Non-coding RNA Detection

1 Prerequisites to get started

1.1 Typographical Conventions

- `Constant width font` is used for program names, variable names and other literal text like input and output in the terminal window.
- Lines starting with a `$` within a literal text block are commands. You should type the text following the `$` into your terminal window finishing by hitting the return-key. (The `$` signifies the command line prompt, which may look different on your system).
- All other lines within a literal text block are the output from the command you just typed.

1.2 Data Files

Data files containing the sequences used in the examples below are available at `http://www.tbi.univie.ac.at/~ivo/RNA/tutorial/Data/`

1.3 Terminal, Command line and Editor

- You can get a **terminal** by moving your mouse-pointer to an empty spot on you desktop, clicking the right mouse-button on choosing "Open Terminal" from the pull-down menu.
- You can **run commands** in the terminal by typing them next to the command line prompt (usually something like $) followed by hitting the return-button.

```
$ date
Mon Mar 13 13:40:00 CET 2006
```

- To get more information about a command type **man** followed by the *command-name* and hitting of the return-button.

```
$ man date
```

Core Unix Commands

'|' ties stdout to stdin

'<' redirects stdout to stdin

'>' redirects stdout to a file.

- **cd** change the working directory (initially your "HOME")
- **ls** list files in the current (or a specified) directory

- **less** Show FILE(s) one page at a time
- **echo** Echo the STRING(s) to standard output

```
$ ls > file_list
$ less file_list
$ rm file_list
$ ls | less
```

The `wc` command prints the number of newlines, words, and bytes in files. Use `cd ../` to change to the parent directory and `cd FOO` to change to the directory `FOO`. With `ls -F` you can get a directory listing. The command `pwd` displays the path to the current working directory. `rm FOO.txt` to remove file `FOO.txt`.

The Emacs Editor

Use the mouse and the pull-down menu!!

- To edit the file FOO.seq run

  ```
  $ emacs FOO.seq
  ```

- type a short "random" DNA sequence into Emacs e.g. ATGAAGATGA"
- save the file via the menu `File->Save`
- replace all T by U to get a RNA sequence via the menu `Edit->Search->Replace`
  (space to leave unchanged; ! to replace all occurrences without asking, ...)

2 Consensus Structure Prediction

2.1 The Program `RNAalifold`

`RNAalifold` generalizes the folding algorithm for sequence alignments, treating the entire alignment as a single "generalized sequence". To assign an energy to a structure on such a generalized sequence, the energy is simply averaged over all sequences in the alignment. This average energy is augmented by a covariance term, that assigns a bonus or penalty to every possible base pair (i, j) based on the sequence variation in columns i and j of the alignment.

Sequence co-variations are a direct consequence of RNA base pairing rules. RNA helices normally contain only 6 out of the 16 possible combinations: the Watson-Crick pairs GC, CG, AU, UA, and the somewhat weaker wobble pairs GU and UG. Mutations in helical regions therefore have to be correlated. In particular we often find "compensatory mutations" where a mutation on one side of the helix is compensated by a second mutation on the other side, e.g. a C·G pair changes into a U·A pair. Mutations where only one pairing partner changes (such as C·G to U·G) are termed "consistent mutations".

Compensatory mutations are a strong indication of structural conservation, while consistent mutations provide a weaker signal. The covariance term used by `RNAalifold` therefore assigns a bonus of 1 kcal/mol to each consistent and

2 kcal/mol for each compensatory mutation. Sequences that cannot form a standard base pair incur a penalty of -1 kcal/mol. Thus, for every possible consensus pair between two columns i and j of the alignment a covariance score C_{ij} is computed by counting the fraction of sequence pairs exhibiting consistent and compensatory mutations, as well as the fraction of sequences that are inconsistent with the pair. The weight of the covariance term relative to the normal energy function, as well as the penalty for inconsistent mutations can be changed via command line parameters.

Apart from the covariance term, the folding algorithm in `RNAalifold` is essentially the same as for single sequence folding. In particular, folding an alignment containing just one sequence will give the same result as single sequence folding using `RNAfold`. For N sequences of length n the required CPU time scales as $\mathcal{O}(N \cdot n^2 + n^3)$ while memory requirements grow as the square of the sequence length. Thus `RNAalifold` is in general faster than folding each sequence individually. The main advantage, however, is that the accuracy of consensus structure predictions is generally much higher than for single sequence folding, where typically only between 40% and 70% of the base pairs are predicted correctly.

Apart from prediction of `MFE` structures `RNAalifold` also implements an algorithm to compute the partition function over all possible (consensus) structures and the thermodynamic equilibrium probability for each possible pair. These base pairing probabilities are useful to see structural alternatives, and to distinguish well defined regions, where the predicted structure is most likely correct, from ambiguous regions.

As a first example we'll produce a consensus structure prediction for the following four tRNA sequences.

```
$  cat four.seq
>M10740 Yeast-PHE
GCGGAUUUAGCUCAGUUGGGAGAGCGCCAGACUGAAGAUUUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCA
>K00349 Drosophila-PHE
GCCGAAAUAGCUCAGUUGGGAGAGCGUUAGACUGAAGAUCUAAAGGUCCCCGGUUCAAUCCCGGGUUUCGGCA
>K00283 Halobacterium volcanii Lys-1
GGGCCGGUAGCUCAUUUAGGCAGAGCGUCUGACUCUUAAUCAGACGGUCGCGUGUUCGAAUCGCGUCCGGCCCA
>AF346993
CAGAGUGUAGCUUAACACAAAGCACCCAACUUACACUUAGGAGAUUUCAACUUAACUUGACCGCUCUGA
```

`RNAalifold` uses aligned sequences as input. Thus, our first step will be to align the sequences. We use `clustalw` in this example, since it's one of the most widely used alignment programs and has been shown to work well on structural RNAs Other alignment programs can be used (including programs that attempt to do structural alignment of RNAs), but the resulting multiple sequence alignment must be in `Clustal` format.

Consensus Structure from related Sequences

(a) Prepare a sequence file (use file `tRNAs.seq`)

(b) Align the sequences

(c) Compute the consensus structure from the alignment

(d) Inspect the output files `alifold.out`,`alirna.ps`, `alidot.ps`

(e) for comparison fold the sequences individually using `RNAfold`

```
$ clustalw four.seq > four.out
$ RNAalifold -p tRNAs.aln
$ RNAfold -p < four.seq
```

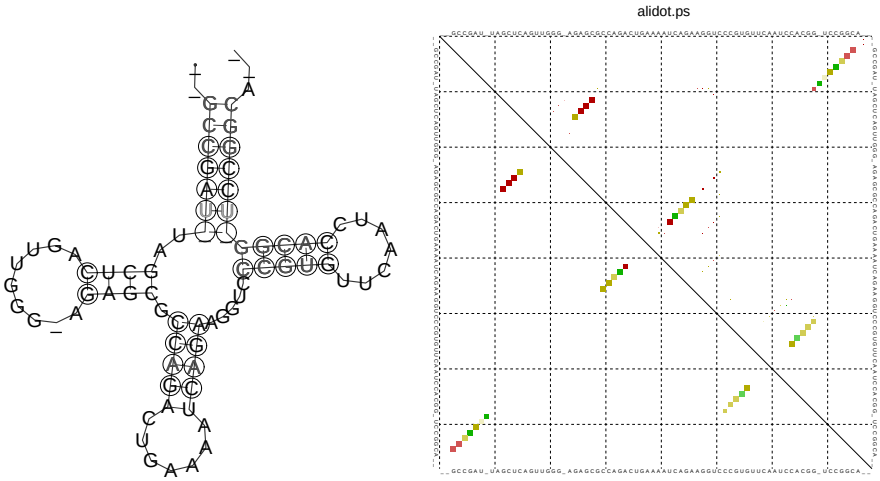
```
__GCCGAU_UAGCUCAGUUGGG_AGAGCGCCAGACUGAAAAUCAGAAGGUCCCGUGUUCAAUCCACGG_UCCGGCA__
..(((((((..((((.........)))).(((((.......)))))).....(((((.......)))))))))))...
 minimum free energy = -17.00 kcal/mol (-14.62 +  -2.38)
..(((((((,.((((.........)))).(((((.......)))))).....{(((((.......)))))))))))...
 free energy of ensemble = -17.95 kcal/mol
 frequency of mfe structure in ensemble 0.213582
```

The output contains a consensus sequence and the consensus structure in *bracket* notation. The consensus structure has an energy of -9.46 kcal/mol, which in turn consists of the average free energy of the structure -5.70 kcal/mol and the covariance term -3.76 kcal/mol. The strongly negative covariance term shows that there must be a fair number of consistent and compensatory mutations, but in contrast to the average free energy it's not meaningful in the biophysical sense.

Compare the predicted consensus structure with the structures predicted for the individual sequences using `RNAfold`. How often is the correct "clover-leaf" shape predicted?

`RNAalifold` Output Files

```
4 sequence; length of alignment 78
alifold output
    6    72  0  99.9%   0.006 GC:2    GU:1    AU:1
    7    71  0  99.5%   0.020 GC:2    AU:2
    5    73  1  99.9%   0.002 CG:2    GC:1
   33    43  0  97.9%   0.122 GC:2    GU:1    AU:1
    4    74  1  99.8%   0.005 CG:3
   31    45  0  98.2%   0.062 CG:3    UA:1
    3    75  1  99.0%   0.030 GC:3
   30    46  0  96.9%   0.097 CG:2    UA:2
```



The last output file produced by `RNAalifold -p`, named `alifold.out`, is a plain text file with detailed information on all plausible base pairs sorted by the likelihood of the pair. In the example above we see that the pair $(6, 72)$ has no

inconsistent sequences, is predicted almost with probability 1, and occurs as a GC pair in two sequences, a GU pair in one, and a AU pair in another.
`RNAalifold` automatically produces a drawing of the consensus structure in Postscript format and writes it to the file "alirna.ps". In the structure graph consistent and compensatory mutations are marked by a circle around the variable base(s), i.e. pairs where one pairing partner is encircled exhibit consistent mutations, whereas pairs supported by compensatory mutations have both bases marked. Pairs that cannot be formed by some of the sequences are shown gray instead of black. In the example given, many pairs show such inconsistencies. This is because one of the sequences (AF346993) is not aligned well by clustalw.

Note, that subsequent calls to `RNAalifold` will overwrite any existing output `alirna.ps` (`alidot.ps`, `alifold.out`) files in the current directory. Be sure to rename any files you want to keep.

3 Structural Alignments

3.1 Manually correcting Alignments

As the tRNA example above demonstrates, sequence alignments are often unsuitable as a basis for determining consensus structures. As a last resort, one may always try manually correcting an alignment. Sequence editors that are structure-aware may help in this task. In particular the SARSE `http://http://sarse.kvl.dk/` editor, and the `ralee-mode` for emacs `http://www.sanger.ac.uk/Users/sgj/ralee/` are useful, `ralee` has been pre-installed on your machines.

Try correcting the Clustal generated alignment `four.aln` from the example above. For this we first have to convert it to Stockholm format. Fortunately the formats are similar. Make a copy of the file add the correct header line and the consensus structure from RNAalifold:

```
$ cp four.aln four.stk
$ emacs four.stk
.....
$ cat four.stk
```

```
# STOCKHOLM 1.0

K00349          --GCCGAAAUAGCUCAGUUGGG-AGAGCGUUAGACUGAAGAUCUAAAGGUCCCCGGUUCAAUCCCGGGUUUCGGCA--
K00283          GGGCCG--GUAGCUCAUUUAGGCAGAGCGUCUGACUCUUAAUCAGACGGUCGCGUGUUCGAAUC--GCGUCCGGCCCA
M10740          --GCGGAUUUAGCUCAGUUGGG-AGAGCGCCAGACUGAAGAUUUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCA--
AF346993        --CAGAGUGUAGCUUAAC---ACAAAGCACCCAACUUACACUUAGGAGAUUUCAACUUAACUUGACCGCUCUGA----
#=GC SS_cons    ..(((((((..((((.........)))).(((((.......)))))).....(((((.......)))))))))))...
```

Now use the functions under the Edit menu to improve the alignment, the coloring by structure should help highlight misaligned positions.

3.2 Automatic structural alignments

Next, we'll compute alignments using two structural alignment programs: `locarna` and `stral`. `LocARNA` is an implementation of the Sankoff algorithm for simultaneous folding and alignment (i.e. it will generate both alignment and consensus structure. `StrAl` uses a much simpler string alignment algorithm that takes structure only approximately into account (a bit like only distinguishing paired

or un-paired, but forgetting about who pairs with whom). `StrAl` is thus a much faster program suitable for longer sequences than `LocARNA`.

Both programs can read the fasta file `four.seq`.

```
$ mlocarna-p four.seq
...
CLUSTAL --- LocARNA - Local Alignment of RNA --- Score: 2555

   (((((((..((((.........)))).........................((((.......)))).)))))))).
K00283_Halobacte   GGGCCGGUAGCUCAUUUAGGCAGAGCGUCUGACUCUUAAUCAGACGGUCGCGUGUUCGAAUCGCGUCCGGCCCA
K00349_Drosophil   GCCGAAAUAGCUCAGUU-GGGAGAGCGUUAGACUGAAGAUCUAAAGGUCCCCGGUUCAAUCCCGGGUUUCGGCA
M10740_Yeast_PHE   GCGGAUUUAGCUCAGUU-GGGAGAGCGCCAGACUGAAGAUUUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCA
AF346993           CAGAGUGUAGCUUA---ACACAAAGCACCCAACUUACACUUAGGAGAU-UUCAACU-UAACUUGACCGCUCUGA
   (((((((..((((.---.....))))..................-.((((..-....)))).))))))).

$ stral four.seq
$ cat resultDIR/four.seq
> M10740
GCGGAUUUAGCUCAGUU-GGGAGAGCGCCAGACUGAAGAUUUGGAGGUCCUGUGUUCGAUCCACAGAAUUCGCA
> K00349
GCCGAAAUAGCUCAGUU-GGGAGAGCGUUAGACUGAAGAUCUAAAGGUCCCCGGUUCAAUCCCGGGUUUCGGCA
> K00283
GGGCCGGUAGCUCAUUUAGGCAGAGCGUCUGACUCUUAAUCAGACGGUCGCGUGUUCGAAUCGCGUCCGGCCCA
> AF346993
CAGAGUGUAGCUUA-AC--ACAAAGCACCCAACUUACACUUAGGAGAUUUCAACUU--AACUUGACCGCUCUGA
```

Use RNAalifold to predict structures for all your alignments (Clustal, handcrafted, StrAl, and LocARNA) and compare them. The handcrafted and LocARNA alignments should be essentially perfect, in the StrAl alignment the T-arm is slipped for AF346993.

Other interesting approaches to structural alignment include `CMfinder`, `dynalign`, and `stemloc`.

4 Noncoding RNA gene prediction

Prediction of ncRNAs is still a largely unsolved problem in bioinformatics. Unlike protein coding genes, ncRNAs do not have any statistically significant features in primary sequences that could be used for reliable prediction. A large class of ncRNAs, however, depend on a defined secondary structure for their function. As a consequence, evolutionarily conserved secondary structures can be used as characteristic signal to detect ncRNAs. All currently available programs for *de novo* prediction make use of this principle and are therefore, by construction, limited to structured RNAs.

Programs to predict structural RNAs

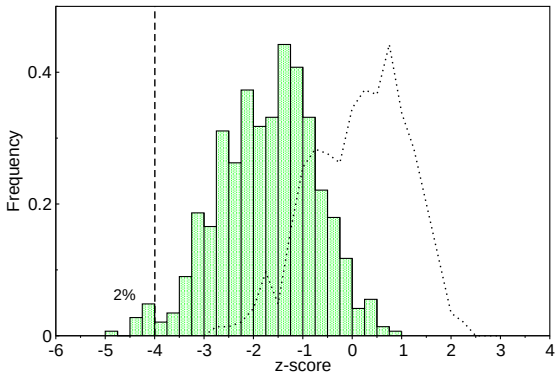
- `QRNA` (Eddy & Rivas, 2001)
- `ddbRNA` (di Bernardo, Down & Hubbard, 2003)
- `MSARi` (Coventry, Kleitman & Berger, 2004)
- `AlifoldZ` (Washietl & Hofacker, 2004)
- `RNAz` (Washietl, Hofacker & Stadler, 2005)
- `EvoFold` (Pedersen et al, 2006)

4.1 AlifoldZ

`AlifoldZ` is based on an old hypothesis: functional RNAs are thermodynamically more stable than expected by chance. This hypothesis can be statistically tested by calculating z-scores: Calculate the MFE m of the native RNA and the mean μ and standard deviation σ of the background distribution of a large number of random (shuffled) RNAs. The normalized z-score $z = (m - \mu)/\sigma$ expresses how many standard deviations the native RNA is more stable than random sequences.

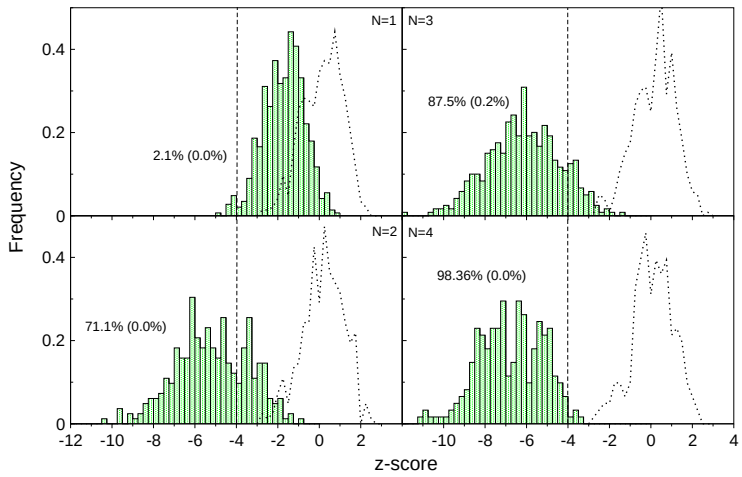
Unfortunately, most ncRNAs are not significantly more stable than the background. See for example the distribution of z-scores of some tRNAs.

z-score distribution of tRNAs



`AlifoldZ` calculates z-scores for consensus structures folded by `RNAalifold`. This significantly improves the detection performance compared to single sequence folding:

z-score distribution of tRNA consensus folds



Installation and basic usage of AlifoldZ

- Available at: `http://www.tbi.univie.ac.at/papers/SUPPLEMENTS/Alifoldz/`

- One single Perl script (needs `RNAfold` and `RNAalifold`)

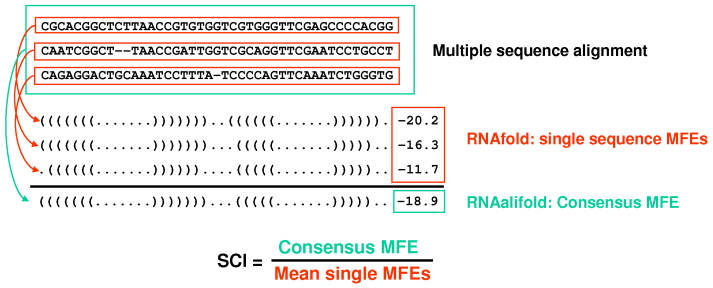
```
$ alifoldz.pl -h
$ alifoldz.pl miRNA.aln
$ alifoldz.pl -w 120 -x 100 < unknown.aln
```

4.2 RNAz

`AlifoldZ` has some shortcomings that limits its usefulness in practice: The z-scores are not deterministic, i.e. you get a different score each time you run `AlifoldZ`. To get stable z-scores you need to sample a large number of random alignments which is computationally expensive. Moreover, `AlifoldZ` is extremely sensitive to alignment errors.

The program `RNAz` overcomes these problems by using a different approach to asses a multiple sequence alignment for significant RNA structures. It is based on two key innovations: (i) The structure conservation index (SCI) to measure structural conservation in an alignment and (ii) z-scores that are calculated by regression without sampling. Both measures are combined to an overall score that is used to classify an alignment as "structured RNA" or "other".

The structure conservation index



- The structure conservation index is an easy way to normalize an `RNAalifold` consensus MFE.

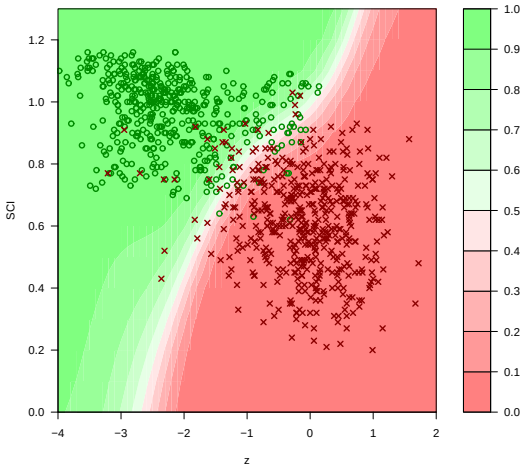
z-score regression

- The mean μ and standard deviation σ of random samples of a given sequence are functions of the length and the base composition:

$$\mu, \sigma(length, \frac{GC}{AT}, \frac{G}{C}, \frac{A}{T})$$

- It is therefore be possible to *calculate* z-scores by solving this 5 dimensional regression problem.

SVM Classification



- A support vector machine learning algorithm is used to classify an alignment based on z-score and structure conservation index.

Installation of RNAz

- Available at: `www.tbi.univie.ac.at/~wash/RNAz`
- Package includes the core program `RNAz` in ISO `C`, a set of helper programs in Perl, and an extensive manual.
- Standard installation (requires root privileges):

```
$ tar -xzf RNAz-1.0-tar.gz
$ cd RNAz-1.0
$ ./configure
$ make
$ make install
$ cp /usr/local/share/RNAz/perl/* /usr/local/bin
```

Basic usage of RNAz

- `RNAz` reads one or more multiple sequence alignments in ClustalW or MAF format.

```
$ RNAz --help
$ RNAz tRNA.aln
$ RNAz --both-strands --predict-strand tRNA.maf
```

Advanced usage of RNAz

- `RNAz` is limited to a maximum alignment length of 400 columns and a maximum number of 6 sequences. To process larger alignments a set of Perl helper scripts are used.

- Selecting one or more subsets of sequences from an alignment with more than 6 sequences:

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

- Scoring long alignments in overlapping windows:

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

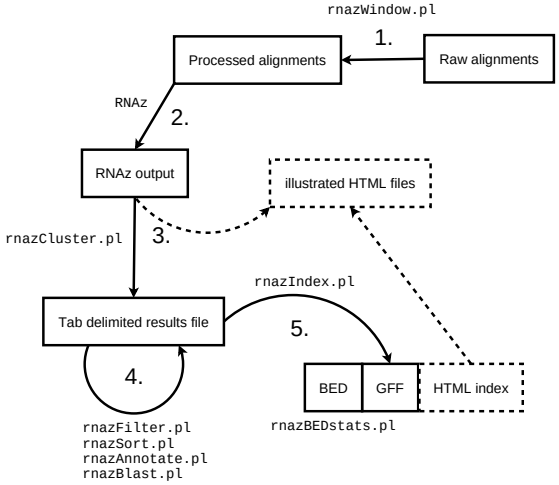
4.3 Large scale screens

The `RNAz` package provides a set of Perl scripts that implement a complete analysis pipeline suitable for medium to large scale screens of genomic data.

General procedure

(a) Obtain or create multiple sequence alignments in MAF format

(b) Run through the `RNAz` pipeline:



Examples in this tutorial

Analyze snoRNA cluster in the human genome for conserved RNA structures: download pre-computed alignments from the UCSC genome browser and run it through the `RNAz` pipeline

Example b: Obtaining pre-computed alignments from UCSC

- Go to the UCSC genome browser (`genome.ucsc.edu`) and go to "Tables". Download "multiz17" alignments in MAF format for the region: chr11:93103000-93108000

Example b: Running the pipeline

- The Perl scripts are run in the same order as in Example 1:

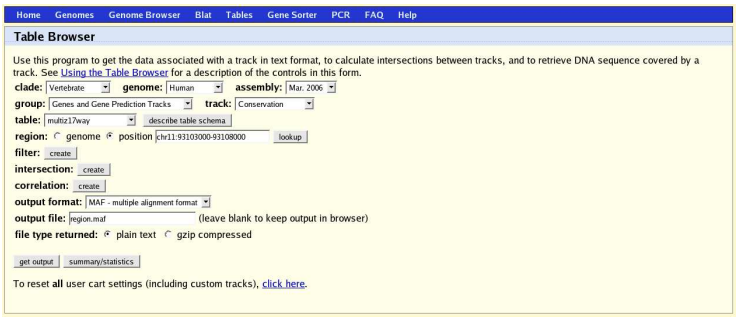
```
$ rnazWindow.pl --min-seqs=4 region.maf > windows.maf
$ RNAz --both-strands --show-gaps --cutoff=0.5 windows.maf > rnaz.out
$ rnazCluster.pl --html rnaz.out > results.dat
$ rnazAnnotate.pl --bed annotation.bed results.dat > results_annotated.dat
$ rnazIndex.pl --html results_annotated.dat > results/index.html
```

- The results can be exported as UCSC BED file which can be displayed in the genome browser:

```
$ rnazIndex.pl --bed --ucsc results.dat > prediction.bed
```

Example b: Visualizing the results on the genome browser

- Upload the BED file as "Custom Track"...



- ...and have a look at the results: