



RNA Structure Framework (v1.1.0)

User Manual

Developer/Maintainer: Danny Incarnato
Contact: [danny.incarnato\[at\]hugef-torino.org](mailto:danny.incarnato[at]hugef-torino.org)
[dincarnato\[at\]nextgenintelligence.com](mailto:dincarnato[at]nextgenintelligence.com)

Date released: 15th June 2016

HuGeF – Human Genetics Foundation – Epigenetics Unit
NGI – Next Generation Intelligence

1. Introduction

The recent advent of high-throughput methods for probing RNA secondary structures has enabled transcriptome-scale analysis of the *RNA structurome*. Despite the establishment of several methods for querying RNA secondary structures on a genome-wide scale (CIRS-seq, SHAPE-seq, Structure-seq, DMS-seq, PARS), no tool has been developed to date to enable the rapid analysis and interpretation of these data.

The **RNA Structure Framework** is a modular toolkit developed to deal with RNA structure probing high-throughput data, from reads mapping to structure inference.

Its main features are:

- Automatic reference transcriptome creation
- Automatic reads preprocessing (adapter clipping and trimming) and mapping
- Scoring and data normalization
- Accurate RNA folding prediction by incorporating structural probing data

2. Requirements

- Linux/Mac system
- Bowtie v1.0.0 (<http://bowtie-bio.sourceforge.net/index.shtml>)
- SAMTools v1.2 or greater (<http://www.htslib.org/>)
- BEDTools v2.0 or greater (<https://github.com/arq5x/bedtools2/>)
- Cutadapt v1.10 or greater (<http://cutadapt.readthedocs.io/en/stable/index.html>)
- ViennaRNA Package v2.2.0 or greater (<http://www.tbi.univie.ac.at/RNA/>)
- RNAstructure v5.6 or greater (<http://rna.urmc.rochester.edu/RNAstructure.html>)
- Perl v5.12 (or greater), with ithreads support
- Perl non-CORE modules (<http://search.cpan.org/>):
 - DBD::MySQL
 - LWP::UserAgent
 - RNA (part of the ViennaRNA package)
 - XML::LibXML::Simple

3. List of toolkit components

| CORE modules | |
|------------------|--|
| rsf-index | Automatically queries UCSC genome database and builds the transcriptome Bowtie reference index for the RT Count module |
| rt-count | Performs reads pre-processing and mapping (where needed), and calculates per-base RT-stops and coverage |
| rsf-norm | Performs whole-transcriptome normalization of structure probing data |
| rsf-fold | Produces secondary structures for the analyzed transcripts using structure probing data to guide folding |

| Utilities | |
|--------------------|--|
| rsf-combine | Allows combining SPD files from multiple experiments into a single structure profile |
| rsf-compare | Compares inferred secondary structures with a set of reference structures, computing PPV and sensitivity |
| rsf-silico | Produces SPD files by calculating the partition function folding for a given RNA, and reporting the probability of each base of being unpaired |

4. Usage

4.1. *rsf-index*

The RSF Index tool is designed to automatically generate a Bowtie reference index, that will be used by the RT Count module for reads mapping.

This tool requires an internet connection, since it relies on querying the UCSC Genome database to obtain transcripts annotation and reference genome's sequence.

To list the required parameters, simply type:

```
$ rsf-index -h (or --help)
```

| Parameter | Description |
|--------------------------------|---|
| -o or --output-dir | Bowtie index output directory (Default: <assembly>_<annotation>, e.g. "mm9_refFlat") |
| -ow or --overwrite | Overwrites the output directory if already exists |
| -g or --genome-assembly | Genome assembly for the species of interest (Default: mm9). For a complete list of UCSC available assemblies, please refer to Appendix A, or to the UCSC website (https://genome.ucsc.edu/FAQ/FAQreleases.html) |
| -a or --annotation | Name of the UCSC table containing the genes annotation (Default: refFlat). For a complete list of tables available for the chosen assembly, please refer to the UCSC website (https://genome.ucsc.edu/cgi-bin/hgTables) |
| -n or --gene-name | When possible, gene names will be used instead of gene IDs/accessions |
| -t or --timeout | Connection's timeout in seconds (Default: 180) |
| -r or --reference | Path to a FASTA file containing chromosome (or scaffold) sequences for the chosen genome assembly. Note: if no file is specified, RSF Index will try to obtain sequences from the UCSC DAS server. This process may take up to hours, depending on your connection's speed. |
| -b or --bowtie-build | Path to bowtie-build executable (Default: assumes bowtie-build is in PATH) |
| -e or --bedtools | Path to BEDTools executable (Default: assumes BEDTools is in PATH) |

Note: For RNA structure probing experiments conducted over synthetic RNAs (or custom pools of RNAs), a reference can be generated by invoking directly the *bowtie-build* command, however it is necessary to first sort lexicographically the FASTA file by sequence IDs:

```
$ awk 'BEGIN{RS=">"} NR>1 {gsub("\n", "\t"); print ">"$0}' reference_unsorted.fa | \
LC_ALL=C sort -T "" -t ' ' -k2,2 | awk '{sub("\t", "\n"); gsub("\t", ""); print $0}' > reference_sorted.fa
```

```
$ bowtie-build reference_sorted.fa reference_sorted
```

```
$ ls -l
```

```
-rwxrwxrwx 1 danny epigenetics 96041105 5 mar 10.50 reference_sorted.1.ebwt
-rwxrwxrwx 1 danny epigenetics 37313744 5 mar 10.50 reference_sorted.2.ebwt
-rwxrwxrwx 1 danny epigenetics 1844468 5 mar 10.28 reference_sorted.3.ebwt
-rwxrwxrwx 1 danny epigenetics 74627475 5 mar 10.28 reference_sorted.4.ebwt
-rwxrwxrwx 1 danny epigenetics 302198817 5 mar 10.28 reference_sorted.fa
-rwxrwxrwx 1 danny epigenetics 96041105 5 mar 11.11 reference_sorted.rev.1.ebwt
-rwxrwxrwx 1 danny epigenetics 37313744 5 mar 11.11 reference_sorted.rev.2.ebwt
-rwxrwxrwx 1 danny epigenetics 302198817 5 mar 10.28 reference_unsorted.fa
```

4.2. *rt-count*

The RT Count module is the core component of the toolkit. It can process any number of both FastQ or SAM/BAM files, also mixed. In case FastQ files are passed, reads are pre-processed (trimming and clipping), and mapped to the reference transcriptome. Each SAM/BAM file is then processed to calculate per-base RT-stops and reads coverage on each transcript.

To list the required parameters, simply type:

```
$ rt-count -h (or --help)
```

| Parameter | Description |
|---------------------------------|---|
| -p or --processors | Number of processors (threads) to use (Default: 1) |
| -wt or --working-threads | Number of working threads to use for each instance of SAMTools/Bowtie (Default: 1). Note: RT Counter executes 1 instance of SAMTools/Bowtie for each processor specified by -p. At least -p <processors> * -wt <threads> processors are required. |
| -t or --tmp-dir | Path to a directory for temporary files creation (Default: /tmp) Note: If the provided directory does not exist, it will be created |
| -o or --output-dir | Output directory for writing counts in RTC (RT Count) format (Default: rt_count/) |
| -ow or --overwrite | Overwrites the output directory if already exists |
| -k or --keep | Keeps SAM/BAM files after reads mapping (in case FastQ files are passed) Note: If unsorted SAM/BAM files are passed, this option will cause RT Count to keep the sorted SAM/BAM file |
| -n or --no-bam | Disables conversion of SAM files to BAM format (requires -k) |
| -b or --bowtie | Path to Bowtie v1 executable (Default: assumes Bowtie is in PATH) |
| -c or --cutadapt | Path to Cutadapt executable (Default: assumes Cutadapt is in PATH) |
| -s or --samtools | Path to SAMTools executable (Default: assumes SAMTools is in PATH) |
| -r or --sorted | In case SAM/BAM files are passed, assumes that they are already sorted lexicographically by transcript ID, and numerically by position |

| | |
|-----------------------------|---|
| -t5 or --trim-5prime | <p>In case SAM/BAM files are passed, allows to specify a comma separated list (no spaces) of values indicating the number of bases trimmed from the 5'-end of reads in the respective sample SAM/BAM files (Default: 0)</p> <p>Note #1: Values must be provided in the same order as the input files (e.g. <i>rt-count -t5 0,5 file1.bam file2.bam</i>, will consider 0 bases trimmed from <i>file1</i> reads, and 5 bases trimmed from <i>file2</i> reads).</p> <p>Note #2: If a single value is specified along with multiple SAM/BAM files, it will be used for all files.</p> |
| -f or --fasta | <p>Path to a FASTA file containing the reference transcripts</p> <p>Note #1: Transcripts in this file must match transcripts in SAM/BAM file headers.</p> <p>Note #2: This can be omitted if a Bowtie index is specified by -bi (or --bowtie-index)</p> |

| FASTX Clipper options | |
|------------------------------------|--|
| -ca or --cutadapt-adapter | Sequence of 3' adapter for clipping (Default: TGGAATTCTCGGGTGCCAAGG, Illumina TruSeq Small RNA 3' Adapter) |
| -cl or --cutadapt-len | Minimum length to keep reads after clipping (≥ 10 , Default: 25) |
| -cm or --cutadapt-min-align | Minimum alignment in nt to adapter's sequence (> 0 , Default: 1) |
| -cp or --clipped | Assumes that reads have been already clipped |

| Bowtie options | |
|---------------------------------|---|
| -bn or --bowtie-n | Use Bowtie mapper in -n mode (0-3 mismatches, Default: 2) |
| -bv or --bowtie-v | Use Bowtie mapper in -v mode (0-3 mismatches, Default: disabled). If both --bowtie-v and --bowtie-n parameters are passed, the -n mode will be overridden by the -v mode. |
| -bm or --bowtie-max | Discard alignment if more than this number of alignments exist (Default: 1) |
| -bk or --bowtie-multimap | Report up to this number of equally scoring positions for multi-mapping reads (Default: 1) |
| -ba or --bowtie-all | Report all equally scoring positions for multi-mapping reads (Default: disabled, reports only uniquely mapped reads) |

| | |
|--|---|
| -bc <i>or</i> --bowtie-chunkmbs | Maximum MB of RAM for best-first search frames (Default: 128) |
| -bi <i>or</i> --bowtie-index | Path to transcriptome reference index (see paragraph 4.1) |

4.2.1. RTC format

RT Count produces a RTC (RT Count) file for each analyzed sample. RTC files are proprietary binary files, that store transcript's sequence, per-base RT-stop counts, and per-base reads coverage. These files can be indexed for fast random access.

Each entry in a RTC file is structured as follows:

| Field | Description | Type |
|--------------------------|--|-------------------------|
| len_transcript_id | Length of the transcript ID (plus 1, including NULL) | uint32_t |
| transcript_id | Transcript ID (NULL terminated) | char[len_transcript_id] |
| len_seq | Length of sequence | uint32_t |
| seq | 4-bit encoded sequence: 'ACGTN' -> [0,4] (High nybble first) | uint8_t[(len_seq+1)/2] |
| stops | RT-stops at each base of transcript | uint32_t[len_seq] |
| cov | Coverage at each base of transcript | uint32_t[len_seq] |

The RTC file EOF marker (last 8 bytes of file) is “\x5b\x65\x6f\x66\x72\x74\x63\x5d”.
If the marker is absent, then the file is truncated or corrupted.

RTI index files are binary files, structured as follows:

| Field | Description | Type |
|--------------------------|--|-------------------------|
| len_transcript_id | Length of the transcript ID (plus 1, including NULL) | uint32_t |
| transcript_id | Transcript ID (NULL terminated) | char[len_transcript_id] |
| offset | Offset position of transcript in the RTC file | uint32_t |

4.3. rsf-norm

The RSF Norm tool takes one (Rouskin method), or two (Ding method) RTC files generated by the RT Count module, and performs normalization to obtain a per-base reactivity score for each transcript. Reactivity scores can be computed using two methods:

[1] Ding *et al.*, 2014

In this scoring approach, the signal per-base is calculated as the natural log (ln) of the ratio between the raw count of RT-stops/Nuclease cuts at a given position of a transcript, and the average of the ln of RT-stops/Nuclease cuts along the whole transcript's length:

$$U_i = \frac{\ln(n_{1i}+p)}{\left(\sum_{j=0}^l \frac{\ln(n_{1j}+p)}{l}\right)} \quad \text{and} \quad T_i = \frac{\ln(n_{2i}+p)}{\left(\sum_{j=0}^l \frac{\ln(n_{2j}+p)}{l}\right)}$$

where n_{1i} and n_{2i} are respectively the raw read counts in the untreated (or RNase V1) and treated (DMS, CMCT, SHAPE, or Nuclease S1) experiments at position i of the transcript, l is transcript's length, and p is a pseudocount added to deal with non-covered regions. U_i and T_i are respectively the normalized number of RT-stops at position i in the untreated and treated samples.

Score at position i is then calculated as:

$$S_i = \max(0, (T_i - U_i))$$

Note: Since version 1.1.0, RSF Norm allows Ding scoring scheme to be applied within sliding windows

[2] Rouskin *et al.*, 2014

In this scoring approach, the untreated sample is not considered. Signal per-base is calculated within fixed size windows, by dividing the number of RT-stops on each residue, by the number of RT-stops on the most reactive residue within the same window after removing the outliers.

Once computed, reactivity scores are normalized. Three normalization methods are actually provided:

| Parameter | Description |
|-------------------------------|--|
| 2-8% Normalization | From the top 10% of values, the top 2% is ignored, then any reactivity value along the entire transcript is divided by the average of the remaining 8% |
| 90% Winsorising | Each reactivity value above the 95th percentile is set to the 95th percentile, and the reactivity at each position of the transcript is divided by the 95th percentile |
| Box-plot Normalization | Values greater than 1.5x the interquartile range (numerical distance between the 25th and 75th percentiles) above the 75th percentile are removed. After excluding these outliers, the next 10% of reactivities are averaged, and all reactivities (including outliers) are divided by this value. |

Since Box-plot normalization will return values ranging from 0 to 1.5-2.2, normalized reactivities can be further remapped to values ranging from 0 to 1 according to Zarrinhalam *et al.*, 2012.

In this approach, values < 0.25 are linearly mapped to $[0-0.35]$, values ≥ 0.25 and < 0.3 are linearly mapped to $[0.35-0.55]$, values ≥ 0.3 and < 0.7 are linearly mapped to $[0.55-0.85]$, and values ≥ 0.7 are linearly mapped to $[0.85-1]$.

To list the parameters required to the RSF Norm tool, simply type:

\$ *rsf-norm -h* (or *--help*)

| Parameter | Description |
|----------------------------------|--|
| -u or --untreated | Path to the RTC file for the non-treated (or RNase V1) sample |
| -t or --treated | Path to the RTC file for the treated (DMS, SHAPE, or Nuclease S1) sample |
| -i or --index | <p>A comma separated (no spaces) list of RTI index files for the provided RTC files</p> <p>Note #1: RTI files must be provided in the order 1. Untreated, 2. Treated</p> <p>Note #2: If a single RTI file is specified along with both untreated and treated samples, it will be used for both samples</p> <p>Note #3: If no RTI index is provided, it will be generated at runtime, and stored in the same folder of the untreated/treated samples</p> |
| -p or --processors | Number of processors (threads) to use (Default: 1) |
| -o or --output-dir | Output directory for writing normalized data in SPD (Structure Probing Data file) format (Default: <treated>_vs_<untreated>_norm/ for Ding method, <treated>_norm/ for Rouskin method) |
| -ow or --overwrite | Overwrites the output directory if already exists |
| -c or --config-file | <p>Path to a configuration file with normalization parameters (see paragraph 4.2.1)</p> <p>Note #1: If the provided file exists, the loaded configuration will override any command-line specified parameter</p> <p>Note #2: If the provided file doesn't exist, it will be generated using the command-line specified (or the default) parameters</p> |
| -sm or --scoring-method | Score calculation method (1-2, Default: 1), where: 1. Ding <i>et al.</i> , 2014; 2. Rouskin <i>et al.</i> , 2014 |
| -nm or --norm-method | Score normalization method (1-3, Default: 1), where: 1. 2-8% normalization; 2. 90% Winsorising; 3. Box-plot normalization |
| -rb or --reactive-bases | <p>Reactive bases to consider for signal normalization (Default: N [ACGT])</p> <p>Note: This parameter accepts any IUPAC code, or combinations of them (e.g. <i>-rb M</i>, or <i>-rb AC</i>). Reactivity for any other base will be reported as NaN.</p> |
| -ni or --norm-independent | Each one of the reactive bases will be normalized independently (e.g. <i>-rb AC -ni</i> will normalize independently A and C residues) |

| | |
|---------------------------------|---|
| -mc or --mean-coverage | Discards any transcript with mean coverage below this threshold (≥ 0 , Default: 1) |
| -ec or --median-coverage | Discards any transcript with median coverage below this threshold (≥ 0 , Default: 1) |
| -nw or --norm-window | Window size (in nt) for signal normalization (≥ 3 , Default: whole transcript [Ding], 50 [Rouskin]) |
| -wo or --window-offset | Offset for sliding window during normalization (Default: none [Ding], 50 (non-overlapping windows) [Rouskin]) |
| -d or --decimals | Number of decimals for reporting reactivities (1-10, Default: 3) |
| -n or --nan | Non-covered transcript positions will be reported as NaN in the reactivity profile |

Scoring method #1 options (Ding *et al.*, 2014)

| | |
|-----------------------------|--|
| -pc or --pseudocount | Pseudocount added to reactivities to avoid division by 0 (> 0 , Default: 1) |
| -s or --max-score | Score threshold for capping raw reactivities (> 1 , Default: 10) |

Normalization method #3 options (Box-plot normalization)

| | |
|------------------------------------|---|
| -rm or --remap-reactivities | Remaps reactivities to values ranging from 0 to 1 according to Zarrinhalam <i>et al.</i> , 2012 |
|------------------------------------|---|

4.3.1. Configuration files

RSF Norm configuration files are used to provide normalization parameters for the analysis, without the need to manually specify them from the command-line.

Configuration files are composed of a list of key/value pairs, separated by the equal sign (=), or by the colon punctuation mark (:). Keys and values are *case-insensitive*.

Accepted key/value pairs are:

| Parameter | Accepted values | Default value |
|------------------------|--|----------------------------|
| scoreMethod | "Ding" (or 1); "Rouskin" (or 2) | Ding |
| normMethod | "2-8%" (or 1); "90% Winsorising" (or 2); "Box-plot" (or 3) | 2-8% |
| reactiveBases | [ACGTURYSWKMBDHAVN] (or "all") | all |
| normIndependent | TRUE/FALSE; Yes/No; 1/0 | FALSE |
| normWindow | Positive integer ≥ 3 | 1e9 [Ding] 50 [Rouskin] |
| windowOffset | Positive integer ≥ 0 | 1e9 [Ding] 50 [Rouskin] |
| meanCoverage | Positive integer ≥ 0 | 1 |
| medianCoverage | Positive integer ≥ 0 | 1 |

| Scoring method #1 parameters (Ding <i>et al.</i> , 2014) | | |
|--|---------------------------|----|
| maxScore | Positive integer ≥ 1 | 10 |
| pseudoCount | Positive integer ≥ 1 | 1 |

| Normalization method #3 parameters (Box-plot normalization) | | |
|---|-------------------------|-------|
| remapReactivities | TRUE/FALSE; Yes/No; 1/0 | FALSE |

e.g. A typical configuration file

```
scoreMethod=Ding
normMethod=2-8%
maxScore=50
pseudoCount=1
reactiveBases=ACGT
normIndependent=no
normWindow=1e9
windowOffset=1e9
meanCoverage=1
```

4.3.2. Structure Probing Data (SPD) files

RSF Norm produces a set of SPD files, one for each transcript being analyzed. These files are essentially XML files, and therefore can be parsed using any standard XML parsing library. SPD files tree structure is the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<data [attributes]>
  <transcript id="Transcript ID" length="Transcript length">
    <sequence>
      Transcript sequence
    </sequence>
    <reactivity>
      Comma-separated list of reactivity values
    </reactivity>
    <error>
      Comma-separated list of standard deviations from multiple experiments
    </error>
  </transcript>
</data>
```

The “error” tag is optional. It is introduced by RSF Combine when multiple experiments are combined into a single SPD profile, and stores the per-base standard deviation of the normalized reactivity.

The “data” tag’s attributes allow keeping track of the analysis performed to obtain the SPD file:

| Attribute | Description |
|-----------------|--|
| combined | Whether multiple experiments have been combined into a single profile (TRUE or FALSE, see paragraph 5.1) |
| scoring | Scoring method (Ding or Rouskin) |
| norm | Normalization method (2-8%, Winsorising 90%, or Box-plot) |
| reactive | Reactive bases |
| win | Normalization window’s size (in nt) |
| offset | Offset for normalization window sliding |

| Scoring method #1 attributes (Ding <i>et al.</i> , 2014) | |
|--|--|
| max | Score threshold for reactivity capping |
| pseudo | Pseudocount added during score calculation |

| Normalization method #3 attributes (Box-plot normalization) | |
|---|--|
| remap | Whether box-plot normalized reactivities have been remapped according to Zarringhalam <i>et al.</i> , 2012 |

4.3. rsf-fold

The RSF Fold tool is designed to allow the transcriptome-wide reconstruction of RNA structures, starting from SPD files generated using the RSF Norm tool.

This tool can process a single, or an entire directory of SPD files, and produces the inferred secondary structures (either in dot-bracket notation, or CT format) and their graphical representation (either in Postscript, or SVG format).

To allow higher analysis flexibility, the tool incorporates two different prediction methods:

[1] ViennaRNA

[2] RNAstructure

Note #1: Only the new v2.2.0 ViennaRNA soft-constraint approach is provided, since hard-constraint predictions are in most cases totally unreliable.

Note #2: The Iterative Cluster Refinement method (Incarnato *et al.*, 2016) has been temporary removed due to a change in the ViennaRNA APIs, and will be added back in a future release.

To list the parameters required to the RSF Fold tool, simply type:

```
$ rsf-fold -h (or --help)
```

| Parameter | Description |
|------------------------------------|---|
| -o or --output-dir | Output directory for writing structural data (Default: structurome/) |
| -ow or --overwrite | Overwrites output directory (if the specified path already exists) |
| -ct or --connectivity-table | Writes predicted structures in CT format (Default: Dot-bracket notation) |
| -m or --folding-method | Specifies the folding method (1-2, Default: 1): 1. ViennaRNA; 2. RNAstructure |
| -p or --processors | Number of processors to use for the analysis (Default: 1) |
| -g or --img | Enables generation of structure representations (Default: Postscript format) |
| -s or --svg | Structure representations are generated in SVG format (requires -g) |
| -sl or --slope | Sets slope used with structure probing data restraints (Default: 1.8 [kcal/mol]) |
| -in or --intercept | Sets intercept used with structure probing data restraints (Default: -0.6 [kcal/mol]) |
| -m or --maximum-distance | Sets the maximum pairing distance in nucleotides between transcript's residues (Default: 0, [no limit]) |

| Folding method #1 options (ViennaRNA) | |
|---------------------------------------|--|
| -v or --viennarna | Path to ViennaRNA RNAfold executable (Default: assumes RNAfold is in PATH) |
| -nlp or --no-lonely-pairs | Disallows lonely (unstacked) base-pairs inside predicted structure |
| -ngu or --no-closing-gu | Disallows G:U wobbles at the end of helices |
| -cm or --constraint-method | Method for converting SPD reactivities into pseudo-energies (1-2, Default: 1): 1. Zarringhalam <i>et al.</i> , 2012; 2. Deigan <i>et al.</i> , 2009 |

| Zarringhalam <i>et al.</i> , 2012 method options | |
|--|---|
| -cc or --constraint-conversion | Method for converting SPD reactivities to pairing probabilities (1-5, Default: 1): 1. Skip normalization step (SPD reactivities are treated as pairing probabilities) 2. Linear mapping according to Zarringhalam <i>et al.</i> , 2012 3. Use a cutoff to divide into paired and unpaired nucleotides 4. Linear model for converting SPD reactivities into probabilities of being unpaired 5. Linear model for converting the logarithm of SPD reactivities into probabilities of being unpaired |
| -bf or --beta-factor | Sets the magnitude of penalties for deviations from the observed pairing probabilities (Default: 0.5) |
| -f or --cutoff | Cutoff for constraining a position as unpaired (0-1, Default: 0.7) Note: This option requires constraint conversion method #3 |
| -ms or --model-slope | Sets the slope used by the linear model (Default: 0.68 [Method #4], or 1.6 [Method #5]) Note: This option requires constraint conversion methods #4 or #5 |
| -mi or --model-intercept | Sets the intercept used by the linear model (Default: 0.2 [Method #4], or -2.29 [Method #5]) Note: This option requires constraint conversion methods #4 or #5 |

| Folding method #2 options (RNAstructure) | |
|--|--|
| -r or --rnastructure | Path to RNAstructure Fold executable (Default: assumes RNAstructure is in PATH) |
| -dp or --data-path | Path to RNAstructure data tables (Default: assumes DATAPATH environment variable is already set) |

Note: ViennaRNA constraint method #2 (Deigan *et al.*, 2009) is essentially the same employed by RNAstructure, therefore the two approaches should yield approximately the same results.

For additional details relatively to ViennaRNA soft-constraint prediction methods, please refer to the ViennaRNA manual, or to Lorenz *et al.*, 2016.

5. Utilities

5.1. *rsf-combine*

RSF Combine allows combining multiple experiments into a single reactivity profile. This is useful for example when performing CIRS-seq experiments, to combine into a single profile both the reactivity of A/C residues probed with DMS, and of G/U residues probed with CMCT. Alternatively, RSF Combine is able to combine into a single profile multiple replicates of the same probing experiment. In these cases, the resulting SPD file may contain the optional “error” tag, in which the per-base standard deviation of the reactivity is reported.

When combining SPD files generated using the *-n* (or *--nan*) option of RSF Norm, only positions covered in all experiments will be combined, while the others will be reported as NaN.

There’s no limit to the number of experiments that RSF Combine can handle. Moreover, it can be used both on individual SPD files, or on whole SPD folders generated by RSF Norm.

Note: RSF Combine does not allow combining SPD files generated using different scoring/normalization methods, since this will produce inconsistent data

To list the parameters required to the RSF Combine tool, simply type:

\$ *rsf-combine -h* (or *--help*)

| Parameter | Description |
|---------------------------|--|
| -p or --processors | Number of processors (threads) to use (Default: 1) |
| -o or --output-dir | Output directory for writing combined SPD (Structure Probing Data) files (Default: combined/) |
| -ow or --overwrite | Overwrites the output directory if already exists |
| -s or --stdev | When multiple replicates are combined, an optional “error” tag will be reported within the output SPD files, containing the per-base reactivity’s standard deviation |
| -d or --decimals | Number of decimals for reporting reactivities (1-10, Default: 3) |

5.2. *rsf-compare*

RSF Compare allows comparing inferred secondary structures from RSF Fold, with a reference of known secondary structures, reporting for each comparison the PPV (Positive Predictive Value, the fraction of base-pairs present in the predicted structure that are also present in the reference structure) and the sensitivity (the fraction of base-pairs present in the reference structure that are also in the predicted structure). Reference structures must be provided in Vienna format:

```
>Transcript_1
AAAAAAAAAAAAAAAAUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU
.((((((((((((((((((((.....))))))))))))))))))
>Transcript_2
CCCCCCCCCCCCCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGG
((((((((((((((((((((.....))))))))))))))))))

-- cut --

>Transcript_n
GCUAGCUAGCUAGCUAGCUAGCUAGCUAGCUAGCUAGCUAGCUAGCU
((((((((((((((((((((.....)))))))))))))).....
```

RSF Compare can be invoked both on a single structure, or on an entire folder of RSF Fold predicted structure files. Structures can be provided either in CT or Vienna (dot-bracket) format.

To list the parameters required to the RSF Combine tool, simply type:

```
$ rsf-compare -h (or --help)
```

| Parameter | Description |
|--------------------------|---|
| -r or --reference | A file containing reference structures in Vienna format (dot-bracket) |

5.3. *rsf-silico*

RSF Silico calculates partition function folding for a set of given RNAs, using either ViennaRNA, RNAstructure, or their combination. The probability of each base of being unpaired is then reported in the form of a SPD file. To keep compatibility with RSF Norm, it is possible to specify which bases should be excluded from the analysis. Those bases are reported as NaN in the resulting SPD file.

To list the parameters required to the RSF Silico tool, simply type:

\$ *rsf-silico -h* (or *--help*)

| Parameter | Description |
|----------------------------------|---|
| -p or --processors | Number of processors (threads) to use (Default: 1) |
| -o or --output-dir | Output directory for writing combined SPD (Structure Probing Data) files (Default: combined/) |
| -ow or --overwrite | Overwrites the output directory if already exists |
| -t or --tmp-dir | Path to a directory for temporary files creation (Default: /tmp) Note: If the provided directory does not exist, it will be created |
| -m or --method | Partition function calculation method (1-3, Default: 1), where: 1. ViennaRNA; 2. RNAstructure; 3. Combined Note: Method #3 (Combined) calculates base-pair probabilities using both ViennaRNA and RNAstructure, and produces a SPD file containing the per-base average of the two methods |
| -e or --temperature | Temperature in Celsius degrees (Default: 37) |
| -md or --maximum-distance | Sets the maximum pairing distance in nucleotides between transcript's residues (Default: 0 [No limit]) |
| -v or --viennarna | Path to ViennaRNA RNAfold executable (Default: assumes RNAfold is in PATH) |
| -pr or --partition | Path to RNAstructure Partition executable (Default: assumes Partition is in PATH) |
| -pp or --probability-plot | Path to RNAstructure ProbabilityPlot executable (Default: assumes ProbabilityPlot is in PATH) |
| -dp or --data-path | Path to RNAstructure data tables (Default: assumes DATAPATH environment variable is already set) |

| | |
|--------------------------------------|---|
| -w <i>or</i> --window-size | Window size in nt for base-pair probability calculation (≥ 3 , Default: full transcript) |
| -wo <i>or</i> --window-offset | Offset for window sliding (Default: none) |
| -kb <i>or</i> --keep-bases | <p>Bases to report in the SPD file (Default: N [ACGT])</p> <p>Note: This parameter accepts any IUPAC code, or combinations of them (e.g. <i>-kb M</i>, or <i>-kb AC</i>). Reactivity for any other base will be reported as NaN.</p> |
| -d <i>or</i> --decimals | Number of decimals for reporting reactivities (1-10, Default: 3) |

6. Application case

Analysis of PARS data on GM12878 native deproteinized RNA structures

Reference: Wan, Y. *et al.* (2014) Landscape and variation of RNA secondary structure across the human transcriptome. *Nature*, **505**, 706–709.

GEO Dataset: GSE50676

Description: In this work, Wan and colleagues used PARS to probe transcriptome-wide the structures of RNA from GM12878 cells in native deproteinized conformation, following phenol-chloroform extraction.

First of all, data in Sequence Read Archive (SRA) format should be downloaded from the Gene Expression Omnibus database:

```
$ wget 'ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-
instant/reads/ByExp/sra/SRX%2FSRX346%2FSRX346863/SRR972714/SRR972714.sra' -O S1.sra
$ wget 'ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-
instant/reads/ByExp/sra/SRX%2FSRX346%2FSRX346864/SRR972715/SRR972715.sra' -O V1.sra
```

These commands will download the SRA files for the Nuclease S1 and RNase V1 treatments. Once downloaded, SRA files should be converted to FastQ format. To perform conversion, it is necessary first to download and install the NCBI SRA Toolkit (<http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software>). Issue the following commands to convert the SRA files into FastQ files:

```
$ fastq-dump S1.sra
$ fastq-dump V1.sra
```

To build the reference index for the *Homo sapiens* genome (hg19 assembly), using the Ensembl gene annotation, run the *RSF Reference Builder* module with the following parameters:

```
$ rsf-index -g hg19 -a ensGene
```

```
[+] Making output directory...
[+] Connecting to UCSC genome database (genome-mysql.cse.ucsc.edu:3306)...
[+] Connected. Searching annotation...
[+] Annotation found. Validating columns...
[+] Downloading annotation data. Please wait...

[!] Warning: No reference multi-FASTA file has been provided.
            RSF Index will now try to download reference genome sequence
            from UCSC DAS server.
            This may take up to hours, depending on your connection's speed.

[+] Downloading sequence data for 52 chromosomes. Please wait...

[*] Chromosome chr1 [DONE]
[*] Chromosome chr10 [DONE]
[*] Chromosome chr11 [DONE]
[*] Chromosome chr12 [DONE]
[*] Chromosome chr13 [DONE]
[*] Chromosome chr14 [DONE]
[*] Chromosome chr15 [DONE]
[*] Chromosome chr16 [DONE]
[*] Chromosome chr17 [DONE]
[*] Chromosome chr17_ctg5_hap1 [DONE]
[*] Chromosome chr17_g1000205_random [DONE]
[*] Chromosome chr18 [DONE]
[*] Chromosome chr19 [DONE]
[*] Chromosome chr19_g1000209_random [DONE]
[*] Chromosome chr1_g1000191_random [DONE]
[*] Chromosome chr1_g1000192_random [DONE]
[*] Chromosome chr2 [DONE]
[*] Chromosome chr20 [DONE]
[*] Chromosome chr21 [DONE]
[*] Chromosome chr22 [DONE]
[*] Chromosome chr3 [DONE]
[*] Chromosome chr4 [DONE]
[*] Chromosome chr4_ctg9_hap1 [DONE]
[*] Chromosome chr4_g1000193_random [DONE]
[*] Chromosome chr4_g1000194_random [DONE]
[*] Chromosome chr5 [DONE]
```

```

[*] Chromosome chr6 [DONE]
[*] Chromosome chr6_apd_hap1 [DONE]
[*] Chromosome chr6_cox_hap2 [DONE]
[*] Chromosome chr6_dbb_hap3 [DONE]
[*] Chromosome chr6_mann_hap4 [DONE]
[*] Chromosome chr6_mcf_hap5 [DONE]
[*] Chromosome chr6_qbl_hap6 [DONE]
[*] Chromosome chr6_ssto_hap7 [DONE]
[*] Chromosome chr7 [DONE]
[*] Chromosome chr7_gl000195_random [DONE]
[*] Chromosome chr8 [DONE]
[*] Chromosome chr9 [DONE]
[*] Chromosome chrUn_gl000211 [DONE]
[*] Chromosome chrUn_gl000212 [DONE]
[*] Chromosome chrUn_gl000213 [DONE]
[*] Chromosome chrUn_gl000215 [DONE]
[*] Chromosome chrUn_gl000218 [DONE]
[*] Chromosome chrUn_gl000219 [DONE]
[*] Chromosome chrUn_gl000220 [DONE]
[*] Chromosome chrUn_gl000222 [DONE]
[*] Chromosome chrUn_gl000223 [DONE]
[*] Chromosome chrUn_gl000227 [DONE]
[*] Chromosome chrUn_gl000228 [DONE]
[*] Chromosome chrUn_gl000241 [DONE]
[*] Chromosome chrX [DONE]
[*] Chromosome chrY [DONE]

```

```

[+] Extracting transcript sequences...
[+] Building Bowtie transcriptome index from sequences. Please wait...
[+] All done.

```

Once the reference index has been prepared, the “*hg19_ensGene*” folder should appear in the current path, containing all the relevant index files. The FastQ files can now be passed to the RT Count module that will perform reads mapping, and compute normalized reactivity scores for all covered transcripts. The module will also perform all the necessary pre-processing steps on the FastQ files (trimming and adapter clipping). These steps are not mandatory, and can be skipped by simply setting “-t5 0” or “-t3 0” to either disable trimming from the 5’- or 3’-end of the reads, and “--clipped” to disable adapter clipping. According to the GEO datasets page, the last 51 nt of each read should be trimmed (moreover, following analysis of FastQ files with FASTX Toolkit, we also decided to trim the first 3 nt of each read). To perform reads mapping, and data normalization, run the RT Counter module with the following parameters:

```
$ rt-count -t tmp/ -k -b5 3 -b3 51 -c -bm 20 -bc 3200000 -bi hg19_ensGene/hg19_ensGene S1.fq V1.fq
```

```

[+] Reference FASTA file is present. Skipping FASTA regeneration...
[+] Making output directory...
[+] Guessing file types:

```

| <u>Sample</u> | <u>Type</u> | <u>5'-end trimming</u> |
|---------------|-------------|------------------------|
| S1 | FastQ | 3 nt |
| V1 | FastQ | 3 nt |

```

[+] Processing FastQ files...
[+] Input FastQ files are already clipped. Skipping adapter clipping...
[+] Mapping reads to transcriptome...

```

```

[-] Mapping sample "S1" (PID: 19759)
[-] Mapping sample "V1" (PID: 19760)

```

```
[+] Mapping statistics:
```

```

[*] Sample "S1" [Mapped: 68.82%; Failed: 29.08%; Suppressed: 2.11%]
[*] Sample "V1" [Mapped: 65.15%; Failed: 29.81%; Suppressed: 5.04%]

```

```
[+] Sorting BAM files...
```

```

[-] Sorting sample "S1.bam" (PID: 19879)
[-] Sorting sample "V1.bam" (PID: 19880)

```

```

[+] Getting transcripts from reference, and building count table base structure...
[+] Validating SAM/BAM file headers...
[+] Calculating per-base RT-stops and coverage. This may take a while...

```

```

[-] Processing sample "S1" (Thread #2)...
[-] Processing sample "V1" (Thread #1)...

```

```
[+] Statistics:

[*] Sample "S1": 58725 transcripts covered
[*] Sample "V1": 98452 transcripts covered

[+] Cleaning up temporary files...
[+] All done.
```

The program reports mapping statistics, and the number of covered transcripts. RT Count has generated a folder named *“rt_counter”*, which contains two folders:

1. a *“BAM”* folder containing mapped reads for both samples
2. a *“counts”* folder containing per-base transcript RT-stops in RTC (RT Count) format (see paragraph 4.2.1), and a RTI index file for fast RTC files random access:

\$ ls -l

```
rt_counter/BAM:
-rw-r--r-- 1 danny epigenetics 136030227  8 feb 12.04 S1.bam
-rw-r--r-- 1 danny epigenetics 130537439  8 feb 12.07 V1.bam

rt_counter/counts:
-rw-r--r-- 1 danny epigenetics   102  8 feb 12.02 index.rti
-rw-r--r-- 1 danny epigenetics 63658  8 feb 12.17 S1.rtc
-rw-r--r-- 1 danny epigenetics 63658  8 feb 12.22 V1.rtc
```

The RTC files can now be used as input for the RSF Norm module, that will normalize per-base signals. In this case, the V1 sample will be used as the untreated sample, while the S1 sample as the treated sample:

\$ rsf-norm -u rt_counter/counts/V1.rtc -t rt_counter/counts/S1.rtc -c parameters.conf

```
[+] Parsing configuration...

[!] Warning: Provided configuration file doesn't exist. Will be created...

[+] Configuration summary:

  Parameter                                Value

Scoring method                               Ding
Normalization method                         2-8%
Pseudocount                                  1
Maximum score                                10
Reactive bases                               ACGT
Normalize each base independently             No
Minimum mean coverage                        1
Minimum median coverage                      1

[+] Making output directory...
[+] Regenerating RTI index files...
[+] Loading transcript IDs... 204940 transcripts loaded.
[+] Normalizing reactivities [Last: ENST00000610125]
[+] Normalization statistics:

  [*] Covered transcripts:  3359
  [*] Discarded transcripts: 201581 total
                             201581 insufficient coverage
                             0 mismatch between treated and untreated sample sequence
                             0 absent in untreated sample reference

[+] All done.
```

RSF Norm has generated a folder containing one SPD (Structure Probing Data) file (see paragraph 4.3.2) for each transcript being analyzed, named *“S1_vs_V1_norm”*, and a configuration file with the parameters used for the analysis named *“parameters.conf”*:

\$ ls -l

```
-rw-r--r-- 1 danny epigenetics      129 12 feb 17.51 parameters.conf
drwxr-xr-x 3 danny epigenetics     4096 13 feb 10.36 rt_counter
drwxr-xr-x 2 danny epigenetics    135168 13 feb 12.52 S1_vs_V1_norm
```

SPD files can now be used as input for the RSF Fold module to perform data-guided folding prediction. In the following example, we will pass the RSF-normalized reactivity profile for the U1 snRNA (Ensembl ID: ENST00000383861) to the RSF Fold module to perform prediction using the ViennaRNA package:

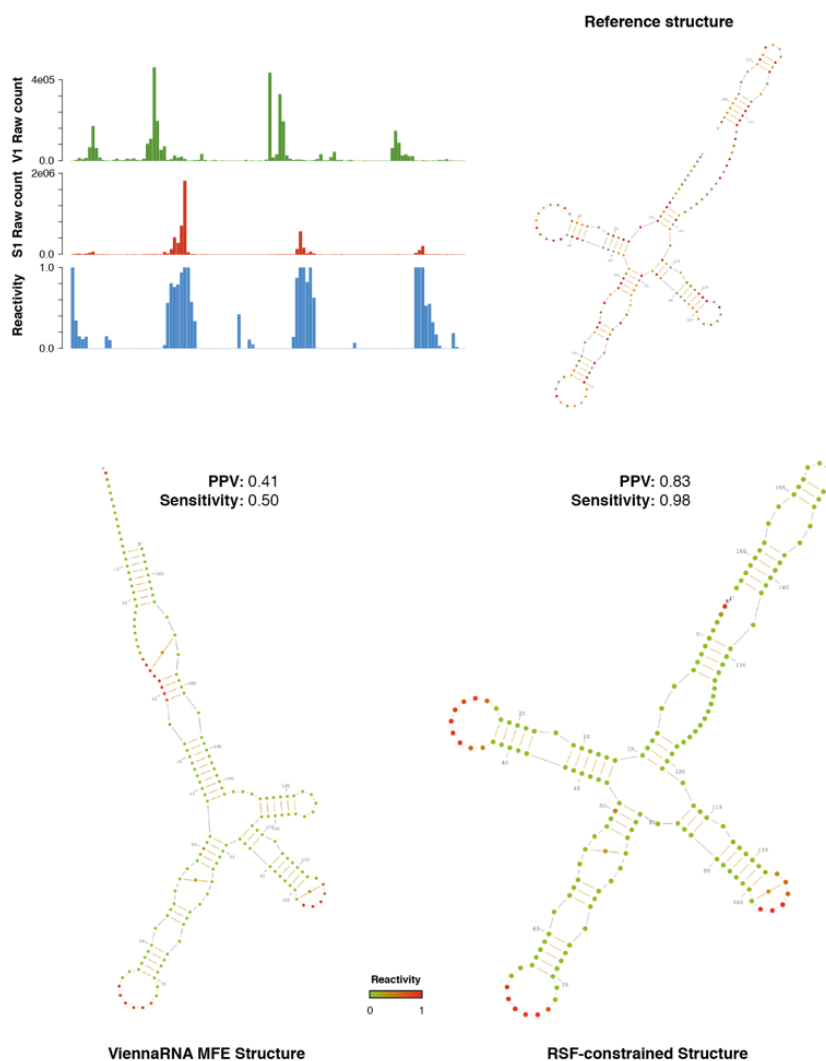
```
$ rsf-fold -m 1 -cm 1 -cc 3 -f 0.7 -g S1_vs_V1_norm/ENST00000383861.spd
```

```
[+] Checking method's requirements...
[+] Making output directory tree...
[+] Importing SPD file(s) [1 imported]
[+] Building RNA structurome [Last: ENST00000383861]
[+] Folding statistics:

  [*] Folded transcripts: 1
  [*] Discarded transcripts: 0 total
                        0 SPD parsing failed
                        0 constraint file generation failed
                        0 folding failed
                        0 I/O error

[+] All done.
```

As shown in the following figure, the RSF-constrained structure for the U1 snRNA (bottom right) better recapitulates the known reference structure (upper right) in terms of both Sensitivity and Positive Predictive Value (PPV) than the unconstrained MFE structure does (bottom left).



(Structure plots in this figure have been generated using the Assemble2 software, <http://bioinformatics.org/s2s/>)

Appendix A. List of UCSC genome assembly releases (<https://genome.ucsc.edu/FAQ/FAQreleases.html>)

| Species | UCSC Version | Release date | Release name | Status |
|------------------------|--------------|--------------|--|-----------------------------|
| Mammals | | | | |
| Human | hg38 | Dec. 2013 | Genome Reference Consortium GRCh38 | Available |
| | hg19 | Feb. 2009 | Genome Reference Consortium GRCh37 | Available |
| | hg18 | Mar. 2006 | NCBI Build 36.1 | Available |
| | hg17 | May 2004 | NCBI Build 35 | Available |
| | hg16 | Jul. 2003 | NCBI Build 34 | Available |
| | hg15 | Apr. 2003 | NCBI Build 33 | Archived |
| | hg13 | Nov. 2002 | NCBI Build 31 | Archived |
| | hg12 | Jun. 2002 | NCBI Build 30 | Archived |
| | hg11 | Apr. 2002 | NCBI Build 29 | Archived (data only) |
| | hg10 | Dec. 2001 | NCBI Build 28 | Archived (data only) |
| | hg8 | Aug. 2001 | UCSC-assembled | Archived (data only) |
| | hg7 | Apr. 2001 | UCSC-assembled | Archived (data only) |
| | hg6 | Dec. 2000 | UCSC-assembled | Archived (data only) |
| | hg5 | Oct. 2000 | UCSC-assembled | Archived (data only) |
| | hg4 | Sep. 2000 | UCSC-assembled | Archived (data only) |
| | hg3 | Jul. 2000 | UCSC-assembled | Archived (data only) |
| | hg2 | Jun. 2000 | UCSC-assembled | Archived (data only) |
| | hg1 | May 2000 | UCSC-assembled | Archived (data only) |
| Alpaca | vicPac2 | Mar. 2013 | Broad Institute Vicugna_pacos-2.0.1 | Available |
| | vicPac1 | Jul. 2008 | Broad Institute VicPac1.0 | Available |
| Armadillo | dasNov3 | Dec. 2011 | Broad Institute DasNov3 | Available |
| Bushbaby | otoGar3 | Mar. 2011 | Broad Institute OtoGar3 | Available |
| Baboon | papHam1 | Nov. 2008 | Baylor College of Medicine HGSC Pham_1.0 | Available |
| | papAnu2 | Mar. 2012 | Baylor College of Medicine Panu_2.0 | Available |
| Cat | felCat5 | Sep. 2011 | ICGSC Felis_catus-6.2 | Available |
| | felCat4 | Dec. 2008 | NHGRI catChrV17e | Available |
| | felCat3 | Mar. 2006 | Broad Institute Release 3 | Available |
| Chimp | panTro4 | Feb. 2011 | CGSC Build 2.1.4 | Available |
| | panTro3 | Oct. 2010 | CGSC Build 2.1.3 | Available |
| | panTro2 | Mar. 2006 | CGSC Build 2.1 | Available |
| | panTro1 | Nov. 2003 | CGSC Build 1.1 | Available |
| Chinese hamster | criGri1 | Jul. 2013 | Beijing Genomics Institution-Shenzhen C_griseus_v1.0 | Available |
| Cow | bosTau7 | Oct. 2011 | Baylor College of Medicine HGSC Btau_4.6.1 | Available |
| | bosTau6 | Nov. 2009 | University of Maryland v3.1 | Available |
| | bosTau4 | Oct. 2007 | Baylor College of Medicine HGSC Btau_4.0 | Available |
| | bosTau3 | Aug. 2006 | Baylor College of Medicine HGSC Btau_3.1 | Available |
| | bosTau2 | Mar. 2005 | Baylor College of Medicine HGSC Btau_2.0 | Available |
| | bosTau1 | Sep. 2004 | Baylor College of Medicine HGSC Btau_1.0 | Archived |
| Dog | canFam3 | Sep. 2011 | Broad Institute v3.1 | Available |
| | canFam2 | May 2005 | Broad Institute v2.0 | Available |
| | canFam1 | Jul. 2004 | Broad Institute v1.0 | Available |
| Dolphin | turTru2 | Oct. 2011 | Baylor College of Medicine Ttru_1.4 | Available |
| Elephant | loxAfr3 | Jul. 2009 | Broad Institute LoxAfr3 | Available |
| Ferret | musFur1 | Apr. 2011 | Ferret Genome Sequencing Consortium MusPutFur1.0 | Available |
| Gibbon | nomLeu3 | Oct. 2012 | Gibbon Genome Sequencing Consortium Nleu3.0 | Available |

| | | | | |
|-----------------------|---------|-----------|--|----------------------|
| | nomLeu2 | Jun. 2011 | Gibbon Genome Sequencing Consortium Nleu1.1 | Available |
| | nomLeu1 | Jan. 2010 | Gibbon Genome Sequencing Consortium Nleu1.0 | Available |
| Gorilla | gorGor3 | May 2011 | Wellcome Trust Sanger Institute gorGor3.1 | Available |
| Guinea pig | cavPor3 | Feb. 2008 | Broad Institute cavPor3 | Available |
| Hedgehog | eriEur2 | May 2012 | Broad Institute EriEur2.0 | Available |
| | eriEur1 | Jun. 2006 | Broad Institute Draft_v1 | Available |
| Horse | equCab2 | Sep. 2007 | Broad Institute EquCab2 | Available |
| | equCab1 | Jan. 2007 | Broad Institute EquCab1 | Available |
| Kangaroo rat | dipOrd1 | Jul. 2008 | Baylor/Broad Institute DipOrd1.0 | Available |
| Manatee | triMan1 | Oct. 2011 | Broad Institute TriManLat1.0 | Available |
| Marmoset | calJac3 | Mar. 2009 | WUSTL Callithrix_jacchus-v3.2 | Available |
| | calJac1 | Jun. 2007 | WUSTL Callithrix_jacchus-v2.0.2 | Available |
| Megabat | pteVam1 | Jul. 2008 | Broad Institute Ptevap1.0 | Available |
| Microbat | myoLuc2 | Jul. 2010 | Broad Institute MyoLuc2.0 | Available |
| Minke whale | balAcu1 | Oct. 2013 | KORDI BalAcu1.0 | Available |
| Mouse | mm10 | Dec. 2011 | Genome Reference Consortium GRCh38 | Available |
| | mm9 | Jul. 2007 | NCBI Build 37 | Available |
| | mm8 | Feb. 2006 | NCBI Build 36 | Available |
| | mm7 | Aug. 2005 | NCBI Build 35 | Available |
| | mm6 | Mar. 2005 | NCBI Build 34 | Archived |
| | mm5 | May 2004 | NCBI Build 33 | Archived |
| | mm4 | Oct. 2003 | NCBI Build 32 | Archived |
| | mm3 | Feb. 2003 | NCBI Build 30 | Archived |
| | mm2 | Feb. 2002 | MGSCv3 | Archived |
| | mm1 | Nov. 2001 | MGSCv2 | Archived (data only) |
| Mouse lemur | micMur1 | Jul. 2007 | Broad Institute MicMur1.0 | Available |
| Naked mole-rat | hetGla2 | Jan. 2012 | Broad Institute HetGla_female_1.0 | Available |
| | hetGla1 | Jul. 2011 | Beijing Genomics Institute HetGla_1.0 | Available |
| Opossum | monDom5 | Oct. 2006 | Broad Institute release MonDom5 | Available |
| | monDom4 | Jan. 2006 | Broad Institute release MonDom4 | Available |
| | monDom1 | Oct. 2004 | Broad Institute release MonDom1 | Available |
| Orangutan | ponAbe2 | Jul. 2007 | WUSTL Pongo_albelii-2.0.2 | Available |
| Panda | ailMel1 | Dec. 2009 | BGI-Shenzhen AilMel 1.0 | Available |
| Pig | susScr3 | Aug. 2011 | Swine Genome Sequencing Consortium Sscrofa10.2 | Available |
| | susScr2 | Nov. 2009 | Swine Genome Sequencing Consortium Sscrofa9.2 | Available |
| Pika | ochPri2 | Jul. 2008 | Broad Institute release OchPri2 | Available |
| Platypus | ornAna1 | Mar. 2007 | WUSTL v5.0.1 | Available |
| Rabbit | oryCun2 | Apr. 2009 | Broad Institute release OryCun2 | Available |
| Rat | rn5 | Oct. 2011 | RGSC Rnor_5.0 | Available |
| | rn4 | Nov. 2004 | Baylor College of Medicine HGSC v3.4 | Available |
| | rn3 | Jun. 2003 | Baylor College of Medicine HGSC v3.1 | Available |
| | rn2 | Jan. 2003 | Baylor College of Medicine HGSC v2.1 | Archived |
| | rn1 | Nov. 2002 | Baylor College of Medicine HGSC v1.0 | Archived |
| Rhesus | rheMac3 | Oct. 2010 | Beijing Genomics Institute CR_1.0 | Available |
| | rheMac2 | Jan. 2006 | Baylor College of Medicine HGSC v1.0 Mmul_051212 | Available |
| | rheMac1 | Jan. 2005 | Baylor College of Medicine HGSC Mmul_0.1 | Archived |
| Rock hyrax | proCap1 | Jul. 2008 | Baylor College of Medicine HGSC Procap1.0 | Available |
| Sheep | oviAri3 | Aug. 2012 | ISGC Oar_v3.1 | Available |
| | oviAri1 | Feb. 2010 | ISGC Ovis aries 1.0 | Available |
| Shrew | sorAra1 | Jun. 2006 | Broad Institute SorAra1.0 | Available |
| Sloth | choHof1 | Jul. 2008 | Broad Institute ChoHof1.0 | Available |

| | | | | |
|-------------------------------|---------|-----------|---|------------------|
| Squirrel | speTri2 | Nov. 2011 | Broad Institute SpeTri2.0 | Available |
| Squirrel monkey | saiBol1 | Oct. 2011 | Broad Institute SaiBol1.0 | Available |
| Tarsier | tarSyr1 | Aug. 2008 | WUSTL/Broad Institute Tarsyr1.0 | Available |
| Tasmanian devil | sarHar1 | Feb. 2011 | Wellcome Trust Sanger Institute Devil_refv7.0 | Available |
| Tenrec | echTel2 | Nov. 2012 | Broad Institute EchTel2.0 | Available |
| | echTel1 | Jul. 2005 | Broad Institute echTel1 | Available |
| Tree shrew | tupBel1 | Dec. 2006 | Broad Institute Tupbel1.0 | Available |
| Wallaby | macEug2 | Sep. 2009 | Tammar Wallaby Genome Sequencing Consortium Meug_1.1 | Available |
| White rhinoceros | cerSim1 | May 2012 | Broad Institute CerSimSim1.0 | Available |
| American alligator | allMis1 | Aug. 2012 | Int. Crocodilian Genomes Working Group allMis0.2 | Available |
| Atlantic cod | gadMor1 | May 2010 | Genofisk GadMor_May2010 | Available |
| Budgerigar | melUnd1 | Sep. 2011 | WUSTL v6.3 | Available |
| Chicken | galGal4 | Nov. 2011 | ICGC Gallus-gallus-4.0 | Available |
| | galGal3 | May 2006 | WUSTL Gallus-gallus-2.1 | Available |
| | galGal2 | Feb. 2004 | WUSTL Gallus-gallus-1.0 | Available |
| Coelacanth | latCha1 | Aug. 2011 | Broad Institute LatCha1 | Available |
| Elephant shark | calMil1 | Dec. 2013 | IMCB Callorhinchus_milli_6.1.3 | Available |
| Fugu | fr3 | Oct. 2011 | JGI v5.0 | Available |
| | fr2 | Oct. 2004 | JGI v4.0 | Available |
| | fr1 | Aug. 2002 | JGI v3.0 | Available |
| Lamprey | petMar2 | Sep. 2010 | WUGSC 7.0 | Available |
| | petMar1 | Mar. 2007 | WUSTL v3.0 | Available |
| Lizard | anoCar2 | May 2010 | Broad Institute AnoCar2 | Available |
| | anoCar1 | Feb. 2007 | Broad Institute AnoCar1 | Available |
| Medaka | oryLat2 | Oct. 2005 | NIG v1.0 | Available |
| Medium ground finch | geoFor1 | Apr. 2012 | BGI GeoFor_1.0 / NCBI 13302 | Available |
| Nile tilapia | oreNil2 | Jan. 2011 | Broad Institute Release OreNil1.1 | Available |
| Painted turtle | chrPic1 | Dec. 2011 | IPTGSC Chrysemys_picta_bellii-3.0.1 | Available |
| Stickleback | gasAcu1 | Feb. 2006 | Broad Institute Release 1.0 | Available |
| Tetraodon | tetNig2 | Mar. 2007 | Genoscope v7 | Available |
| | tetNig1 | Feb. 2004 | Genoscope v7 | Available |
| Turkey | melGal1 | Dec. 2009 | Turkey Genome Consortium v2.01 | Available |
| <i>X. tropicalis</i> | xenTro3 | Nov. 2009 | JGI v.4.2 | Available |
| | xenTro2 | Aug. 2005 | JGI v.4.1 | Available |
| | xenTro1 | Oct. 2004 | JGI v.3.0 | Available |
| Zebra finch | taeGut2 | Feb. 2013 | WUSTL v3.2.4 | Available |
| | taeGut1 | Jul. 2008 | WUSTL v3.2.4 | Available |
| Zebrafish | danRer7 | Jul. 2010 | Sanger Institute Zv9 | Available |
| | danRer6 | Dec. 2008 | Sanger Institute Zv8 | Available |
| | danRer5 | Jul. 2007 | Sanger Institute Zv7 | Available |
| | danRer4 | Mar. 2006 | Sanger Institute Zv6 | Available |
| | danRer3 | May 2005 | Sanger Institute Zv5 | Available |
| | danRer2 | Jun. 2004 | Sanger Institute Zv4 | Archived |
| | danRer1 | Nov. 2003 | Sanger Institute Zv3 | Archived |
| Deuterostomes | | | | |
| <i>C. intestinalis</i> | ci2 | Mar. 2005 | JGI v2.0 | Available |
| | ci1 | Dec. 2002 | JGI v1.0 | Available |
| Lancelet | braFlo1 | Mar. 2006 | JGI v1.0 | Available |
| <i>S. purpuratus</i> | strPur2 | Sep. 2006 | Baylor College of Medicine HGSC v. Spur 2.1 | Available |
| | strPur1 | Apr. 2005 | Baylor College of Medicine HGSC v. Spur_0.5 | Available |
| Insects | | | | |
| <i>A. mellifera</i> | apiMel2 | Jan. 2005 | Baylor College of Medicine HGSC v.Amel_2.0 | Available |

| | | | | |
|-------------------------|---------|-----------|---|-----------|
| | apiMel1 | Jul. 2004 | Baylor College of Medicine HGSC v.Amel_1.2 | Available |
| <i>A. gambiae</i> | anoGam1 | Feb. 2003 | IAGP v.MOZ2 | Available |
| <i>D. ananassae</i> | droAna2 | Aug. 2005 | Agencourt Arachne release | Available |
| | droAna1 | Jul. 2004 | TIGR Celera release | Available |
| <i>D. erecta</i> | droEre1 | Aug. 2005 | Agencourt Arachne release | Available |
| <i>D. grimshawi</i> | droGri1 | Aug. 2005 | Agencourt Arachne release | Available |
| <i>D. melanogaster</i> | dm3 | Apr. 2006 | BDGP Release 5 | Available |
| <i>D. melanogaster</i> | dm2 | Apr. 2004 | BDGP Release 4 | Available |
| | dm1 | Jan. 2003 | BDGP Release 3 | Available |
| <i>D. mojavensis</i> | droMoj2 | Aug. 2005 | Agencourt Arachne release | Available |
| | droMoj1 | Aug. 2004 | Agencourt Arachne release | Available |
| <i>D. persimilis</i> | droPer1 | Oct. 2005 | Broad Institute release | Available |
| <i>D. pseudoobscura</i> | dp3 | Nov. 2004 | Flybase Release 1.0 | Available |
| | dp2 | Aug. 2003 | Baylor College of Medicine HGSC Freeze 1 | Available |
| <i>D. sechellia</i> | droSec1 | Oct. 2005 | Broad Institute Release 1.0 | Available |
| <i>D. simulans</i> | droSim1 | Apr. 2005 | WUSTL Release 1.0 | Available |
| <i>D. virilis</i> | droVir2 | Aug. 2005 | Agencourt Arachne release | Available |
| | droVir1 | Jul. 2004 | Agencourt Arachne release | Available |
| <i>D. yakuba</i> | droYak2 | Nov. 2005 | WUSTL Release 2.0 | Available |
| | droYak1 | Apr. 2004 | WUSTL Release 1.0 | Available |
| Nematodes | | | | |
| <i>C. brenneri</i> | caePb2 | Feb. 2008 | WUSTL 6.0.1 | Available |
| | caePb1 | Jan. 2007 | WUSTL 4.0 | Available |
| <i>C. briggsae</i> | cb3 | Jan. 2007 | WUSTL Cb3 | Available |
| | cb1 | Jul. 2002 | WormBase v. cb25.agp8 | Available |
| <i>C. elegans</i> | ce10 | Oct. 2010 | WormBase v. WS220 | Available |
| | ce6 | May 2008 | WormBase v. WS190 | Available |
| | ce4 | Jan. 2007 | WormBase v. WS170 | Available |
| | ce2 | Mar. 2004 | WormBase v. WS120 | Available |
| | ce1 | May 2003 | WormBase v. WS100 | Archived |
| <i>C. japonica</i> | caeJap1 | Mar. 2008 | WUSTL 3.0.2 | Available |
| <i>C. remanei</i> | caeRem3 | May 2007 | WUSTL 15.0.1 | Available |
| | caeRem2 | Mar. 2006 | WUSTL 1.0 | Available |
| <i>P. pacificus</i> | priPac1 | Feb. 2007 | WUSTL 5.0 | Available |
| Other | | | | |
| Sea Hare | aplCal1 | Sep. 2008 | Broad Release Aplcal2.0 | Available |
| Yeast | sacCer3 | apr-11 | SGD April 2011 sequence | Available |
| | sacCer2 | June 2008 | SGD June 2008 sequence | Available |

References

1. Incarnato, D., Neri, F., Anselmi, F. & Oliviero, S. Genome-wide profiling of mouse RNA secondary structures reveals key features of the mammalian transcriptome. *Genome Biol.* **15**, 491 (2014).
2. Ding, Y. *et al.* In vivo genome-wide profiling of RNA secondary structure reveals novel regulatory features. *Nature* **505**, 696–700 (2014).
3. Rouskin, S., Zubradt, M., Washietl, S., Kellis, M. & Weissman, J. S. Genome-wide probing of RNA structure reveals active unfolding of mRNA structures in vivo. *Nature* **505**, 701–705 (2014).
4. Wan, Y. *et al.* Landscape and variation of RNA secondary structure across the human transcriptome. *Nature* **505**, 706–709 (2014).
5. Incarnato, D., Neri, F., Anselmi, F. & Oliviero, S. RNA structure framework: automated transcriptome-wide reconstruction of RNA secondary structures from high-throughput structure probing data. *Bioinformatics* **32**, 459–461 (2016).
6. Zarringhalam, K., Meyer, M. M., Dotu, I., Chuang, J. H. & Clote, P. Integrating chemical footprinting data into RNA secondary structure prediction. *PLoS ONE* **7**, e45160 (2012).
7. Lorenz, R., Luntzer, D., Hofacker, I. L., Stadler, P. F. & Wolfinger, M. T. SHAPE directed RNA folding. *Bioinformatics* **32**, 145–147 (2016).