RNAshapes 2.1.5 manual

Peter Steffen[1], Björn Voß[2], Marc Rehmsmeier[3], Jens Reeder[1] and Robert Giegerich[1]

[1]Faculty of Technology, Bielefeld University, 33594 Bielefeld, Germany,
[2]Institute of Biology II, Experimental Bioinformatics, Freiburg University,
Schänzlestr. 1, 79104 Freiburg, Germany,
[3]Center for Biotechnology (CeBiTec), Bielefeld University, 33594 Bielefeld, Germany

April 15, 2008

Contents

1 Introduction

This manual describes RNAshapes, a software package that integrates three RNA analysis tools based on the abstract shapes approach [6]: the analysis of shape representatives [1], the calculation of shape probabilities [7], and the consensus shapes approach [4]. This new package is completely reimplemented in C, and outruns the original implementations significantly in runtime and memory requirements. Additionally, we added a number of useful features like suboptimal folding with correct dangling energies, structure graph output, shape matching, and a sliding window approach.

In the following, we will shortly review the notion of abstract shapes, and explain where its power comes from. We will then provide an overview of the problems that can be approached in the new way. Section 2 gives an exemplified introduction to the RNAshapes interface. Section 3 describes all program options in detail.

The abstract shapes approach. An RNA shape is an abstract representation of an RNA secondary structure. It is inspired by the dot-bracket representation known from the Vienna RNA package [3]. Consider the following sequence and two secondary structures from its folding space in dot-bracket representation:

```
AUCGGCGCACAGGACAUCCUAGGUACAAGGCCGCCCGUU
..(((.((..(((....))).(((.....)))))))..
..(((.....(((....))).(((.....)))..)))..
```

The shapes approach offers five abstraction levels – or *shape types* – ordered in their degree of abstraction. Common to all levels is that they abstract from loop and stack lengths, where unpaired regions are represented by an underscore and stacking regions by a pair of squared brackets. This is the least abstract shape type 1, so the two example secondary structures become:

```
_[_[_[_]_[_]]]_
_[_[_]_[_]_]_
```

The succeeding shape types gradually increase abstraction, ending in type 5, where no unpaired regions are included and nested helices are combined. In this type, our example structures are both represented as:

```
[[][]]
```

For a detailed description of all shapes types see Section 3. These abstractions form the basis of all applications of RNA abstract shape analysis. In the following we give a short overview of the main applications, all integrated in the RNAshapes package.

Shape representative analysis. Current RNA folding algorithms either calculate a single, minimum free energy prediction, or a huge number of suboptimal structures, most of which are quite similar and therefore redundant. With shapes, we abstract from the concrete secondary structures and only consider classes of structures that fall into different shapes. The *shape representative* (in short: *shrep*) of a shape is the structure with the minimum free energy inside a shape class.

Shape probabilities. In [7], we extended the shapes approach to the computation of shape probabilities. The probability of a shape is the sum of the probabilities of all structures that fall into this shape. Several analyses indicate that this approach is quite effective. For example, an analysis of a conformational switch shows the existence of two shapes with probabilities approximately $\frac{2}{3}$ vs. $\frac{1}{3}$, whereas the analysis of a micro RNA precursor reveals the hairpin shape with a probability near to 1.0 [7].

The new implementation contains three approaches for probability analysis, suitable for different input sizes: Complete probability analysis, sampling shapes probability analysis, and fast high probability shape analysis (see Section 2).

Consensus shapes. The well-known Sankoff algorithm [5] for simultaneous RNA sequence alignment and folding is currently considered an ideal, but computationally over-expensive method. Available tools implement this algorithm under various pragmatic restrictions. In [4], we proposed to redefine the consensus structure prediction problem in a way that does not imply a multiple sequence alignment step. For a family of RNA sequences, our method RNAcast explicitly and independently enumerates the near-optimal abstract shape space, and predicts as the consensus an abstract shape common to all sequences. For each sequence, it delivers the thermodynamically best structure that has this common shape. Since the shape space is much smaller than the structure space, and identification of common shapes can be done in linear time (in the number of shapes considered), the method is essentially linear in the number of sequences. Our evaluation showed that the new method compares favorably with available alternatives [4]. It is particularly useful on sequences with low conservation, where methods based on sequence alignment cannot be employed. We have now integrated RNAcast into the RNAshapes package.

2 The RNAshapes interface by example

In the following, we give an exemplified introduction to the RNAshapes interface. All sequence files used here can be found in the directory `examples`.

2.1 Shape representative analysis

As an example sequence, we use the alanine tRNA of Natronobacterium pharaonis (gb: AB003409.1/96-167). It is contained in sequence file `pharaonis.seq` in the examples directory. With option **-f**, we

specify the name of the input file for RNAshapes. When given no additional parameters, RNAshapes works in shape representative analysis mode:

```
RNAshapes -f pharaonis.seq
```

Alternatively, we can also call RNAshapes directly with the sequence:

```
RNAshapes GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
```

This calls both give the following result:

```
        GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
-35.90  ((((((((((((((.((((.....((((((.....))))))..)))))))))).........))))))))).  []
```

The predicted mfe-structure is one hairpin with internal loops. In default setting, RNAshapes considers the energy range 10% above the mfe. In this example, all structures in this energy range (-35.90 to -32.31) fall into the hairpin shape `[]`, and the printed structure is the best (i.e. with lowest energy) structure of this shape class. This is the shape representative structure (in short: *shrep*).

To increase the energy range, we can use the switches **-e** (for absolute difference above the mfe), or **-c** (for a percent value above the mfe). Let's try with option **-e** 5:

```
RNAshapes -f pharaonis.seq -e 5
        GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
-35.90  ((((((((((((((.((((.....((((((.....))))))..)))))))))).........))))))))).  []
-32.20  (((((((.....((.((((.....((((((.....))))))..))))))(((.......))).)))))))).  [[][]]
-31.70  ((((((...((((.......)))).((((((.....))))))..... (((((.......))))).))))))).  [[][][]]
```

This gives three results. The biologically active structure of the alanine tRNA is the clover-leaf structure, the third result of the above call.

By default, RNAshapes uses the most abstract shape type (type 5). The shape type can be changed with option **-t**. For example, the same analysis with shape type 3 gives the following results:

```
RNAshapes -f pharaonis.seq -e 5 -t 3
        GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
-35.90  ((((((((((((((.((((.....((((((.....))))))..)))))))))).........))))))))).  [[[[]]]]
-33.20  ((((((((((((((.(((...((.((((((.....))))))..)))))))))).........))))))))).  [[[[[]]]]]
-32.20  (((((((.....((.((((.....((((((.....))))))..))))))(((.......))).)))))))).  [[[[]]][]]
-31.90  ((((((..........((((.....((((((.....))))))..))))(((((.......))))).))))))).  [[[]][]]
-31.70  ((((((...((((.......)))).((((((.....))))))..... (((((.......))))).))))))).  [[][][]]
```

With option **-Z**, we can tell RNAshapes to color all dotbracket and shape strings in the output such that corresponding structural elements have the same color in both representations. This is especially useful for examining and learning the different shape types.

```
RNAshapes -f pharaonis.seq -e 5 -t 3 -Z
```

To get a more visual impression, we can also generate structure graphs of these results with option **-g**. As parameter, **-g** receives the number of graphs that should be generated:

```
RNAshapes -f pharaonis.seq -e 5 -g 3
```

This call generates the postscript files `rna_1.ps`, `rna_2.ps`, and `rna_3.ps` (Figure 1).

So far, we have only entered a single sequence as input for RNAshapes. In practice, multiple sequence files are often more convenient. RNAshapes supports the fasta file format. We call RNAshapes with option **-f** and the example file `random.fasta`, that contains two random sequences:

```
RNAshapes -f random.fasta
>random1
        AGUCUUGACGGUCGCAGCCGUCUCGCGUGGGGCCGCGUCUUUAGGGAUCCAAGGCAUGAGAAUUGAUCGU
-19.30  .((((((.(((((.((((......)).)).)))))).((((....)))).))))))..............  [][]]
-18.70  .......(((((((((((((((((((((....)))))).....)))))....)))).))).....)))))))  []
-18.60  .((((((.(((((.((((......)).)).)))))).((((....)))).))))))..((.......)).  [][]][]
>random2
        UUUGAAUAUCAGCUCAACAGGCAACGGUCCGGUGUAAUCUUUCUGUUAUGGCAGUUAUACCCCGGGAUGAUCACCGCAACUGCGUAGGGU
-20.60  ..(((.((((.(((.....)))..(((...(((((((.....((((....))))))))))))))).)))).))).(((....))).....  [][]][]
-19.00  ..........((((.........((..((((.(((......((((....))))...))).))))..)).....(((....))).))))  [][]]
```

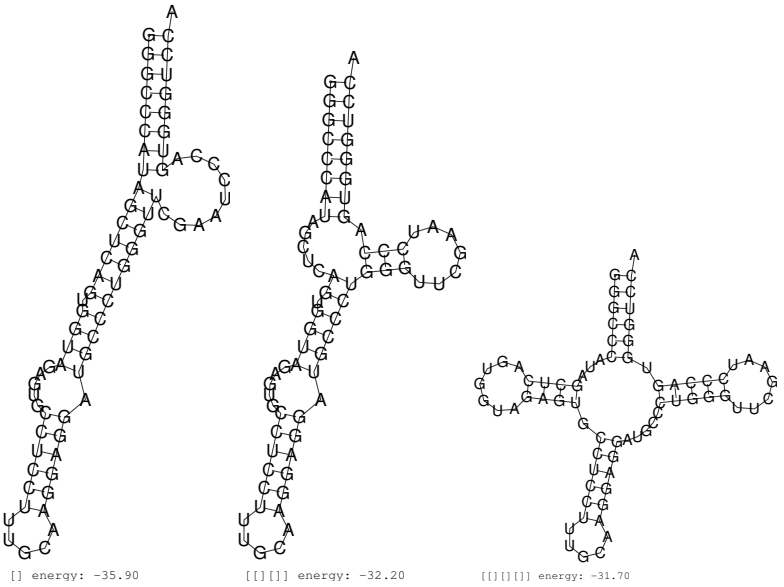

Figure 1: Postscript structure graphs generated with option **-g**

The sequences in the input file are processed one after another, and the results are printed together with the corresponding sequence descriptor.

We can also feed RNAshapes by standard input, which is useful for directly processing sequence output generated by other programs. For example, the call

```
cat random.fasta | RNAshapes
```

is equivalent to the **-f** call above. Finally, we can also enter sequences interactively by calling RNAshapes without any input sequence (but possibly with parameters):

```
RNAshapes -e 5
Interactive mode. Try ‘RNAshapes -h‘ for more information.

Input sequence (upper or lower case); :q to quit, -h for help.
....,....1....,....2....,....3....,....4....,....5....,....6....,....7....,....8
```

The interactive mode supports the following features:

- Direct input of sequences (or alternatively by cut-and-paste).
- Input history with keyboard keys `UP` and `DOWN` (library `editline` required).
- Complete interface to all program options. Instead of a sequence, simply type one or several RNAshapes commands. Examples:
  - `-h` shows the command overview.
  - `-H <option>` shows a detailed help for the given option.
  - `-c 20` sets the energy range to 20%.
  - `-C -f ires.fasta` switches to consensus shapes mode and starts the analysis with file `ires.fasta`.
  - `:s` shows the current settings.

2.2 Shape probabilities

RNAshapes offers a number of options to calculate structure and shape probabilities. The probability of a shape is the sum of the probabilities of all structures that fall into this shape.

The probability analysis modes are controlled with several options (**-p**, **-q**, **-P**, **-i**, **-r**, **-F**, **-T**). The most convenient (but computational most expensive one) is **-q**, the complete probability analysis with simultaneous shrep calculation. Applying this to the alanine tRNA from above gives the following result:

```
RNAshapes -q -f pharaonis.seq
        GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
-35.90  ((((((((((((((.((((.....((((((.....))))))..)))))))))))).........)))))))).  0.9897545  []
-32.20  (((((((.....((.((((.....((((((.....))))))..))))))(((.......))).)))))))).  0.0089891  [[][]]
-31.70  ((((((...((((.......)))).((((((.....))))))).....((((.......))))).))))))).  0.0012534  [[][][]]
-27.70  ((((((.((((((.......)))).((((((.....))))))..))..((((.......))))).))))))).  0.0000029  [[[][]][]]
```

The shape holding the mfe-structure, `[]`, is also the most probable one (98.97545%), whereas the other shapes have probabilities below 1%. This means, that this unmodified tRNA is very unlikely to occur in the clover-leaf shape (this is consistent with biological knowledge and clearly expresses the need for other mechanisms, such as base modifications, to ensure that the clover-leaf structure is actually achieved).

As a second example, we use the pheS-pheT-Attenuator of E. coli (embl:V00291.1/3682-3754). It is known to switch from a translationally inactive to a translationally active conformation under specific conditions. These two conformations correspond to two valleys in the structure landscape that are separated by a saddle point (energy barrier). In terms of shape analysis, this means that two shapes with reasonable probability should be present. The corresponding experiment delivers the following results (example file `attenuator.fasta`):

```
RNAshapes -q -f attenuator.fasta
> pheS-pheT-Attenuator of E. coli (embl:V00291.1/3682-3754)
        ATCCAGGAGGCTAGCGCGTGAGAAGAGAAACGGAAAACAGCGCCTGAAAGCCTCCCAGTGGAGGCTTTTTTTG
-21.20  ...((((..(((....(((..........)))......))).)))).(((((((((....)))))))))......  0.5381946  [][]
-20.93  .((((((((((.((((((.................)).))))).....)))))))...))))...........  0.3243870  []
-19.33  ...(((((.....((((((.................)).))))...((((((((((....))))))))))))))  0.0975740  [[][]]
-19.73  ..((....))...((((((.................)).))))..((((((((((....))))))))))....  0.0388666  [][][]
-17.30  ...((((..((....))............((........))))))..(((((((((....)))))))).....  0.0008492  [[][]][]
-15.13  ..((....)).(((((((.................)).))))..((((((((((....))))))))))..))  0.0001033  [][[][]]
-14.20  ...(((((.((....))............((.........))..((((((((((....))))))))))))))  0.0000243  [[][][]]
-13.10  ..((....))......(((..........)))......(((...))).(((((((((....)))))))).....  0.0000010  [][][][]
```

The shape with two hairpins has a probability of 0.5381946, whereas the one hairpin shape has a probability of 0.3243870. Shape 1 corresponds to the "off" position of the switch and the higher probability resembles that this is its native position. The "on" position (Shape 2) is less probable, expressing the need for some external effect to trigger the switch.

As stated above, RNAshapes offers several options to control probability calculation, where option **-q** is the computational most expensive one. This is due to the fact, that this analysis performs an analysis of the complete folding space. Although the computational effort depends only on the size of the shape space (which is much smaller than the complete folding space), it requires a substantial amount of computer main memory. In our experience, option **-q** can be used with sequences up to 250 bases on a computer with 2GB main memory. Additionally, this restriction depends not only on the sequence length, but also on the structural properties of the sequence. To get an impression of the expected running time for a certain calculation, we can use the option **-B** that shows a progress bar:

```
RNAshapes -q -B -f attenuator.fasta
```

A slightly more efficient variant of option **-q** above is option **-p**. This performs the same analysis, but without calculation of the corresponding shreps. This option should work with sequences up to a length of around 300 bases (for the presentation, we only use the "short" attenuator sequence here):

```
RNAshapes -p -f attenuator.fasta
> pheS-pheT-Attenuator of E. coli (embl:V00291.1/3682-3754)
ATCCAGGAGGCTAGCGCGTGAGAAGAGAAACGGAAAACAGCGCCTGAAAGCCTCCCAGTGGAGGCTTTTTTTG
0.5381946  [][]
0.3243870  []
0.0975740  [[][]]
0.0388666  [][][]
0.0008492  [[][]][]
0.0001033  [][[][]]
0.0000243  [[][][]]
0.0000010  [][][][]
```

Now we take a look at option **-P**. This option only calculates probabilities for shapes with the lowest free energy shreps. These are often also the shapes of highest probability (but not necessarily so). This mode is implemented as a two-step process. In the first step, the lowest free energy shapes are calculated as in shape representative analysis. Then, for each of these shapes, the probability is calculated individually. The parameter for **-P** specifies the number of lowest free energy shapes for which the probability should be calculated:

```
RNAshapes -P 5 -f attenuator.fasta
> pheS-pheT-Attenuator of E. coli (embl:V00291.1/3682-3754)
        ATCCAGGAGGCTAGCGCGTGAGAAGAGAAACGGAAAACAGCGCCTGAAAGCCTCCCAGTGGAGGCTTTTTTTG
-21.20  ...((((..(((....(((.........))).....))).)))).(((((((....)))))))).....  0.5381930  [][]
-20.93  .((((((((((.((((((.................)).)))).....)))))))...)))).........  0.3243860  []
-19.33  ...(((((.....((((((.................)).)))).)...(((((((((....)))))))))))  0.0975737  [][]]
-19.73  ..((....))...((((((.................)).)))).)..((((((((((....))))))))))....  0.0388665  [][][]
(Only 4 of 5 probabilities calculated. To get more results, increase energy range (-e or -c).)
```

By default, the lowest free energy shapes are calculated for an energy range up to 10% above the mfe. In cases, where this energy range contains fewer shapes than requested with the **-P** option, we have to increase the energy range with options **-e** or **-c**. In our experience, option **-P** works with sequences up to 500 bases.

The last, and most powerful approach for long sequences, is the sampling shapes probability analysis. The sampling shapes approach works in the same manner as Ding and Lawrence's Sfold program. In each step of the recursive backtracing procedure, base pairs and the structural element they belong to are sampled according to their probability, which is obtained from the partition function. For each sample, we calculate its corresponding shape. The shape probability then results from its frequency in the sample space. The sampling shapes approach is activated with option **-i**, together with the desired number of samples:

```
RNAshapes -i 1000 -f attenuator.fasta
... samples ...
Results for 1000 iterations:

-21.20  ...((((..(((....(((.........))).....))).)))).(((((((....)))))))).....  0.5050000  [][]
-20.93  .((((((((((.((((((.................)).)))).....)))))))...)))).........  0.3560000  []
-19.33  ...(((((.....((((((.................)).)))).)...(((((((((....)))))))))))  0.0990000  [][]]
-19.73  ..((....))...((((((.................)).)))).)..((((((((((....))))))))))....  0.0390000  [][][]
-13.60  .........((....)).....((((((..(((.........))))..(((((((....)))))))))))).  0.0010000  [][[][]]
```

Our experiments revealed that 1000 samples are sufficient to achieve reasonable results (confidence value of 95% allowing 10% deviation) for shapes with high probability and that this number is independent of the sequence length (details in [7]). The sampling approach is feasible with an input length of up to 1500 bases.

Finally, we can also calculate the probabilities of individual structures by adding the option **-r** to the program call. This option can be used with any analysis mode of RNAshapes. For example, with our first analysis from above, we receive the following result:

```
RNAshapes -f pharaonis.seq -e 5 -r
                    GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
-35.90  (0.4012148)  ((((((((((((((.((((.....((((((.....))))))).)))))))))))).........))))))).  []
-32.20  (0.0009912)  (((((((.....((.((((.....((((((.....))))))).))))))((.......))).))))))).  [][]]
-31.70  (0.0004404)  ((((((...((((.......)))).((((((.....))))))).....(((((.......))))).))))))  [[][][]]
```

The structure probabilities are printed in the second column. Here, the mfe structure has a probability of 40.12%.

2.3 Consensus shapes analysis

RNAshapes integrates the RNAcast approach for prediction of consensus structures [4]. For a family of RNA sequences, the method independently enumerates the near-optimal shape space, and predicts as the consensus an abstract shape common to all sequences. As an example, we use a family of IRES elements of Picornaviridae viruses (sequence file `ires.fasta`). The consensus shape mode is activated with option **-C**:

```
RNAshapes -C -f ires.fasta
1) Shape: [[][]]  Score: -223.50  Ratio of MFE: 0.99
> EMCBCG
        CCUUUGCAGGCAGCGGAAAUCCCCACCUGGUAACAGGUGCCUCUGCGGCCAAAAGCCACGUGUAUAAGAUACACCUGCAAAGG
-34.10  ((((((((((..(((((......((((((....))))))..))))))(((.....)))..((((.....)))).))))))))))  [[][]]  R = 2
> MNGPOLY
        CCUUUGCAGGCAGCGGAAUCCCCCACCUGGUGACAGGUGCCUCUGCGGCCGAAAGCCACGUGUGUAAGACACACCUGCAAAGG
-39.10  ((((((((((..(((((......((((((....))))))..))))))(((.....)))..((((.....)))).))))))))))  [[][]]  R = 2
> FDI251473
        GCACGCAAGCCGCGGGAACUCCCCCUUGGUAACAAGGACCCGCGGGGCCGAAAGCCACGUUCUCUGAACCUUGCGUGU
-34.10  ((((((((((((((((.......(((((....)))))..)))))))))(((.....)))..((((...))))))))))))  [[][]]  R = 2
> FMDVALF
        GCAUGAUGGCUGUGGGAACUCCCCCUUGGUAACAAGGACCCACGGGGCCAAAAGCCACGUCCUCACGGACCCAUCAUGC
-34.70  ((((((((((((((((.......(((((....)))))..)))))))))(((.....)))..((((....))))))))))))  [[][]]  R = 2
> FAN133359
        GCAUGACGGCCGUGGGAACUCCUCCUUGGUAACAAGGACCCACGGGGCCAAAAGCCACGCCCACACGGGCCCGUCAUGU
-41.90  ((((((((((((((((......((((((....))))))))))))))))(((.....)))..((((....))))))))))  [[][]]  R = 1
> PIFMDV2
        GCAUGUUGGCCGUGGGAACACCUCCUUGGUAACAAGGACCCACGGGGCCGAAAGCCAUGUCCUAACGGACCCAACAUGU
-39.60  ((((((((((((((((......((((((....))))))))))))))))(((.....)))..((((....))))))))))  [[][]]  R = 1
```

The consensus shapes are printed in order of their score. Here, the score is the sum of the shrep energies. The "ratio of MFE"-value specifies the ratio of this score to the sum of the minimum free energies. A ratio near 1.0 means a good conservation, a lower ratio means less conservation. The R-value is the rank of the shape in the shape space of the corresponding sequence.

As in all other analysis modes, the shape type can be controlled with option **-t**. For consensus shape analysis, we generally prefer to work with the less abstract level 3:

```
RNAshapes -C -f ires.fasta -t 3
```

To get more results, we can also increase the energy range with options **-e** and **-c**, for example:

```
RNAshapes -C -f ires.fasta -t 3 -e 10
```

We propose to use the output of the consensus shapes analysis as input for RNAforester [2], a multiple RNA structure alignment program. Use output type **-o** f together with option **-C** to generate suitable input for RNAforester. For example:

```
RNAshapes -C -f ires.fasta -o f | RNAforester -m
```

Note that with output type **-o** f only the result for the first consensus is printed (otherwise RNAforester would not work properly). Use the shape match option **-m** to get alternative results. RNAforester is now part of the Vienna RNA package and can be downloaded at `http://www.tbi.univie.ac.at/~ivo/RNA/`.

2.4 Additional options

Sliding window mode. Apart from consensus shapes, each analysis mode can be used with a sliding window mechanism. Here, the complete input sequence is processed by individual calculations of subsequences of the specified window size.

For example, the file `V00291` contains the E.coli thrS, infC, rplT, pheS, pheT and himA genes (embl:V00291.1) and has a length of 7784 bases. To start the probability analysis with a window size of 73, we call (this will take some time, maybe stop the calculation with `ctrl-c`):

```
RNAshapes -q -f V00291 -w 73
>V00291 V00291.1 12-SEP-2005
        1                                                                       73
        gattcagtttatgctgctgtaaatccgctcgcagtaaacctttcagacgcaccgtgatgttatcagttgttct
-8.80   ...........((((((.((......))..))))))........(((.(((...((((...))))..))))))  0.6994365  [][]
-8.20   (((.((.....((((((.((......))..))))))..........(((...))).))..)))........  0.2448787  [[][]]
-7.50   ((....(((..((((((.((......))..)))))))))...))..........................  0.0425042  []
-5.70   ((......)).((((((.((......))..))))))........(((.(((...((((...))))..))))))  0.0101577  [][][]
-4.90   ((..(((....((((((.((......))..))))))..........(((...))).((....)).))).)).  0.0014580  [[][][]]
```

```
-4.40   ...........((((((.((......)).))))))........((((((...))).((....)).....)))  0.0007761  [][[][]]
-4.30   ......((...((((((.((......)).))))))...((...)).))....((((...))))........  0.0005654  [[][]][]
-3.30   ((.........((((((.((......)).))))))........((..(((...))).((....)).)).)).  0.0001179  [[][[][]]]
-3.60   ((....((...((((((.((......)).))))))...((...)).))....((((...))))).....)).  0.0000690  [[[][]][]]
-2.40   ((......)).((((((.((......)).))))))..........(((...))).((....)).........  0.0000288  [][][][]
-1.30   ((......)).((((((.((......)).))))))........((((((...))).((....)).....)))  0.0000046  [][][[][]]
-1.00   ((..((((....))))((((..........)))).........)).((((...))).((....)).........  0.0000021  [[][]][][]
-1.30   ((..(((....((((((.((......)).))))))...((...)).(((...))).((....)).))).)).  0.0000010  [[][][][]]
```

This is the result for the first "window", the bases 1-73. After this, the window is moved by one base, and the calculation continues with bases 2-74, and so on. Note that these succeeding calculations are faster than the first one, since only a single column of the dynamic programming matrices has to be calculated in each window step. The window step size can be changed with option **-W**.

The result for each window is the same as if we would calculate the corresponding subsequence individually. For example, the window 3682-3754 gives the same result as our pheS-pheT-Attenuator analysis from above, as this is exactly the same sequence.

It would be nice to extend the window approach in a way that the results for the individual windows are merged to a result for the complete sequence. Currently, this is not implemented in RNAshapes.

Suboptimal folding. RNAshapes offers a complete suboptimal folding mode. Its implementation is based on a nonambiguous RNA grammar and handles dangling energies correctly. It is activated with option **-s**:

```
RNAshapes -f pharaonis.seq -s -e 5
        GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
-31.10  (((((((((....(((.((((.....(((((((..)))))))..)))))))((........)).)))))))).  [][]]
-31.10  (((((((((....(((.((((.....(((((((.....)))))))..)))))))((........)).)))))))).  [][]]
-32.20  (((((((((.....((.((((.....(((((((..)))))))..)))))))(((.......))).)))))))).  [][]]
.... (69 results more) ...
```

Shape matching. To see only those structures, that fall into a certain shape, we can use the "shape match" option **-m**. For example, to see all clover-leaf structures in the energy range 5 kcal/mol above the mfe, we call:

```
RNAshapes -f pharaonis.seq -s -e 5 -m '[[][][]]'
        GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUGGGUUCGAAUCCCAGUGGGUCCA
-31.60  (((((((..((((.......)))).(((((((...)))))))).....(((((.......)))))))))))).  [][][]]
-31.60  (((((((..((((.......)))).(((((((.....)))))))).....(((((.......)))))))))))).  [][][]]
-31.70  ((((((...((((.......)))).(((((((...)))))))).....((((.......)))))).)))))).  [][][]]
-31.70  ((((((...((((.......)))).(((((((.....)))))))).....((((.......)))))).)))))).  [][][]]
```

The "shape match" option is also available in all other analysis modes of RNAshapes.

Output control. RNAshapes offers a number of options to control the program output. The first option, **-S**, splits structures into smaller parts. This is especially useful when working with long sequences where the program output can be quite confusing for manual inspection. For example:

```
RNAshapes -f pharaonis.seq -S 50
-35.90  1                                                 50
        GGGCCCAUAGCUCAGUGGUAGAGUGCCUCCUUUGCAAGGAGGAUGCCCUG
        (((((((((((((((.((((.....((((((.....))))))..)))))))
        51                  72
        GGUUCGAAUCCCAGUGGGUCCA
        ))))........)))))))).                               []
```

The option **-O** can be used to "fine-tune" the format of the printed results, for example when we are only interested in parts of the result, or when results of RNAshapes should be used as input for other programs. See Section 3 for a detailed description.

3 Options

The following section gives a complete description of the RNAshapes command line interface.

-h Display command option overview

-H `option` Display detailed information on `option`

This displays the corresponding section of the RNAshapes manual for the given command line option.

-v Show version

This shows the version number of RNAshapes.

3.1 Sequence analysis modes

-a Shape folding (standard mode)

RNA folding based on abstract shapes. This is the standard mode of operation when no other options are given. It calculates the shapes and the corresponding shreps based on free energy minimization. The energy range can be set by **-e** and **-c**. When not specified, the energy range is set to 10% of the minimum free energy.

-s Complete suboptimal folding

Complete suboptimal folding of RNA. This mode uses a non-ambiguous grammar that also handles dangling bases of multiloop components in a non-ambiguous way. The energy range can be set by **-e** and **-c**. When not specified, the energy range is set to 10% of the minimum free energy.

-p Shape probabilities

Shape probability mode. This option calculates the shape probabilities based on partition function. The probability of a shape is the sum of the probabilities of all structures that fall into this shape. On a computer with 2GB main memory, sequences up to a length of 300 bases can be processed with this mode.

-q Shape probabilities (including shreps)

Shape probability mode. Calculates the shape probabilities based on partition function. This is the same as **-p**, and additionally, the corresponding shreps with their minimum free energies are calculated. Note that this mode is slightly slower than **-p** and can be used with sequences up to a length of 250 bases.

-P `value` Shape probabilities for mfe-best shapes

Shape probability mode. This mode first calculates the best `value` shapes based on free energy minimization. In a second step, it calculates the probability for each of these best shapes. This mode has lower memory requirements than modes **-p** and **-q** and can be used for longer sequences (up to 500 bases). The energy range must be specified with **-e** or **-c** in order to get the desired number of results.

-i `value` Sampling with `value` iterations

Probabilistic sampling based on partition function. This mode combines stochastic sampling with a-posteriori shape abstraction. A sample from the structure space holds M structures together with their shapes, on which classification is performed. The probability of a shape can then be approximated by its frequency in the sample.

Sequences up to a length of around 1500 can be handled with this mode. In our experience, 1000 iterations are sufficient to achieve reasonable results for shapes with high probability.

-C Consensus shapes (RNAcast)

For a family of RNA sequences, this method independently enumerates the near-optimal abstract shape space, and predicts as the consensus an abstract shape common to all sequences. For each sequence, it delivers the thermodynamically best structure which has this common shape. Since the shape space is much smaller than the structure space, and identification of common shapes can be done in linear time (in the number of shapes considered), the method is essentially linear in the number of sequences. Input for RNAcast must be provided in multiple fasta format.

We propose to use the output of the consensus shapes analysis as input for RNAforester [2], a multiple RNA structure alignment program. Use output type **-o** f together with option **-C** to generate suitable input for RNAforester. For example:

```
RNAshapes -f test.fasta -C -o f | RNAforester -m
```

Note that with output type **-o** f only the result for the first consensus is printed (otherwise RNAforester would not work properly). Use the shape match option **-m** to get alternative results. RNAforester is now part of the Vienna RNA package and can be downloaded at http://www.tbi.univie.ac.at/~ivo/RNA/.

3.2 Additional modes (use with any of the above)

-r Calculate structure probabilities

This calculates the probability of every computed structure. It can be combined with any sequence analysis mode. Note that this option increases processing time of modes **-a**, **-s** and **-C**.

-w `value` Specify window size

Beginning with position 1 of the input sequence, the analysis is repeatedly processed on subsequences of the specified size. After each calculation, the results are printed out and the window is moved by the window position increment (**-W**), until the end of the input sequence is reached.

-W `value` Specify window position increment (use with **-w**) (default: 1)

This specifies the increment for the window analysis mode (**-w**).

-m `shape` Match shape (use with **-a**, **-s**, **-p**, **-q**, or **-C**)

Specify a shape for the corresponding mode of operation. For example, with options **-p -m** '[]' the probability of shape [] is computed.

3.3 Analysis control

-e `value` Set energy range (kcal/mol)

This sets the energy range for shape folding (**-a**), complete suboptimal folding (**-s**), probability analysis with **-P**, and consensus shapes analysis (**-C**). `value` is the difference to the minimum free energy for the sequence.

-c `value` Set energy range (%) (default: 10)

This sets the energy range as percentage value of the minimum free energy. For example, when **-c** 10 is specified, and the minimum free energy is -10.0 kcal/mol, the energy range is set to -9.0 to -10.0 kcal/mol.

-t `value` Specify shape type (1-5) (default: 5)

The shape type is the level of abstraction or dissimilarity which defines a different shape. In general, helical regions are depicted by a pair of opening and closing square brackets and unpaired regions are represented as a single underscore. The differences of the shape types are due to whether a structural element (bulge loop, internal loop, multiloop, hairpin loop, stacking region and external loop) contributes to the shape representation: Five types are implemented. Their differences are shown in the following example:

```
AUCGGCGCACAGGACAUCCUAGGUACAAGGCCGCCCGUU
..(((.((..(((....))).(((.....)))))))..
```

Type 5: Most abstract - helix nesting pattern and no unpaired regions

```
[[]_[]]
```

Type 4: helix nesting pattern and unpaired regions in external loop and multiloop

```
[[]_[]]
```

Type 3: nesting pattern for all loop types but no unpaired regions

```
[[[]_[]]]
```

Type 2: nesting pattern for all loop types and unpaired regions in external loop and multiloop

```
[_[[]_[]]]
```

Type 1: Most accurate - all loops and all unpaired

```
_[_[_[]_[]]]_
```

-F `value` Set probability cutoff filter (use with **-p**, **-q** or **-P**)

This option sets a barrier for filtering out results with very low probabilities during calculation. The default value here is 0.000001, which gives a significant speedup compared to a disabled filter. Note that this filter can have a slight influence on the overall results. To disable this filter, use option **-F** 0.

-T `value` Set probability output filter (use with **-p**, **-q** or **-P**)

This option sets a filter for omitting low probability results during output. Unlike **-F**, this option does not have any influence on probabilities beyond this value.

-M `value` Set maximum loop length (default: 30) (use -M n for unrestricted)

This option sets the maximum lengths of the considered internal and bulge loops. The default value here is 30. Note that this restriction can have a very slight influence on the calculated structure and shape probabilities. For unrestricted loop lengths, use option **-M** n. This will increase calculation times and memory requirements.

-y `value` Set minimal shape length

This option sets the minimal shape length. Subshapes smaller than `value` are omitted from the analysis.

-l Allow lonely base pairs

In default mode, RNAshapes only considers helices of length 2 or longer. With option **-l**, lonely base pairs are also included.

-u Ignore unstable structures (use with **-a**, **-s** or **-C**)

This option filters out closed structures with positive free energy.

3.4 Input/Output

-o `value` Specify output type (1-4,f) (default: 2)

Specifies the output type. Output type 1 mimics RNAfold and RNAsubopt. Type 2 is the default RNAshapes output. Type 3 is similar to type 2, but without parentheses and with only a single space between results. This output type can be used for exporting results as a comma separated text-file to other applications like Microsoft Excel. Type 4 is a colored variant of type 2. Additional output types can be defined with option **-O**.

In consensus shapes analysis (**-C**), output type f can be used to generate suitable input for RNAforester (a multiple RNA structure alignment program; see **-C** for details).

-O `string` Specify output format string

The option **-O** can be used to "fine-tune" the format of the printed results, for example when we are only interested in parts of the result, or when results of RNAshapes should be used as input for other programs. The syntax is as follows:

```
TYPE{FORMAT}...TYPE{FORMAT}
```

where TYPE specifies the result element:

```
D: structure in dot-bracket notation
S: shape string
E: energy
P: shape probability
R: structure probability (option -r)
C: shape rank (option -C)
V: verbatim output, independent of result element
```

FORMAT is the C-format string that shall be used to print the corresponding result element. Typical C-format strings are %.2f for a floating point number with two decimal places and %s for a string. For example, to print only the structure followed by its energy, we can use -O 'D{%s\t}E{%.2f}V{\n}'. The symbol '\n' performs a line break, the symbol '\t' a tabulator. An ANSI escape sequence can be used with symbol '\e' (see Example 4 below).

The standard output types (option **-o**) are defined as follows:

```
1) 'D{%s  }E{(%.2f)  }R{(%.7f)  }P{%.7f  }S{%s}C{  R = %d}V{\n}'
2) 'E{%-8.2f}R{(%.7f)  }D{%s  }P{%.7f  }S{%s}C{  R = %d}V{\n}'
3) 'E{%.2f }R{%.7f }D{%s }P{%.7f }S{%s}C{ %d}V{\n}'
4) 'E{%-8.2f}R{(%.7f)  }D{\e[1;31m%s\e[0m  }
    P{\e[1;30m%.7f\e[0m  }S{%s}C{  R = %d}V{\n}'
```

-S `value` Specify output width for structures

This splits the structure strings into parts of the specified length. This option is useful when displaying results for long sequences that would otherwise not fit onto the screen.

-# `value` Print only the first `value` results

This option specifies the total number of results to be printed. When this number is reached, the program terminates. Note that this option does not reduce calculation time or memory requirements (except for modes **-s** and **-i**).

-g `value` Generate structure graphs for first `value` structures

This generates postscript structure graphs for the first `value` structures computed for a sequence. If multiple sequences are given, `value` graphs are generated for each sequence.

The filenames of the structure graphs consist of several parts:

1. When the input sequence is given in fasta format, the first 12 characters of the sequence description are taken. White-spaces and special characters are removed. When no description is available, "rna" is chosen as standard name.

2. The sequence position in window mode (option **-w**).

3. The running number of the result.

For example, the first result of a sequence called "xyz" at position 7 in window mode will be saved in file xyz_7_1.ps.

-L Highlight uppercase characters in structure graphs

Used with option **-g**, this generates postscript structure graphs where all uppercase characters in the input sequence are highlighted. This option is useful for marking interesting regions of the input sequence.

-N Do not include additional information in graph output file

In standard operation, the postscript structure graph generation (option **-g**) generates files with shape, energy, and shape probability (if available) included at the bottom. Use this option to suppress this.

-A Omit samples in output

Omit output of samples in sampling mode (**-i**)

-f `file` Read input from `file`

Let RNAshapes load its input data from `file`. `file` can contain a plain single sequence, or multiple sequences in fasta format. When given multiple sequences, each sequence is processed separately in the order of input.

Valid characters in an input sequence are "ACGU" and "acgu". "T" and "t" will be converted to "U". Other letters are mapped to "N" and will not be paired. All other characters are ignored.

-B Show progress bar (use with **-p**, **-q** or **-P**)

Setting this option activates a progress bar. This is useful when experimenting with options **-p** and **-q**, to get an impression of the expected running time.

-z Enable colors (in interactive mode: disable colors)

This option enables colored output. In interactive mode, this is the default setting, so use **-z** to disable colors here.

-Z Enable colors for dotbracket and shape strings

This option colors dotbracket and shape strings in the result output, such that corresponding structural elements have the same color in both representations.

-D `string` Convert dotbracket-string to shape (choose type with **-t**)

Convert a dotbracket-string into a shape. Choose the shape type with option **-t**. The default shape type is 5. For example:

```
RNAshapes -D '.((((.(((....))).))))...(((....))).' -t 4
_[]_[]_
```

-U Start graphical user interface

This option starts the graphical user interface included in the RNAshapes distribution. It requires Java 1.4.2 or later (download from http://java.sun.com/). Note that the RNAshapes distribution for Microsoft Windows includes a slightly different user interface. It does not require Java and additionaly, it offers an interactive visualization of the calculated RNA structures.

3.5 Additional interactive mode commands

:s Show current configuration

This command shows the current settings in an interactive session.

:d Reset configuration

This command sets all settings to their default values.

:e `string` Execute system command

Command **:e** executes a system command. For example, we can use the command **:e** gv rna_1.ps to open a structure graph file created with option **-g** (on a unix machine with gv installed).

:q Quit

This command quits an interactive RNAshapes session.

References

[1] R. Giegerich, B. Voß, and M. Rehmsmeier. Abstract Shapes of RNA. *Nucleic Acids Res.*, 32(16):4843–4851, 2004.

[2] M. Höchsmann, B. Voß, and R. Giegerich. Pure multiple RNA secondary structure alignments: A progressive profile approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):53–62, 2004.

[3] I.L. Hofacker, W. Fontana, P.F. Stadler, L.S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, 125:167–188, 1994.

[4] J. Reeder and R. Giegerich. Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction. *Bioinformatics*, 21(17):3516–3523, 2005.

[5] D. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl Math*, 45(5):810–825, 1985.

[6] P. Steffen, B. Voß, M. Rehmsmeier, J. Reeder, and R. Giegerich. RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 2005. Epub ahead of print.

[7] B. Voß, R. Giegerich, and M. Rehmsmeier. Complete probabilistic analysis of RNA shapes. 2005. Manuscript under review.