Chapter 9

Finding Instances of Riboswitches and Ribozymes by Homology Search of Structured RNA with Infernal

Amell El Korbi, Jonathan Ouellet, Mohammad Reza Naghdi, and Jonathan Perreault

Abstract

In the genomics era, computational tools are essential to extract information from sequences and annotate them to allow easy access to genes. Fortunately, many of these tools are now part of standard pipelines. As a consequence, a cornucopia of genomic features is available in multiple databases. Nevertheless, as novel genomes are sequenced and new structured RNAs are discovered, homology searches and additional analyses need to be performed. In this chapter, we propose simple ways of finding instances of riboswitches and ribozymes in databases or in unannotated genomes, as well as ways of finding variants that deviate from the typical consensus.

Key words ncRNA, Noncoding RNA, Infernal, Covariation, Homology search, RNA structure, Secondary structure, Riboswitches, Ribozymes

1 Introduction

The diversity of roles attributed to noncoding RNA (ncRNA) has increased at a rapid pace in the last decade. As additional classes of RNAs were discovered and studied, so were their structures [1]. At the same time, sequence databases have grown exponentially, largely due to next-generation sequencing technologies. Public databases such as GenBank [2] or the metagenome-focussed CAMERA [3] database provide incredible opportunities to discover functional structured RNAs with computational screening which have proven extraordinarily useful for many ground-breaking discoveries of new ncRNAs [4–7].

Several computational methods have been used to that end. Some of the most commonly used tools for de novo prediction of ncRNAs include Evofold [8], QRNA [9], RNAz [10–12], CMfinder [13], Dynalign [14], LocARNA [15], Pfold [16], and the Vienna RNA package [17]. A comprehensive list can also be found in this wikipedia page [18].

Daniel Lafontaine and Audrey Dubé (eds.), *Therapeutic Applications of Ribozymes and Riboswitches: Methods and Protocols*, vol. 1103, DOI 10.1007/978-1-62703-730-3_9,

These tools allowed many research groups to find various ncRNAs: small RNAs that base pair on multiple target mRNAs to inhibit gene expression [19], self-cleaving ribozymes [20, 21], and riboswitches that bind a metabolite with their aptamer domain to change gene expression through their expression platform [22].

In parallel to the blossoming field of ncRNAs, the increasing rate of DNA sequencing requires efficient methods to annotate known RNAs. This becomes as important as ever since some of these RNAs are used as antibiotic targets. Indeed, a few riboswitches are already known to be sensitive to natural antibiotics analogous to their ligand [23–27] and new ligand-analogs are being developed in order to overcome the increasing worldwide resistance of bacteria against antibiotics [28–30]. Therefore, finding all instances of a targeted RNA can help determine the sensitive pathogenic strains as well as potentially sensitive beneficial strains.

1.1 Browsing Rfam: The RNA Families Database

In that regard, the collection of RNA families database (Rfam [31]) is particularly useful. The Sanger Institute performs homology searches with the Infernal software suite [32] for all known RNA families, which includes riboswitches and ribozymes, to update the database approximately once a year. Therefore, the quickest way of finding a riboswitch that could be a potential antibiotic target in any organism is by browsing Rfam. The cutoff scores typically used by Rfam to accept a predicted RNA with a relatively complex structure, such as for riboswitches, are high enough that there are almost no false positives in microbial genomes. On this subject, readers might also be interested to look at other recent publications in Methods in Molecular Biology [33, 34].

1.2 Search for a Motif in New Genomes with Infernal

However, there are some instances where Rfam does not provide the information needed, particularly if a given riboswitch exists in a newly sequenced genome that has not been screened by Rfam yet. This paper aims to circumvent such problems by presenting a simple step-by-step approach to look at a genome and evaluate the presence of an ncRNA of interest within that unannotated genome. It is targeted towards a general audience with minimal bioinformatics skills, although some basic knowledge of shell command lines would be useful.

1.3 Search for Variants of a Known Motif in All Bacterial Genomes

Occasionally, researchers that are very knowledgeable about a specific riboswitch have reasons to hypothesize that more instances exist. For example, divergence from the structure consensus of a riboswitch class could prevent Infernal from finding such a riboswitch's sub-family. In such cases, it could be desirable to perform a new Infernal search with less stringent criteria to reveal these "hidden" hits [35]. While Rfam lists most of the instances of riboswitches and ribozymes that can easily be found with a relatively high confidence (low E-values), it can occasionally ignore cases

that diverge from the consensus. This has been previously illustrated several times, notably for the *glmS* and the hammerhead ribozymes [21, 35–40]. In the case of the *glmS* ribozyme, Infernal was used with a very high E-value tolerance, as high as 5,000 on all microbial genomes from NCBI's Refseq38 sequence dataset. The resulting hits therefore included a vast majority of spurious hits, but also a number of previously unannotated *glmS* ribozyme instances [35]. Homology searches with very relaxed parameters should not be performed on a routine basis, but rather if there are good hints that additional riboswitches or ribozymes could be found in this manner. Indications on how to manage such searches and the resulting hits will be provided in Subheading 3.3.

2 Materials

Infernal requires the Linux/UNIX system to run the program and is accessible on Janelia'server [41]. Another alternative is the use of Mac OS X, which is a certified UNIX platform (*see* **Note 1**).

3 Methods

3.1 Browsing Rfam: The RNA Families Database

The simplest way to verify the presence of a specific riboswitch in target bacteria is to look in Rfam. As long as the bacterial genomes have been sequenced and annotated by Rfam, browsing the genomes section [42] would allow anyone to rapidly find which of the known riboswitches are found in a given genome by examining the ncRNAs found in the "chromosomes" tabs. Conversely, browsing the "families" section provides a quick overview of all species that have a specific riboswitch within their genome. This could be especially useful in the context of the development of a new antibiotic to target only a desired group of bacteria and leave most of the natural microbiota intact. Of course, the presence of a riboswitch in a bacterial species does not warrant microbicidal effect of the newly made antibiotic compound. Indeed, studies have already shown potent compounds capable of binding a specific riboswitch to prevent gene regulation via a competition against its native ligand. This competition can affect the growth of some bacteria that have the riboswitch, while leaving others unscathed although the targeted riboswitch is present in both cases. Depending on which genes are regulated by these riboswitches, significant differences of sensitivity can be observed [29].

In the few cases where Rfam would not be useful, any sequence can be screened for the presence of riboswitches with Infernal, which is described in more detail in Subheadings 3.2 and 3.3.

3.2 Search for a Motif in New Genomes with Infernal

Most of what is described herein can be found with additional details in the Infernal user guide [43] and additional papers [34, 44]. The intent here is to provide inexperienced users a rapid start-up guide. For this tutorial, the motif of purine riboswitches is used as an example where Infernal builds a covariance model from an alignment with structural annotation.

The latest version of Infernal is available to download here [41]. At the time of writing, the latest release of Infernal is 1.0.2 (30 Oct 2009) [45]. Once the source file downloaded, expand the "tar file" at a convenient location. For a basic installation, execute the two commands "configure" and "make" from the "infernal-1.0.2" directory (*see* **Note 2**):

```
# ./configure
# make
```

To run the optional testsuite, execute the following command:

```
# make check
```

Once Infernal installed, the ncRNA covariance model is built. The first step is to generate the "purine" seed alignment in Stockholm format from Rfam at this location [46]. Once the alignment is generated, the file should be downloaded and saved under "purine.sto" (*see* **Note 3**) in the "infernal-1.0.2" directory. The Stockholm format describes the secondary structure of an RNA sequence alignment. Base pairs are annotated as "<" (for the opening base of the pair) and ">" (for the closing base). Other, base pair annotations such as (,), [,], {or} are also used sometimes for base pairs of stems enclosing a multistem junction. Single stranded regions are annotated with other characters, typically ".", but sometimes "_" for loops and "," for junctions. A similar notation is used in Infernal's output for a regular "cmsearch." For simplicity, we assume that all the following commands are executed from the infernal directory.

Build the "cm file" (covariance model) using the command "cmbuild":

```
# src/cmbuild purine.cm purine.sto
```

Execute the command "cmcalibrate" which may take more than 1 h:

```
# src/cmcalibrate purine.cm
```

To search for the presence of that purine motif in a new genome, copy the sequence file of the genome of interest (FASTA format) to the "infernal-1.0.2" directory and use:

```
# src/cmsearch purine.cm genome.fa
```

For the purpose of this example, the search will be performed in a known bacterial genome downloaded from NCBI. The genome sequence of *Bacillus subtilis* (in FASTA format) is downloaded from this link [47]. The file is downloaded by pressing the

"send" option while selecting "Destination file" and the "FASTA" format. Rename the file as "sequence_Bsubs.fa" and move it to the same directory as "purine.cm".

Execute the "cmsearch" command:

```
# src/cmsearch --ga purine.cm sequence_Bsubs. fa
```

where the --ga option sets the bit-score cutoff value as the one used by Rfam curators according to the "GA cutoff" value in the purine.sto file downloaded from Rfam. When a large number of hits are expected, "cmsearch" has the useful additional option --tabfile to get a tabular representation of the search results (*see* **Note 4**). However, the current example does not need the option since it is a simple search for one riboswitch in a single genome. To create an output file, the command line would be:

```
# src/cmsearch --ga purine.cm sequence_Bsubs.fa>output.txt
```

Below is the output.

```
# command:    src/cmsearch --ga purine.cm sequence_Bsubs.fa
CM: Purine
>gi|223666304|ref|NZ_CM000487.1|

  Plus strand results:

 Query = 1 - 102, Target = 697666 - 697767
 Score = 85.74, E = 1.742e-18, P = 7.494e-25, GC =  42

           ::::::::::::::::::(((((((((,,,<<<<<<<_______>>>>>>>,,,,,,,,<<<
         1 aaaaaaaaaaaaaaaaucacuCgUAUAAucccgggAAUAUGGcccgggaGUUUCUACCag 60
             A+ AAA+ AAAA A   :  C:UAUAAU  :GGGAAUAUGGCCC:  AGUUUCUACC:G
    697666 CAUGAAAUCAAAACACGACCUCAUAUAAUCUUGGGAAUAUGGCCCAUAAGUUUCUACCCG 697725

           <<<<_______>>>>>>>,,)))))))):::::::::::::::
        61 gcaaCCGUAAAuugccuGACUAcGagugaaauuauuaaaaau 102
           GCAACCGUAAAUUGCC:GACUA:G  : AAA U +U  A+A+
    697726 GCAACCGUAAAUUGCCGGA**C**UAUGCAGGAAAGUGAUCGAUAA 697767

 Query = 1 - 102, Target = 693731 - 693832
 Score = 85.00, E = 2.717e-18, P = 1.169e-24, GC =  38

           ::::::::::::::::::(((((((((,,,<<<<<<<_______>>>>>>>,,,,,,,,<<<
         1 aaaaaaaaaaaaaaaaucacuCgUAUAAucccgggAAUAUGGcccgggaGUUUCUACCag 60
           A AAA+ AAA+AA A+   : CGUAUAAU :CG GAAUAUGGC CG: AGU UCUACCA:
    693731 AGAAAUCAAAUAAGAUGAAUUCGUAUAAUCGCGGGAAUAUGGCUCGCAAGUCUCUACCAA 693790

           <<<<_______>>>>>>>,,)))))))):::::::::::::::
        61 gcaaCCGUAAAuugccuGACUAcGagugaaauuauuaaaaau 102
           GC ACCGUAAAU GC:UGACUACG :   A+UU UU+  ++U
    693791 GCUACCGUAAAUGGCUUGA**C**UACGUAAACAUUUCUUUCGUUU 693832

 Query = 1 - 102, Target = 4004455 - 4004556
 Score = 73.67, E = 2.539e-15, P = 1.092e-21, GC =  28

           ::::::::::::::::::(((((((((,,,<<<<<<<_______>>>>>>>,,,,,,,,<<<
         1 aaaaaaaaaaaaaaaaucacuCgUAUAAucccgggAAUAUGGcccgggaGUUUCUACCag 60
             A+ ++A AAAAA A +::CU:GUAUA  ::C:G AAUAUGG C:G:: GUUUCUACC::
   4004455 CAUCUUAGAAAAAGACAUUCUUGUAUAUGAUCAGUAAUAUGGUCUGAUUGUUUCUACCUA 4004514
```

```
           <<<<_______>>>>>>>,,)))))))):::::::::::::::
        61 gcaaCCGUAAAuugccuGACUAcGagugaaauuauuaaaaau 102
           G:  CCGUAAA  :C::GACUAC:AG::A +UU +++AAA+U
   4004515 GUAACCGUAAAAAACUAGACUACAAGAAAGUUUGAAUAAAUU 4004556

 Minus strand results:

Query = 1 - 102, Target = 2319369 - 2319270
Score = 82.19, E = 1.481e-17, P = 6.372e-24, GC =  46

           ::::::::::::::::::(((((((,,,<<<<<<<_______>>>>>>>,,,,,,,,<<<
         1 aaaaaaaaaaaaaaaaaucacuCgUAUAAucccgggAAUAUGGcccgggaGUUUCUACCag 60
            +A AA+A+AA+A  A+CAC:C:UAUAAU :CG:G AUAUGGC:CG: AGUUUCUACC:G
   2319369 UUACAAUAUAAUAGGAACACUCAUAUAAUCGCGUGGAUAUGGCACGCAAGUUUCUACCGG 2319310

           <<<<_______>>>>>>>,,)))))))):::::::::::::::
        61 gcaaCCGUAAAuugccuGACUAcGagugaaauuauuaaaaau 102
            CA CCGUAAA UG C:GACUA:G:GUGA  +++  AA
   2319309 GCA-CCGUAAA-UGUCCGACUAUGGGUGAGCAAUGGAACCGC 2319270

Query = 1 - 102, Target = 625950 - 625851
Score = 65.20, E = 4.193e-13, P = 1.804e-19, GC =  30

           ::::::::::::::::::(((((((,,,<<<<<<<_______>>>>>>>,,,,,,,,<<<
         1 aaaaaaaaaaaaaaaaaucacuCgUAUAAucccgggAAUAUGGcccgggaGUUUCUACCag 60
           AA++AAA+A  +A++AU   U:GUAUAA:C:C:: AAUAUGG ::G:G:GU UCUACCAG
    625950 AAUUAAAUAGCUAUUAUCACUUGUAUAACCUCAAUAAUAUGGUUUGAGGGUGUCUACCAG 625891

           <<<<_______>>>>>>>,,)))))))):::::::::::::::
        61 gcaaCCGUAAAuugccuGACUAcGagugaaauuauuaaaaau 102
           G:  CCGUAAA  :CCUGA UAC:A    + UU++U A A+U
    625890 GAA-CCGUAAA-AUCCUGAUUACAAAAUUUGUUUAUGACAUU 625851
```

The header (not shown here) has information on the version of Infernal, the files used and the run time. The next line is the name of the covariance model used (CM), followed by the sequence name in the FASTA file on another line. For files with multiple sequences in FASTA, the name of the sequence is displayed for each group of corresponding hits in Infernal's output. Also displayed is the strand polarity in which the RNA was found. The corresponding positions of the hits are indicated for the query (which is often the entire query, but can also be only a portion of it, especially in "local searches") and for the target sequence. An evaluation of the validity of the hit is shown on the next line, the "score" reflects how well the hit matches the model, while the "E-value" corresponds to the expected number of hits with that "score" (or better) in a random sequence of the same size as the one you are looking at, i.e., the number of expected false positives. Similarly, a hit with a low P-value has a low probability of being a false positive. Finally, "GC" corresponds to the GC-content (in percentage). The following lines correspond to pairwise alignments where the first line describes the secondary structure, the second line is the "query" sequence, and the third line highlights homologous regions with the fourth line, which is the sequence of the hit. In the case shown above, all the hits have very good E-values (approximately ranging from 10^{-13} to 10^{-18}). However, in the case of ambiguous hits, with E-values closer

to 1, manual inspection can provide the additional clues needed to confirm the presence of a riboswitch at that position. For example, a loop–loop base-pairing interaction forms between the two loops of the purine riboswitch (here, the loops are annotated with "_"). This feature is not evaluated by Infernal and therefore does not contribute to the E-value. Observing this interaction in a relatively poor hit, with an E-value of 0.2 for instance, would mean this hit is more likely to be a true riboswitch than suggested strictly by the E-value. For purine riboswitches, detailed knowledge of the riboswitch is also useful to discriminate between adenine and guanine riboswitches. Because these two differ by a single base, Infernal finds both types of riboswitches during the same search. In the output shown above, the first four hits are guanine riboswitches and the last one is an adenine riboswitch. The former have a "C" at the last base of the junction, while the adenine riboswitch has a "U" at that position (shown in bold in the alignments above and annotated with ",").

3.3 Search for Variants of a Known Motif in All Bacterial Genomes

The Infernal suite can be used to find atypical riboswitches or ribozymes, but if many genomes are evaluated for poor E-values, thousands of hits will be generated and will require a lot of CPU time (*see* **Note 1**). Afterwards, knowing the structure of the RNA in detail can help to sort through the haystack of hits that would ensue a search accepting E-values as high as 5,000. Evaluating the presence of pseudoknots or the relevance of the downstream gene being regulated by this RNA are examples of how one can judge whether hits are likely real ncRNAs. This entire process takes much more time than what is described in Subheading 3.2 and is not recommended for all homology searches. This approach is more feasible in a case where only a few genomes are to be scrutinized, although the E-value should be set closer to 1 since it corresponds to the number of false positives expected to be found with that score (or better) in a database of this size. Therefore, a "cmsearch" allowing a maximum E-value of 5 (1 is default) could be performed and the hits could be manually screened one by one with the criteria mentioned hereafter when specifically interested in the genome of one bacteria. However, before performing such a search on all microbial genomes (with a maximal E-value of 1,000 for instance), one should have a strong basis to believe more riboswitches or ribozymes can be found since it would be CPU-time intensive and would generate a lot of false positives requiring a lot of time to sort through.

The steps described in Subheading 3.2 are also valid for searches in all available genomes. However, to get hits with E-values as high as 100 (for example), the "-E" option with "cmsearch" is used (*see* **Note 4** for more information on "--tabfile").

```
# cmsearch -E 100 --tabfile results.tab purine.cm
sequence.fa
```

Where "sequence.fa" could be a large file with all microbial genomes (available from NCBI [48], note that this compressed file is approximately 2 Gb in size). Some adjustments can be useful to determine the best value for "-E" (*see* **Note 5**). Different softwares can help visualizing a large number of hits. In that regard, the "RALEE" major mode in "Emacs" (a text editor program) is very useful [49]. It can color Stockholm alignments according to conservation, stems, or covariation. Both "RALEE" [50] and Emacs [51] are available for any Operating System platform.

When sorting the hits, as many criteria as possible should ideally be used to distinguish potentially good hits from spurious ones. Here are a few noticeable features, that have already proven useful in other works [4, 5, 35]:

1. Pseudoknots: Infernal does not take the base pairs of pseudoknot in consideration. Thus, it cannot account for pseudoknots in its E-value, which means that manually confirming the presence of a known pseudoknot in the ncRNA greatly improves this hit's likelihood of being real.
2. Essential bases: when the structure has been studied enough to determine which bases are absolutely crucial for the RNA's function, the hits that do not have these bases can be considered as spurious. However, one must be careful with such criteria since an apparent deleterious mutation at a specific position could be compensated by different bases at other positions, as in the case of the core-conserved C3G8 base-pair within the hammerhead ribozyme core, which was sometime found to be U3A8 [38, 52].
3. Intergenic versus coding sequence (CDS): even though riboswitches could theoretically be found in coding sequences, to our knowledge there is no natural riboswitch found yet that is completely embedded in the coding sequence. Therefore, if the hit is in an intergenic region, it should be regarded as more likely to be real, and, conversely, as spurious if it is in a CDS.
4. Functional relevance: when the riboswitch's ligand is known, the connection with the genes is often obvious. For example, in the above list of purine riboswitches, a hypoxanthine/guanine permease can be found downstream of a hit, as well as other genes involved in purine synthesis for other hits. The absence of a clear connection between the candidate riboswitch and the function of the downstream gene does not automatically means a false positive, but an obvious connection does help for its validation.
5. Expression platforms: to exert their effect on expression, the aptamer portion of riboswitches is usually close (or even

overlapping) to a terminator (*see* **Note 6**) or ribosome binding site in 5′ untranslated regions (UTR) [53]. Furthermore, if the downstream gene is not in the same orientation as the putative RNA, it is unlikely to be a true riboswitch.

After discarding most false positives, the resulting alignment (or the complete alignment from Rfam) is likely to have subgroups that have some of the diverging positions in common. This subgroup can be used as a secondary alignment for a more specialized "cmsearch." A remarkable example of such a subgroup was noticed in the purine alignment and led to discovery of a novel deoxyguanosine riboswitch, although in that case the secondary alignment did not provide any additional hits [54].

In some cases where ample data is available on the structure requirements of a RNA, such as the hammerhead ribozyme, artificial alignments can be constructed to find a conformation expected to be functional but not known in nature (*see* **Note 7**). For example, the following Stockholm alignment combines two structural types of hammerhead ribozyme sequences to form a hypothetical type, for which artificial constructs were known to be active but not known in nature until recently [21, 37–40]. The types of the hammerhead ribozymes are defined by the identity of the closing stem, the two others being simple hairpins. Here, alignments of type 1 and 3 were combined to search for type 2 hammerhead ribozymes (i.e., where stem 1 and 3 are simple hairpins and stem 2 is the closing stem).

```
# STOCKHOLM 1.0
#=GF ID Hammerhead_2_synthetic_alignment
J1           UGUCC..GAAACGCU.....GCGA.......AGCGUCUAGGC...GUUAU......GCCUACUGAUGA...GGAC
J2           AGGAU..GAAACCAUAC...CAUAGU...GUAUGGUCGGAUAA..UAUUUUU...UUAUCCCUGAUGA...AUCC
J3           UGUCC..GAAA.........CUCG...........UCUGCCCC..UGAUGCC...GGGGCACUGACGAU..GGAC
N1           CGUCAG.GAAACC.......ACUA.........GGUCUGCCCC..UGAUGCC...GGGGCACUGACGAU.CUGAC
N2           .GCUGG.GAAACGUC.....AACA.......GACGUCGUGAUC..UGAAACUC..GAUCACCUGAUGA..CCGGC
N3           UGAC...GAAACACCAACAGUGGGGGCUGUUGGUGUCUGAGCG..UGAUACC...CGCUCACUGAAGAU...GUC
N4           UGAC...GAAACAUCAACAGUGGGGGCUGUUGGUGUUUCGAG...CCA........CACGGCUGAUGAA...GUC
N5           UGCCAG.GAAACC.......CAAUU........GGUUUCGAGC..ACACCG....GCUCUGCUGAUGAA.CUGGC
N6           UGUUC..GAAA.........CUAG...........UCCAGGG...UUA........CCCUGCUGACGA...GAAC
N7           .AUCGG.GAAACGC......UUGG.........GCGUCUGGACC..UGAUGCC...GGUCCACUGAUGA..CCGAU
N8           AGCCGG.GAAACCU......UAAC........AGGUCUCCAGAU.GUGU.....GUCUGGACUGAUGA..CCGGC
N9           .GCUGGUGAAACGUC.....AAUC.......GACGUUAAAUG...UCAUACA....CAUUUCUGAGGA.ACCGGC
N10          CGACUAUGAAACCUCU....UUUAU.....AGAGGUUUCGAGC..ACACCG....GCUCUGCUGAUGAAAUAGUC
N11          NGUCC..GAAAC........GUGC..........GUCCAGGG...UUA........CCCUGCUGACGA...GGAC
N12          GGUAU..GAAAAA.......AAAU.........UUUCCGGGAG..UAAUGCU...CUCCCGCUGAUGAU..AUAC
N13          CGA....GAAACAG......GGC.........CUGUUAAAUG...UCAUACA....CAUUUCUGAGGA.....UC
N14          CAUUU..GAAAUAGC.....CAA........GCUGUCGGAAA...GUGUGCGC...UUUCCCUGAUGA...AAAU
N15          AGCUC..GAAACC.......UGUUG........GGUUUCGAG...CCA........CACGGCUGAUGAA..GAGC
N16          CGUCC..GAAAC........CCCA..........GUAUCGGGAU.UGUG.....GUCCCGGCUGAUGA...GGGC
#=GC SS_cons .<<<<<<...<<<<<<<<<<......>>>>>>>>>>.<<<<<<<<........>>>>>>>>........>>>>>>
//
```

After building and calibrating this model, a "cmsearch" can be performed (as seen in Subheading 3.2). Here is a portion of the results of a search against the chromosome 1 of Agrobacterium tumefaciens (NC_003062) [55].

```
CM: Hammerhead_2_synthetic_alignment
>gi|159184118|ref|NC_003062.2|

  Plus strand results:

 Query = 1 - 51, Target = 1183818 - 1183877
 Score = 28.65, E = 0.0002488, P = 2.328e-10, GC =  65

           :((((,,,<<<<......____..>>>>,<<<<<<._______.>>>>>>,,,,,,,,.))
         1 uGuCcGAAACau......uaaa..auGUCuggggc.UgAUaCu.gccccaCUGAUGA.gG 49
           U::C GAAAC::      UA    ::GUC: GGG: U+    U :CCC :CUGAUGA  G
   1183818 UACC-GAAACCGgcucccUAGGguCGGUCGUGGGGcUUGGCAUgCCCCGCCUGAUGAu-G 1183875

           ))
        50 aC 51
           ::
   1183876 GU 1183877


 Query = 1 - 51, Target = 1956021 - 1956068
 Score = 19.83, E = 0.09822, P = 9.192e-08, GC =  52

           :((((,,,<<<<____.....>>>>,<<<<<<_______>>>>>>,,,,,,.,,))))
         1 uGuCcGAAACauuaaa.....auGUCuggggcUgAUaCugccccaCUGAU.GAgGaC 51
           +GUCCGAAA : + A       : U :GG:::       :::CC:CUGAU GAGGAC
   1956021 AGUCCGAAAUC-AUAGaucag-GGUACGGUCA-------UGACCGCUGAUcGAGGAC 1956068


...
```

The first hit is a confirmed ribozyme [38, 40], while the second is more likely to be a false positive because the stem 3 is weak (due to a stem of only 3 base pairs) and there is a "C" insertion in the core (lower case "c" in "CUGAUcGA"). Note that even if this approach works relatively well, the type 2 hammerheads have been discovered by other methods [21]. Also, this approach can only be used for a few RNAs for which the structure is very well understood, which is necessary to manipulate sequence alignments in order to carefully simulate a new structural version of a known ncRNA.

4 Notes

1. While searching a single genome for a ribozyme or riboswitch is typically done within a few hours on a desktop computer (3 GHz for example). It would take several weeks to look at all available genomes. Running the Infernal software on a computer cluster is preferable to perform multiple "cmsearches" on all the sequenced microbes.
2. Using a Mac, the GNU C compiler gcc is not installed by default. Installation of Xcode will resolve this.
3. Alternatively, pre-built and pre-calibrated models are available from Rfam here [56]. The "cm" corresponding to the RNA of

interest can easily be copied and pasted as a text file for which the extension has to be changed to ".cm" before using it.

4. The --tabfile option allows the use of the tab file as an input to the "Easel" miniapp "esl-sfetch" (found in a subdirectory of Infernal). The miniapp "esl-sfetch" extracts the sequences of all hits found from the genome sequence file to a new FASTA file. This file is useful to get a new alignment with the CM file of the motif ("purine.cm" in the example) using the command "cmalign." To get a tabular version of the search results, the command line is:

```
# cmsearch --ga --tabfile results.tab purine.cm sequence_Bsubs.fa
```

Now, to use the tabfile "results.tab" as an input to fetch the hits sequences:

```
# easel/miniapps/esl-sfetch -C -f --tabfile sequence_Bsubs.fa results.tab
```

An error of this type: "`Failed to open SSI index`" may occur. In this case the "sequence_Bsubs.fa" (or the file containing the new genome) has to be indexed. This is done with this step:

```
# easel/miniapps/esl-sfetch --index sequence_Bsubs.fa
```

Now that the file containing the genome sequence is indexed, the "sfetch" command can be re-executed as above. The hits sequences are displayed in FASTA format. To get the output in a new FASTA file, the command line would be:

```
# easel/miniapps/esl-sfetch -C -f --tabfile sequence_Bsubs.fa results.
  tab>hitsequences.fa
```

The tabular version has the format shown below.

```
# model name  target name        start        stop  start   stop    bit sc   E-value  GC%
# ----------  -------------  ---------  ----------  -----  -----  --------  --------  ---
  Purine      gi|223666304|     697666      697767      1    102     85.74  1.74e-18   42
  Purine      gi|223666304|     693731      693832      1    102     85.00  2.72e-18   38
  Purine      gi|223666304|    4004455     4004556      1    102     73.67  2.54e-15   28
  Purine      gi|223666304|    2319369     2319270      1    102     82.19  1.48e-17   46
  Purine      gi|223666304|     625950      625851      1    102     65.20  4.19e-13   30
```

Shown below is the beginning of the file "hitsequences.fa" containing the sequences of the hits found in the *Bacillus subtilis* genome:

```
>gi|223666304|ref|NZ_CM000487.1|/697666-697767/Purine/B85.74/E1.7e-1
8/GC42 Bacillus subtilis subsp. subtilis str. 168 chromosome, whole g
enome shotgun sequence
CATGAAATCAAAACACGACCTCATATAATCTTGGGAATATGGCCCATAAGTTTCTACCCG
GCAACCGTAAATTGCCGGACTATGCAGGAAAGTGATCGATAA
>gi|223666304|ref|NZ_CM000487.1|/693731-693832/Purine/B85.00/E2.7e-
18/GC38 Bacillus subtilis subsp. subtilis str. 168 chromosome, whole g
enome shotgun sequence
AGAAATCAAATAAGATGAATTCGTATAATCGCGGGAATATGGCTCGCAAGTCTCTACCAA
GCTACCGTAAATGGCTTGACTACGTAAACATTTCTTTCGTTT
...
```

These sequences are aligned using the "purine.cm" as a seed to obtain a new motif that can be used in further searches:

```
# src/cmalign purine.cm hitsequences.fa
```

This should give the following (the output was slightly modified to fit on the page):

```
# STOCKHOLM 1.0
#=GF AU Infernal 1.0.2

gi|223666304|...       CAUGAAAUCAAAACACGACCUCAUAUAAUCUUGGGAAUAUGGCCCAUAAG
gi|223666304|...       AGAAAUCAAAUAAGAUGAAUUCGUAUAAUCGCGGGAAUAUGGCUCGCAAG
gi|223666304|...       CAUCUUAGAAAAAGACAUUCUUGUAUAUGAUCAGUAAUAUGGUCUGAUUG
gi|223666304|...       UUACAAUAUAAUAGGAACACUCAUAUAAUCGCGUGGAUAUGGCACGCAAG
gi|223666304|...       AAUUAAAUAGCUAUUAUCACUUGUAUAACCUCAAUAAUAUGGUUUGAGGG
#=GC SS_cons           ::::::::::::::::::(((((((,,,<<<<<<<_______>>>>>>>,
#=GC RF                aaaaaaaaaaaaaaaaucacuCgUAUAAucccgggAAUAUGGcccgggaG

gi|223666304|...       UUUCUACCCGGCAACCGUAAAUUGCCGGACUAUGCAGGAAAGUGAUCGAU
gi|223666304|...       UCUCUACCAAGCUACCGUAAAUGGCUUGACUACGUAAACAUUUCUUUCGU
gi|223666304|...       UUUCUACCUAGUAACCGUAAAAAACUAGACUACAAGAAAGUUUGAAUAAA
gi|223666304|...       UUUCUACCGGGCA-CCGUAAA-UGUCCGACUAUGGGUGAGCAAUGGAACC
gi|223666304|...       UGUCUACCAGGAA-CCGUAAA-AUCCUGAUUACAAAAUUUGUUUAUGACA
#=GC SS_cons           ,,,,,,,<<<<<<<_______>>>>>>>,,)))))))):::::::::::::
#=GC RF                UUUCUACCaggcaaCCGUAAAuugccuGACUAcGagugaaauuauuaaaa

gi|223666304|...       AA
gi|223666304|...       UU
gi|223666304|...       UU
gi|223666304|...       GC
gi|223666304|...       UU
#=GC SS_cons           ::
#=GC RF                au
```

5. To compromise between the number of spurious hits and the potential of getting new valid hits, it can be useful to test a few different E-values. Ideally, an instance that diverges from the consensus and that is known (or strongly suspected) to be a true riboswitch should be used. In that case, you can always expect to find more hits by increasing the E-value until this hit is also detected.
6. Some tools have been developed to detect terminators, "RNIE" being the most up-to-date and is available here [57, 58]. "RNIE" works with the Infernal suite, it can be used to find all transcription terminators in a genome with the following command:

```
# src/cmsearch -T 16 -g --fil-no-qdb --fil-T-hmm 2 --cyk --beta 0.05 CM
  sequence.fa
```

7. Alignments can be modified in many ways. In the case presented above, the first half of stem 2, the GAAA and stem 3 were taken from a type 1 alignment, while stem 1 and the CUGANGA were taken from a stem 3 alignment. A simpler modification that could be made to an alignment could be to change the bases of a conserved pseudoknot with the hopes that the CM built from that new "synthetic alignment" would

find instances that have been missed because their pseudoknot's sequence diverges from the current model.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) discovery grant (RGPIN 418240-2012) and by a grant from The Banting Research Foundation to JP.

References

1. Wan Y, Kertesz M, Spitale RC, Segal E, Chang HY (2011) Nat Rev Genet 12:641–655
2. Benson DA, Karsch-Mizrachi I, Clark K, Lipman DJ, Ostell J, Sayers EW (2012) Nucleic Acids Res 40:D48–D53
3. Sun S, Chen J, Li W, Altintas I, Lin A, Peltier S, Stocks K, Allen EE, Ellisman M, Grethe J, Wooley J (2011) Nucleic Acids Res 39:D546–D551
4. Weinberg Z, Wang JX, Bogue J, Yang J, Corbino K, Moy RH, Breaker RR (2010) Genome Biol 11:R31
5. Weinberg Z, Perreault J, Meyer MM, Breaker RR (2009) Nature 462:656–659
6. Shi Y, Tyson GW, DeLong EF (2009) Nature 459:266–269
7. Livny J, Waldor MK (2007) Curr Opin Microbiol 10:96–101
8. Pedersen JS, Bejerano G, Siepel A, Rosenbloom K, Lindblad-Toh K, Lander ES, Kent J, Miller W, Haussler D (2006) PLoS Comput Biol 2:e33
9. Rivas E, Eddy SR (2001) BMC Bioinformatics 2:8
10. Washietl S (2007) Methods Mol Biol 395:503–526
11. Gruber AR, Neubock R, Hofacker IL, Washietl S (2007) Nucleic Acids Res 35:W335–W338
12. Washietl S, Hofacker IL, Stadler PF (2005) Proc Natl Acad Sci U S A 102:2454–2459
13. Yao Z, Weinberg Z, Ruzzo WL (2006) Bioinformatics 22:445–452
14. Harmanci AO, Sharma G, Mathews DH (2007) BMC Bioinformatics 8:130
15. Will S, Reiche K, Hofacker IL, Stadler PF, Backofen R (2007) PLoS Comput Biol 3:e65
16. Knudsen B, Hein J (2003) Nucleic Acids Res 31:3423–3428
17. Lorenz R, Bernhart SH, Honer Zu Siederdissen C, Tafer H, Flamm C, Stadler PF, Hofacker IL (2011) Algorithms Mol Biol 6:26
18. http://en.wikipedia.org/wiki/List_of_RNA_structure_prediction_software
19. Storz G, Vogel J, Wassarman KM (2011) Mol Cell 43:880–891
20. Cochrane JC, Strobel SA (2008) Acc Chem Res 41:1027–1035
21. Hammann C, Luptak A, Perreault J, de la Pena M (2012) RNA 18:871–885
22. Roth A, Breaker RR (2009) Annu Rev Biochem 78:305–334
23. Lee ER, Blount KF, Breaker RR (2009) RNA Biol 6:187–194
24. Blount KF, Wang JX, Lim J, Sudarsan N, Breaker RR (2007) Nat Chem Biol 3:44–49
25. Blount KF, Breaker RR (2006) Nat Biotechnol 24:1558–1564
26. Sudarsan N, Cohen-Chalamish S, Nakamura S, Emilsson GM, Breaker RR (2005) Chem Biol 12:1325–1335
27. Ott E, Stolz J, Lehmann M, Mack M (2009) RNA Biol 6:276–280
28. Kim JN, Blount KF, Puskarz I, Lim J, Link KH, Breaker RR (2009) ACS Chem Biol 4:915–927
29. Mulhbacher J, Brouillette E, Allard M, Fortier LC, Malouin F, Lafontaine DA (2010) PLoS Pathog 6:e1000865
30. Lunse CE, Schmidt MS, Wittmann V, Mayer G (2011) ACS Chem Biol 6:675–678
31. http://rfam.sanger.ac.uk
32. Nawrocki EP, Kolbe DL, Eddy SR (2009) Bioinformatics 25:1335–1337
33. Hoeppner MP, Barquist L, Gardner PP. Methods in Molecular Biology (In press)
34. Barquist L, Burge SW, Gardner PP. Methods in Molecular Biology (In press)
35. McCown PJ, Roth A, Breaker RR (2011) RNA 17:728–736

36. de la Pena M, Garcia-Robles I (2010) EMBO Rep 11:711–716
37. Jimenez RM, Delwart E, Luptak A (2011) J Biol Chem 286:7737–7743
38. Perreault J, Weinberg Z, Roth A, Popescu O, Chartrand P, Ferbeyre G, Breaker RR (2011) PLoS Comput Biol 7:e1002031
39. Seehafer C, Kalweit A, Steger G, Graf S, Hammann C (2011) RNA 17:21–26
40. de la Pena M, Garcia-Robles I (2010) RNA 16:1943–1950
41. http://infernal.janelia.org/
42. http://rfam.sanger.ac.uk/genome/browse#A
43. Nawrocki EP, Kolbe DL, Eddy SD (2009) ftp://selab.janelia.org/pub/software/infernal/Userguide.pdf
44. Nawrocki EP. Methods in Molecular Biology (In press)
45. ftp://selab.janelia.org/pub/software/infernal/infernal-1.0.2.tar.gz
46. http://rfam.sanger.ac.uk/family/RF00167#tabview=tab2
47. http://www.ncbi.nlm.nih.gov/nuccore/223666304
48. ftp://ftp.ncbi.nih.gov/genomes/Bacteria/all.fna.tar.gz
49. Griffiths-Jones S (2005) Bioinformatics 21:257–259
50. http://personalpages.manchester.ac.uk/staff/sam.griffiths-jones/software/ralee/
51. http://www.gnu.org/software/emacs
52. Przybilski R, Hammann C (2007) RNA 13:1625–1630
53. Barrick JE, Breaker RR (2007) Genome Biol 8:R239
54. Kim JN, Roth A, Breaker RR (2007) Proc Natl Acad Sci U S A 104:16092–16097
55. http://www.ncbi.nlm.nih.gov/nuccore/159184118?report=fasta
56. ftp://ftp.sanger.ac.uk/pub/databases/Rfam/CURRENT/Rfam.cm.gz
57. http://github.com/ppgardne/RNIE
58. Gardner PP, Barquist L, Bateman A, Nawrocki EP, Weinberg Z (2011) Nucleic Acids Res 39:5845–5852