14 June 2005

eQRNA-2.0.3c documentation

Elena Rivas and Sean R. Eddy

{elena,eddy}@genetics.wustl.edu.

Howard Hughes Medical Institute
Department of Genetics
Washington University, St. Louis, MO 63110, USA

Abstract

Basic how–to guide to install and run eQRNA-2.0.3c.
eQRNA-2.0.3cis a new eQRNA generation that uses probabilistic evolutionary models to be able to tune all the parameters (transition and emission probabilities) of the three eQRNA models to any possible degree of sequence divergence.

Install eQRNA

```
Source distribution (qrna-2.0.3c.tar.Z):

    setenv QRNADB  (HOME)/qrna-2.0.3c/lib  assign a location for the libraries.

    tar zxf qrna-2.0.3c.tar.Z       Unpacks the archive. (makes a new directory, qrna-2.0.3c).
    cd qrna-2.0.3c                  Moves into the distribution toplevel directory.
    cd squid                        Moves into the directory squid.
    make                            Builds the binaries for the squid library.
    cd squid02                      Moves into the directory squid02.
    make                            Builds the binaries for the squid library.
    cd ..                           Goes back to toplevel directory.
    cd src                          Moves into the source directory.
    make                            Builds the binaries.
```

It should build cleanly on just about any UNIX machine.

The directory eQRNA-2.0.3c includes the following subdirectories,

Demos	A collection of files to demonstrate how to use eQRNA.
documentation	Contains a userguide manual.
lib	Contains all the additional files used as input by eQRNA. This directory is accessed by setting the environment variable QRNADB.
Licenses	The license for this software.
scripts	The wrap around Perl scripts used with eQRNA.
squid	An old version of Sean Eddy's squid library for sequence handling.
squid02	An newer version of Sean Eddy's squid library for sequence handling.
src	The source files, and headers. It also includes a Makefile. After building eQRNA the executable is located in this directory.

Create a eQRNA input file from a blast output

- Start with a WUBLASTN output file **foo.blast**.
  Example, the file in the 'qrna-2.0.3c/Demos/" directory **HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast**, which is the result of blasting a collection of 850 annotated Human RNA genes against the mouse genome.
- Then you need to run the Perl scripts **qrna-2.0.3c/scripts/blastn2qrnadepth.pl**.

```
usage: blastn2qrnadepth.pl [options] <blastfile>
options:
-c             :  DONOT resolve conflicts of dept  [ default: do by truncating alignments  ]
-d <depth>     :  max number of alignment coverage per position  [ default depth = 1       ]
-e <max_eval>  :  maximum evalue   of blast hits allowed         [ default max_eval = 0.01 ]
-g <org_name>  :  name of the blasting organism                  [ default 'org'           ]
-h             :  make histogram of depth per position           [ ]
-i <min_id>    :  minimum identity of blast hits allowed         [ default min_id = 0      ]
-j <max_id>    :  maximum identity of blast hits allowed         [ default max_id = 100    ]
-l <min_len>   :  minimum length   of blast hits allowed         [ default min_len = 1     ]
-o <outfile>   :  output qfile                                   [ default = 'blastfile.q' ]
-r             :  calculate depth respect to the database instead of the query
```

```
-s <shift>     :  position shift when calculating depth          [ default shift = 1       ]
-w <which>     :  criteria to pick alignments                    [ default which = 'SC'    ]
                                    ID -- best % identity
                                    SC -- best score
-x <name>      : ignore given name, use this one for gff outputs

-q             : gff file write the database entrie instead of the query
```

- "depth" is a parameter to limit the number of blast alignments that include a given position. If you do not want to constrain the number of blast alignments this way, set "depth" to a very large value. Limiting the depth is important for eukariotic genomes with large amount of repetitive sequences. The default depth = 1 will give the best alignment per position. If you want to allow some redundancy a depth of 5 or 10 is recommended.
- If the option -c is not activated, the program truncates alignments when the depth exceeds the allowed one, and the conflict cannot be resolved simply by omitting the whole alignment. If the option -c is activated, alignments are not altered at all, but then the depth becomes an approximate parameters; they may be some positions with a depth larger than the allowed one.
- "max_eval" removes blastn alignments with E values above this cutoff.
- "org_name" is the name of the organism as it will appear in the gff file
- "min_id" is the minimum identity of blastn alignments allowed.
- "max_id" is the maximum identity of blastn alignments allowed.
- "min_len" is the minimum length of blastn alignments allowed.
- "shift" is a parameter related to "depth". It indicates how many nucleotides to skip after every check for depth. I recommend not to change it from its default value.
- "which" has to values "SC" or "ID". It determines the criteria to pick the best alignments when applying the depth constraint.

Notice: Even if you do not want to prune **foo.blast**, you have to run this script.

- The result of running
  qrna-2.0.3c/scripts/blastn2qrnadepth.pl foo.blast
  is the three files:
  - **foo.blast.E<eval>.D<depth>.q** is the input file taken by eQRNA.
    It is a collection of sequences in fasta format, where two consecutive sequences are the two component of an alignment with the gaps left in place.
  - **foo.blast.E<eval>.D<depth>.q.rep** is a report of the BLASTN alignment that have been pruned in the process of creating **foo.blast.E<eval>.D<depth>.q** according to the options used in **blastn2qrnadepth.pl**.
  - **foo.blast.E<eval>.D<depth>.q.gff** is the gff format version of the surviving aligned regions of the query genome.

- The result of running
  qrna-2.0.3c/scripts/blastn2qrnadepth.pl -g human HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast
  is the three files in the directory "qrna-2.0.3c/Demos/":
  HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast.E0.01.D1.q
  HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast.E0.01.D1.q.rep
  HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast.E0.01.D1.q.gff
  Looking at the .rep file, we observe that initially the .blast file had 69,553 alignments. And that after using the default filters, we end up with 867 alignments approximately one alignment per RNA gene which is reasonable since by default we have assumed depth = 1. Of those 867 alignments 44 have been trimmed to avoid depths larger than one.

Run eQRNA

- eQRNA is a collection of programs, written in C.
  Source code is in "qrna-2.0.3c/src/". The executable is called "qrna".
- The "makefile" at "qrna-2.0.3c/src/" can be modified to change the C compiler or to change compilation flags.
- To run eQRNA :
  qrna-2.0.3c/src/qrna [options] foo.blast.q
- Options are:

```
qrna -- scores an alignment with the 3 models
         2.0.3c (Wed May 11 11:44:37 CDT 2005) using squid 1.5m (Sept 1997)
Usage: eqrna [-options] <input_file.q>
where options are:
   -A             : do an all-to-all comparison between the two input files
   -a             : print alignment
   -B             : sre_shuffle the alignment keeping the gap structure of the window
   -b             : shuffle the alignment
   -c <cfgfile>   : <cfgfile> to use to train the rna model (default = tRNA+rRNA)
   -C             : con_shuffle the alignment shuffling conserved, mutated, and gap positions independe
   -D <codfile>   : include a file of coding-coding frequencies for the coding model
   -d             : log2 form (default = log2-odds space )
   -e <num>       : number of sequen skipped in second file (for multiple comparisons). default 0.
   -E <censor>    : generate histogram of scores and calcule gumble parameters [censor=0 (no censorship
   -F             : change the overall base composition of the 3 models, based on nts frequencies in the
   -f             : use full dp for the probabilistic models (do not conserve the aligment-default is d
   -G             : change the overall base composition of the 3 models, based on nts frequencies for e
   -g             : do forward (default is viterbi)
   -H <Hexfile>   : include a file of Hexamer frequencies for the coding model
   -h             : print short help and usage info
   -i             : evolutionary time factor (default i=1)
   -j             : use semi-full dp for the probabilistic model (use the alignment created by OTH)
   -k             : allow pseudoknots (not implemented)
   -l <minlenhit> : change the minlenhit parameter (default 0)
   -L <maxlenhit> : change the maxlenhit parameter (default provided by longuest sequence)
   -m             : do Forward and Viterbi Diagonal dp
   -n             : do Forward and Viterbi Full     dp
   -N <num>       : in combination with -E number of shuffles. default 1.
   -o <outfile>   : direct structure-annotated sequence to <outfile>
   -p <pamfile>   : <pamfile> to use (default = BLOSUM62)
   -P             : pedantic, check your evolutionary models for inconsistencies
   -q             : do Forward and Viterbi semi-full dp
   -r             : do Nussinov rna model (default is a 3-state model)
   -R <ribofile>  : <ribofile> to use to train the rna model (default = RIBOPROB85-60)
   -s             : do global (not dp)
   -S             : sweep a collection of motifs(seqfile1) across another bunch of sequences(seqfile2)
   -t             : print traceback
   -v             : verbose debugging output
   -w <num>       : scanning window (default is full length)
   -x <num>       : slide positions (default is 50)
   -y <num>       : grab n sequences at random from the second data file to compare to each one of the

   --cyk         : use CYK algorithm to calculate RNA score (default is Inside).
   --latte       : I just called starbucks with your order...
   --ones        : score with the three models only the given strand.
   --parse       : input is a selex file. Por a given ss, it calculates the probablity of either the be
   --print <f>   : print to file <f> the actual alignments scored (useful if you want to store shuffled
   --rnass       : print the alignment with the predicted RNA secondary structure.
   --noends      : do not evaluate ends. Default now: calculate the actual boundaries of the model call
   --scan        : scanning version. no windows. faster.
   --shtoo       : qrna the alignment and also give one shuffled score.
   --twindow     : select the divergence time based on the %id of the window. Default is by %id of alig

 Debugging, experimentation:
   --regress <f> : save regression test information to file <f>
```

Some useful flags:

- For alignments that are too long, you can score them in chunks using -w ⟨windowsize⟩. The option -x ⟨slide⟩ to decide how many nucleotides to move before you score another chunk of the given alignment. Every window analyzed starts with "length alignment:".
- Option -L determines the maximum length allowed for an alignment to be scored. The default is 1,000. I use this variable to control the memory usage of the program. If you are using the "window" version, memory is determined by the window so you can set -L as large as you want. If not using a window, I would not use a max length larger than $1,500$.
- There are three different scoring algorithms:

  global (-s)

  local Viterbi (default)

  local forward (-g).

  You can report any of them or any desired combination of them. I would recommend to use the default.
- -w <win> -x <slide>
  Scores an alignments in windows of length <win>, moving nucleotides <slide> each time.
- –scan -w <win> -x <slide>
  Same results as before but using a smarter algorithm that saves time. This version is more memory consuming, so do not use it for very large window sizes. This option is incompatible with "–rnass" and shuffling the alignment by windows.
- –rnass
  Calculates a secondary structure for the aligned sequences using the posterior probabilities of each of the possible pairs of positions being basepaired. Implements the Outside algorithm for the RNA grammar. It then uses an unambiguous grammar by R. Dowell and S. Eddy (BMC bioinformatics **5**:71, 2004.) to use those posterior probabilities and generate the best structure.

Looking at a eQRNA output:

For file "qrna-2.0.3c/Demos/5s_rRNA.q" which contains an alignment of two 5S rRNAs we have,

qrna-2.0.3c/src/eqrna -a qrna-2.0.3c/Demos/5s_rRNA.q > **qrna-2.0.3c/Demos/5s_rRNA.q.eqrna**

The output looks like:

```
#-------------------------------------------------------------------------------
#      qrna 2.0.3c (Wed May 11 11:44:37 CDT 2005) using squid 1.5m (Sept 1997)
#-------------------------------------------------------------------------------
#      Rate-generating PAM model =  BLOSUM62
#-------------------------------------------------------------------------------
#      Rate-generating RIBOPROB matrix =  /RIBOPROB85-60.SEP04.mat
#-------------------------------------------------------------------------------
```

```
#      seq file  =  Demos/5s_rRNA.q
#                   #seqs: 2 (max_len = 143)
#-----------------------------------------------------------------------------------
#      full length version:  -- length range = [0,1000]
#-----------------------------------------------------------------------------------
# 1  [both strands]
>RF00001.5S_rRNA.U26684/296-411 (143)
>RF00001.5S_rRNA.M35563/5-120 (143)

Divergence time (variable): 0.141119 0.141770 0.135666
[alignment ID = 82.05 MUT = 16.24 GAP = 1.71]

length alignment: 117 (id=82.05) (mut=16.24) (gap=1.71)
posX: 0-116 [0-115](116) -- (0.27 0.24 0.28 0.22)
posY: 0-116 [0-115](116) -- (0.23 0.29 0.29 0.18)


        RF00001.5S_rRNA CCTGATACCCATAGAGCTGTGGTACCACCTGAATCCATGCCGAACTCAGA
        RF00001.5S_rRNA CCTGGTGTCCATAGAGCACTGGAACCACCTGATCCCATCCCGAACTCAGA

        RF00001.5S_rRNA AGTGAAACGCAGCATCGCCGATGGTAGTGTGAG.GTCTCCTCATGTGAGA
        RF00001.5S_rRNA AGTGAAACGGTGCATCGCCGATGGTAGTGTG.GGGCCTCCCCATGTGAGA

        RF00001.5S_rRNA GTAGGACAGTATCAGGT
        RF00001.5S_rRNA GTAGGTCAACGCCAGGC

LOCAL_DIAG_VITERBI -- [Inside SCFG]
OTH ends  (+) =  (0..[117]..116)
OTH ends *(-) =  (0..[117]..116)
COD ends  (+) =  (80..[4]..83)
COD ends *(-) =  (48..[37]..84)
RNA ends *(+) =  (0..[117]..116)
RNA ends  (-) =  (0..[117]..116)
winner = RNA
              OTH =      245.664              COD =      248.285              RNA =      263.894
   logoddspostOTH =        0.000  logoddspostCOD =        2.621  logoddspostRNA =       18.230
     sigmoidalOTH =      -18.230    sigmoidalCOD =      -15.609    sigmoidalRNA =       15.391
```

- Every new blast alignment starts with two lines:
  ">Query_name"
  ">Subject_name"
- "Divergence time" indicates the particular time parameterization of eQRNA used. By default eQRNA decides on the divergence time for the three different models (in this case $t = 0.141119$ for the OTH model, $t = 0.141770$ for the COD model, and $t = 0.135666$ for the RNA model.) given the percentage identity of the alignment (82.05%) [Divergence time (variable)]. You can set a particular divergence time using the option "-i" [Divergence time (fixed)].
- For each model, and for each strand, you are given the actual local regions (they could be more than one per model and strand) that score according to the model. The notation is (from..[length]..to). Coordinates for *both* strands are given relative to the positive strand. The "*" indicates the strand with the strongest signal for a given model.
- For a given scoring algorithm you get three rows of numbers:

  row 1 The scores of the alignment under each of the three models. The "null" model is a forth model which assumes that the two sequences in the alignment are independent from each other.

$$\log \frac{P(data|OTH)}{P(data|null)} \qquad \log \frac{P(data|COD)}{P(data|null)} \qquad \log \frac{P(data|RNA)}{P(data|null)}$$

row 2 The two (COD and RNA) log–odds posterior probabilities respect to the OTH model.

$$\log \frac{P(OTH|data)}{P(OTH|data)} \qquad \log \frac{P(COD|data)}{P(OTH|data)} \qquad \log \frac{P(RNA|data)}{P(OTH|data)}$$

row 3 The row you should be paying the most attention to. The three sigmoidal scores calculated using the other two models as null models. The model with the highest sigmoidal score is the winner.

$$\begin{aligned}
\sigma_{\text{OTH}} &= \log \frac{P(data|OTH)P(OTH)}{P(data|COD)P(COD) + P(data|RNA)P(RNA)} \\
\sigma_{\text{COD}} &= \log \frac{P(data|COD)P(COD)}{P(data|OTH)P(OTH) + P(data|RNA)P(RNA)} \\
\sigma_{\text{RNA}} &= \log \frac{P(data|RNA)P(RNA)}{P(data|OTH)P(OTH) + P(data|COD)P(COD)}
\end{aligned}$$

These scores are called sigmoidal because for each model M_i

$$P(M_i|data) = \frac{e^{\sigma_{M_i}}}{1 + e^{\sigma_{M_i}}},$$

which is a sigmoidal function. This function has the nice property that

$$P(M_i|data) > 1/2 \Longleftrightarrow \sigma_{M_i} > 0.$$

We assume a flat prior for the three models.

- Option -a prints the scored alignment.
- Option -B scrambles the columns of the pairwise alignment. The result is an alignment with the same percentage identity, but without any column correlations. This shuffling mantains the gaps structure of the original alignments.
- Option -C scrambles the columns of the pairwise alignment while mantaining the gap and conserved structure of the original alignment; that is, it shuffles columns with gaps amongst themshelves, conserved columns amongst themshelves, and mutated columns amongst themshelves. This is the most conservative type of shufling. Compare the results of using -C for the alignment of two 5S rRNAs given before:

qrna-2.0.3c/src/eqrna -a -C qrna-2.0.3c/Demos/5s_rRNA.q > qrna-2.0.3c/Demos/5s_rRNA.q.con_shuffle.eqrna

---

```
#-------------------------------------------------------------------------------
#      qrna 2.0.3c (Wed May 11 11:44:37 CDT 2005) using squid 1.5m (Sept 1997)
#-------------------------------------------------------------------------------
#      Rate-generating PAM model =  BLOSUM62
#-------------------------------------------------------------------------------
#      Rate-generating RIBOPROB matrix =  /RIBOPROB85-60.SEP04.mat
#-------------------------------------------------------------------------------
#      seq file  =  5s_rRNA.q
#                   #seqs: 2 (max_len = 143)
#-------------------------------------------------------------------------------
#      full length version:  -- length range = [0,1000]
#-------------------------------------------------------------------------------
# 1  [both strands] (con_shuffled)
>RF00001.5S_rRNA.U26684/296-411 (143)
>RF00001.5S_rRNA.M35563/5-120 (143)

Divergence time (variable): 0.141119 0.141770 0.135666
[alignment ID = 82.05 MUT = 16.24 GAP = 1.71]

length alignment: 117 (id=82.05) (mut=16.24) (gap=1.71)(con_shuffled)
```

```
posX: 0-116 [0-115](116) -- (0.27 0.24 0.28 0.22)
posY: 0-116 [0-115](116) -- (0.23 0.29 0.29 0.18)


        RF00001.5S_rRNA AACACGTGCCGTAAAACTTGGAGGCGGAGGATCAGTCGATCGACAGCAAC
        RF00001.5S_rRNA AACATGCACCGTAAAACCAGGACGCGGAGGATGTGTCGTTCGACAGCAAC

        RF00001.5S_rRNA TGGTAGTCTTAGGCCTGTTGGCGAAAGATCGAG.CTCTTTACCCTCACCC
        RF00001.5S_rRNA TGGTAGTCTCGGGCCTGTTGGCGAAAGATCG.GGCCCTTTGCCCTCACCC

        RF00001.5S_rRNA TACGCTGGATGAAAGAT
        RF00001.5S_rRNA TACGCCGGGACTAAGAC

LOCAL_DIAG_VITERBI -- [Inside SCFG]
OTH ends  (+) =  (0..[117]..116)
OTH ends *(-) =  (0..[117]..116)
COD ends  (+) =  (74..[10]..83)
COD ends *(-) =  (60..[25]..84)
RNA ends  (+) =  (30..[87]..116)
RNA ends *(-) =  (30..[87]..116)
winner = OTH
             OTH =      245.664             COD =      243.373             RNA =      242.141
   logoddspostOTH =        0.000  logoddspostCOD =       -2.291  logoddspostRNA =       -3.523
     sigmoidalOTH =        1.779    sigmoidalCOD =       -2.411    sigmoidalRNA =       -3.791
```

The alignment still has 82.05% identity as the original one, but now it gets classified as OTH since the RNA structure has been destroyed. We use the -C option to estimate false positives.

Example using the scanning version with a window:

Consider file "qrna-2.0.3c/Demos/Scerevisiae_orf_v_other_yeasts.q" which contains an alignment of a *S. cerevisiae* ORF The alignment has 514 nucleotides, and we would like to score it with eQRNA using a window of 150 nucleotides, and moving the window 50 nucleotides each time.

qrna-2.0.3c/src/eqrna -w 150 -x 50 qrna-2.0.3c/Demos/Scerevisiae_orf_v_other_yeasts.q > Scerevisiae_orf_v_other_yeasts.q.W150.X50.eqrna

The output for the first two windows looks like:

```
#-------------------------------------------------------------------------------
#      qrna 2.0.3c (Wed May 11 11:44:37 CDT 2005) using squid 1.5m (Sept 1997)
#-------------------------------------------------------------------------------
#      Rate-generating PAM model =  BLOSUM62
#-------------------------------------------------------------------------------
#      Rate-generating RIBOPROB matrix =  /RIBOPROB85-60.SEP04.mat
#-------------------------------------------------------------------------------
#      seq file  =  Demos/Scerevisiae_orf_v_other_yeasts.q
#                   #seqs: 2 (max_len = 545)
#-------------------------------------------------------------------------------
#      window version: window = 150   slide = 50 -- length range = [0,9999999]
#-------------------------------------------------------------------------------
# 1  [both strands]
>EFB1_I-142172-143158-80<621- (545)
>Contig1207-5-454>998- (545)
```

```
length of whole alignment after removing common gaps: 545
Divergence time (variable): 0.053647 0.055579 0.053819
[alignment ID = 92.66 MUT = 6.79 GAP = 0.55]

length alignment: 150 (id=96.67) (mut=3.33) (gap=0.00)
posX: 0-149 [0-149](150) -- (0.29 0.20 0.18 0.33)
posY: 0-149 [0-149](150) -- (0.27 0.21 0.20 0.32)
LOCAL_DIAG_VITERBI -- [Inside SCFG]
OTH ends *(+) =  (0..[150]..149)
OTH ends  (-) =  (0..[150]..149)
COD ends  (+) =  (78..[69]..146)
COD ends *(-) =  (6..[144]..149)
RNA ends *(+) =  (0..[150]..149)
RNA ends  (-) =  (0..[150]..149)
winner = COD
              OTH =      434.236              COD =      440.590              RNA =      435.769
   logoddspostOTH =        0.000  logoddspostCOD =        6.354  logoddspostRNA =        1.533
     sigmoidalOTH =       -6.404    sigmoidalCOD =        4.393    sigmoidalRNA =       -4.838

length alignment: 150 (id=94.67) (mut=5.33) (gap=0.00)
posX: 50-199 [50-199](150) -- (0.27 0.25 0.18 0.30)
posY: 50-199 [50-199](150) -- (0.27 0.24 0.21 0.28)
LOCAL_DIAG_VITERBI -- [Inside SCFG]
OTH ends *(+) =  (50..[150]..199)
OTH ends  (-) =  (50..[150]..199)
COD ends  (+) =  (78..[72]..149)
COD ends *(-) =  (54..[144]..197)
RNA ends  (+) =  (50..[150]..199)
RNA ends *(-) =  (50..[150]..199)
winner = COD
              OTH =      416.872              COD =      428.104              RNA =      414.360
   logoddspostOTH =        0.000  logoddspostCOD =       11.232  logoddspostRNA =       -2.512
     sigmoidalOTH =      -11.233    sigmoidalCOD =       10.999    sigmoidalRNA =      -13.745
```

How to read the information for each analyzed window:

- Each new analyzed window starts with the line:

  ```
  length alignment:
  ```

- For each window and for each sequence in the alignment we have a line of the form:

  ```
  posX: 50-199 [50-199](150) -- (0.27 0.25 0.18 0.30)
  ```

  The first pair of number represent the first and last coordinates of the window respect to the beginning of the alignment. The pair of numbers in brackets represent the mapping of the window into the coordinate system of sequence X (after removing gaps). The adjacent number in parenthesis is the length of that segment in sequence X. Finally the four decimal numbers in parenthesis are the fraction of A, C, G, and T's in the segment of sequence X involved in that particular window.

Example starting with a blastn output:

File **"qrna-2.0.3c/Demos/HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast"** is a typical WUBLASTN output file. This file is the result of blasting a collection of 850 annotated Human ncRNAs to the mouse genome.

Typing:

qrna-2.0.3c/scripts/blastn2qrnadepth.pl -g human qrna-2.0.3c/Demos/HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast

creates the file
"qrna-2.0.3c/Demos/HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast.E0.01.D1.q".
This file has selected those WUBLASTN alignment of any length, and any % identity, and E-value ≤ 0.01, and with depth = 1.
The two aligned sequence are now ready to be sent to eQRNA.

Typing:
qrna-2.0.3c/src/eqrna -w 150 -x 50 /qrna-2.0.3c/Demo/HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast.E0.01.D1.q > qrna-2.0.3c/Demo/HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast.E0.01.D1.q.W150.X50.eqrna

produces an output that starts with:

```
#---------------------------------------------------------------------------------
#      qrna 2.0.3c (Wed May 11 11:44:37 CDT 2005) using squid 1.5m (Sept 1997)
#---------------------------------------------------------------------------------
#      Rate-generating PAM model =  BLOSUM62
#---------------------------------------------------------------------------------
#      Rate-generating RIBOPROB matrix =  /RIBOPROB85-60.SEP04.mat
#---------------------------------------------------------------------------------
#      seq file  =  Demos/HG_13_RNAs_gene.fa.MGSCv3.fragchrom.blast.E0.01.D1.q
#                   #seqs: 1734 (max_len = 344)
#---------------------------------------------------------------------------------
#      window version: window = 150   slide = 50 -- length range = [0,9999999]
#---------------------------------------------------------------------------------
# 1  [both strands]
>NT_034563\-2555387\-2555837-71>406- (344)
>chr3\-frag968\-96700001\-96801000-15935>16265- (344)

length of whole alignment after removing common gaps: 344
Divergence time (variable): 0.244728 0.237432 0.233189
[alignment ID = 69.48 MUT = 24.42 GAP = 6.10]

length alignment: 150 (id=72.00) (mut=20.00) (gap=8.00)
posX: 0-149 [0-146](147) -- (0.13 0.39 0.34 0.14)
posY: 0-149 [0-140](141) -- (0.11 0.40 0.35 0.13)
LOCAL_DIAG_VITERBI -- [Inside SCFG]
OTH ends *(+) =  (0..[150]..149)
OTH ends  (-) =  (0..[150]..149)
COD ends *(+) =
COD ends  (-) =
RNA ends  (+) =  (48..[102]..149)
RNA ends *(-) =  (48..[102]..149)
winner = OTH
              OTH =      226.420              COD =      210.326              RNA =      226.210
   logoddspostOTH =        0.000  logoddspostCOD =      -16.094  logoddspostRNA =       -0.210
     sigmoidalOTH =        0.210    sigmoidalCOD =      -16.993    sigmoidalRNA =       -0.210
```

When alignments have been created using BLASTN, the name of the sequences involved in the alignment contains information about the alignment. That information is extracted from the BLASTN file, and processed by blastn2qrnadepth.pl in the following format:

```
>name-from[><]to-more_info (length_alignment)
```

where > (<) indicates the alignment involves the positive (negative) strand.
For instance in the previous example,

```
>NT_034563-2555387-2555837-71>406-Telomerase_RNA (344)
```

name = NT_034563-2555387-2555837
alignment involves positions 71 to 406 of that clone fragment, in the positive strand.
more_info = telomerase_RNA

```
>chr3-frag968-96700001-96801000-15935>16265- (344)
```

name = chr3-frag968-96700001-96801000
alignment involves positions 15935 to 16265 of the chromosome fragment, in the positive strand.
more_info = none

Analyze the results

There are several Perl scripts to parse through a eQRNA output file, and obtain different types of information.

To exemplify some of those possibilities in directory "qrna-2.0.3c/Demos/" I have included the following files

- "ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q"
  which is an already processed collection of 133 WUBLASTN alignments of intergenic *E. coli* to *Yersinia pestis.*
- "ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.eqrna"
  which is the result of running the command line:
  qrna-2.0.3c/src/eqrna -w 200 -x 50
  qrna-2.0.3c/Demos/ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q

qrna2gff.pl

This script converts a eQRNA output to gff format.
Command line:
qrna-2.0.3c/scripts/qrna2gff.pl [options] foo.eqrna

```
usage:  qrna2gff.pl [options] file.qrna

options:
-c <case>          :  cases (default is case = 1)
                        possible cases are:
                        0=GLOBAL
                        1=LOCAL_DIAG_VITERBI 2=LOCAL_DIAG_FORWARD
                        3=LOCAL_SEMI_VITERBI 4=LOCAL_SEMI_FORWARD
                        5=LOCAL_FULL_VITERBI 6=LOCAL_FULL_FORWARD
-I <min_id>        : min ID for analysis                      [default min_id = 0]
-J <max_id>        : max ID for analysis                      [default max_id = 100]
-G <min_gc>        : min GC for analysis                      [default min_gc = 0]
-H <max_gc>        : max GC for analysis                      [default max_gc = 100]
-g <typetarget>    :  which type of loci you want to analyze (default is all)
                        possible types of loci are:
                        OTH | COD | RNA
-l <lambda>        : lambda parameter of an EVD fit
-m <mu>            : mu parameter of an EVD fit
-n <size>          : size of database
-q <file.q>        : include qfile to get the actual ends of the qrna call
-s <type_of_score> : type of score (sigmoidal | simple)       [default = sigmoidal]
-u <cutoff>        : default is cutoff = 0
-w <whichorg>      : default is whichorg = 1  (use 1-for-org1 2-for-org2 12-for-both)
```

```
-x <name>          : ignore given name, use this one for gff outputs

-z <file>          : give a table of cutoffs for %gc ranges
```

We can convert the file
"ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W150.X50.eqrna"
to gff format using the following command,
qrna-2.0.3c/scripts/qrna2gff.pl -u 0.0
-q ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q
ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.qq.W150.X50.eqrna

As a result "qrna2gff.pl" produces an gff output file:
"ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W15.X50.eqrna.all.CUTOFF0.0.ID[100:0].GC[100:0].gf
which contains the coordinates and winning scores of all windows which score above a given cutoff. The beginning of the file looks like this,

---

```
Chromosome      QRNA    OTH     21079   21179   2.61840460601519        .       .       gene 'Chromosome' id
Chromosome      QRNA    RNA     29271   29316   3.75234675708036        .       .       gene 'Chromosome' id
Chromosome      QRNA    OTH     29318   29511   6.13015338253808        .       .       gene 'Chromosome' id
Chromosome      QRNA    OTH     29367   29559   2.80440056924974        .       .       gene 'Chromosome' id
Chromosome      QRNA    OTH     29417   29606   5.59321779216388        .       .       gene 'Chromosome' id
Chromosome      QRNA    OTH     29465   29650   5.6461631223522 .       .       gene 'Chromosome' id '65.98'
Chromosome      QRNA    OTH     57110   57275   1.32268486355351        .       .       gene 'Chromosome' id
Chromosome      QRNA    OTH     113262  113425  3.75112094661129        .       .       gene 'Chromosome' id
Chromosome      QRNA    OTH     121898  122091  1.70460148654424        .       .       gene 'Chromosome' id
Chromosome      QRNA    OTH     121849  122043  2.33771767947887        .       .       gene 'Chromosome' id
Chromosome      QRNA    OTH     121791  121943  3.84589012195515        .       .       gene 'Chromosome' id
Chromosome      QRNA    COD     159094  159129  0.204083583744543       .       .       gene 'Chromosome' id
```

If one is just interested on RNA candidates, the command
qrna-2.0.3c/scripts/qrna2gff.pl -u 0.0 -g RNA
-q ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q
ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.qq.W150.X50.eqrna

produces an gff output file:
"ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W15.X50.eqrna.RNA.CUTOFF0.0.ID[100:0].GC[100:0]
which contains the coordinates fo the windows with RNA scores above 0.0. The beginning of the file looks like this,

---

```
Chromosome      QRNA    RNA     29271   29316   3.75234675708036        .       .       gene 'Chromosome' id
Chromosome      QRNA    RNA     223560  223650  0.95699951153242        .       .       gene 'Chromosome' id
Chromosome      QRNA    RNA     223513  223675  0.946987126535079       .       .       gene 'Chromosome' id
Chromosome      QRNA    RNA     223465  223658  6.54202258217173        .       .       gene 'Chromosome' id
Chromosome      QRNA    RNA     484847  484967  0.839933401410345       .       .       gene 'Chromosome' id
```

---

loci_from_gff.pl

After a genome–to–genome eQRNA screen, one needs to post process the eQRNA output. The first kind of processing one would like to do with a eQRNA output is to extract the actual independent genomic regions (loci) that are identified as RNAs or coding by eQRNA. A locus can be composed of a single window, a collection of continuous windows from a given alignment, or a collection of windows from different alignments that include that particular region, and are identified with the same function by eQRNA. To obtain loci we have the script **loci_from_gff**.

To use this script you need to have your eQRNA output in gff format (see section before), and a gff formated file with the coordinates of the sequences that were originally blasted, in the example we are considered, the file of intergenic ecoli regions **qrna-2.0.3c/Demos/ ecolim56.noCDSnoRNAnoRep.gff**.
Command line:
qrna-2.0.3c/scripts/loci_from_gff.pl [options] reference.gff foo.eqrna.gff

---

```
usage:  loci_from_gff.pl [options] gff1 gff2

options:
-f <feature1>     :  which type of feature from gffDB    [ default is all]
-g <feature2>     :  which type of feature from gff    [ default is all]
-l <len>          :  until which coordinate    [ default is all]
-o <output>       :  output file [default = gff2.loci.gff]
```

We can extract the RNA genomic loci with scores larger than 0.0 using the following command,
qrna-2.0.3c/scripts/loci_from_gff.pl qrna-2.0.3c/Demos/ecolim56.noCDSnoRNAnoRep.gff
qrna-2.0.3c/Demos/ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.eqrna.
RNA.CUTOFF0.0.ID[100:0].GC[100:0].gff

The output file, ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.eqrna.
RNA.CUTOFF0.0.ID[100:0].GC[100:0].gff.loci.gff
contains 33 loci, and has the following form:

```
Chromosome      chromosome      all_loci        29271   29316   .       .       .
Chromosome      chromosome      all_loci        223465  223675  .       .       .
Chromosome      chromosome      all_loci        484847  484967  .       .       .
Chromosome      chromosome      all_loci        812328  812354  .       .       .
Chromosome      chromosome      all_loci        953829  954019  .       .       .
Chromosome      chromosome      all_loci        1014799 1014937 .       .       .
Chromosome      chromosome      all_loci        1255827 1255941 .       .       .
Chromosome      chromosome      all_loci        1286570 1286720 .       .       .
Chromosome      chromosome      all_loci        1407093 1407275 .       .       .
Chromosome      chromosome      all_loci        2088071 2088174 .       .       .
Chromosome      chromosome      all_loci        2403095 2403284 .       .       .
Chromosome      chromosome      all_loci        2496318 2496507 .       .       .
Chromosome      chromosome      all_loci        2531417 2531579 .       .       .
Chromosome      chromosome      all_loci        2744208 2744317 .       .       .
Chromosome      chromosome      all_loci        2967335 2967528 .       .       .
Chromosome      chromosome      all_loci        3208677 3208802 .       .       .
Chromosome      chromosome      all_loci        3267868 3268234 .       .       .
Chromosome      chromosome      all_loci        3316035 3316209 .       .       .
Chromosome      chromosome      all_loci        3376674 3376868 .       .       .
Chromosome      chromosome      all_loci        3408100 3408215 .       .       .
Chromosome      chromosome      all_loci        3426828 3427037 .       .       .
Chromosome      chromosome      all_loci        3440573 3440620 .       .       .
Chromosome      chromosome      all_loci        3451293 3451513 .       .       .
Chromosome      chromosome      all_loci        3472104 3472199 .       .       .
Chromosome      chromosome      all_loci        3948444 3948582 .       .       .
Chromosome      chromosome      all_loci        3955867 3955912 .       .       .
Chromosome      chromosome      all_loci        3999179 3999387 .       .       .
Chromosome      chromosome      all_loci        4056180 4056294 .       .       .
Chromosome      chromosome      all_loci        4140284 4140476 .       .       .
Chromosome      chromosome      all_loci        4175158 4175337 .       .       .
Chromosome      chromosome      all_loci        4177799 4177910 .       .       .
Chromosome      chromosome      all_loci        4178956 4179110 .       .       .
Chromosome      chromosome      all_loci        4205863 4206115 .       .       .
```

In the absence of the .q file (through the option -q), the eQRNA scores reported are associated to the whole scored window. The option -q allows us to allocate eQRNA local scores to the actual query fragment that scored as suppose to the whole window.

phase_count_fast.pl

Despite the name this method of obtaining loci is much slower than the previous. It is your only option if you do not know the actual coordinates of your query. This script gives you some additional information respect to the previous one, such as the number the window that contributed to construct a particular locus, and the average score for those windows, but if that information is not important to you, this script can be ignored.
Command line:

qrna-2.0.3c/scripts/phase_count_fast.pl [options] foo.eqrna query_organism database_organism

---

```
usage:  phase_count_fast.pl [options] file.qrna org1 org2

options:
-c <case>          :  cases (default is case = 1)
                        possible cases are:
                        0=GLOBAL
                        1=LOCAL_DIAG_VITERBI 2=LOCAL_DIAG_FORWARD
                        3=LOCAL_SEMI_VITERBI 4=LOCAL_SEMI_FORWARD
                        5=LOCAL_FULL_VITERBI 6=LOCAL_FULL_FORWARD
-I <min_id>        : min ID for analysis                      [default min_id = 0]
-J <max_id>        : max ID for analysis                      [default max_id = 100]
-G <min_gc>        : min GC for analysis                      [default min_gc = 0]
-H <max_gc>        : max GC for analysis                      [default max_gc = 100]
-g <typetarget>    :  which type of loci you want to analyze (default is all three)
                        possible types of loci are:
                        OTH | COD | RNA
-l <lambda>        : lambda parameter of an EVD fit
-m <mu>            : mu parameter of an EVD fit
-n <size>          : size of database
-o <output>        : output file [default = ]
-q <file.q>        : include qfile to get the actual ends of the qrna call
-s <type_of_score> : type of score (sigmoidal | simple)       [default = sigmoidal]
-t                 : towhomness -- obtains corresponding loci in the other organism
-u <cutoff>        : default is cutoff = 5
-v <loci_overlap>  :  minimun overlap required to build loci (default is loci_overlap = -1)
-w <whichorg>      : default is whichorg = 1  (use 1-for-org1 2-for-org2 12-for-both)
-x <name>          : ignore given name, use this one for gff outputs
```

We can extract the genomic loci (with scores larger than a cutoff set with option -u to 0 bits) from the file "ecolim56.noCDSnoRNAnoRe
CO92.dust.blast.E1e-05.D1.q.W200.X50.eqrna" using the following command,
qrna-2.0.3c/scripts/phase_count_fast.pl
-u 0.0 -q ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q
qrna-2.0.3c/Demos/ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.eqrna
ecoli.m56 Ypestis

There are three output files to this script.

- **qrna-2.0.3c/Demos/ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.e**
  Lists all loci ("RNA", "COD", and "OTH") with score larger than 0.0 for both organisms.

- **qrna-2.0.3c/Demos/ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.e**
  .ecoli.m56.gff
  List in gff format of all loci ("RNA", "COD", and "OTH") with score larger than 0.0 for the query organism, in this case "ecoli.m62".

- qrna-2.0.3c/Demos/ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.eqrna.all
  .Ypestis.gff
  List in gff format of all loci ("RNA", "COD", and "OTH") with score larger than 0.0 for the database organism, in this case "Ypestis".

The first output file **qrna-2.0.3c/Demos/**
ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.eqrna..
allloci.CUTOFF0.0.ID[**100:0**].**GC**[**100:0**].

qrna-2.0.3c/Demos/m52nc.fas-salmonella_typhi.q.W150.X50.eqrna.allloci.CUTOFF0.0.ID[**100:0**].**GC**[**100:0**] lists all the loci for both organism, starting with the "RNA" loci, and then proceeding with the "COD" loci, and finally the "OTH" loci. The file looks like,

---

```
---------------Some General Statistics-------------------
FILE:                   ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200
method:                 LOCAL_DIAG_VITERBI
Cutoff:                 0.0

gc:[0:100) ** cutoff = 0.0
min id:                 0
max id:                 100
min gc:                 0
max gc:                 100

# blastn hits:          133
# windows:              293
---------------------------------------------------------

---------------Statistics by Windows---------------------
# windows:              293

RNA>0:                  50/293
RNA>cutoff:             50/293

COD>0:                  54/293
COD>cutoff:             54/293

 in phases:             292/293
        RNA:                    50/292
        COD:                    54/292
        OTH:                    188/292

 in transitions:        0/293
        RNA/COD:                0/0
        RNA/OTH:                0/0
        COD/OTH:                0/0
        RNA/COD/OTH:            0/0
---------------------------------------------------------

---------------Statistics for RNA loci (ecoli.m56):-------------------
# loci: 33
ave_length:     184.36
```

```
1-loci Chromosome 29271 29316 (46) 1 RNA -10.94 3.75
2-loci Chromosome 223465 223675 (211) 3 RNA -18.95 2.82
3-loci Chromosome 484847 484967 (121) 1 RNA -11.88 0.84
4-loci Chromosome 812328 812354 (27) 3 RNA -11.23 4.41
```

For instance, we see that 33 independent RNA loci have been identified in *E. coli* that score as RNA above 0 bits out of the 293 overlapping windows analyzed.
The file then proceeds to list the coordinates of each of those loci. The information given for each locus has the following form

```
num-loci name_seq loc_from loc_to (loc_length) number_wind type_loc COD_sc RNA_sc
```

Therefore,

```
1-loci Chromosome 29271 29316 (46) 1 RNA -10.94 3.75
```

means that the first *E. coli* RNA locus corresponds to sequence named “Chromosome” (whole genome). The RNA locus has a length of 46 nucleotides and covers the region from nucleotide 29271 to 29316. Only one window have contributed to this RNA locus, and the average sigmoidal score for the coding model is -10.94 bits, while the average sigmoidal score for the RNA model is 3.75 bits.

The same information is included in gff format separated by organism. For instance for the intergenic ecoli all the loci in gff format can be found in file
**qrna-2.0.3c/Demos/
ecolim56.noCDSnoRNAnoRep.gff.all.fa_Yersinia.pestis-CO92.dust.blast.E1e-05.D1.q.W200.X50.eqrna.
allloci.CUTOFF0.0.ID[100:0].GC[100:0].ecoli.m56.gff**.
The beginning of the file looks like,

```
Chromosome      QRNA_loci       RNA     29271   29316   3.75234675708036        .       .       gene "Chromo
Chromosome      QRNA_loci       RNA     223465  223675  2.81533640674641        .       .       gene "Chromo
Chromosome      QRNA_loci       RNA     484847  484967  0.839933401410345       .       .       gene "Chromo
Chromosome      QRNA_loci       RNA     812328  812354  4.41093963462041        .       .       gene "Chromo
C
```

Here we report the beginning and end coordinates of a given locus respect to the “Chromosome” sequence, with the corresponding eQRNA winning score, which is an average of the eQRNA winning scores of all the windows contributing to define that locus.

qrna2col.pl

This script converts a eQRNA output to “col” format.
Command line:
qrna-2.0.3c/scripts/qrna2col.pl [options] foo.eqrna

```
usage:  qrna2col.pl [options] file.qrna

options:
-c <case>          :  cases (default is case = 1)
                        possible cases are:
                        0=GLOBAL
                        1=LOCAL_DIAG_VITERBI 2=LOCAL_DIAG_FORWARD
                        3=LOCAL_SEMI_VITERBI 4=LOCAL_SEMI_FORWARD
                        5=LOCAL_FULL_VITERBI 6=LOCAL_FULL_FORWARD
-d <qfile_dir>     : location of the .q file. (default is assumes ../qfile)
-f                 :  create qfile-type file with the sequences
-g <typetarget>    :  which type of loci you want to analyze (default is all)
```

```
                        possible types of loci are:
                        OTH | COD | RNA
-i <id_max>         :  max identity of alignments analysed (default is id_max = 100)
-l <id_max>         :  min fraction of the window that has to score for it to be taken (default all)
-q <qfile>          : name of the qfile  (default given by the qrna name)
-r                  : remove non-canonical pairs
-s <type_of_score> : type of score (sigmoidal | simple)       [default = sigmoidal]
-u <cutoff>         : default is cutoff = 0
```

Using the file“5S_arch.stk.fa.blast.D10.q.rnass.eqrna”, which is the result of running the command line:
qrna-2.0.3c/src/eqrna –rnass qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q,

the command line:
qrna-2.0.3c/scripts/qrna2col.pl -d qrna-2.0.3c/Demos -g RNA -u 0.0
qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q.rnass.eqrna

produces the file
qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q.rnass.eqrna.all.CUTOFF0.L0.allpairs.col
as output. The first two entries of this file are the first two sequences fragments of the first window that scores as “RNA” with a score higher than 0.0.
The file **qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q.rnass.eqrna** was the result of running eQRNA with the option “–rnass” which produces a *a posteriory* secondary structure of the region of the window scored as “RNA”. As a result the “col”–format file has a sixth column which includes the position a base is basepaired to or a “.” is single stranded. If the option “–rnass”is not used, the sixth column is blank.

```
; generated by qrna2col.pl
; ==========================================================
; QRNA  qrna 2.0.3c (Wed May 11 11:44:37 CDT 2005) using squid 1.5m (Sept 1997)
; TYPE  RNA (cutoff=0.0)
; COL1  label
; COL2  residue
; COL3  seqpos
; COL4  alignpos
; COL5  align_bp
; ENTRY Desumobi-1>132-
; LEN_A 132
; START 1
; ID    100.00
; SCORE 16.782306856792
; ISREV 1
; ----------
N       C       1       1       .
N       G       2       2       132
N       G       3       3       131
N       T       4       4       130
N       G       5       5       129
N       C       6       6       128
N       C       7       7       127
N       C       8       8       126
N       G       9       9       125
N       A       10      10      124
N       C       11      11      123
N       C       12      12      122
N       C       13      13      121
N       G       14      14      .
N       G       15      15      .
N       C       16      16      .
```

N	C	17	17	.
N	A	18	18	.
N	T	19	19	.
N	A	20	20	.
N	G	21	21	72
N	T	22	22	71
N	G	23	23	70
N	G	24	24	69
N	C	25	25	68
N	C	26	26	67
N	G	27	27	66
N	G	28	28	65
N	G	29	29	.
N	C	30	30	.
N	A	31	31	.
N	A	32	32	.
N	C	33	33	.
N	A	34	34	.
N	C	35	35	.
N	C	36	36	.
N	C	37	37	.
N	G	38	38	.
N	G	39	39	.
N	T	40	40	.
N	C	41	41	.
N	T	42	42	.
N	C	43	43	.
N	G	44	44	.
N	T	45	45	.
N	T	46	46	.
N	T	47	47	.
N	C	48	48	.
N	G	49	49	.
N	A	50	50	.
N	A	51	51	.
N	C	52	52	.
N	C	53	53	.
N	C	54	54	.
N	G	55	55	.
N	G	56	56	.
N	A	57	57	.
N	A	58	58	.
N	G	59	59	.
N	T	60	60	.
N	T	61	61	.
N	A	62	62	.
N	A	63	63	.
N	G	64	64	.
N	C	65	65	28
N	C	66	66	27
N	G	67	67	26
N	G	68	68	25
N	C	69	69	24
N	C	70	70	23
N	A	71	71	22
N	C	72	72	21
N	G	73	73	.

```
N       T       74      74      116
N       C       75      75      115
N       A       76      76      114
N       G       77      77      113
N       A       78      78      112
N       A       79      79      111
N       C       80      80      110
N       G       81      81      109
N       G       82      82      108
N       C       83      83      107
N       C       84      84      106
N       G       85      85      105
N       T       86      86      104
N       G       87      87      103
N       A       88      88      102
N       G       89      89      101
N       G       90      90      100
N       T       91      91      .
N       C       92      92      .
N       C       93      93      .
N       G       94      94      .
N       A       95      95      .
N       G       96      96      .
N       A       97      97      .
N       G       98      98      .
N       G       99      99      .
N       C       100     100     90
N       C       101     101     89
N       T       102     102     88
N       C       103     103     87
N       G       104     104     86
N       C       105     105     85
N       A       106     106     84
N       G       107     107     83
N       C       108     108     82
N       C       109     109     81
N       G       110     110     80
N       T       111     111     79
N       T       112     112     78
N       C       113     113     77
N       T       114     114     76
N       G       115     115     75
N       A       116     116     74
N       G       117     117     .
N       C       118     118     .
N       T       119     119     .
N       G       120     120     .
N       G       121     121     13
N       G       122     122     12
N       A       123     123     11
N       T       124     124     10
N       C       125     125     9
N       G       126     126     8
N       G       127     127     7
N       G       128     128     6
N       C       129     129     5
N       A       130     130     4
```

```
N       C       131     131     3
N       C       132     132     2
; **********
```

---

plot_eqrna.pl

Another kind of information that one might want to extract from a eQRNA output is a distribution of scores according to percentage identity of the alignments involved. For this purpose we have the script **plot_eqrna.pl**.

Command line:
   qrna-2.0.3c/scripts/plot_eqrna.pl [options] foo.eqrna (foo.shuffle.eqrna)

```
usage: plot_eqrna.pl [options] qrnafile qrnafilesh
-c <case>               :  cases (default is case = 1)
                              possible cases are:
                              0=GLOBAL
                              1=LOCAL_DIAG_VITERBI 2=LOCAL_DIAG_FORWARD
                              3=LOCAL_SEMI_VITERBI 4=LOCAL_SEMI_FORWARD
                              5=LOCAL_FULL_VITERBI 6=LOCAL_FULL_FORWARD
-a <max_n_ali>          : max number of alignments                [default all]
-d <divide_lodsc>       : for fits, lodsc for break of behaviour   [default divide_lodsc = 0]
-e <evalue>             : fit to an e-value
-i <min_id_plot>        : for ID plots                             [default min_id_plot = 50]
-j <max_id_plot>        : for ID plots                             [default max_id_plot = 100]
-I <min_id>             : min ID for analysis                      [default min_id = 0]
-J <max_id>             : max ID for analysis                      [default max_id = 100]
-G <min_gc>             : min GC for analysis                      [default min_gc = 0]
-H <max_gc>             : max GC for analysis                      [default max_gc = 100]
-g <max_gap_plot>       : for GAP plots                            [default max_gap_plot = 50]
-l <max_num_win>        : max number of windows to be considered   [default all]
-m <max_mut_plot>       : for MUT plots                            [default max_mut_plot = 50]
-n <time increments>    : for TIME histo                           [default inc_t = 0.001]
-o <outfile>            : outfile                                  [default = qrnafile.timeplot]
-p <paramfile>          : paramfile                                [default = none]
-s <type_of_score>      : type of score (sigmoidal | simple)       [default = sigmoidal]
-t <max_time_plot>      : for plots                                [default max_time_plot = 0.8]
-T <max_time>           : max time for analysis                    [default max_time = 0.8]
-S <min_time>           : min time for analysis                    [default min_time = 0]
-u <cutoff>             : plot all logodds, but add stats for those above cutoff [default cutoff = 0]
-w <displaycutoff>      : plot logodds larger than dislplaycutoff              [default display_cutoff = 0
-y <min_ord_rna_plot3>  : min ordinate for plotting ave and std RNA scores
-Y <max_ord_rna_plot3>  : max ordinate for plotting ave and std RNA scores
-z <min_ord_rna_plot4>  : min ordinate for plotting max RNA scores
-Z <max_ord_rna_plot4>  : max ordinate for plotting max RNA scores
-x <min_ord_cod_plot3>  : min ordinate for plotting ave and std COD scores
-X <max_ord_cod_plot3>  : max ordinate for plotting ave and std COD scores
-v <min_ord_cod_plot4>  : min ordinate for plotting max COD scores
-V <max_ord_cod_plot4>  : max ordinate for plotting max COD scores
```

To exemplify some of those possibilities in directory "qrna-2.0.3c/Demos/" I have included the following files

- "5S_arch.stk.fa.blast.D10.q"
  which is an already eQRNA–processed collection of 945 WUBLASTN alignments between the 58 archaeal 5S RNAs from the 5S Ribosomal RNA Database by Szymanski *et al*, NAR 30:176–178 (2002).

- "5S_arch.stk.fa.blast.D10.q.eqrna"
  which is the result of running the command line:
  qrna-2.0.3c/src/eqrna   qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q
- "5S_arch.stk.fa.blast.D10.q.con_shuffle.eqrna"
  which is the result of running the command line:
  qrna-2.0.3c/src/eqrna   -C   qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q

The command line
**qrna-2.0.3c/scripts/plot_eqrna.pl   -u 4.0   5S_arch.stk.fa.blast.D10.q.eqrna
5S_arch.stk.fa.blast.D10.q.con_shuffle.eqrna**

produces four output files in postscript:

- for COD scores
  - qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q.eqrna.ID[100:0].GC[100:0].id.spreadcod_histo_with_sh.ps
  - qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q.eqrna.ID[100:0].GC[100:0].id.spreadcod_scores_with_sh.ps
- for RNA scores
  - qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q.eqrna.ID[100:0].GC[100:0].id.spreadrna_histo_with_sh.ps
  - qrna-2.0.3c/Demos/5S_arch.stk.fa.blast.D10.q.eqrna.ID[100:0].GC[100:0].id.spreadrna_scores_with_sh.ps

For each kind of score (coding or RNA), the two postscript files include the following information,

- The first postscript file is a histogram of the number of coding (RNA) scoring windows above a chosen cutoff (zero bits by default). See Figure 1 for an example.
- The second files produces a distribution of the coding (RNA) scores (average with standard deviation) with the percentage identity of the alignments. See Figure 2 for an example.

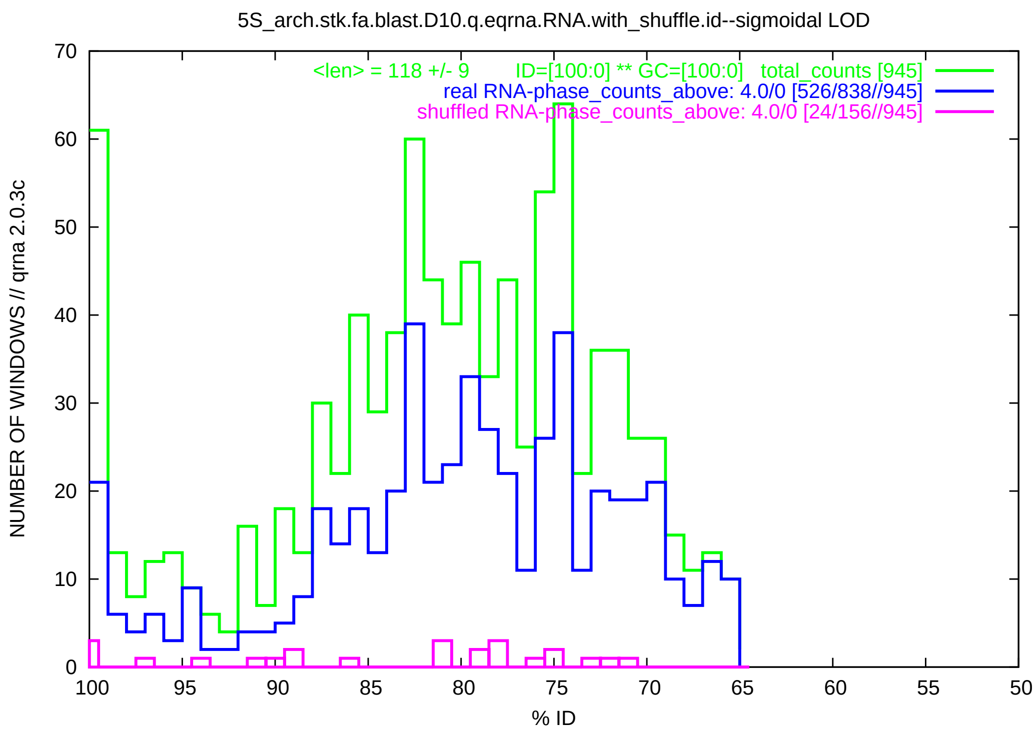

Figure 1: Analysis of the collection of 945 BLASTN alignments between 5S archaeal rRNAs contained in file “5S_arch.stk.fa.blast.D10.q”. Alignments have been grouped by percentage identity. This figure represent the histogram of the number of alignments bined in each percentage identity interval. In green we find the histogram for the total number of windows analyzed. In blue we have the histogram for those windows that score as RNA above a cutoff of 4.0 bits. In red we find the histogram for the windows that score as RNA above a cutoff of 4.0 bits after shuffling the alignments (false positives).

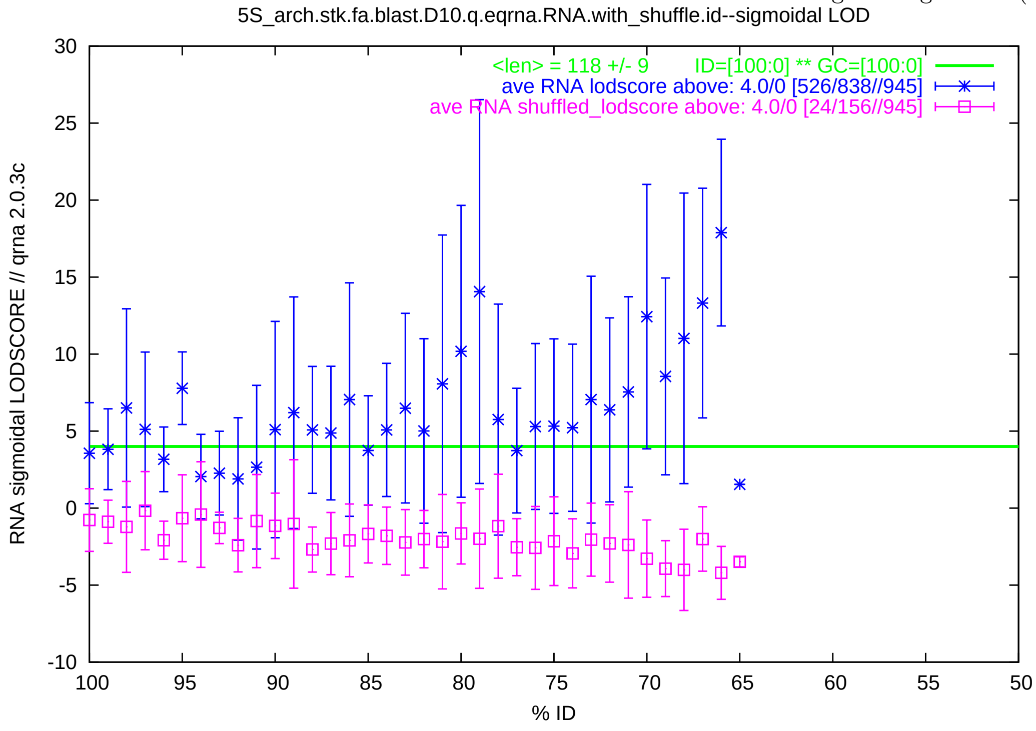

Figure 2: Analysis of the collection of 945 BLASTN alignments between 5S archaeal rRNAs contained in file “5S_arch.stk.fa.blast.D10.q”. Alignments have been grouped by percentage identity. This figure represent the RNA scores of all the alignments as a function of the percentage identity of the alignments. “$*$” represents the average of the RNA scores. “□” represents the average of the RNA scores of the corresponding shuffled alignments (false positives). The error bars correspond to one standard deviation.