Efficient RNA pairwise structure comparison by SETTER method

David Hoksza\textsuperscript{1,2,*} and Daniel Svozil\textsuperscript{2,*}

\textsuperscript{1}SIRET Research Group, Department of Software Engineering, FMP, Charles University in Prague, 11800 Czech Republic and \textsuperscript{2}Laboratory of Informatics and Chemistry, Institute of Chemical Technology Prague, 16628 Czech Republic

\textbf{ABSTRACT}

Motivation: Understanding the architecture and function of RNA molecules requires methods for comparing and analyzing their 3D structures. Although a structural alignment of short RNAs is achievable in a reasonable amount of time, large structures represent much bigger challenge. However, the growth of the number of large RNAs deposited in the PDB database calls for the development of fast and accurate methods for analyzing their structures, as well as for rapid similarity searches in databases.

\textbf{Results:} In this article a novel algorithm for an RNA structural comparison \textit{SETTER} (SE\textit{condary} Structure-based \textit{TERtiary Structure Similarity Algorithm) is introduced. \textit{SETTER} uses a pairwise comparison method based on 3D similarity of the so-called generalized secondary structure units. For each pair of structures, \textit{SETTER} produces a distance score and an indication of its statistical significance. \textit{SETTER} can be used both for the structural alignments of structures that are already known to be homologous, as well as for 3D structure similarity searches and functional annotation. The algorithm presented is both accurate and fast and does not impose limits on the size of aligned RNA structures.

\textbf{Availability:} The \textit{SETTER} program, as well as all datasets, is freely available from \url{http://siret.cz/hoksza/projects/setter/}.

\textbf{Contact:} hoksza@ksi.mff.cuni.cz, or svozil@vscht.cz

\textbf{ Supplementary information:} Supplementary data are available at \textit{Bioinformatics} online.

Received on November 18, 2011; revised on May 15, 2012; accepted on May 16, 2012

\section{INTRODUCTION}

In addition to its role in the transfer of biological information, the evidence shows that RNA molecules also play key roles in a variety of cellular processes\cite{Mattick2005}. RNA shows, among others, an enzymatic activity in ribosomes\cite{Kim1974}, it plays part in the transcription regulation\cite{Bartel2004}, and it is among others, an enzymatic activity in ribozymes\cite{Scott2007}, of cellular processes\cite{Mattick2006}. RNA shows, evidence shows that RNA molecules also play key roles in a variety of cellular processes, such as continuous interhelical base stacking, RNA domain structure and helical packing.

The function of an RNA molecule is largely determined by the 3D structure that is typically more evolutionarily conserved than its sequence\cite{Chursov2012}. Thus, methods for the comparative RNA function annotation based on structural similarity usually yield much better results than sequence-based approaches. Although detecting optimal structural similarity between two biomolecules in 3D has been shown to be NP-hard \cite{Kokolov2004}, the development of automatic tools capable of an efficient and accurate RNA structural alignment has become an important part of structural bioinformatics. The study of RNA tertiary and quaternary structures must be facilitated by the software that is able to work both with small and large RNA molecules. To be computationally tractable, currently available software tools for comparing two RNA 3D structures, such as ARTS\cite{Dror2008}, DIAL\cite{Dror2004, Ferré2005}, PARTS\cite{Chang2008, Chang2010}, SARSA\cite{Dror2005, Dror2006}, and R3D Align\cite{Rahrig2010}, are therefore based on heuristic approaches. ARTS\cite{Dror2004}, DIAL\cite{Dror2004, Ferré2005} and R3D Align\cite{Rahrig2010} detects a maximum common substructure between two RNA 3D structures using backbone phosphate atoms. Based on 3D similarity between 1333 solved RNA structures assessed by the ARTS algorithm, a database of hierarchically classified structures DARTS was subsequently developed\cite{Abraham2008}. ARTS is not practical for comparison of large RNA molecules due to its cubic time complexity. To overcome this problem, the DIAL server using a dynamic programming algorithm and running in a quadratic time was developed\cite{Ferré2007}. The DIAL alignment algorithm is based on torsion and/or pseudorotorion similarity sequence similarity and base pairing similarity, and it provides access to global, local and semi-global structural alignments. An improvement in the speed over the DIAL algorithm was later brought by PARTS\cite{Chang2010}, an algorithm based on the so-called structural alphabet (SA). SA is an emerging concept in the structural biology of proteins. A protein structure is represented as a limited series of ‘letters’ each assigned to a well-characterized conformation\cite{deBrev}.
Three important elements are recognized in the GSSU: a loop, a neck and a to form a global alignment by using a maximum clique algorithm. Local alignments are then merged by nucleotide superpositions that effectively accommodate the flexibility of RNA molecules. Local alignments are then merged to form a global alignment by using a maximum clique algorithm on a specially defined ‘local alignment’ graph.

In this article, a novel pairwise RNA comparison method (SEcondary xTructure-based TEriary Structure Similarity Algorithm) SETTER is proposed. The method divides the whole RNA structure into non-overlapping generalized secondary structure units (GSSUs). The structural alignment is then obtained by using a distance measure based on RMSD transformations between all unit spheres of consecutive unit spheres are computed and used as scoring matrix for the dynamic programming-based global alignment. Highly accurate alignments of homologous molecules are produced by the R3D Align algorithm \( \text{Rahrig et al.} \) which is based on local nucleotide by nucleotide superpositions that effectively accommodate the flexibility of RNA molecules. Local alignments are then merged to form a global alignment by using a maximum clique algorithm on a specially defined ‘local alignment’ graph.

In the case of ambiguity, a maximum length is assigned to the substring such that each nucleotide \( nt_i \) is paired with \( nt_j \). However, this does not allow an unambiguous distinction between GSSUs. To identify all GSSUs, the process iteratively applies two following steps. The RNA structure is processed in a sequence order and in the first step, nucleotides are stored on a stack. This process stops by encountering a nucleotide \( nt_i \) which is already in the stack. Then, in the second step, a new GSSU \( G \) starts to be formed from the pair \( \{ nt_i, nt_j \} \) (i.e. the neck) and from all nucleotides found between \( nt_i \) and \( nt_j \) (i.e. the loop). These residues are then removed from the stack. Finally, the stem is formed from all residues encountered either before the residue WC bonded with the residue that was pushed on the stack before the previous GSSU was generated. By repeating these two steps, the algorithm iteratively searches for GSSUs, and it stops when the end of the sequence is reached. All residues remaining on the stack (if any) then form the last GSSU. Note that even a structure without a single WC pair has a GSSU which is identical with the structure itself. A detailed description of the process of generating GSSUs from Fig. 1 as well as the pseudocode algorithm are given in Sections 2 and 3 in Supplementary Information.

2 METHODS

2.1 GSSU identification

Three important elements are recognized in the GSSU: a loop, a neck and a stem (Fig. 1). A formal description of the GSSU is given by the following definition.

Definition 1. Let \( R \) be an RNA structure with a nucleotide sequence \( \{ nt_i \} \) and let \( WC \subseteq R \) denote its subset participating in a Watson–Crick (WC) base pair. By a GSSU \( G \), we understand a pair of substrings of \( R \), \( \{ nt_{1,\max} \} \) and \( \{ nt_{1,\min} \} \) such that each nucleotide \( nt_i \) is paired with \( nt_j \) of maximum length such that each nucleotide \( nt_i \) is paired with \( nt_j \) where \( j_1 \leq j_2 \leq \cdots \leq j_n \).

In case of ambiguity, a maximum length is assigned to the substring occurring earlier in the sequence. Let \( \max \) and \( \min \) be the highest/lowest indices of the WC paired bases in \( G \). We define a loop as \( L = \{ nt_{1,\max} \} \) and a neck as the pair \( (nt_{1,\min}, nt_{1,\max}) \).

Nucleotides are represented by their P atoms. WC hydrogen bonds are identified using 3DNA \( \text{et al.} \) and OGB \( \text{et al.} \). Non-WC pairs are not used because they often mediate RNA tertiary contacts, the presence of which does not allow an unambiguous distinction between GSSUs.

To identify all GSSUs, the process iteratively applies two following steps. The RNA structure is processed in a sequence order and in the first step, nucleotides are stored on a stack. This process stops by encountering a nucleotide \( nt_i \) WC bonded with a nucleotide \( nt_j \) already in the stack. Then, in the second step, a new GSSU \( G \) starts to be formed from the pair \( \{ nt_{1,\min}, nt_{1,\max} \} \) (i.e. the neck) and from all nucleotides found between \( nt_i \) and \( nt_j \) (i.e. the loop). These residues are then removed from the stack. Finally, the stem is formed from all residues encountered either before the residue WC bonded with the residue that was pushed on the stack before the previous GSSU was generated. By repeating these two steps, the algorithm iteratively searches for GSSUs, and it stops when the end of the sequence is reached. All residues remaining on the stack (if any) then form the last GSSU. Note that even a structure without a single WC pair has a GSSU which is identical with the structure itself. A detailed description of the process of generating GSSUs from Fig. 1 as well as the pseudocode algorithm are given in Sections 2 and 3 in Supplementary Information.

2.2 Comparing two GSSUs

GSSU pairwise comparison lies in the heart of the method. Each GSSU is represented by an ordered set of 3D coordinates of P atoms annotated with bonding and nucleotide-atom-type information. A common way to assess similarity between two sets of points is to define a pairing between them. The sets are then superposed by finding translation and rotation of one of them over the other minimizing the mutual distances of the respective paired points. Usually, the RMSD is used as the distance measure and two structures can be superposed given a pairing (alignment) in polynomial time \( O(n^2) \). However, finding the optimal alignment is an NP-hard problem \( \text{Khachiyan and Liptak} \). The optimal solution can be found by exhaustive search, which is computationally not feasible. This problem can be resolved by identifying suboptimal alignments that will likely participate in the optimal alignment. This is the principle idea behind SETTER’s structure comparison process. SETTER generates a set of short alignments, the quality of which is evaluated by the Kabach \( \text{et al.} \) RMSD algorithm. Working with relatively short alignments allows to superpose even the largest RNA structures in a reasonable amount of time.

To superpose two GSSUs means to match their loops which implies matching their necks (Fig. 1). To define the superposition in 3D space unambiguously at least three pairs of points are needed. A set of these points is called a triplet and the alignment is formed by matching triplets between two given structures. SETTER aligns necks first, and then the final pair in the triplet is identified by aligning each possible pair of loops’ nucleotides. For example, if two GSSUs with loops consisting of \( n \) and \( m \) nucleotides are to be aligned, \( n \times m \) alignments are generated (Fig. 1).
The whole process is formalized by equation (1).

\[ G_{ij} \] stands for two GSSUs to be compared, \( G \) stands for the 11th nucleotide in the sequence of \( G \) and \( |G| \) for its length. \( N_{i}(x, G) \) is the Euclidean distance from the nucleotide \( x \) to its nearest neighbor in \( G \). If \( x \) and its nearest neighbor share the same nucleotide type (the function \( r(x) \) in the formula), the distance is modified by a factor \( \zeta \). It takes values from the interval \( 0 < \zeta \leq 1 \), the lower the value, the more matching identical nucleotides are rewarded. \( \delta \) computes the raw distance \( T \) is the set of transpositions resulting from the candidate triplet alignments and \( v(G, r) \) transposes GSSU \( G \) using the transposition \( r \). The \( S \)-distance is then normalized by the function \( y \) counting a number of nearest neighbors within the distance \( \epsilon \) after the optimal transposition \( \alpha_{opt} \).

Since hydrogen bonds are identified using simple geometric criteria, their formation is performed for each of the candidate alignments decreasing the \( \lambda \)-score. Each pair of GSSUs just being aligned, and such a translation is already implicitly translation is not included explicitly. The translation is limited only to the \( \lambda \), \( \delta \), \( \phi \), \( \epsilon \) and \( \chi \) being the best result so far, the comparison computation process is implemented (Fig. 3):

\[ \text{if } \{ \frac{G_1}{G_2} \} \text{ are iteratively added to the } \mathcal{T}-\text{distance.} \]

For each structure \( \mathcal{T}_{A} \) and \( \mathcal{T}_{B} \), each GSSU from \( \mathcal{T}_{A} \) is compared with each GSSU from \( \mathcal{T}_{B} \), but only top \( \kappa \) pairs with the minimum distance is processed further. For each of the \( \kappa \) selected pairs \( \langle G_{i}^{A}, G_{j}^{B} \rangle \) the value of \( \mathcal{T} \) is set to \( S(G_{i}^{A}, G_{j}^{B}) \). In the second step, \( \mathcal{T} \)-distances of the neighboring GSSU pairs are iteratively added to the \( \mathcal{T} \)-distance. For the GSSU pair \( \langle G_{i+1}^{A}, G_{j+1}^{B} \rangle \) the value of \( S(G_{i+1}^{A}, G_{j+1}^{B}) \) and the penalty for the rotation needed to transform the structures from the state corresponding to \( S(G_{i}^{A}, G_{j}^{B}) \) to the state corresponding to \( S(G_{i}^{A}, G_{j+1}^{B}) \) are added to \( \mathcal{T} \). This process goes from \( i+1, j+1 \) until either \( i+1 > \lambda \) or \( j+1 \) reaches the number of GSSUs in \( \mathcal{T}_{A} \) or \( \mathcal{T}_{B} \). Similarly, the other ends of the structures need to be aligned and so the process is repeated for \( \{ i-1, j-1 \} \). However, the case when GSSUs in the RNA structure are oriented in the opposite direction must also be considered, and another \( \kappa \) alignments must be performed aligning \( i-1 \) residues with \( j+1 \) residues (not shown in Fig. 3).

The rotation between two GSSUs imposes a penalty to the \( \mathcal{T} \). This penalty is calculated as a distance between the rotation matrices describing two consequent GSSU superpositions (Fig. 3). However, the penalty for translation is not included explicitly. The translation is limited only to the pair of GSSUs just being aligned, and such a translation is already implicitly present in the \( \mathcal{T} \)-distance (see Section 2.3 and Fig. 3).

Currently, there is no provision for a situation in which one structure is missing a GSSU that is present in the other structure. This potential limitation may have a possible undesirable effect on the alignment; however, it cannot be improved without an increase in computational demands.

2.4 Early termination

The nearest neighbor search, which is a part of the \( \mathcal{T} \)-distance computation, has \( O(n^3) \) time complexity with respect to the GSSU’s length \( n \). In addition, the search is performed for each of the candidate alignments decreasing the efficiency of SETTER. To increase the algorithm’s speed, a simple early termination condition is thus implemented. Alignments that are not likely to be the part of the optimal superposition are identified, and for these, the nearest neighbor search is skipped. Such alignments will very likely have the triplet \( \mathcal{T} \)-distance higher than the lowest \( \mathcal{T} \)-distance obtained up to that time. Specifically, triplet-based \( \mathcal{T} \)-distance will probably be lower than ‘real’ \( \mathcal{T} \)-distance. If the triplet \( \mathcal{T}_{A} \subset \mathcal{T}_{B} \) is aligned with the triplet \( \mathcal{T}_{A} \subset \mathcal{T}_{B} \) with \( S(G_{i}^{A}, G_{j}^{B}) = \gamma \) being the best result so far, the comparison computation can be terminated if \( S(T_{A}^{\lambda} \cdot T_{B}^{\gamma}) = l \cdot \lambda > \chi \). Since the early termination is a heuristic, \( (T_{A}^{\lambda} \cdot T_{B}^{\gamma}) \cdot S(G_{i}^{A}, G_{j}^{B}) \) does not have to be valid, the early termination condition is strengthened by introducing the parameter \( l \geq 1 \). By varying the \( l \) parameter, the trade-off between accuracy and speed can be set. The higher the \( l \), the less often early termination occurs and the more accurate and slower the algorithm is. The effect of the \( \lambda \) parameter

Fig. 2. The alignment of the GSSU from the 16S rRNA of the transfer-messenger RNA (PDB code 1P6V) with the GSSU from the glutamine tRNA (PDB code 1EXD). The final structural alignment is defined by three nucleotide pairs forming a triplet (Lines 1–3). To find the optimal superposition for the given neck pairs (Lines 1 and 2), the position of the middle pair is varied (Line 3).

For each alignment, the rotation matrix and the translation vector defining optimal superposition of two triplets are calculated. Although these are optimal for the given triplet pair only, they are used to superpose whole GSSUs. This possible inaccuracy is the trade-off for an efficiency.

After the superposition, for each mismatch in GSSU B is found, and its distance is added to the distance of two GSSUs referred to as \( \mathcal{T} \)-distance. Finally, the \( \mathcal{T} \)-distance is normalized. The whole process is formalized by equation (1).

\[ \begin{align*}
\text{if } x=r(G_{i}) & \quad \text{then } \min_{x \neq y} |G_{i}(x) - G_{j}(y)| = 0; \\
\delta(G_{i}^{A}, G_{j}^{B}) & = \sum_{x=1}^{n} \left( \sum_{y=1}^{n} \min_{x \neq y} |G_{i}(x) - G_{j}(y)| \right) \times \zeta; \\
\delta(G_{i}^{A}, G_{j}^{B}) & = \sum_{x=1}^{n} \left( \sum_{y=1}^{n} \min_{x \neq y} |G_{i}(x) - G_{j}(y)| \right) \times \zeta; \\
\delta(G_{i}^{A}, G_{j}^{B}) & = \sum_{x=1}^{n} \left( \sum_{y=1}^{n} \min_{x \neq y} |G_{i}(x) - G_{j}(y)| \right) \times \zeta; \\
\delta(G_{i}^{A}, G_{j}^{B}) & = \sum_{x=1}^{n} \left( \sum_{y=1}^{n} \min_{x \neq y} |G_{i}(x) - G_{j}(y)| \right) \times \zeta.
\end{align*} \]
A schematic representation of two RNA structures

The assessment of the quality of structural alignments is not an easy task

individual GSSUs;

on the quality of the functional annotation is demonstrated in Section 4 of

rotation angles

the rotation angle needed to get from the state \( G_A \) to \( G_B \) (first down arrow in the figure) is thus known. The current state is changed to \( G_B \), and the process is repeated for the pair \( G_B, G_A \) (second down arrow in the figure).

Similarly, the algorithm must also process in the opposite direction from the position of \( G_A \) and \( G_B \) (up arrow in the figure).

The rotation angles \( \pi \) from the previous step are used in the penalty function \( X \), which represents a weight function for the GSSU distances. To get the final \( \Sigma \)-distance, the sum of weighted GSSU \( \Sigma \)-distances is normalized by the ratio of non-aligned GSSUs over the maximum number of non-aligned GSSUs. The parameter \( \beta \) was empirically set to 6

on the quality of the functional annotation is demonstrated in Section 4 of Supplementary Information.

For each alignment, SETTER outputs the list of residues forming individual GSSUs. \( \Sigma \)-distance characterizing the overall quality of the alignment. \( P \)-value quantifying the statistical significance of the alignment; list of aligned GSSUs, rotation and translation matrices; 3D coordinates of each residue after the superposition; triplet pair of the best scoring GSSU pair and list of residues with their respective nearest neighbors and the corresponding distances.

2.5 Structural alignment accuracy

The assessment of the quality of structural alignments is not an easy task because it is not possible to define a perfect 3D-to-3D alignment is [Brown et al. 2004]. The commonly used measures such as e.g. the RMSD requires the knowledge of which residues are aligned against which ones. However, because SETTER is not based on a sequence alignment algorithm such information is missing. Therefore, the list of aligned residues was generated using a simple geometric approach. Two residues \( A \) and \( B \) are considered to be aligned if \( A \) is the closest residue to \( B \) and, at the same time, \( B \) is the closest residue to \( A \). Such a definition is, in our opinion, suitable for an evaluation of the quality of the superposition and can be used for an approximate comparison with other alignment-based methods.

In the present work, the quality of the structural alignments was assessed by using the following measures: the RMSD, the percentage of structural identity (PSI), the percentage of sequence identity (PID), Capriotti and Marti-Renom, 2004, 2009 and the number of nucleotides aligned and the number of exact base matches [Rahman et al. 2010]. RMSD captures the general 3D shape of RNA, but it can be misleading as the errors are spread over the whole molecule. PSI is defined as a percentage of superimposed residues within 4.0 Å with respect to the length of the shorter of the two structures. PID is the percentage of aligned nucleotides of the same type with respect to the length of the shorter of the two structures. The number of aligned and number of exact base matches give similar information as PSI and PID. We note that these measures do not account for the specificity of RNA base pairing and base-stacking interactions. Therefore, some new metrics particularly suitable to RNA structure comparison have been developed by [Parisien et al. 2004]. However, these are not used in the present study as they would not allow to compare SETTER results with other approaches.

2.6 Statistical significance of the structural alignment

The quality of the structural alignment can be assessed by means of statistical hypothesis testing. The key idea is to create a set of randomly generated structures, to align them and to fit the distribution of their \( \Sigma \)-distances. For the given \( \Sigma \)-distance its \( P \)-value can then be calculated. The alignment is a good one if its \( \Sigma \)-distance is good compared with the distribution of \( \Sigma \)-distances. This is reflected by its low \( P \)-value; the smaller the \( P \)-value, the more statistically significant the \( \Sigma \)-distance is. To show how well data follow the fitted distribution a visual inspection of quantile–quantile plots (QQ-plots) can be used, or the fit can be tested by two-sample Kolmogorov–Smirnov test.

\( \Sigma \)-distance follows the log-normal distribution (see Section 5 in Supplementary Information), the probability density function \( \rho(x) \) of which is given as

\[
\rho(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]

where parameters \( \mu \) and \( \sigma \) are the mean and standard deviation, respectively, of the variable’s natural logarithm that is by definition, normally distributed. On a non-logarithmized scale, \( \mu \) is called a location parameter and \( \sigma \) a scale parameter. These parameters must be determined, and once they are known, they can be used to derive the statistical significance of the particular alignment given as its \( P \)-value. \( P \)-value corresponds to the probability \( P(X \leq x) \) that the variable \( X \) takes a value lower or equal to \( x \),

where \( \text{erf}(x) \) is the error function defined as

\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt.
\]

For the determination of \( \mu \) and \( \sigma \) parameters, a set of reasonably unrelated structures was prepared. Such a set should cover the whole range of alignments starting from the exceptionally good ones going up to the very bad ones. The unrelatedness of the structures was based on their sequence similarity. The used threshold of 80% sequence similarity guarantees the uniform coverage of the alignments in terms of their quality. Because the \( \mu \) and \( \sigma \) parameters depend on the length of the shorter structure \( N \) in the alignment they must be determined for different lengths separately. For each length, a dataset containing 50,000 structure pairs was generated by randomly cutting the regions of the given length from structures longer than \( N \). The datasets of lengths 5, 10, 15, 20, … , 300 residues were prepared. The \( \Sigma \)-distance was determined for each alignment in the given dataset and the parameters \( \mu \) and \( \sigma \) of the log-normal distribution were found by a maximum-likelihood fitting. All statistical calculations were performed using the R system version 2.13.1 with the package MASS (version 7.3-14).

2.7 Functional annotation accuracy

The quality of the functional annotation was assessed by SETTER’s ability to correctly assign the SCOR functional classification to the query RNA
The area under the ROC curve (AUC), a threshold independent measure, is calculated as a percentage of correctly classified structures. To obtain the AUC, the alignments of all pairs of RNA structures were sorted by their -values. We also notice that it is difficult to obtain high values of both ACC and AUC simultaneously (see Section 3.2.1). The ability of SETTER to produce good structural alignments is demonstrated on the visualizations of the superpositions of several 23S rRNA, 16S rRNA, 5S rRNA, tRNA and other RNA structures (see Section 1 in the Supplementary Information).

3 RESULTS AND DISCUSSION

3.1 Assessment of the structural alignment quality

SETTER structural alignments were compared with SARA by calculating PSI values for the all-to-all comparison using the FSCOR dataset. The results are summarized in Figure 4 showing that SETTER yields less alignments with very low PSI (up to 20%), and SARA returns slightly more alignments with PSI >90%. In terms of remaining PSI levels, both methods perform similarly.

In addition, SETTER was compared with R3D Align, ARTS, SARA and DIAL by calculating various measures reflecting the quality of the alignment of two 16S rRNA structures and of the alignment of the sarcin/ricin domain from 28S rRNA with the central part of the 5S rRNA. The results summarized in Section 7 in Supplementary Information also demonstrate that SETTER produces alignments of the quality comparable with other automated approaches considering its approximate nature in obtaining the list of aligned residues (see Section 3.2). The ability of SETTER to produce good structural alignments is demonstrated on the visualizations of the superpositions of several 23S rRNA, 16S rRNA, 5S rRNA, tRNA and other RNA structures (see Section 1 in the Supplementary Information).

3.2 Assessment of the functional annotation quality

3.2.1 Statistical parameters evaluation

QQ-plots show that data follow the fitted log-normal distribution very closely except for the region of high -distances (see Section 5 in Supplementary Information). However, poor fitting in this region will not seriously influence the results of database searching, as we are generally not interested in highly dissimilar structures. The quality of the fit was further confirmed by Kolmogorov–Smirnov test that provided . The relations (2) and (3) provide a simple way to calculate and parameters for any sequence length .
Table 1. ACC and AUC comparison of SETTER, iPARTS and SARA on the FSCOR and T/R-FSCOR datasets

<table>
<thead>
<tr>
<th></th>
<th>FSCOR</th>
<th>T/R-FSCOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>ACC</td>
</tr>
<tr>
<td>iPARTS</td>
<td>72/92</td>
<td>?</td>
</tr>
<tr>
<td>STR(_{p=0.0})</td>
<td>82/91</td>
<td>61.8/72.8</td>
</tr>
<tr>
<td>SARA</td>
<td>61/63</td>
<td>81.4/95.3</td>
</tr>
<tr>
<td>STR(_{p=0.01})</td>
<td>71/87</td>
<td>80.5/95.1</td>
</tr>
</tbody>
</table>

The values are given in % and are reported for exact/similar classification. iPARTS should be compared with SETTER with the P-value threshold of 1.0 (i.e., no filtering applied), and SARA should be compared with SETTER with the P-value threshold of 0.01. For iPARTS, ACC was not reported and necessary tests cannot be performed using the iPARTS web interface. ROC curves from which AUC value were calculated are shown in Section 8 of the Supplementary Information.

3.2.2 Effectiveness comparison

The capabilities of SETTER for a functional annotation of new RNA structures were compared with SARA and iPARTS approaches using the published AUC (SARA, iPARTS) and ACC (SARA only) values on the FSCOR and T/R-FSCOR datasets. Following settings were used in SETTER: \( \xi = 0.2, \beta = 2, \epsilon = 6, \lambda = 3 \) and \( \lambda = 1 \) (see Section 3.3 for details).

The results are summarized in Table 1. The percentage of classified structures for the given P-value threshold is called a coverage and two sets of results with different coverage are presented for SETTER. STR\(_{p=0.0}\) corresponds to the classification where the structures are sorted according to their P-values but no threshold is applied (coverage equals to 100%). At this coverage, SETTER is compared with iPARTS which does not use any filtering. Results in Table 1 show that SETTER outperforms iPARTS in ACC for exact classification both in FSCOR and T/R-FSCOR datasets and performs comparably for similar classification.

SARA was evaluated at a coverage of 58.7%. To achieve this coverage in SETTER, the P-value threshold of 0.013 was used. At this coverage, SETTER performs better than SARA in terms of ACC both for the FSCOR and T/R-FSCOR datasets (Table 1). In addition, SETTER can also be compared with SARA in terms of ACC. SETTER’s ACC is comparable to that of SARA in the similar classification both for the FSCOR and T/R-FSCOR datasets, and in the exact classification for the FSCOR dataset. However, it is much higher (by 13.7%) for the exact classification for the T/R-FSCOR dataset.

Thus, it can be concluded that SETTER performs better than iPARTS and SARA in terms of ACC and is comparable with SARA in terms of AUC.

3.3 Speed comparison

SETTER’s nearest neighbor identification procedure scales as \( O(n^2) \) with the size of the GSSU (not with the size of structure!), and employment of the heuristic speed optimization with \( \lambda = 1 \) further reduces the number of \( O(n^2) \) computations. The speed of SETTER was compared with that of iPARTS and SARA measuring the runtimes of all-to-all comparisons on four datasets containing RNA structures of various sizes (Table 2). The runtime of SETTER was measured on Linux machine with 4 Intel(R) Xeon(R) CPUs E7540, 2 GHz (the algorithm is not parallelized thus only one core per comparison was used) and 132 GB of RAM (however, the average memory size needed to store the representations of all

Table 2. Runtime comparison of iPARTS, SARA and SETTER for datasets of RNA structures of various size

<table>
<thead>
<tr>
<th></th>
<th>iPARTS</th>
<th>SARA</th>
<th>SETTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1.1 s</td>
<td>1.7 s</td>
<td>0.3 s</td>
</tr>
<tr>
<td>D2</td>
<td>2.6 s</td>
<td>9.2 s</td>
<td>2.4 s</td>
</tr>
<tr>
<td>D3</td>
<td>17.0 s</td>
<td>?</td>
<td>3.6 s</td>
</tr>
<tr>
<td>D4</td>
<td>2.8 min</td>
<td>?</td>
<td>21.3 s</td>
</tr>
<tr>
<td>D5</td>
<td>?</td>
<td>?</td>
<td>79.8 s</td>
</tr>
</tbody>
</table>

The D1 set contains RNA structures 1EHE:A, 1HEE:B, 1BVF:A, 2TRA:A and 1YFG:A (average length 76 nucleotides), D2 set contains ribosome P4-P6 domain (G1D-A, 1BRE:A and 1L5A:A (average length 157 nucleotides), D3 contains domain 9 of 25S tRNA 1EHE:A and 1IP09:A (average length 496 nucleotides), D4 contains 16S tRNA 133:A and 24VY:A (average length 1522 nucleotides) and D5 contains the currently largest RNA-structures in PDB— yeast 25S tRNA 3O58:1 and 3O5H:1 (average length 3396 nucleotides). The runtimes of SARA and iPARTS were obtained from their server versions. A comparison with SARA is difficult, because for three of five datasets, the server times out and return no results.

4 CONCLUSIONS

- The SETTER method divides the RNA structure into non-overlapping structural elements called GSSUs. The structural alignment is then based on the pairwise comparison using 3D similarity of the GSSUs.
- SETTER was not developed for aligning RNA molecules not containing any secondary structure. However, such cases are very rare, no such a structure is present either in the FSCOR dataset or in large 16S or 23S tRNAs.
- The SETTER algorithm scales as \( O(n^2) \) with the size of GSSU and as \( O(n) \) with the number of GSSUs in the structure. This scaling gives SETTER its unprecedented speed as the average size of GSSU remains constant irrespective of the size of the structure. However, it has been noted that due to the complexity of RNA 3D alignment, the quadratic time algorithms (or better) cannot be expected to be highly accurate (Bert et al., 2009). Therefore, the main utility of the SETTER can be in identifying potential alignment regions which can further be processed by more accurate but computationally intensive methods such as R3D Align, ARTS, SARA and DIAL approaches. The results demonstrate that SETTER produces structural alignments of comparable quality.
• The functional assignment was benchmarked against iPARTS and SARA using the classification accuracy (ACC) and the area under the ROC curve (AUC) measures for three datasets from the SCOR database. SETTER performs better than iPARTS and SARA in terms of AUC and is comparable with SARA in terms of ACC.

• The SETTER method is capable of aligning even the largest RNA structures deposited in the PDB database in a reasonable amount of time. For example, two structures of the 25S rRNA each having 3396 nucleotides and represented by 89 GSSUs are aligned in 1 min and 20 s respectively, and this represents an important addition to the portfolio of automatic RNA structural analysis tools.

ACKNOWLEDGEMENTS

The authors thank anonymous referees whose stimulating critique and comments led to the significant improvements of this article.

Funding: Czech Science Foundation (GACR) (project Nr. P202/11/0968) and Ministry of Education of the Czech Republic (MSM6046137302).

Conflict of Interest: none declared.

REFERENCES


