Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Parallel Computing 34 (2008) 661–680

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

RNAVLab: A virtual laboratory for studying RNA secondary structures based on grid computing technology

Michela Taufer [a,*], Ming-Ying Leung [b,c,d], Thamar Solorio [e], Abel Licon [a], David Mireles [f], Roberto Araiza [f], Kyle L. Johnson [g,d]

[a] Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, United States
[b] Department of Mathematical Sciences, The University of Texas at El Paso, El Paso, TX 79968, United States
[c] Bioinformatics Program, The University of Texas at El Paso, El Paso, TX 79968, United States
[d] Border Biomedical Research Center, The University of Texas at El Paso, El Paso, TX 79968, United States
[e] Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, United States
[f] Department of Computer Science, The University of Texas at El Paso, El Paso, TX 79968, United States
[g] Department of Biological Sciences, The University of Texas at El Paso, El Paso, TX 79968, United States

ARTICLE INFO

Article history:
Received 5 June 2007
Received in revised form 6 June 2008
Accepted 21 August 2008
Available online 18 September 2008

Keywords:
Pseudoknots
Condor
Family *Nodaviridae*

ABSTRACT

As ribonucleic acid (RNA) molecules play important roles in many biological processes including gene expression and regulation, their secondary structures have been the focus of many recent studies. Despite the computing power of supercomputers, computationally predicting secondary structures with thermodynamic methods is still not feasible when the RNA molecules have long nucleotide sequences and include complex motifs such as pseudoknots. This paper presents RNAVLab (RNA Virtual Laboratory), a virtual laboratory for studying RNA secondary structures including pseudoknots that allows scientists to address this challenge. Two important case studies show the versatility and functionalities of RNAVLab. The first study quantifies its capability to rebuild longer secondary structures from motifs found in systematically sampled nucleotide segments. The extensive sampling and predictions are made feasible in a short turnaround time because of the grid technology used. The second study shows how RNAVLab allows scientists to study the viral RNA genome replication mechanisms used by members of the virus family *Nodaviridae*.

1. Introduction

Ribonucleic acid (RNA) is made up of four types of nucleotide bases: adenine (A), cytosine (C), guanine (G), and uracil (U). A sequence of these bases is strung together to form an RNA molecule that, unlike deoxyribonucleic acid (DNA), is often single-stranded. RNA molecules vary greatly in size, ranging from about 20 nucleotide bases in microRNAs to long polymers of over 30,000 bases comprising entire viral genomes [1]. Among the four nucleotide bases, C and G form a complementary base pair by hydrogen bonding, as do A and U and, less frequently, G and U. Although a single-stranded RNA molecule is a linear polymer, it tends to fold back on itself to form a three dimensional (3D) functional structure mostly by pairing between complementary bases. 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 time consuming and rather costly process. However, useful information about an RNA molecule can be gained from knowing its secondary structure, which refers to the collection of hydrogen bonded base pairs in the molecule. Essentially, all RNA secondary structures are made up of elements

* Corresponding author.
E-mail address: taufer@acm.org (M. Taufer).

0167-8191/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.parco.2008.08.002

that can be classified into two basic categories: stem-loops and pseudoknots (see Fig. 1). Both kinds of secondary structure elements have been implicated in important biological processes such as gene expression and regulation for stem-loops [2] and pseudoknots [3,4]. We also note that in both stem-loops and pseudoknots, it is necessary to have a stretch of nucleotide sequence (ACCGUC in Fig. 1a and b) followed by its inverted complementary sequence (GACGGU) downstream. For simplicity, we shall refer to these kinds of patterns as close inversions. The development of mathematical models and computational prediction algorithms for stem-loop structures based on thermodynamic models started in the 1980s [5,6]. Pseudoknots, because of the extra base pairings, must be represented by more complex models and data structures. Despite the computing power of supercomputers and emerging advanced technologies, e.g., multi-core architectures, the prediction of secondary structures of long RNA sequences (on the order of thousands of nucleotides) based on thermodynamic methods, e.g. [7], is still not feasible, especially if the structures include complex secondary structures like pseudoknots. The time and space required for accurate predictions of pseudoknots based on energy minimizations grow very rapidly with the sequence length. Fig. 2 shows the time and memory (in logarithmic scale) allocated for the prediction of RNA pseudoknots with various lengths using one of the most accurate prediction programs, Pknots-RE [8]. The algorithm underlying Pknots-RE has a runtime and memory demand in the order of n^6 and n^4, respectively, where n is the length of the input sequence [8]. The program conducts an exhaustive search for the optimal structure with the lowest free energy and has the capability to predict rather complex structures, even some non-planar structures, for short RNA segments up to 200 nucleotides in length. To overcome the tremendous demand on computing resources needed by pseudoknot prediction, alternative algorithms have been proposed (e.g., Pknots-RG [9], ILM [10], and HotKnots [11]) that tend either to restrict the types of pseudoknots or the length of the secondary structure to be predicted to keep runtime and memory size under control. For instance, the program Pknots-RG [9] limits the types of pseudoknots to simpler structures for longer segments, up to 800 nucleotides. However, a large variety of pseudoknots have been shown to occur in nature. Their omission from computational methods may significantly affect the prediction accuracy. Furthermore, even the simplified programs are not able to predict secondary structures on the order of thousands of nucleotides.

Our analysis of the length of pseudoknots in a widely used database known as PseudoBase [12], which collects 245 pseudoknots, shows that most pseudoknots known to date are formed by RNA segments whose lengths are less than 200 nucleotides, i.e., 95% of the segments in the database have lengths that range between 20 and 200 nucleotides. The range of lengths between 30 (lower quartile) and 67.5 (upper quartile) nucleotides covers 50% of all segments. This observation leads to the idea of overcoming the computing constraints presented above by developing a strategy for cutting long RNA sequences into segments not longer than 200 bases in length and distributing the task of structure prediction of each segment to be done simultaneously on different computers. Ideally, if two segments that are cut from the same RNA sequence overlap each other, then the predicted structures of their overlapping part should be consistent with one another. Such consistency is important for the final structure assembly. In preliminary work, we observed that arbitrarily cutting the RNA sequence into overlapping segments is not advantageous for consistency. It is well conceivable that when an arbitrary cut goes through the middle of a close inversion, the bases forming the pairings do not get into the same segment, resulting in the omission of the structure from that prediction. For instance, consider a 100 base segment of the Severe Acute Respiratory Syndrome (SARS) coronavirus genome, which is one of the coronavirus genomes analyzed by Chew et al. in [13], from position 25,884 to 25,983 and another segment from position 25,923 to 26,022. When the program Pknots-RE is applied to these segments, two predictions are produced which are shown in Fig. 3. Note that, over the stretch of the 62 bases that the two segments overlap one another, the two predictions are different. This kind of inconsistency poses a serious problem when the predicted structures of the segments need to be assembled.

If the prediction of secondary structures for long RNA sequences is not feasible with thermodynamic methods even with powerful supercomputers, arbitrarily cutting an RNA sequence into shorter overlapping segments makes the single segment predictions feasible but not advantageous for consistency, unless the cutting algorithm uses the locations of high concentration of close inversions. In addition, once motifs are predicted from sampled segments, they have still to be assembled

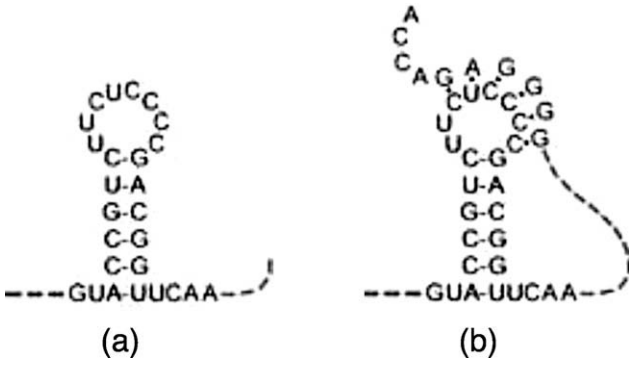


Fig. 1. Examples of stem-loop (a) and a pseudoknot (b).

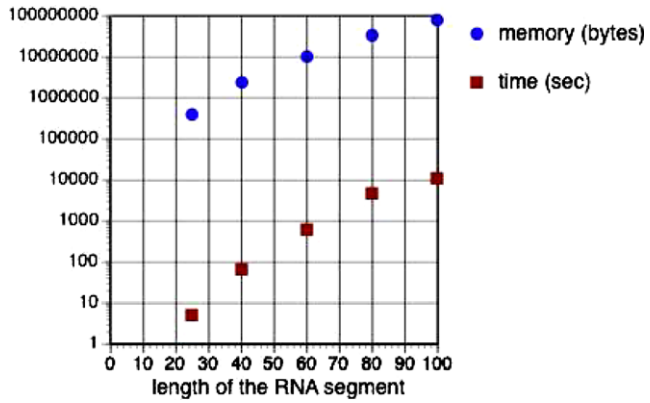


Fig. 2. Time (sec) and memory (bytes) used by Pknots-RE for predicting PseudoBase sequences with different lengths.

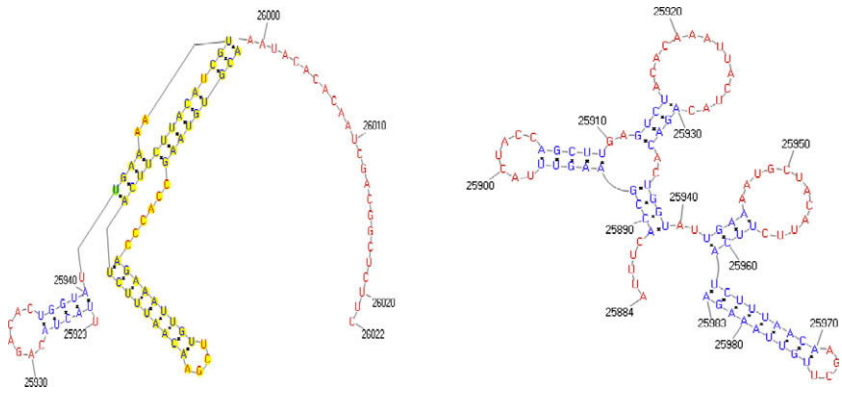


Fig. 3. Pknots-RE predictions of two overlapping SARS coronavirus segments, including bases 25,884–25,983 (left) and 25,923–26,022 (right).

together into complete secondary structures. Both predictions and rebuilding can significantly benefit from the combined prediction capability of different codes, as opposed to using the single codes separately [14]. Prediction of large numbers of short, overlapping segments is still computationally demanding but it also allows massive task parallelism that can be addressed with grid computing resources. Of course, the scientist who uses such an approach of sampling and rebuilding from segments to predict longer secondary structures has to benefit from the computing capabilities of such a framework without being required to cut and paste results from one code output to another, redirecting or reformatting output files (e.g., from FASTA to EMBL format) before forwarding them to the next step in the analysis, or dealing with distributed computer systems. Therefore, the ideal computing environment for the scientist should integrate key services and functionalities by combining different mechanisms and programs in an automated, computationally powerful, and reliable cyberinfrastructure accessible through an easy-to-use, familiar interface.

These overall needs and key services are integrated in RNAVLab (RNA Virtual Laboratory), a virtual laboratory for the study of RNA secondary structures. RNAVLab addresses the challenges above by combining sampling of nucleotide sequences, predictions based on different codes and supported by grid computing technology, and analysis of large sets of secondary structures with different scientific scopes. Scientific scopes include associating stem-loop and pseudoknot types with functions in genomes. In this paper two important case studies using RNAVLab are presented:

- To rebuild secondary structures in longer sequences by systematically sampling nucleotide segments from an RNA molecule and assembling the significant motifs found in the secondary structures of the segments (i.e., stem-loops and pseudoknots). The extensive and systematic sampling of nucleotide segments is vital for overcoming the inconsistency outlined above; the computing power needed for the prediction of the numerous segments is provided by grid technology. Motifs are identified in the secondary structures of each segment and assembled in a single structure based on their recurrences and statistical insights.
- To study the association of stem-loop and pseudoknot structures with the viral RNA replication mechanisms for the genomes of members of the virus family *Nodaviridae*. In other words, RNAVLab helps scientists to address questions such as ‘‘what are the mechanisms of nodavirus RNA replication?’’. The association of predicted secondary structure near the 3′

terminus of the RNA2 genome segments from seven members of the nodavirus family with their RNA replication mechanisms is targeted. The nodavirus genomes provide an excellent model system for the study of RNA replication due to their genetic simplicity, the robust yield of RNA replication products, and the ability of the RNAs to replicate in cells from a wide variety of organisms.

The rest of this paper is organized as follows: Section 2 discusses significant related work in the field. Section 3 presents the RNAVLab virtual laboratory and its components. Section 4 shows how to use RNAVLab for rebuilding longer RNA secondary structures from RNA segments and to assist in the study of nodavirus RNA replication. Section 5 concludes and briefly presents future work.

2. Related work

When dealing with RNA secondary structures, scientists have several sources of data and tools available. However, to retrieve pieces of information from these sources as well as to sort and elaborate the data with these tools, scientists have to do significant handwork by sorting, computing, merging, and comparing results as well as extrapolating conclusions. For instance, when dealing with pseudoknots, scientists need to access databases such as PseudoBase [12], which are not always provided with advanced search engines. The data from the database has to be copied into files of different format (e.g., FASTA and EMBL). Ultimately the scientists have to download and install codes on platforms that do not always fully support the code execution. Even if portable, some of these codes require significant amount of computing power that is not always available on the scientist's PC. An alternative is to submit the retrieved data to portals that provide prediction and visualization functions. However, the portals provide single prediction approaches that force the scientists to find multiple portals for their prediction. Ideally scientists should be led through the different steps by a unified, easy-to-use environment that screens them from database issues providing them with powerful tools for search, prediction, analysis, and visualization.

While past efforts have been focused on increasing prediction accuracy of sequential RNA prediction programs [8,9] and prediction efficiency has been improved by using massively parallel high performance machines [15] or local clusters [16], not much is known about RNA prediction systems based on grid computing technology (i.e., heterogeneous, volatile computers, ranging from supercomputers to clusters and PCs connected to the Internet, spread out in different locations). Previously, grid technology was applied successfully to protein structure prediction [17,18] and similar achievements are expected for RNA secondary structure prediction. With their significant computing power, these computing systems allow the scientists to explore larger spaces of RNA secondary structures.

Among the several tools based on thermodynamics that are available for RNA secondary structure predictions and analysis, the Vienna Package [19] is one of the most well-known packages. It consists of a C code library and several stand-alone programs for the prediction and comparison of RNA secondary structures. The stand-alone programs are not integrated in a unified environment and do not address multiple prediction approaches but instead deploy the Zuker algorithm [6]. Therefore, the Vienna Package does not include the prediction and analysis of pseudoknots. Last but not least, the package does not integrate grid technology.

As an alternative to thermodynamic methods for RNA secondary structure prediction, Stochastic Context Free Grammars (SCFG) have been proposed for secondary structure prediction [20]. These approaches rely on estimating probability distributions over a set of transformation rules that define how the fold is formed. SCFGs have the ability to learn the parameters of a generative model by observing a set of sequences with their corresponding secondary structures. In general, SCFGs are outperformed by physics based approaches, although recently Do et al. proposed a generalization of SCFGs where a flexible and richer feature set allows to include free energy parameters more akin to thermodynamic models [21]. However, the complexity of the RNA secondary structures predicted by these methods is restricted by the expressibility of their grammars, thus highly complex structures, such as pseudoknots, cannot be predicted by SCFGs. In addition, the above-mentioned tools are single purpose; they can only be used for secondary structure prediction.

For finding consensus motifs that can be associated with RNA functionalities, most of the previous work takes as input a set of primary sequences and generates as output the set of structural motifs identified, and the differences lie in the search strategy for identifying common motifs. For instance, work presented in [22,23] uses suffix arrays for efficiently exploring the space of valid secondary structures in their Seed method. In Seed, the search space is constrained by the seed sequence, which is just one of the sequences in the set used to instantiate valid motifs. Seed ranks motifs using a metric that combines the entropy of the segment with the free energy of the secondary structure, as computed by MFOLD [7]. This ranking function reached good results and the top motifs had also the highest Matthews Correlation Coefficient [24]. A drawback of Seed is the fact that it is limited to finding patterns in stem regions only, i.e., no loops or pseudoknots can be identified by the Seed method.

Ashlock and Schonfeld propose a depth annotation scheme to identify common motifs that uses an evolutionary algorithm to cluster folds by projecting them in a two dimensional Euclidean space [25]. This method can identify pseudoknots by assigning a unique identifier to stems. The intuition behind this approach is that similar folds will be placed closer by the projection algorithm. To identify motifs, we need to analyze the output of the projection. Since the method provides a visual representation of the similarity between bricks, it is simple to identify motifs by just looking for clusters. However, as the number of bricks increases, spotting the clusters become less straightforward and we need the help of a clustering algorithm.

Another shortcoming of this method is that the distances between the pairs of depth annotations depend on a specific size of segment. Thus prior knowledge of the sequences is needed in order to define an appropriate window size.

There are other approaches to motif finding, see for example [26], but most of them give the desired results provided the secondary structure is not complex (no loops or pseudoknots are included), or provided that we have enough prior knowledge regarding the identity of the motifs. On the contrary, our automated method targets motifs that are as general as possible and exhaustively explores the search space of all the sequences of nucleotides. It is a strictly structural method in the sense that, for the experiments presented here, we only looked at the secondary structure predicted by Pknots-RG. Our preliminary results show that our method can find motifs as simple as small stems and as complex as pseudoknots and loops.

With reference to the deployment of RNAVLab proposed in this paper, to our knowledge, these existing tools have not been used for such an exhaustive survey of the link between RNA secondary structure prediction and nodavirus genome replication. Lastly the highly modular design of RNAVLab makes it easy to add new tools. The tools described above can then be integrated into the array of tools that already exist in RNAVLab.

3. RNAVLab overview

RNAVLab has a modular structure that makes it easy to integrate new features. As shown in Fig. 4, RNAVLab has three major components: (1) a segment sampler component (*Sampling*) to sample RNA sequence segments guided by simple heuristics and more sophisticated mathematical methods capable of identifying palindrome distributions; (2) a structure prediction component (*Prediction*) to predict the structures of the sampled segments using different programs on heterogeneous computers; and (3) a structure analysis component (*Analysis*) to compare and contrast predictions with observed structures as well as identify similarities and other characteristics across predictions. Each component, described in more detail in this section, is shown in Fig. 5. RNAVLab also includes a database of pseudoknot structures, PseudoBase++ and an easy-to-use interface; both the database and the interface are also described in this section.

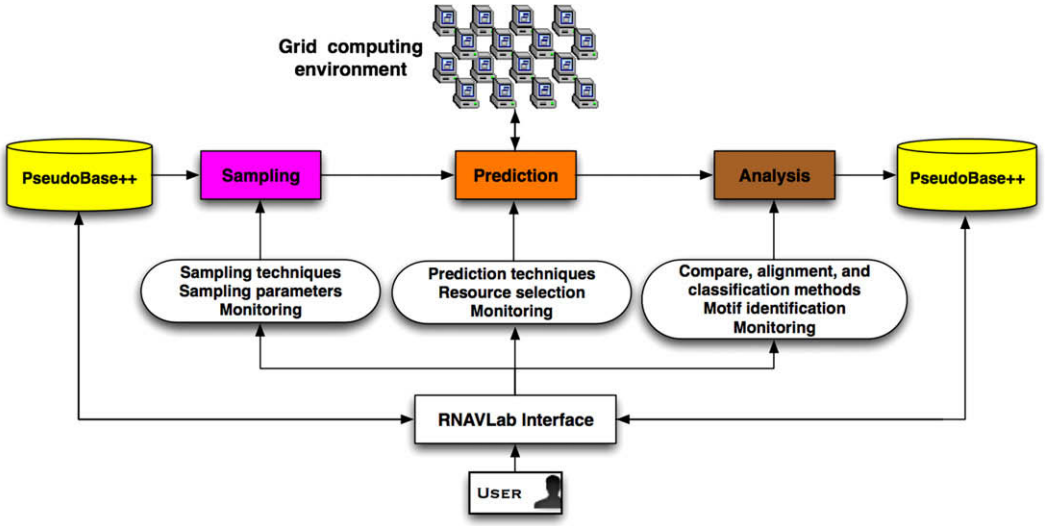


Fig. 4. Overview of RNAVLab functionalities, software components, and user interface.

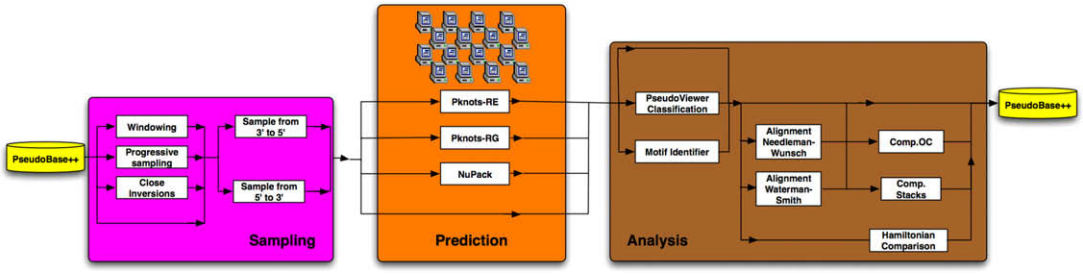


Fig. 5. Detailed overview of the tools in the three major components of RNAVLab.

3.1. Sampling component

The segment sampler component (*Sampling*) samples overlapping segments in RNA sequences and passes them to the structure prediction component for the search of significant motifs. Generally, RNA segments containing a high concentration of close inversions have greater tendency to form local secondary structures because the symmetry facilitates base pairing required in the formation of stem-loops and pseudoknots [27,13]. Currently RNAVLab includes two sampling strategies: a windowing strategy and a progressive segmentation strategy.

In the windowing sampling approach, each set of segments has a fixed size (*window size*) and a fixed sliding step (*window step*). The segments in a set are generated by progressively sliding the fixed-size window forward for a fixed number of steps. At each step, the nucleotides within the window form a segment. For an extensive sampling, this procedure is repeated to generate several sets by increasing the window sizes and/or the window steps, each time generating a new set of segments. In our experiments the window sizes are increased by five bases. The maximum length of a window is $n/2$, where n is the length of the RNA sequence we want to rebuild the secondary structure for. Window steps range from 1 base to $w - 1$ bases, where w is the window size.

In the progressive sampling approach the user defines a starting point, ending point, and a "step size"; the sampler generates a series of segments by progressively removing "step size" bases from the original segment, whose length is defined by the beginning and ending points (5′ and 3′ termini, respectively) given by the user, starting from the beginning point and progressing in a 5′-to-3′ direction. The series of segments with progressively decreasing lengths are forwarded to the prediction component for prediction. Segments can be inverted before being forwarded. The extension of this component to employ more sophisticated statistics-based sampling methods using the distributional properties of close inversion on random RNA sequences is work in progress.

3.2. Structure prediction component

The structure prediction component (*Prediction*) harnesses heterogeneous computing resources across the University of Texas at El Paso (UTEP) campus to rebuild RNA secondary structures from the segments generated by sampling, using a variety of prediction programs. Currently RNAVLab supports the following prediction programs: Pknots-RE [8], Pknots-RG [9], and NuPack [28]. The structure prediction component links the segment sampler component to the structure analysis component. Its main functionalities are: (1) to dispatch jobs provided by the segment sampler to available computing resources across the UTEP campus and (2) to identify completed jobs so that results can be forwarded to the structure analysis component.

To setup a grid computing environment, RNAVLab relies on the Condor framework [29], for a variety of reasons. Condor provides all of the functionalities needed to implement a seamless grid layer that allows for faster prediction of RNA secondary structures by harnessing the idle cycles of computers, i.e., workstations and clusters, across the campus. The pool of machines on which RNAVLab is currently relying consists of 23 single-, double-, and quad-processor 32- and 64-bit machines, and while all these machines run Linux (i.e., Fedora, Kubuntu, SuSE), Condor can also be installed on Unix, Windows (e.g., 2000, XP), and Macintosh (OS X) machines, among others. Condor handles all the details of sending executable and data files to computing resources and retrieving the computation results. Furthermore, Condor provides other useful features, such as checkpointing and job migration that only require re-linking the prediction programs used by RNAVLab with Condor libraries. These features can be very helpful, especially with predictions that take a long time: if the application is interrupted, checkpointing saves the computation's state so it can be resumed later (instead of starting from scratch), and job migration allows a saved state to resume execution in a different machine. RNAVLab successfully re-links Pknots-RE and NuPack, but not Pknots-RG, due to its use of *pthreads*. Therefore, Pknots-RG cannot use checkpointing and job migration, but it can still be dispatched to the computing resources. Pknots-RG is usually the fastest of all three programs and therefore checkpointing may not be necessary or even helpful.

To interface RNAVLab with Condor and dispatch jobs, an XML file describing the submitted jobs is generated when the user invokes the use of global resources through the RNAVLab interface. Each job consists of a unique identifier, the name of the prediction program to be used (e.g., Pknots-RE, Pknots-RG, and NuPack), and the command-line parameters (i.e., the input files with the RNA sequences). The XML format is converted into a Condor submit file format, and Condor is used to submit the jobs to the pool. Condor also provides the functionality needed to check whether a submitted job has completed execution: this functionality is blocked while the job is running and only returns when the job is completed. Once all the jobs are submitted through Condor, RNAVLab sequentially checks for the completion of jobs. If RNAVLab stalls because a job is not finished, it does not stall the other jobs, since they are already on the queue. Once jobs are completed, their results are forwarded to the structure analysis component and visualized, if required by the user, on the interface. Although the current pool of machines used by RNAVLab consists only of Linux machines, future work includes its extension to clusters available across the campus as well as the integration and support of BOINC (Berkeley Open Infrastructure for Network Computing) [30] to allow researchers to deploy desktop and laptop PCs owned by students or administration personnel outside the campus when these computers are idle. Work in [17] shows that adding idle cycles of PCs significantly increases available computing power.

3.3. Structure analysis component

The structure analysis component (*Analysis*) evaluates the consistency of the various predictions collected by the structure prediction component. Currently a set of tools allows the end-user to perform secondary structure classifications, comparisons, alignments, and motif identification. In general, a motif is a repeated pattern in a set of sequences of nucleotides (primary structure), or in a set of secondary structures. An innovative aspect of this component is that RNAVLab deals with secondary structure rather than nucleotide sequences for classification, comparison, alignment, and identification. Secondary structures are considered in terms of strings of brackets, i.e., "(" and ")", "[" and "]", "{" and "}", dots ".", and colons ":". Two paired nucleotides are represented with a pair of brackets collocated in the string at the same positions as the related nucleotides in the input segment.

With reference to the classification of secondary structures and more in particular of pseudoknots, the *PseudoViewer Classification* tool deploys the classification of pseudoknots in [31] that clusters them into six different simple types, i.e., H-, HH-, HHH-, HL_out-, HL_in-, and LL_in-type. Note that "H" means hairpin loop, "L" means bulge loop, "in" means internal loop or multiple internal loops, and "out" means external loop or multiple external loops. The tool for classification works on the string of brackets to extract the proper type. Fig. 6 shows the six pseudoknot types.

Comparisons of observed and predicted structures, or across predicted structures using different programs are performed on aligned or non-aligned strings of brackets. Three different algorithms can be used for comparisons:

- A variant of the Hamilton algorithm (*Hamiltonian Comparison*) – the algorithm assigns each kind of nucleotide bond a numerical tag. Bonds GC, CG, UA, and AU are assigned tags from 1 to 4, respectively. The closures of the bonds are all assigned 0. The resulting numerical strings are compared and when two non-zero numbers and their respective closing 0 match, it is counted as a true pair. This approach is useful when the types of nucleotide bonds are important. Fig. 7 shows

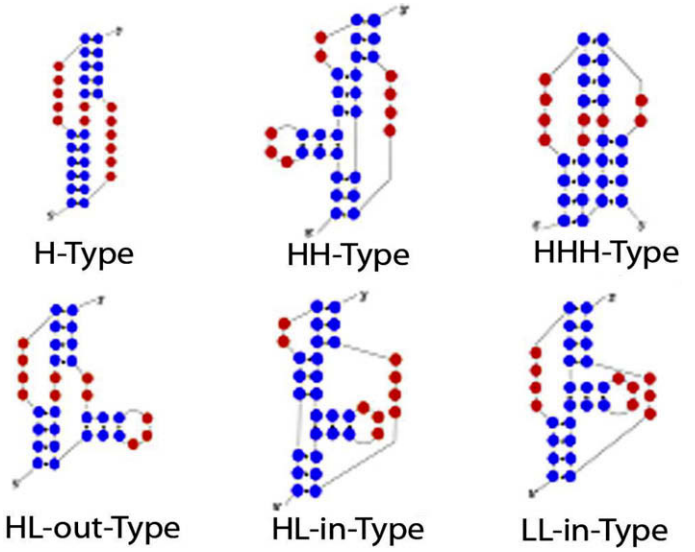


Fig. 6. The six pseudoknots types.

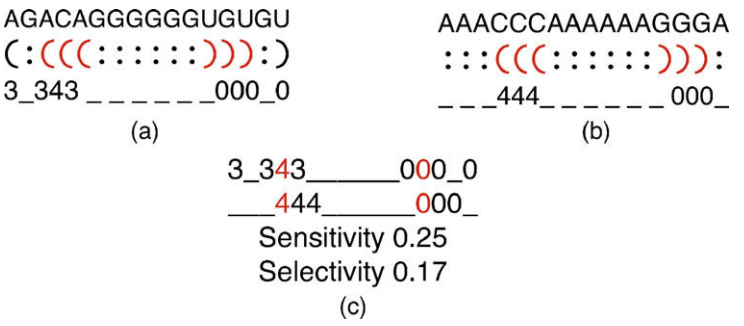


Fig. 7. Example of comparison with a variant of the Hamilton algorithm.

an example of comparison of a predicted string (Fig. 7a) with an observed string (Fig. 7b) by using the variant of the Hamilton algorithm. The two strings of brackets associated with the prediction and observed secondary structures are converted to two strings of numbers in Fig. 7c. The strings of numbers are compared and the related sensitivity and selectivity are computed. Here, high sensitivity expresses the ability to predict all observed pairs and high selectivity expresses the ability to only predict observed pairs.

- A stack based algorithm (*Comp. Stack*) – the algorithm uses stacks to keep track of the positions of open parenthesis and brackets in secondary structures with and without pseudoknots. When an open bracket or parenthesis is encountered, its position is pushed into a stack associated to a stem-loop. Pseudoknots are considered to be a combination of two stem-loops and therefore use two stacks. When a closed bracket or parenthesis is encountered, the position is popped from the associated stack. If a bracket or parenthesis is encountered at the same time in both structures and the position popped from both stacks is the same, this is counted as a true pair. This approach is useful when the identification of exactly alike structures is important. Fig. 8 shows two examples of comparison with this approach: in Fig. 8a the secondary structures are very similar but the bonding nucleotides are shifted and therefore, the comparison has sensitivity and selectivity equal to zero. In Fig. 8b, three bonds in the first string have the same position as three bonds in the second string and this matching results in a higher score for this example in terms of sensitivity and selectivity.
- A lenient algorithm (*Comp. OC – Open–Close parenthesis*) – the algorithm uses simple string comparisons that allows for similar yet shifted structures to receive high comparison scores. The algorithm traverses two bracket strings and counts how many times an open bracket or parenthesis is in the same position and how many times a closed bracket or parenthesis is in the same position for the strings, without considering the type of nucleotides involved. The smaller of these two values is the amount of true pairs. Fig. 9 compares the two strings already compared in Fig. 8a with the stack based algorithm. By using the lenient algorithm, higher sensitivity and selectivity are achieved.

The alignment of two or more secondary structures is performed by aligning their bracket strings using variants of well-known alignment algorithms such as the Smith–Waterman [32] and Needleman–Wunsch [33] algorithms. Unlike the aforementioned algorithms that align string of nucleotides, i.e., "A", "U", "C", and "G", the variant algorithms align strings of brackets, i.e., ":", "(", and "[". Shifts are indicated with an underscore, i.e., "_". The alignment of secondary structure is very important to identify secondary structures that are similar in their shape but are shifted: this can happen when e.g., a predicted structure is compared with an experimentally observed structure or when the genome structures of two viruses belonging to the same family are compared. Fig. 10 shows an example of how the alignment with the variant of the Smith–Waterman algorithm can improve comparison sensitivity and selectivity. Fig. 10a shows the comparison of two strings that have not been a priori aligned; Fig. 10b shows the comparison of the same strings once they have been aligned. The alignment allows for the identification of the two similar structures and affects the sensitivity and selectivity by increasing their final values.

The *Motif Identifier* tool performs the identification of common motifs that can be ultimately used for: (1) rebuilding large secondary structures from smaller ones belonging to overlapping RNA segments of the same virus; (2) quantifying the capability of the several prediction programs to capture secondary structures that have been experimentally observed; and (3) classifying unknown viruses by matching common motifs present in their RNA with similar motifs in viruses belonging to a known family. The tool proceeds as follows: first it identifies all the valid secondary structures, from the most simple (e.g., a hairpin comprising a few base pairs) to the most complex (e.g., pseudoknots), that can be generated from the input of secondary structures. Then by using an associative array of linked lists, the tool finds and stores the locations of each substructure generated in the previous step. To narrow down the number of motifs and identify the most relevant ones, ranking

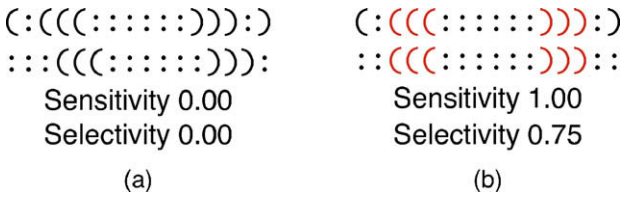


Fig. 8. Example of comparison with a stack based algorithm.

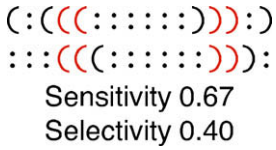


Fig. 9. Example of comparison with the lenient algorithm.

Fig. 10. Example of secondary structure alignment.

Let S be a set of *n* secondary structure sequences
Let M be the set of all valid structural motifs
1. M = {}
2. For every s_i in S
   2.1. Generate a new set VS_i with all the valid motifs in s_i
   2.2 M = M U VS_i
3. For every motif m_i in M
   3.1 Search and store the location of all occurrences of m_i in S
4. For every motif m_i in M
   4.1 Rank m_i according to the set of criteria C
   4. 2 Remove m_i if the scoring is below a given threshold

Fig. 11. The Motif Identifier algorithm.

techniques are applied. Ranking criteria include: the frequency of the motif over the maximal number of possible occurrences, the number of bonding nucleotides, the length of the secondary structure, and the motif location in the RNA segment. Other possible ranking criteria can include information of the primary structure such as the percentage of bases correctly matched, and/or free energy of the structure as in [34]. In RNAVLab, we score motifs based on their frequency (f), number of base pairs (s), and the length of the overlapping region (o):

$$\text{score} = \frac{f * s}{o}. \tag{1}$$

The simple intuitive motivation behind this scoring function is that more accurate secondary structures are more likely to be predicted in overlapping segments with higher frequency.

To assemble the significant motifs and rebuild the final secondary structure out of the segment motifs, we use the scoring function above. We project motifs, in descending order according to their score, into a final structure until there are no more mutually exclusive motifs in the set. In other words, we only project different motifs found in segments when they do not overlap with each other. As part of the rebuilding algorithm, we also define the minimum frequency that a motif present in overlapping segments has to meet in order to be projected in the resulting rebuilt sequence (*threshold*). Threshold values can range from 1 to 9. Finally, we compute the energy of the rebuilt structures as a whole by using the same energy algorithm used in Pknots-RG and NuPack. Fig. 11 shows the pseudocode of the tool.

In contrast to other approaches for finding common motifs described in Section 2, the Motif Identifier tool finds strictly structural motifs, but it only looks for motifs composed of the folds defined in the initial set: the search space explored is defined by all the valid secondary structures in the set and the tool does not use any information from the primary structures. The algorithm in Fig. 11 was inspired by previous work such as [22,23,34]. A key difference is the lack of an ad hoc design: no prior knowledge is assumed about existing motifs and the goal is to discover the motifs that are more likely to be part of the native structure. The algorithm of the Motif Identifier is simple but particularly powerful because it allows identifying complex structures such as pseudoknots. Lastly, since the design is modular new features such as different rank-

ing functions can be easily added. The different tools in the Analysis component can be combined to perform more complex operations on the secondary structures. For instance, motifs that have been identified by the Motif Identifier can be aligned and compared by any of the alignment and comparison tools. Information on the consistency of the prediction results collected can be fed back to the database and made available to the segment sampler component to adjust the sampling strategy and adaptively identify new RNA segments for predictions.

3.4. The database source PseudoBase++

PseudoBase++ (http://pseudobaseplusplus.utep.edu) is part of RNAVLab and includes a database of pseudoknots, a search engine to access the data, and a user interface to select, visualize and insert new data through any web-browser. The database is an extension of PseudoBase [12]: it contains the data related to pseudoknots that is already provided in PseudoBase and other additional data that enriches the information associated to each pseudoknot entry. PseudoBase++ is currently focusing on pseudoknot structures. The primary source of data in PseudoBase++ is PseudoBase: 257 structures are from this database.

3.5. The user interface

The RNAVLab user interface is designed and implemented around the RNAVLab computational environment with a strong focus on compatibility. The implementation idea is also derived from the key concept of encompassing high functionality within a simple but comprehensive interface. By maintaining these two design concepts throughout the process of implementation, the resulting application successfully supports this easy-to-use interface with the rich functionality provided by RNAVLab. The interface is developed in JAVA, thus in nature, preventing any operating system dependencies.

The RNAVLab interface is designed to accommodate the visual structure of a basic media player. This is to create an immediate familiarity for the users as well as maintain a simple comprehensive infrastructure. The interface includes four tabs: database, tools, full results, and previous results. The *database tab* is a representation of RNA secondary structures in the database. The *tools tab* is divided into two main sections. One of the sections maintains total functionality with visual representations for the database of sequences, results pertaining to the user's requests, and previously obtained results. This section is conveniently organized with a tabled panel allowing quick access to any sub-section (Fig. 12, section on the left). The other main section of the interface provides a constant representation of all concurrent processes being performed by the user as well as a list of each of these processes once they are completed (Fig. 12, section on the right). By selecting any of these processes, the results pertaining to that particular process are displayed conveniently on an information panel located just below in the section on the right in the information window. Though the results are saved and displayed in the

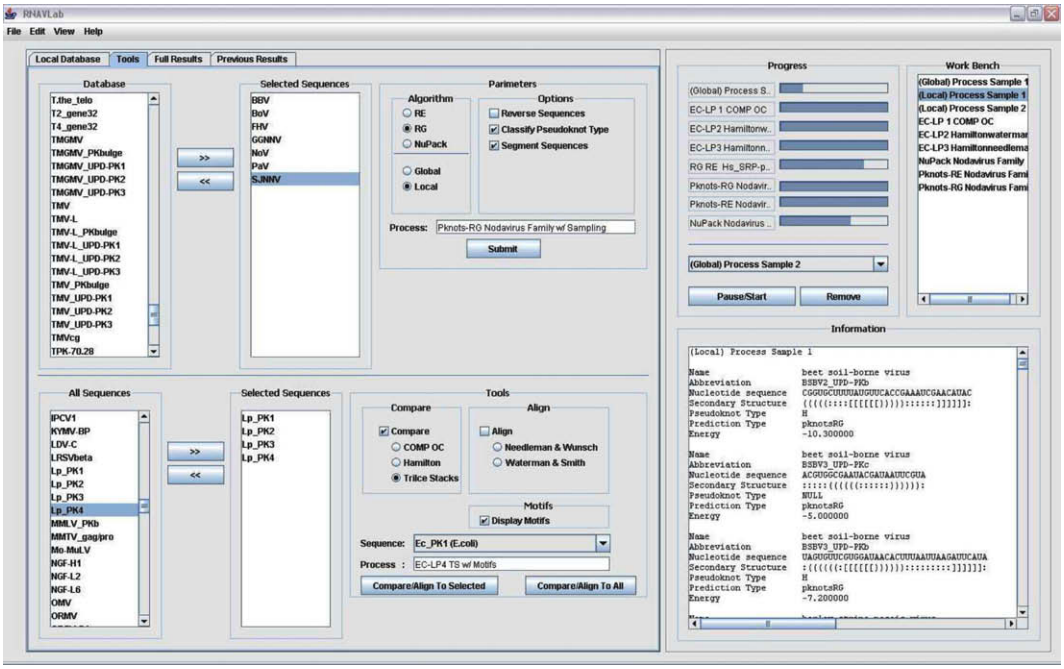


Fig. 12. Snapshot of the user interface.

full result tab and *previous results tab*, the visualization in the information window is implemented to allow direct comparisons to several outcomes simultaneously. The interface is capable of supporting two different types of processes through the tools tab, one of sampling and prediction and the other of analysis. Each process has its own convenient selection section with easy checkbox and radio-button options. Each section maintains a copy of the database of sequences with independent sub-lists designated to separate desired sequences or groups of sequences. The first section on the top of the tools tab allows the sampling and prediction with Pknots-RE, Pknots-RG, and NuPack either locally (on the local machine) or globally (across the UTEP campus machines) in accordance to the particular selected sequences or groups of sequences. Reversing and sampling the resulting sequences and classifying the resulting pseudoknot types are also sub-options included in this section. The second section on the bottom of the tools tab allows the user to analyze selected secondary sequences using the tools described in Section 3.3. The tools can be used simultaneously in different combinations.

4. Using RNAVLab: two significant case studies

We present two case studies that address the potentials of RNAVLab from two different points of view. In Section 4.1, we present a study in which we address two important analysis components of RNAVLab. First, we quantify its capability to capture the secondary structure observed experimentally. We compare performance and accuracy (in terms of sensitivity and selectivity) of the RNAVLab rebuilding algorithm based on nucleotide segments against a traditional algorithm using the same prediction code and the entire sequence. Second, we statistically quantify the effectiveness of the RNAVLab naive approach for sampling nucleotide segments and we measure whether the extensive sampling and predictions can compensate for the fact that no attention is paid to the type of nucleotides in the segments, i.e., whether or not there are palindromic sequences.

In Section 4.2, we present a second case study in which we show how RNAVLab can be used for studying the correlation between viral RNA replication mechanisms used by members of the nodavirus family and the secondary structures adopted by the 3′ ends of their RNA2 segments, which are hypothesized to play a role in the initiation of complementary strand synthesis during RNA replication.

4.1. Case study I: rebuilding longer RNA secondary structures from motifs in RNA segments

Secondary structures for long RNA sequences i.e., on the order of thousands of nucleotides, that have been experimentally validated are rare. When available, our method can deal with the prediction of these sequences but other methods that predict secondary structures using the entire sequence as a whole cannot, making a comparison between the two approaches infeasible. Therefore, for our analysis in this paper we used the 39 longest nucleotide sequences from Group A in [21] that have lengths ranging from 100 to 482 bases and are still predictable as a whole by the Pknots-RG code. Note that since we are not considering the exact same set as in [21], we cannot perform a direct comparison against those results.

The sampling, motif identification, and rebuilding were executed on the RNAVLab server. Window sizes, window steps, and thresholds used in the experiments are defined in the previous section. The predictions were performed on a 64-node cluster (each node consists of 2 AMD Opteron processors running at 2 GHz with 4 Gbyte of RAM and a local 120 Gbyte hard disk) that is part of the on-campus grid resources of RNAVLab. The accuracy of predictions is measured in terms of sensitivity (i.e., ability to predict all true pairs) and selectivity (i.e., ability to only predict true pairs). Predictions are compared with the experimental secondary structures provided in [21].

Fig. 13 presents the 39 sequences (*Sequence*), their length in bases (*Length*), their number of rebuilt structures including those that, when compared with the experimental secondary structures, have sensitivity and selectivity equal to zero (*Predictions Attempted*), the number of rebuilt structures that have a positive sensitivity and selectivity (*Predictions Used*), the total time in seconds needed for all the segment predictions on the cluster (*Rebuilt Time*), and the time in seconds used for the prediction of sequences as a whole when using Pknots-RG (*Pred. Time*). The figure underlines the high cost in terms of computation needed for our approach. RNAVLab makes our approach feasible by allowing us to perform the computation on idle resources across the campus.

4.1.1. Accuracy of rebuilt structures

In Fig. 14, we present a summary of the accuracies: the oracle or upper bound on sensitivity and selectivity for our method (*Rebuilt Sen.* and *Rebuilt Sel.*) is compared with the sensitivity and selectivity of Pknots-RG when considering the entire sequence for prediction (*Pred. Sen.* and *Pred. Sel.*) as well as the sensitivity and selectivity achieved by our algorithm when selecting those structures with the lowest free energy (*Min En.*, *Min En. Sen.*, and *Min En. Sel.*).

Since we are using Pknots-RG for the prediction of the segments, intuitively we would expect our algorithm to achieve results that are at most equally accurate as those achieved by this prediction code when predicting the whole sequence. However, because we are allowing the prediction of segments starting at different positions in the primary structure, our method can find structures that are very different from those predicted by the code on the entire sequence. Out of the total 39 sequences presented in the figure, the oracle outperformed Pknots-RG on sensitivity and/or selectivity for 24 sequences (see bold values in the figure). However, the selection criteria based on the minimum free energy (*Min En. Sen.* and *Min En. Sel.*) are not as accurate: only in 7 out of the 39 cases did these criteria yield better or equal results than Pknots-RG.

Sequence	Length (nt)	Predictions Attempted	Predictions Used	Rebuilt Time (sec)	Pred. Time (sec)
RF00167_A	100	1701	362	680.40	0.18
RF00374_A	101	1701	351	683.10	0.20
RF00499_A	102	2142	495	831.78	0.21
RF00162_A	103	2142	586	835.47	0.22
RF00198_A	104	2142	451	854.28	0.23
RF00435_A	109	2142	499	905.22	0.26
RF00485_A	114	2628	449	1118.52	0.31
RF00020_A	115	2628	508	1141.65	0.32
RF00001_A	117	2628	678	1168.20	0.23
RF00383_A	117	2628	604	1157.22	0.32
RF00286_A	118	2628	510	1161.36	0.31
RF00463_A	127	3159	824	1490.04	0.42
RF00182_A	129	3159	733	1519.47	0.41
RF00373_A	133	3735	951	1809.54	0.44
RF00290_A	140	3735	974	1924.92	0.53
RF00004_A	145	4356	1277	2348.55	0.61
RF00484_A	149	4356	1246	2376.90	0.66
RF00025_A	152	5022	854	2818.53	0.72
RF00050_A	157	5022	1252	2987.10	0.84
RF00171_A	168	5733	1490	3807.09	1.13
RF00387_A	168	5733	1549	3706.74	1.01
RF00259_A	169	5733	1426	3743.64	1.10
RF00232_A	170	5733	1423	3775.05	1.08
RF00391_A	171	5733	1466	3735.18	0.96
RF00013_A	183	7290	1885	5428.89	1.44
RF00458_A	202	9027	2618	8174.16	2.03
RF00193_A	273	16524	5267	31195.71	6.66
RF00231_A	275	16524	4519	31961.34	6.79
RF00503_A	293	19071	5988	47340.99	9.41
RF00030_A	297	19071	5845	47387.79	9.47
RF00216_A	302	20412	4855	51796.98	9.85
RF00010_A	312	21798	7393	61489.98	10.59
RF00009_A	320	21798	6056	62566.20	11.67
RF00100_A	330	23229	6393	74524.86	13.56
RF00036_A	337	24705	8555	86616.63	14.60
RF00209_A	379	31059	10631	154066.14	22.35
RF00024_A	451	43911	12050	471230.46	53.44
RF00210_A	462	47988	17172	526495.41	50.79
RF00177_A	482	52245	19114	668287.08	59.13

Fig. 13. Performance comparison of predictions performed with our rebuilding algorithm based on sampled segments and the same predictions using Pknots-RG and the entire sequence.

The results of our rebuilding algorithm are promising, especially considering the potential of our approach in overcoming the limitations of current prediction methods on the length and complexity of the sequences. Currently, the most salient weakness of our method involves the selection of the final rebuilt structure. The minimum free energy is not by itself a good factor for selection, probably due to what is already common belief that native structures will often be near-optimal in terms of the minimum free energy.

4.1.2. Effectiveness of sampling approach

In cutting an RNA sequence into segments of overlapping sequences, we experimented with various window sizes and window step sizes. In rebuilding the overall structures from the segments, different threshold values were used. We noticed

Sequence	Length (nt)	Rebuilt Sen	Rebuilt Sel	Pred Sen	Pred Sel	Min En.	Min En. Sen	Min En. Sel
RF00167_A	100	0.73	0.64	1.00	0.79	-23.5	0.64	0.47
RF00374_A	101	**0.81**	0.65	0.81	0.67	-41.6	**0.81**	0.65
RF00499_A	102	0.76	0.71	0.91	0.86	-34.6	0.73	0.59
RF00162_A	103	0.70	0.52	0.85	0.66	-24.1	0.59	0.46
RF00198_A	104	**0.96**	**0.70**	0.92	0.55	-35.4	**0.92**	**0.63**
RF00435_A	109	0.97	0.97	1.00	1.00	-52.9	0.62	0.46
RF00485_A	114	0.54	0.34	0.71	0.37	-24.2	0.33	0.18
RF00020_A	115	0.73	0.58	0.97	0.74	-36.9	0.73	0.58
RF00001_A	117	0.55	0.41	0.82	0.61	-39.4	0.55	0.40
RF00383_A	117	**0.75**	**0.36**	0.75	0.32	-36.7	0.06	0.02
RF00286_A	118	0.86	0.51	0.95	0.57	-37.5	0.86	0.51
RF00463_A	127	**0.80**	**0.80**	0.61	0.42	-53.1	0.41	0.29
RF00182_A	129	0.53	0.36	0.83	0.56	-35.76	0.5	0.3
RF00373_A	133	**0.64**	**0.45**	0.64	0.18	-22.68	0.18	0.05
RF00290_A	140	**0.93**	**0.93**	0.77	0.49	-33.4	0.73	0.54
RF00004_A	145	**0.97**	**0.81**	0.77	0.48	-45.9	**0.77**	**0.48**
RF00484_A	149	**0.58**	**0.44**	0.39	0.21	-40.7	0.27	0.15
RF00025_A	152	**0.73**	**0.53**	0.7	0.49	-21.56	0.55	0.35
RF00050_A	157	0.56	**0.30**	0.68	0.28	-71.0	0.24	0.09
RF00171_A	168	**0.94**	**0.61**	0.91	0.57	-47.5	**0.91**	**0.60**
RF00387_A	168	0.73	0.71	0.96	0.96	-50.34	0.63	0.53
RF00259_A	169	0.52	0.38	0.59	0.47	-33.0	0.34	0.24
RF00232_A	170	0.61	**0.49**	0.63	0.49	-58.9	0.59	0.45
RF00391_A	171	**0.66**	**0.41**	0.5	0.27	-45.3	0.28	0.13
RF00013_A	183	0.72	0.54	0.98	0.87	-62.3	0.72	0.54
RF00458_A	202	**0.78**	**0.64**	0.58	0.39	-49.9	**0.63**	**0.45**
RF00193_A	273	**0.86**	**0.73**	0.79	0.6	-73.6	0.55	0.38
RF00231_A	275	**0.97**	**0.71**	0.71	0.41	-89.0	**0.92**	**0.63**
RF00503_A	293	**0.95**	**0.87**	0.70	0.44	-55.4	0.84	0.63
RF00030_A	297	**0.70**	**0.53**	0.68	0.48	-83.77	0.46	0.29
RF00216_A	302	**0.63**	**0.46**	0.40	0.21	-117.24	0.49	0.29
RF00010_A	312	0.68	0.57	0.77	0.63	-117.7	0.67	0.54
RF00009_A	320	**0.77**	**0.35**	0.57	0.22	-87.4	0.34	0.13
RF00100_A	330	**0.75**	**0.58**	0.40	0.23	-102.9	**0.71**	**0.50**
RF00036_A	337	0.63	0.50	0.94	0.86	-116.04	0.63	0.49
RF00209_A	379	**0.77**	**0.54**	0.75	0.46	-139.1	0.63	0.38
RF00024_A	451	**0.86**	**0.53**	0.80	0.48	-215.2	0.73	0.41
RF00210_A	462	**0.91**	**0.69**	0.80	0.56	-175.5	0.74	0.51
RF00177_A	482	0.82	0.63	0.93	0.74	-239.5	0.72	0.50
Average		0.75	0.58	0.76	0.53		0.59	0.41

Fig. 14. Accuracy comparison (in terms of sensitivity and selectivity) of the upper bound rebuilt predictions based on sampled segments, the same predictions with the entire sequence, and the rebuilt prediction with lowest free energy.

that as values of these parameters vary, the overall accuracy, measured by sensitivity and selectivity of the rebuilt structures, also changes. In order to check whether any significant systematic relationship exists between the accuracy of the rebuilt structures and the parameters, we carried out a multiple regression analysis on each of the 39 sequences in Table 2 with sensitivity and selectivity as response variables and (window size, window step, threshold) as prediction variables.

In all except one sequence, both sensitivity and selectivity are significantly (p-value <0.005) related to the three prediction variables. Both response variables correlate positively with window size, but negatively with window step and threshold. The positive correlation with window size agrees with our expectation that having a larger sequence segment, which constitutes a larger portion of the whole RNA molecule, in a single prediction should generally be beneficial to the accuracy

of the rebuilt structure. On the other hand, a larger window step would mean that successive sequence segments overlap less with each other, so that it is easier to miss those secondary structures spanning both segments but not captured within either one, resulting in the negative correlation with the window step parameter. The negative correlation of threshold with structure accuracy implies that every motif detected in a sequence segment should be taken into account in the rebuilt structure.

A very strong positive correlation between sensitivity and selectivity (correlation coefficient >0.9) has been detected in each of the 39 sequences while the prediction variables are being varied. This suggests that our structure rebuilding approach can be made highly effective simultaneously in both measures of accuracy. It is also interesting to note that the minimum free energy of a rebuilt structure generally shows a negative correlation with sensitivity, but a positive correlation with selectivity, suggesting that the minimum free energy does not necessarily reflect the accuracy of the rebuilt structure. While the minimum free energy is the quantity used pervasively in many secondary structure prediction algorithms for determining the identity of the optimal structure, there seems to be a necessity for seeking an alternative measure.

4.2. Case study II: studying viral RNA genome replication mechanisms for members of the virus family Nodaviridae

For testing the usability of RNAVLab, we considered as our biological system several members of the virus family *Nodaviridae* collectively known as the nodaviruses, a family of tiny, icosahedral viruses with bipartite, single-stranded RNA genomes. The abundant replication and small genomes of these viruses have made them attractive models for the study of virus structure, virus assembly, and RNA replication. The family *Nodaviridae* is comprised of two genera: alphanodaviruses and betanodaviruses. While the betanodaviruses have been isolated only from fish, the alphanodaviruses infect predominantly insects; the alphanodavirus *Nodamura virus* (NoV) also infects mice. Other members of the alphanodavirus genus include *Black beetle virus* (BBV), *Boolarra virus* (BoV), *Flock House virus* (FHV), and *Pariacoto virus* (PaV). The betanodavirus genus includes many members, including the type species of the genus, *Striped jack nervous necrosis virus* (SJNNV), and *Greasy grouper nervous necrosis virus* (GGNNV). These seven viruses were selected here for further study on the basis of the availability of cDNA clones of their genomic RNAs, reagents that will enable us to perform functional assays to determine whether the predicted RNA structures play a role in the viral life cycle. Table 1 shows the abbreviation of the seven viruses, their lengths in terms of nucleotides (nt), and the hosts from which the viruses were isolated, i.e., NoV [35], BBV [36], BoV [37], FHV [38], PaV [39], SJNNV [40], and GGNNV [41].

The nodavirus genome is divided into two segments of positive-strand RNA: RNA1 encodes the viral RNA-dependent RNA polymerase (RdRp) that replicates both genomic segments, while RNA2 encodes the precursor to the protein that comprises the viral outer coat (capsid) [42]. A small subgenomic RNA3, which is not encapsidated into viral particles, encodes a protein that suppresses host defense mechanisms like RNA interference. During viral RNA replication, the genomic RNA is copied first into a complementary negative strand, which is then used as a template for further positive-strand synthesis. The role of RNA secondary structure in the genome replication of other RNA viruses, e.g., members of the plant tombusvirus, potexvirus, and bromovirus families and the animal picornavirus, coronavirus, and flavivirus families, has been well established in the literature. For example, the RdRp of brome mosaic virus initiates negative strand synthesis at a tRNA-like structure at the 3′ end of the positive-sense RNA template [43]. For the Nodavirus family, Kaesberg et al. [44] had previously used the method of Zuker and Stiegler [45] to perform RNA secondary structure analysis on the 3′ noncoding regions of genomic RNA1 and RNA2 of four nodaviruses (BBV, FHV, NoV, and BoV). This method was able only to predict simple stem-loop structures and not pseudoknots. These authors predicted the presence of stem-loop structures near the 3′ terminus of RNA2 for each of these viruses. However, the cloning and sequencing of three additional nodavirus genomes allowed us to revisit the issue of nodavirus RNA secondary structures and technical advances in the field allow us to test the effect of secondary structure on genome replication in cultured cells. The role of secondary structure on nodavirus RNA replication has been studied previously for only one member, *Flock House virus*. A long-range interaction between two regions of RNA1 is required for synthesis of subgenomic RNA3 [46]. The results of genetic experiments suggest that a similar long-range interaction may be also required for synthesis of the RNA3 of another member of the family, *Nodamura virus* (NoV) as well [47]. However, the role of RNA secondary structure in replication of nodavirus genomic RNAs remains unclear. Defining this role is crucial to understanding the mechanism of nodavirus RNA replication. The predictive approaches used in RNAVLab will greatly facilitate our molecular studies by providing a "road map" to elements of possible structural importance, allowing these sequences to be targeted by site-directed mutagenesis.

Previous studies with FHV showed that sequences at the 3′ end of RNA2, particularly within the terminal 50 nucleotides, were critical for RNA replication, and could direct replication of chimeric RNAs that contained heterologous core sequences flanked by RNA2 sequences [48]. By replacing the center of RNA2 with the same heterologous sequence, the work in [48] created a family of RNA molecules that differed only at their termini. The different properties of these molecules could be therefore confidently attributed to these termini. This system established a uniform assay for the different RNAs, using a single probe to the common central core region for Northern blot hybridization experiments. Since such chimeric RNA molecules replicate efficiently, they provide an ideal model substrate for secondary structure prediction and analysis.

4.2.1. Computational results

We used RNAVLab to computationally predict and identify secondary structure motifs in the terminal nucleotides in the 3′ end of RNA2 that could potentially be critical for RNA replication [47]. We analyzed predicted RNA secondary structures of

Table 1
Selected members of the family *Nodaviridae*

Virus	Abbrev.	Natural host	Accession number	Length of RNA2 (nt)
Nodamura	NoV	Mosquito, *Culex tritaeniorhynchus*	NC_002691	1336
Black beetle	BBV	Scarab beetle, *Heteronychus arator*	NC_002037	1399
Boolarra	BoV	Underground grass grub, *Oncopera intricoides*	NC_004145	1305
Flock house	FHV	Grass grub, *Costelytra zealandica*	NC_004144	1400
Pariacoto	PaV	Southern armyworm, *Spodoptera eridania*	NC_003692	1311
Striped jack nervous necrosis	SJNNV	Striped jack, *Pseudocaranx dentex*	NC_003449	1410
Greasy grouper nervous necrosis	GGNNV	Greasy grouper, *Epinephelus tauvina* (Singapore)	NC_004136	1433

progressively shorter lengths from the 3′ end of RNA2 from the seven members of the nodavirus family presented in Table 1. Our goal was to identify common motifs across samples, code predictions, and viruses. Due to the dynamic nature of the prediction programs, the final secondary structures are heavily dependent on neighboring structures: having a certain substructure present in all the predictions, independently from the starting and ending points of the segments, may indicate a strong base pairing that ultimately may be present in nature.

For each virus, the terminal 100 nucleotides of the 3′ end of RNA2 were sampled using the progressive segmentation strategy with a step size of 10 nucleotides. The three different prediction programs currently available in RNAVLab were used, i.e., Pknots-RG, Pknots-RE, and NuPack. All the final predictions, obtained from genome segments from different viruses, with different lengths, and different prediction programs, were processed by the tool in the structure analysis component that allows for identifying common motifs, i.e., pseudoknots or stem-loops, and then were aligned for identification of shifted and overlapping structures. The prediction time for the several secondary structures ranged from several hours for long segments predicted using Pknots-RE to a couple of seconds for short segments predicted using Pknots-RG and NuPack. The predictions were performed in parallel across the pool of machines managed by Condor.

For each of the seven viruses, 10 sample segments per virus were processed for prediction using the three codes, for a total of 210 predictions. Within the 210 predictions, 1982 common motifs were found, ranging from simple motif with a single bond to more complex structures such as pseudoknots. To reduce the number of motifs and identify the most significant ones, overlapping motifs were merged into canonical motifs. Overlapping motifs are those contained within larger motifs in terms of nucleotide length and number of bonds but with the same prediction frequency for the same prediction codes and viral genomes. Moreover, simpler motifs, i.e., hairpins with fewer than 4 base pairs, and less frequent motifs, i.e., motifs that were predicted by a single code or had a frequency below 33%, were not considered. The frequency of a motif is measured as the number of times the motif was indeed predicted over the maximum number of times the motif could be predicted by the three codes in samples with a suitable length to accommodate the motif length and its starting portion. This

Table 2
List of identified common motifs

Virus	Length (nt)	Number of bonds	Frequency	Predictions per code (E, R, N)	Motif
NoV	24	7	0.86	7, 7, 4	(((((((:::::::::::)))))))
BBV	15	10	0.83	5, 6, 4	(((((:::::)))))
BBV	16	10	0.63	5, 5, 5	(((((::::::)))))
BoV	16	6	1.00	9, 9, 9	((((((:::::))))))
BoV	17	5	0.67	5, 1, 7	[[[[[:::::::]]]]]
BoV	56	9	0.56	3, 0, 2	((((:::::))))::::::::::::::::::::::::::[[[[[:::::::]]]]]
BoV	34	11	0.39	4, 0, 3	[[[[[:::::::]]]]]:((((((:::::))))))
FHV	14	10	0.78	6, 6, 2	(((((:::::)))))
FHV	12	8	0.67	7, 5, 4	((((:::::))))
FHV	56	11	0.42	2, 3, 0	[[[[[[:::::::::((((((:::::))))))::::::::::::::::::::]]]]]]
FHV	64	9	0.42	2, 3, 0	(((:::::[[[[[[]))):::::::::::::::::::::::::::::::::::::::]]]]]]
FHV	56	6	0.42	2, 3, 0	[[[[[[:::]]]]]]
BoV	24	9	0.39	4, 0, 3	((((:::[[[[[]))))):::]]]]]
PaV	18	10	0.67	5, 4, 1	((((::::::::))))
PaV	30	7	0.39	4, 1, 2	[[[[::::::::::]]]]:(((::::)))
PaV	19	4	0.39	4, 1, 2	[[[[::::::::::]]]]
PaV	26	8	0.33	4, 1, 1	((((::[[[[:))))::::::]]]]
GGNNV	24	10	0.56	4, 5, 1	(((((:[[[[[]))))):::]]]]
GGNNV	22	5	0.54	8, 4, 1	(((((::::::::::::)))))
GGNNV	41	10	0.50	4, 4, 1	[[[[[::::::::]]]]]:(((((::::::::::::)))))
GGNNV	18	5	0.48	4, 5, 1	[[[[[::::::::]]]]]
SJNNV	21	6	0.83	2, 2, 1	(((:::(((::::)))::)))
SJNNV	42	11	0.50	6, 3, 0	[[[[[::::::::]]]]](((((::::::::::::)))))
SJNNV	24	10	0.50	6, 3, 0	(((((:[[[[[]))))):::]]]]]
SJNNV	24	6	0.43	6, 3, 0	((((((::::::::::::))))))
SJNNV	18	5	0.43	6, 3, 0	[[[[[::::::::]]]]]

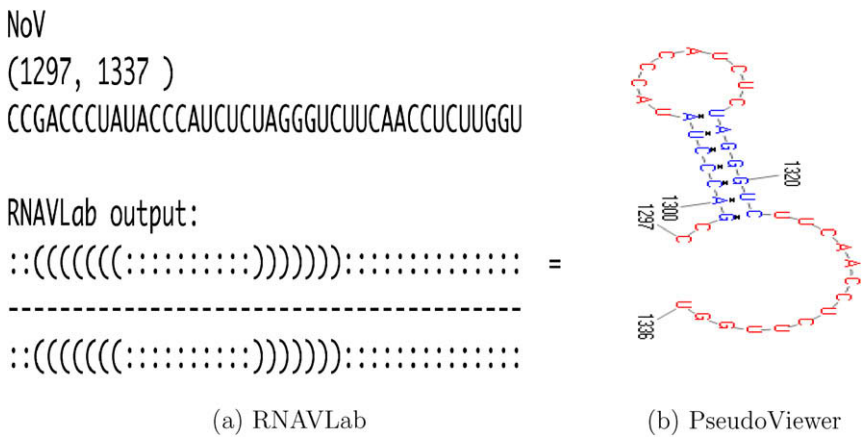


Fig. 15. RNAVLab output and PseudoViewer result for NoV.

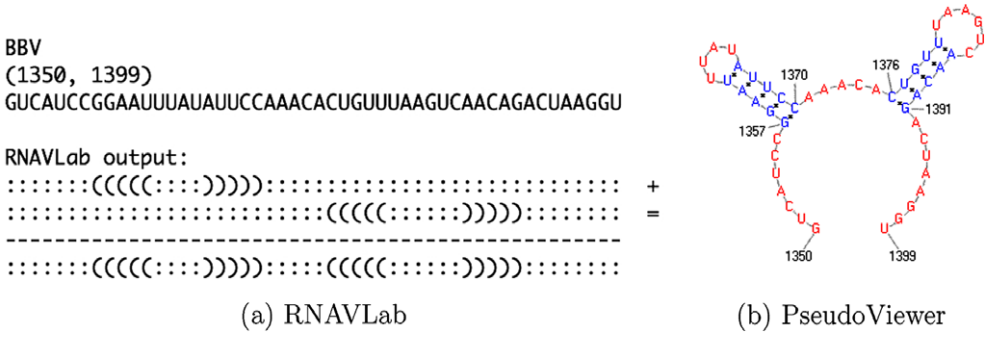


Fig. 16. RNAVLab output and PseudoViewer result for BBV.

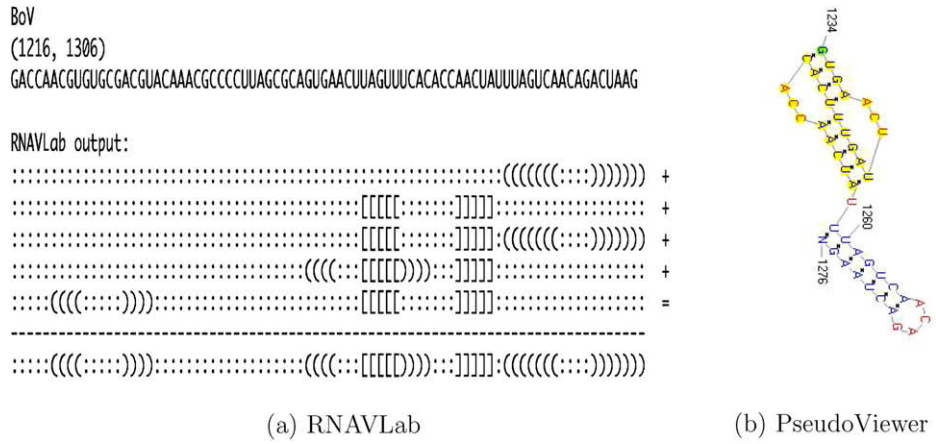


Fig. 17. RNAVLab output and PseudoViewer result for BoV.

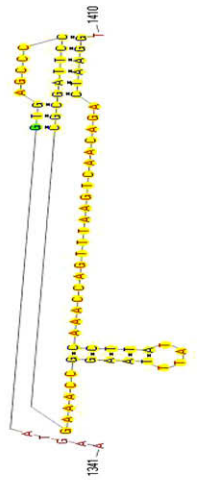
approach reduced the number of common motifs to a set of 28 motifs shown in Table 2. For each virus, Table 2 presents the motif length, the number of nucleotide bonds, the frequency of the predictions across codes and samples, the number of predictions per code (where E stands for Pknots-RE, R stands for Pknots-RG, and N stands for NuPack), and the motif secondary structures in terms of brackets.

The alignment of the 28 motifs allows for rebuilding larger motifs. Figs. 15–21 show the output of the RNAVLab analysis (a) and the PseudoViewer representation of the RNA2 terminal part for each virus with the common motif that has resulted from the analysis of the several predictions (b).

FHV
(1341, 1401)
AAGTAGTGAGCCCCCTTAGCGCGAAACCGGAATTTATATTCCAAACCAGTTTAAGTCAACAGACTAAGGT

RNAVLab output:

```
:::::::::::::::::::::::::::::((((((::::)))))):::::::::::::::::::::::::::::  +
:::::::::::::::::::::::::::::::::::::::::::::::((((::::)))):::::::::::::::  (+)
:::::::::::::[[[[[[:::::::::((((((::::)))))):::::::::::::::::::::]]]]]]:  +
:::::(((:::::[[[[[[)))::::::::::::::::::::::::::::::::::::::::::::]]]]]]:  +
:::::::::::::[[[[[[::::::::::::::::::::::::::::::::::::::::::::::]]]]]]:  =
------------------------------------------------------------------------
:::::(((:::::[[[[[[)))::::::((((((::::)))))):::::::::::::::::::::]]]]]]:
```



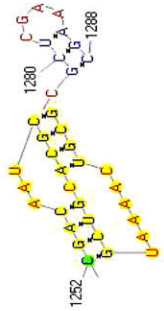
(a) RNAVLab (b) PseudoViewer

Fig. 18. RNAVLab output and PseudoViewer result for FHV.

PaV
(1252, 1312)
CGACAAUCGCACGUCGUAAAACUGCGCGCUCGAAAGCUCAUAAAAGAAACAACCAUGGCC

RNAVLab outputs:

```
((((::::::::))))::::::::::::::::::::::::::::::::::::::::::::::::::::::::  +
:::::::[[[[::::::::::::]]]]:(((::::)))::::::::::::::::::::::::::::::::::  +
:::::::[[[[::::::::::::]]]]::::::::::::::::::::::::::::::::::::::::::::  +
((((:::[[[[:))))::::::]]]]::::::::::::::::::::::::::::::::::::::::::::  =
------------------------------------------------------------------------
((((:::[[[[:))))::::::]]]]:(((::::)))::::::::::::::::::::::::::::::::::
```



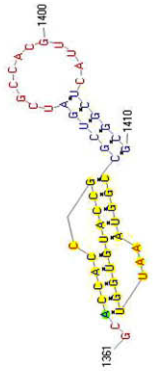
(a) RNAVLab (b) PseudoViewer

Fig. 19. RNAVLab output and PseudoViewer result for PaV.

SJNNV
(1321, 1411)
UCUUUGGGCUUGUUGGGUUACCGUUAGCUCCGCGCAGUGAGCACCACCGCCAUGUGGUUAAAUGGCCGCUGAUCGCCACGUUACUCGGCG

RNAVLab outputs:

```
:::::::(((:::(((::::)))::)))::::::::::::::::::::::::::::::::::::::::::::::::::::::  +
::::::::::::::::::::::::::::::::::::::::::::::::[[[[[::::::::]]]]](((((((::::::::::::)))))))  +
::::::::::::::::::::::::::::::::::::::::::::::(((((:[[[[[)))))):::]]]]::::::::::::::::::::::  +
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::(((((((::::::::::::)))))))  +
::::::::::::::::::::::::::::::::::::::::::::::::[[[[[::::::::]]]]]::::::::::::::::::::::::::  =
------------------------------------------------------------------------------------------
:::::::(((:::(((::::)))::)))::::::::::::::(((((:[[[[[)))))):::]]]](((((((::::::::::::)))))))
```



(a) RNAVLab (b) PseudoViewer

Fig. 20. RNAVLab output and PseudoViewer result for SJNNV.

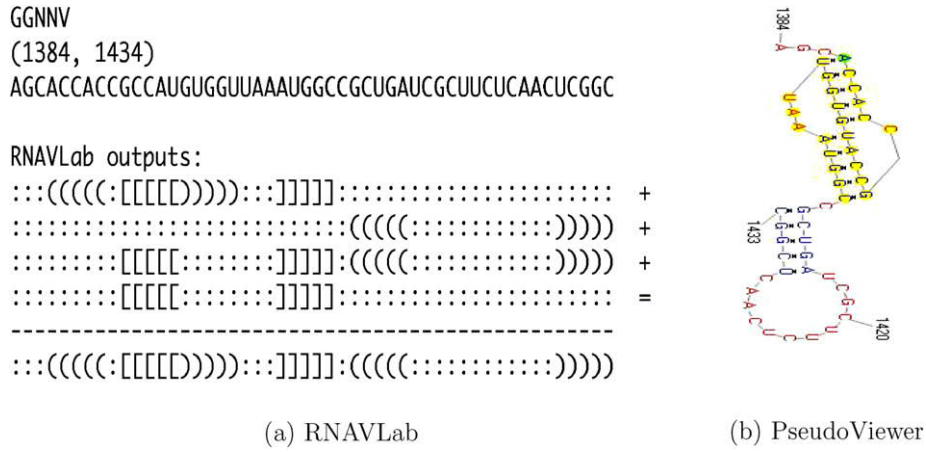


Fig. 21. RNAVLab output and PseudoViewer result for GGNNV.

For NoV (Fig. 15) and BBV (Fig. 16), RNAVLab consistently predicts the presence of stem-loop structures near the 3′ end of RNA2. In particular, for NoV a hairpin is predicted from nt 1299 to nt 1322, within the last 50 nucleotides of RNA2. For segments shorter than 40 nucleotides, the stem-loop is no longer predicted. For BBV, we predict two hairpins within the last 50 nucleotides: from nt 1357 to nt 1370 and from nt 1376 to nt 1391, respectively.

For both SJNNV (Fig. 20) and GGNNV (Fig. 21), RNAVLab identifies a pseudoknot followed by a hairpin at the terminal part of their RNA2 segment: while the pseudoknot is identical, the terminal hairpin in SJNNV resembles the hairpin in GGNNV but it has an additional pair of nucleotides. A similar pseudoknot followed by a hairpin is also found at the end of BoV (Fig. 17) and at the terminal part of PaV (Fig. 18), although for the latter virus, this structure is followed by a tail of 23 nucleotides that is predicted to fold into a variety of hairpins of different sizes that vary with the program used. A pseudoknot is found at the very end of FHV (Fig. 19) together with a hairpin that is located inside the pseudoknot to form an HL-out type pseudoknot.

These results found with RNAVLab are consistent with the work conducted experimentally in [48]. We are currently studying whether the pseudoknots and stem-loops are indeed critical structures that drive the genome replication of these viruses. To address this critical question we are combining computational and experimental methods. Driven by the computational results presented in this paper, we will ultimately address the question of whether the biological relevance of the pseudoknots and stem-loops can indeed be experimentally verified. This will be accomplished using two approaches. First, to determine whether the predicted structures can be verified in solution, we will generate a nuclease map using single- and double-strand-specific ribonucleases (RNases) as described by Tuplin et al. in [49]. Viral RNA will be transcribed in vitro and treated with RNases A, T1, and V1, to cleave the 3′ end of single-stranded uracils and cytosines, the 3′ end of single-stranded guanines, and double-stranded nucleotides, respectively. A labeled primer will be used for reverse-transcription of the digested RNAs and sequencing of the undigested cDNA transcription templates. This technique will allow us to confirm or refute predicted structural elements.

Second, to test the role of these predicted structures in the viral life cycle, we will use a process called reverse genetics. Because it is technically difficult to make changes to RNA directly, we will use cloned complementary DNA (cDNA) copies of the viral genomic RNA instead. We will use site-directed mutagenesis to make specific deletions and substitutions of nucleotides within the predicted structural elements. The altered cDNAs will be introduced into cultured cells, where they will be transcribed into RNA by cellular DNA-dependent RNA polymerases. These primary RNA transcripts will then be able to replicate in the presence of the viral RNA-dependent RNA polymerase (RdRp). Therefore, we will test the effect of the RNA2 mutations by transfecting the mutant form into cells together with RNA1 as a source of RdRp. Following incubation to allow the viral RNA to replicate, we will isolate total cellular RNA from the cells and assay for the presence of RNA replication products (negative strand intermediates of RNA1 and RNA2, together with production of subgenomic RNA3) using strand-specific Northern blot hybridization. For example, to test the function of the predicted stem-loop near the 3′ end of NoV RNA2, we could delete it altogether or change specific nucleotides to disrupt base pairing interactions within the predicted stem. If these structures are important for viral RNA replication, we will observe a decrease in the level of replication products detected (relative to wild-type RNA2 controls). Together these methods will allow us to experimentally test whether the structures we predict in the RNAVLab are physically present in the viral RNAs and whether they have functional significance in the viral life cycle.

5. Conclusions and future work

RNAVLab is a virtual laboratory that facilitates the study of RNA secondary structures, i.e., prediction, alignment, comparison, identification, and classification of common secondary structures motifs across viruses, through an automated, compu-

tationally powerful approach: the scientist's intervention is minimized and grid computing technologies are used to address computing intensive tasks such as the prediction of long RNA secondary structures including pseudoknots.

In this paper, RNAVLab is used for rebuilding long secondary structures from significant motifs in RNA segments as well as the computational study of mechanisms that guide RNA replication in the virus family *Nodaviridae*. For 24 of the 39 RNA sequences with different lengths that we used for the validation of our rebuilding approach, we obtained more accurate (either in terms of sensitivity or selectivity or both) rebuilt structures than those predicted by the same code applied to the whole sequences. The regression analysis results indicated that (1) there is a significant relationship between the accuracy of the rebuilt structure (in terms of sensitivity and selectivity) and the sampling factors (i.e., window size, window step, and threshold values); (2) our method equally targets both the measurements of accuracy, i.e., sensitivity and selectivity; and (3) the minimum free energy cannot be trusted by itself as a quality measure of the secondary structure.

By predicting RNA secondary structures of progressively shorter lengths from the 3′ end on the *Nodamura virus* RNA2, RNAVLab indicates that, across prediction programs and with different sampled segments, a hairpin structure from nt 1299 to nt 1322 is consistently predicted in the terminal segment of the RNA2 for the *Nodamura virus* and two hairpins are predicted for the *Black beetle virus*, from nt 1273 to nt 1286 and from nt 1292 to nt 1307, respectively. These results for NoV and BBV are consistent with the prediction described in [44] except that their prediction included a different fold for BBV nt 1376-1391. Similar secondary structures, i.e., a pseudoknot followed by a hairpin, are predicted in the other members of the family, i.e., *Boolarra virus*, *Pariacoto virus*, *Striped jack nervous necrosis virus*, *Greasy grouper nervous necrosis virus*. Only in the *Flock house virus* the pseudoknot includes the hairpin.

Ongoing work includes studying whether the computational findings in both the case studies can indeed be experimentally verified. If this is the case, RNAVLab will be a powerful tool to drive molecular studies by providing a "road map" to elements of possible structural importance, allowing these sequences to be targeted by site-directed mutagenesis.

Acknowledgments

This material is based in part upon work supported by the Texas Advanced Research Program under Grant Nos. 003661-0008-2006 and 0036661-0008-2007, and the National Institutes of Health under Grant Nos. S06GM08012-37 (NIH-GM), 5G12RR008124-11 (NIH-NCRR) number corrected), and 3T34GM008048-20S1. Financial support through the National Science Foundation (Grant DUE-631168, "SHiPPER: Spreading High-Performance computing Participation in undergraduate Education and Research" and Grant DMS-0800266, "Mathematical Models for RNA") is acknowledged. We would also like to thank the BBRC DNA Analysis Core Facility for services and facilities provided.

References

[1] V. Thiel et al, Mechanisms and enzymes involved in SARS coronavirus genome expression, J. Gen. Virol. 84 (2003) 2305–2315.
[2] M. Petrillo, G. Silvestro, P.P. Di Nocera, A. Boccia, G. Paolella, Stem-loop structures in prokaryotic genomes, BMC Genom. 7 (2006) 170.
[3] M.-C. Su et al, An atypical RNA pseudoknot simulator and an upstream attenuation signal for −1 ribosomal frameshifting of SARS coronavirus, Nucleic Acids Res. 33 (13) (2005) 4265–4275.
[4] S.R. Wilkinson, M.D. Been, A pseudoknot in the 3′ non-core region of the glmS ribozyme enhances self-cleavage activity, RNA 11 (2005) 1788–1794.
[5] D. Sankoff, Simultaneous solution of the RNA folding, alignment, and protosequence problems, SIAM J. Appl. Math. 45 (1985) 810–825.
[6] M. Zuker, Computer prediction of RNA structure, Methods Enzymol. 180 (1989) 262–288.
[7] M. Zuker, D. Mathews, D. Turner, Algorithms and thermodynamics for RNA secondary structure prediction: a practical guide, in: RNA Biochemistry and Biotechnology, Kluwer Academic Publishers, 1999.
[8] E. Rivas, S.R. Eddy, A dynamic programming algorithm for RNA structure prediction including pseudoknots, J. Mol. Biol. 285 (1999) 2053–2068.
[9] J. Reeder, R. Giegerich, Design, implementation, and evaluation of a practical pseudoknot folding algorithm based on thermodynamics, BMC Bioinform. 5 (2004) 104.
[10] J. Ruan, G.D. Stormo, W. Zhang, ILM: a web server for predicting RNA secondary structures with pseudoknots, Nucleic Acids Res. 32 (Web Server issue) (2004) W146–W149.
[11] J. Ren, B. Rastegari, A. Condon, H.H. Hoos, HotKnots: heuristic prediction of RNA secondary structures including pseudoknots, RNA 11 (2005) 1494–1504.
[12] F.H.D. Batenburg van et al, PseudoBase: a database with RNA pseudoknots, Nucleic Acids Res. 28 (1) (2000) 201–204.
[13] D. Chew, K. Choi, H. Heidner, M. Leung, Palindromes in SARS and other coronaviruses, INFORMS J. Comput. 16 (2004) 331–340.
[14] T. Estrada, A. Licon, M. Taufer, CompPknots: a framework for parallel prediction and comparison of RNA secondary structures with pseudoknots, in: Proc. First Frontier High Perf. Comp. Network. Workshop, 2006.
[15] B.A. Shapiro, An algorithm for comparing multiple RNA secondary structures, Comput. Appl. Biosci. 4 (3) (1988) 387–393.
[16] B.A. Shapiro, K.Z. Zhang, Comparing multiple RNA secondary structures using tree comparisons, Comput. Appl. Biosci. 6 (4) (1990) 309–318.
[17] M. Taufer et al, Predictor@Home: a protein structure prediction supercomputer based on global computing, IEEE Trans. Parallel Distrib. Syst. 17 (8) (2006) 786–796.
[18] B. Zagrovic, C.D. Snow, M.R. Shirts, V.S. Pande, Simulation of folding of a small alpha-helical protein in atomistic detail using worldwide distributed computing, J. Mol. Biol. 323 (2002) 927–937.
[19] I.L. Hofacker, Vienna RNA secondary structure server, Nucleic Acids Res. 31 (13) (2003) 3429–3431.
[20] B. Knudsen, J. Klein, RNA secondary structure prediction using stochastic context-free grammars and evolutionary history, Bioinformatics 15 (6) (1999) 446–454.
[21] C.B. Do, D. Woods, S. Batzoglou, CONTRAfold: RNA secondary structure prediction without physics-based models, Bioinformatics 22 (14) (2006) e90–e98.
[22] T. Nguyen, M. Turcotte, Exploring the space of RNA secondary structure motifs using suffix arrays, in: Proc. Sixth Int. Sympos. Comput. Biol. Genome Inform. (CBGI'05), 2005.
[23] M. Anwar, T. Nguyen, M. Turcotte, Identification of consensus RNA secondary structures using suffix arrays, BMC Bioinform. 7 (2006) 244.
[24] B. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochim. Biophys. Acta (405) (1975) 442–451.

[25] D. Ashlock, J. Schonfeld, Depth annotation of RNA folds for secondary structure motif search, in: Proc. 2005 IEEE Sympos. Comput. Intell. Bioinf. Comp. Biology, 2005.
[26] O. Bergig, D. Barash, K. Kedem, RNA motif search using the structure to string str^2 method, in: Proc. 2004 IEEE Comp. Syst. Bioinf. Conf., 2004.
[27] M.-Y. Leung et al, Nonrandom clusters of palindromes in herpesvirus genomes, J. Comput. Biol. 12 (3) (2005) 331–354.
[28] N.A. Pierce, NuPack: a software suite for the analysis and design of nucleic acids, 2006. <http://www.nupack.org>.
[29] D. Thain, T. Tannenbaum, L. Livny, Distributed computing in practice: the Condor experience, Concurr. Comput.: Practice Exp. 17 (2–4) (2004) 323–356.
[30] D.P. Anderson, BOINC: a system for public-resource computing and storage, in: Proc. Fifth IEEE/ACM Int. Workshop Grid Comput. (GRID'04), 2004.
[31] K. Han, Y. Byun, PseudoViewer2: visualization of RNA pseudoknots of any type, Nucleic Acids Res. 31 (2003) 3432–3440.
[32] T. Smith, M. Waterman, Identification of common molecular subsequences, J. Mol. Biol. 147 (1981) 195–197.
[33] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequences of two proteins, J. Mol. Biol. 48 (1970) 444–453.
[34] M. Anwar, M. Turcotte, An approach to selecting putative RNA motifs using MDL principle, in: Proc. 2006 Int. Conf. Bioinform. Comput. Biol. (BIOCOMP'06), 2006.
[35] W.F. Scherer, H.S. Hurlbut, Nodamura virus from Japan: a new and unusual arbovirus resistant to diethyl ether and chloroform, Am. J. Epidemiol. 86 (1967) 271–285.
[36] J.F. Longworth, G.P. Carey, A small RNA virus with a divided genome from *Heteronychus arator* (F.) [Coleoptera: Scarabaeidae], J. Gen. Virol. 33 (1975) 31–40.
[37] C. Reinganum, J.B. Bashiruddin, G.F. Cross, Boolarra virus: a member of the *Nodaviridae* isolated from *Oncopera intricoides* (Lepidoptera: Hepialidae), Intervirology 24 (1985) 10–17.
[38] P.D. Scotti, S. Dearing, D.W. Mossup, Flock House virus: a nodavirus isolated from *Costelytra zealandica* (White) (Coleoptera: Scarabaeidae), Arch. Virol. 75 (1983) 181–189.
[39] J.-L. Zeddam, J.L. Rodriguez, M. Ravallec, A. Lagnaoui, A noda-like virus isolated from the sweet potato pest *Spodoptera eridania* (Cramer) (Lepidoptera: Noctuidae), J. Invertebr. Pathol. 74 (1999) 267–274.
[40] K.-I. Mori, T. Nakai, K. Muroga, M. Arimoto, K. Mushiake, I. Furusawa, Properties of a new virus belonging to *Nodaviridae* found in larval striped jack (*Pseudocaranx dentex*) with nervous necrosis, Virology 187 (1992) 368–371.
[41] C. Tan, B. Huang, S.F. Chang, G.H. Ngoh, B.L. Munday, S.C. Chen, J. Kwang, Determination of the complete nucleotide sequences of RNA1 and RNA2 from greasy grouper (*Epinephelus tauvina*) nervous necrosis virus, Singapore strain, J. Gen. Virol. 82 (2001) 647–653.
[42] A. Schneemann, L.A. Ball, C. Delsert, J.E. Johnson, T. Nishizawa, *Nodaviridae*, in: C.M. Fauquet, M.A. Mayo, J. Maniloff, U. Desselberger, L.A. Ball (Eds.), Virus Taxonomy, Eighth Report of the International Committee on Taxonomy of Viruses, Elsevier Academic Press, San Diego, CA, 2005, pp. 865–872.
[43] W.A. Miller, J.J. Bujarski, T.W. Dreher, T.C. Hall, Minus-strand initiation by brome mosaic virus replicase within the 3′ tRNA-like structure of native and modified RNA templates, J. Mol. Biol. 187 (1986) 537–546.
[44] P. Kaesberg, R. Dasgupta, J.-Y. Sgro, J.-P. Wery, B.H. Selling, M.V. Hosur, J.E. Johnson, Structural homology among four nodaviruses as deduced by sequencing and x-ray crystallography, J. Mol. Biol. 214 (1990) 423–435.
[45] M. Zuker, P. Stiegler, Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information, Nucleic Acids Res. 9 (1981) 133–148.
[46] B.D. Lindenbach, J.-Y. Sgro, P. Ahlquist, Long-distance base pairing in Flock House virus RNA1 regulates subgenomic RNA3 synthesis and RNA2 replication, J. Virol. 76 (2002) 3905–3919.
[47] K.L. Johnson, B.D. Price, L.A. Ball, Recovery of infectivity from cDNA clones of Nodamura virus and identification of small nonstructural proteins, Virology 305 (2003) 436–451.
[48] C.G. Albarino et al, The cis-acting replication signal at the 3′ end of Flock House virus RNA2 is RNA3-dependent, Virology 311 (2003) 181–191.
[49] A. Tuplin, D.J. Evans, P. Simmonds, Detailed mapping of RNA secondary structures in core and NS5B-encoding region sequences of hepatitis C virus by RNase cleavage and novel bioinformatic prediction methods, J. Gen. Virol. 85 (2004) 3037–3047.