FASTR: A novel data format for concomitant representation of RNA sequence and secondary structure information

Tungadri Bose, Anirban Dutta, Mohammed MH, Hemang Gandhi and Sharmila S Mande*

Bio-Sciences R&D Division, TCS Innovation Labs, Tata Consultancy Services Limited,
Pune 411 013, India

**Corresponding author (Email, sharmila.mande@tcs.com)*

Given the importance of RNA secondary structures in defining their biological role, it would be convenient for researchers seeking RNA data if both sequence and structural information pertaining to RNA molecules are made available together. Current nucleotide data repositories archive only RNA sequence data. Furthermore, storage formats which can frugally represent RNA sequence as well as structure data in a single file, are currently unavailable. This article proposes a novel storage format, 'FASTR', for concomitant representation of RNA sequence and structure. The storage efficiency of the proposed FASTR format has been evaluated using RNA data from various microorganisms. Results indicate that the size of FASTR formatted files (containing both RNA sequence as well as structure information) are equivalent to that of FASTA-format files, which contain only RNA sequence information. RNA secondary structure is typically represented using a combination of a string of nucleotide characters along with the corresponding dot-bracket notation indicating structural attributes. 'FASTR' – the novel storage format proposed in the present study enables a frugal representation of both RNA sequence and structural information in the form of a single string. In spite of having a relatively smaller storage footprint, the resultant 'fastr' string(s) retain all sequence as well as secondary structural information that could be stored using a dot-bracket notation. An implementation of the 'FASTR' methodology is available for download at *http://metagenomics.atc.tcs.com/compression/fastr*.

[Bose T, Dutta A, Mohammed MH, Gandhi H and Mande SS 2015 FASTR: A novel data format for concomitant representation of RNA sequence and secondary structure information. *J. Biosci.* **40** 571–577] **DOI** 10.1007/s12038-015-9546-0

1. Background

Non-coding RNAs (ncRNAs) play several regulatory roles that are inherently dependent on their secondary and tertiary structures. Exploring the structural diversity of the RNA population is one of the goals of researchers working in the 'Rnomics' area. Recent years have witnessed substantial growth and significant advancements in research pertaining to riboswitches (Breaker 2011), RNA structurome (Westhof and Romby 2010; McManus and Graveley 2011; Wan *et al.* 2011), RNA interference based gene silencing studies (Ossowski S *et al*., 2008), etc. Researchers in this field are trying to decipher the relationship between the structure and function of different classes of non-coding RNAs. It is therefore important to have efficient computational methods for comparison and classification of RNA molecules. Furthermore, the availability of large amount of data generated by the present generation sequencing technologies and other associated high-throughput experiments necessitate the development of novel methods/strategies for efficient archival and dissemination of the generated data.

Coding RNAs (or mRNAs), mediating translation of the genetic blueprint into functional protein molecules, play a central role in protein synthesis, On the other hand, non-coding RNAs, viz., rRNA (ribosomal RNAs) and tRNAs (transfer RNAs), are the principal actuators of the translation

Keywords. Data archival; data dissemination; file format; RNA; RNA secondary structure

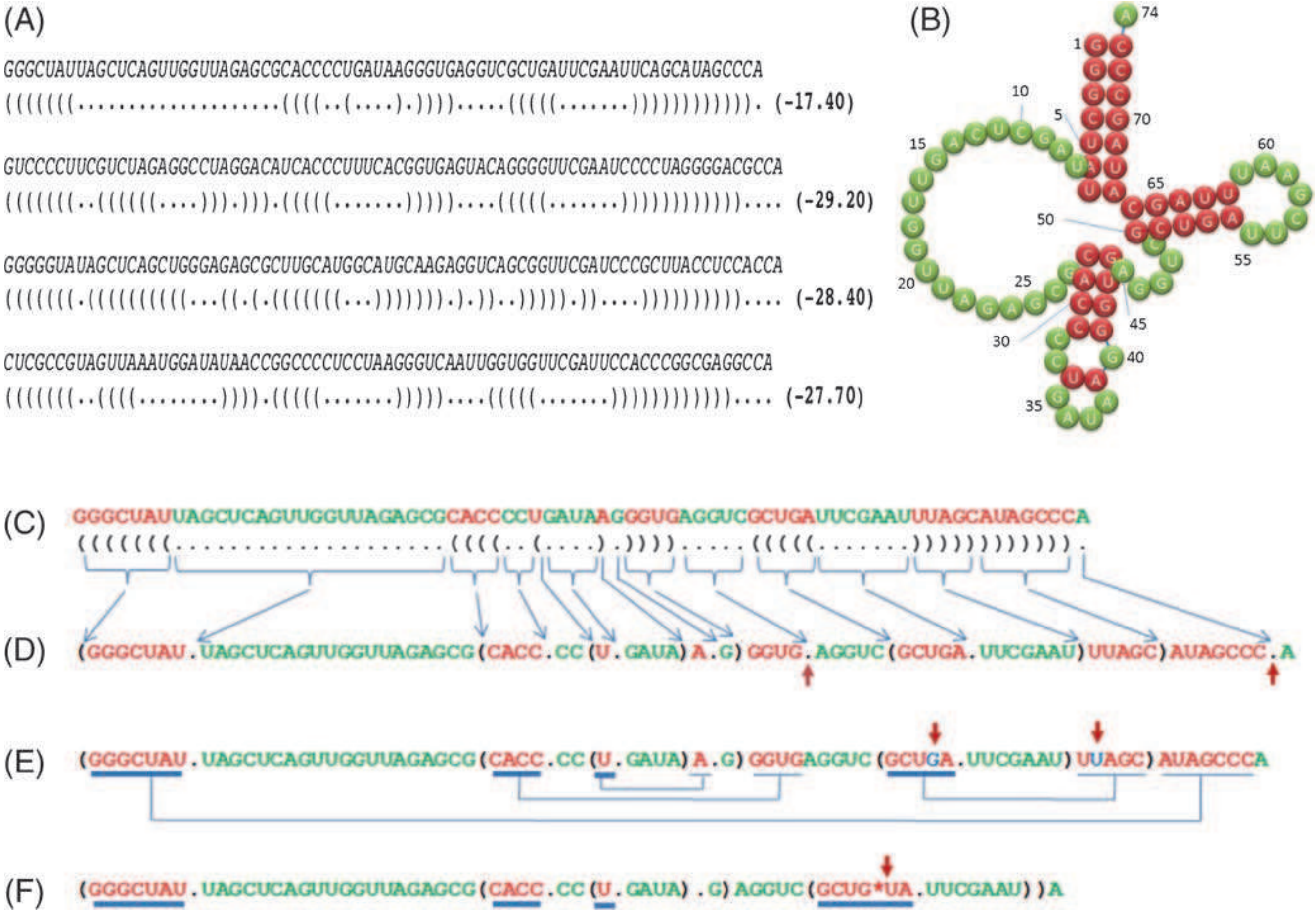
Supplementary materials pertaining to this article are available on the *Journal of Biosciences* Website at *http://www.ias.ac.in/jbiosci/sep2015/supp/Bose.pdf*

Published online: 19 August 2015

machinery. Numerous other types of non-coding RNAs (e.g. snRNA, siRNA, miRNA, etc.), additionally take part in a variety of cellular processes. In contrast to mRNAs, which code for functional molecules (proteins), non-coding RNAs themselves act as functional moieties. Studying the sequences as well as the corresponding structures is therefore equally important for understanding non-coding RNAs. A few recent studies have suggested that this dependency of RNA function on their specific secondary (and tertiary) structure holds true for coding RNAs as well (Westhof and Romby 2010; McManus and Graveley 2011; Wan *et al.* 2011). Consequently, biologists working in RNA research often require secondary structural information along with the RNA sequence data.

Several algorithms are available for predicting the secondary structure of RNA molecules from their sequences (Andronescu *et al.* 2003; Zuker 2003; Gruber *et al.* 2008; Sato *et al.* 2009; Smith *et al.* 2010; Sato *et al.* 2011). Given the importance of secondary structural information, and the easy availability of sequence-to-structure prediction tools, it is evident that archiving (and disseminating) structural information along with sequence data would be of great benefit to researchers working in this field. However, most (nucleotide) sequence repositories (e.g. Genbank, DDBJ etc.) do not store structural information along with the sequences, even for non-coding RNA sequences (wherein function is primarily governed by structural conformation).

The 'dot-bracket' notation, generated by the Vienna package (Gruber *et al.* 2008), is one of the simplest and widely accepted format for representing RNA secondary structure (figure 1A). This format uses a single character (either a 'dot' or a 'bracket') for representing the structural attribute of each ribonucleotide in a RNA sequence. Consequently, storing



```
GGGCUAUUAGCUCAGUUGGUUAGAGCGCACCCCUGAUAAGGGUGAGGUCGCUGAUUCGAAUUCAGCAUAGCCCA
(((((((....................((((..(....).)))).....(((((.......)))))))))))). (-17.40)

GUCCCCUUCGUCUAGAGGCCUAGGACAUCACCCUUUCACGGUGAGUACAGGGGUUCGAAUCCCCUAGGGGACGCCA
(((((((..((((((....))).))).(((((.......)))))....(((((.......)))))))))))).... (-29.20)

GGGGGUAUAGCUCAGCUGGGAGAGCGCUUGCAUGGCAUGCAAGAGGUCAGCGGUUCGAUCCCGCUUACCUCCACCA
(((((((.((((((((((...((.(.(((((((...))))))).).))..))))).))....)))))))))).... (-28.40)

CUCGCCGUAGUUAAAUGGAUAUAACCGGCCCCUCCUAAGGGUCAAUUGGUGGUUCGAUUCCACCCGGCGAGGCCA
(((((((..((((........)))).(((((.......)))))....(((((.......)))))))))))).... (-27.70)
```

(A)
(B)
74
1
10
70
5
15
60
65
50
55
20
25
45
30
40
35
(C)
GGGCUAUUAGCUCAGUUGGUUAGAGCGCACCCCUGAUAAGGGUGAGGUCGCUGAUUCGAAUUUAGCAUAGCCCA
(((((((....................((((..(....).)))).....(((((.......)))))))))))).
(D)
(GGGCUAU.UAGCUCAGUUGGUUAGAGCG(CACC.CC(U.GAUA)A.G)GGUG.AGGUC(GCUGA.UUCGAAU)UUAGC)AUAGCCC.A
(E)
(GGGCUAU.UAGCUCAGUUGGUUAGAGCG(CACC.CC(U.GAUA)A.G)GGUGAGGUC(GCUGA.UUCGAAU)UUAGC)AUAGCCCA
(F)
(GGGCUAU.UAGCUCAGUUGGUUAGAGCG(CACC.CC(U.GAUA).G)AGGUC(GCUG*UA.UUCGAAU))A

Figure 1. Different representations of RNA secondary structure (**A**) A typical output from the RNA-fold program (Vienna package). Alternate lines depict RNA sequences (comprising of contiguous strings of alphabets A, U, G or C), and the (predicted) secondary structures (dot-bracket notation) and the free-energy values of folding; (**B**) A schematic diagram depicting the secondary structure of a hypothetical RNA molecule in stem-and-loop representation. The 'paired' bases forming the stem are indicated in red and the 'un-paired' bases constituting the loop regions are marked in green; (**C**–**F**) The procedure adopted for generating a single 'FASTR' string from a RNA sequence and its corresponding secondary structure (in dot-bracket format); (**C**) represents the RNA sequence and the corresponding structure (represented in 'dot-bracket' format) in two consecutive lines. (**D**) and (**E**) depicts the different transformation steps followed for generating the final 'FASTR' string (**F**).

‘both’ sequence and structure information (of RNA molecules) together would require double the storage space (as compared to storing only the sequence information). Although the volume of sequence information pertaining to non-coding RNAs is not very high at this moment, it is expected to grow substantially due to recent advances in sequencing technologies. In addition, growing interest in the structural analysis of genome-wide transcriptomes is expected to generate humongous volumes of data (in the order of hundreds of MBs per sample). It would therefore be beneficial to have a file/data storage format that can accommodate both sequence and structure information without entailing any significant increase in storage requirements. An earlier work by Achawanantakun and co-workers (Achawanantakun *et al.* 2011) proposed an approach for representing RNA sequence and secondary structure information in a single string. However, this representation (also called grammar strings) archives in a lossy storage format, wherein the information pertaining to the ribonucleotide characters representing stem-regions are discarded. Consequently, the complete RNA sequence information cannot be retrieved from such ‘grammar strings’ rendering this format unsuitable for usage by data repositories.

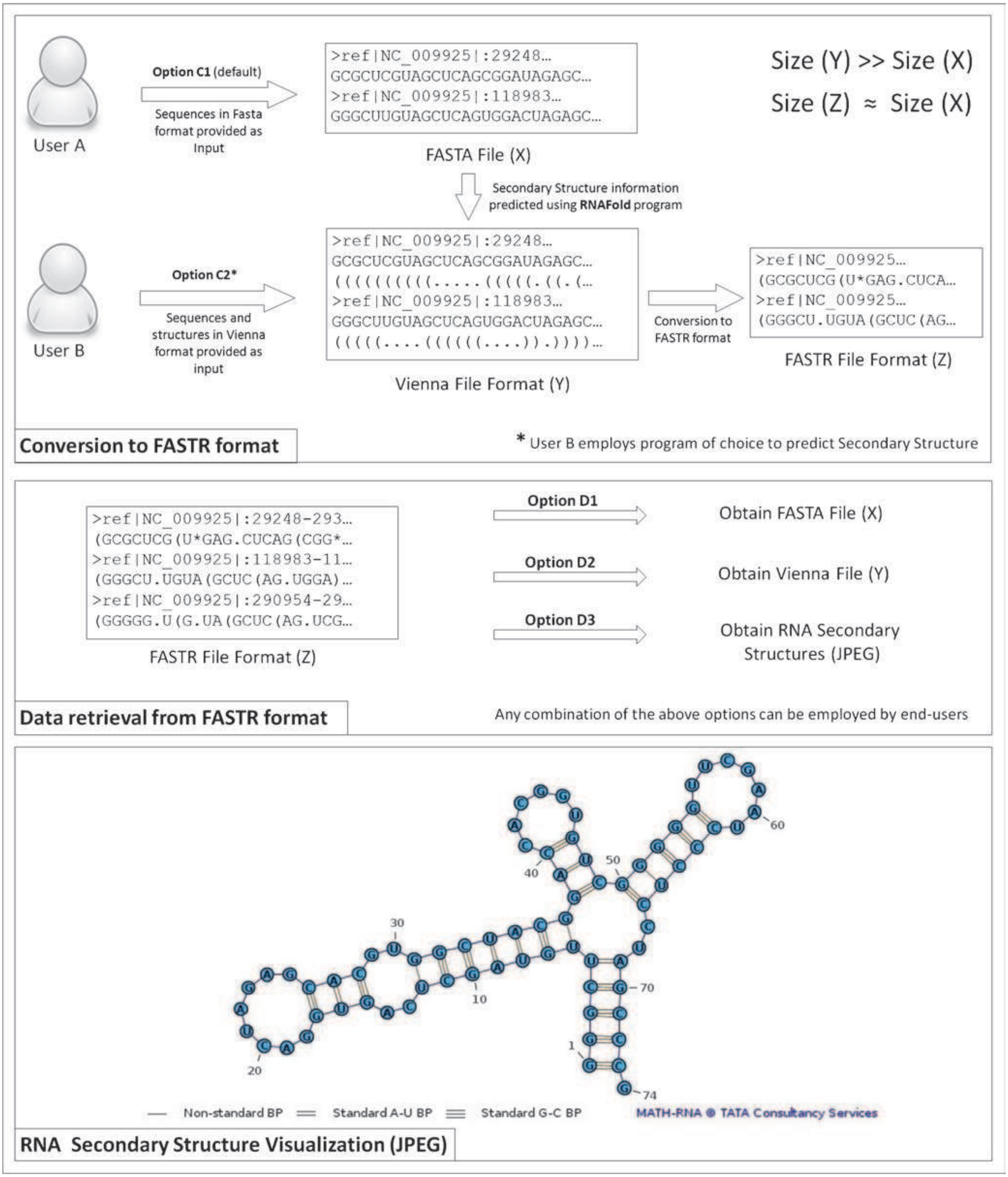
This paper describes ‘FASTR’ – a novel loss-less storage format, wherein, sequence and structural information can be frugally represented in the form of a single string. The size of the generated ‘FASTR’ string (containing both sequence and structural information) is equivalent to the size of a string that stores the sequence information alone, and therefore the proposed format does not necessitate any additional storage requirement. Moreover, considering that the ‘FASTR’ format can store information pertaining to both sequence and structure, adoption of this storage format by data repositories will confer certain additional advantages. Primarily, end-users downloading FASTR data can access both sequence and structure information without the need to run the structure prediction step. In addition, researchers depositing new RNA sequences in data repositories can opt for a submission in ‘FASTR’ format thereby incorporating the structure information (obtained in his/her analysis) along with sequence information. The availability of structural information for RNA sequences (either deposited by researchers or predicted by data archivers prior to compiling the FASTR formatted files) would also provide an opportunity to data repositories to organize data entries into structural classes.

2. Methods

Nucleotide sequences (including RNA sequences) are normally stored in FASTA format. In addition to the sequence information, a FASTA file includes unique sequence identifiers (i.e. a header) for each of the sequences it stores. The secondary structure information for a particular RNA sequence may be further obtained using any available RNA structure prediction algorithms (Andronescu *et al.* 2003; Zuker 2003; Gruber *et al.* 2008; Sato *et al.* 2009; Smith *et al.* 2010; Sato *et al.* 2011). The following section describes the steps involved in generating ‘FASTR’ strings from FASTA formatted representations of individual RNA sequences:

1. *Obtaining secondary structure information:* The sequences in the FASTA file are provided as input to the widely used RNA-fold program (included in Vienna package – *www.tbi.univie.ac.at/RNA*) for predicting secondary structures of given RNA sequences. The output of this program consists of RNA sequences, their corresponding headers and their secondary structures represented in a ‘dot-bracket’ notation (figure 1A).
2. *Finding base-pairing patterns:* This step involves utilizing the dot-bracket notation (obtained for every individual sequence) to identify stem and loop regions in each RNA sequence. Figure 1B depicts a schematic of the secondary structural conformation of a hypothetical RNA molecule, the dot-bracket notation of which is provided in figure 1C. Corresponding opening and closing brackets (in the dot-bracket notation) represent paired bases forming a ‘stem’. Accordingly, traversing the brackets in a way similar to solving an algebraic equation, allows identification of base pairing patterns. For example, opening brackets corresponding to bases 1–7 (in the RNA sequence depicted in figure 1C) have corresponding closing brackets for bases at positions 67–73. In this case, base 1 (G) pairs with base 73 (C), base 2 (G) pairs with base 72 (C), and so on.
3. *Removing redundant structural information:* In order to generate a ‘FASTR’ string, the RNA sequence and its corresponding secondary structure (in dot-bracket format) are first converted into a single modified sequence string by removing redundant structural information. In order to do this, the linear stretches of nucleotides belonging to the same structural component (i.e. the same stem or a loop) are first prefixed with a single character (i.e. an opening bracket, a closing bracket or a dot) that represents the structural attribute of the corresponding stretch (figure 1D). For example, the first seven (ribo)nucleotides in the sequence depicted in figure 1C have the same structural attribute, represented by seven contiguous opening brackets (belonging to the same stem region). This structural attribute is therefore prefixed to the corresponding sequence stretch as a single opening bracket. Figure 1D represents the single string generated by repeating this procedure for all similar stretches. The stretches of brackets and dots that have been reduced to single representative characters are indicated in this figure. It may also be noted that bases present at the extreme ends (5′ or 3′ ends) of the modified sequence string, or those occurring at a junction of two stems (i.e. a position which is preceded by one closing bracket and followed by one opening bracket), always remain unpaired. Two such

regions/positions (bases 45–49 and base 74) in the example sequence are indicated in figure 1D (with red arrows). Consequently, secondary structural information of such bases (represented by dots) are removed from the string. Figure 1E illustrates the string obtained at the end of this transformation.


User A
Option C1 (default)
Sequences in Fasta format provided as Input
FASTA File (X)
Size (Y) >> Size (X)
Size (Z) ≈ Size (X)
Secondary Structure information predicted using RNAFold program
User B
Option C2*
Sequences and structures in Vienna format provided as input
Vienna File Format (Y)
Conversion to FASTR format
FASTR File Format (Z)
Conversion to FASTR format
* User B employs program of choice to predict Secondary Structure
Option D1
Obtain FASTA File (X)
Option D2
Obtain Vienna File (Y)
Option D3
Obtain RNA Secondary Structures (JPEG)
FASTR File Format (Z)
Data retrieval from FASTR format
Any combination of the above options can be employed by end-users
1
10
20
30
40
50
60
70
74
Non-standard BP
Standard A-U BP
Standard G-C BP
MATH-RNA © TATA Consultancy Services
RNA Secondary Structure Visualization (JPEG)

```
>ref|NC_009925|:29248...
GCGCUCGUAGCUCAGCGGAUAGAGC...
>ref|NC_009925|:118983...
GGGCUUGUAGCUCAGUGGACUAGAGC...
```

```
>ref|NC_009925|:29248...
GCGCUCGUAGCUCAGCGGAUAGAGC...
((((((((((.....(((((.((.(...
>ref|NC_009925|:118983...
GGGCUUGUAGCUCAGUGGACUAGAGC...
(((((....((((((....)).)))))...
```

```
>ref|NC_009925...
(GCGCUCG(U*GAG.CUCA...
>ref|NC_009925...
(GGGCU.UGUA(GCUC(AG...
```

```
>ref|NC_009925|:29248-293...
(GCGCUCG(U*GAG.CUCAG(CGG*...
>ref|NC_009925|:118983-11...
(GGGCU.UGUA(GCUC(AG.UGGA)...
>ref|NC_009925|:290954-29...
(GGGGG.U(G.UA(GCUC(AG.UCG...
```

Figure 2. An overview of various features available in the FASTR implementation.

4. *Removing redundant sequence information:* Since base-pairing mostly occurs between two complementary bases (A pairs with U; G pairs with C), it is sufficient to store information of any one of the paired bases. Guided by this principle, our encoding method retains the first base in every complementary base-pair in the string (when the sequence is read from 5′ to 3′). In Figure 1E, underscores have been used to indicate corresponding complementary regions. While, the characters underscored with a thicker line are retained in the string, the corresponding complementary regions (underscored by thinner lines) are discarded. In cases, where the standard complementarity rules are violated (for example base G at position 53 incorrectly paired with U at position 63), the first base of such a base-pair, (the base occurring near the 5′ end of the string) is retained in its place and is immediately followed by an asterisk ('*') character and its base-pair. The second base is thereafter removed from its original position in the string. Figure 1F represents the 'FASTR' string after redundant sequence information has been removed in the above mentioned manner. The bases underscored with thinner lines (complementary regions) in figure 1E have been removed in figure 1F.

The 'FASTR' strings thus obtained for a given set of RNAs, contain all the information pertaining to their sequences and corresponding secondary structures. Appropriate decoding steps can be employed to regenerate the sequence and/or structure data from the FASTR strings. Flow-charts in supplementary figure 1 depict the encoding and decoding steps of FASTR.

3. Results

The utility of the 'FASTR' storage format was evaluated using a RNA dataset (all.frn.tar.gz) downloaded from *ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/all.frn.tar.gz* (NCBI database). This archive consists of 1,66,775 RNA sequences (both rRNA and tRNA) belonging to approximately 2,300 bacterial and archeal species (at the time of download). All sequences (along with their corresponding header information) were concatenated into a single file (all-frn.FASTA). The resulting file, consisting of multiple RNA sequences in FASTA format (multi-FASTA file), was provided as input to the RNA-fold program. The output of the RNA-fold program (all-frn.FASTA.out), consisting of sequence information (along with corresponding headers) and the predicted structural information (in dot-bracket notation), was then processed following the 'FASTR' encoding steps (as described in the Methods section). The generated 'FASTR' file contained information pertaining to (a) sequence, (b) predicted secondary structure, (c) sequence headers, as well as the (d) the predicted free-energy information corresponding to each individual sequence.

The sizes of the files generated at various stages of the process (described above), along with a description of the information content that they hold, are given in Table 1. It is evident from these results that in spite of holding several additional fields of information (as compared to the original FASTA file), the size of the 'FASTR' file is equivalent to the size of the original FASTA file. This indicates the efficiency of the proposed (FASTA to FASTR) conversion strategy, which provides a convenient way of storing both the sequence and structural information pertaining to a RNA molecule frugally, and in a single unified file format.

Table 1. A comparison of the size of files generated at various stages of FASTR conversion, along with a description of the information content

File name	Contents	Description	File size (megabytes)
all-frn.fasta	RNA Sequences with their corresponding Headers	File generated by downloading all.frn.tar.gz archive from NCBI and subsequently compiling all constituent sequences into a multi-fasta formatted file.	63.74
all-frn.fasta.out	RNA Sequences with their corresponding Headers + Predicted Secondary Structures in Dot-Bracket Notation (along with corresponding Free Energy Values)	File generated by providing all-frn.fasta as input to the RNA-fold program (Vienna Package).	118.19
all-frn.fastr	Fastr strings that capture (in a single string) both RNA sequence information and Predicted Secondary Structure information + Corresponding Headers and Predicted Free Energy Values	Output file generated by providing the all-frn.fasta.out file as input to fastr.pl program.	63.77

4. Discussion

Major sequence repositories like the NCBI, DDBJ, etc., store ribonucleotide data in FASTA formatted files which does not contain any structural information. Consequently, researchers intending to examine the secondary structural conformation of any RNA molecule are required to perform a secondary structure prediction step, using one of the available methods. Since 'FASTR' format allows concomitant storage of both sequence and structure, adoption of such format by curated database providers will benefit the research community. Additionally, scientists intending to upload RNA information to a data archive will also be able to deposit both the sequence as well as the secondary structure (which is either experimentally determined or predicted *in silico*) of a RNA molecule using the 'FASTR' format. Inspite of being a loss-less archival format, the 'FASTR' representation does not entail any extra storage requirement for storing both sequence and secondary structure information (as compared to the corresponding FASTA string). Therefore the storage archivers are also expected to benefit from this new data storage format.

It may be noted that the Vienna format is unable to represent the pseudoknot architecture in RNA molecules. Given that the input format for generating of 'FASTR' string is the 'dot-bracket' notation in Vienna format, the 'FASTR' strings are also incapable of storing structural information pertaining to pseudoknots. However, it is also significant to note that inspite of the functional importance of pseudoknots in RNA structure, the occurrence of pseudoknots in RNA molecules are relatively infrequent (Aalberts and Hodas 2005). Furthermore, most of the available tools for the downstream (bioinformatic) analysis of RNA data cannot handle pseudoknot structures (Smit *et al.* 2008). Consequently, pseudoknot removal from the RNA structure models is an active area of research (Smit *et al.* 2008; Chiu and Chen 2014) and is an important pre-processing step for most computational analysis. The incapability of 'FASTR' format to address pseudoknots may thus be considered to be an acceptable trade-off, especially in light of the frugal representation and storage advantage it has to offer.

Both 32-bit and 64-bit implementations of the 'FASTR' methodology are available for download at *http://metagenomics.atc.tcs.com/compression/fastr*. An overview of various features, input and output formats available in the FASTR implementation is depicted in figure 2. For using 'FASTR', end-users would require a standard Linux OS desktop with Java Runtime Environment (ver. 1.6 or higher) installed. The execution of the program is also dependent on the availability of the *RNAfold* program (from Vienna-Package). For the convenience of end-users, the FASTR webserver also provides download links for (a) various dependencies of the FASTR program viz. JRE, *RNAfold* program (b) tools for inter-converting between various RNA secondary structure file formats (Wiese *et al.* 2005; Lorenz *et al.* 2011; Antczak *et al.* 2014), and (c) other nucleotide sequence archival tools like DSRC (Deorowicz and Grabowski 2011), DELIMINATE (Mohammed *et al.* 2012), BIND (Bose *et al.* 2012), MFCompress (Pinho and Pratas 2014) and FQC (Dutta *et al.* 2015).

5. Conclusions

RNA secondary structure is pivotal in understanding the underlying mechanisms behind the functioning of riboswitches, gene splicing activities, epigenetics, ncRNA based activities, etc. It is therefore necessary to capture and store the secondary structural information along with the RNA sequences. The proposed 'FASTR' file format has been designed to concomitantly store both the sequence as well as structural information pertaining to RNA molecules. Unlike other representations (like the Vienna Format), this frugal representation of RNA data is highly efficient and ensures that no extra storage space is entailed (as compared to the FASTA representation). This novel approach of concomitantly storing RNA sequence and structure information (in a loss-less manner) is expected to immensely benefit the researchers associated with RNA biology.

Acknowledgements

Tungadri Bose is also a PhD scholar in the Indian Institute of Technology, Bombay, and would like to acknowledge the Institute for its support. We also thank Mr Pranav Nawathe for assisting in the development of the FASTR webpage.

References

Aalberts DP and Hodas NO 2005 Asymmetry in RNA pseudoknots: observation and theory. *Nucleic Acids Res.* **33** 2210–2214

Achawanantakun R, Sun Y and Takyar SS 2011 ncRNA consensus secondary structure derivation using grammar strings. *J. Bioinform. Comput. Biol.* **9** 317–337

Andronescu M, Aguirre-Hernández R, Condon A and Hoos HH 2003 RNAsoft: A suite of RNA secondary structure prediction and design software tools. *Nucleic Acids Res.* **31** 3416–3422

Antczak M, Zok T, Popenda M, Lukasiak P, Adamiak RW, Blazewicz J and Szachniuk M 2014 RNApdbee–a webserver to derive secondary structures from pdb files of knotted and unknotted RNAs. *Nucleic Acids Res.* **42** W368–W372

Bose T, Mohammed MH, Dutta A and Mande SS 2012 BIND - an algorithm for loss-less compression of nucleotide sequence data. *J. Biosci.* **37** 785–789

Breaker RR 2011 Prospects for riboswitch discovery and analysis. *Mol. Cell.* **43** 867–879

Chiu JK and Chen YP 2014 Efficient conversion of RNA pseudoknots to knot-free structures using a graphical model. *IEEE Trans. Biomed. Eng.* Dec 2

Deorowicz S and Grabowski Sz 2011 Compression of DNA sequence reads in FASTQ format. *Bioinformatics* **27**(6) 860–862

Dutta A, Haque MM, Bose T, Reddy CVSK and Mande SS 2015 FQC: A novel approach for efficient compression, archival, and dissemination of fastq datasets. *J. Bioinforma. Comput. Biol.* **13** 1541003

Gruber AR, Lorenz R, Bernhart SH, Neuböck R and Hofacker IL 2008 The Vienna RNA websuite. *Nucleic Acids Res.* **36** W70–W74

Lorenz R, Bernhart SH, Siederdissen CH, zu Tafer H, Flamm C, Stadler PF and Hofacker IL 2011 Vienna RNA package 2.0. *Algorithms Mol. Biol.* **6** 26

McManus CJ and Graveley BR 2011 RNA structure and the mechanisms of alternative splicing. *Curr. Opin. Genet. Dev.* **21** 373–379

Mohammed MH, Dutta A, Bose T, Chadaram S and Mande SS 2012 DELIMINATE–a fast and efficient method for loss-less compression of genomic sequences. *Bioinformatics.* **28** 2527–2529

Ossowski S, Schwab R and Weigel D 2008 Gene silencing in plants using artificial microRNAs and other small RNAs. *Plant J.* **53** 674–690

Pinho AJ and Pratas D 2014 MFCompress: a compression tool for FASTA and multi-FASTA data. *Bioinformatics* **30** 117–118

Sato K, Hamada M, Asai K and Mituyama T 2009 CENTROIDFOLD: a web server for RNA secondary structure prediction. *Nucleic Acids Res.* **37** W277–W280

Sato K, Kato Y, Hamada M, Akutsu T and Asai K 2011 IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics* **27** i85–i93

Smit S, Rother K, Heringa J and Knight R 2008 From knotted to nested RNA structures: a variety of computational methods for pseudoknot removal. *RNA* **14** 410–416

Smith C, Heyne S, Richter AS, Will S and Backofen R 2010 Freiburg RNA Tools: a web server integrating INTARNA, EXPARNA and LOCARNA. *Nucleic Acids Res.* **38** W373–W377

Wan Y, Kertesz M, Spitale RC, Segal E and Chang HY 2011 Understanding the transcriptome through RNA structure. *Nat. Rev. Genet.* **12** 641–655

Westhof E and Romby P 2010 The RNA structurome: high-throughput probing. *Nat. Methods.* **7** 965–967

Wiese KC, Glen E and Vasudevan A 2005 JViz.Rna–a Java tool for RNA secondary structure visualization. *IEEE Trans. Nanobiosci.* **4** 212–218

Zuker M 2003 Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* **31** 3406–3415

MS received 18 February 2015; accepted 10 July 2015

Corresponding editor: Mandar V Deshmukh