Supplementary Material

Four RNA families with functional transient structures

Jing Yun A. Zhu and Irmtraud M. Meyer

Centre for High-Throughput Biology and Department of Computer Science and
Department of Medical Genetics, University of British Columbia,
2125 East Mall, Vancouver, BC,
Canada V6T 1Z4, irmtraud.meyer@cantab.net

Oct 19, 2014

1 Data availability

The data generated or used by this project is available in the following folders:

- The curated alignments in stockholm format with structural annotation are in the folder alignments_annotated.

  1. HDV ribozyme: HDV.stk
  2. Levivirus 5'UTR of maturation protein: levivirus.stk
  3. SAM riboswitch: samRiboswitch.stk
  4. Trp operon leader: trpOperonLeader.stk

- The CM(covariation model) files built and calibrated from the curated alignments above are in folder CM_files. For each family, the cm files for all alternative structures are included. However, the pseudoknotted hairpin is removed from both HDV ribozyme (the active structure) and SAM riboswitch (the SAM-bound structure) as INFERNAL[1] does not handle pseudoknotted structure. In the active structure of HDV ribozyme, a 6-bp hairpin and a 2-bp hairpin located near the 5' end, causing pseudoknot, are removed; thus, the cm file is made of the remaining active structure. In the SAM-bound structure of SAM riboswitch, a 4-bp hairpin causing pseudoknot is removed, and the cm file is made of the remaining SAM-bound structure.

   1. HDV ribozyme: HDV_active.cm, HDV_alternative1.cm, HDV_alternative2.cm
   2. Levivirus 5'UTR of maturation protein: levivirus_transient.cm, levivirus_finalInactive.cm
   3. SAM riboswitch: samRiboswitch_SAM_bound.cm, samRiboswitch_SAM_unbound.cm
   4. Trp operon leader: trpOperonLeader_terminator.cm, trpOperonLeader_antiterminator.cm

- The initial (ungapped) structures directly read off figures from literature consulted and the corresponding (ungapped) reference sequences are in folder initial_structure_refSeq:
   1. HDV ribozyme: HDV.fasta
   2. Levivirus 5'UTR of maturation protein: levivirus.fasta
   3. SAM riboswitch: samRiboswitch.fasta
   4. Trp operon leader: trpOperonLeader.fasta

2 Source literature of the structural annotations

In this study, the structures and the corresponding ungapped reference sequences are directly and manually read off figures from literature consulted. Such structures are then mapped onto our alignment based on the reference sequence. The consulted papers and the corresponding figures wherein the dot-bracket structures are extracted are shown in the table below:

Structure	Source literature
Trp operon leader (terminator)	Figure 2, 4 in Yanofsky (1981)[2]; Figure 2, 3 in Kolter and Yanofsky (1982)[3]
Trp operon leader (anti-terminator)	Figure 2, 4 in Yanofsky (1981)[2]; Figure 2, 3 in Kolter and Yanofsky (1982)[3]
Levivirus (final inactive)	Figure 2 in Groeneveld *et al.* (1995)[4]
Levivirus (transient)	Figure 1 in Meerten *et al.* (2001)[5]
HDV (active)	Figure 1, 2 in Ferre-D'Amare *et al.*(1998)[6]
HDV (Alternative 1)	Figure 1 in Chadalavada *et al.* (2000)[7]
Chadalavada2002 HDV (Alternative 2)	Figure 6, 7 in Chadalavada *et al.* (2000)[7]
SAM (SAM-bound)	Figure 5 in Winkler *et al.* (2003)[8] and Rfam[9]
SAM (SAM-unbound)	Figure 5 in Winkler *et al.* (2003)[8] and Rfam[9]

Table 1: The source literatures where the transient/alternative and dominant structures are derived are organized in this table. For each literature, the exact figure number where the dot-bracket structure and the representative sequence are read off is clarified.

3 Programs and Parameters

This section describes what functions are used in each program, and the corresponding parameters are included if there is any.

3.1 Infernal[1]

The version of Infernal used is infernal-1.1rc2.

3.1.1 Building the primary or secondary Covariation Model

Two functions from Infernal[1] are used to make cm files.

- cmbuild: we use the default setting, and run the function as :

  cmbuild -o ⟨stockholm file storing the annotated alignment returned by cmbuild⟩ -F ⟨cm file for the output Covariation Model⟩ ⟨stockholm file for the input alignment⟩

- cmcalibrate: we use the default setting, and run the function as:

  cmcalibrate ⟨cm file for the Covariation Model to be calibrated⟩

3.1.2 Selecting related sequences to add into the small MSA

Two functions from Infernal[1] are used to search for and align new related sequences so as to add into the high-quality MSA.

- cmsearch: we use the default setting, and run the function as :

  cmsearch –tblout ⟨hits returned by the search in a tab format, with score, E-value, and *etc.*⟩ -o ⟨detailed information about the hits, and the hits are ranked in this file⟩ -A ⟨alignment of the searched sequences⟩ –verbose ⟨cm file for the Covariation Model⟩ ⟨fasta file for database to search⟩

- cmalign: we use the default setting, and run the function as:

  cmalign -o ⟨stockholm alignment of all the sequences using the given Covariation Model⟩ –mapali ⟨cm file used to align the new sequences⟩ ⟨fasta file storing the new sequences⟩

3.2 USEARCH[10]

Two functions from USEARCH[10] are used to cluster sequences based on their primary sequence:

- Sort by sequence length:

  usearch –sort ⟨input fasta alignment⟩ –output ⟨output sorted alignment⟩

- Cluster according to (ungapped) primary sequence:

  usearch –cluster ⟨fasta file of the sorted alignment⟩ –uc ⟨the clustered results, with all the clades and the subordinate sequences⟩ –id ⟨Cutoff for the primary sequence similarity⟩

  For each RNA family, we tried cutoffs from about 90% to 98% until the clustering step generates about 50 to 100 clusters which is a tangible number of clusters to work on in the subsequent selection step.

3.3 Ranking Sequences

We use some home-made perl scripts to rank the alignment sequences based on their structural fitting in accordance with the reference sequence, contribution to the covariation, gappiness, and etc. There are some score numbers that we used in the scripts during evaluating and ranking the covariation score of sequences. These scores are assigned to one-sided covariation, double-sided covariation, mismatch, gappiness etc. However, due to the difference in the phylogenetic diversity of alignments, there is no universal set of scores working for them all. Instead, we had to adjust these scores frequently until the resultant alignment is satisfactory, as visually contemplated in Ralee[11]. As a starting point for such ranking process, the following set of score could be used initially (and might be substantially adjusted later). Users can then recursively adjust these scores based on visual observation using Ralee[11]:

- nucleotides of base pair match the reference sequence: +1
- nucleotides of base pair has one-sided covariation: +1.5
- nucleotides of base pair has double-sided covariation: +2
- nucleotides of base pair has (gap-free) non-canonical base pair: -1
- nucleotides of base pair has one-sided gap: -1.5
- nucleotides of base pair has double-sided gap: -1

4 Manual Curation

As introduced in Programs and Parameters (section Infernal[1]), running the cmalign and cmsearch functions generates alignment of which the quality need to be improved via manual curation in order to serve the subsequent step (Materials and Methods in the main manuscript). The manual curation is done on

Ralee[11] for each alignment sequence. The arduous process of manual curation requires judgments based on experimental evidence derived from the literature consulted for each family.

4.1 Steps of Manual Curation

During manual curation, the goal is to optimally align the sequences against the structure and correct any obvious misalignment based on the literature consulted. (i) convert the raw alignment generated by cmalign/cmsearch to stockholm format, and display it in Ralee[11] which will color the alignment sequences based on their structural fitting against the structure (ii) zoom into the mismatching nucleotides in each sequence. If it's obvious that they are misaligned and should be aligned to adjoining region, then push them to the neighboring position. Homologous regions must be aligned based on their primary sequence. They cannot be shuffled around merely to satisfy the reference structure if they are obvious to be homologous to a neighbouring region and not to the reference helix region. (iii) cmalign/cmsearch is operated using cm files built from each of the alternative structures for the families, respectively. Thus, each alignment is specific to only one structure. For each RNA family, repeat (i) and (ii) for all alternative structures, respectively. (iv) for each RNA family, compare the curated alignments(*i.e.* after (ii)) pertaining to each alternative structure, and identify sequence regions where both of the alternative structures involve. We call such region structural overlapping regions. After being identified, structural overlapping regions could be manually improved to make them simultaneously fit both structures better. This is done by correcting obvious misaligned nucleotides, and inserting gap columns if necessary to make this region more flexible to accommodate both structures.

During this manual curation, it is noticed, in only rare occasion, that some alternative but conserved homologous helices can be flexible in terms of size, or their position relative to the dominant structure. Some of such sequences are not included in our alignment, and this type of flexible conservation cannot be captured in alignment format.

4.2 Post-processing

Once the manual curation is done for all the structural regions of the alignment sequences as explained above, we use home-made perl scripts to splice out the unstructured (based on the reference structure) region of sequences in an alignment, and realign them via MUSCLE[12]. Then the realigned unstructured regions and the curated structural regions are stitched together. The all-gap columns are subsequently removed.

For accuracy checking, the alignment sequences are ungapped and matched against NCBI database to make sure no base is altered during this curation process. This checking process also ensures the accuracy of the starting and ending coordinates of the sequence on the genome with the given accession.

References

1. Nawrocki, E. P. & Eddy, S. R. Infernal 1.1: 100-fold faster RNA homology searches. Bioinformatics 2013; 29:2933–2935.

2. Yanofsky, C. Attenuation in the control of expression of bacterial operons. Nature 1981; 289:751–758.

3. Kolter, R. & Yanofsky, C. Attenuation in amino acid biosynthetic operons. Annu Rev Genet 1982; 16:113–34.

4. Groeneveld, H., Thimon, K. & van Duin, J. Translational control of maturation-protein synthesis in phage MS2: a role for the kinetics of RNA folding? RNA 1995; 1:79–88.

5. Van Meerten, D., Girard, G. & Van Duin, J. Translational control by delayed RNA folding: Identification of the kinetic trap. RNA 2001; 7:483–494.

6. Ferre-D'Amare, A. R., Zhou, K. & Doudna, J. A. Crystal structure of a hepatitis delta virus ribozyme. Nature 1998; 395:567–74.

7. Chadalavada, D. M., Knudsen, S. M., Nakano, S. & Bevilacqua, P. C. A role for upstream RNA structure in facilitating the catalytic fold of the genomic hepatitis delta virus ribozyme. J Mol Biol 2000; 301:349–367.

8. Winkler, W. C., Nahvi, A., Sudarsan, N., Barrick, J. E. & Breaker, R. R. An mRNA structure that controls gene expression by binding S-adenosylmethionine. Nature Structural Biology 2003; 10:701–707.

9. Burge, S. W. *et al.* Rfam 11.0: 10 years of RNA families. Nucleic Acids Res 2013; 41:D226–32.

10. Edgar, R. C. Search and clustering orders of magnitude faster than blast. Bioinformatics 2010; 26:2460–2461.

11. Griffiths-Jones, S. RALEE–RNA ALignment editor in Emacs. Bioinformatics 2005; 21:257–259.

12. Edgar, R. C MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research 2004; 32:1792–1797.