Prediction of Structural Elements in Long Non-Coding RNAs using *RNAz*

Kaswara Kraibooj Dominic Rose

Bioinformatics Group
Department of Computer Science
Albert-Ludwigs-University Freiburg
Georges-Köhler-Allee 106
79110 Freiburg
Germany

kkraibooj1@yahoo.com
dominic@informatik.uni-freiburg.de

June 25, 2012

Abstract

In this paper we present an analysis of human long intergenic non-coding RNAs transcripts (8195 transcripts of hg19). The key problem of this kind of RNAs is that they do not have common statistically significant features in their primary sequence (e.g. open reading frames or codon bias). Therefore, the analysis was done by the tool *RNAz* which could solve this problem by employing comparative genomics and making use of two measurements: (1) The thermodynamic stability and (2) The structural conservation of long non-coding RNAs. *RNAz* detected 2700 sequences (loci) of high probability (>0.9) to have structural conserved structures and to be thermostatically stable. 1823 (67.51 %) of them are located in introns. 867 are located in exons and 197 (7.29 %) span splice sites. Then, using the tool *RNAclust.pl* we detected the loci which have common secondary structure motifs.

1 Introduction

Non-coding RNAs have a big spectrum of molecules which are heterogeneous in structure and function. While in general ncRNAs play important regulatory roles in the cell, their functions are very miscellaneous, and in most cases are not yet discovered. Most of them do not have the features which were exploited by efficient algorithms to predict the coding genes. However, some classes of them such as the small ones ($<$ 200 bp) could be predicted using their secondary structure.
In this paper, we present an analysis of the class ”lincRNAs”. **LincRNAs** [9] stand for **long** ($>$ 200 bp) intergenic non-coding RNAs. The expression **”intergenic”** refers to being transcribed from the non-coding DNA sequences. **Non coding** means this RNA does not code a protein.
Detecting substructures of lincRNA depends on the assumption that their function is defined by two main characteristics:
(1) Thermodynamic stability.
(2) Conservation of secondary structure.
The tool *RNAz* exploits these two features making use of comparative studies to achieve the prediction. The classification is done by a support vector machine learning algorithm (SVM) which is trained on a large set of well known ncRNAs.

General view of the analysis

We did the analysis by *RNAz* as the following: First, we aligned the transcripts. Then we used them as input to *RNAz* after preparing them by the tool *rnazWindow.pl*. Next, having applied *RNAz* with a probability class of more than 0.5, we got the hits which are called windows (retrieved from *rnazWindow.pl*). We used this class probability because the hits of probability more than 0.5 are considered as functional. Then we clustered these resulting hits (overlapping windows) using the tool *rnazCluster.pl* to loci. Before continuing, we measured the accuracy of the number of our hits. We did this by estimating the false discovery rate (FDR).
To calculate FDR we randomized the alignments using the tool *SISSIz* and run *RNAz* again on them. Comparing the resulting hits of this shuffled screen with the hits of the original one we could calculate the FDR. As a last step we used the tool [6] to detect similar loci of different transcripts.
We worked on two screens (named screen1 and screen2). The difference between them was that screen1 contains only exons and splice sites. But screen2 had contained exons, splice sites and introns. We made this difference by using two different methods of alignments. We used screen2 to make some comparisons with screen1. But we did not complete the analysis of screen2 since it is the same as screen1 except for the introns.
In the rest of this report: section 2 describes the data set, the tools and the methods used in the analysis; Section 3 describes the work flow of the analysis; Section 4 presents the results; Section 5 discusses the results.

2 Data set, Tools and Methods

2.1 Data set

The input is a BED-formatted file of 8195 human lincRNAs transcripts (hg19) [9]. The BED format is described on the web page [1].

2.2 *RNAz*

RNAz [11] is a package of tools. The main tool of this package is *RNAz*. Its input is an alignment and its output are the hits of some class probabilities. The class probability is specified by the parameter (-p). The other tools pre-process the input of *RNAz*, manipulate the output or perform other helpful functions.

2.3 *SISSIz*

The input of *SISSIz* [7] is an alignment. The output is a shuffled alignment. This tool preserves the dinucleotide content. In our project, we applied this tool to create a control screen. The control screen is needed for estimating the false discovery rate which reflects the precision of the number of hits gotten by *RNAz*.

2.4 *rnazWindow.pl*

This tool [12] is one of the tools of *RNAz*. Its input are the non-processed alignments. its output are windows. Windows represent an optimal input of *RNAz*. *rnazWindow.pl* does the following:
1.Reduces the gaps and repeats.
2.Splices the big alignments into small windows.
Gaps and repeats distort the analysis. *RNAz* can not analyze alignments which are too long. Additionally, we want to analyze local structure, not global ones.

2.5 FDR (False Discovery Rate)

FDR [4] measures the percentage of falsely expected elements of true elements of some test. A high value is not good. The lower FDR value, the better. The general relation of FDR is:
FDR = $\frac{V}{V+S}$; where S is the number of true positives, and V is the number of false positives.
In this project, FDR is defined by three equations:

$$\text{FDR1}= \frac{number\ of\ windows\ in\ the\ shuffled\ screen}{number\ of\ windows\ in\ the\ original\ screen}$$

$$\text{FDR2}= \frac{number\ of\ loci\ in\ the\ shuffled\ screen}{numberof\ loci\ in\ the\ original\ screen}$$

$$\text{FDR3}= \frac{length\ of\ loci\ in\ the\ shuffled\ screen}{length\ of\ loci\ in\ the\ original\ screen}$$

Locus (pl. loci) is the stretch over the overlapping windows which have a shared hit from the first position of the first window to the last position of the last window.

2.6 *RNAclust*

The purpose of using this tool [6] is to find the loci which have common secondary structure motif. The input of this tool is a FASTA file of all sequences (here our loci sequences). The final output of it is a hierarchical cluster-tree. The leaves represent the input sequences. The internal nodes represent the clusters which share common secondary structure motif.

2.7 *iTOL*

It is a free web server [5] for displaying and annotating phylogenetic trees. The input of it is a tree file of text format gotten from the *RNAclust* tool. The output is a graphical phylogenetic tree. We can color this tree according to categories we choose. Then we can see whether our category contains phylogenetic relations.

2.8 *Soupviewer*

This tool [8] gives a more comfortable way of analyzing the big trees which come from *RNAsoup* resulting from *RNAclust* tool in our case.
Additional tools were used to achieve helping roles such as Galaxy [10] (free public web server) to fetch the alignments we need, *rnazCluster.pl* to cluster the windows to loci, *rnazOutpotSort.pl* to sort the output of RNAz, and *rnazIndex.pl* to retrieve the sequences of the loci,

3 The procedure

Fetch alignments

The first step is to align these transcripts, because the input of *RNAz* is an alignment. To do this we used Galaxy as the following: First of all, we uploaded our BED file to the Galaxy server using the tool *Get Data* without forgetting to choose the correct Genome which is in our project *Human Feb. 2009 (GRCh37/hg19) (hg19).* Then we used the tool *Fetch Alignment* which provides us with a variety of alignment methods. For our analysis we chose *Stitch Gene blocks given a set of coding exon intervals* method for screen1, and *Extract MAF blocks given a set of genomic intervals* method for screen2. We have done a complete analysis for screen1 and used the screen2 to give us a general understanding of the transcripts by comparing them. The method for screen1 does the following:
It finds MAF blocks that overlaps the coding regions, sorts MAF blocks by alignment score, stitches blocks and resolves overlaps based on alignment score and finally outputs alignments in FASTA format [3]. The tool offers to choose sequences among many species to be aligned with our sequences. We chose all the species. Finally, we downloaded the resulting FASTA file to our machine. The method for screen2 has a similar effect except for that it does not ignore the intronic region as the first screen does.

Prepare the alignment

The second step is to predict the secondary structures of these transcripts using the alignments we have. We should pre-process the alignments before we use them as input for the tool *RNAz*. Otherwise, it would be too difficult to continue with non-processed alignments which may be too long, have a lot of gaps and repeats. The tool *rnazWindow.pl* achieves the preprocessing. We can specify the length of a window by passing it as a parameter to the tool. We took the default settings with window length of 120 nt and a shift of 40 nt which are the default values (optimal behaviour). *rnazWindow.pl* takes the resulting alignment file as input and returns the processed alignments as output. Since the format of the input must be either MAF or CLUSTAL, we had to change the FASTA format into one of those formats. We used the tool *ClustalW* [2] to change the FASTA format into ClustalW one.

Prediction (run *RNAz*)

At this point, the alignments are ready to be an ideal input for *RNAz*. We run *RNAz* on the alignment windows passing the parameter -p 0.5. The option -p specifies the class probability, i.e *RNAz* outputs the hits with just p>X (p>0.5). The hits of 0.5 probability are classified as functional ones. later we will filter the hits of 0.9 probability in order to study the most reliable hits.

Estimation of FDR

Now to estimate the precision of the number of our hits, we calculated the false discovery rate (FDR) using the three equations. To do this, we generated a control screen by shuffling the alignments. Then running *RNAz* again to get the hits of shuffled alignments at the same probability classes (0.5, 0.9). To shuffle the alignments preserving its characteristics we used the tool *SISSIz*.

Clustering the overlapping windows to loci

Then, we clustered the hits of RNAz using the tool *rnazCluster.pl*. This tool takes the (RNAz output) as input and combines the hits in overlapping windows to loci. The input must be sorted according to the genomic locations of the reference sequence.

3.1 Retrieval of loci sequences

The output of the last step are the loci without their sequences. The tool *rnazIndex.pl* was used to retrieve the sequence of the loci. At this point, we have the loci names and sequences which are the input of the next step.

Clustering the loci sequences

Now, to identify the hits (loci sequences) which share a secondary structure motif, we clustered them using *RNAclust.pl* [6] which gave us a hierarchical cluster-tree. Each internal node of the tree represents a cluster of common secondary structure motif. Each leaf represents one of the loci.

Tree Visualization

Then we used the tool *Soupviewer* and the tool *iTOL* to graphically show the tree and easily identify the best clusters.

Figure 1 is a diagram of the work flow of the whole analysis starting by the data set and ending by the visualization.

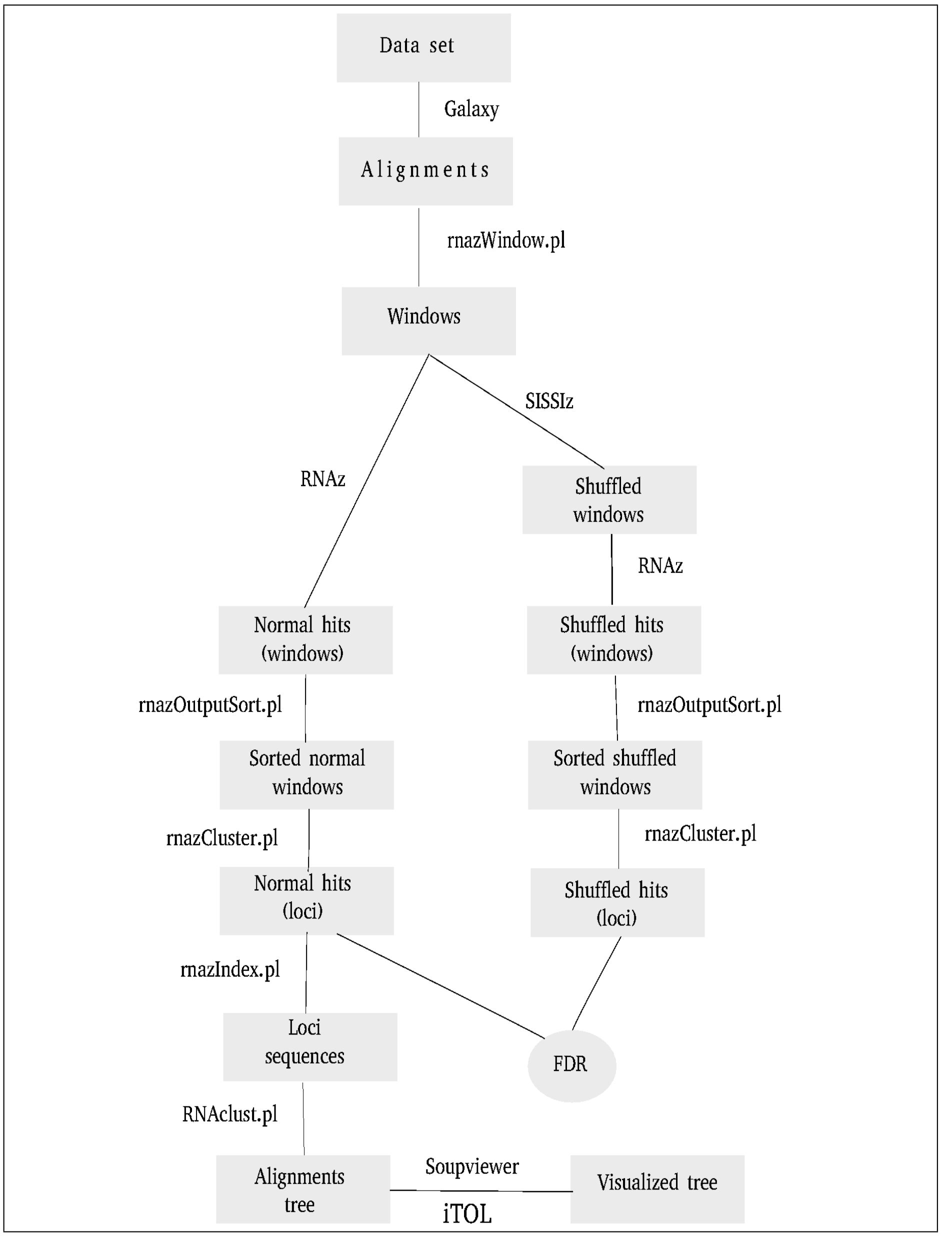


Figure 1: The general work flow diagram of the Analysis using RNAz

4 Results

4.1 Statistics of both screens for comparison

Table 1 gives statistics for class probability higher than 0.5 which are classified as functional, and Table 2 gives statistics for class probability higher than 0.9 which are classified as functional of high reliability.

Table 1: several statistics of both screens at p>0.5

Statistics (0.5)	screen1	screen2
number of loci	2279	8158
number of windows	3236	7559
total length of loci (nt)	325359	539198

Table 2: several statistics of both screens at p>0.9

Statistics (0.9)	screen1	screen2
number of loci	877	2700
number of windows	1079	2554
total length of loci	115414	178849

Having compared the number of loci of both screens (0.5), we found that 2279 (27.93 %) of 8158 loci are located in exonic regions or splice sites, whereas the remaining 5879 loci (72.07%) are located in intronic regions.
And having compared the number of loci of both screens (0.9), we found that 1823 (67.51 %) of 2700 loci are located in intronic regions, 197 loci (7.29 %) span splice sites and 680 (25.18%) loci are located in exons.

4.2 Statistics of transcripts of screen1

Table 3 enables us to compare the resulting hits with the original transcripts using two parameters: the length of the sequences and the number of transcripts. As we notice, we lost a very large part of the sequences by aligning and windowing the transcripts. Therefore, the total length of the loci is too small to consider the results sufficient one.

Table 3: The number and length of transcripts of all analysis stages

Statistics (transcripts)	Number of transcripts	Length of transcripts (nt)
Original	8195 (100%)	426642903 (100%)
Alignments	8175 (99.75%)	133021650 (31.17%)
Windows	5426 (66.21%)	4043902 (0.94%)
Loci (0.5)	1276 (15.57%)	325359 (0.076%)
Loci (0.9)	638 (7.78%)	115414 (0.027%)

4.3 FDR

The next tables show the FDR values of both screens at the two probability classes 0.9 and 0.5.
As we see the FDR of screen1 in Table 4 and Table 6 is optimistic, whereas the FDR of screen2 is pessimistic as we see in Table 5 and Table 7.

Table 4: FDR calculated on screen1 of 1000 alignments p>0.9

FDR (all alignments) (0.9)	normal	shuffled	FDR
number of loci	877	268	0.306
number of windows	1079	281	0.260
total length of loci	115414	34277	0.296

Table 5: FDR calculated on screen2 of 1000 alignments p>0.9

FDR (1000 alignments) (0.9)	normal	shuffled	FDR
number of loci	25	14	0.560
number of windows	31	14	0.451
total length of loci	2078	1001	0.481

Table 6: FDR calculated on screen1 of 1000 alignments p>0.5

FDR (all alignments) (0.5)	normal	shuffled	FDR
number of loci	2279	706	0.309
number of windows	3236	945	0.292
total length of loci	325359	87846	0.269

Table 7: FDR calculated on screen2 of 1000 alignments p>0.5

FDR (1000 alignments) (0.5)	normal	shuffled	FDR
number of loci	52	42	0.807
number of windows	66	44	0.666
total length of loci	4702	3248	0.690

We can notice that FDR value changes in each single screen and between both screens. i.e. FDR of screen2 is higher than FDR of screen1, and in each single screen: FDR of 0.9 class is lower than FDR of 0.5 class.
For a single screen, we can interpret the increase of FDR of class 0.5 in comparison to class 0.9 by the high reliability of prediction (>0.9).
For both screens, the interpretation might depend on the fact that screen2 contains introns while screen1 does not, but this needs further analysis. Otherwise, the chance could be the reason for this effect.

4.4 Statistics of screen1 of class 0.9

We chose the hits (loci) of screen1 of class probability 0.9 to continue the analysis because of its high reliability. In this paragraph we present some statistics of this class.

Table 8 shows how many transcripts have a specific number of loci. One interesting transcript is the one which has 13 hits. This transcript belongs to chromosome 5.

Table 8: The number of transcripts which have specific number of hits

Number of hits (loci)	Number of transcripts
1	476
2	119
3	25
4	13
5	4
6	1
13	1

To get further information of the 13 hits, we presented them using UCSC Genome Broswer, see Figure 2. We can simply notice that the hits cluster themselves in groups such as the first four and the next six hits. This fact could be of biological meaning, for example they might share one function.

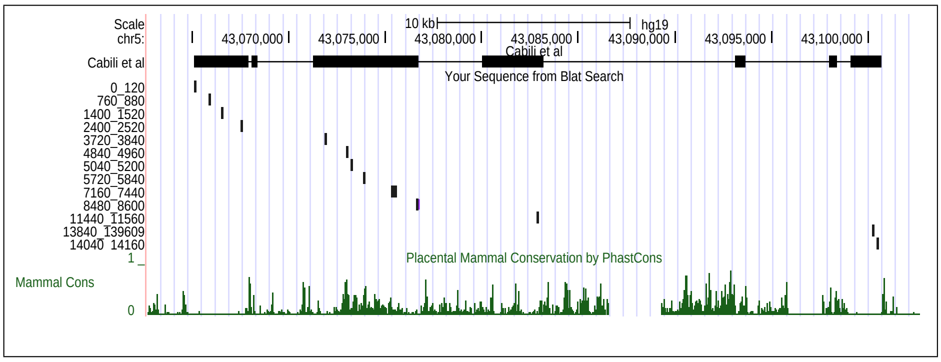


Figure 2: The locations of the 13 hits

Figure 3 shows the number of hits per chromosome. We notice that chr1, chr2 and chr5 have more hits than others. Considering the number of transcripts of each of the three (720, 768, 533), respectively, we see that chr5 has relatively more hits than the others. However, considering a more reliable measurement: the total length of the transcripts of each chromosome (50288230,20504623, 10647790) seems to be more interesting, since chr5 is much smaller than chr1 or chr2. Highlighting this information might be helpful in further studies.

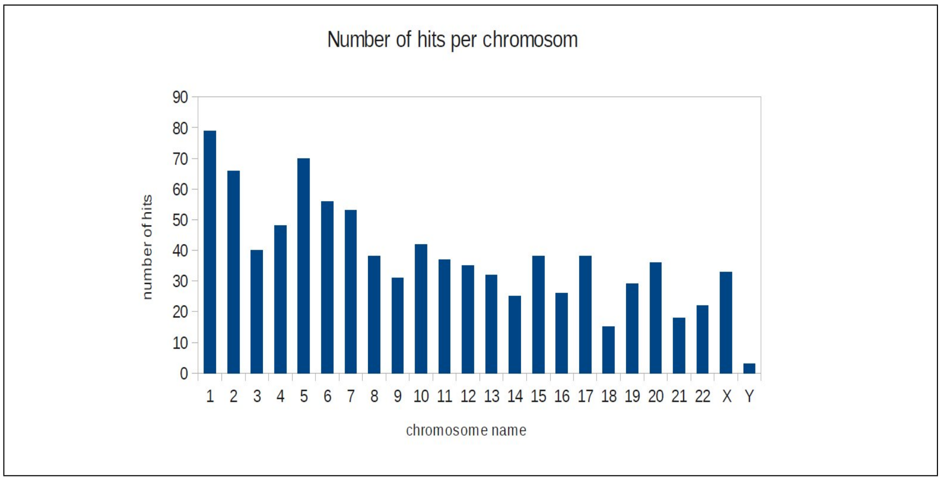


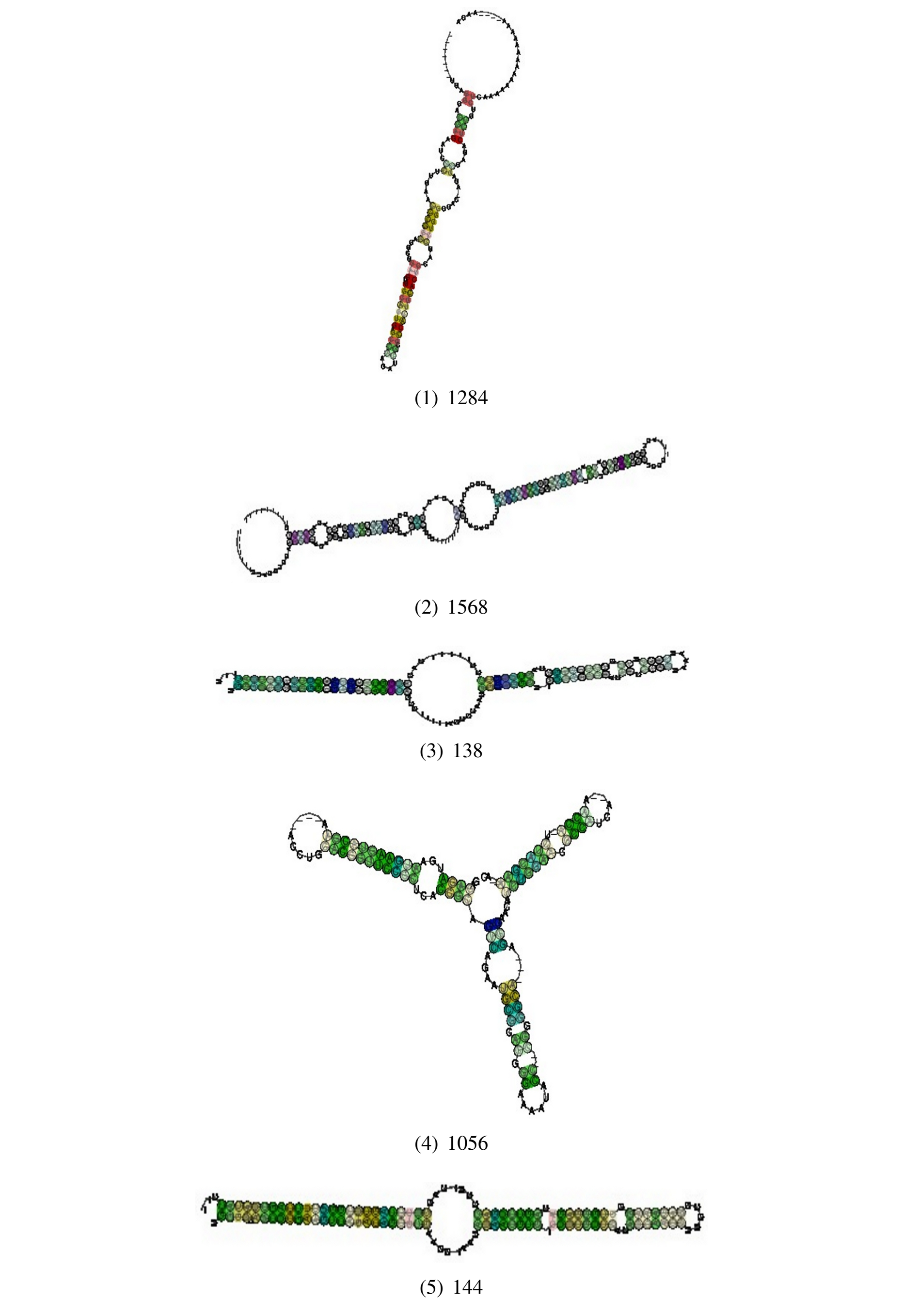
Figure 3: Number of hits per chromosome

4.5 The similarity between the loci (screen1 0.9)

The tree retrieved from*RNAclust.pl* enabled us to reveal clusters of the loci which share secondary structure motifs. We chose the best clusters resulting from the tree according to the minimum free energy (MFE) and the structural conservation index (SCI), taking in consideration the number of sequences which effects both criteria, see table 9. The best family is the first one: 14 loci belong to different transcripts and different locations.

Table 9: The best alignments of the loci (0.9) using their node ids of the resulting tree

Node_ID	Number of sequences	MFE	SCI
1284	14	-24.54	0.566277
1568	10	-42.29	0.573517
138	7	-39.15	0.774459
1056	5	-35.08	0.845261
144	4	-52.33	0.860691



(1) 1284

(2) 1568

(3) 138

(4) 1056

(5) 144

Figure 4: The secondary structures of the best five alignments presented by the tree

Figure 4 shows the secondary structures of the best chosen alignments. The best family resulting from our analysis is the one consisting of 14 sequences (loci). Although it has a relatively large number of sequences, the values of SCI and MFE are relatively good.

Figure 5 presents the tree drawn by *iTOL*. We showed just the best alignments with its own secondary structures next to them.

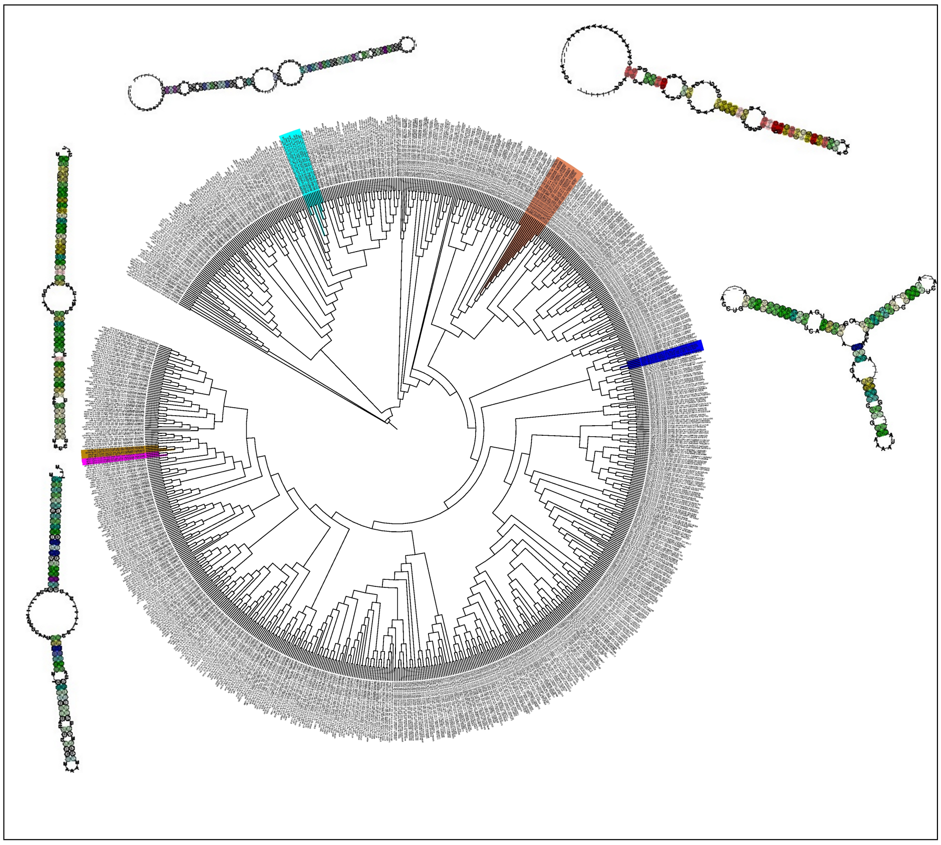


Figure 5: The only chosen alignments are colored on the tree with its secondary structures

4.6 Similarity according to the relative locations of loci

We tried to discover similarity between the loci secondary structures according to their locations at the transcripts. This way, we classified the loci locations to 9 sequential classes starting from the smallest coordinate of the loci, see Figure 6. Then we gave each class a unique color definition. Then we used *iTOL* to draw the tree resulting from *RNAclust*, see Figure 7. We can easily notice that the colors are randomly distributed. i.e. the sequences which locate in the same place of different transcripts do not share secondary structure motifs. This means that we can not benefit from the classification depending on locations to give it a biological meaning.

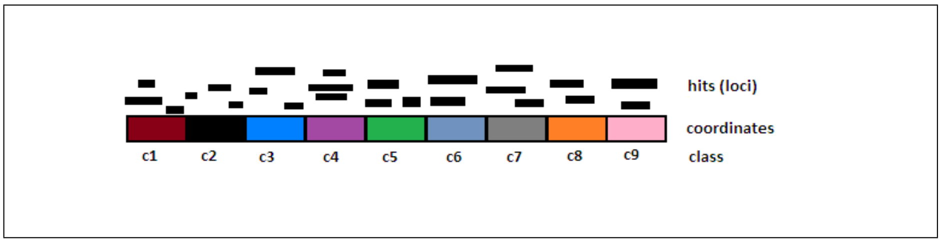


Figure 6: A representative diagram of coordinating the loci according to its relative locations at the transcripts

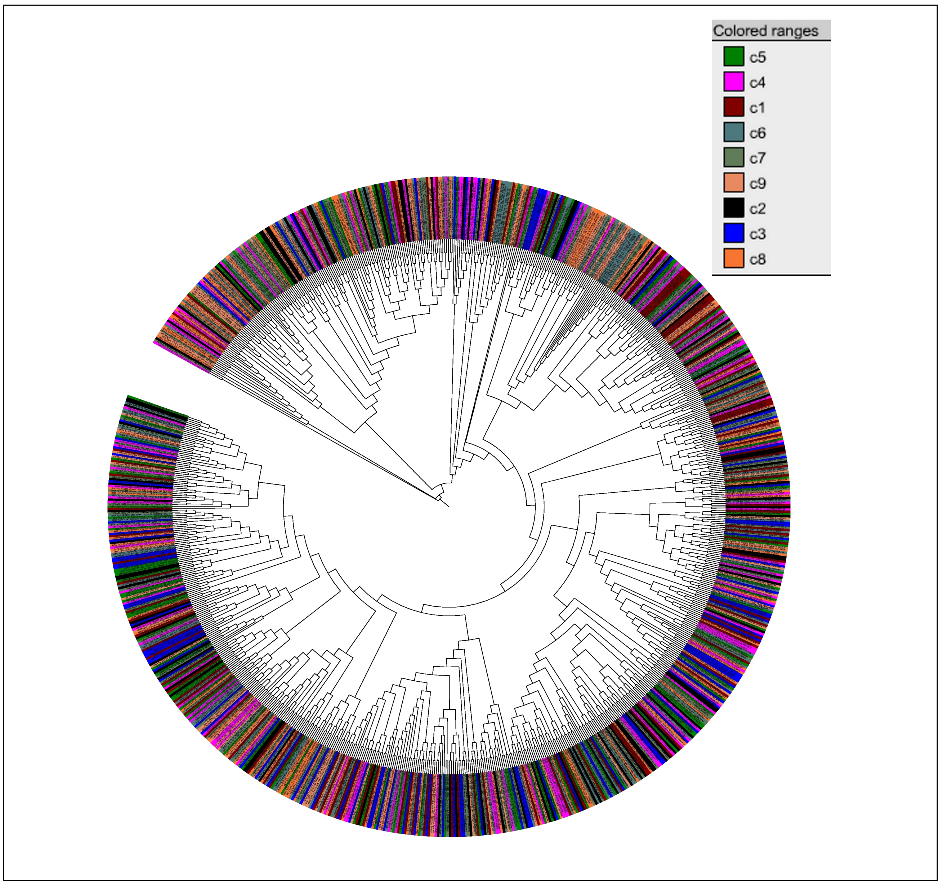


Figure 7: The tree colored according to the loci locations within their transcripts. c1 indicates class 1 which locates in the start segment of the loci, c2 indicates class 2 which locates in the second segment of the transcripts, and so on.

5 Discussion

We worked on two screens of the same set of transcripts. We could, thus, enrich the results by a comparison. From the one hand, the absence of the introns of screen1 helped detecting the hits in exons and splice sites. From the other hand, the existence of the introns (in addition to exons and splice site) in screen2 helped detecting the hits

in introns. Furthermore, we could calculate several values of FDR of both screens.
It is interesting that most loci are located in introns. Nevertheless, this might be interpreted by the fact the several classes of small RNAs (such as miRNAs and snoRNAs) are hosted by lncRNA in intronic regions, and they are well detectable by *RNAz*. However, the interpretation of this fact might be addressed as a future work.

Remarkably, FDR calculated on screen2 was on average bigger than the one calculated on screen1. Since the only difference between the screens is the existence or absence of introns, only this fact may effect FDR. The effect of introns on *RNAz* or *SISSIz* might be a topic of further research to study whether they are biased by the contents of introns , because both tools are used to calculate FDR.
Unfortunately the length of the resulting loci was too small in comparison with the length of the original transcripts. Thus, this analysis can not be considered as a comprehensive or final one, since we lost the vast majority of the transcripts by aligning and windowing them. Nevertheless, the resulting loci contain some small families of shared secondary structure motifs. A drawback of the analysis was the lack of well known transcripts of such kind of ncRNAs which did not help us to make a comparison which might reveal the functions of our loci.
Finally we can say that the very big loss by aligning was the essential drawback (70%) of this analysis. Therefore, a possible future work may be an attempt to create methods which do not depend on the alignments.

Bibliography

[1] BED Format. http://genome.ucsc.edu/FAQ/FAQformat.html.

[2] ClustalW. http://www.clustal.org/clustal2/.

[3] FASTA Format. http://blast.ncbi.nlm.nih.gov/blastcgihelp.shtml.

[4] FDR (False Discovery Rate). http://en.wikipedia.org/wiki/False_*discovery*_*rate*.

[5] iTOL. http://itol.embl.de/.

[6] RNAclust.pl. http://www.bioinf.uni-leipzig.de/ kristin/Software/RNAsoup/.

[7] SISSIz tool. https://github.com/wash/sissiz.

[8] Soupviewer. http://www.bioinf.uni-leipzig.de/ jane/software/soupviewer/manual.php.

[9] M. N. Cabili, C. Trapnell, L. Goff, M. Koziol, B. Tazon-Vega, A. Regev, and J. L. Rinn. Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. *Genes Dev.*, 25(18):1915–1927, Sep 2011.

[10] Galaxy Team.

[11] S. Washietl, I. L. Hofacker, and P. F. Stadler. Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. U.S.A.*, 102(7):2454–2459, Feb 2005.

[12] Stefan Washietl. rnazWindow.pl. http://www.tbi.univie.ac.at/ wash/RNAz/man/rnazWindow.html.