t[i]$ and $t[i_1]$ where $t[i]$ is the parent of $t[i_1]$, then in time $O(m^2)$, instead of $O(m^3)$ as in lemma 6, we can compute $M_p(T[i];[k,l])$ and $D_p(T[i];[k,l])$ from $M_p(T[i_1];[k,l])$ and $D_p(T[i_1];[k,l])$. This idea can be easily extended to bulge loops and internal loops. For a multiple loop $t[i]$, suppose that $t[i_p]$ is the first internal child of $t[i]$, then from $M_p(T[i_p];[k,l])$ and $D_p(T[i_p];[k,l])$ we can first compute $A(T[i_p];[k,l])$ and $D(T[i_p];[k,l])$ in $O(m^2)$ time and then compute $A(F[i_1,i_p];[k,l])$, $D(F[i_1,i_p];[k,l])$, and $I(F[i_1,i_p];[k,l])$ in $O(m^2)$ time. Similarly $A_d(F[i_1,i_p];[k,l])$, $D_d(F[i_1,i_p];[k,l])$, and $I_d(F[i_1,i_p];[k,l])$ can also be computed in $O(m^2)$ time. This means that the computation time for $t[i]$ is $O(d_i m^2 + (dp_i-1)m^3)$. Therefore the total time is $O(\sum_{d_i>0} d_i m^2 + \sum_{dp_i>1}(dp_i-1)m^3)$ =$O(|T|m^2+(hl-1)m^3)$ = $O(nm^2 + hl\ m^3)$.

In practice, the space requirement would be a bottleneck. To reduce the space, the basic idea is straightforward: some computed values are no longer useful after they were used and the space used can be released.

For an internal node $t[i]$ with $d_i > 1$, $M_p(T[i];[k,l])$ and $D_p(T[i];[k,l])$ can be computed using $O(m^2)$ space since for the iteration of s from 1 to d_i, in order to compute the current values for s, $X(F[i_1,i_s];[k,l])$, we only need to maintain the previous values for $s-1$, $X(F[i_1,i_{s-1}];[k,l])$, where $X \in \{A,D,I\}$. If $dp_i > 1$ ($t[i]$ is a multiple loop), then the order we evaluate $t[i]$'s internal children is the child with maximum number of multiple loop descendant nodes first and then from left to right. This will guarantee that during the the computation only $\log(ml)$ nodes need $O(m^2)$ space. Therefore the total space requirement is $O(m^2 \log(ml))$. □

2.6 RNA structural pattern searching

We now briefly consider how to use the algorithm to search an RNA structure from a genome. Here the assumption is that the length n of the given RNA structure R is much smaller compared to the length m of a genome G.

In this situation, there is no meaning to compare R to every substring of G. We only need to consider substrings of G that have similar length of n. Here we can use a con-

Table 1. The two profiles of 5Srna+ and 5Srna- of 5S rRNA query used in HomoStRscan

```
a. The profile 5Srna+
5Srna+_B.sub_matured
TTTGGTGGCG auaGCGAAGA GgtcacACCC GTtcccatac
cgaacACGGa aGTtaagCTC TTCaGCgCCG ATGGTAGTcG
GGGGtttCCC CCtGTGAGAG TAGGaCGCCG CCAAGc
>
>
        (1  )      1       115      10
        (2  )     14        66       2
        (3  )     16        63       6
        (4  )     27        53       2
        (5  )     29        49       4
        (6  )     68       104      11
        (7  )     80        92       5
>
b. The profile 5Srna-
5Srna-_B.sub_matured_RCS
gCTTGGCGGC GtCCTACTCT CACaGGGGGa aaCCCCCgAC
TACCATCGGc GCtGAAGAGc ttaACttCCG Tgttcggtat
gggaACGGGT gtgacCTCTT CGCtatCGCC ACCAAA
>
>
        (1  )      2       116      10
        (2  )     13        49      11
        (3  )     25        37       5
        (4  )     51       103       2
        (5  )     54       101       6
        (6  )     64        90       2
        (7  )     68        88       4
>
```

stant k as a constraint such that only substrings of lengths less or equal to kn will be compared. In practice, k would be a small number less than 3. With this constraint, using a technique in [29], the time and space complexities of the algorithm can be further reduced to $O(n^2m)$ and to $O(nm)$.

We now assume that $k = 2$ and consider a sequence of length $3n = (k+1)n$ form G. From the above discussion, in $O(n^3)$ time and $O(n^2)$ space, we can compute scores of the optimal alignments starting from the first n locations of the sequence since starting from those locations the lengths of the substrings are at least $2n$. For the given genome, for every n locations, we can take a substring of length $3n$ and compute the scores of the optimal alignments. Therefore, the time complexity is $O(n^3m/n) = O(n^2m)$ and space complexity remains $O(n^2)$.

3 Results and Discussion

We reported here the applications of HomoStRscan in searching for 5S ribosomal RNA (rRNA) in three bacterial genome databases. We know that 5S rRNA or tRNA can be encoded in either the positive stranded sequence (PSS) or reverse complementary sequence (RCS) of the listed sequence in the genome database. To distinguish the difference of the structural feature of the query RNA encoded in the PSS and RCS using the same target sequence data we need to design two structural profiles for the query RNA.

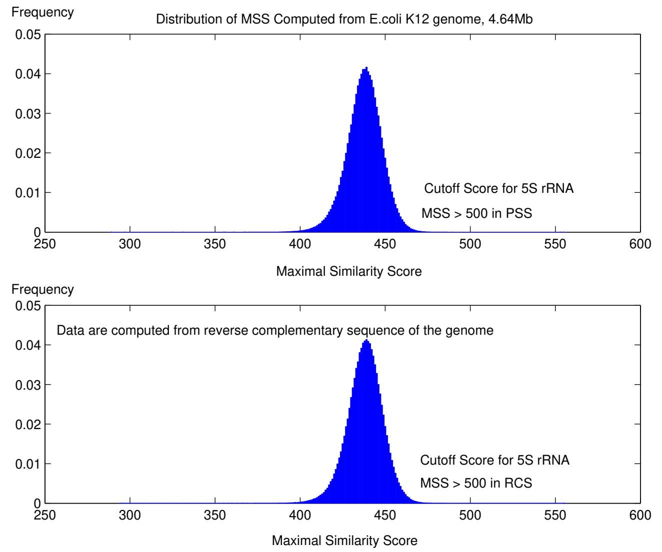

Figure 2: 5S rRNA MSS distributions computed from *E.coli* K12. The MSS scores computed from the PSS of the genomic sequence are shown in the top and those observations computed from the RCS are shown in the bottom. An expanded view of the high-scoring tails of the MSS distribution are shown in Fig. 3.

One is for the detection of those RNAs encoded in the PSS and the other is for those encoded in the RCS in scanning the same sequence. We know that the wobble base pairs, G:U and U:G will be reversed into A:C and C:A in the RCS. To consider the un-complementary, structural feature in the wobble and other non-canonical base pairs (e.g. A:G, G:G) we also need to design the score matrix of base pairs specifically in searching for query RNA encoded in either PSS or RCS, respectively.

The common secondary structure of 5S rRNA from eubacteria was reported based on a multiple sequence alignment of 436 5S rRNAs [25]. The query RNA used in HomoStRscan was composed of two parts, in which the first part is the primary sequence data and the second was the region table of the structural constraints in the folded secondary structure. Based on the secondary structure of *Bacillus subtilis (B.sub)* 5S rRNA [26] we design the two profiles, 5Srna+ and 5Srna- as the two query RNAs. The primary sequence in the query 5Srna- is the RCS of the sequence in the query 5Srna+ (see Table 1).

We know that some non-canonical base pairs, A:G, G:A, A:A, G:G are a unique structural feature in the secondary structure of 5S rRNAs. To characterize the conserved and unique structural feature we design two specific score matrices of base pairs, bp5s+ and bp5s- that are used in finding RNA homologue encoded in the PPS and RCS, respectively. They are shown in Table 2.

For *Escherichia coli (E.coli)* K12, eight 5S rRNAs were annotated in the *E.coli* genome in the database of Bacteria Complete Genomes

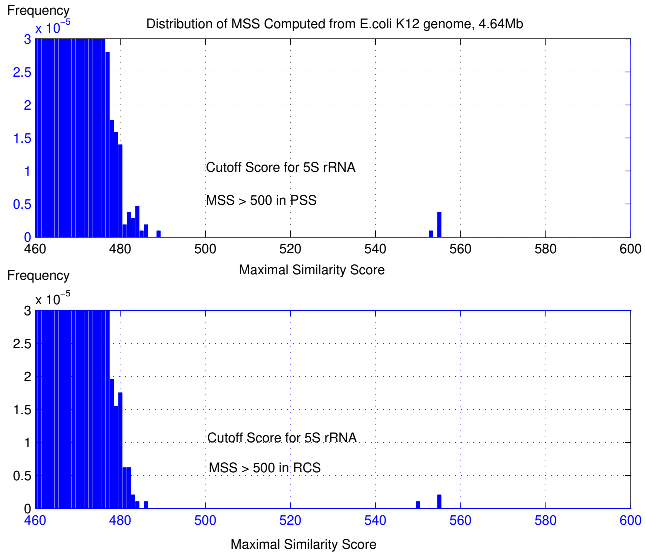

Figure 3: An expanded view of the high-scoring tails of the 5S rRNA MSS distribution computed from *E.coli* K12. For further details see the caption to Figure 2.

(http://www.ncbi.nlm.nih.gov:80/genomes/static/eub_g.html). Among them, five 5S rRNAs were encoded in the PSS and the other three were encoded in the RCS. The target genome has about 4.64 million bases (Mb). In the search for 5S rRNAs, we used the default parameters to compute the MSS distribution in the genome by HomoStRscan. The sample size of the MSS distribution computed from the PSS was 1,073,456 without including those overlapping segments having the same ending positions approximately. The observed distribution of MSS ranged from 289 to 555 (Fig. 2). The computed sample mean and the sample standard deviation (*std*) were 437.6 and 10.41, respectively. Using the cutoff score of MSS, 500 that was equal to the value of the sample mean plus 6 times of *std*, we found only 5 observations. All five homologous 5S rRNAs agree with those listed in the annotation table of *E.coli* K12 genome. Similarly, we had 969,775 observations of MSS computed in the RCS by 5Srna- and score matrix bp5s-. The observed distribution of MSS in RCS ranged from 295 to 555 (Fig. 2). The sample mean and *std* of MSS were 438.0 and 10.37. Using the same rule of selecting cutoff we also had the cutoff MSS = 500 that was equal to the value of the sample mean plus 6 times of *std*. Using that cutoff, we discovered only three observations in the RCS (see Table 3) and they agree completely with those listed in the annotation table. In the example, we showed high sensitivity/specificity ratios in HomoStRscan search. Both the sensitivity and the specificity ratios in the search are 100% in *E.coli* K12 genome (Fig. 3).

There were seven 5S rRNAs listed in the annotation table of the 2.82 Mb complete genome of *Staphylococcus aureus*

Table 2. The score matrices of base pairs, bp5s+ and bp5s- that are used in the 5S rRNA database search by HomoStRscan

a. score matrix bp5s+

	AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU	DD
AC	0	1	0	2	0	0	0	0	0	2	0	1	0	0	0	0	-4
AA	6	0	4	4	0	0	3	0	4	3	2	2	4	0	2	0	-4
AG	4	0	6	4	0	0	4	0	2	3	4	2	3	0	3	0	-4
AU	4	2	4	12	0	0	12	0	3	12	3	10	12	0	8	0	-4
CA	0	0	0	0	1	0	2	0	0	0	0	0	2	0	2	0	-4
CC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4
CG	3	0	4	12	2	0	12	0	3	12	4	8	12	0	10	0	-4
CU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4
GA	4	0	2	3	0	0	3	0	6	4	4	3	4	0	2	0	-4
GC	3	2	3	12	0	0	12	0	4	12	4	10	12	0	8	0	-4
GG	2	0	4	3	0	0	4	0	4	4	6	3	3	0	3	0	-4
GU	2	1	2	10	0	0	8	0	3	10	3	12	8	0	8	0	-4
UA	4	0	3	12	2	0	12	0	4	12	3	8	12	0	10	0	-4
UC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4
UG	2	0	3	8	2	0	10	0	2	8	3	8	10	0	12	0	-4
UU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4
DD	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4

b. score matrix bp5s-

	AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU	DD
AA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4
AC	0	12	0	10	8	3	8	2	0	10	0	2	8	3	0	2	-4
AG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4
AU	0	10	0	12	8	3	12	4	0	12	0	2	12	3	0	4	-4
CA	0	8	0	8	12	3	10	3	0	8	0	0	10	2	1	2	-4
CC	0	3	0	3	3	6	4	4	0	4	0	0	3	4	0	2	-4
CG	0	8	0	12	10	4	12	4	0	12	0	0	12	3	2	3	-4
CU	0	2	0	4	3	4	4	6	0	3	0	0	3	2	0	4	-4
GA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4
GC	0	10	0	12	8	4	12	3	0	12	0	2	12	4	0	3	-4
GG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-4
GU	0	2	0	2	0	0	0	0	0	2	0	1	0	0	0	0	-4
UA	0	8	0	12	10	3	12	3	0	12	0	0	12	4	2	4	-4
UC	0	3	0	3	2	4	3	2	0	4	0	0	4	6	0	4	-4
UG	0	0	0	0	1	0	2	0	0	0	0	0	2	0	1	0	-4
UU	0	2	0	4	2	2	3	4	0	3	0	0	4	4	0	6	-4
DD	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4

Table 3. The predicted 5S rRNAs in the genome of E.coli k12

rRNAs Location		Product	Prodicted	MMS +	MMS -
			MMS > mean+6*std	500	500
228756..228875	+	5S rRNA	228756..228875	555	
2724089..2724208	-	5S rRNA	2724089..2724208		555
3421059..3421179	-	5S rRNA	3421059..3421179		550
3421305..3421424	-	5S rRNA	3421305..3421424		555
3944324..3944443	+	5S rRNA	3944324..3944443	555	
4038097..4038216	+	5S rRNA	4038097..4038216	553	
4169216..4169335	+	5S rRNA	4169216..4169335	555	
4210619..4210738	+	5S rRNA	4210619..4210738	555	
Total number of observations with MMS > 500				5	
Total number of observations with MMS > 500					3

Table 4. The predicted 5S rRNAs in the genome of Staphylococcus aureus subsp. aureus MW2

rRNAs Location	Product		Prodicted MMS > mean+6*std	MMS + 508	MMS - 509
496386..496500	+	5S rRNA	496386..496500	546	
534799..534913	+	5S rRNA	534799..534913	546	
540641..540755	+	5S rRNA	540641..540755	546	
545852..545966	+	5S rRNA	545852..545966	546	
1959247..1959361	-	5S rRNA	1959247..1959361		546
2137179..2137293	-	5S rRNA	2137179..2137293		546
2250748..2250862	-	5S rRNA	2250748..2250862		546

Total number of observations with MMS > 508 4
Total number of observations with MMS > 509 3

Table 5. The predicted 5S rRNAs in the genome of Streptococcus agalactiae 2603V/R by HomoStRscan

rRNAs Location	Product		MMS > mean+6*std	505 +	505 -
16411..17917	+	16S rRNA			
18234..21136	+	23S rRNA			
Predicted	+	5S rRNA	21211..21326	539	
22242..23748	+	16S rRNA			
24065..26967	+	23S rRNA			
Predicted	+	5S rRNA	27042..27157	539	
91219..92725	+	16S rRNA			
93042..95944	+	23S rRNA			
Predicted	+	5S rRNA	96019..96134	539	
165248..166754	+	16S rRNA			
167071..169973	+	23S rRNA			
170027..170188	+	5S rRNA	170048..170163	539	
250375..251881	+	16S rRNA			
252198..255100	+	23S rRNA			
Predicted	+	5S rRNA	255175..255290	539	
348582..350088	+	16S rRNA			
350405..353307	+	23S rRNA			
353361..353522	+	5S rRNA	353382..353497	539	
417726..419232	+	16S rRNA			
419549..422451	+	23S rRNA			
Predicted	+	5S rRNA	422526..422641	539	

Total number of observations with MMS > 505 7
Total number of observations with MMS > 505 0

subsp. aureus MW2 (MW2). Using the same parameters, score matrices and query RNAs as those we used in finding 5S rRNAs of *E.coli* genome, we computed the MSS distributions in both the PSS and RCS by HomoStRscan. We had 652,567 and 586,125 observations in the two MSS distributions of PSS and RCS, respectively. The MSS scores ranged from 301 to 546 in PSS and from 302 to 546 in RCS, respectively. And their sample means were 440.8 and 440.9, and the sample standard deviations were 11.32 and 11.41, respectively. The cutoff MSS for discovering 5S rRNAs in *MW2* genome were selected by the same rule that was used in *E.coli* genome (cutoff = mean + 6*std). Thus we had cutoff MSS = 508 in PSS and MSS = 509 in RCS. Using the two cutoff MSSs we detected four 5S rRNAs in PSS and three in RCS of MW2 genome (see Table 4). This agrees completely with the data listed in the databanks. The computed sensitivity and specificity ratios for finding 5S rRNAs are also 100% in the genome MW2.

Using the same approach and same parameters, we found seven 5S rRNAs encoded in the PSS of the genome of *Streptococcus agalactiae* 2603V/R. The size of the genome *Streptococcus agalactiae* 2603V/R was 2.16 Mb and only two 5S rRNAs encoded in the PSS were noted publicly in the genome database. In addition to those two 5S rRNAs we also found five additional 5S rRNAs in the PSS using the same cutoff of MSS. No 5S rRNAs were found in the RCS by the same cutoff of MSS we used above (Fig. 4). All the predicted seven 5S rRNAs had the same value of MSS score (MSS = 539). Their MSS scores were above the sample means by 9 times the std. This is very statistically significant (Fig. 5). The five 5S rRNAs not presented in the literature are listed in Table 5.

The examples presented here indicate various aspects in using HomoStRscan search tool for detection of homologues of 5S rRNAs in genomes. The procedure used here is also suitable to search for other homologous structural RNAs in general. In searching for tRNAs, we made two query RNAs in either PSS and RCS, respectively. One has the regular size of the variable loop and the other has a bigger variable loop. Though we use the same score matrix for the unpaired bases as we used in the search for 5S rRNAs, we set two new score matrices of base pairs in the tRNA search because those distinct non-canonical base pairs in 5S rRNA, such as A:G, G:A, and A:A, are not frequently found in tRNA. We computed the MSS distribution in the complete genomic sequence data for each tRNA query by HomoStRscan. The tRNAs were predicted using a cutoff MSS that was 5 times the std greater than the sample mean of MSS. The sensitivity/specificity ratios computed from searching the tested complete genome of eubacteria are about 98%. Also our method finds some tRNAs that are not listed in the current database of genome (data are not included).

We have developed a computational method, HomoStRscan, in finding homologue of structured RNAs from the complete genomic sequence. In general, our method can be used to search for any RNA segments with the established secondary structure in the nucleic sequence. The predicted homologous RNAs are predicted by a robust statistical inference from the computed MSS distribution. Our computational experiments for several complete genomic sequences indicate that HomoStRscan will detect 100% of the true 5S rRNAs and give zero false positive in the search. Based on the general searching method presented here, we expect to improve our method to be more efficient with less computing cost for structured RNAs with additional specific

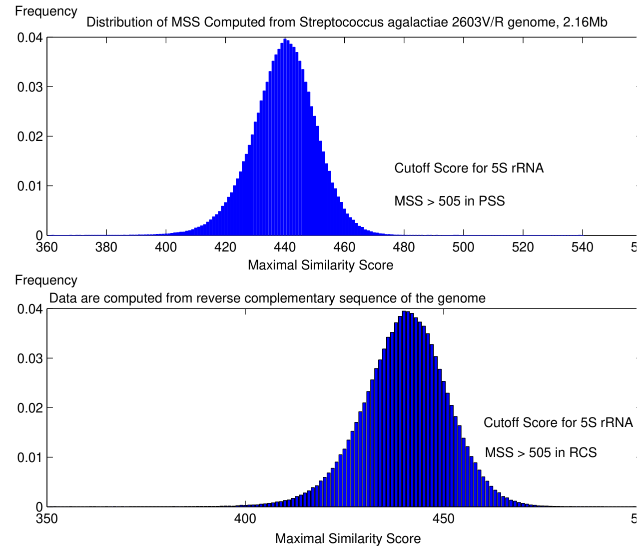

Figure 4: 5S rRNA MSS distributions computed from *Streptococcus agalactiae* 2603V/R. The MSS scores computed from the PSS of the genomic sequence are shown in the top and those observations computed from the RCS were shown in the bottom. An expanded view of the high-scoring tails of the MSS distribution are shown in Fig. 5.

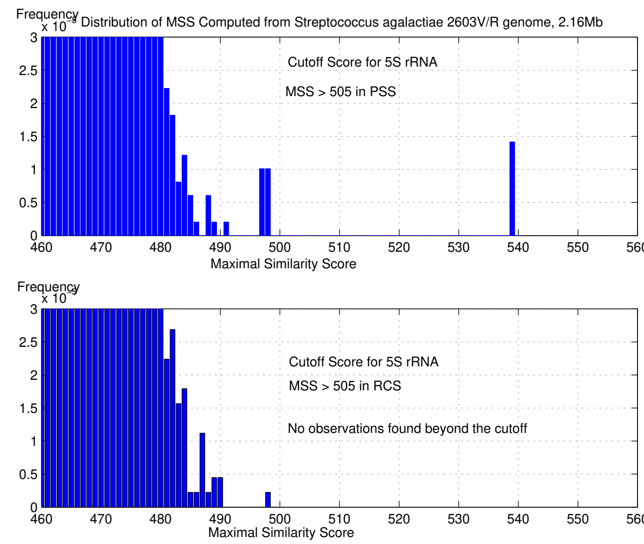

Figure 5: An expanded view of the high-scoring tails of the 5S rRNA MSS distribution computed from *Streptococcus agalactiae* 2603V/R. For further details see the caption to Figure 4.

structural elements.

Acknowledgments

The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government. K. Zhang is supported in part by the Natural Sciences and Engineering Research Council of Canada under research grant no. OGP0046373, a research fellowship from Simon Fraser University and a Sharcnet research fellowship.

References

[1] Simons, R.W. and M. Grunberg-Manago, eds. *RNA Structure and Function* Cold Spring Harbor Lab. Press, New York, 1998.

[2] G. Storz, An expanding universe of noncoding RNAs. *Science* **296,** 2002, pp. 1260-1263.

[3] S.R. Eddy, Non-coding RNA genes and the modern RNA world.. *Nature Rev Genet* **2,** 2001, pp. 919-929.

[4] M.H. Malim, J. Hauber, S.-Y. Le, J.V. Maizel Jr., and B.R. Cullen, The HIV-1 rev trans-activator acts through a structured target sequence to activate nuclear export of unspliced viral mRNA, *Nature* **338,** 1989, pp. 254-257.

[5] P.M. Macdonald, and C.A. Smibert, Translational regulation of maternal mRNAs. *Curr Opin Genet Dev* **6,** 1996, pp. 403-407.

[6] J.S. Mattick, Non-coding RNAs: the architects of eukaryotic complexity. *EMBO Rep.* **2,** 2001, pp. 986-991.

[7] I. Brierley, P. Digard, and S.C. Inglis, Characterization of an efficient coronavirus ribosomal frameshifting signal requirement for an RNA pseudoknot. *Cell* **57,** 1989, pp. 537-547.

[8] J. Dubnau, and G. Struhl, RNA recognition and translational regulation by a homeodomain protein. *Nature* **379,** 1996, pp. 694-699.

[9] C.U.T. Hellen, and P. Sarnow, Internal ribosome entry sites in eukaryotic mRNA molecules. *Genes & Development* **15,** 2001, pp. 1593-1612.

[10] M.W. Hentze, S.W. Caughman, J.L. Casey, D.M. Koeller, T.A. Rouault, J.B. Harford, and R.D. Klausner, A model for the structure and functions of iron-responsive elements. *Gene* **72,** 1988, pp. 201-208.

[11] A. Krol, Evolutionarily different RNA motifs and RNA-protein complexes to achieve selenoprotein synthesis. *Biochimie* **84,** 2002, pp. 765-774.

[12] D.E. Draper, Strategies for RNA folding. *Trends Biochem Sci* **21,** 1996, pp. 145-149.

[13] A. Laferriere, D. Gautheret and R. Cedergren, An RNA pattern matching program with enhanced performance and portability. *Comput. Appl. biosci.* **10,** 1994, pp. 211-212.

[14] B. Billoud, M. Kontic, and A. Viari, Palingol: a declarative programming language to describe nucleic acids' secondary structures and to scan sequence database. *Nucleic Acids Res.* **24,** 1996, pp. 1395-1403.

[15] 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,** 2003, pp. 3608-3612.

[16] M. Dsouza, N. Larsen and R. Overbeek, earching for patterns in genomic data. *Trends Genet.* **13,** 1997, pp. 497-498.

[17] T.J. Macke, D.J. Ecker, R.R. Gutell, D. Gautheret, D.A. Case and R. Sampath, RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Res.* **29,** 2001, pp. 4724-4735.

[18] D. Gautheret and A. Lambert, Direct RNA motif definition and identification from multiple sequence lignments using secondary structure profiles. *J. Mol Biol.* **313,** 2001, pp. 1003-1011.

[19] G.A. Fichant and C. Burks, Identifying potential tRNA genes in genomic DNA equences. *J. Mol Biol.* **220,** 1991, pp. 659-671.

[20] A. Pavesi, F. Conterio, A. Bolchi, G. Dieci and S. Ottonello, Identification of new eukaryotic tRNA genes in genomic DNA databases by a multistep weight matrix analysis of transcriptional control regions. *Nucleic Acids Res.* **22,** 1994, pp. 1247-1256.

[21] S.R. Eddy and R. Durbin, RNA sequence analysis using covariance models. *Nucleic Acids Res.* **22,** 1994, pp. 2079-2088

[22] 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,** 1997, pp. 955-964.

[23] S.Y. Le, K. Zhang and J.V. Maizel, Jr. RNA molecules with structure dependent functions are uniquely folded. *Nucleic Acids Res.* **30,** 2002, pp. 3574-3582.

[24] G.D. Collins, S.Y. Le and K. Zhang, A new algorithm for computing similarity between RNA structures. *Information Sciences* **139,** 2001, pp. 59-77.

[25] M. Szymanski, M.Z. Barciszewska, J. Barciszewski and V.A. Erdmann, 5S ribosomal RNA database Y2K. *Nucleic Acids Res.* **28,** 2000, pp. 166-167.

[26] S.Y. Le, K. Zhang and J.V. Maizel, Jr. A method for predicting common structures of homologous RNAs. *Computers and Biomedical Research* **28,** 1995, pp. 53-66.

[27] K. Zhang, Computing similarity between RNA secondary structures. *Proceedings of IEEE International Joint Symposia on Intelligence and Systems*, Rockville, Maryland, May 1998, pp. 126-132.

[28] B. Ma, L. Wang, and K. Zhang, "Computing similarity between RNA structures", *Theoretical Computer Science*. **276**, 2002, pp. 111-132.

[29] T. Jiang, G. H. Lin, B. Ma and K. Zhang, A general edit distance between RNA structures. *Journal of Computational Biology*, **9,** 2002, pp. 371-388.

[30] O. Gotoh, An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162,** 1982, pp.705-708.

[31] Z. Wang and K. Zhang, Alignment between RNA structures. *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science, Springer-Verlag's Lecture Notes in Computer Science 2136*, 2001, pp. 690-702.

[32] F. Corpet and B. Michot, RNAlign program: Alignment of RNA sequences using both primary and secondary structures. *Comput. Applic. Biosci.* **10,** 1994, pp. 389-399.

[33] H.P. Lenhof, K. Reinert, M. Vingron, A polyhedral approach to RNA sequence structure alignment. *J. Comput. Biol.* **5,** 1998, pp. 517-530.

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