2013 IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum

Secondary Structure Predictions for Long RNA Sequences Based on Inversion Excursions and MapReduce

Daniel T. Yehdego[1], Boyu Zhang[2], Vikram K. R. Kodimala[1], Kyle L. Johnson[1], Michela Taufer[2], Ming-Ying Leung[1]

[1] The University of Texas at El Paso
El Paso, Texas 79968
{dtyehdego,vkodimala,kljohnson,mleung}@utep.edu

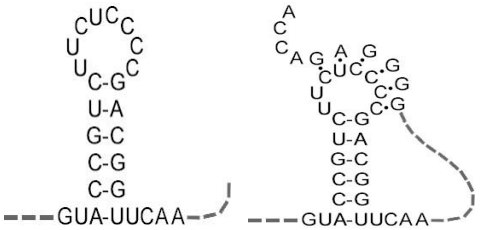
[2] University of Delaware
Newark, Delaware 19716
{bzhang,taufer}@udel.edu

***Abstract*—Secondary structures of ribonucleic acid (RNA) molecules play important roles in many biological processes including gene expression and regulation. Experimental observations and computing limitations suggest that we can approach the secondary structure prediction problem for long RNA sequences by segmenting them into shorter chunks, predicting the secondary structures of each chunk individually using existing prediction programs, and then assembling the results to give the structure of the original sequence. The selection of cutting points is a crucial component of the segmenting step. Noting that stem-loops and pseudoknots always contain an inversion, i.e., a stretch of nucleotides followed closely by its inverse complementary sequence, we developed two cutting methods for segmenting long RNA sequences based on inversion excursions: the centered and optimized method. Each step of searching for inversions, chunking, and predictions can be performed in parallel. In this paper we use a MapReduce framework, i.e., Hadoop, to extensively explore meaningful inversion stem lengths and gap sizes for the segmentation and identify correlations between chunking methods and prediction accuracy. We show that for a set of long RNA sequences in the RFAM database, whose secondary structures are known to contain pseudoknots, our approach predicts secondary structures more accurately than methods that do not segment the sequence, when the latter predictions are possible computationally. We also show that, as sequences exceed certain lengths, some programs cannot computationally predict pseudoknots while our chunking methods can. Overall, our predicted structures still retain the accuracy level of the original prediction programs when compared with known experimental secondary structure.**

***Keywords*-Pseudoknots, RNA segmentation, Hadoop, Performance analysis, Prediction accuracy.**

I. Introduction

RNA is made up of four types of nucleotide bases, i.e., adenine (A), cytosine (C), guanine (G), and uracil (U) and play important roles in many biological processes including gene expression and regulation. Many viral genomes are also made up of RNA. Secondary structural elements in RNA are crucial to their functionality and can be separated into stem-loops and pseudoknots (see Figure 1). In both elements, it is well known that an adenine binds with a uracil and a cytosine binds with a guanine. Any stem-loop or pseudoknot contains an inversion, which is a string of nucleotides followed closely by its inverse complementary sequence. Figure 2 shows an example of an inversion, with the 6-nucleotide string "ACCGCA" followed by its inverse complementary sequence "UGCGGU" after a gap of 3 nucleotides.



(a) Stem-loop (b) Pseudoknot

Figure 1. Two basic elements in RNA secondary structures.

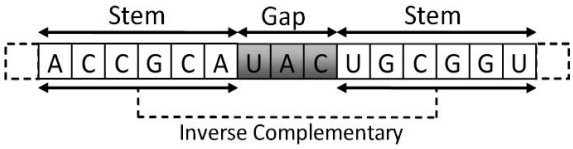


Figure 2. Inversion with stem length 6 and gap size 3.

Most secondary structure prediction algorithms are based on the minimization of a free energy (MFE) function and the search for a thermodynamically most stable structure starts from the whole RNA sequence. The search for a structure with global minimal free energy may be memory and time demanding, especially for long sequences and for pseudoknot predictions. At the same time, minimal energy configurations may not be most favorable for carrying out the biological functions of RNA, which often require the RNA to react and bind with other molecules (e.g., RNA binding proteins). Our current work suggests that local structures formed by pairings among nucleotides in close proximity and based on local minimal free energies, rather than the global minimal free energy, may correlate better with the real molecular structure of long RNA sequences. This hypothesis has yet to be supported by more detailed

978-0-7695-4979-8/13 $26.00 © 2013 IEEE
DOI 10.1109/IPDPSW.2013.109

experimental evidence. However, if proven correct, it will open the door to a new generation of programs based on segmenting long RNA sequences into shorter chunks, predicting the secondary structures of each chunk individually, and then assembling the prediction results to give the structure of the original sequence.

The selection of cutting points in the original RNA sequence is a crucial component of the segmenting step. We propose to approach the problem by searching for and by cutting around inversion excursions. We consider two alternative cutting methods, the centered and optimized methods. Both methods identify regions in the sequence with high concentrations of inversions and avoid cutting into these regions. In the centered method, the longest spanning inversion clusters are centered in the chunks, while in the optimized method, the number of bases covered by inversions is maximized. The prediction of secondary structures for different chunks can be performed in parallel, thus benefiting from parallel computing systems and paradigms. In this paper we use a MapReduce framework, i.e., Hadoop, to extensively explore meaningful combinations of stem length and gap size for the predictions and identify correlations between sequence chunking and prediction accuracy. For each combination, we evaluate the capability of the centered and optimized methods to retain the secondary structure prediction accuracy using several existing prediction algorithms. We compare the accuracy values with a naïve chunking method that does not use knowledge on inversions and with the predictions of the same algorithms when using the whole sequences (no chunking is used). Our datasets are restricted to the experimentally found secondary structures including pseudoknots. Here we use a dataset of 12 non-homologous RNA sequences with known structures available in the RFAM database.

The rest of this paper is organized as follows: Section 2 presents relevant background and related work. Section 3 discusses our cutting methods. Section 4 shows the accuracy retention capabilities of the cutting methods for the pknotsRG algorithm, leading to the conclusion in Section 5 that predictions obtained with these cutting methods can outperform those obtained on the whole RNA sequence without segmentation.

II. BACKGROUND AND RELATED WORK

A. RNA secondary structure predictions

The 3D structure of an RNA molecule is often the key to its function. Because of the instability of RNA molecules, experimental determination of their precise 3D structures is a rather costly process. Useful information about the molecule can be gained from knowing its secondary structure [1]. As noted above, all RNA secondary elements can be classified into stem-loops and pseudoknots (see Figure 1). Both secondary structure elements have been implicated in important biological processes like gene expression and gene regulation [2]. Development of mathematical models and computational prediction algorithms for stem-loop structures began in the early 1980‘s [3], [4], [5]. Pseudoknots, because of the extra base-pairings involved, must be represented by more complex models and data structures which require large amounts of memory and computing time to obtain the optimal and suboptimal structures with minimal free energies [6], [7]. To overcome the tremendous demand on computing resources that pseudoknot prediction poses, alternative algorithms have been proposed that restrict the types of predicted pseudoknots. Yet, most programs available to date for pseudoknot structure prediction can only process sequences of limited lengths on the order of several hundred nucleotides. Thus, these programs cannot be applied directly to long RNA molecules such as the genomic RNA in viruses, which may be thousands of bases in length.

In our previous work, we proposed to approach this problem using three steps: (1) cut a long RNA sequence into shorter non-overlapping chunks; (2) predict the secondary structures of each chunk individually by distributing them to different processors on a Condor grid and (3) assemble the prediction results to give the structure of the original sequence [8]. In our past effort we performed an exhaustive search for all the possible ways to cut a sequence. In the current study, we move away from the exhaustive search and apply cutting methods using statistical information on inversions. Our new approach outperforms our previous work in terms of computing efficiency and confirms the capability of appropriate sequence segmentation methods to retain RNA secondary structure prediction accuracy. Preliminary results were recently shown in a poster [9] and are here extended.

B. MapReduce and Hadoop

The MapReduce (MR) paradigm is a parallel programming model that facilitates the processing of large distributed datasets. It was originally proposed by Google to index and annotate data on the Internet [10]. In this paradigm, the programmer specifies two functions: map and reduce. The map function takes as input a key and value pair, and outputs a list of intermediate key and value pairs which may be different from the input. The reduce function takes as input a key and values pair, and outputs a list of values. Note that the input values to reduce is the list of all the values associated with the same key. MR is appealing to scientific problems because of the simplicity of programming, the automatic load balancing and failure recovery, and the scalability. It has been widely adapted for many bioinformatics applications, e.g., Hong et al. designed a RNA-Seq analysis tool for the estimation of gene expression levels and genomic variant calling [11]; and Langmead et al. designed a next-generation sequencing tool based on MR Hadoop [12]. To the best of our knowledge, this work is the first one to adapt MR into secondary structure predictions of long RNA sequences. Preliminary work on the reasoning behind adapting RNA

secondary structure predictions to the MR paradigm can be found at [13].

III. METHODOLOGY

A. Inversions in RNA sequences

Given a long RNA sequence, we identify regions with high concentrations of inversions by using an adapted version of the "Palindrome" program in the EMBOSS package [14], which is a free open-source software analysis package. Two main reasons for adapting the EMBOSS palindrome program are the fact that the program works correctly on DNA but not RNA sequences and, in future work, we will allow for G-U pairing—a feature that is not available in the EMBOSS Palindrome program. Our adapted program called InversFinder, is written in Java and is available for download at http://rnavlab.utep.edu. InversFinder requires a text file containing the RNA sequence in FASTA format as input. The minimum stem length L and maximum gap size G of the inversion are parameters specified by the user.

B. Inversion excursion plot for RNA sequences

Our cutting methods rely on a general excursion approach first formulated in [15], which has already been applied to a variety of sequence analysis problems but not to RNA secondary structure predictions. In many bioinformatics applications, the problem calls for identifying high concentration regions of a certain property in the nucleotide bases of biomolecular sequences. For example, replication origins in viral genomes have been predicted by looking for regions that are unusually rich in the nucleotides A and T in DNA sequences [16]. In this paper, we follow the same approach for RNA sequences, but our focus is whether or not the nucleotide base is found inside an inversion. We refer to the excursions generated by this property as "inversion excursions". The excursion method requires assigning to each nucleotide a positive score if it is a part of an inversion (including the two stems and the gap between them), and a negative score if it does not. We go through the entire nucleotide sequence accumulating the scores to form inversion excursions.

```
. . GCGAUUGCCGUCAGGCAAUACU. .
. . 0001111111111111111000. .
```

Figure 3. Inversion with stem length 6 and gap size 3.

To facilitate the analysis, we use a parsing program to convert an RNA sequence into a binary sequence with the same length. If a nucleotide base is included in an inversion identified by the InversFinder program, it is given a value of "1"; if not, it is assigned a value of "0", as illustrated in Figure 3. Each "1" in the binary sequence is given a score of 1, and each "0" a negative score of s which is determined as follows. We consider the binary sequence as a realization of a sequence of independent and identically distributed (i.i.d.) random variables, $X_1, X_2, ..., X_n$, where n is the length of the RNA sequence (i.e., number of bases). These random variables take values either 1 or s. Let $p = Pr(X_i = 1)$ and $q = 1 - p = Pr(X_i = s)$. The parameter p is naturally estimated by the percentage of bases contained in one or more inversions in the RNA sequence, i.e., the percentage of "1"s of the binary sequence. We require that the expected score per base $\mu = p + qs$ to be negative. As done in [16] and other applications, the value of s can be conveniently selected by giving μ a value of -0.5, and then determining the value of s according to Equation 1.

$$s = \left\lfloor \frac{\mu - p}{q} \right\rfloor \tag{1}$$

The excursion score E_i at position i of the sequence is defined recursively as in Equations 2 and 3.

$$E_0 = 0 \tag{2}$$

$$E_i = max(E_{i-1} + X_i, 0) \; for \; 1 \leq i \leq n \tag{3}$$

An excursion starts at a point i where E_i is zero, continues with a number of rising and falling stretches of positive values, and ends at $j > i$ where j is the next position with $E_j = 0$. The score then stays at zero until it becomes positive again as the begin of the next excursion. Plotting the excursion scores along the nucleotide positions of the RNA sequence offers an effective visualization of how inversion concentrations vary along the sequence. This plot can serve as a guide for choosing the cut-points for the segmentation process. Figure 4 shows an example of an excursion plot. Rising stretches in the plot indicates the presence of inversions.

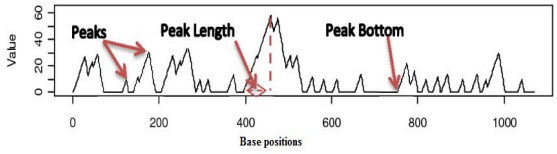


Figure 4. An excursion plot with peaks, peak bottoms, and peak lengths.

After generating the excursion plot, we identify the positions, called peaks, where the excursion scores are local maxima. Then, the bottom of each peak (the last position with zero excursion score right before the peak) is located. After that, the length of the peak (the location difference between a peak and its peak bottom) is calculated. Note that since we require chunk lengths to be smaller than a prescribed maximum c, peak lengths greater than c have to be flagged and analyzed separately. Figure 4 also shows examples of peaks, peak bottoms, and peak lengths. To be used with the centered and optimized cutting methods, the peaks are sorted in decreasing order based on their excursion scores.

C. Methods for sequence segmentation

1) Centered cutting method: The centered method cuts the sequence by identifying inversions and building the chunks around them. Our objective is to segment the RNA sequence in such a way to avoid losing structure information as much as possible by centering the longest spanning inversion clusters in the chunks. After peaks are identified, they are sorted in decreasing order of their excursion values. The peak with the highest excursion value is considered first. Then the second highest peak is considered and so on. The algorithm stops either when all the peaks are exhausted or when all the inversion regions of the sequence (i.e., all "1"s in the binary sequence) have been included in the chunks, whichever occurs first. Overlapping chunks are adjusted so that any nucleotide base is captured by only one chunk, with priority given to the peak with a higher excursion score.

For each of the selected peak, the positions of the inversions or peak length positions are centered within the max chunk-length of c bases where c is defined by the user. We start at the bottom of this peak and follow the excursion until it returns to 0 the very next time and locate the position of the very last peak before the excursion returns to 0. We take the sequence segment between the peak bottom and the position of the very last peak and place the sequence segment in the center of the chunk as illustrated in Figure 5. Suppose this centered segment contains x nucleotide bases. If $(c-x)$ is even, then the resulting chunk will have $(c-x)/2$ bases on each side of the centered segment. If $(c-x)$ is odd, then we will adjust the lengths on each side to the integers below and above $(c-x)/2$, allowing one side (chosen at random) to have one more nucleotide base than the other.

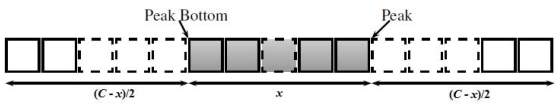


Figure 5. Centered cutting method where x = peak length.

As an example, we applied the method to an RNA sequence RF00209_A from RFAM database with sequence length 379 bases. As shown in Figure 6, the sequence is segmented into six chunks using the centered cutting method. These six segments cover the entire sequence. Labels 1 through 6 in Figure 6 represent the six segments with decreasing order of peak excursion scores. After the peak scores are sorted, the peak with the highest excursion score is considered first. In this example, we use the maximum chunk-length $c = 100$. The highest peak is found at position 297 with peak bottom at 257. As there are other inversions after the highest scoring peak, we follow the entire excursion to the end at position 356. Locating the last peak in this excursion at 343, we center the sequence segment from 257 to 343 to produce the chunk covering the 100 positions from 250 to 349. After this, the second highest scoring peak at position 54 is considered and the above procedure is repeated. This time, the peak bottom is at position 19 and the last peak before the end of this excursion is at position 70. Centering the segment consisting of positions $19-70$ in a chunk of 100 would require 24 positions on each side, extending the chunk beyond the beginning of the sequence. We therefore adjust the chunk to start at position 1 instead. Note that during the segmentation process, we might get a chunk that overlaps with previously established chunks. In those cases, we have to reconcile the situation by reducing one of the chunk lengths. For example, after establishing the first two chunks (labeled 1 and 2 in Figure 6), the next highest peak to be processed is at position 114, with peak bottom at position 89. Centering this peak will produce a chunk from positions 52 to 151, overlapping with chunk 2. We resolve such conflicts by giving priority to the chunk with the higher number of bases within completely contained inversions. With this rule, we give priority to chunk 2, and reduce chunk 3 to positions $101-151$. The process continues for the remaining chunks 4, 5, and 6.

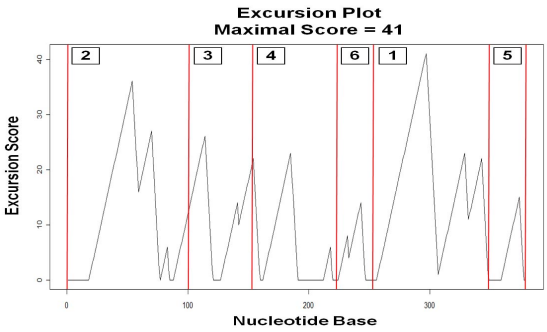


Figure 6. Chunks obtained using the centered method.

2) Optimized cutting method: In the optimized method, cutting points are decided by choosing a segment containing the peak in an optimal position that yields the highest inversion scores for the segment; the score is defined as the total number of nucleotide bases contained in the inversions that are entirely within the chunk. For example, consider a peak with peak length spanning the nucleotide bases between i and j and then all the chunks of size c covering this peak. That is, all segments with length c starting between positions $j-(c-1)$ and $i+(c-1)$ are considered (see Figure 7). The chunk with the maximum inversion score is then selected. Beginning with the highest peak, the above process is repeated until either all the peaks are utilized or all the inversion regions of the sequence are contained in the established chunks, whichever occurs first. When chunks overlap, the cut points are adjusted in a similar way to that principle described for the centered method. The optimized method ensures that peak length positions are

included within a chunk, but not necessarily in the center of the chunk.

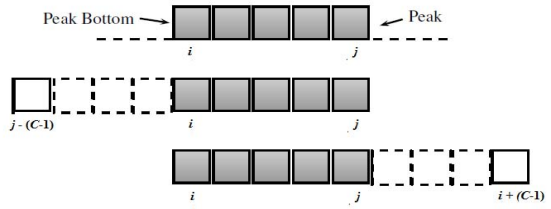


Figure 7. Chunks by optimized method with peak spanning positions i-j.

As an example, we applied the optimized method to the same RF00209_A RNA sequence file from the RFAM database, as shown in Figure 8. The optimized method produced only 5 chunks covering all except the first 18 positions of the sequence. It can be seen from Figure 8 that

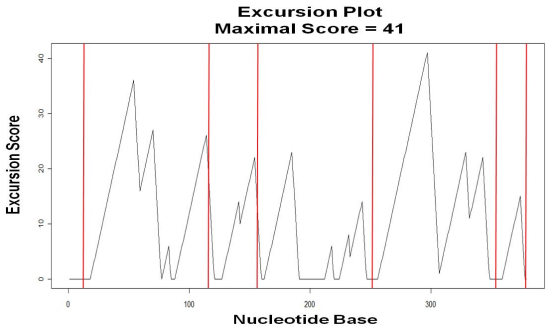


Figure 8. Chunks obtained using the optimized method.

cuts into those sequence segments with rising excursions scores preceding the peaks are avoided by this method. Also, the chunks produced by the optimized method cover only 96.3% of the sequence, leaving out those parts of the sequence where no inversions are found. Therefore, wasting of computing resources is minimal in the optimized method.

3) Regular cutting method: The regular cutting method is the most convenient method of segmentation and is used as a reference method in this paper. This method simply cuts the nucleotide sequence regularly into chunks of a specified maximum chunk-length c until the sequence is exhausted. For example, with $c = 100$, the sequence RF00209_A from the RFAM database with 379 bases will be cut into four chunks made up of nucleotide positions $1-100$, $101-200$, $201-300$, and $301-379$. This method can easily lose important structure information.

D. Structure prediction and assembly

We cut an RNA sequence into chunks and predict each chunk in parallel before merging the predictions into the complete secondary structure. The overall workflow consists of the parallel chunking and prediction, the reconstruction of the secondary structure, and a possible comparison with experimental results or secondary structure predictions obtained from the sequence used as a whole. This workflow naturally fit into the MR paradigm, for which each map function cuts the sequence in chunks and performs the prediction on one or multiple chunks using existing prediction algorithms and their associated programs. Here we use four popular secondary-structure prediction programs with pseudoknot prediction capabilities, i.e., pknotsRG [17], PKNOTS [6], HotKnots [18], and NUPACK [19]. As these widely used programs have sequence length limits of up to a few hundred bases only, chunking approaches will be useful when the programs are applied to analyze large RNA molecules. The partial predictions of the chunks are assembled into the whole secondary structure predictions by concatenating the predicted structures (reduce function). Note that two consecutive chunks do not overlap in our current work and thus the reduce function glues the chunk predictions together. Multiple reductions glue together the chunks resulting from the same sequence and using a given stem length, gap length, and prediction algorithm. More specifically, if we define this approach in a MR language, the input to each map function is a $< k1, v1 >$ value pair, in which $k1$ is a sequence identifier, and $v1$ is the chunking approach (e.g., stem length, gap length, and prediction algorithm). Each map function outputs the list of $< k2, v2 >$ pairs as intermediate output to reduce. The $k2$ is the id of the whole secondary structure to which the predicted chunk belongs, and $v2$ is the predicted secondary structure of the chunk. All the values associated with the same key, i.e., $< k2, list(v2) >$, are passed to a reduce function that reconstructs the whole secondary structure of the sequence using all the $v2$ (predicted chunk structures) associated with the same $k2$.

E. Assessment of segmentation methods

In order to assess whether the experimental secondary structure can be predicted accurately after sequence segmentation and to what degree, we use three metrics: (1) accuracy chunking (AC) is the accuracy of the predicted structure assembled from the chunks when compared with the experimental secondary structure; (2) accuracy whole (AW) is the accuracy of the predicted structure obtained from the whole sequence when compared with the experimental secondary structure; and (3) accuracy retention (AR) is the ratio between AC and AW. Ideally we would like to use AR to capture the capability of the segmentation method to outperform the prediction accuracy using the sequence as a whole. However, several codes predicting pseudoknots cannot process whole sequences with more than 200 nucleotides due to memory limitations. Thus, for these sequences AW is not available and we rely on AC only to assess the accuracy retention.

AC and AW are given by the percentage agreement of the

predicted structure with the known real structure calculated as:

$$\frac{100 * [a + 2 * b]}{n} \quad (4)$$

where a and b respectively represents the number of unpaired bases and the number of base pairs in common between the predicted and real structures, and n is the length of the RNA sequence.

A large AC value (i.e., close to 1) for a predicted structure means that the chunk-based predicted structure is similar to the experimental structure; a large AR value (greater than 1) means that the chunk-based prediction is more accurate than that obtained by predicting the secondary structure with the sequence as a whole; a large AW values (AW close to 1) means that the structure predicted as a whole is similar to the experimental structure. Various statistical tests provided by the R package [20] are applied in the accuracy analysis for the different cutting methods.

IV. RESULTS AND DISCUSSION

A. Test set-up

We ran the MR framework on a cluster composed of 8 dual quad-core compute nodes (64 cores), each with two Intel Xeon 2.50 GHz quad-core processors and a high-speed DDR Infiniband interconnect for application and I/O traffic. Our implementation is based on Hadoop 0.20.2. We used four well-known prediction codes that are able to capture pseudoknots, i.e., pknotsRG [17], HotKnots [18], PKNOTS [6], and NUPACK [19]. As we will see in the following discussion, for the longer sequences ($length > 200$), some prediction codes (i.e., NUPACK, PKNOTS) cannot predict the whole sequence because of resource requirements. In such cases, our chunk-based approach is the only approach available. We used a dataset of 12 sequences in RFAM database [21], each containing a pseudoknot and for which we know their secondary structures experimentally. The lengths of the sequences range from 79 to 451 bases. Note that there are not large datasets of experimentally determined RNA secondary structures including pseudoknots, and to the best of our knowledge the one used in this paper is one of the few available to the public for free.

For the regular cutting method, we used a fixed chunk length of 60 bases for each RNA sequence. Note that the length 60 is the proper tradeoff for our dataset between being too short, which may result in more information lost, and being too long, which may cause many sequences be cut into one single chunk. We have checked the effect of max chunk-length on MAR by increasing the max chunk-length from 60-150 in 10 nucleotide base increments. With the centered and optimized cutting methods, the chunks obtained depended on the value of inversion parameters minimum stem length L and maximum gap size G: each chunk is not longer than 60 bases. We also allowed a range of L values from 3 to 8, and of G values from 3 to 8, resulting in a total of 36 (L, G) pairs. The values are selected because they are scientifically meaningful. Note that for some (L, G) pairs no inversion may be found; for these cases the cutting methods did not apply and we assigned them an AR value of 0.

B. Accuracy

There are five important questions that we want to answer when measuring the accuracy of our chunk-based approach on sequences with pseudoknot structures. First, we want to evaluate to what extent the chunk-based prediction approaches capture the secondary structures within the experimentally observed structures. Second, we want to identify whether predictions based on chunking produce more or less accurate results compared to the predictions using the whole sequence. Third, we want to understand whether the accuracy of chunk-based predictions correlates with the length of the whole sequence. Fourth, we want to understand whether the accuracy of chunk-based predictions correlates with the L, G parameter values. Fifth, we want to quantify the extent to which the inversion based chunking methods—centered (C) and optimized (O)—outperform the naïve chunking method—regular (R)—and which chunking strategy (C or O) is better.

To answer the first and second questions, in Table I, we present the maximum AC (MAC), maximum AW (MAW), and maximum AR (MAR) for each sequence, using pknotsRG as the prediction code, and using maximum chunk-length 60 (for method R, C, and O) with varying L, G values for methods C and O (both range from 3 to 8). A larger AC value means that the chunk-based predicted structure is more similar to the experimentally observed structure(e.g., an AC value 0.7 means that the predicted structure overlaps with the real structure in 70% of the nucleotide bases, hence it captures most of the stem-loop and pseudoknot structures); similarly, a larger AW value means that the predicted structure using the whole sequence is more similar to the experimentally observed one. An AR value larger than 1 means that the chunk-based predicted structure is more similar to the actual experimental structure than the secondary structure predicted by using the whole sequence. From Table I, we observe that in 9 out of 12 sequences (75%), at least one of the chunk-based predictions produces more or equally accurate results compared with the prediction using the whole sequence when using prediction code pknotsRG. Table II shows the same information but using HotKnots as the prediction code. In this case, the chunk-based predictions outperform the prediction using the whole sequences in 11 out of 12 sequences (92%). In Table III, we show the result for using PKNOTS as the prediction code. In all 6 sequences (100%), the chunk-based predictions produce more or equally accurate results comparing with the prediction using the whole sequence. And from Table IV we can see that the number is 6 out of 7 (86%) for prediction code NUPACK. From the

comprehensive results we obtain using different prediction codes, we observe that the chunk-based predictions tend to yield better accuracy than predictions obtained by using the whole sequence.

Table I
MAC/MAW/MAR FOR EACH SEQUENCE USING PKNOTSRG.

Seq.	Len.	MAC			MAW	MAR		
		R	C	O		R	C	O
RF00507_B	79	0.58	0.58	0.58	0.47	1.24	1.24	1.24
RF00233_B	84	0.68	0.68	0.7	0.68	1	1	1.04
RF00094_A	89	0.2	0.21	0.21	0.57	0.35	0.37	0.37
RF00499_A	103	0.7	0.7	0.7	0.7	1	1	1
RF00140_B	112	0.45	0.43	0.42	0.45	1	0.96	0.94
RF00259_A	169	0.33	0.24	0.39	0.34	0.97	0.71	1.14
RF00458_A	202	0.5	0.48	0.51	0.41	1.23	1.17	1.25
RF00261_B	221	0.19	0.46	0.49	0.2	0.93	2.27	2.4
RF00216_A	302	0.5	0.58	0.62	0.23	2.19	2.54	2.7
RF00010_A	312	0.59	0.56	0.6	0.63	0.94	0.89	0.96
RF00061_B	323	0.42	0.49	0.46	0.3	1.39	1.6	1.52
RF00024_A	451	0.3	0.45	0.56	0.6	0.51	0.75	0.94

Table II
MAC/MAW/MAR FOR EACH SEQUENCE USING HOTKNOTS.

Seq.	Len.	MAC			MAW	MAR		
		R	C	O		R	C	O
RF00507_B	79	0.63	0.63	0.63	0.47	1.35	1.35	1.35
RF00233_B	84	0.69	0.69	0.71	0.69	1	1	1.03
RF00094_A	89	0.2	0.44	0.69	0.37	0.55	1.18	1.85
RF00499_A	103	0.77	0.77	0.77	0.77	1	1	1
RF00140_B	112	0.46	0.45	0.47	0.46	1	0.96	1.02
RF00259_A	169	0.31	0.22	0.37	0.35	0.87	0.63	1.03
RF00458_A	202	0.45	0.54	0.46	0.47	0.95	1.16	0.97
RF00261_B	221	0.25	0.46	0.52	0.35	0.71	1.32	1.49
RF00216_A	302	0.47	0.52	0.63	0.41	1.14	1.26	1.51
RF00010_A	312	0.62	0.59	0.6	0.61	1.01	0.97	0.98
RF00061_B	323	0.37	0.48	0.48	0.49	0.76	0.99	0.99
RF00024_A	451	0.29	0.44	0.56	0.48	0.61	0.93	1.17

Table III
MAC/MAW/MAR FOR EACH SEQUENCE USING PKNOTS.

Seq.	Len.	MAC			MAW	MAR		
		R	C	O		R	C	O
RF00507_B	79	0.19	0.19	0.19	0.19	1.00	1.00	1.00
RF00233_B	84	0.37	0.37	0.38	0.37	1.00	1.00	1.03
RF00094_A	89	0.19	0.24	0.22	0.16	1.21	1.50	1.43
RF00499_A	103	0.26	0.26	0.26	0.22	1.23	1.23	1.23
RF00140_B	112	0.29	0.30	0.50	0.34	0.87	0.89	1.47
RF00259_A	169	0.22	0.27	0.28	0.20	1.12	1.39	1.42
RF00458_A	202	0.21	0.31	0.32	-	-	-	-
RF00261_B	221	0.20	0.25	0.24	-	-	-	-
RF00216_A	302	0.31	0.42	0.43	-	-	-	-
RF00010_A	312	0.22	0.32	0.33	-	-	-	-
RF00061_B	323	0.29	0.41	0.39	-	-	-	-
RF00024_A	451	0.26	0.45	0.49	-	-	-	-

To answer the third question, i.e., whether the accuracy of chunk-based predictions correlates with the length of the whole sequence, we draw a scatter plot of the MAC values versus sequence lengths using a different symbol for each prediction code (Figure 9). To quantify the correlation between MAC and sequence length, we computed the Pearson coefficient and the p-values for each prediction code for all the sequences, and the correlation is found not to be statistically significant: r^2 = 0.08009, p-value = 0.3488 for pknotsRG; r^2 = 0.11492, p-value = 0.2572 for HotKnots; r^2 = 0.005476, p-value = 0.8101 for PKNOTS; and r^2 = 0.05429, p-value = 0.4436 for NUPACK. The results are consistent across prediction codes. This indicates that there is no statistically significant dependence of the MAC values on sequence length. Thus, we do not expect decline in prediction accuracy when we increase the sequence length.

Table IV
MAC/MAW/MAR FOR EACH SEQUENCE USING NUPACK.

Seq.	Len.	MAC			MAW	MAR		
		R	C	O		R	C	O
RF00507_B	79	0.32	0.32	0.32	0.2	1.56	1.56	1.56
RF00233_B	84	0.69	0.69	0.71	0.69	1	1	1.03
RF00094_A	89	0.25	0.54	0.47	0.22	1.1	2.4	2.1
RF00499_A	103	0.74	0.74	0.74	0.74	1	1	1
RF00140_B	112	0.44	0.42	0.46	0.44	1	0.96	1.04
RF00259_A	169	0.27	0.24	0.23	0.3	0.9	0.82	0.78
RF00458_A	202	0.45	0.39	0.44	0.43	1.03	0.91	1.01
RF00261_B	221	0.14	0.35	0.34	-	-	-	-
RF00216_A	302	0.52	0.63	0.64	-	-	-	-
RF00010_A	312	0.54	0.53	0.6	-	-	-	-
RF00061_B	323	0.33	0.43	0.45	-	-	-	-
RF00024_A	451	0.28	0.44	0.48	-	-	-	-

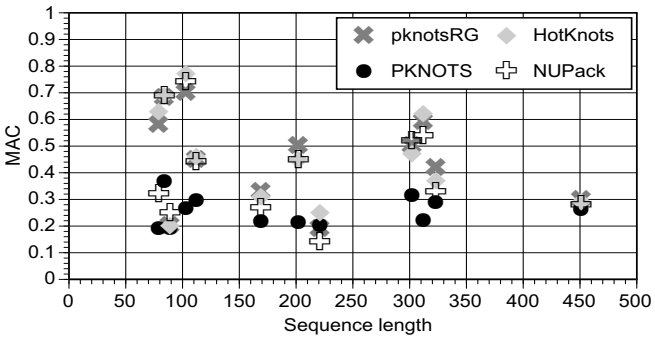


Figure 9. Scatter plot of MAC vs. sequence length for 4 prediction codes.

Unlike the (L, G) parameters which can be freely chosen by the user, the maximum chunk-length c in the cutting method is limited by the particular secondary structure prediction program. Just to explore the possible effect of c on MAR for the cutting method, we run our program with c ranging from 60 to 150 bases; by increasing c by 10 bases each time, we obtain a total of 540 combinations per sequence. Table V represents the count of sequences with highest MAR over all chunk-length combinations for each sequence. The table shows that the average MAR values stay approximately constant as c increases from 60-130 and start increasing as c increases from 130-150.

Table V
AVERAGE MAR FOR EACH MAX CHUNK-LENGTH.

	Average MAR		
Max chunk-length	R	C	O
60	0.84	1.06	1.09
70	0.89	1.07	1.07
80	0.84	1.05	1.05
90	0.79	1.05	1.03
100	0.93	1.45	1.42
110	0.91	1.05	1.06
120	0.93	1.06	1.08
130	0.83	1.07	1.05
140	0.91	1.13	1.12
150	0.97	1.18	1.17

To answer the fourth question, i.e., whether the accuracy of chunk-based predictions correlates with L and G, we present the number of sequences that give MAC with respect to the (L, G) value pairs. For the centered and optimized chunking method, there are 36 combinations of (L, G) value pairs (both range from 3 to 8). For each sequence, the MAC for each method is generated by one or more (L, G) value pairs. We present the number of MAC achieved by each (L, G) value pair in Figure 10. Some (L, G) combinations are missing from the figure since no sequence achieved MAC using any chunking method C or O by any prediction code. Our result shows that the value of the (L, G) parameter pair at which MAR is attained varies from sequence to sequence, with different segmentation methods and maximum chunk-length. In making predictions for sequences with unknown secondary structure, we have to come up with a particular appropriate (L, G) combination for that particular sequence, prediction algorithm, and max chunk-length. We expect that the length and composition of the sequence, as well as any knowledge of the biological characteristics of the RNA, will help us determine criteria by which suitable (L, G) parameters can be chosen. We observe that, in general, the L parameter values 3, 4, and 5 and G parameter values 3, 7, and 8 tend to give the best accuracy.

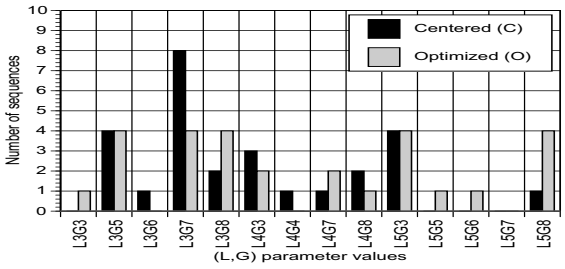
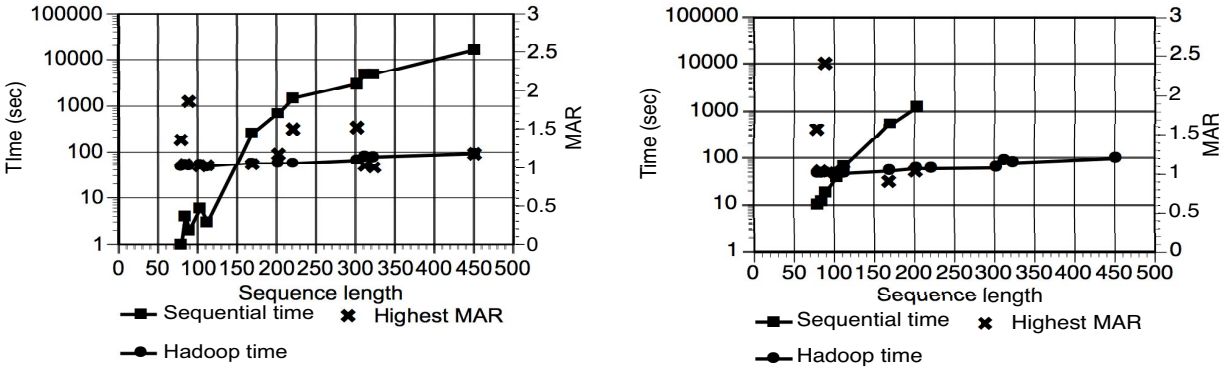


Figure 10. Number of sequences with MAC at (L,G) pairs.

To answer the fifth question, i.e, to quantify to what extent the inversion based chunking methods (C and O) outperform the naïve chunking method (R) and which chunking strategy (C or O) is better, we count the number of sequences where each chunking method gives the highest MAC. For those sequences in which multiple methods produce the highest MAC, the count is split equally among the methods. These counts are shown in Table VI, showing that the optimized method produces the highest MAC in 62% of our data sequences, followed by the centered (24%) and regular (14%) methods. These results suggest that the centered and optimized methods attain higher MAC than a naïve segmentation approach and that the optimized method outperforms the centered method.

To quantify whether there are significant differences among the cutting methods, we performed Friedman tests [22] on the chunking methods. The Friedman test is a non-parametric ANOVA test for repeated measures. This test requires ranking the MAC among the three methods for each data sequence and obtaining the rank sum for each method over the entire dataset. Methods sharing the same MAC are assigned an equal averaged rank. The rank sums for the four prediction codes are shown in Table VII. Table VIII shows the p-values obtained in the Friedman test for different chunking methods using each prediction code. The first column shows the p-values obtained in the tests that compare the R, C, and O methods; the second column for the C and R methods; the third column for the O and R methods; and the last column for the C and O methods. With significance level 0.05, we observe that for prediction codes pknotsRG, HotKnots, and PKNOTS, there are significant differences among the three methods (i.e., p-value$\leq$ 0.05); while for prediction code NUPACK, there are no significant differences among the three chunking methods (i.e., p-value $<$ 0.05). To further understand which two methods are significantly different, we report the p-values for the tests that compare C and R methods, O and R methods, and C and O methods in columns 3 - 5. As we can see, there is no significant difference between the centered and regular methods except when using the PKNOTS prediction code; there is a significant difference between the optimized and regular methods except when using the NUPACK prediction code; and there is no significant difference between the centered and optimized methods except when using the HotKnots prediction code. Even though the dataset size is modest due to the fact that there are not many RNA molecules including pseudoknots whose secondary structures we know experimentally, our tests still show the clear trend that the optimized chunking method performs significantly better than the regular method. It also shows that there is no significant difference between the centered and optimized methods for the majority of the prediction codes. To quantify which prediction program works better with our chunking approaches than others, we performed similar Friedman tests on the various prediction programs used. However, we observed no statistically significant difference among them, hence it is not reported here.



(a) HotKnots

(b) NUPACK

Figure 11. Profile of execution time for sequential, MR (MR) method and accuracy using highest MAR.

Table VI
COUNT OF SEQUENCES WITH HIGHEST MAC.

Regular	Centered	Optimized	Total
6.6	11.6	29.6	$12 \times 4 = 48$

Table VII
MAC RANK SUMS FOR EACH PREDICTION CODE.

Prediction code	Regular	Centered	Optimized
pknotsRG	19.5	22	30.5
HotKnots	18.5	22	31.5
PKNOTS	14.5	26.5	31
NUPACK	20.5	22.5	29

Table VIII
P-VALUES FOR EACH PREDICTION CODE.

Prediction code	C-O-R	C-R	O-R	C-O
pknotsRG	0.0302	0.7389	0.0114	0.0956
HotKnots	0.0085	0.3173	0.0114	0.0196
PKNOTS	0.0006	0.0027	0.0016	0.2059
NUPACK	0.1319	0.7389	0.0578	0.2059

C. Performance

When evaluating the performance of our chunk-based predictions, we want to answer two important questions. First, given a sequence of nucleotides, we want to understand whether the prediction of its secondary structure based on our segmentation approach (chunk-based) takes more or less time than the prediction of the whole sequence without segmentation when using the same prediction code. Second, we want to understand how the execution time of our segmentation-based predictions changes with the length of the sequences. When considering the chunk-based predictions, for each sequence and each prediction code, we run the prediction with Hadoop using the regular, centered, and optimized chunking methods with c equal to 60, L ranging from 3 to 8, and G from 3 to 8. When considering the prediction of each sequence as a whole (no segmentation), we use one of the compute nodes on our cluster. Because of space constraints, in Figure 11 we present the execution time for both the Hadoop chunk-based predictions and the sequential predictions using two of the four prediction codes: (a) HotKnots and (b) NUPACK. In each sub-figure, the x-axis is the 12 RFAM sequences sorted by length; the y-axis on the left is the execution time in seconds in logarithmic scale for both sequential and MR predictions; and the y-axis on the right is the highest MAR value for each sequence. Note that a MAR value ≥ 1 means that the predicted structure obtained by our chunk-based methods is more similar to the experimentally observed structure than the predicted structure obtained by using the whole sequence. Also note that in using the NUPACK program, some long sequences are missing because it cannot predict the whole sequence sequentially due to memory limitations. From Figure 11, we observe that the execution time for chunk-based predictions is larger than the sequential prediction when the sequence length is short (less than 150 bases). When the sequence length grows (> 150 bases), the chunk-based predictions run significantly faster than the sequential prediction. Overall we observed that as the length of the sequence grows, the execution time of the sequential prediction grows exponentially with the length of the sequences for all the four prediction codes. The time complexity of the PknotsRG algorithm is $O(n^4)$ where n is the number of nucleotides in the input sequence. However, in the prediction of RNA secondary structure using a segmentation method and using Hadoop implementation, we observed that the execution time for chunk-based predictions does not grow significantly with the sequence length. This is due to the fact that our chunking methods cut the whole RNA sequence into chunks no longer than 60 bases. In other words, as the length of the whole sequence grows, the chunks still within 60 bases. The predictions are performed in parallel across the Hadoop nodes and the distributed file system. As the sequence is cut into more chunks, the execution time for chunking (overhead) and prediction would increase.

Our measurements show how this overhead is not significant. In the case of very long RNA sequences (e.g., RNA viral genomes with thousands of bases), the chunk-based method is promising for two reasons: it allows us to predict secondary structures that we would not be able to predict otherwise, i.e., when considering the sequence as a whole; it would also allow us to keep under control the total execution time by controlling the maximum chunk length. Figure 11 also shows the highest MAR for each sequence (the higher the better). We can observe that, when the sequence length increases, the accuracy of our chunk-based methods does not decrease significantly compared with the prediction without segmentation - i.e., using the whole sequence.

V. CONCLUSION AND FUTURE WORK

In this paper we present two chunking methods, which are based on inversion excursions, for predicting RNA secondary structures including pseudoknots. We have assessed the accuracy of our methods with a dataset of RNA sequences with pseudoknots and four popular prediction codes (i.e., pknotsRG, HotKnots, PKNOTS, and NUPACK). Results show that our chunk-based methods outperform the prediction method using the whole sequence for a dataset of RNA sequences in 75% of the cases when using pknotsRG, 92% when using HotKnots, 100% when using PKNOTS, and 86% when using NUPACK. The somewhat counter-intuitive results in this paper suggest that local structures formed by pairings among nucleotides in close proximity, rather than the global minimal free energy, may correlate better with the real molecular structure of long RNA sequences. This hypothesis is being tested experimentally on genomic RNA sequences from Nodamura virus (NoV) in co-author Dr. Johnson's molecular virology lab.

ACKNOWLEDGMENTS

D.Y. and B.Z. equally contributed to this work. This work is supported in part by grants DMS 0800272/0800266 and EIA 0080940 from the NSF, RCMI 5G12RR008124-18 and NIMHD 8G12MD007592 from the NIH.

REFERENCES

[1] J. Ren, et al., "HotKnots: Heuristic prediction of RNA secondary structures including pseudoknots," *RNA*, 11:1494–1504, 2004.

[2] I. Brierley, et al., "Viral RNA pseudoknots: versatile motifs in gene expression and replication," *Nature Reviews Microbiology*, 5(8): 598–610, 2007.

[3] R. Nussinov and A. B. Jacobson, "Fast algorithm for predicting the secondary structure of single stranded RNA," *Proc. of the Nat. Acad. of Scien. of the U.S.A.*, 77(11):6309–6313, 1980.

[4] D. Sankoff, "Simultaneous solution of the RNA folding, alignment and protosequence problems," *SIAM J. on Applied Mathematics*, 45(5):810–825, 1985.

[5] M. Zuker, "Mfold web server for nucleic acid folding and hybridization prediction," *Nucleic Acids Research*, 31(13):3406–3415, 2003.

[6] E. Rivas and S. R. Eddy, "A dynamic programming algorithm for RNA structure prediction including pseudoknots," *J. Molecular Biology*, 285(5):2053–2568,1999.

[7] R. M. Dirks and N. A. Pierce, "An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots," *J. Comp. Chem.*, 25(10):1295–1304, 2004.

[8] M. Taufer, et al., "RNAVLab: A virtual laboratory for studying rna secondary structures based on grid computing technology," *J. Parallel Computing*, 34(11):661–680, 2008.

[9] D. Yehdego, et al., "Poster: Secondary structure predictions for long RNA sequences based on inversion excursions," in *Proc. of the ACM Conf. on Bioinformatics, Comp. Biology and Biomed. (BCB)*, 2012.

[10] J. Dean and S. Ghemawat, "MR: Simplified data processing on large clusters," in *Proc. of the 6th Symposium on Operating Systems Design and Implementation*, 2004.

[11] D. Hong, et al., "FX: An RNA-Seq analysis tool on the cloud," *Bioinformatics*, 28(5):721–723, 2012.

[12] B. Langmead, et al., "Cloud-scale RNA-sequencing differential expression analysis with Myrna," *Genome Biology*, 11:R83, 2011.

[13] B. Zhang, et al., "A modularized MR framework to support RNA secondary structure prediction and analysis workflows," in *Proc. of the 2012 Comp. Struct. Bioinfo. Workshop*, 2012.

[14] "Emboss-palindrome," http://emboss.bioinformatics.nl/cgi-bin/emboss/palindrome.

[15] S. Karlin, et al., "Statistical composition of high-scoring segments from molecular sequences," *Annals of Statistics*, 18(2):571–581, 1990.

[16] D. S. Chew, et al., "At excursion: a new approach to predict replication origins in viral genomes by locating at-rich regions," *BMC Bioinformatics*, 8:163, 2007.

[17] J. Reeder and R. Giegerich, "Design, implementation, and evaluation of a practical pseudoknot folding algorithm based on thermodynamics," *BMC Bioinform.*, 5:104–104, 2004.

[18] M. S. Andronescu, et al., "Improved free energy parameters for RNA pseudoknotted secondary structure prediction," *RNA*, 16(1):26–42, 2010.

[19] R. Dirks and N. Pierce, "A partition function algorithm for nucleic acid secondary structure including pseudoknots," *J. Comput. Chemistry*, 24(13):1664–1677, 2003.

[20] R. D. C. Team, *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008.

[21] S. Griffiths-Jones, et al., "RFAM: annotating non-coding rnas in complete genomes," *Nucleic Acids Research*, 33(suppl1):D121–D124, 2005.

[22] W. J. Conover, *Practical Nonparametric Statistics*, New York, NY: John Wiley & Sons, Inc., 1980.