Published in final edited form as:
*Proceedings (IEEE Int Conf Bioinformatics Biomed)*. 2011 December 31; 2011: 618–622. doi:10.1109/BIBM.2011.74

# R-PASS: A Fast Structure-based RNA Sequence Alignment Algorithm

**Yanan Jiang**[a], **Weijia Xu**[b], **Lee Parnell Thompson**[c], **Robin R. Gutell**[a], and **Daniel P. Miranker**[c]

Yanan Jiang: yanan.jiang@utexas.edu; Weijia Xu: xwj@tacc.utexas.edu; Lee Parnell Thompson: parnell@cs.utexas.edu; Robin R. Gutell: robin.gutell@mail.utexas.edu; Daniel P. Miranker: miranker@cs.utexas.edu

[a]Institute for Cellular and Molecular Biology, The University of Texas at Austin, Austin, USA

[b]Texas Advanced Computing Center, The University of Texas at Austin, Austin, USA

[c]Department of Computer Science, The University of Texas at Austin, Austin, USA

## Abstract

We present a fast pairwise RNA sequence alignment method using structural information, named R-PASS (RNA Pairwise Alignment of Structure and Sequence), which shows good accuracy on sequences with low sequence identity and significantly faster than alternative methods. The method begins by representing RNA secondary structure as a set of structure motifs. The motifs from two RNAs are then used as input into a bipartite graph-matching algorithm, which determines the structure matches. The matches are then used as constraints in a constrained dynamic programming sequence alignment procedure. The R-PASS method has an O(nm) complexity. We compare our method with two other structure-based alignment methods, LARA and ExpaLoc, and with a sequence-based alignment method, MAFFT, across three benchmarks and obtain favorable results in accuracy and orders of magnitude faster in speed.

### Keywords

RNA pairwise structural alignment; structure motif; bipartite graph matching; constraint sequence alignment

## I. Introduction

A trend in the sequence analysis is to process increasingly larger scales of sequences in order to detect sequence homologues, predict consensus secondary structures[1], identify structure motifs[2] and infer phylogenetic relationships[3]. Non-canonical base pairs and structure motifs are found based on a MSA of 2,240 16S rRNA sequences[2] and a set of statistical free energy values are computed from more than 50,000 ncRNA sequences[4]. Furthermore, genome wide sequence alignments identifying ncRNAs have become increasingly routine. RNAz[5] predicted over 30,000 structured RNA elements in human genomes from 438,788 alignments of non-coding regions. Scalable and fast computation method is a key to make large scale analysis feasible.

Sequence-based RNA sequence alignment programs, e.g. MAFFT[6], generate accurate alignments when the RNA sequences are conserved. However, these programs are unable to

produce reliable alignments when sequence identity drops below 50–60%[7]. Exploiting the phenomenon of the coevolution of base-pairs and the preservation of secondary structure are promising approaches to improve RNA alignment accuracy[7].

Although many structure-based programs exist, most of them have high complexity and are not applicable to long RNAs. In this paper, we present a method, R-PASS (RNA Pairwise Alignment of Structure and Sequence). We evaluated our method compared with two state-of-art structure-based alignment programs, LARA[8] and ExpaLoc[9], and a popular sequence-based alignment program MAFFT. Of the programs tested, R-PASS is the fastest. The results also show improved accuracy upon MAFFT and ExpaLoc and comparable accuracy with LARA.

## II. RELATED WORK

Most structure-based alignment programs continue the tradition of the Sankoff's algorithm[10], where it simultaneously folds and aligns a set of pseudo knot-free RNAs using a dynamic programming approach (DP). Although efforts have been made to reduce the time and space complexity, this approach still requires $O(n^4)$ time in the pairwise alignment case. Thus most structure-based programs are not practical for long RNAs[8].

R-PASS assumes the structure information is available for both sequences. We compare our program to the two most recent structure based alignment programs that target the same problem, LARA[8] and ExpaLoc[9]. LARA adopts a graph-based representation and models the alignment as an integer linear program. ExpaLoc combines ExpaRNA[9] and LocARNA[11], where ExpaRNA detects the longest exact pattern match of two RNA structures and LocARNA fills in the unaligned space between those patterns.

Our program differs with LARA in that the RNA structures are matched at the structure motif level instead of at the nucleotide level and an optimal alignment is found by a DP algorithm. Unlike ExpaRNA which finds the exact pattern matches, our matching algorithm is more flexible, so more structure constraints can be used in alignment construction. Also, the alignment building process in our program is still sequence-based, and thus has a much lower computation complexity than LocARNA.

To evaluate the effectiveness of using additional structure information, we also compare our program with MAFFT[6]. Its iterative refinement method L-INS-I is evaluated to be one of the most accurate sequence-based alignment programs that produce high quality alignments with average pairwise sequence identity above 55%[12].

## III. ALGORITHM

Given two RNA sequences with known secondary structures, we parse the annotated base pairs into a set of structure motifs. The feature vectors of the structure motifs are computed and form the vertices of a bipartite graph. The weight of an edge is based on the similarity of two feature vectors. A set of edges which represent the correspondence between structural motifs are obtained by a bipartite graph matching algorithm. These correspondences are then applied as anchor points to construct an optimal sequence alignment using a DP algorithm.

## A. Structure Matching

**1) Infering motifs from base pairing annotation—**The secondary structure of a RNA sequences consists of a set of nested base pairs. The majority of the nucleotides in a RNA structure form canonical base pairings in a regular helix region. The remaining unpaired nucleotides form loops, which can be further categorized to bulges, internal, hairpin and multi-stem loops. Thus a RNA secondary structure can be viewed as a set of those RNA motifs.

Stem is the continuous base-paired double strand region and is composed of a 5' end half and a 3' end half. Bulge is a loop appearing only on one strand of a stem. Internal loop is the unpaired regions occurring on both strands of a stem. Hairpin loop is the loop linking the 5' end and 3' end of a stem. Multi-stem loop is the intersection of three or more stems. It is separated by single strand sequences. Free end is the unpaired region at the 5' end or 3' end of a structure.

Given a RNA secondary structure annotated by its base pairings, we first segment it into Stem, Bulge, Internal loop and Hairpin loop motifs. Then we merge the basic motifs into Compound Stem and Hairpin. A Compound Stem is the discontinuous base-paired region merged by Stems, Bulges and Internal loops. A Hairpin is formed by combining a Hairpin loop and its flanking Stem/Compound Stem. The Compound Stem, Stem, Hairpin, Multi-stem loop and Free end form a complete RNA structure. To reduce computation, we only consider Compound Stem, Stem and Hairpin motifs in the following matching step. The other motifs are spontaneously matched after matching of those three motifs.

**2) Vector model of motifs—**For each Stem/Compound Stem/Hairpin motif we compute a feature vector which includes the start ($p_s$) and end position ($p_e$) of the motif and the number of nucleotides in it ($l$). Since a Stem/Compound Stem consist of two halves, the end position of the first half ($p_{se}$) and the start position of the second half ($p_{es}$) are also included in its feature vector. The feature vector of a Hairpin is the same as the feature vector of its stem component. Therefore $f_{\{Stem, \textbf{Compound Stem, Ha}irpin\}} = (p_s, p_{se}, p_{es}, p_e, l)$. A RNA secondary structure $S$ can thus be represented as $S = \{f_{ti}\}, i \in (1, k)$, where $k$ is the number of motifs in $S$ and $t \in$ {Stem, Compound Stem, Hairpin}.

The similarity of two structure motifs is measured by how close their relative positions are in the corresponding global sequences and how similar their lengths are. The similarity is only computed between motifs of the same type, i.e. Hairpin with Hairpin and Stem with Stem or Compound Stem. We define the following heuristic similarity function composed of position similarity (*pos*) and length similarity (*len*), where $s_l$ is the global sequence length, and $w_l$ is the weight of length similarity.

$$d(f_a, f_b) = \begin{cases} 0, & \text{if } a, b \text{ are not of the same type} \\ pos(f_a, f_b).len(f_a, f_b), & otherwise \end{cases}$$

$$len(f_a, f_b) = 1 - w_l \cdot \left| \frac{l_a - l_b}{l_a + l_b} \right|$$

$$pos(f_a, f_b) = 1 - max \left( \begin{array}{c} min \left( \left| \frac{p_{sa}}{sl_a} - \frac{p_{sb}}{sl_b} \right|, \left| \frac{p_{sea}}{sl_a} - \frac{p_{seb}}{sl_b} \right| \right), \\ min \left( \left| \frac{p_{esa}}{sl_a} - \frac{p_{esb}}{sl_b} \right|, \left| \frac{p_{ea}}{sl_a} - \frac{p_{eb}}{sl_b} \right| \right) \end{array} \right), \text{if } a, b \in Stem$$

In the above equation, the score for different types of the structural motif is always 0. For the comparison of the same motif type, the score is the product of the relative position similarity and the length similarity. The relative position difference for a single strand is defined as the minimum difference as measured by start and end position. For a Stem/Compound Stem structure, the relative position difference is first evaluated on each strand individually and then the maximum score between the two strands is used. For simplicity, only the weight on length difference ($w_l$) is used. The similarity score ranges between 0 and 1 inclusively.

**3) Structure motif matching algorithm—**The structure motifs from two RNA structures are matched by a bipartite graph matching (BGM). BGM is a powerful matching technique that can achieve global and optimal matching results in polynomial time. It has been applied to protein structure alignment[13].

We denote a bipartite graph $G$ as $G = (V = L \cup R, E)$. $V$ is a set of vertices which can be divided into two subsets, $L$ and $R$. Each subset individually represents structure motifs from one sequence and contains their feature vectors as vertices, i.e. $L = S_1$, $R = S_2$. $E$ is a set of weighted edges that link between vertices in $L$ and vertices in $R$. The edge weight is the similarity score between two feature vectors. A threshold for edge weight $t$ is used in a preprocessing step of graph construction, so only edges with weight no less than $t$ are used for matching. The limitation $t > 0$ ensures that the matches are only between motifs of the same type. There is no edge linking vertices from the same subset. Hence, $E = \{e_{Li,Rj} = d(f_{Li}, f_{Rj}) \geq t\}$, $i \in (1, k)$, $j \in (1, h)$, where $k$ is the number of motifs in $S_1$ and $h$ is the number of motifs in $S_2$.

Based on this bipartite graph, the correspondence of structure motifs are then found by a stable matching algorithm[14]. The algorithm generates a set of matched structure motifs, more specifically the feature vectors of structure motifs, $M = \{f_{Li}, f_{Ri}\}$, $i \in (1, c)$, where $c$ is the number of matches. The stable matching ensures that no two motifs are better matched together than with the motif they are currently matched with. The complexity of this algorithm is $O(kh)$. Any graph matching algorithm could be used as a substitute for the stable matching, such as a maximum weighted matching algorithm. The stable matching was chosen for its fast runtime.

**4) Computing matching blocks—**The matching structure motifs are then converted into sequence blocks. The boundaries of the matched motifs are obtained based on their feature

vectors. For a Hairpin motif, its sequence is divided into three parts, 5' end stem, Hairpin loop and 3' end stem, and the Stem and Compound Stem motif are segmented into the 5' end stem and 3' end stem. Thus the matching segments from two sequences form a match block, of which the boundaries are determined by the start and end positions of the segments (Fig. 1).

The above generated sequence block set may contain crossing blocks caused by mismatches in BGM. In cases where the structures are unknown and require prediction, overlapping blocks may also appear if the predicted structures contain a set of overlapping candidate motifs. To solve this problem, the Dijkstra's shortest path algorithm is applied[15]. We construct a graph where the vertices are the blocks and two pseudo blocks representing the start and the end of the sequences. The edge weight is the distance between two blocks. It is set to be 0 for crossing blocks and the total number of position gaps between the blocks on both sequences for non-crossing blocks. Only edges with positive weight are added to the graph. A shortest path is then computed between the start block and the end block. The match blocks on the path are used in the next step.

### B. Constraint sequence alignment

For sequence $A = \{a_i\}$ and $B = \{b_j\}$, $i \in (1, n)$, $j \in (1, m)$, where $n$ is the length of sequence $A$ and $m$ is the length of sequence $B$, a constraint alignment is then computed (Fig. 1). The optimal path is built only through match blocks in the DP matrix using an affine gap penalty model[16]. Given a matrix $H$, where $H(i, j)$ is the best alignment score of ($a_1$,…, $a_i$, $b_1$,…, $b_j$) with $a_i$ aligned to a gap; a matrix $V$, where $V(i, j)$ is the best alignment score up to $a_i$ and $b_j$ with $b_j$ aligned to a gap, and a DP matrix $D$, where $D(i, j)$ is the best alignment score up to $a_i$ and $b_j$, the recurrence relation for the constrained DP is then calculated as:

$$H(i, j) = max(D(i-1, j) + o, H(i-1, j) + e)$$
$$V(i, j) = max(D(i, j-1) + o, V(i, j-1) + e)$$
$$D(i, j) = \begin{cases} max\,(D(i-1, j-1) + S(a_i, b_j), H(i, j), V(i, j)), & if\ (i, j)\ in\ block \\ -\infty, & otherwise \end{cases}$$

In the above equations, $o$ is the gap open penalty, $e$ is the gap extension penalty and $S(a_i, b_j)$ is the substitution score between nucleotides $a_i$ and $b_j$. The complexity of the alignment step is $O(nm)$.

## IV. Results

The program generated pairwise alignment is evaluated by comparison to a gold standard reference alignment at the nucleotide level. A pairwise alignment $A$ can be denoted as a set of nucleotide correspondence arranged in sequential order, so $A = \{a_i, b_i\}$, $i \in (1, n)$, where $n$ is the smaller length of the two sequences, and $a_i$ and $b_i$ are the matched nucleotides from two sequences. Let the reference alignment be $A$ and the testing alignment be $A' = \{a'_i, b'_i\}$, we define a correct correspondence as $a_i = a'_i$ and $b_i = b'_i$. The "=" here means two nucleotides have the same nucleotide index in the sequence. The alignment accuracy is the

percentage of correct correspondence over the number of all correspondence in the reference alignment *A*.

## A. Testing data

We created the pairwise sequence alignment testing data with structure information from three benchmarks, Bralibase 2.1[7], Rfam 10.1[17] and CRW Site[18]. Bralibase and Rfam are popular alignment benchmarks used by program evaluations and provide consensus secondary structures for various ncRNA families. The CRW Site well known for its high quality alignments provides individual RNA secondary structure in various formats which serve our purpose well.

We took three RNA families from Rfam: U2 spliceosomal RNA (RF00004), nuclear RNase P (RF00009) and Bacterial RNase P class A (RF00010); four RNA families from Bralibase data-set 1 and data-set 2: g2intron, U5 spliceosomal RNA, tRNA and 5S rRNA and two RNA families from CRW Site: 5S rRNA and 16S rRNA.

Besides the data-set2 from Bralibase which consists of pairwise sequence alignments, all the other tests are created by breaking the multiple sequence alignments into pairwise sequence alignments. The secondary structure of each sequence is either inferred from consensus structure provided in the original dataset or retrieved from the CRW Site. The pseudoknots are excluded from the structures.

A single test contains two RNA sequences with their corresponded RNA secondary structures annotated in dot bracket format[19]. Based on RNA sequence length, the datasets are grouped into small (< 200nt), middle (200-1000nt) and large (> 1000nt) RNA (Table 1). Tests in each RNA family are further divided into at least three subsets: 40, 60 and 80 by sequence identity. For example, subset 40 indicates the pairwise sequence identity of each test is between 40 and 60. Due to the page limit, the details of data sets and all the results are available upon request.

## B. Comparison with other programs

We implemented the algorithm in the tool R-PASS. For sequence alignment, we use the RIBOSUM scoring matrix [20] with gap open penalty of -8 and gap extension penalty of -1. R-PASS is written in Java, LARA, ExpaLoc and MAFFT are implemented in C. ExpaLoc is tested on the small RNA datasets, while the other three programs are tested on all datasets. Each test contains two sequences as input. Except for MAFFT-L-INS-I which only use the sequence information, the structure annotation is provided to all the other three programs. For ExpaLoc, ExpaRNA is first executed and the output constraints are then piped into LocARNA to obtain complete alignments. All programs are executed with default setting in a desktop with Intel processor at 3.16G Hz.

**1) Alignment accuracy—**Using the scoring method described above, we evaluated the alignment quality generated by all four programs with the corresponding reference alignments. As shown in Fig. 2, all four programs can achieve > 90% accuracy in small and large RNA datasets with sequence identity above 60% and in middle RNA datasets with sequence identity above 80%. In this zone, using structure information does not improve

alignment quality significantly. The additional structure information has a remarkable effect in the twilight zone, where the sequence identity is below 60%. While MAFFT performance drops sharply with decrease of sequence identity, R-PASS and LARA can maintain high alignment quality (Fig. 2).

ExpaLoc does not show superiority over MAFFT in the twilight zone in the small RNA datasets. One possible explanation is that as the sequence identity drops, the number of exact pattern matches, i.e., the number of structure constraints also decreases, thus the degree of freedom expands as LocARNA aligns the sequences, causing potential alignment errors.

In the twilight zone, R-PASS is comparable with LARA in terms of alignment quality. R-PASS outperforms all other programs in the tRNA dataset from Bralibase and RNase P datasets from Rfam (Fig. 2). In the other datasets except U5, the largest difference between R-PASS and LARA is less than 3%.

**2) Running time—**The total running time of each program on datasets of the same sized RNA group is used. ExpaLoc costs the most CPU time in the small RNA datasets (data not shown here).

As shown in Table 1, R-PASS is the fastest among all programs. The advantage over LARA is more obvious as the length of the RNA increases (Fig. 3). R-PASS is 10 times faster than LARA and 27 times faster than MAFFT in small RNA datasets; and is 97 times faster than LARA and 33 times faster than MAFFT for middle RNA datasets. It is 4 times faster than the sequence-based alignment program MAFFT and above 1,100 times faster than LARA in large RNA datasets while maintaining comparable alignment quality (Fig. 2). Therefore, our approach is more suitable for large-scale RNA sequences.

**3) R-PASS structure motif matching with LARA subroutine—**In R-PASS, individual sequence blocks generated after structure motif match can be fed into any alignment algorithm to produce a global alignment. Since LARA finds the structure motif correspondence at the base pair level, it performs better than R-PASS in some datasets in the twilight zone. Thus integrating LARA to align the stem regions may improve R-PASS performance in those datasets. We compute the structure match blocks by R-PASS and use LARA to align the blocks generated from Stem motifs and Gotoh algorithm to align the blocks of loop and free end motifs. Each local alignment is then linked in sequential order to form a global alignment.

We tested the integrated R-PASS and LARA in the small RNA datasets. This approach performs better than R-PASS and is comparable with LARA. In some cases, e.g. U5 (Fig. 4), the integrated approach improves the alignment accuracy upon LARA.

## V. CONCLUSION AND FUTURE WORK

We have developed a new workflow for pairwise alignment of RNA sequences with known structure information, named R-PASS. We utilize the RNA structure motif correspondence found in a bipartite graph framework to constrain the sequence alignment problem and perform the final alignment. The complexity of our algorithm is *O(nm)*, yet it can achieve

alignment quality comparable with the current state of the art programs. This is especially apparent in the twilight zone. R-PASS is also significantly faster than competing methods, often by orders of magnitude, which makes our method well suited for use in iterative algorithms and high-throughput RNA analysis.

We are currently working on various improvements to the R-PASS framework described in this paper. The alignment can be refined by matching the basic motifs in a Compound Stem/Hairpin motif. Improvements can also be extended to the nucleotide level where all the base pairings in each sequence could be used as constraints.

The similarity function between structural motifs we use is established by practical experience and the values are determined based on preliminary experiments. This function can be further enhanced by including additional information such as base pair similarity and nucleotide similarity. The additional information can improve the sensitivity and the selectivity of the matching algorithm.

Our results show that incorporating structure information significantly improves alignment accuracy upon sequence-based alignment methods, especially for less conserved sequences. While our current algorithm focuses on aligning known structures, it can be adapted to align unknown structures by structure prediction using RNA folding algorithms or generating all potential structure motifs and find correct matches by an advanced similarity function.

The fast performance of our alignment program also makes it promising to compute a multiple sequence alignment over a large set of sequences. While R-PASS focuses on pairwise alignment, the result can be extended to produce multiple sequence alignments using a guide tree or a progressive approach. Our method can also target template-based alignment problems, where new sequences are added into an existing alignment by matching the new sequence with the consensus sequence of the alignment.

## Acknowledgments

This research is supported by the National Institutes of Health grants, GM085337 and GM067317.

## References

1. Gutell RR, Weiser B, Woese CR, Noller HF. Comparative anatomy of 16-S-like ribosomal RNA. Prog Nucleic Acid Res Mol Biol. 1985; 32:155–216. [PubMed: 3911275]
2. Gutell RR, Larsen N, Woese CR. Lessons from an evolving rRNA: 16S and 23S rRNA structures from a comparative perspective. Microbiol Rev. Mar; 1994 58(1):10–26. [PubMed: 8177168]
3. Phillips A, Janies D, Wheeler W. Multiple sequence alignment in phylogenetic analysis. Mol Phylogenet Evol. Sep; 2000 16(3):317–330. [PubMed: 10991785]
4. Wu JC, Gardner DP, Ozer S, Gutell RR, Ren P. Correlation of RNA secondary structure statistics with thermodynamic stability and applications to folding. J Mol Biol. Aug 28; 2009 391(4):769–783. [PubMed: 19540243]
5. Washietl S, Hofacker IL, Lukasser M, Huttenhofer A, Stadler PF. Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. Nature Biotechnology. Nov; 2005 23(11):1383–1390.
6. Katoh K, Kuma K, Miyata T, Toh H. Improvement in the accuracy of multiple sequence alignment program MAFFT. Genome Inform. 2005; 16(1):22–33. [PubMed: 16362903]

7. Gardner PP, Wilm A, Washietl S. A benchmark of multiple sequence alignment programs upon structural RNAs. Nucleic Acids Res. 2005; 33(8):2433–2439. [PubMed: 15860779]
8. Bauer M, Klau GW, Reinert K. Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. BMC Bioinformatics. 2007; 8:271. [PubMed: 17662141]
9. Heyne S, Will S, Beckstette M, Backofen R. Lightweight comparison of RNAs based on exact sequence-structure matches. Bioinformatics. Aug 15; 2009 25(16):2095–2102. [PubMed: 19189979]
10. Sankoff D. Simultaneous solution of the RNA folding, alignment, and proto-sequence problems. SIAM J Appl Mat. 1985; 45:810–825.
11. Will S, Reiche K, Hofacker IL, Stadler PF, Backofen R. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. PLoS Comput Biol. Apr 13.2007 3(4):e65. [PubMed: 17432929]
12. Wilm A, Mainz I, Steger G. An enhanced RNA alignment benchmark for sequence alignment programs. Algorithms Mol Biol. 2006; 1:19. [PubMed: 17062125]
13. Wang Y, Makedon F, Ford J, Huang H. A bipartite graph matching framework for finding correspondences between structural elements in two proteins. Conf Proc IEEE Eng Med Biol Soc. 2004; 4:2972–2975. [PubMed: 17270902]
14. G D, Shapley LS. College Admissions and the Stability of Marriage. American Mathematical Monthly. 1962; 69:9–14.
15. Dijkstra EW. A note on two problems in connexion with graphs. Numerische Mathematik. 1959; 1:269–271.
16. Gotoh O. An improved algorithm for matching biological sequences. J Mol Biol. Dec 15; 1982 162(3):705–708. [PubMed: 7166760]
17. Gardner PP, Daub J, Tate J, Moore BL, Osuch IH, Griffiths-Jones S, Finn RD, Nawrocki EP, Kolbe DL, Eddy SR, Bateman A. Rfam: Wikipedia, clans and the “decimal” release. Nucleic Acids Res. Jan; 2011 39(Database issue):D141–145. [PubMed: 21062808]
18. Cannone JJ, Subramanian S, Schnare MN, Collett JR, D’Souza LM, Du Y, Feng B, Lin N, Madabusi LV, Muller KM, Pande N, Shang Z, Yu N, Gutell RR. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. BMC Bioinformatics. 2002; 3:2. [PubMed: 11869452]
19. Gruber AR, Lorenz R, Bernhart SH, Neubock R, Hofacker IL. The Vienna RNA websuite. Nucleic Acids Res. Jul 1; 2008 36(Web Server issue):W70–74. [PubMed: 18424795]
20. Klein RJ, Eddy SR. RSEARCH: finding homologs of single structured RNA sequences. BMC Bioinformatics. Sep 22.2003 4:44. [PubMed: 14499004]

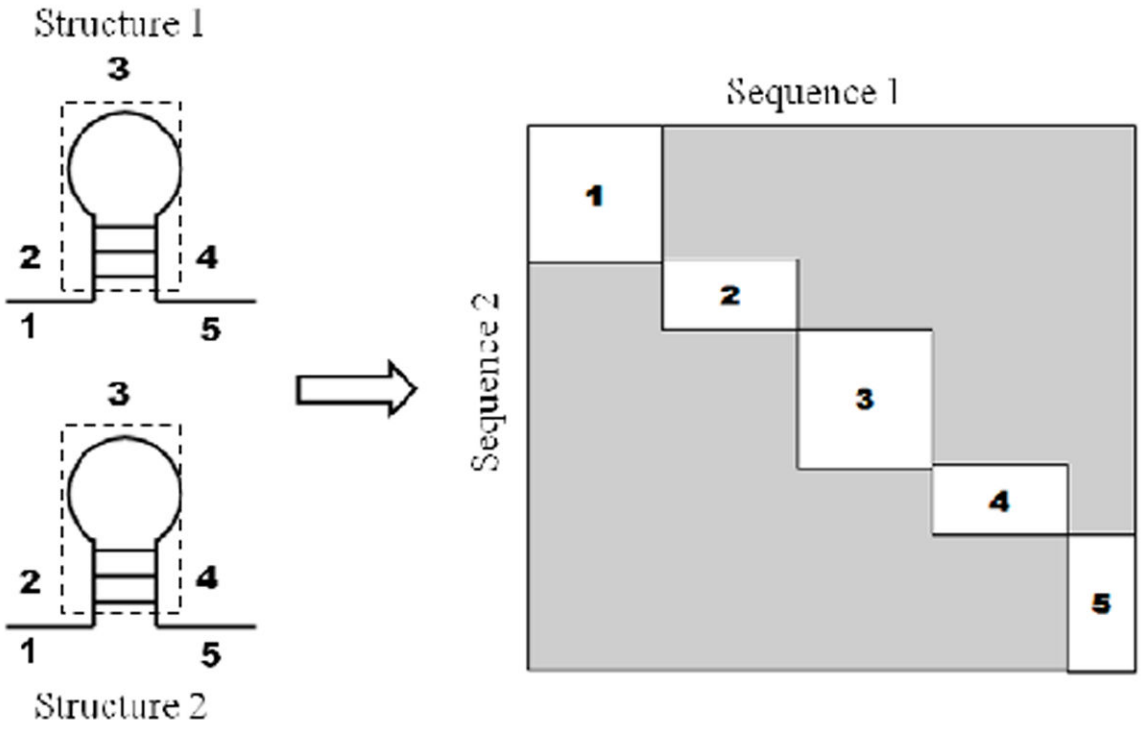


**Figure 1.**
Convert structure alignment to sequence alignment.

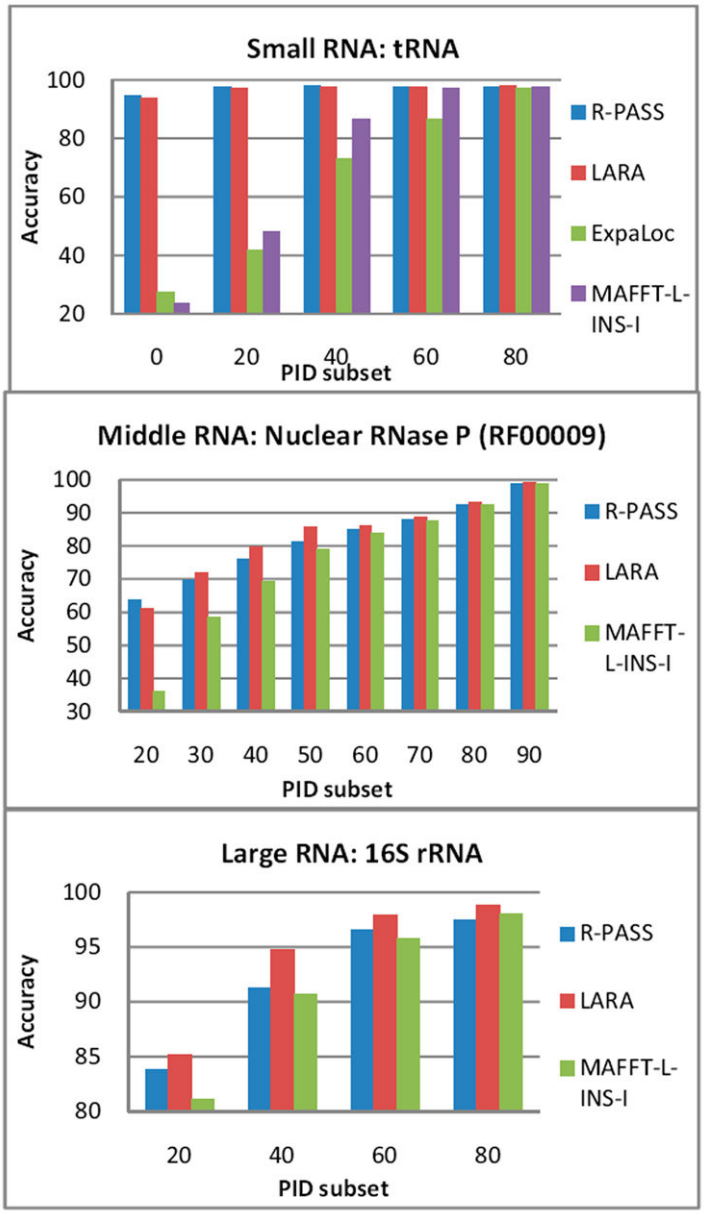


**Figure 2.**
Accuracy comparisons in small, middle and large RNA datasets.

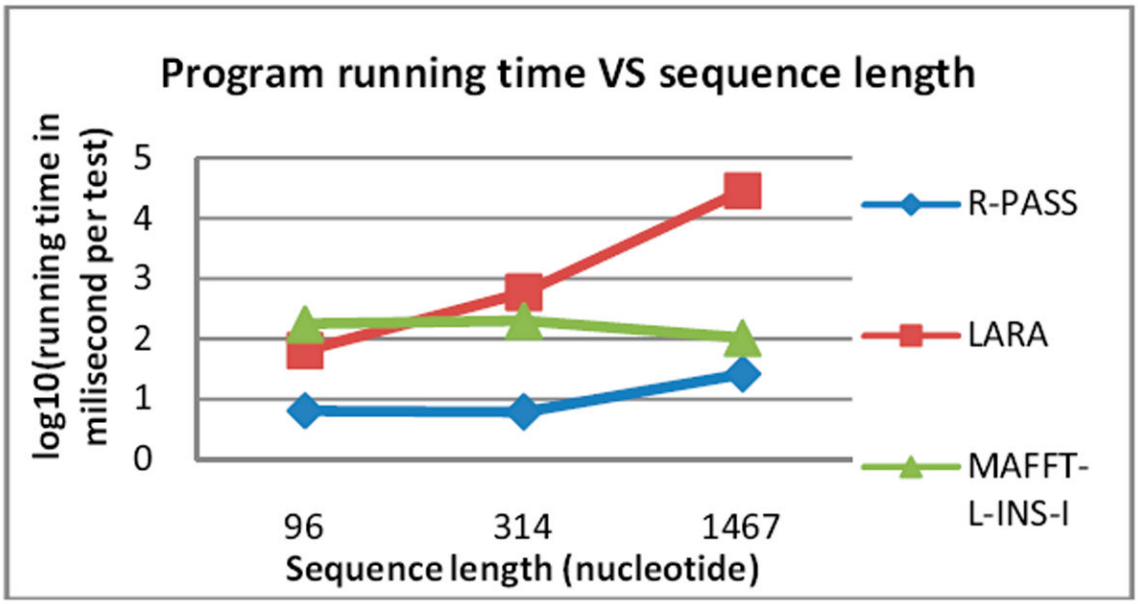


**Figure 3.**
Program running time versus sequence length.

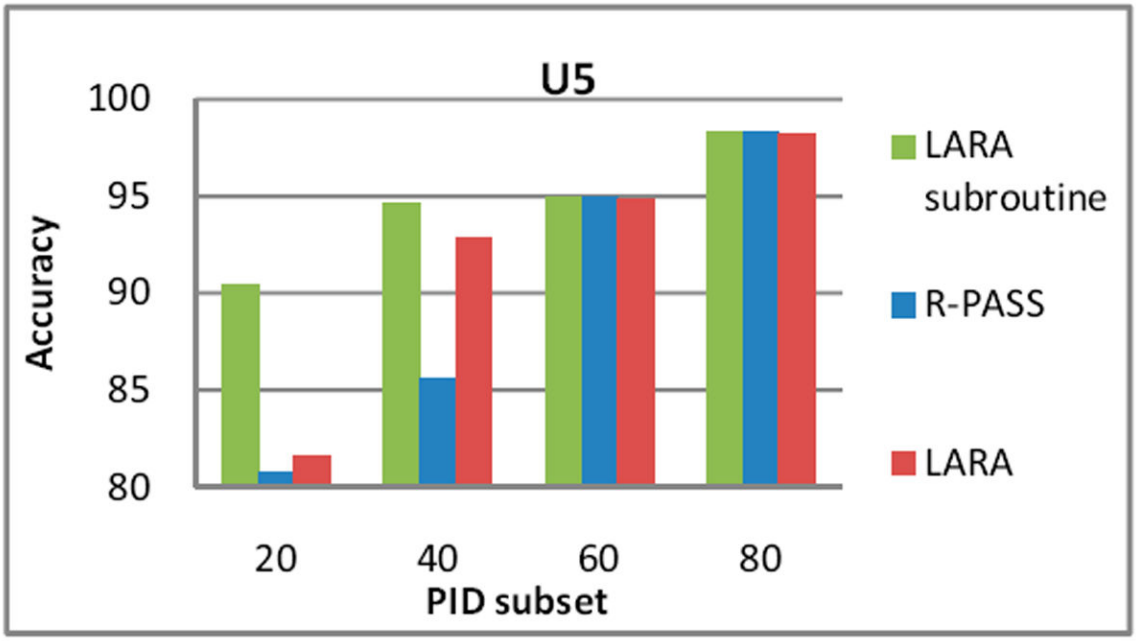


**Figure 4.**
Accuracy comparison on U5.

**TABLE I**

PROGRAM RUNNING TIME IN SECONDS.

| Program | Small | Middle | Large |
|---|---|---|---|
| R-PASS | 17.8 | 17.6 | 27 |
| LARA | 178.24 (10)[a] | 1701 (97) | 30000 (1111) |
| MAFFT-L-NS-I | 491.34 (27) | 573 (33) | 108 (4) |
| # tests | 2806 | 2893 | 1046 |
| Avg. length | 96 | 314 | 1467 |

[a]The number of times that R-PASS is faster than the program