Current Protocols in Bioinformatics Identifying structural non-coding RNAs using RNAz

Stefan Washietl, Ivo L. Hofacker
Institute for Theoretical Chemistry and Structural Biology
University of Vienna
Währingerstr. 17
1170 Vienna, AUSTRIA
phone: ++43-1-4277-527-38
fax: ++43-1-4277-527-93
email ivo,wash@tbi.univie.ac.at

Introduction

The program `RNAz` Washietl et al. (2005) can be used to detect functional RNA structures in alignments of homologous nucleotide sequences. Functional RNA structures can be found in many noncoding RNAs as well as cis-acting regulatory elements of mRNAs. `RNAz` predicts functional RNAs on the basis of two key characteristics: (i) Thermodynamic stability and (ii) evolutionary conservation of secondary structures.

The `RNAz` package consists of the `RNAz` core program and a series of helper programs. All programs run under Linux/Unix, OSX and Windows.

Analysis of simple alignments can be carried out using the `RNAz` core program (Basic Protocol 1). For more complex alignments (longer than 400 columns, more than six sequences) some pre-processing steps may be necessary. This can be achieved using the helper programs in combination with the `RNAz` core program (Basic Protocol 2). The `RNAz` package also contains the program `alifoldz.pl`

Washietl and Hofacker (2004a), a predecessor of RNAz. It is sometimes helpful to use this program in addition to RNAz in order to get additional support for a prediction (Alternate Protocol 1). For analyzing a large number of genomic alignments with RNAz (e.g. scanning a complete genome), the helper programs provide a complete analysis pipeline that automatizes all necessary steps (Basic protocol 3).

RNAz and all helper programs are purely command-line driven and come without any graphical user interface. The protocols in this unit describe how to use these programs on the command-line. However, as an alternative the user can choose to use a web-service which allows to carry out most types of analysis on the web (Alternate Protocol 2).

Contributed by Stefan Washietl and Ivo L. Hofacker
University of Vienna
Vienna, Austria

Basic Protocol 1: Using RNAz to analyze a simple alignment

Short alignments (2–6 sequences, not more than 400 columns) can be directly analyzed using the RNAz program on the command-line.

Necessary Resources

Hardware Personal computer (Linux or Windows), Apple Macintosh (OSX) or Unix workstation (e.g. SGI, Sun). CPU and memory requirements of RNAz are moderate and all examples in this unit can be run within reasonable time on a single modern desktop or laptop computer.

Software RNAz 1.x,

Ghostview/GSview or any other program for viewing Postscript images

Files A multiple sequence alignment in `ClustalW` or `MAF` format. An example of a `ClustalW` formatted alignment can be found in Fig. 1. The MAF format is mainly used for genomic screens as described in Basic Protocol 3. The example files used in this protocol are part of the `RNAz` package and are installed to `/usr/local/share/RNAz/examples` (Linux, Unix and OSX) or `C:\Program Files\RNAz\examples` (Windows).

1. Download and install `RNAz` (see Support Protocol).

2. Prepare an alignment of homologous sequences that you want to analyze for a conserved RNA structure. We recommend using `ClustalW`/`ClustalX` (see unit 2.3) for this purpose but you can also use any other sequence alignment program. As an example, we use here the alignment of an IRE (iron responsive element) that can be found in untranslated regions of vertebrate mRNAs. The file name is `IRE.aln` (see Fig. 1).

3. Analyze the alignment using `RNAz`. Open a command prompt, change into the directory where your alignment resides and run the following command:

```
RNAz IRE.aln
```

4. Understanding the output. The results are shown in Fig. 2. The header of the output starts with displaying important characteristics of the alignments (length, number of sequences and mean pairwise identity). For each single sequence in the alignment a secondary structure is predicted using energy minimization algorithms. The mean of the minimum free energies is reported. More negative MFEs mean more stable RNAs. Since the absolute values of the MFEs depend on length and base composition, `RNAz` calculates a normalized z-score from these MFEs. Also here, more negative z-scores mean more stable RNAs. To assess structural conservation, `RNAz` computes a consensus structure and reports a consensus MFE, which is also calculated by the same minimum free energy minimization algorithm, with the additional constraint that all sequences are forced to fold into a common fold. If there exists a common fold, a good consensus MFE is found. Again, the consensus MFE must be normalized to account for length and

base composition in the alignment. A structure conservation index (SCI) is calculated, which roughly ranges from 0 (no conserved structure) to 1 (perfectly conserved structure). Based on z-score and SCI, `RNAz` calculates a combined score, the so-called "RNA class probability" which is referred to also as "P-value". If P> 0.5, `RNAz` classifies an alignment as "RNA", meaning that `RNAz` has detected an unusually stable and/or unusually conserved RNA structure. If $P < 0.5$ `RNAz` classifies the alignment as "other", meaning that there is no detectable evidence for a stable/conserved structure. In the lower part of the output, the structure prediction is explicitly shown in dot/bracket notation. Each pair of brackets "`(`" and "`)`" correspond to a base-pair, while dots "`.`" indicate unpaired regions, see also UNIT 12.2.

5. Analyzing reading direction. If a not annotated DNA sequence alignment is analyzed, there is usually no prior information on which strand a potential RNA is encoded. Therefore both forward and reverse complement need to be analyzed. This is accomplished by the following command using the `--both-strands option`:

```
RNAz --both-strands --predict-strand IRE.aln
```

Due to the near symmetry of RNA base pairing, a functional RNA structure is often predicted on both strands. While the RNA class probability is usually somewhat higher for the correct strand, the difference can be small. `RNAz` therefore includes an additional classification algorithm for strand prediction. This is still an experimental feature and can be invoked using the `--predict-strand` options. In this example, the forward direction is found to be the correct strand with a "strand probability" of 0.96. Note that the P-values are almost indistinguishable (both $>$0.999).

6. Graphical output.

If the `--plot` option is set, `RNAz` generates two graphics in PostScript format: `rna.ps` and `aln.ps` showing a structure annotated alignment and a representation of the predicted consensus structure, respectively.

```
RNAz --plot IRE.aln
```

The color code indicates the number of compensatory/consistent and incompatible mutations (Fig. 3). To view the PostScript files under Linux/Unix run

```
gv rna.ps
gv aln.ps
```

Under OSX/Windows double click on the file icons.

Basic Protocol 2: Analyzing more complex alignments

Long alignments (longer than 400 columns) or alignments with more than six sequences need some pre-processing, before they can be scored with `RNAz`. The helper programs `rnazWindow.pl` and `rnazSelectSeqs.pl` are used for this purpose.

Necessary Resources

Hardware Personal computer (Linux or Windows), Apple Macintosh (OSX) or Unix workstation (e.g. SGI, Sun).

Software RNAz 1.x,

Perl 5.8.x

Files A multiple sequence alignment in `ClustalW` or `MAF` format.

1. Download and install `RNAz` and the helper programs (see Support Protocol).

2a. Scanning long alignments in overlapping windows. `RNAz` analyzes alignments *globally*, i.e. the given alignment is scored as a whole. If the given alignment is long, for example the alignment of a whole chromosome, it is necessary to score such alignments in short overlapping windows. For this purpose the program `rnazWindow.pl` is used:

```
rnazWindow.pl --window=120 --slide=40 unknown.aln \
  | RNAz --both-strands
```

This commands slices the example alignment `unknown.aln` in windows of size 120 using a step size of 40. The output of `rnazWindow.pl` is directly used as input for `RNAz` via the pipe operator "|". Unless you have prior information on the size of the secondary structures to be detected, we recommend slicing alignments longer than 200 columns using a window size of 120. This window size appears large enough to detect local secondary structures within long ncRNAs and, on the other hand, small enough to find short secondary structures without loosing the signal in a much too long window. The program `rnazWindow.pl` not only performs slicing of long alignments but also a series of other pre-processing steps which are described in Basic protocol 3.

2b. Analyzing alignments with more than six sequences. Since `RNAz` is limited to a maximum number of six sequences, a subset of sequences has to be selected in cases where your alignment contains more sequences. Either this subset can be chosen manually or with the help of the program `rnazSelectSeqs.pl`. The following command selects from the example file `miRNA.maf` which contains 12 microRNAs a subset of six sequences:

```
rnazSelectSeqs.pl miRNA.maf | RNAz
```

With default parameters, the program selects six sequences optimized for a mean pairwise identity of 80% in the subset. As before, the output of `rnazSelectSeqs.pl` can be directly piped into `RNAz`.

If there are many sequences in the alignment, it makes sense to analyze more than one subset. The following command samples three different sets with four sequences which are subsequently scored with `RNAz`:

```
rnazSelectSeqs.pl --num-seqs=4 --num-samples=3 miRNA.maf | RNAz
```

Alternate Protocol 1
Using alifoldz.pl

As alternative to RNAz, an alignment can also be analyzed by `alifoldz.pl`, the predecessor of RNAz. Comparing results of both algorithms might be insightful.

Necessary Resources

Hardware Personal computer (Linux or Windows), Apple Macintosh (OSX) or Unix workstation (e.g. SGI, Sun). CPU requirements of `alifoldz.pl` are high. Although a few short alignments can be analyzed within reasonable time on a single modern desktop or laptop computer, more powerful computing facilities (computing cluster) are recommended if many alignments need to be analyzed.

Software RNAz 1.x package (includes `alifoldz.pl`)

Perl 5.8.x

Vienna RNA package 1.6

Files A multiple sequence alignment in `ClustalW` format. The example file `IRE.aln` used in this protocol is part of the RNAz package and is installed to `/usr/local/share/RNAz/examples` (Linux, Unix and OSX) or `C:\Program Files\RNAz\examples` (Windows).

1. Download and install the RNAz package and the Perl programs that include `alifoldz.pl` (see Support Protocol).

2. Run `alifoldz.pl` on the alignment like this

```
alifoldz.pl IRE.aln
```

The output is shown in Fig. 4. `alifoldz.pl` uses a different approach to analyze an alignment for a stable and conserved RNA structure. It folds the alignment using `RNAalifold` and calculates the consensus minimum free energy (MFE) of the alignment. To assess the significance of the consensus MFE it generates 100 random alignments and calculates the mean μ and standard deviation σ of the consensus MFEs of the random samples. The significance of the native MFE m is than expressed as normalized z-score $z = (m - \mu)/\sigma$. Negative z-scores indicate that the native alignment contains a more stable and conserved fold than expected by chance. In practice, z-scores below -3.5 or -4 indicate an RNA signal that could be of interest. Compared to `RNAz`, `alifoldz.pl` is more stringent and not all ncRNAs score below -3.5 or -4 although they can be detected by `RNAz`. However, significant `alifoldz.pl` z-scores can corroborate `RNAz` predictions. z-scores from `alifoldz.pl` must not be confused with the z-scores calculated by `RNAz`. In `RNAz`, z-scores measure the stability of the single sequences which is then combined with an independent conservation score to get the final classification. In `alifoldz.pl`, the z-score is the final score which measures implicitly both stability and conservation. Drawbacks of the `alifoldz.pl` approach are the relatively strong sensitivity to alignment errors, varying results due to the random sampling, and slower performance. The latter two problems are connected: The more samples are used to calculate the background distribution, the more stable is the final z-score. However, large sample numbers require long calculation time. In practice sample number of 100 or 1000 are recommended.

Basic Protocol 3: Using RNAz perform a large scale genomic screen

A pipeline of helper programs is available to simplify and automatize the screening of large numbers of genomic alignments. This protocol describes all parts of this pipeline which is summarized in Fig. 5.

Necessary Resources

Hardware Personal computer (Linux or Windows), Apple Macintosh (OSX) or Unix workstation (e.g. SGI, Sun).

Software RNAz 1.x,

Perl 5.8.x,

Ghostscript,

Vienna RNA package 1.6,

NCBI blast

Files Multiple sequence alignment in MAF format

Optional for annotation:

Annotation file in BED format

Database of known RNA sequences in FASTA format.

See Fig. 6 for examples and explanation of the file formats. The example files used in this protocol can be found in the following packages: `yeast-examples.tar.gz`/`yeast-exa` available on the the Current Protocols web-site:....

1. Obtaining multiple sequence alignment

Generating multiple alignments of long genomic regions is still a subject of heavy research. A few software packages are available for this task: `TBA/Multiz` (`www.bx.psu.edu/miller_lab/`), `MLAGAN` (`lagan.stanford.edu/lagan_web/`), `PECAN` (`http://www.ebi.ac.uk/~bjp/pecan/`). Usage of these programs are not covered in this unit, please refer to the documentation available for these programs. In many cases pre-computed alignments are available from the various sequencing projects.

As example for this protocol, we use alignments of intergenic regions of *Saccharomyces cerevisiae* aligned to 6 other yeast species. They were prepared using `TBA/Multiz` and were downloaded from the UCSC genome browser. It is essential that all alignment blocks in the `MAF` alignment contain a reference sequence which must be always given as the first sequence in each block. We use here the *Saccharomyces cerevisiae* as reference sequence.

2. Pre-processing of raw alignments

Long alignments must be sliced in overlapping windows. In addition to this step, several filtering steps are necessary to get alignments usable for `RNAz`. Genomic

alignments generally are full of gap-rich regions, dubious aligned fragments, or low complexity regions. The slicing and filtering steps are all carried out by a single call of the `rnazWindow.pl` program:

```
rnazWindow.pl --min-seqs=4 input.maf > windows.maf
```

This command slices and filters the input alignment using default parameters. A detailed description is provided in the manual page of the program (call `rnazWindow.pl --man`). In essence, this command slices the alignment in windows of size 120 using a step-size of 40; discards sequences with more than 25% gaps with respect to the reference sequence; discards sequences outside the definition range of `RNAz` (e.g. shorter than 50 nucleotides, GC content $>$75%); chooses a subset of at most six sequences; discards alignments completely if there are fewer than `--min-seqs` sequences left after filtering (in our case 4, default is 2), or the reference sequence has been discarded in a previous step. After this command, the file `windows.maf` contains alignment blocks suitable for scoring with `RNAz`.

3. Running RNAz

The pre-processed alignments are run through `RNAz` with the following commands:

```
RNAz --both-strands --show-gaps --cutoff=0.5 windows.maf > rnaz.out
```

The `RNAz` output for all alignment windows are now stored in the file `rnaz.out`. The `--both-strands` option is used to scan both forward and reverse direction, the `--show-gaps` option displays gaps in the output which is essential for some visualization steps further downstream in the pipeline. The `--cutoff` value is set to 0.5, so that output is only generated for positive predictions.

4a. Collecting and clustering of the results

It is possible that several overlapping windows give positive predictions. The genomic regions of these windows must be extracted from the raw output in `rnaz.out` and combined to a single genomic region which we refer to as *locus* in this context. Note that a locus is simply a region where one or more `RNAz` hits were encountered in either of the two strands. It must not be understood as

RNA gene prediction in the sense of a genetic unit: It is possible that several RNAz loci are predicted for one RNA gene, or on the other hand, that one RNAz locus consists of several short ncRNAs (e.g. microRNA cluster).

```
rnazCluster.pl rnaz.out > results.dat
```

The results are now stored in `results.dat` with tabulator delimited data fields. Each window is assigned a window ID, and all overlapping windows are combined into a locus with a locus ID. For each window and locus the genomic coordinates as well alignment characteristics and RNAz scores are stored. See the manual page for `rnazCluster.pl` for details.

4b. Collecting and clustering of the results with HTML output

If you have Ghostscript and the Vienna RNA package installed, you can easily generate a website that summarizes all results and provides different ways of visualizations. Simply use the `--html` option:

```
rnazCluster.pl --html rnaz.out > results.dat
```

This creates in addition to the `results.dat` file a subdirectory called "`results`" where the HTML files are stored.

5. Generate HTML overview page

The result file `results.dat` with its idiosyncratic format is mainly intended for internal use by other programs downstream in the pipeline. You can convert the results file to a HTML formatted overview page which links to the HTML pages that you created in step 4b:

```
rnazIndex.pl --html results.dat > results/results.html
```

You can now open the file `results.html` with a web-browser and browse through the list of predictions.

6. Export results to annotation files

The results file `results.dat` can also be exported to standard annotation file formats like `GFF` or `BED`:

```
rnazIndex.pl --gff results.dat > results.gff
```

7. Filtering the results

Using the `rnazFilter.pl` program it is possible to filter the predicted loci by different criteria (e.g. z-score, SCI, P-value,...). The following command gives you only loci with a `RNAz` P-value of at least 0.9:

```
rnazFilter.pl "P>0.9" results.dat > highscoring.dat
```

For explanation of the different fields and examples of more sophisticated filter statements refer to the manual (`rnazFilter.pl --man`).

8. Sorting the results

It is also possible to sort the entries in the `results.dat` file by various criteria. To get a list of loci sorted by support from compensatory/consistent mutations one can run the following command:

```
rnazSort.pl combPerPair highscoring.dat > highscoring_sorted.dat
```

It sorts the file `highscoring.dat` by the the "combPerPair" field. This value is the number of different base pair combinations per predicted pair in the consensus structure. Note that `rnazFilter.pl` and `rnazSort.pl` read tab-delimited results files and write tab-delimited results files. Therefore they can be used repeatedly and in different combinations.

9. Compare results with available annotation

Having produced a set of predictions, it is of interest which loci correspond to already annotated regions in a genome. You can compare your predictions with an annotation file in BED format (see Fig. 6):

```
rnazAnnotate.pl --bed ../sgdRNA.bed results.dat > annotated.dat
```

The file `annotated.dat` now contains an additional field which contains the name of the annotation that overlaps a predicted locus.

10. Compare results with database of known RNAs

Another way to annotate predicted loci is to compare the sequence of the loci to a database of known RNAs. Here we compare the predictions to the Rfam database Griffiths-Jones et al. (2005), which is contained in the FASTA formatted file `rfam`. First a blast index is generated:

```
formatdb -t rfam -i rfam -p F
```

Then the following command compares each predicted locus which each sequence in the database:

```
rnazBlast.pl --database rfam --seq-dir=seq \
 --blast-dir=rfam results.dat > annotated.dat
```

You have to specify the directories where your index files reside with the `--blast-dir` option. Also it is necessary that you have the original FASTA formatted sequence files of your reference sequence. Specify the location of these files by the `--seq-dir option`. If a significant homology has been detected (default E-value$< 10^{-6}$)) `rnazBlast.pl` adds the name of the database match to a new field in `results.dat`. At this stage you may want to repeat step 5 with the file `annotated.dat`. This includes the annotation in the HTML overview page.

11. Randomize alignments for a control screen

To get a rough estimate of the false discovery rate in a screen it is useful to repeat the complete procedure on randomized alignments:

```
rnazRandomizeAln.pl input.maf > random-input.maf
```

The program `rnazRandomizeAln.pl` shuffles the position in the alignment. It aims to remove any correlations arising from a secondary structure while preserving important alignment and sequence characteristics like mean pairwise identity or base composition.

12. Gathering basic statistics of the screen

A few additional helper programs are available that allow to gather some basic statistics on a screen. `rnazBEDstats.pl` counts the entries of a `BED` file and calculates the bases covered by the predictions. It assumes that the `BED` file is sorted by coordinates and therefore should be used in conjunction with `rnazBEDsort.pl`:

```
rnazIndex.pl --bed results.dat \
     | rnazBEDsort.pl | rnazBEDstats.pl
```

To get statistics for the input alignments, the program `rnazMAF2BED.pl` can be used, which converts the coordinates of a given species in a `MAF` alignment to `BED` format:

```
rnazMAF2BED.pl --seq-id=sacCer windows.maf \
       | rnazBEDsort.pl | rnazBEDstats.pl
```

Using these commands you can get an idea which fraction of the input regions is predicted as RNA.

Alternate Protocol 2
Using the RNAz web-server

Most of the steps described in Basic Protocols 1, 2 and 3 can also be carried out online without the need to install any programs locally.

Necessary Resources

Hardware Computer connected to the Internet.

Software Web Browser

Files Multiple sequence alignment in one of the following formats: ClustalW, MAF, (multiple or extended multiple) FASTA, PHYLIP, NEXUS.

Optional for annotation:

Annotation file in BED format

1. Open the RNAz web service in your browser: rna.tbi.univie.ac.at

2. Choose an analysis mode: *Standard Analysis* or *Genomic Screen*. In *Standard Analysis* mode you can analyze one or more independent alignments (corresponds to Basic Protocols 1 and 2). In *Genomic Screen* mode you can analyze genomic alignments with a reference sequence (corresponds to Basic Protocol 3).

3. Upload alignment. Paste one or more alignments into the entry field, or upload an alignment file. In *Standard analysis* mode you can choose between ClustalW, MAF, multiple or extended multiple FASTA, PHYLIP or NEXUS format. In *Genomic Screen* modus you need either a MAF or extended multiple FASTA format (XMFA, see XXX) with all necessary information as described (in Fig. 6). Optionally, you can also upload a BED file with annotation information (see Basic Protocol 1, step 9).

4. Set analysis options. At this stage you can customize the way the alignments are pre-processed and analyzed. These options essentially correspond to the options of the rnazWindow.pl program described in Basic protocol 2 and 3. Click on the help icons to get online help on all parameters. The system suggests reasonable defaults depending on your uploaded alignment.

5. Set formatting options. Here you can choose which kind of output should be generated.

6. Submit job. For large alignments we recommend to input your e-mail address to which a notification message is sent upon completion of the job.

7. Examine the results. After you have submitted your job you are redirected to a results page. Here, you can monitor the progress of your job and, upon completion, you can view and download the results.

Support Protocol
Installing necessary software

Hardware Personal computer (Linux or Windows), Apple Macintosh (OSX) or Unix workstation (e.g. SGI, Sun).

Software `C` compiler and necessary build tools, (usually available on Linux/Unix, on OSX make sure that you have installed the "XCode" tools, on Windows no C-compiler is necessary)

`RNAz 1.x`, (available here: `www.tbi.univie.ac.at/~wash/RNAz`, download latest `*.tar.gz` package for Linux/Unix/OSX and latest `*.msi` package for Windows)

`Perl 5.8.x`, (usually installed on Linux/Unix/OSX, for Windows available here: `http://www.activestate.com/store/activeperl/download/`)

`Ghostscript`, (usually available for all Linux distributions in pre-compiled packages, for OSX either try to get a pre-compiled package from `fink.sourceforge.net` or `darwinports.opendarwin.org` or, alternatively, try to compile from source which is available here: `www.ghostscript.com`)

`Ghostview`/`GSview`, (usually available for all Linux distributions in precompiled packages; not necessary for OSX that can display PostScript files without additional software; available for Windows here: `http://www.cs.wisc.edu/~ghost/gsview`)

Vienna RNA package 1.6 (source available as `*.tar.gz` package here: `www.tbi.univie.ac.at/~ivo/RNA`, not necessary for Windows since all re-

quired programs are provided with the RNAz Windows package)

NCBI blast: available here: `ftp://ftp.ncbi.nih.gov/blast`

1. Install RNAz core program. On Linux/Unix/OSX run the following commands:

```
tar -xzf RNAz-1.x.tar.gz
cd RNAz-1.x
./configure
make
su
make install
```

If you do not have root privileges on your system or want to install RNAz to another location for some other reason you can use the `--prefix` and `--datadir` options for configure. This example installs the package into a self contained directory RNAz in the home directory of some user stefan:

```
./configure --prefix=/home/stefan/RNAz \
            --datadir=/home/stefan/RNAz/share
```

On windows simply double click on the file RNAz-1.x-win32.msi and follow the instructions.

2. Installing the RNAz Perl helper programs. If you run the installation in step 1 under Linux/Unix/OSX, all Perl programs are by default installed to `/usr/local/share/RNAz/perl`. To make these programs available on your command-line, either add this directory to your PATH of executables (use `export` or `setenv` depending on the type of your shell) or simply copy all Perl programs to a directory which is already in your PATH of executables. The following should work for the default installation on all systems:

```
cp /usr/local/share/RNAz/perl/* /usr/local/bin
```

Under Windows, the Perl programs are automatically available on your command-line upon installation of the package. No further steps are necessary.

3. Installing the Perl interpreter. Under Linux/Unix/OSX the Perl interpreter which is necessary to run the Perl programs is usually already installed. Under Windows download the latest MSI package for your system and double-click on it. Follow the instructions. You can use default values in all dialogs. Make sure that you have selected the options ”Add Perl to the PATH environment variable” and ”Create Perl file extension association”.

4. Installing Ghostscript. There are pre-compiled packages available for most Linux distributions which are often installed by default. Also for OSX, pre-compiled versions are available through third-party package systems like Fink or DarwinPorts. Alternatively, Ghostscript can also be installed from source following the instructions contained in the *.tar.gz installer package. Under Windows download the self-extracting installer file (named gs854w32-gpl.exe as of writing this text) and run it. You can use default settings. To make the executable available on the command-line, you have to copy it to a directory which is in the PATH of executables. That is the case for the `RNAz` directory which has been created in step 1. So you can run:

```
cd \
copy "Program Files\gs\gs8.54\bin\gswin32c.exe" "Program Files\RNAz\
```

This command copies the file gswin32c.exe to the directory where `RNAz` executable resides and renames it to gs.exe. Adjust the directory names if you chose non-default locations for your installation.

5. Installing Ghostview. Ghostview is usually available as pre-compiled package for all Linux distributions. For OSX no separate PostScript viewer is necessary since the system can internally convert PostScript format to PDF which can be displayed without problems. Under Windows you have to run the self-extracting installer file (named `gsv48w32.exe` as of writing this text). Follow the instruction and make sure that you choose “Associate Postscript files (*.ps, *.eps) with GSview”.

6. Installing the Vienna RNA package. The package can be installed in the same manner as `RNAz` using `./configure` and `make`. Please refer to the documentation of the package or read the detailed instructions given in unit 12.2 of this book.

Windows user do not need to install the package separately since all necessary programs are automatically installed together with the RNAz windows installer.

7. Installing NCBI Blast. Under Linux/Unix/OSX, download the `blast-2.*.tar.gz` file matching your system. Unzip and untar this package. The executables are contained in the subdirectory `bin`. To make the programs available, add this directory to your PATH of executables or copy the executable program files to a directory which is already in your PATH. Under Windows, create a new folder (e.g. c:\Program Files\blast) and download the `blast-2.*-win32.exe` to this folder. Double click on the `blast-2.*-win32.exe` file which extracts the programs and data. Add the `bin` subdirectory to your Path: Right click *My Computer*, then click *Properties*. Select *Advanced/Environment variables/New*. Add the complete path of the blast bin directory to the variable Path, use ";" as separator.

Guidelines for understanding the results

Basic limitations of RNAz

Since RNAz employs machine learning techniques for classification, it is limited to alignments whose properties are not too different from the training set. In general RNAz will produce a warning if the alignment parameters (such as GC content, or sequence identity) fall outside this range.

More importantly, the methods described here miss all ncRNAs that do not depend on a secondary structure in their function. For example, ncRNAs that act solely by antisense interaction cannot be detected. There is also a nice example of a noncoding transcript that regulates a neighboring gene by interfering with its transcription Martens et al. (2004). In this case only the act of transcription is important. There is no constraint on sequence or structure of the transcript itself which is therefore inevitably missed. Similarly, RNAz cannot distinguish between independent "RNA genes" and cis-acting elements of mRNAs or predict boundaries of ncRNA transcripts. All predictions should be regarded as candidates for local RNA structures. Any subsequent interpretation of these results is within the responsibility of the user.

Alignment quality

RNAz is a comparative method and thus relies on the availability of two or more sequences within a useful range of sequence similarity. For sequence similarities above 90% there will be few covariations to support structure conservation, and RNAz will classify alignments mainly on the basis of the z-score leading to somewhat lower accuracy. At low sequence similarity frequent alignment errors make it impossible to predict conserved structures, limiting the sensitivity of RNAz. In our experience, sequence alignment programs such as ClustalW perform reasonably well for alignments with mean-pairwise identity above 60%.

In principle one might try various methods for "structural alignment" for low homology sequences. However, one should keep in mind that RNAz has been trained on pure sequence alignments. Thus, any efforts to structurally enhance an alignment would give artifactually high P-values even for sequences without conserved RNA structure.

The problem of false positives

As of writing this protocol, RNAz is one of the most accurate programs for predicting functional RNA structures. Still, sensible interpretation of the results is necessary to get the most out of the program for any particular application.

For genomic screens, the main problem of RNAz is limited specificity. From randomized alignments we expect appr. 1% false positives for hits with P$>$0.9 and appr. 5% false positives for hits with P$>$0.5. Thus, a large numbers of alignments will translate into a large number of false positives. If the actual number of true ncRNAs is low, the resulting signal to-noise-ratio will be poor. In general it is useful to manually select a set of promising candidates for further analysis.

Selecting good candidates by critical inspection of predictions

Looking critically at the different scores displayed by RNAz as well as the graphical output can help to select good candidates. Often one can weed out obvious false positives easily. Weird gap-patterns, low complexity runs of single letters or short repeats usually are not found in functional RNA structures but often called erroneously by RNAz. In general, it can be insightful to inspect z-score and structure conservation index independently rather than simply relying on the combined

score (P-value).

Average z-scores below -3 or -4 can be considered as "good" stability scores. The significance of structure conservation index (SCI) can only be interpreted if the sequence variation is taken into account. If you have 100% conserved sequences the SCI will be 1 by construction. As a (rough) rule-of-thumb, SCIs around the mean pairwise identity of the alignment are "good" SCI scores. However, the SCI also depends on the number of sequences. On a pairwise alignment with 70% identity, a SCI of say 0.6 does not give any strong indication that there is a conserved fold. However, on an alignment with 6 sequences and mean pairwise identity of 60%, the same SCI of 0.6 can be significant.

In general, the P-value helps in the decision, but there are cases where high P-values only result from a very stable fold while SCI is low. There are examples of real ncRNAs where this is the case but to select highly probable candidates one might first consider those candidates with additional evidence from structural conservation.

For some applications, it might be useful to use the P-value (e.g. cutoff 0.9) only as first rough pre-filter an then explicitly filter by z-score and SCI (e.g using the `rnazFilter.pl` program, see Basic Protocol 3, step 7). Also visual inspection of the mutational pattern in the colored alignment can be an useful additional filter for selecting good candidates. If there are stems with many compensatory/consistent mutations, this is strong evidence that these stems are indeed part of a functional structure. In a genomic screen, the hits can be sorted by the number of compensatory/consistent mutations (see Basic Protocol 3, step 8) making it easier to identify candidates based on this feature.

It is always a good idea in a genomic screen to have a look at the hits found in known ncRNAs. You will realize that structural ncRNAs can differ remarkably with respect to z-score, SCI and compensatory mutations. For example microRNA precursors are extremely stable (z-scores in most cases below -4) but you will rarely find a significant number of compensatory mutations. tRNAs, on the other hand, are generally not very stable (therefore sometimes missed completely by `RNAz`) but strictly structurally conserved. As a result, they usually contain many compensatory mutations (given there is sequence variation in the alignment). Hofacker et al. (2002)

COMMENTARY

Background Information

The challenge of ncRNA prediction

Methods for prediction of noncoding RNAs are still in their infancy, and comprehensive automatic annotation of all ncRNAs in a genome is still out of reach. The main reason is that noncoding RNAs form a heterogeneous class of genes. In particular in complex genomes like those of mammals, noncoding transcription seems to be more abundant and complex than anticipated. There are small ncRNAs processed from introns or transcribed from intergenic regions, long polyadenylated mRNA-like ncRNAs, antisense transcripts, alternatively spliced noncoding transcripts from protein gene loci, or transcribed pseudogenes Frith et al. (2005). The functional relevance of all these transcripts has yet to be explored and it can be safely assumed that we have not seen the full functional spectrum of ncRNAs. However, it is already clear that, from the perspective of gene prediction, ncRNAs are a moving target and there will probably never be one general purpose RNA gene finding tool capable of detecting all sorts of functional ncRNAs.

The RNAz approach

`RNAz` aims at detecting ncRNAs by prediction of functional secondary structures. This strategy seems to be currently the most promising approach for de novo prediction of ncRNAs. Many known ncRNAs are "structural" ncRNAs, i.e. they depend on a defined secondary structure for their function. Structural constraints may derive from their role in ribonucleoprotein complexes as in snRNAs or the signal recognition particle RNA, from particular processing pathways as in the case of microRNA precursors, or other steric requirements as in the case of tRNAs. Some secondary structures are also directly required for the catalytic function of the RNA itself (e.g. RNAaseP RNA, and group I and II introns) Bompfünewerer et al. (2005); The Athanasius F. Bompfünewerer RNA bioinformatics consortium. (2006).

Two criteria are used to detect functional RNA structures are (i) thermodynamic stability and (ii) structure conservation.

In principle, one can fold sequences using `RNAfold` (Hofacker et al. (1994),

UNIT 12.2) and use the minimum free energy (MFE) as measure for thermodynamic stability. However, the absolute values of the MFEs depends on the length and sequence composition. Long GC-rich sequences give lower (i.e. more stable) MFEs than short AU-rich sequences. To get a normalized measure, one usually calculates the background distribution of MFEs of random sequences of the given length and base composition. Let μ and σ be the mean and standard deviation, resp., of the MFEs of a large number of random permutations of a sequence. The significance of the MFE m can then be expressed as z-score $z = (m - \mu)/\sigma$. Negative z-scores indicate that the given sequence is more stable than expected by chance. Unfortunately, to obtain a single meaningful z-score one has to perform at least 1000 folding calculations. To overcome this performance problem, `RNAz` approximates z-scores using a regression approach. It can thus calculate accurate z-scores almost instantaneously.

To measure structural conservation, `RNAz` first calculates a consensus secondary structure using the `RNAalifold` algorithm (Hofacker et al. (2002), Unit 12.2). This algorithm works almost exactly as single sequence folding algorithms (e.g. `RNAfold`), with the main difference that the energy model is augmented by covariance information. Compensatory mutations (e.g. a CG pair mutates to a UA pair) and consistent mutations (e.g. AU mutates to GU) give a "bonus" energy while inconsistent mutations (e.g. CG mutates to CA) yield a penalty. This results in a "consensus MFE". Again it is difficult to assess the significance of this value. It is straightforward to calculate z-scores of this consensus MFE, an approach that is implemented in the program `alifoldz.pl` Washietl and Hofacker (2004b). However this approach poses again the performance problem. Therefore, `RNAz` uses a different way of normalization. `RNAz` compares the consensus MFE to the *average* MFEs of the individual sequences in the alignment and calculates a structure conservation index: SCI $= E_A/\bar{E}$, where E_A and $\bar{E}$ are the consensus MFE and the mean MFEs of the individual sequences, respectively. The SCI will be high if the sequences fold together equally well as if folded individually. On the other hand, SCI will be low if no consensus fold can be found.

`RNAz` has now calculated two independent characteristics of the alignment. Real structural ncRNAs will have low z-scores and high SCIs. To get a final classification, both scores need to be combined to an overall score. `RNAz` uses a "support vector machine" (SVM), i.e. a machine learning technique trained using known ncRNAs, to calculate an "RNA class probability" which efficiently combines both features.

Literature Cited

Bompfünewerer, A., Flamm, C., Fried, C., Fritzsch, G., Hofacker, I., Lehmann, J., Missal, K., Mosig, A., Müller, B., Prohaska, S., Stadler, B., Stadler, P., Tanzer, A., Washietl, S., and Witwer, C. (2005). Evolutionary patterns of non-coding RNAs. *Theor. Biosci.*, 123:301–369.

Frith, M. C., Pheasant, M., and Mattick, J. S. (2005). The amazing complexity of the human transcriptome. *Eur J Hum Genet*, 13(8):894–7.

Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R., and Bateman, A. (2005). Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res*, 33:D121–D124.

Hofacker, I. L., Fekete, M., and Stadler, P. F. (2002). Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, 319:1059–1066.

Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatsh. Chemie*, 125:167–188.

Martens, J. A., Laprade, L., and Winston, F. (2004). Intergenic transcription is required to repress the *Saccheromyces cerevisiae SER3* gene. *Nature*, 429:571–574.

The Athanasius F. Bompfünewerer RNA bioinformatics consortium. (2006). RNAs everywhere: genome-wide annotation of structured RNAs. *J Exp Zoolog B Mol Dev Evol*.

Washietl, S. and Hofacker, I. L. (2004a). Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. *J. Mol. Biol.*, 342:19–30.

Washietl, S. and Hofacker, I. L. (2004b). Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. *J. Mol. Biol.*, 342:19–39.

Washietl, S., Hofacker, I. L., and Stadler, P. F. (2005). Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. USA*, 102:2454–2459.

Key References

Hofacker IL, Fekete M and Stadler PF (2002). Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.* **319**: 1059–1066.

Washietl S and Hofacker IL (2004). Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. *J. Mol. Biol.* **342**: 19–30.

Washietl S, Hofacker IL and Stadler PF (2005). Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. USA* **102**: 2454–2459.

Internet Resources

`http://www.tbi.univie.ac.at/~wash/RNAz/` Download the latest version of `RNAz`, read online manuals.

`http://rna.tbi.univie.ac.at` `RNAz` web-server.

Figures

Figure 1

```
CLUSTAL W (1.83) multiple sequence alignment


Fugu            TAAAAGCATTTCTTTCCAACTTCAGCTACAGTGTTAGCTAAGTTTGGAGGGGAGGAAAAC
Zebrafish       -AAGGTTATTTCTCTCCGACTTCAGCTACAGTGATAGCTAAGTTTGGAGAGGAGAGAAGG
Mouse           TAAGGCTTTGGCTTTCCAACTTCAGCTACAGTGTTAGCTAAGTTTGGAAAGAAGACAAAA
Rat             TAAGGCTTTAGCTTTCCAACTTCAGCTACAGTGTTAGCTAAGTTTGGAAAGAAGACATAA
                 **     *  ** *** *************** **************  * **  *

Fugu            GGGAG
Zebrafish       GAGA-
Mouse           AGAAG
Rat             AGAAG
                   *
```

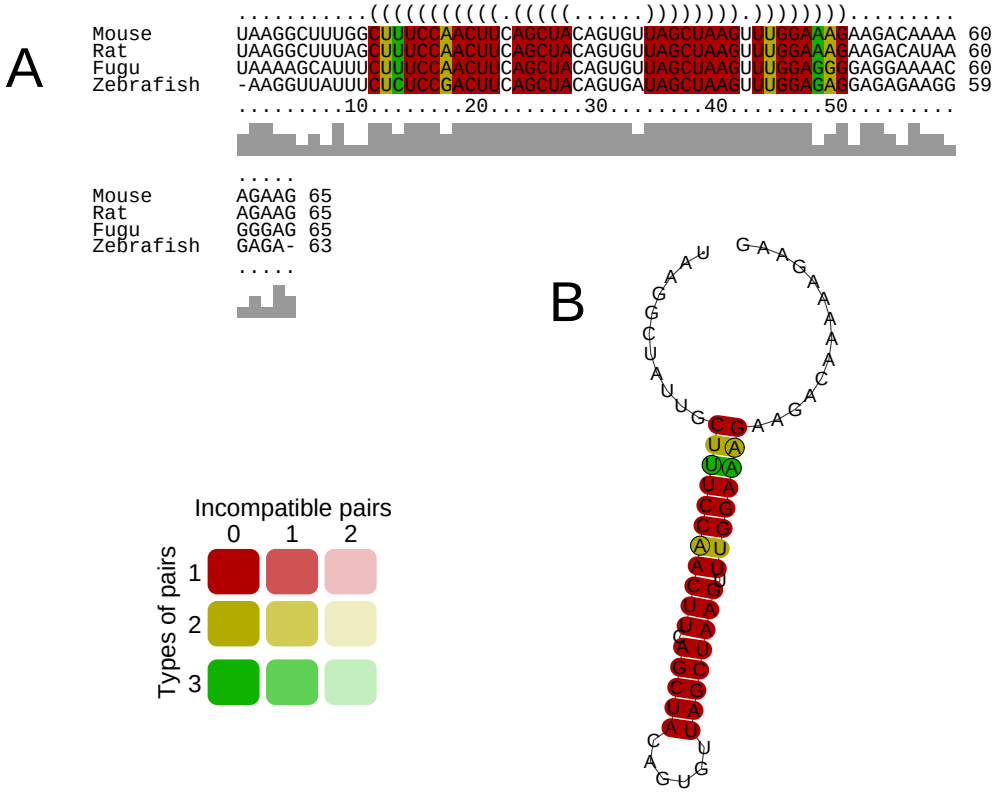
`ClustalW` formatted alignment of an iron responsive element (IRE) conserved in vertebrates. This format is read by `RNAz`. The first line must include the word `CLUSTAL`, the conservation line with asterisks is optional. The block length can be of any size (default:60 columns).

Figure 2

```
############################  RNAz 1.0  ##############################

 Sequences: 4
 Columns: 65
 Reading direction: forward
 Mean pairwise identity:  78.72
 Mean single sequence MFE: -19.23
 Consensus MFE: -17.76
 Energy contribution: -16.95
 Covariance contribution:  -0.81
 Combinations/Pair:   1.25
 Mean z-score:  -3.24
 Structure conservation index:   0.92
 SVM decision value:   3.78
 SVM RNA-class probability: 0.999608
 Prediction: RNA

########################################################################

>Mouse
UAAGGCUUUGGCUUUCCAACUUCAGCUACAGUGUUAGCUAAGUUUGGAAAGAAGACAAAAAGAAG
......((((.(((((((((((.(((((......))))))))).)))))))).....))))..... ( -18.40)
>Rat
UAAGGCUUUAGCUUUCCAACUUCAGCUACAGUGUUAGCUAAGUUUGGAAAGAAGACAUAAAGAAG
.....(((((.(((((((((((.(((((......))))))))).)))))))).....)))))... ( -18.50)
>Fugu
UAAAAGCAUUUCUUUCCAACUUCAGCUACAGUGUUAGCUAAGUUUGGAGGGGAGGAAAACGGGAG
........(((((((((((((.(((((......))))))))).)))))))))).......... ( -18.50)
>Zebrafish
-AAGGUUAUUUCUCUCCGACUUCAGCUACAGUGAUAGCUAAGUUUGGAGAGGAGAGAAGGGAGA-
-.......(((((((((((((.(((((......))))))))).)))))))))).........- ( -21.50)
>consensus
UAAGGCUAUUGCUUUCCAACUUCAGCUACAGUGUUAGCUAAGUUUGGAAAGAAGACAAAAAGAAG
..........((((((((((.(((((......))))))))).))))))).............
                                                  (-17.76 = -16.95 +  -0.81)
```

RNAz output of the alignment shown in Fig. 1. The output consists of two parts: The header shows important characteristics of the input alignment and all scores calculated by RNAz. The lower part shows explicitly the secondary structure predictions for the single sequences and the consensus structure prediction for the alignment.

Figure 3



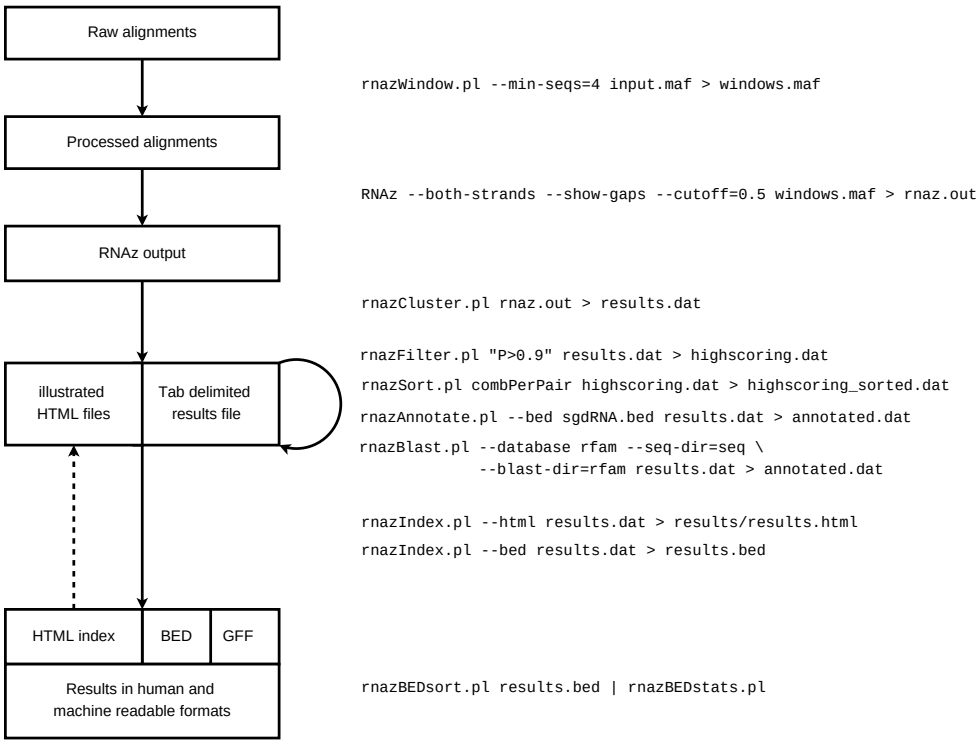
Graphical output of RNAz. (A) Structure annotated alignment. The consensus structure is shown in dot/bracket notation in the first line. The colors of the shadings indicate the number of different types of letter combinations that form a base-pair. Red, ochre, green means that there are 1, 2, 3 different base-pair combinations, respectively. If a base-pair cannot be formed in one or more sequences, the colors are shown faded in different levels (not visible in this example because there are no sequences in the alignment that are incompatible with the consensus fold). (B) RNA secondary structure drawing. A model of the consensus secondary structure is shown. Variable positions are circled (one circle: consistent mutation, two circles: compensatory mutation). The coloring scheme is the same as in (A).

Figure 4

```
#########################################################################
# alifoldz.pl
#
#           Input: 4 sequences of 65 columns
#   Sample Number: 100
#          Window: full sequence
#           Slide: full sequence
#          Strand: forward and reverse
#   MFE threshold: -3
#    Re-alignment: OFF#  Random control: OFF
#    Program call: RNAalifold
#
#########################################################################

  From      To    Strand    Native MFE    Mean MFE     STDV       Z
 ------------------------------------------------------------------------
    1       65       +        -17.76       -5.49       1.93    -6.4
    1       65       -        -16.77       -4.59       2.12    -5.7
```

`alifoldz.pl` output of the alignment shown in Fig. 1. The header shows the program settings used. The complete alignment was scored in both forward and reverse complement direction. A sample size of 100 was used to calculate the z-scores. The table below shows the results of the calculation. The consensus MFE of the native alignment, mean and standard deviation of MFEs of 100 random alignments and the z-score are shown. We observe a significant z-scores of -6.4 in the forward direction. Note that due to the random component of the algorithm your results on the same alignment may differ.

Figure 5



Overview over the analysis pipeline described in Basic Protocol 3.

Figure 6

A

```
##maf version=1
a score=119673.000000
s sacCer1.chr4      1352453 73 - 1531914 GCCTTGTTGGCGCAATCGGTAGCGCGTATGACTC...
s sacBay.contig_465   14962 73 -   57401 GCCTTGTTGGCGCAATCGGTAGCGCGTATGACTC...
s sacKlu.Contig1694     137 73 +    4878 GCCTTGTTGGCGCAATCGGTAGCGCGTATGACTC...
s sacCas.Contig128      258 73 +     663 GCTTCAGTAGCTCAGTCGGAAGAGCGTCAGTCTC...
```

B

```
chr4  1402906 1403030 SNR13
chr4  1461702 1461817 TL(CAA)D
chr4  1492481 1493031 RUF1
chr5  61351   61433   SNR67
chr5  61698   61789   SNR53
chr5  61889   61960   TG(GCC)E
```

C

```
>RF00339;snoR60
UUGCAAAUGAUGAUCUUACUUGGAUUCCCCUUCAGAUCAUUAUGGGUGAUGCAAUCUCCA
AGUUUCUGAUGUA
>RF00247;mir-160
AUGUGCCUGGCUCCCUGUAUGCCACUCAUCUAGAGCAACAAACUUCUGCGAGAGGUUGCC
UAUGAUGGAUGGCGUGCACGGAGCCAAGCAUAU
```

File formats used for genomic screens. (A) Multiple sequence alignment in MAF format. It consists of several alignment blocks. Each block consists of a line starting with "`a score=`" where the alignment score is given. The existence of this line is important for `RNAz` although the value of the score is ignored. The first line is followed by two or more sequence lines starting with `s`. These lines require six fields: (1) a unique identifier of the source sequence, (2) the start position of the aligned subsequence with respect to this source sequence, (3) the length of the aligned subsequence without gaps, (4)"`+`" or "`-`" indicating if the sequence is in the same reading direction of the source sequence or the reverse complement, (5) the sequence length of the complete source sequence, (6) the aligned subsequence with gaps. All fields are required except field 5 which is ignored by `RNAz` and if the correct value is not available it can be filled with arbitrary values. (B) BED annotation file format. This format is used mainly because of its simplicity. In its basic form, it consists of 4 tabulator delimited fields: (1) sequence identifier (2) start coordinates (3) end coordinates, and (4) name of entry. (C) For blast annotation , we use a database of sequences in FASTA format. Each entry starts

with a header line with a leading “>” and the name of the sequence followed by the sequence itself.