`BlockClust`: efficient clustering and classification of non-coding RNAs from short read RNA-Seq profiles

Pavankumar Videm[1], Dominic Rose[1,2], Fabrizio Costa[1], Rolf Backofen[1,3,4,5]

[1]Bioinformatics Group, Department of Computer Science, Albert-Ludwigs-University Freiburg, Georges-Köhler-Allee 106, D-79110 Freiburg, Germany. [2]Munich Leukemia Laboratory (MLL), Munich, Germany. [3]Centre for Biological Signalling Studies (BIOSS), Albert-Ludwigs-University Freiburg, D-79104 Freiburg, Germany. [4]Centre for Biological Systems Analysis (ZBSA), Albert-Ludwigs-University Freiburg, Habsburgerstr. 49, D-79104 Freiburg, Germany. [5]Centre for Non-coding RNA in Technology and Health, University of Copenhagen, Gronnegardsvej 3, DK-1870 Frederiksberg C, Denmark.

{videmp, rose, costa, backofen}@informatik.uni-freiburg.de

Abstract: Sequence and secondary structure analysis can be used to assign putative functions to non-coding RNAs. However sequence information is changed by post-transcriptional modifications and secondary structure is only a proxy for the true 3D conformation of the RNA polymer. In order to tackle these issues we can extract a different type of description using the pattern of processing that can be observed through the traces left in small RNA-seq reads data. To obtain an efficient and scalable procedure, we propose to encode expression profiles in discrete structures, and process them using fast graph-kernel techniques.

We present `BlockClust` for both clustering and classification of small non-coding RNA transcripts with similar processing patterns. We show how the proposed approach is scalable, accurate and robust across different organisms, tissues and cell lines. `BlockClust` was successfully applied on a comprehensive set of eukaryotic data. It is the first tool for eukaryotic non-coding RNA analysis available on the `galaxy` framework.

1 Motivation

The study of non-coding RNAs (ncRNAs) is nowadays becoming important to fully understand cellular functions. On the one hand, most of the transcribed DNA is non-protein-coding [Jac09]; on the other hand ncRNAs play a vital role in many cellular processes. Although up to 450 000 ncRNAs were predicted in the human genome [RBT+10], the large majority is still missing functional annotation. Sequence and secondary structure analysis can be used to assign putative functions to ncRNAs, however sequence information is changed by post-transcriptional modifications [FLSH11], and secondary structure is only a proxy for the true 3D conformation of the RNA polymer. A different type of information that does not suffer from these issues and that can be used for the detection of RNA functional classes, is the pattern of processing

that can be observed through the traces left in small RNA-seq reads data. For example the primary microRNA transcript cleaved by the Drosha complex and forms hairpin like pre-miRNA with 2-nt 3' overhang where Dicer binds and processes into double stranded miRNA and (complementary) miRNA* duplex [GST+08]. The miRNA strand then binds to Ago2 proteins to form RNA-induced silencing complex, which subsequently targets mRNA for regulation while the remaining miRNA* strand is degraded. Traces of this process are often observed in RNA-seq data of miRNA precursor as two adjacent *piles* of reads separated by few bases (length of hairpin). While one of the pile that corresponds miRNA strand is expressed, the other one that corresponds to the miRNA* strand, is not. Computational approaches such as `mirDeep` [FCA+08] rely on this miRNA biogenesis for annotation. Other examples involve snoRNAs, where snoRNA-derived fragments size and position distributions are conserved across species [TGL+09]. The tRNA molecules also undergo post-transcriptional cleavage to form smaller tRNA fragments which carry distinct expression levels and possibly different regulatory functions [GP13].

In this article, we propose `BlockClust` [VRCB14] as a novel technique to capture these processing patterns and detect transcripts that can have evolutionary relationship.

2 Methods

The core idea of the `BlockClust` is to characterize transcripts from small RNA-seq data by extracting characteristic attributes from their expression profiles. Those attributes are encoded into compact discrete structures, which can be processed using fast graph-kernel techniques to find similar expression profiles.

Given the mapped reads we consider only unique reads in the sample (*tags*). For each tag, the expression is normalized by dividing the number of reads associated with that tag by the number of times the tag is mapped to the reference genome. The notion of tags allows the elimination of duplicated data, hence speeding up subsequent processing. We use the `blockbuster` tool [LBSH+09] to identify consecutive tags with high expression and group then into *blocks.* Adjacent blocks, that are either overlapping or that are within a small distance, are then grouped into larger *blockgroups*. Here we assume that a ncRNA gene can span at most a single blockgroup. Each blockgroup is then encoded as a discrete graph. We consider different types of information, ranging from the information available for each individual block, to the relation between two consecutive blocks and finally also the information available globally on the whole blockgroup. For the whole blockgroup we measure quantities such as: *the entropy of read starts*, *the entropy of read ends*, *the entropy of read lengths*, *the median of normalized read expressions and the normalized read expression levels in the first quartile*. For each block we measure: *the number of multi-mapped reads*, *the entropy of read lengths*, *the entropy of read expressions*, *the minimum read length and the block length*. All measures are then discretized into a small number of discretization levels using an equal-frequency algorithm. The discretized attributes are then used to label the nodes of the resulting graph representation.

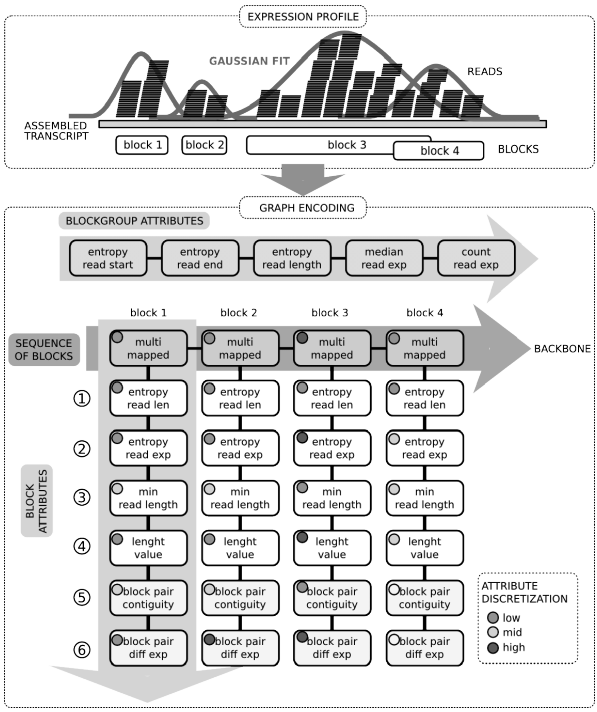

Figure 1: Graph encoding of the expression profile using descretized block and blockgroup attributes.

Figure 1 shows the graph encoding of a blockgroup with actual attributes used in the method. Each graph is made up of two disconnected components: the first one is used to encode the blockgroup attributes (shown as BLOCKGROUP ATTRIBUTES in Figure 1), while the second one represents the sequence of individual blocks and their attributes (shown as BLOCK ATTRIBUTES in Figure 1) . Finally, the resulting graphs are processed using a fast graph kernel called Neighbourhood Subgraph Pairwise Distance Kernel (`NSPDK`) [CG10]. This type of kernel evaluates the similarity between two graphs as the fraction of neighbourhood subgraph pairs that are in common. This similarity notion is parametrized by the maximal size of the neighbourhood subgraphs and by the maximal distance allowed between the subgraphs in each pair. Intuitively this approach can be considered as an extension of the gapped k-mer similarity for strings to the graph domain. Formally: a neighbourhood graph is a subgraph specified by a root vertex v and a radius R, consisting of all vertices that are at a distance (the distance between two vertices v and u on a graph is defined as the number of edges in the shortest path between v and u) not greater than R from v. All pairs of such neighbourhood subgraphs whose root vertices are at a maximum distance D are extracted by the kernel (see Figure 2 for an illustration of the subgraph pair extraction by `NSPDK`). Since neighbourhood subgraphs can be efficiently enumerated in near linear time, the resulting approach has in practice linear complexity and can be used in large scale settings.

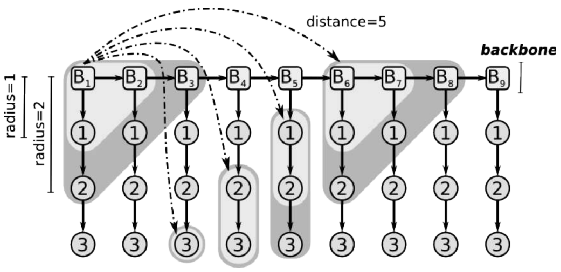

Figure 2: Extraction of subgraph pairs rooted at B_1 with radius 2 and distance 5. All possible roots which are at distance 5 from B_1 are selected and neighbourhood subgraphs of radius at most 2 are extracted.

The resulting pairwise similarity matrix can be used with several existing clustering algorithms. In this work we used the Markov Cluster Algorithm (`MCL`) [EVDO02] on the nearest neighbour adjacency matrix.

When functional annotation is available, we can design a supervised task and build a classifier for each specific ncRNA family using kernelized Support Vector Machine models. Currently, we offer models for three families, namely: miRNA, tRNA and CD-box snoRNA.

3 Results and Discussion

We applied `BlockClust` on several datasets to evaluate the predictive performance and its robustness. Finally we have compared `BlockClust` to other state-of-art tools.

3.1 Datasets and processing

To train our predictive models we have used NGS data generated by Illumina sequencing of human embryoid body and embryonic stem cells, H1 cell line and IMR90 cell line (`Development Data`). In order to compare to other tools and evaluate the robustness of `BlockClust` we have used a comprehensive collection of test datasets (`Benchmark Data`), that includes 32 samples from human, mouse, fly, chimp, worm and plant in a variety of tissues and cell lines.

`BlockClust` is a pipeline that combining several tools namely: `blockbuster`, `NSPDK` and `MCL`. In order to achieve optimal predictive performance, we have optimized the hyper-parameters of each tool. For the `blockbuster` tool we need to specify the minimum distance between two blockgroups (*cluster distance*) and the standard deviation of a single read (*scale*); for `NSPDK` the *radius* and the *distance* are the parameters of choice in order to extract the neighbourhood subgraph pairs; in `MCL` the cluster granularities were controlled via the *inflation* and *pre-inflation* parameters.

In addition to the parameter optimization of the tools used, we also have to choose the number of discretization levels for attributes and select the most discriminative attributes. Table 1 shows the overview of the value ranges, the search step size and the selected optimal values for the aforementioned parameters.

Component	Parameter	Interval	Step	Optimum
`blockbuster`	Cluster distance	20–100	10	40
`blockbuster`	Scale of standard deviation	0.2–0.8	0.1	0.5
Encoding	Discretization bins	3, 5, 7	2	3
`NSPDK`	Radius R	1,3,5,7	2	5
`MCL`	Inflation	1–30	0.3	20
`MCL`	Pre-inflation	1–30	0.3	20

Table 1: Parameter optimization. Overview of tools and probed parameter values and selected optimal values. Note that distance is defined as a function of radius: $D = 2 \times R + 1$.

All the parameters were optimized by splitting the `Development Data` into train/validation/test sets with sizes 35/35/30% respectively. Hyper-parameter were set using the train and validation sets, whereas the predictive performance is reported on the test set alone.

3.2 Performance of `BlockClust`

To assess the quality of the similarity notion generated by our approach, we measured the tendency for transcripts of functionally identical RNAs to be neighbours. We computed the Area Under the Curve for the Receiver Operating Characteristic (AUC ROC) using the distance as a predictor function to evaluate the quality of the induced metric; in addition we computed the purity of the partition generated by the `MCL` approach to evaluate the clustering quality (see Table 2).

ncRNA class	#transcripts	AUC	#clusters	cluster purity
miRNA	168	0.896	10	0.855
tRNA	173	0.741	17	0.837
C/D-box snoRNA	78	0.731	7	0.683
H/ACA-box snoRNA	4	0.838	0	0
rRNA	20	0.872	2	0.956
snRNA	7	0.637	0	0
Y_RNA	8	0.685	0	0
Weighted average	458	0.805	36	0.813

Table 2: Clustering performance of BlockClust averaged over 10 random test splits of Development Data.

Out of 458 known transcripts in the test set miRNA, tRNA and C/D-box snoRNAs contribute to the majority. There are quite less number of known profiles from the remaining four classes. After clustering with MCL, we could capture only 2 clusters of rRNAs out of these four classes, while for the majority classes we got a decent number of clusters. On average we observed a good AUC of 0.8 for the the similarity notion. The best performance was found for miRNA in terms of similarity notion and cluster precisions, followed by rRNAs, tRNAs and C/D-box snoRNAs. Though H/ACA-box snoRNAs have a good AUC, due to their low population MCL could not cluster them together. Poor performance can be seen for Y_RNA and snRNA classes.

In Table we report instead the classification performance on the test set split of Development Data when we train family specific models in a one-vs-all setting. We chose Positive Predictive Value (PPV) and Recall as performance measures. The PPV for all three classes are very good (≈ 0.9). The miRNA model could successfully retrieve 89% of the miRNAs, while 80% and only 48% recalls were observed for tNAs and C/D-box snoRNAs respectively.

ncRNA class	#transcripts	PPV	Recall
miRNA	168	0.901	0.886
tRNA	173	0.899	0.796
C/D-box snoRNA	78	0.870	0.474

Table 3: Classification performance of BlockClust averaged over 10 random test splits of Development Data.

3.3 Comparison with other tools

We compared BlockClust to other tools that can process read profiles of small ncRNAs from RNA-seq data and perform predictions or clustering. The deepBlockAlign [LPE+12]) is a tool which uses a variant of Sankoff algorithm to align all input blockgroups and cluster them. DARIO [FLB+11] is a web server which is used for annotating miRNA, tRNA and snoRNAs from deep sequencing data using a random forest classifier. The comparison with deepBlockAlign was done on the whole Benchmark Data.

Note that since DARIO is not available as a standalone tool, we considered only one of the Benchmark Data (Gene Expression Omnibus[1] (GEO) sample id: GSM769510) for comparison.

[1] http://www.ncbi.nlm.nih.gov/geo/

To compare the similarity notion of BlockClust and deepBlockAlign we computed AUC ROC on similarity matrices of both tools. Compared to deepBlockAlign, BlockClust performs better on average (AUC 0.84 vs. 0.7) and also in each individual class. See Table 4 for AUCs of both tools for each individual class and weighted average over all classes. In terms of computational complexity BlockClust is very competitive, achieving a 60-fold speed-up (50 seconds as compared to 58 minutes of deepBlockAlign on a dataset of ≈600 profiles). This is due to BlockClust quasi-linear complexity compared to the $O(m^2)O(n^6)$ complexity of the Sankoff algorithm used in deepBlockAlign (where n is the number of blocks per instance and m is the number of sequences).

ncRNA class	#transcripts	BlockClust AUC ROC	deepBlockAlign AUC ROC
miRNA	3869	0.925	0.714
tRNA	4988	0.795	0.701
C/D-box snoRNA	731	0.762	0.615
H/ACA-box snoRNA	142	0.859	0.720
rRNA	770	0.873	0.759
snRNA	240	0.698	0.610
Y_RNA	244	0.694	0.656
Weighted average	11061	0.839	0.700

Table 4: Comparison of BlockClust vs. deepBlockAlign on whole Benchmark Data.

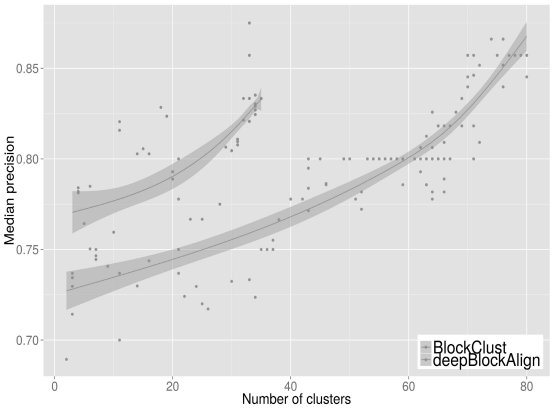

Figure 3: Cluster purities `BlockClust` (red) vs. `deepBlockAlign` (blue). The median of cluster precisions with respect to number of clusters generated by `MCL` clustering algorithm at different inflation values.

In order to compare the precision of the clusters that can be obtained from `BlockClust` and `deepBlockAlign`, we applied `MCL` on similarity matrices from both tools. We used one sample from the `Benchmark Data` (GEO sample id: GSM450239) for comparison. The inflation and the pre-inflation parameters of `MCL` affect the cluster granularity, so by varying these parameters we obtained varying number of clusters for both tools. Figure 3 depicts the median cluster purities for number of clusters obtained at different inflations. In theory, with increasing number of clusters, the cluster sizes decrease. In turn, the smaller clusters tend to be more pure than larger ones. At all inflation settings, `BlockClust` produced less number of clusters with higher median precisions compared `deepBlockAlign`. Hence `BlockClust` potentially produces larger clusters with a higher precision.

Please note that the `deepBlockAlign` is an algorithm designed and optimized to identify similar processing patterns regardless of the ncRNA class. Therefore it might not give optimal results when used to cluster the ncRNAs into the families of their primary function.

	miRNA		tRNA		snoRNA C/D-box	
ncRNA class	PPV	Recall	PPV	Recall	PPV	Recall
`BlockClust`	0.88	0.89	0.95	0.80	0.74	0.39
`DARIO`	0.85	0.81	0.92	0.88	0.46	0.52

Table 5: Comparison of classification performance of `BlockClust` against `DARIO`.

Compared to `DARIO`, `BlockClust` exhibits a better precision for all three ncRNA classes and also slightly better recall for miRNAs. Whereas, `DARIO` achieves a better

recall for the remaining two classes. Please refer to Table 5 for comparison of BlockClust and DARIO. Note that since DARIO is available only as a web server we could not reliably assess its run times.

3.4 Clustering analysis

To examine whether the BlockClust encoding of the attributes is discriminative enough to cluster ncRNA classes, we analysed the clusters generated by the BlockClust as follows. First we clustered all blockgroups in one sample from Benchmark Data (GEO sample id: GSM768988) using BlockClust. Then for each ncRNA family, we considered the clusters with highest precision. The hierarchical clustering of these cluster instances along with the representative expression profiles are shown in Figure 4.

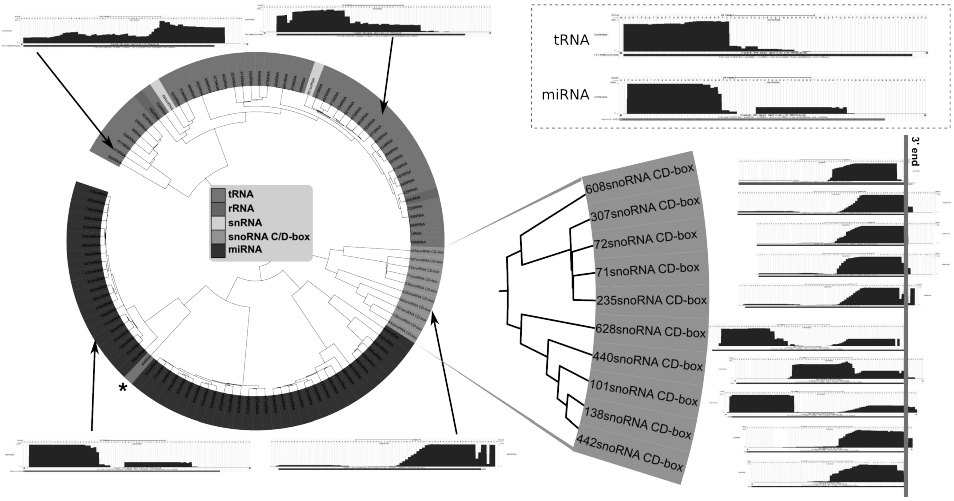

Figure 4: Hierarchical clustering of the BlockClust clusters of each family with highest precision. One representative read profile for miRNAs and snoRNAs, and two for tRNAs are shown. The annotation of the ncRNA can be seen under each profile as an horizontal bar.

From clustering (Figure 4), we observed that the tRNA (blue) branch is constitute two different representative profiles for 5'- and 3'- derived fragments. For miRNAs (purple), the classic 2-*block* profile can be found, where expressed block represents the miRNA and non-expressed block represents the degraded miRNA*. According to literature [TGL+09], the CD-box snoRNAs are mostly 5'-derived fragments. Surprisingly, in our example dataset, we observe CD-box snoRNAs with consistent 3'-derived fragments.

Finally, we investigate the tRNA (marked with *) that was clustered together with the miRNAs. Similar to miRNAs the read profile of this tRNA has a precisely cut 5'-derived

fragment (see top right corner box in Figure 4). It has been already demonstrated that such 5'-derived tRNA fragments could possibly processed by dicer as miRNAs [GP13] and have functional characteristics of miRNAs [MSS+13].

4 Conclusion

We presented `BlockClust`, an approach that can exploit processing traces of small ncRNAs to reliably and efficiently identify functional non-coding genes. We encode read expression profiles in compact discrete structures in order to use fast graph kernel approaches, obtaining competitive predictive performance and a significant speed-up compared to existing approaches. The complete work-flow of `BlockClust` and its tool dependencies are easily installable and usable from the galaxy [GNT+10] main toolshed: http://toolshed.g2.bx.psu.edu/view/rnateam/blockclust_workflow

Funding

German Research Foundation (DFG-grant SFB 992/1 and BA 2168/3-1 to R.B.).

References

[CG10] Fabrizio Costa and Kurt De Grave. Fast Neighborhood Subgraph Pairwise Distance Kernel. In Proceedings of the 26 th International Conference on Machine Learning, pages 255–262. Omnipress, 2010.

[EVDO02] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res, 30(7):1575–84, 2002.

[FCA+08] Marc R. Friedlander, Wei Chen, Catherine Adamidi, Jonas Maaskola, Ralf Einspanier, Signe Knespel, and Nikolaus Rajewsky. Discovering microRNAs from deep sequencing data using miRDeep. Nat Biotechnol, 26(4):407–15, 2008.

[FLB+11] Mario Fasold, David Langenberger, Hans Binder, Peter F. Stadler, and Steve Hoffmann. DARIO: a ncRNA detection and analysis tool for next-generation sequencing experiments. Nucleic Acids Res, 39(Web Server issue):W112–7, 2011.

[FLSH11] Sven Findeiss, David Langenberger, Peter F. Stadler, and Steve Hoffmann. Traces of post-transcriptional RNA modifications in deep sequencing data. Biol Chem, 392(4):305–13, 2011.

[GNT+10] Jeremy Goecks, Anton Nekrutenko, James Taylor, et al. Galaxy: a comprehensive approach for supporting accessible, reproducible,and transparent computational research in the life sciences. Genome Biol, 11(8):R86, 2010.

[GP13] Jennifer Gebetsberger and Norbert Polacek. Slicing tRNAs to boost functional ncRNA diversity. RNA Biol, 10(12), 2013.

[GST+08] Jianhua Gan, Gary Shaw, Joseph E. Tropea, David S. Waugh, Donald L. Court, and Xinhua Ji. A stepwise model for double-stranded RNA processing by ribonuclease III. Mol Microbiol, 67(1):143–54, 2008.

[Jac09] Alain Jacquier. The complex eukaryotic transcriptome: unexpected pervasive transcription and novel small RNAs. Nat Rev Genet, 10(12):833–44, 2009.

[LBSH+09] David Langenberger, Clara Bermudez-Santana, Jana Hertel, Steve Hoffmann, Philipp Khaitovich, and Peter F. Stadler. Evidence for human microRNA-offset RNAs in small RNA sequencing data. Bioinformatics, 25(18):2298–301, 2009.

[LPE+12] David Langenberger, Sachin Pundhir, Claus T. Ekstrom, Peter F. Stadler, Steve Hoffmann, and Jan Gorodkin. deepBlockAlign: a tool for aligning RNA-seq profiles of read block patterns. Bioinformatics, 28(1):17–24, 2012.

[MSS+13] Roy L. Maute, Christof Schneider, Pavel Sumazin, Antony Holmes, Andrea Califano, Katia Basso, and Riccardo Dalla-Favera. tRNA-derived microRNA modulates proliferation and the DNA damage response and is down-regulated in B cell lymphoma. Proc Natl Acad Sci USA, 110(4):1404–9, 2013.

[RBT+10] Mathieu Rederstorff, Stephan H. Bernhart, Andrea Tanzer, Marek Zywicki, Katrin Perfler, Melanie Lukasser, Ivo L. Hofacker, and Alexander Huttenhofer. RNPomics: defining the ncRNA transcriptome by cDNA library generation from ribonucleo-protein particles. Nucleic Acids Res, 38(10):e113, 2010.

[TGL+09] Ryan J. Taft, Evgeny A. Glazov, Timo Lassmann, Yoshihide Hayashizaki, Piero Carninci, and John S. Mattick. Small RNAs derived from snoRNAs. RNA, 15(7):1233–40, 2009.

[VRCB14] Pavankumar Videm, Dominic Rose, Fabrizio Costa, and Rolf Backofen. BlockClust: efficient clustering and classification of non-coding RNAs from short read RNA-seq profiles. Bioinformatics, 30(12):i274–i282, 2014.