Published online 12 May 2009 *Nucleic Acids Research, 2009, Vol. 37, Web Server issue* ***W287–W295***
doi:10.1093/nar/gkp330

FASTR3D: a fast and accurate search tool for similar RNA 3D structures

Chin-En Lai[1], Ming-Yuan Tsai[1], Yun-Chen Liu[1], Chih-Wei Wang[1], Kun-Tze Chen[1] and Chin Lung Lu[1,2,*]

[1]Institute of Bioinformatics and Systems Biology and [2]Department of Biological Science and Technology, National Chiao Tung University, Hsinchu 300, Taiwan

Received February 21, 2009; Revised April 15, 2009; Accepted April 19, 2009

ABSTRACT

FASTR3D is a web-based search tool that allows the user to fast and accurately search the PDB database for structurally similar RNAs. Currently, it allows the user to input three types of queries: (i) a PDB code of an RNA tertiary structure (default), optionally with specified residue range, (ii) an RNA secondary structure, optionally with primary sequence, in the dot-bracket notation and (iii) an RNA primary sequence in the FASTA format. In addition, the user can run FASTR3D with specifying additional filtering options: (i) the released date of RNA structures in the PDB database, and (ii) the experimental methods used to determine RNA structures and their least resolutions. In the output page, FASTR3D will show the user-queried RNA molecule, as well as user-specified options, followed by a detailed list of identified structurally similar RNAs. Particularly, when queried with RNA tertiary structures, FASTR3D provides a graphical display to show the structural superposition of the query structure and each of identified structures. FASTR3D is now available online at http://bioalgorithm.life.nctu.edu.tw/FASTR3D/.

INTRODUCTION

In recent years, there is a fast growing interest in non-coding RNAs (ncRNAs) because, although their transcripts are not translated into proteins, they play essential roles in many cellular processes, including gene regulation, RNA modification and chromosome replication (1–4). However, the function of most ncRNAs has yet to be determined. Likewise to proteins, a common and useful approach for annotating the function of an ncRNA is by searching databases for similar RNA molecules whose functions are already known. For this purpose, several databases of ncRNAs have been proposed, such as NONCODE (5), RNAdb (6), miRBase (7), fRNAdb (8) and ncRNAdb (9). For these databases, however, the search is performed solely by querying keywords, accession numbers, transcript/organism names and/or sequences. Compared with the 20-letter protein alphabet, the 4-letter RNA alphabet is smaller and less informative, leading to that searching for similar RNA molecules based on sequence comparison/alignment is not as accurate and powerful as it does for proteins.

Actually, a more reliable way for determining the functions of ncRNAs is from the analysis on the structure level, since structures of molecules are typically more evolutionarily conserved than their sequences. In this regard, a series of recent efforts and studies has led to a substantial increase in both the number and the size of solved RNA structures deposited in the PDB and NDB databases (10,11). Therefore, it has become more and more crucial to develop automatic tools that are able to efficiently and accurately search for structurally similar RNA substructures and motifs against the PDB/NDB database. Basically, detecting structural similarities in two RNA molecules at secondary structure level is an easy job, whereas it is intractable at tertiary structure level, because it has been shown to be an nondeterministic polynomial time (NP)-hard problem even to find a constant ratio approximation algorithm for computing a pair of maximal substructures from two RNA (or protein) tertiary [three-dimensional (3D)] structures with exhibiting the highest degree of similarity (12). Therefore, currently available tools, such as ARTS (13,14), DIAL (15), SARSA (16) and SARA (17), are all based on some heuristic approaches for comparing the similarities of two RNA tertiary structures. All these methods, however, have at least quadratic-time complexity and hence are impractical for searching ever-increasing databases of RNA tertiary structures. Currently, there are several tools that can be used to search motifs in RNA structures, including FR3D (18), PRIMOS (19) and RNAMotif (20). FR3D uses a base-centred method to perform a geometric search of RNA local/composite 3D motifs. PRIMOS searches for locally structural similarities of consecutive RNA

*To whom correspondence should be addressed. Tel: +886-3-5712121 (ext. 56949); Fax: +886-3-5729288; Email: cllu@mail.nctu.edu.tw

fragments by comparing their pseudotorsion angles. RNAMotif finds the fragments of an RNA sequence that conform to a predefined descriptor of defining a particular motif of secondary structure.

In this study, we have developed a web server, called FASTR3D ('Fast and Accurate Search Tool for RNA 3D structures'), based on a hashing algorithm that is able to fast and accurately find structural similarities for a query of RNA molecule in the PDB database. In principle, this hashing algorithm consists of three main procedures as follows. The first procedure is to derive the primary sequence, secondary structure and tertiary structure information of all RNA molecules currently deposited in the PDB database and then store the derived second structures in a hash table. The secondary procedure is to derive some possible secondary structures of the query RNA if it is a primary sequence or tertiary structure. The third procedure is to search the hash table for all candidate RNAs whose secondary structures exactly match that of the query RNA, followed by primary sequence filter and/or tertiary structure filter to screen out those candidates whose primary sequences and/or tertiary structures are not equal to that of the query RNA. The FASTR3D web server is now available online at http://bioalgorithm.life.nctu.edu.tw/FASTR3D/ for public access.

In addition, our FASTR3D was tested with a number of RNA primary sequences, secondary structures and tertiary structures, and its experimental results on querying RNA primary sequences and secondary structures were also compared with those obtained by the search tool of RNA FRABASE (http://rnafrabase.ibch.poznan.pl/), which was developed by Popenda *et al.* (21) on the basis of RNA primary sequences and/or secondary structures using the methods of regular expression and pattern recognition. The comparison of experimental results on querying secondary structures reveals that FASTR3D has a comparable performance as RNA FRABASE, both with returning the search results in a short time. However, our FASTR3D is able to find more structurally similar RNAs for a query of RNA primary sequence, when compared with RNA FRABASE, because FASTR3D searches for structurally similar RNAs using the secondary structure derived from the query sequence, while RNA FRABASE searches them solely based on the primary sequence. In addition, the function of querying RNA tertiary structures in FASTR3D, as well as the online graphical display of showing the structural superposition of the query and identified structures, is not available in RNA FRABASE.

METHODS

Our FASTR3D was implemented based on a hashing algorithm whose procedure flowchart, as shown in Figure 1, consists of three major procedures. The first procedure is a preprocessing job that is to derive the primary sequence, secondary structure and tertiary structure information of all RNAs in the PDB database and particularly store the derived secondary structures (i.e. standard Watson–Crick and wobble base pairs) in a hash table. Note that the secondary structure information was derived using the RNAView program (22), while the tertiary structure information of pseudotorsion angles η and θ values was derived using the AMIGOS program (23). The second procedure is to derive the secondary structure information for the RNA queried by the user. Currently, the user can input any of the following three types of queries: (i) a PDB code of an RNA tertiary structure optionally with specified residue range, (ii) an RNA secondary structure, optionally with primary sequence, in the dot-bracket notation, and (iii) an RNA primary sequence in the FASTA format. If the query is a PDB code of an RNA tertiary structure, then its secondary structure is derived from its PDB file, which is downloaded from the PDB database, using the RNAView program (22). If the query is an RNA primary sequence, then a set of at most X suboptimal secondary structures is derived using the RNAsubopt program (24), where the default value of X is 16. It is often observed that the suboptimal secondary structure predicted by RNAsubopt for an RNA molecule may not be the true secondary structure. Therefore, we design an alternative approach as follows to derive a set of at most X most frequently occurring true secondary structures for the query RNA sequence. First, we search the PDB database for all the RNAs whose primary sequences are equal to the query sequence. Then, we use RNAView to derive all the secondary structures from the PDB files of these RNAs and from them we finally select at most X most frequently occurring secondary structures. The third procedure is to use the hash table to quickly search for all candidate RNAs whose secondary structures exactly match that of the query RNA (or any of X predicted/true secondary structures for the query RNA), followed by primary sequence filter (if the query RNA has primary sequence information) and/or tertiary structure filter (if the query is an RNA tertiary structure) to screen out those candidates whose primary sequences and/or tertiary structures are not equal to that of the query RNA.

In the following, we describe the details of the significant steps in the above procedures, including how to prepare the hash table of the secondary structures of all RNA molecules currently deposited in the PDB database, how to use this hash table to search for RNA structural similarities and how to utilize the η and θ values to efficiently screen out structurally non-similar candidates. For simplicity, we let $D = \{S_1, S_2, \ldots, S_m\}$ denote the database of the secondary structures derived from the PDB database using the RNAView program (22), and let Q be the secondary structure of the query RNA. Note that in the structural database D, each structure S_i is labelled with an integer i, to which we refer as the *index* of S_i. Moreover, we denote by the *k-tuple* a consecutive sequence of k nt (residues) within an RNA molecule. Clearly, there are $(|S| - k + 1)$ *overlapping* k-tuples for a given RNA secondary structure S with $|S|$ residues. The *offset* of a k-tuple within S is defined to be the position of its first residue with respect to the first residue of S. For convenience, we use the letter j to denote offset and use the notation $w_j(S)$ to denote the k-tuple of S that has offset j. Therefore, the position of each occurrence of each k-tuple

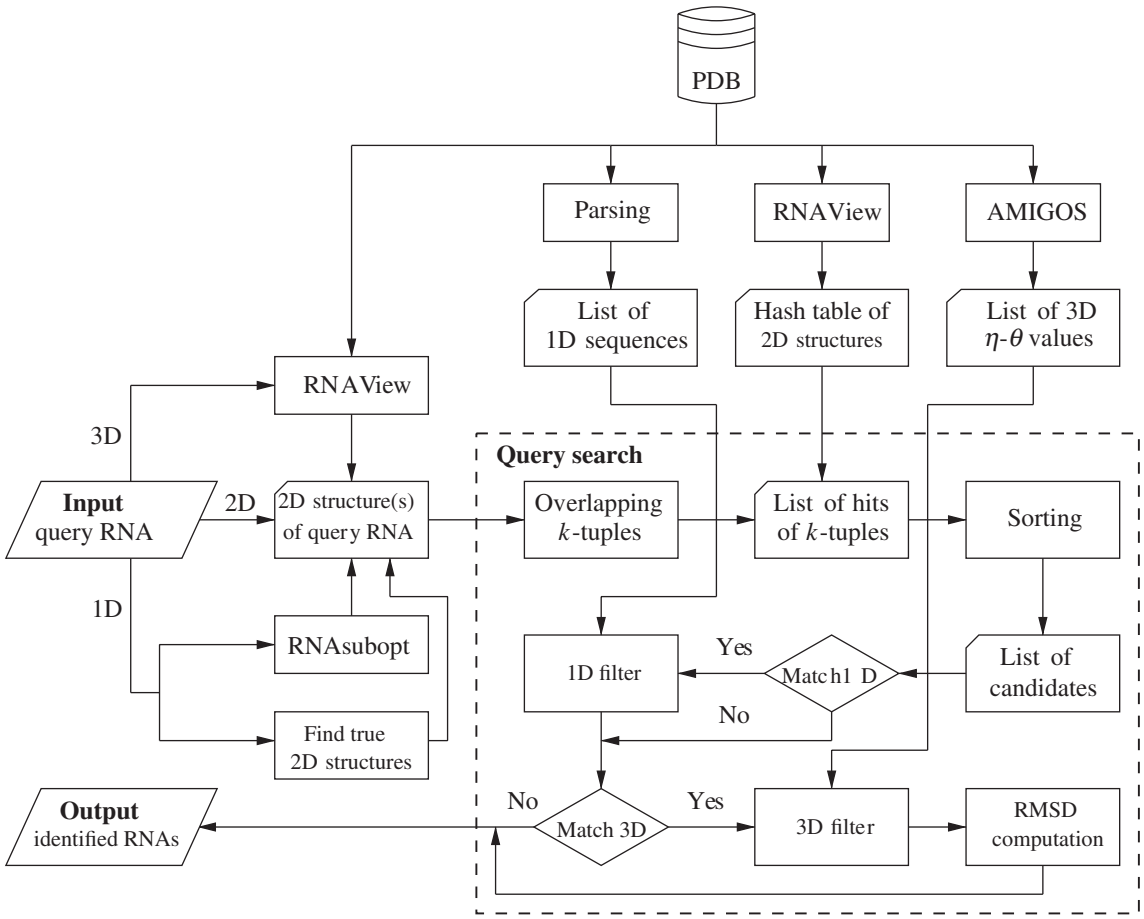


Figure 1. The procedure flowchart of FASTR3D, where 1D, 2D and 3D refer to primary, secondary and tertiary, respectively.

within a structure S_i of D can be represented by an (i, j) pair.

Hash table construction for a structural database

Here, we reorganize the structural database D by using a hash table to store the position of each occurrence of each k-tuple. Note that each RNA tertiary structure S_i in the structural database D is represented by its secondary structure in the *dot-bracket* format, where an unpaired nucleotide is denoted by a dot and a Watson–Crick (e.g. AU, UA, CG, GC) or wobble (e.g. GU and UG) base pair by a pair of opening and closing round brackets (e.g. '(' and ')'). Moreover, to correctly represent complicated secondary structures in RNA molecules, the bracket notation used in this study is extended by allowing the user to use additional squared brackets (e.g. '[' and ']') to represent simple pseudoknots and kissing loops, and curly brackets (e.g. '{' and '}') to represent high-order pseudoknotted structures.

To simplify our implementation, all the brackets appearing in an RNA secondary structure are transformed into the round brackets, since their exact pairing relationships between the opening and closing brackets are already recorded in advance using a data structure of 1D array. For each secondary structure S_i with $|S_i|$ residues, we break it into $\lceil \frac{|S_i|}{k} \rceil$ non-overlapping k-tuples and store the position of each occurrence of each k-tuple in the hash table. Recall that for any k-tuple $w = r_1 r_2 \ldots r_k$, each residue r_x, where $1 \leq x \leq k$, can be either a dot, opening bracket or closing bracket. Therefore, each of these three possible symbols is then encoded as a base-3 digit as follows: $e(\cdot) = 0_3$, $e(() = 1_3$ and $e()) = 2_3$. Using this encoding, w can be represented uniquely by a decimal integer $E(w) = \sum_{x=1}^{k} 3^{x-1} e(r_x)$. Finally, the hash table of the structural database D is represented by two data structures, a list of positions L and an array A of pointers into L. Basically, there are 3^k pointers in A, with one pointer corresponding to each of the 3^k possible k-tuples. More clearly, the pointer at position $E(w)$ of A points to the entry of L that describes the positions of the first occurrence of the k-tuple w in the database D. Then we can obtain the positions of all occurrences of w in D by traversing L from this position until we reach the location pointed by the pointer located at position $E(w) + 1$ of A. Below, we illustrate the above hash table construction with a simple example. For simplicity, we let $k = 2$ and D consist of two RNAs S_1 and S_2 whose secondary structures are S_1 = '(((···)·))' and S_2 = '·((····)) ·', respectively. In Table 1, each row contains the list of the positions of all

Table 1. A 2-tuple hash table for S_1 = '(((···)·))' and S_2 = '·((····)) ·'

2-tuple w	$E(w)$	Position lists
..	0	(1, 5), (2, 5)
·(	1	(2, 1)
·)	2	(2, 7)
(·	3	(1, 3), (2, 3)
((	4	(1, 1)
()	5	
)·	6	(1, 7), (2, 9)
)(	7	
))	8	(1, 9)

Table 2. The search of the query secondary structure Q = '(···)·'

p	$w_p(Q)$	Positions	H	M
1	(·	(1, 3)	(1, 2, 3)	**(1, 2, 3)**
		(2, 3)	(2, 2, 3)	**(1, 2, 5)**
2	..	(1, 5)	(1, 3, 5)	**(1, 2, 7)**
		(2, 5)	(2, 3, 5)	(1, 3, 5)
3	..	(1, 5)	(1, 2, 5)	(2, 2, 3)
		(2, 5)	(2, 2, 5)	(2, 2, 5)
4	·)	(2, 7)	(2, 3, 7)	(2, 3, 5)
5	)·	(1, 7)	(1, 2, 7)	(2, 3, 7)
		(2, 9)	(2, 4, 9)	(2, 4, 9)

occurrences for each of the nine possible 2-tuples, denoted by w. Then the pointer at $E(w)$ of A points to the beginning of the position list corresponding to w and the concatenation of the nine position lists in the order from top to bottom forms L.

Query substructure search

In the following, we describe how to use the hash table of the structural database D as constructed above to search for all occurrences of a query Q of an RNA secondary structure. Suppose that the length of Q is n. Then we can proceed position-by-position along Q from position 1 to $n-k+1$. At position p, where $1 \le p \le n-k+1$, we obtain the list of the positions of all the occurrences of the k-tuple $w_p(Q)$ from the hash table of D via the pointer of $E(w_p(Q))$. Let this list contain q positions, say $(i_1, j_1), (i_2, j_2), \ldots, (i_q, j_q)$. From this list, we derive a list of *hits* $H_1 = (i_1, j_1 - p, j_1), H_2 = (i_2, j_2 - p, j_2), \ldots, H_q = (i_q, j_q - p, j_q)$. This list of hits is then added to a master list M of hits that accumulates all the hits we derived when p runs from 1 to $n-k+1$. For convenience, the elements of a hit are referred to as the *index*, *shift* and *offset*. Next, we sort all the elements in M first by index and then by shift. Finally, we scan through M by looking for *runs* of hits for which the index and shift are identical. Clearly, by further sorting each of these runs by offset, we can determine the region of some structure in D that exactly matches the query structure Q. For example, we search for the query of an RNA secondary structure Q = '(···) ·' within the hash table of D as constructed in Table 1. In Table 2, column 3 displays the occurrence positions in D for each 2-tuple of Q, with corresponding hits shown in column 4, and column 5 shows the sorted M in which the run of three hits highlighted in bold indicates that there is a match between Q and S_1 that starts at the third nucleotide and ends at the eighth nucleotide. Basically, the search speed of the above hashing algorithm is proportional to the size of the master list M, which falls off rapidly with increasing the value of k. Although a greater k increases the search speed, the condition $|Q| \ge 2k-1$ should be satisfied to guarantee that the hashing algorithm will find a hit at some point in the matching region. For example, suppose that S = '((····))' and Q = '(····)'. If $k = 4$, then none of three overlapping 4-tuples in Q is able to match any of two non-overlapping 4-tuples in S. In addition, the hash table is generated in advance for a fixed k in our algorithm. Therefore, to achieve the best search speed and reduce the storage requirement, we set the value of k as $\min\{20, \lceil \frac{|Q|}{2} \rceil\}$.

Tertiary structure filter using pseudotorsion angles

Basically, the comparison of RNA conformation is a high-dimensional problem, because six standard torsion angles (α, β, γ, δ, ε and ζ) are needed to specify the backbone conformation of a single nucleotide. Duarte and Pyle (23), however, pointed out that the pseudotorsion angles η ($C4'_{i-1} - P_i - C4'_i - P_{i+1}$) and θ ($P_i - C4'_i - P_{i+1} - C4'_{i+1}$) are at least as descriptive of backbone morphology as standard torsion angles and they may be even superior in terms of specifying the backbone conformation of an individual nucleotide. This suggests that the η–θ plot can provide us a 2D representation of the conformation properties of an entire RNA molecule, so that we can carry out the rapid and accurate comparison of RNA conformations. Duarte *et al.* (19) further called such an ordered set of η–θ coordinates as an RNA *worm*. As was used by Duarte *et al.* (19), we can detect the conformation difference of two RNAs by comparing their worms based on a Euclidean metric as follows. Let Q' denote an identified candidate RNA whose secondary structure matches that of the query RNA Q with n residues, and let the worms of Q and Q' denoted by $\{(\eta_{1,1}, \theta_{1,1}), \ldots, (\eta_{1,n}, \theta_{1,n})\}$ and $\{(\eta_{2,1}, \theta_{2,1}), \ldots, (\eta_{2,n}, \theta_{2,n})\}$, respectively. The *conformational difference* between two residues $(\eta_{1,i}, \theta_{1,i})$ and $(\eta_{2,i}, \theta_{2,i})$ is defined to be $\Delta(\eta, \theta)_i = \sqrt{\Delta\eta_i^2 + \Delta\theta_i^2}$, where $\Delta\eta_i = \min\{|\eta_{1,i} - \eta_{2,i}|, \ 360 - |\eta_{1,i} - \eta_{2,i}|\}$ and $\Delta\theta_i = \min\{|\theta_{1,i} - \theta_{2,i}|, \ 360 - |\theta_{1,i} - \theta_{2,i}|\}$ (since 0° and 360° are the same). As was also pointed out by Duarte *et al.* (19), two residues $(\eta_{1,i}, \theta_{1,i})$ and $(\eta_{2,i}, \theta_{2,i})$ can be considered structurally identical if $\Delta(\eta, \theta)_i < 25°$. Therefore, based on this property, we design our tertiary structure filter to discard the identified RNA Q' from consideration if the average conformation difference $\overline{\Delta(\eta, \theta)}$ between Q and Q' is greater than or equal to a predefined cutoff, where $\overline{\Delta(\eta,\theta)} = \sqrt{(\sum_{i=1}^{n} (\Delta(\eta,\theta)_i)^2)/(n)}$ and for our purpose, the cutoff value is set as 55°.

USAGE OF FASTR3D

Input

FASTR3D provides an intuitive user interface as illustrated in Figure 2. In basic search, the user can submit

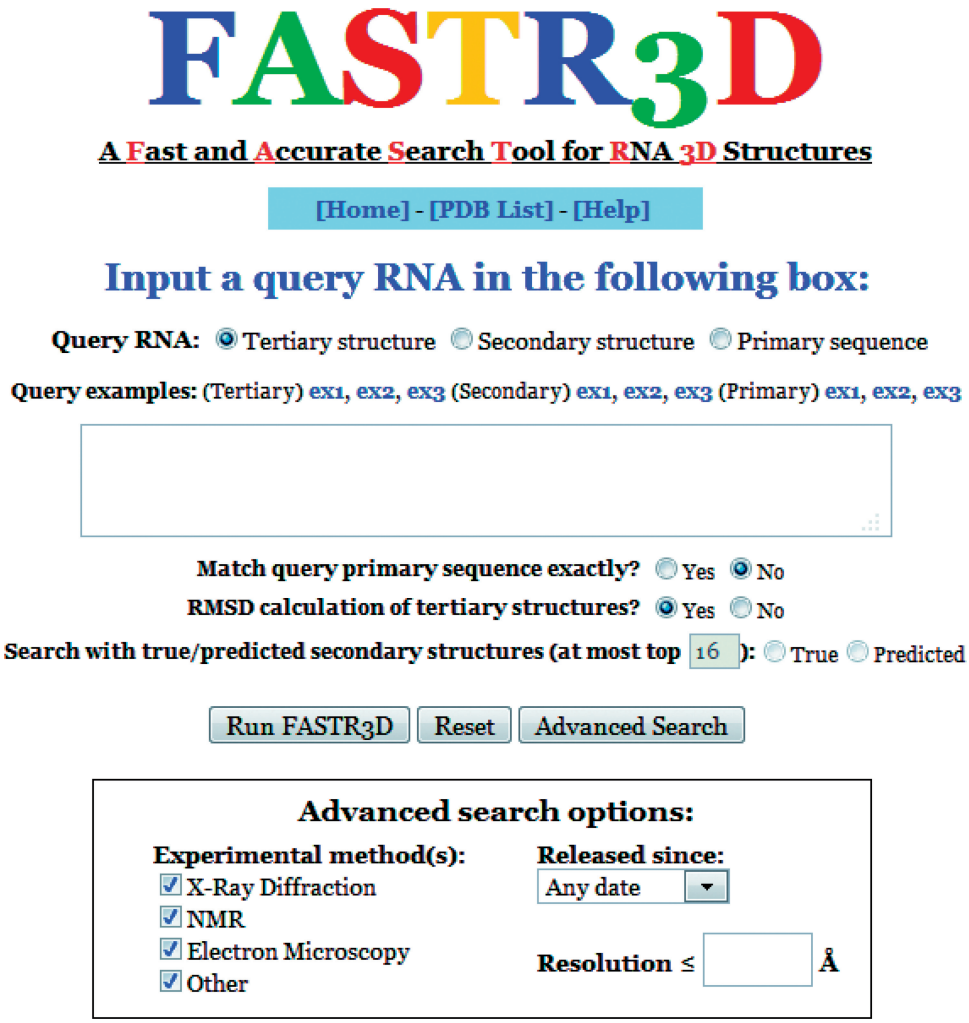


Figure 2. The web interface of FASTR3D.

a job by entering or pasting one of the following three types of queries to search for structurally similar RNA structures: (i) a PDB code of an RNA tertiary structure (default), optionally with specified residue range, (ii) an RNA secondary structure, optionally with primary sequence, in the RNA FRABASE format (i.e. a kind of dot-bracket notation) and (iii) an RNA primary sequence in the FASTA format. In addition, the user can further restrict FASTR3D to return those RNAs whose primary sequences exactly match that of the query RNA if the query RNA contains the information of its primary sequence. If the query is an RNA tertiary structure, then the user can determine whether to calculate the RMSD between the query RNA and identified candidate RNAs with the considerations of computational performance. If the query is a primary sequence, then the user can choose to use either at most X true, frequently occurring secondary structures or predicted suboptimal secondary structures to perform the PDB database search. The default value of X is 16 and can be changed by the user. In advanced search, the user can run FASTR3D with specifying additional filtering options: (i) the released date of identified RNA structures in the PDB database, and (ii) the experimental methods used to determine identified RNA structures and their least resolutions.

Output

In the output page, FASTR3D will first show the user-queried RNA molecule, as well as user-specified options. Next, it will show a detailed list of identified structurally similar RNAs (Figure 3 for an example), including corresponding PDB ID, primary sequence, secondary structure, tertiary structure, RMSD between the query and identified structures, chain ID, starting and ending nucleotide numbers, experimental method used to determine the structure, classification of RNA molecule (based on function, metabolic role, molecule type, cellular location and so on), released date in the PDB database and solved resolution. Particularly, if the query RNA is a tertiary structure, then FASTR3D allows the user to visually view, rotate and enlarge the superposition of the query RNA and each of identified RNA (Figure 4). If the query RNA is a primary sequence or secondary structure, then the user still can visually view, rotate and enlarge the tertiary structure of each identified RNA.

No.	PDB id	Primary Sequence	Secondary Structure	Tertiary Structure	RMSD	Chain	Start	End	Method	Class	Released Date	Å
1	1Y27	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	0.000	X X	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	28-DEC-04	2.4
2	2G9C	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	0.981	A A	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	21-NOV-06	1.7
3	2B57	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	0.987	A A	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	23-MAY-06	2.15
4	2EEW	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	0.989	A A	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	13-NOV-07	2.25
5	2EEU	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	0.991	A A	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	13-NOV-07	1.95
6	2EES	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	0.992	A A	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	13-NOV-07	1.75
7	1U8D	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	1.008	A A	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	23-NOV-04	1.95
8	2EET	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	1.011	A A	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	13-NOV-07	1.95
9	2EEV	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	1.014	A A	27 54	43 72	X-Ray Difraction	RIBONUCLEIC ACID	13-NOV-07	1.95
10	3DS7	GCGUGGAUAUGGCACGC CGGGCACCGUAAAUGUCCG	((((([[))))) ((((((]].....))))))	Jmol 3D	1.036	A A	27 54	43 72	X-Ray Difraction	RNA	17-FEB-09	1.85

Figure 3. The output of FASTR3D on querying an RNA tertiary structure.

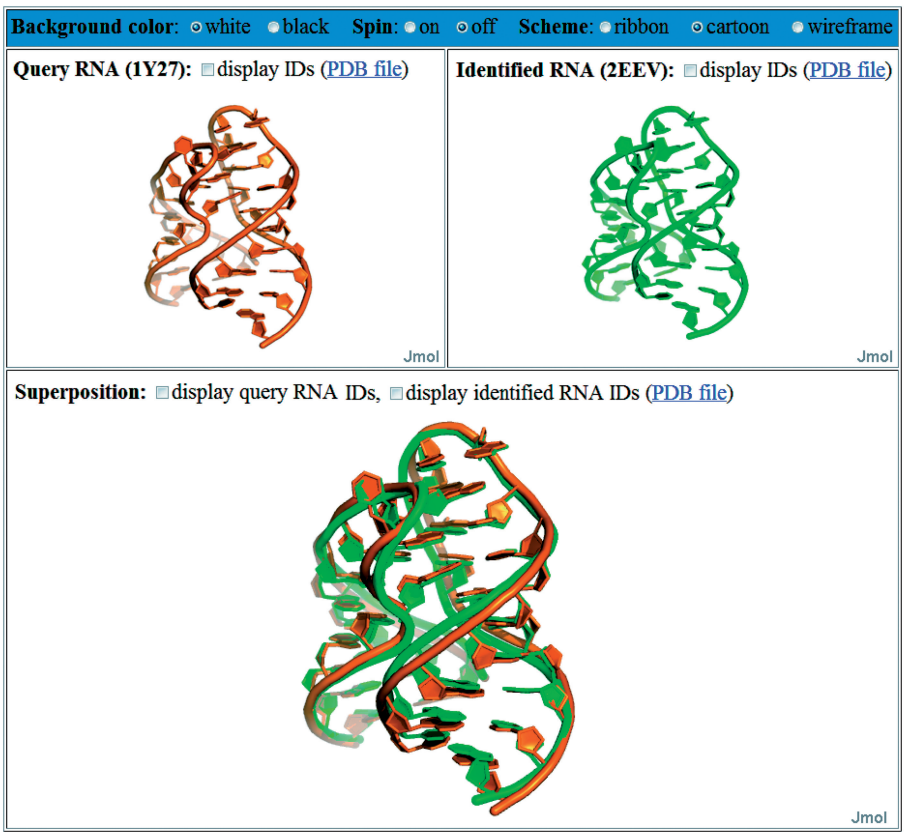


Figure 4. The visual display of query RNA (top left panel) and an identified RNA (top right panel) and their superposition (bottom panel).

EXPERIMENTAL RESULTS

For the purpose of evaluation, our FASTR3D was tested with a number of RNA primary sequences and secondary/tertiary structures, and its experimental results on querying RNA primary sequences and secondary structures were also compared with those obtained by RNA FRABASE. Basically, our FASTR3D has a comparable performance as RNA FRABASE on querying RNA secondary structures, because the basic principles behind these two tools are the same, even though they were

implemented based on different algorithms. As to the queries of RNA primary sequences, the search result of our FASTR3D is greatly different from those obtained by RNA FRABASE. Recall that, when queried with an RNA primary sequence, our FASTR3D searches for query-matching substructures (fragments) within RNA molecules using the secondary structure information of the query sequence, while RNA FRABASE searches them solely based on the query sequence. As mentioned before, RNA structures are more evolutionarily conserved than their sequences and, therefore, it can be commonly observed that different RNA sequences have the same/similar structures. This indicates that our FASTR3D may be able to find more structurally similar RNA fragments, when compared with RNA FRABASE. For the purpose of demonstration, we selected a fragment from the large subunit of the ribosome in *Haloarcula marismortui* (PDB ID: 1FFK, chain: 0, nucleotide number: 2558–2575) and applied its sequence (GGGGCUGAAG AAGGUCCC) to RNA FRABASE (with default parameters) and our FASTR3D (with searching frequently occurring true secondary structures and without matching the query sequence). Consequently, RNA FRABASE found 51 candidate RNAs that have the same primary sequence as the query, while our FASTR3D found 304 candidates that have the same secondary structure as that of the query derived by the program RNAsubopt. By further verification, we found that 94 out of the 304 tertiary substructures returned by our FASTR3D are highly similar to that of the query. This experiment demonstrates that the number of structurally similar substructures identified by our FASTR3D is greater than that by RNA FRABASE.

In the following, we demonstrate the utility of our FASTR3D on querying RNA tertiary structures, which is currently not available in RNA FRABASE. First of all, we used the tertiary substructure of a riboswitch (PDB ID: 1Y27, chain: X, nucleotide numbers: 27–43 and 54–72), as shown in Figure 5, to test our FASTR3D for its capability of searching the PDB database for structurally similar riboswitches. The so-called riboswitches are genetic regulatory elements typically found in the non-coding regions of various bacterial mRNAs. They are to regulate the expression of the genes encoded by their downstream mRNAs, via the binding of small metabolites that do not require the assistance of any protein factor (25). More importantly, it has been suggested by recent studies that riboswitches can serve as antibacterial drug targets, due to their importance to the control of genes in many bacteria (26). Basically, riboswitches are composed of a ligand binding aptamer domain and an expression platform that interfaces with RNA elements involved in gene expression. Particularly, the aptamer domain for guanine-responsive riboswitches consists of three stems and two hairpin loops. It has been reported that the interaction between these two hairpin loops, as was illustrated in Figure 5, is required for the biological function of the guanine-responsive riboswitches (27). In this experiment, FASTR3D quickly found other nine riboswitches (PDB IDs: 2G9C, 2B57, 2EEW, 2EEU, 2EES, 1U8D, 2EET, 2EEV and 3DS7) that possess substructures highly similar

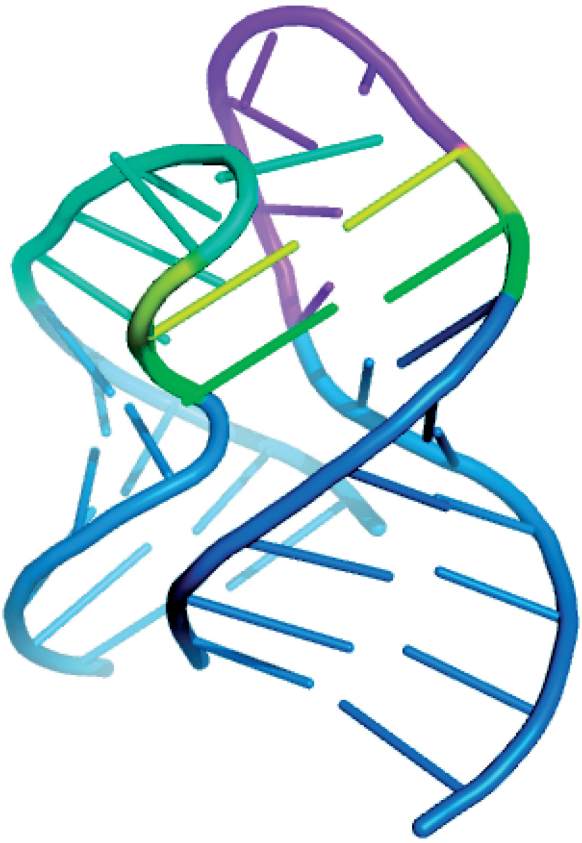
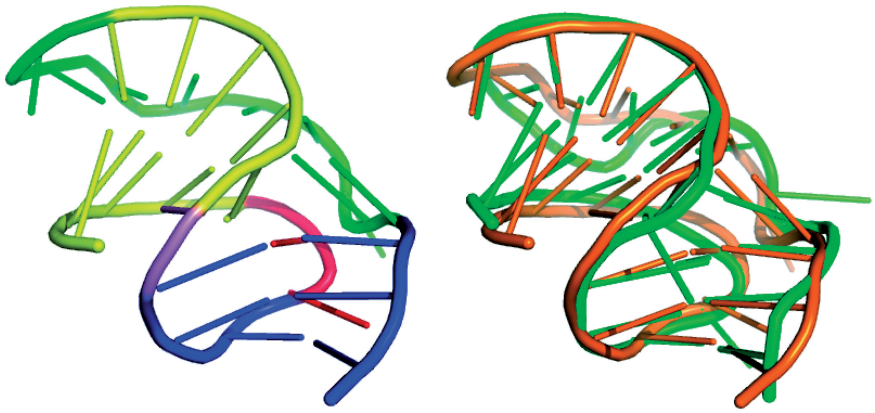


Figure 5. The interaction between two hairpin loops from the guanine-responsive riboswitch (PDB ID: 1Y27, chain: X, nucleotide numbers: 27–43 and 54–72). One loop is in cyan and the other is in magenta, with interacting residues in the loops colored yellow and green. Helical stems of the hairpin loops are in blue. This figure was prepared using the program PyMoL (http://www.pymol.org/).

to the query, where their RMSDs to the query range from 0.98 Å to 1.04 Å (Figure 3 for other details). The superposition of the query and the identified substructure in 2EEV is shown in the bottom panel in Figure 4.

Next, we tested our FASTR3D using a frameshifting pseudoknot (PDB ID: 1YG3, chain: A, nucleotide numbers: 3–30) from sugarcane yellow leaf virus (ScYLV), as shown in Figure 6a. Programmed −1 ribosomal frameshifting (−1 PRF) is a recoding mechanism by which the translational ribosome switches from the zero reading frame to the −1 reading frame at a specific position and continues its translation in the new frame. The recording of −1 PRF leads to an expression of an alternative protein, which is different from that produced by standard translation. To date, this recoding mechanism has been found to occur in many viruses, as well as a few cellular genes (28,29). The mechanism allows viruses to produce different proteins from the same mRNA and hence increases the diversity of their proteins. In most cases (but not all), the −1 PRF is commonly stimulated by an RNA pseudoknot located downstream from a heptanucleotide slip site where the −1 PRF event takes place. It has been shown that the absence or destabilization of a stable pseudoknot can eliminate efficient stimulation



(a) (b)

Figure 6. (**a**) Tertiary structure of a frameshifting pseudoknot (PDB ID: 1YG3, chain: A, nucleotide numbers: 3–30). Stem 1 is in yellow, stem 2 is in blue, loop 1 is in red, loop 2 is in green and the nucleotide (A13) between the two stems is in violet. (**b**) The superposition between the query pseudoknot (1YG3) colored orange and an identified pseudoknot (2AP5) colored green with an RMSD of 2.97 Å.

of −1 PRF in ScYLV (30). In this experiment, FASTR3D quickly found other three RNA pseudoknots (PDB IDs: 1YG4, 2AP0 and 2AP5) in the PDB database whose 3D structures are very similar to that of the query, where their RMSDs to the query is between 1.71 Å and 2.97 Å. Figure 6b displays the superposition of the query and the identified pseudoknot 2AP5 whose RMSD is 2.97 Å.

For more details on the above experiments, as well as other experiments, we refer the reader to help page of our FASTR3D at http://bioalgorithm.life.nctu.edu.tw/FASTR3D/help.html. Basically, when queried with RNA primary sequences, our FASTR3D can provide more unintended structures than RNA FRABASE as the query sequences are not as conserved as their secondary structures. On the other hand, the search results by our FASTR3D using RNA tertiary structures have the intended structures with more various sequences than those by RNA FRABASE using their primary sequences and secondary structures as the input.

SUMMARY

FASTR3D is a web-based search tool that allows the user to quickly and accurately search the PDB database for structural similarities of a query RNA. The user can query this tool by using either an RNA tertiary structure, an RNA secondary structure or an RNA primary sequence. Since the hashing algorithm, as well as tertiary structure filter, behind our FASTR3D is highly efficient, a typical query can be done in a short time. It is worth mentioning again that the function of querying RNA tertiary structures in our FASTR3D, as well as the online graphical display of showing structural superposition, is not available in RNA FRABASE. Therefore, we believe that our FASTR3D can serve as a useful tool in the study of structural biology.

FUNDING

National Science Council of Republic of China (NSC97-2221-E-009-081-MY3 in part). Funding for open access charge: ATU plan of MOE.

Conflict of interest statement. None declared.

REFERENCES

1. Doudna,J.A. (2000) Structural genomics of RNA. *Nat. Struct. Biol.*, **7**, 954–956.
2. Eddy,S.R. (2001) Non-coding RNA genes and the modern RNA world. *Nat. Rev. Genet.*, **2**, 919–929.
3. Mattick,J.S. and Makunin,I.V. (2006) Non-coding RNA. *Hum. Mol. Genet.*, **15**, R17–R29.
4. Storz,G. (2002) An expanding universe of noncoding RNAs. *Science*, **296**, 1260–1263.
5. He,S., Liu,C., Skogerbo,G., Zhao,H., Wang,J., Liu,T., Bai,B., Zhao,Y. and Chen,R. (2008) NONCODE v2.0: decoding the non-coding. *Nucleic Acids Res.*, **36**, D170–D172.
6. Pang,K.C., Stephen,S., Dinger,M.E., Engstrom,P.G., Lenhard,B. and Mattick,J.S. (2007) RNAdb 2.0–an expanded database of mammalian non-coding RNAs. *Nucleic Acids Res.*, **35**, D178–D182.
7. Griffiths-Jones,S., Saini,H.K., van Dongen,S. and Enright,A.J. (2008) miRBase: tools for microRNA genomics. *Nucleic Acids Res.*, **36**, D154–D158.
8. Kin,T., Yamada,K., Terai,G., Okida,H., Yoshinari,Y., Ono,Y., Kojima,A., Kimura,Y., Komori,T. and Asai,K. (2007) fRNAdb: a platform for mining/annotating functional RNA candidates from non-coding RNA sequences. *Nucleic Acids Res.*, **35**, D145–D148.
9. Szymanski,M., Erdmann,V.A. and Barciszewski,J. (2007) Noncoding RNAs database (ncRNAdb). *Nucleic Acids Res.*, **35**, D162–D164.
10. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.
11. Berman,H.M., Westbrook,J., Feng,Z., Iype,L., Schneider,B. and Zardecki,C. (2002) The nucleic acid database. *Acta Crystallogr. D Biol. Crystallogr.*, **58**, 889–898.
12. Kolodny,R. and Linial,N. (2004) Approximate protein structural alignment in polynomial time. *Proc. Natl Acad. Sci. USA*, **101**, 12201–12206.
13. Dror,O., Nussinov,R. and Wolfson,H. (2005) ARTS: alignment of RNA tertiary structures. *Bioinformatics*, **21 (Suppl. 2)**, 47–53.

14. Dror,O., Nussinov,R. and Wolfson,H.J. (2006) The ARTS web server for aligning RNA tertiary structures. *Nucleic Acids Res.*, **34**, W412–W415.
15. Ferrè,F., Ponty,Y., Lorenz,W.A. and Clote,P. (2007) DIAL: a web server for the pairwise alignment of two RNA three-dimensional structures using nucleotide, dihedral angle and base-pairing similarities. *Nucleic Acids Res.*, **35**, W659–W668.
16. Chang,Y.F., Huang,Y.L. and Lu,C.L. (2008) SARSA: a web tool for structural alignment of RNA using a structural alphabet. *Nucleic Acids Res.*, **36**, W19–W24.
17. Capriotti,E. and Marti-Renom,M.A. (2008) RNA structure alignment by a unit-vector approach. *Bioinformatics*, **24**, i112–i118.
18. Sarver,M., Zirbel,C.L., Stombaugh,J., Mokdad,A. and Leontis,N.B. (2008) FR3D: finding local and composite recurrent structural motifs in RNA 3D structures. *J. Mol. Biol.*, **56**, 215–252.
19. Duarte,C.M., Wadley,L.M. and Pyle,A.M. (2003) RNA structure comparison, motif search and discovery using a reduced representation of RNA conformational space. *Nucleic Acids Res.*, **31**, 4755–4761.
20. Macke,T.J., Ecker,D.J., Gutell,R.R., Gautheret,D., Case,D.A. and Sampath,R. (2001) RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Res.*, **29**, 4724–4735.
21. Popenda,M., Blazewicz,M., Szachniuk,M. and Adamiak,R.W. (2008) RNA FRABASE version 1.0: an engine with a database to search for the three-dimensional fragments within RNA structures. *Nucleic Acids Res.*, **36**, D386–D391.
22. Yang,H., Jossinet,F., Leontis,N., Chen,L., Westbrook,J., Berman,H. and Westhof,E. (2003) Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Res.*, **31**, 3450–3460.
23. Duarte,C.M. and Pyle,A.M. (1998) Stepping through an RNA structure: a novel approach to conformational analysis. *J. Mol. Biol.*, **284**, 1465–1478.
24. Wuchty,S., Fontana,W., Hofacker,I.L. and Schuster,P. (1999) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**, 145–165.
25. Mandal,M. and Breaker,R.R. (2004) Gene regulation by riboswitches. *Nat. Rev. Mol. Cell Biol.*, **5**, 451–463.
26. Blount,K.F. and Breaker,R.R. (2006) Riboswitches as antibacterial drug targets. *Nat. Biotechnol.*, **24**, 1558–1564.
27. Batey,R.T., Gilbert,S.D. and Montange,R.K. (2004) Structure of a natural guanine-responsive riboswitch complexed with the metabolite hypoxanthine. *Nature*, **432**, 411–415.
28. Farabaugh,P.J. (1996) Programmed translational frameshifting. *Microbiol. Rev.*, **60**, 103–134.
29. Namy,O., Rousset,J.P., Napthine,S. and Brierley,I. (2004) Reprogrammed genetic decoding in cellular gene expression. *Mol. Cell*, **13**, 157–168.
30. Cornish,P.V., Hennig,M. and Giedroc,D.P. (2005) A loop 2 cytidine-stem 1 minor groove interaction as a positive determinant for pseudoknot-stimulated -1 ribosomal frameshifting. *Proc. Natl Acad. Sci. USA*, **102**, 12694–12699.