FUNCTIONAL CLASSIFICATION OF LONG NON-CODING RNAS

Jessime Kirk

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Curriculum in Bioinformatics and Computational Biology

Chapel Hill
2019

Approved by:

J. Mauro Calabrese

Peter J. Mucha

Timothy Elston

Alain Laederach

Daniel Dominguez

ABSTRACT

Jessime Kirk: Functional classification of long non-coding RNAs
(Under the direction of J. Mauro Calabrese and Peter J. Mucha)

Long non-coding RNAs (lncRNA) play important roles in mammalian development and health. The relationships between lncRNAs' sequences and their functions, however, are poorly understood. Unlike proteins, which often contain recognizable functional domain and are evolutionarily conserved, lncRNAs lack significant linear sequence homology. To address this issue and enable the prediction of a lncRNA's biological properties from its sequence, we developed a non-linear sequence similarity algorithm called SEEKR. SEEKR allows for the comparison of sequences based on the non-linear abundance of short motifs. These short motifs, or k-mers, may represent potential protein binding sites within the lncRNA.

We used SEEKR to form communities of similar lncRNAs from both human and mouse transcriptomes. We were then able to demonstrate that these communities predicted the biological properties of the lncRNAs within a given community, including cellular localization and protein binding. We also show we can predict RNAs' repressive activity in *vivo* using SEEKR.

Additionally, SEEKR provided evidence of similarity between certain pre-mRNA transcripts and known repressive lncRNAs. We hypothesized that some of these pre-mRNAs may have localized repressive capabilities. We demonstrated that pre-mRNAs are detectable at physiologically relevant levels in human cells and that some pre-mRNAs with repressive-like sequences may also interact with transcription regulating proteins in *cis*.

Finally, we have packaged SEEKR into a user-friendly command line tool, which is free and open source. Here, we provide an extensive tutorial describing both how we have used SEEKR and how to perform common analysis tasks.

To the friends we made along the way.

ACKNOWLEDGEMENTS

This works was completed with the help of many other people. First, I would like to thank Mauro Calabrese, who initially came up with the idea for this work. More importantly, he has been as good of a mentor and leader as I could have hoped for. I would also like to thank the other members of the Calabrese lab, particularly Megan Schertzer and David Lee, who have made it a pleasure to go to work every day. Similarly, I would like to thank the other members of my committee, Peter Mucha, Tim Elston, and Alain Laederach, Daniel Dominguez, as well as Praveen Sethupathy, who have provided me with guidance and suggestions over the years.

I would like to thank all of my fellow graduate school friends who have fielded ideas, provided insights, participated in side projects, helped with homework, listened to grievances, shared milestones, and have been part of this journey. Additionally, I would like to thank all my other friends as well, who have provide laughs, adventures, endless inside jokes, dinner parties, and holiday celebrations, and generally provided me with a life outside of graduate school. I wish I could name you all here, but you know who you are.

Finally, I would like to thank my wife, Laura, who has been there through it all, and provided me with everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANOVA	Analysis of Variance
CASP	Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction
CHAMP	Convex Hull of Admissible Modularity Partitions
CHIP	Chromatin immunoprecipitation
eCLIP	Enhanced cross-linking immunoprecipitation
ENCODE	Encyclopedia of DNA Elements
GTF	Gene transfer format
H3K27me3	Histone H3 Lysine 27 trimethylation
HSD	Honestly significant difference
HMM	Hidden Markov Model
IQR	Inner quartile range
ITR	Inverted terminal repeats
Kb	Kilobase
lncRNA	Long non-coding RNA
LRT	Likelihood ratio test
Mb	Megabase
MFE	Minimum free energy
mRNA	Messenger RNA
NLP	Natural language processing
PWM	Position weight matrix
RPBP	Reads per base pair
RPKM	Reads per kilobase million
SEEKR	SEquence Evaluation from Kmer Representation
SHAPE-MaP	Selective 2′-hydroxyl acylation analyzed by primer extension and Mutational profiling

TETRIS	Transposable element to test RNA’s effect on transcription in *cis*
TRE	Tetracycline Responsive Element
TSC	Trophoblast stem cells
XIST	X-inactive Specific Transcript

CHAPTER 1

Introduction

1.1 Importance of bioinformatics models to genomic studies

One of the key goals of bioinformatics as applied to genomics is to enable the prediction of the biological properties of a nucleotide sequence based solely on the contents of that sequence. Specific sequence content determines secondary and tertiary structures of molecules. It regulates interaction partners with all other types of biomolecules. Ultimately, sequence determines the biological role of nucleic acids in both healthy and disease cells.

The inherent combinatorial complexity of genomic sequences makes the task of accurately predicting biological properties from sequence alone extremely challenging. The number of possible sequences for even very short nucleic acids—let alone the length of the human genome—is effectively infinite. Therefore, it is impossible to create a complete mapping of nucleic acid sequences to biological roles. Instead, an effective approach to understanding a given molecule's cellular role is to build computational models which leverage knowledge of underlying biological principles to make predictions about the molecule.

Several subfields of bioinformatics and genomics have made great progress with this approach. Although the work is currently still unpublished, DeepMind recently made headlines for winning the yearly CASP competition[1]. The goal of the competition is to accurately predict the three-dimensional structure of a given protein based only on its sequence. Despite their newcomer status, DeepMind used their deep learning system AlphaFold to demonstrate that there's still much to discover in the sequence prediction field. Similarly, tools like nhmmer have successfully been used to measure the evolutionary relationship between known functional domains and newly discovered sequences of interest, information which can be used to predict the function of the new sequence.

Despite the success in some sub-fields, other nucleic acid sequences of interest have proved more difficult to study.

1.2 Long non-coding RNAs

Long non-coding RNAs, or lncRNAs, are defined as RNA transcripts longer than 200 base pairs that do not code for a protein. This class of RNAs is found in all eukaryotic genomes, and tens of thousands of transcripts have been annotated in humans[2]. A large majority of these have never been studied in any meaningful way, yet several have been shown to play important roles in normal development while others have been implicated in various diseases[3].

Even within the few lncRNAs with known functions, detailed mechanisms of action are generally unknown. Determining mechanisms is complicated by the significantly lower levels of evolutionary conservation of lncRNAs relative to mRNA[4]. It's currently not possible to use a tool such as nhmmer to discover conserved functional domains within a given lncRNA. Besides conservation levels, this failure is also attributable to the fact that tools like nhmmer were designed to study and take into account the biological roles of mRNA but do not do the same for other transcripts like lncRNAs.

An additional complication in studying mechanisms of lncRNA action is the lack of catalytic activity of lncRNAs. Instead, they likely function primarily through the set of RNA binding proteins with which they interact. lncRNAs acting as "scaffolds" or "guides" for proteins has been proposed as a reoccurring mechanism[5]. A lncRNA "scaffold" acts as a method for coordinating multiple protein binding events—of either the same or different proteins—simultaneously. As an example, binding multiple different proteins may help streamline multi-stage enzymatic reactions necessary for some biological phenomenon. lncRNAs that bind many copies of the same protein may enable competitive inhibition, where proteins which would otherwise be performing a function elsewhere in the cell are instead tethered to a lncRNA.

Similarly, lncRNA “guides” may act by recruiting a protein at a specific time and place in the cell. An important recently discovered role for lncRNAs is transcriptional regulation. A given lncRNA, through a variety of mechanisms, may influence the expression levels of other transcripts in the genome. A lncRNA, either while actively being transcribed or by interacting with proximal DNA shortly after transcription, could be localized to a specific genomic locus. This lncRNA could then influence the transcription levels of nearby loci by the recruitment of transcription factors that activate or repress transcription. By providing multiple protein binding sites throughout the sequence and localizing to a specific chromatic location, the lncRNA acts as a “guide” for some subset of RNA binding proteins.

1.3 XIST as a model lncRNA

The transcriptional repressor XIST is the most well characterized lncRNA due to its role in X-inactivation and is found in all placental mammals[6]. X-inactivation is the process by which mammalian females shut down one of their two X-chromosomes for gene dosage compensation[7]. Proper XIST expression and function is necessary for X-inactivation, and XIST is required for silencing virtually all genes on the inactive-X. While X-inactivation is complex, involving many additional factors beyond XIST, and mechanistic details are still an area of active research, XIST provides a well-studied example of lncRNA activity.

Throughout this study, we used XIST as an approximate ground truth by which to compare other less studied lncRNAs of interest. Much of the work presented here is generalizable to the lncRNA field at large, but, due to the convenience of XIST as a model for lncRNAs, we continually use XIST as a reference example.

1.4 Inspiration from natural language processing

In order to create a tool capable of providing insights into lncRNA biology based solely on the transcript, we build a sequence similarity algorithm which takes into account lncRNA biology, particularly the likelihood of lncRNAs providing function through protein binding. The technical details of this model are described fully in the following chapter, but a less formal introduction is described here, drawing off the similarity between our approach towards sequence analysis and a popular NLP technique known as a bag-of-words model[8].

Understanding the meanings or sentiments of a set of documents and their words is a common task in NLP. Examples include creating a summary of a document, classifying the tone of a tweet as “positive” or “negative”, or deciding if a given email should be sorted into spam. This last task (deciding if an email is spam) is analogous to deciding if an individual lncRNA should be classified as a potential repressor or not. Using a bag-of-words model, an email can be classified as spam by counting the word frequencies in the email. After counting all the words in the email, each word frequency can be normalized by the prevalence of its usage in the English language. The words “the” and “a” are likely two of the most common words in the email, but it is unlikely that their frequencies significantly deviate from standard usage. High frequencies of other words, such as “new”, “free”, “hurry”, “limited”, regardless of their exact position in the email or their grammatical context, are likely to indicate spam. This can be measured by comparing the normalized frequency of these words to a set of emails known to be spam. On the other hand, an email which has high normalized word frequencies of “aunt”, “family”, “dog”, “baby”, again regardless of context, is more likely to be personal in nature. Analogously, counting and normalizing sub-sequences in a transcript of interest is a viable approach for providing biological insight into a lncRNA.

In both spam detection and lncRNA function prediction, more sophisticated approaches exist. There are numerous benefits to these simplified methods, however, including speed and interpretability, which will be further explored in the following chapters.

REFERENCES

1 Evans, R. *et al.* De novo structure prediction with deep-learning based scoring. in *Thirteenth Critical Assessment of Techniques for Protein Structure Prediction* (2018).

2 Iyer, M. K. *et al.* The landscape of long noncoding RNAs in the human transcriptome. *Nat Genet* **47**, 199–208 (2015).

3 Wapinski, O. & Chang, H. Y. Long noncoding RNAs and human disease. *Trends in Cell Biology* **21**, 354–361 (2011).

4 Johnsson, P., Lipovich, L., Grandér, D. & Morris, K. V. Evolutionary conservation of long non-coding RNAs; sequence, structure, function. *Biochimica et Biophysica Acta (BBA) - General Subjects* **1840**, 1063–1071 (2014).

5 Wang, K. C. & Chang, H. Y. Molecular Mechanisms of Long Noncoding RNAs. *Molecular Cell* **43**, 904–914 (2011).

6 Brown, C. J. *et al.* The human XIST gene: Analysis of a 17 kb inactive X-specific RNA that contains conserved repeats and is highly localized within the nucleus. *Cell* **71**, 527–542 (1992).

7 Penny, G. D., Kay, G. F., Sheardown, S. A., Rastan, S. & Brockdorff, N. Requirement for Xist in X chromosome inactivation. *Nature* **379**, 131–137 (1996).

8 Zhang, Y., Jin, R. & Zhou, Z.-H. Understanding bag-of-words model: a statistical framework. *Int. J. Mach. Learn. & Cyber.* **1**, 43–52 (2010).

9 Derrien, T. *et al.* The GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. *Genome Res.* **22**, 1775–1789 (2012).

CHAPTER 2

Functional classification of long non-coding RNAs by kmer content

2.1 Introduction

The human genome expresses thousands of lncRNAs, several of which regulate fundamental cellular processes. Still, the overwhelming majority of lncRNAs lack characterized function and it is likely that physiologically important lncRNAs remain to be identified. Moreover, the mechanisms through which most lncRNAs act are not clear, limiting our understanding of the biology that they govern in cells [1-12].

A significant roadblock to progress remains the inability to detect recurrent relationships between lncRNA sequence and function. An understanding of analogous relationships in proteins has enabled the classification of protein families, functional domains, and mechanisms that, in turn, have led to discoveries that have improved the diagnosis and treatment of disease [13,14]. However, with rare exceptions, the functions of lncRNAs are unrecognizable from computational analyses and must be determined empirically [10-12,15-20]. As a result, classification of function in one lncRNA often provides no information about function in others. For example, the *Xist* and *Kcnq1ot1* lncRNAs both repress gene expression in *cis* (meaning on the same chromosome from which they were transcribed), and both require the Polycomb Repressive Complex to do so [7]. Yet, despite similarities in mechanism, the two lncRNAs share almost no sequence similarity by standard metrics. Using two common sequence alignment algorithms, nhmmer [21] and Stretcher [22], *Xist* and *Kcnq1ot1* appear just as similar to each other as they do to randomly generated sequences (Supplementary Fig. 1). Thus, comparing the sequence of *Kcnq1ot1* to a known *cis*-repressive lncRNA (*Xist*) provides no indication that *Kcnq1ot1* is also a *cis*-repressive lncRNA. This problem extends to the thousands of lncRNAs that lack characterized functions.

2.2 Results

2.2.1 Kmer-based quantitation as a means to compare lncRNA sequence content

We hypothesized that lncRNAs with shared functions should harbor sequence similarities that confer the shared functions, even if conventional alignment algorithms do not detect the similarity. Our rationale follows. First, most lncRNAs likely have no catalytic activity, suggesting that the proteins they bind in cells define their function. Second, proteins often bind RNA through short motifs, or kmers, that are between 3 to 8 bases in length, where “k” specifies the length of the motif [23,24]. Third, the mere presence of a set of protein binding motifs may be more important than their relative positioning within a lncRNA, meaning that functionally related lncRNAs could harbor related motif contents and still lack linear sequence similarity.

To test our hypothesis, we developed a method of sequence comparison, called SEEKR (*SE*quence *E*valuation from *K*mer *R*epresentation). In SEEKR, all kmers of a specified length “k” (i.e. k= 4, 5, or 6) are counted in one-nucleotide increments across each lncRNA in a user-defined group, such as the GENCODE annotation set [12]. Kmer counts for each lncRNA are then normalized by lncRNA length and standardized across the group to derive a matrix of kmer profiles, which consist of z-scores for each kmer in each lncRNA. The relative similarity of kmer profiles between any pair of lncRNAs can then be determined via Pearson’s correlation (Fig. 1A, B; Methods).

SEEKR offers advantages relative to existing alignment algorithms. Foremost, SEEKR does not consider positional information in similarity calculations, allowing it to quantify nonlinear sequence relationships. For reasons described above, this functionality might suit lncRNAs better than traditional alignment algorithms developed to detect linear sequence homology between evolutionarily related entities [21,22,25,26]. Second, whereas traditional alignment algorithms can only quantify similarity, SEEKR can quantify similarities and differences using Pearson's correlation. Third, SEEKR can quantify relationships in groups of lncRNAs despite differences in overall length, whereas length differences can confound traditional alignment algorithms. For example, conventional alignment of a 20kb and 4kb RNA is barely informative (80% of the 20kb RNA would not align), but their kmer contents can be compared via SEEKR. Lastly, SEEKR is algorithmically efficient; all pairwise comparisons between human GENCODE lncRNAs can be computed in under a minute.

Initially, we assessed whether SEEKR could detect previously identified sequence similarities in lncRNAs. We compared kmer profiles via SEEKR for all pairwise combinations in a set of 161 lncRNAs recently described to be conserved between human and mouse [27]. We also aligned the lncRNAs to each other using two existing alignment algorithms, the hidden Markov model based nhmmer [21], and Stretcher, an implementation of the global alignment algorithm Needleman-Wunsch [22]. In this test, SEEKR detected known lncRNA homologues nearly as well as or better than both algorithms (Fig. 1C). We defined signal to background in this assay as the ratio between the median similarity of homologous and non-homologous lncRNAs. By this metric, nhmmer detected homologues the most clearly, as expected (signal-to-background ratio of 0.606 : 0.000), followed by SEEKR (signal-to-background of 0.152 : -0.003 at kmer length k=6), and Stretcher (signal-to-background of 0.525 : 0.307; Fig. 1D). We conclude that kmer-based classification can detect sequence similarity between evolutionarily related lncRNAs.

We next examined if SEEKR could detect novel forms of similarity between lncRNAs with no known sequence homology. We created kmer profiles for all lncRNAs in the human and mouse GENCODE databases [12], as well as for select lncRNAs that were not included in GENCODE. Next, we compared kmer profiles between all lncRNAs in each organism using Pearson's correlation and hierarchically clustered the resulting matrices to examine the patterns that emerged. Consistent with our hypothesis, clustering lncRNAs by SEEKR grouped many by known function in human and mouse (Fig. 2). Several known *cis*-repressive lncRNAs, including *XIST*, *TSIX, KCNQ1OT1*, *UBE3A-ATS*, *ANRIL/CDKN2B-AS1*, and *Airn* clustered together due to high abundance of AU-rich kmers, whereas several *cis*-activating lncRNAs, including *PCAT6*, *HOTTIP*, *LINC00570*, *DBE-T*, and *HOTAIRM1*, clustered separately due to high abundance of GC-rich kmers (Figs. 2A and D). These patterns were robust over differing kmer lengths (Supplementary Fig. 2). To determine if this level of clustering was significant, we curated lists of human and mouse *cis*-activating and *cis*-repressive lncRNAs from the literature (Supplementary Table 1), and compared average pairwise kmer similarities between lncRNAs in each list to pairwise similarities of 10,000 size-matched lists of randomly selected lncRNAs from the respective organism. Human and mouse *cis*-repressors, and human *cis*-activators (but not mouse *cis*-activators), were significantly more similar to each other than expected by random chance (Supplementary Table 2). Concordantly, SEEKR detected significant similarity between the *cis*-repressive *Kcnq1ot1* and *Xist* lncRNAs where none was found by conventional alignment algorithms (Supplementary Fig. 1). We conclude that lncRNAs of related function can have related kmer profiles even if they lack linear sequence similarity.

Unexpected relationships also emerged in the hierarchical clusters of Fig 2. Most notably, the lncRNAs *NEAT1* and *MALAT1* showed greater than average similarity to *XIST* in both human and mouse. Among all human lncRNA pairwise comparisons, their Pearson's r values fell in the 99.99th and 99.60th percentile, respectively. Likewise, in mouse, the similarities were in the 97.15th and 95.32nd percentiles. The meaning of the similarity between the three lncRNAs is unclear, but we note that all three lncRNAs seed the formation of sub-nuclear compartments and engage with actively transcribed regions of the genome [28-33]. We speculate that their kmer similarity is related to these shared actions.

2.2.2 LncRNAs can be partitioned into communities of related kmer content

We next used a network-based approach to partition lncRNAs into communities of related kmer profiles, reasoning that such communities would provide a framework to understand the predictive value of lncRNA kmer content. We created networks of relationships between all human and mouse lncRNAs in which weighted edges connected lncRNAs in an organism if the Pearson's correlation between their standardized kmer profiles met a threshold for similarity (Methods). We then used the Louvain method to assign lncRNAs within the largest connected component of the network representations to communities of related kmer profiles [34]. Approximately half of all GENCODE lncRNAs grouped into five major communities in both human and mouse. LncRNAs not assigned to the five most populated communities were assigned to a "null" community. Our network-based approach and hierarchical clustering grouped lncRNAs in similar ways (p < 1e-324, Chi-squared; Supplementary Tables 3 and 4), signaling community robustness. LncRNA community assignments and associated summary statistics are provided in Supplementary Tables 5-12 and Supplementary Fig. 3. Differences in human and mouse community structures may be due in part to differences in completeness of lncRNA annotation. In the versions of GENCODE used for this work, there were about twice as many lncRNAs annotated in human (v22, n=15953) as there were annotated in mouse (vM5, n=8245 [12]).

2.2.3 Kmer content correlates with localization and protein binding

We next examined whether lncRNAs with related kmer profiles shared biological properties. For this analysis, we focused on human lncRNAs, where data from the ENCODE project allowed us to examine lncRNA subcellular localization and protein associations, transcriptome-wide. To determine whether kmer content provides information about lncRNA localization, we examined ENCODE subcellular fractionation RNA-Seq experiments performed in HepG2 and K562 cells [35]. For each lncRNA expressed in each cell type, we computed its nuclear ratio and determined whether the distributions of nuclear ratios differed between communities. The majority of communities showed slight but significant differences in their distribution of nuclear ratios, with the largest differences found between communities #1 and #3 (Fig. 3A and Supplementary Tables 13-16). Concordantly, lncRNAs that associate with polysomes in K562 cells [36] were also non-uniformly distributed between communities (p = 3.5e-5, Chi-squared), and were the most over- and under-represented in the most cytoplasmic and nuclear lncRNA communities, respectively (communities #3 and #1 being the most cytoplasmic and nuclear, respectively; Supplementary Table 17). Lastly, we used ENCODE data to identify the most cytoplasmic and nuclear lncRNAs in HepG2 and K562 cells and determine which kmers were asymmetrically distributed between lncRNAs in the two compartments. 360 and 27 kmers were significantly enriched in cytoplasmic and nuclear lncRNAs, respectively (p-adjusted <0.05; Kolmogorov–Smirnov test; Supplementary Table 18). Consistent with our RNA-Seq and polysome analyses, 58 and 93% of the cytoplasmic- and nuclear-biased kmers were the most enriched in the most cytoplasmic and nuclear lncRNA communities, respectively (communities #3 and #1; Supplementary Table 18, last column). We conclude that kmer content provides information about the subcellular localization of a lncRNA.

To determine if kmer content provides information about protein binding in lncRNAs, we examined ENCODE data for 156 eCLIP experiments performed for 109 proteins in HepG2 and K562 cells [37]. We created binary vectors for each experiment that recorded whether the lncRNAs bound or did not bind a given protein, then built separate logistic regression models for each protein to determine if kmer community assignments could improve prediction of lncRNA/protein associations over a null model that only included lncRNA length and expression as covariates. LncRNA community assignments significantly increased the log-likelihood of detecting lncRNA/protein associations for the majority of proteins examined (p-adjusted <0.05;146/156, ~94%; Fig. 3B and Supplementary Table 19). Increases in precision and recall in community-informed models were generally modest but significant (Fig. 3B and Supplementary Table 20). In total, ~17% (25/146) of our models had an increase in precision and/or recall of 5% or more. Notably, in all cases in which recall increased, precision also increased, indicating that kmer community information increased the ability to predict true lncRNA-protein associations and simultaneously increased the fidelity of those predictions. When we used individual 6mers instead of lncRNA communities as predictive features, results were no better than the null model that used only lncRNA length and expression as predictive features. Models with more features than samples are prone to learning noise in their training set, and often lose predictive power, due to overfitting [38]. Using individual 6mers brought the number of features being evaluated to 4099, more than the number of lncRNAs expressed in HepG2 and K562 cells (3745). We conclude that kmer content provides information about the protein-binding potential of a lncRNA, but that no single kmer provides an overwhelming portion of that information, and, that kmer communities provide a way to collapse high-dimensional kmer matrices down to representative variables for predictive purposes.

Protein binding to RNA is difficult to assess from motif content alone due to the degeneracy of most motifs and the challenge of predicting the effects of RNA structure [24,39-41]). Supporting this notion, we found that the abundance of motif-matching kmers was consistently, but not always, higher in the communities enriched for binding of specific proteins than in the cognate communities not enriched for binding, indicating that factors in addition to motif abundance control protein/lncRNA associations (Fig. 3C). We therefore sought to determine if kmer content could distinguish between motif matches in lncRNAs that coincide with protein binding events and those that do not. We searched the lncRNAs expressed in HepG2 and K562 cells for matches to binding motifs of the 17 proteins in Fig. 3C, whose position weight matrices were determined from biochemical assays in [23]. We annotated motif matches that fell inside and outside of CLIP peaks as true and false positive matches, respectively. As expected, the majority of motif matches fell outside of CLIP peaks (i.e., they were false positive matches; Supplementary Table 21). We then used SEEKR to compare regional kmer content in 300 nucleotide windows surrounding true and false positive motif matches. Remarkably, for 13 of 17 proteins examined, kmer profiles of true positive binding regions were more similar to each other than kmer profiles of randomly selected, size-matched sets of false positive regions (p-value < 0.005; Supplementary Fig. 4). These data support the notion that binding modules for the same protein in different RNAs often have sequence similarity that extends beyond the protein binding motif, and that this similarity can be quantified, in part, by local kmer content.

Moreover, SEEKR provides a simple way to visualize the density of specific kmers within CLIP enriched regions. We compared the most overrepresented kmers in true positive binding regions to protein binding motifs measured *in vitro* [23], and found that their relationships differed substantially from protein to protein (Fig 3D and Supplementary Fig. 5). For certain proteins, such as HNRNPC, KHDRBS1, and QKI, the most enriched kmers in true positive regions matched the PWM for the protein that was determined *in vitro* [23]. We interpret this observation to mean that for these proteins, motif density plays a dominant role in determining RNA binding *in vivo*, because our kmer data show that motif-matching kmers are more abundant in true positive regions than they are in false positive regions. For other proteins, such as FXR1, IGFBP1, and TIA1, the most enriched kmers in true positive regions did not match the PWM determined *in vitro* [23]. For these proteins, sequence beyond the binding motif may play a dominant role in dictating association with RNA, possibly due to effects from RNA structure. When PWMs were extracted from eCLIP peaks, similar relationships between kmers and *in vitro* defined motifs were observed (Supplementary Fig. 5). These results show how SEEKR can be used to augment traditional motif-based analyses and provide insights into mechanisms of RNA-protein interaction. SEEKR provides a way to quantify sequence similarities between any number of protein binding regions, which in turn, can provide predictive power and identify shared characteristics that are not apparent from PWM-based motif analyses.

2.2.4 Similarities in lncRNA communities between organisms

Given (i) that kmer content provides some indication of protein binding potential in a lncRNA, (ii) that sequence specificities of many RNA binding proteins are conserved [23,24], and (iii) that protein binding likely dictates lncRNA function, we hypothesized that kmer contents between communities of functionally related lncRNAs could be conserved even if the lncRNAs themselves lack known evolutionary relationships. In support of this idea, we identified extensive similarity between certain human and mouse lncRNA communities via SEEKR (Methods; Supplementary Fig. 6). Most notably, lncRNAs in human community #1 (the "*XIST*" community) had kmer profiles that were, as a group, nearly indistinguishable from lncRNAs in mouse community #1 (the "*Xist*" community) and were also similar to lncRNAs in mouse community #4 ($p<0.0001$ for both comparisons). Human community #2 and community #3 (the "*HOTTIP*" community) were both similar to mouse community #2 (the "*Hottip*" community; $p<0.0001$). No other major similarities between mouse and human were apparent. Extending this analysis across greater evolutionary distance, we found *HOTTIP*-like lncRNA communities in ten of ten vertebrates examined as well as in the sea urchin *S. purpuratus*, and *XIST*-like lncRNA communities in seven of ten vertebrates examined (Supplementary Figs. 7-9; [10]). These analyses demonstrate that, at the level of kmers, subsets of human lncRNAs are more similar to lncRNAs in other genomes than they are similar to lncRNAs in their own genome, supporting the idea that groups of lncRNAs have similar function in different organisms despite lacking obvious linear sequence similarity.

2.2.5 SEEKR can predict *Xist*-like regulatory potential in lncRNAs

We next directly tested whether kmer profiles could be used to predict lncRNA regulatory potential. We focused on the ability of certain lncRNAs to repress transcription in *cis*. *Cis*-repression was one of the earliest characterized functions of lncRNAs, and is essential for normal human health and development. In the most striking example, the *XIST* lncRNA silences nearly all genes across an entire chromosome during X-chromosome Inactivation [7]. *Cis*-repression is also one of most straightforward lncRNA functions to study because, by definition, *cis* acting lncRNAs act near their site of transcription.

We developed a reductionist assay to study lncRNA *cis*-repressive activity in a normalized genomic context, called TETRIS (transposable element to test RNA's effect on transcription in *cis*). TETRIS enables the sequence of a lncRNA and an adjacent reporter gene to be manipulated in a plasmid, but then rapidly inserted into chromosomes via the piggyBac transposase [42,43], so that effects of the lncRNA on the reporter can be studied in genomic chromatin (Fig. 4A and Methods). Under our assay conditions, piggyBac catalyzes 4-7 insertions of each cargo per stably selected cell, and cell density estimates suggest between 100,000 to 500,000 cells receive insertions and survive selection (Fig. 4B and not shown). Thus, each TETRIS assay likely surveys 400,000 to 3.5 million insertion events. Insertion-site dependent variation in lncRNA-induced effects are averaged out in the population, bypassing the need to isolate clones of modified cells, and providing the means to quantify lncRNA regulatory potential without influence from genomic position.

We validated TETRIS by comparing effects that expression of different lncRNAs had on luciferase activity. A cell line created from a vector that lacked a lncRNA insert (TETRIS-Empty) showed a ~2-fold increase in luciferase activity upon addition of doxycycline, representing our baseline for the assay (Fig. 4C). We attribute this mild activation to the close proximity of the dox-inducible and luciferase promoters, and to the fact that both promoters are contained within the same insulated domain [44]. By contrast, expression of the first 2kb of *Xist* repressed luciferase 5-fold relative to uninduced control (Fig. 4C). The 2-fold activation and 5-fold repression were stable across nine and 16 independent derivations of TETRIS-Empty and TETRIS-*Xist*-2kb cell lines, respectively (mean ± standard deviation of 2.03 ± .50 and 0.23 ± .08), demonstrating that TETRIS assays result in reproducible effects on luciferase activity. For its repressive effect, *Xist* requires "Repeat A," a 425-nucleotide long element contained within its first 2kb [45]. In the context of TETRIS, deletion of Repeat A resulted in a significant, but not complete, de-repression of luciferase, whereas expression of Repeat A alone resulted in repression relative to control, but at reduced levels compared to *Xist*-2kb (*"ΔrepA"* and "*repA only"*; Fig. 4C). Similarly, expression of the first 5.5kb of *Xist* caused a 5-fold repression of luciferase, whereas deletion of the first 2kb from the 5.5kb construct caused complete loss of repressive activity (*"Xist-5.5kb"* and "*Xist-2-5.5"*; Fig. 4C). Expression of either the final 3.3kb of *Xist* or the *Hottip* lncRNA had no repressive effect (Fig. 4C). These experiments demonstrate (i) that TETRIS is a suitable assay to measure repression by *cis*-acting lncRNAs in a normalized genomic context, and (ii) in the assay, sequence elements in addition to Repeat A cooperate to encode repressive function in the 5´ end of *Xist*.

We next used TETRIS and SEEKR to test our hypothesis that kmer content can predict lncRNA regulatory potential. We reasoned that we could design entirely synthetic lncRNAs that lacked linear sequence similarity to any known lncRNA but nonetheless had robust *Xist*-like repressive activity. We generated six synthetic lncRNA sequences in silico with varying levels of kmer similarity to the first 2kb of *Xist*, and cloned them into TETRIS to measure their effects on luciferase activity. As measured by SEEKR, the lncRNAs had Pearson's similarities to *Xist* that ranged from average (a Pearson's r of ~0) to three standard deviations above the mean similarity for all mouse lncRNAs (a Pearson's r of 0.19, more similar to *Xist*-2kb than all other lncRNAs the mouse genome; Fig. 4D). Using nhmmer or Stretcher to align the synthetic lncRNAs to the first 2kb of *Xist* produced either no alignments (nhmmer) or alignments that differed by only three percent across all six synthetic lncRNAs (Stretcher; Fig. 4E, grid below graph). Via BLAST, the lncRNAs had no significant similarity to the mouse genome or to each other (not shown). The lack of informative alignments was expected because the synthetic lncRNAs have no evolutionary relationship with *Xist*, any region in the genome, or each other. Nevertheless, as envisioned, the synthetic fragments that SEEKR classified to be most similar to *Xist* had the highest repressive activity (Fig. 4E). These data directly demonstrate that evolutionarily unrelated lncRNAs can encode similar function through different spatial arrangements of related sequence motifs. Thus, kmer content can be used to predict lncRNA regulatory potential.

We next examined whether SEEKR could predict *Xist*-like repressive activity in endogenous lncRNAs. We cloned into TETRIS thirty-three lncRNAs or lncRNA fragments that had a range of kmer similarities to the first 2kb of *Xist*. Included in our final set of fragments were several conserved lncRNAs and/or shorter fragments contained within them (*Airn, Hottip, Kcnq1ot1, Malat1, Neat1,* and *Pvt1*), as well as many lncRNAs with uncharacterized functions (Supplementary Table 22). Again, the more *Xist*-like a lncRNA fragment was at the level of kmers, the more likely it was to repress in TETRIS; the Pearson's r value between *Xist*-likeness at a kmer length of 6 and luciferase activity upon dox addition was -0.41 (p=0.02). Including the six synthetic lncRNAs in the correlation brought the Pearson's r value to -0.52 (p=0.0007; Fig. 4F). Nhmmer and Stretcher had no ability to predict repressive activity, demonstrating that these algorithms cannot detect sequence signatures correlated with repressive activity in this setting (p=0.32 and 0.91, respectively; Fig. 4G and H). LncRNA fragment length also had no ability to predict repressive activity (r=0.03, p-value=0.84).

Lastly, we examined whether kmer profiles associated with sequence elements required for repression by *Xist*-2kb might increase our ability to predict repressive activity in other lncRNAs. To determine the elements in *Xist*-2kb required for repression, we made a series of 26 deletions (Fig. 5). Surprisingly, 15 of the deletions, including ones that removed predicted stable structures, pseudoknots, and ~40% of Repeat A ("ΔSS1", "ΔSS2", "ΔPK2", "ΔSS3", "ΔSS4"; bottom panel in Fig. 5; [41]), had no significant effect on repression. However, removal of all eight GC-rich portions of Repeat A, but not its U-rich linkers, caused a ~3-fold reduction in repression ("ΔGC repeat in rA" vs "ΔU spacer in rA"), as did removal of three predicted stable structures and their intervening sequences in the 742 nucleotides immediately downstream of Repeat A ("ΔSS2/3/4 broad"; [41]). Co-deletion of Repeat A and the stable structures had an additive effect, causing a near complete loss of repression (the "ΔrAΔSS234 br." mutant), whereas expression of Repeat A or the stable structures alone had half the repressive potency of *Xist*-2kb ("Only rA" and "Only SS234"). Expression of both regions together had the same repressive potency as *Xist*-2kb ("Minimal"). Thus, in TETRIS, the major elements required for repression are contained between nucleotides 308 and 1,476 of *Xist*. Based on prior structural models [41,46], we infer that the elements are comprised of protein binding sites, spacer sequences, and stable structures.

Having mapped the elements responsible for repression in *Xist*-2kb, we attempted to extract subsets of 6mers from them that increased our ability to predict *Xist*-like repression. We also examined if kmer variance across lncRNA communities or kmer nucleotide composition could be used to extract subsets of outperforming 6mers, and if different kmer lengths had better predictive power than k=6. No rationally designed subset of 6mers could predict repression better than the full 6mer profile of *Xist*-2kb, nor could any other kmer length (Supplementary Fig. 10). These results support the ideas that different lncRNAs can encode similar function through related, but not necessarily identical, sequence solutions, and that the full complement of 6mers may be a broadly effective search tool to identify such similarities (not too relaxed, not too stringent).

2.3 Discussion

Collectively, our data support the notion that many lncRNAs function through recruitment of proteins that harbor degenerate RNA binding motifs, and that spatial relationships between protein binding motifs in these lncRNAs are often of secondary importance to the concentration and effectiveness of the motifs themselves. By this logic, a lncRNA may merely need to present the appropriate motifs embedded within the appropriate structural contexts to achieve a specific function. Thus, different lncRNAs likely encode similar function through vastly different sequence solutions, and nonlinear sequence comparisons can be used to discover similarities between them. By extension, because the RNA binding motifs of many proteins are conserved [23,24], it is likely that groups of lncRNAs rely on similar motifs to encode related function in different organisms even though they lack direct evolutionary relationships. This concept is supported by our observation that lncRNA communities with related kmer contents exist in human, mouse, and other organisms. We propose that nonlinear sequence homology – in which the relative abundance of a set of protein binding motifs is conserved, but the sequential relationships between them are not – is prevalent in lncRNAs. To quantify nonlinear homology, we introduce SEEKR, a method to compare sequence content between any group of lncRNAs, regardless of the size of the group, the evolutionary relationships between the lncRNAs being analyzed, or the differences in their lengths. Each lncRNA (and each functional domain within each lncRNA) has its own kmer signature, which can encode information about protein binding and RNA structure. SEEKR provides a simple way to tie this information to a biological property.

2.4 Methods

2.4.1 Kcnq1ot1 versus Xist comparison

Kcnq1ot1 was aligned to *Xist* using nhmmer and Stretcher with default parameters. To assess significance of the alignments, we generated 1,000 pseudo-*Kcnq1ot1*s that were the same length of real *Kcnq1ot1* but composed of nucleotides randomly selected from a distribution of the mononucleotide content of *Kcnq1ot1* (0.335 A: 0.205 G: 0.202 C: 0.258 T). We then aligned the pseudo-lncRNAs to *Xist* with nhmmer and Stretcher as well as compared their kmer contents relative to all other mouse lncRNAs at kmer length k=6 via SEEKR.

2.4.2 SEEKR

In SEEKR, a matrix of kmer counts for a user-defined set of lncRNAs is created by counting all occurrences of each kmer in each lncRNA in one-nucleotide increments, and then dividing those counts by the length of the corresponding lncRNA. Z-scores are then derived for each kmer in each lncRNA by subtracting the mean length-normalized abundance of each kmer in the group of lncRNAs being analyzed from the length-normalized abundance of the kmer in the lncRNA in question, and then dividing that difference by the standard deviation in abundance of that kmer in the group of lncRNAs being analyzed. We refer to the array of z-scores for each kmer in a given lncRNA as its kmer profile. Similarity between any two lncRNAs can be calculated by comparing their kmer profiles with Pearson's correlation.

Our rationale for length normalization in SEEKR follows. Without length normalization, kmer profiles become difficult to interpret for lncRNAs of different lengths. For example, an RNA that is 10x longer than another RNA will have 10x the number of kmers. Without normalization, these lncRNAs would be considered dissimilar by SEEKR, regardless of the similarity in their relative concentrations of kmers. By length normalizing, SEEKR creates a list of relative kmer concentrations in a given lncRNA that is robust to differences in length. The idea that length normalization is important is supported by studies of known *cis*-repressive lncRNAs. At 18kb, the *Xist* lncRNA is the most potent *cis*-repressive lncRNA known. At least three other known *cis*-repressive lncRNAs are longer than *Xist*, but less potent: *Airn*, *Kcnq1ot1*, and *Ube3a-ATS*, are 90kb, 85kb, and 1.1Mb, respectively [7]. Of these, the longest lncRNA, *Ube3a-ATS*, is the least potent, arguing that length alone does not account for lncRNA potency. In certain biological contexts, lncRNA length may not be relevant, or it may have varying influence on lncRNA function. However, what these contexts might be and to what extent length does or does not affect lncRNA function in them are not known and difficult to predict. We also note that Pearson's correlation inherently normalizes for length. Thus, comparisons of kmer content that use Pearson's correlation will eliminate length as a variable.

2.4.3 GENCODE lncRNA annotations

All GENCODE annotations used in this work were from human build v22 and mouse build vM5 [12]. For each lncRNA, only the major splice annotation was considered (the -001 isoform). In total, there were 15953 human and 8245 mouse transcripts. The heat maps in Fig. 2 were generated with GENCODE annotations plus the additional lncRNA sequences downloaded from the UCSC genome browser [47]: *SAMMSON*, *XACT*, *UBE3A-ATS*, *MORRBID,* and *NESPAS,* (Human), and unspliced *Airn*, *Anril*, *Bvht*, *Haunt*, *Morrbid*, unspliced *Tsix*, *Ube3a-ATS, XistAR, and Upperhand* (Mouse).

2.4.4 Conservation analysis

Ninety-three pairs of human and mouse GENCODE lncRNAs were recently identified as putative homologues due to their high conservation at the DNA level [27]. These 93 lncRNAs, plus an additional 68 lncRNA pairs that had equivalent names in mouse and human GENCODE annotations, formed the final set of 161 homologues that were used for the conservation analysis of Fig. 1C. For the Fig 1C. experiment, "signal" values were computed as the mean of the 161 homologue-to-homologue measurements in each of the three algorithms; likewise, background values were computed as the mean of the remaining 12880 non-homologous comparisons. Homologous pairs were defined as being "detected" if the signal value/average similarity (as determined via SEEKR, nhmmer, or Stretcher) was higher for homologue-to-homologue measurements than it was for all other lncRNA-to-non-homologue comparisons. For this analysis, nhmmer was downloaded as part of the HMMER package (URLs) and was run with --nonull2, --nobias, --noali, and -o flags set. Stretcher was used as part of Biopython (URLs) and was run with --gapopen=16, and –gapextend=4.

2.4.5 Hierarchical clustering and labeling

Hierarchical clustering was performed with the R package "amap" using Pearson's as a distance metric and average linkage [48], and was visualized with Java Treeview [49]. We used kmer length k=6 for our main analyses because it performed well in evolutionary comparisons (Fig. 1C), and it provided a feature number (4^6 = 4096 features) that is only marginally larger than the average length of a GENCODE lncRNA (1152 and 1471 nucleotides for human and mouse lncRNAs, respectively).

2.4.6 Clustering of known *cis*-activating and *cis*-repressive lncRNAs

We performed a literature review to curate lists of experimentally verified *cis*-repressive and *cis*-activating lncRNAs in mouse and human (Supplementary Table 1). We calculated the mean pairwise similarity between all lncRNAs in each of these groups, and compared those means the distribution of mean similarities calculated from pairwise comparisons of 10,000 randomly selected, size-matched groups of lncRNAs in their respective organism to generate p-values that describe the likelihood that the similarity observed between the functionally related *cis*-acting lncRNAs was greater than would have been expected from random chance (Supplementary Table 2).

2.4.7 Network analysis and lncRNA community definition

Networks of lncRNAs were formed from a weighted adjacency matrix in which edges between any two lncRNAs were kept only if their Pearson's r-value was at least 0.13. We selected the lncRNAs within the largest connected component of this network representation and used the Louvain algorithm [34] at default resolution parameter to assign lncRNAs to communities of related kmer profiles (using the Python package "louvain-igraph"). This decision was supported through use of the recently developed CHAMP algorithm [50] (URLs), which found a wide domain of optimality around the default resolution parameter. We retained assignments for the lncRNAs present in the top five most populated communities, and assigned the remaining lncRNAs, including those not found in the largest connected component of the network representation, to the "null" community, which served as an important outgroup for our comparisons of kmer content and biological properties in Fig. 3. Multiple Pearson's r value thresholds between 0.12 and 0.21 were tested for human lncRNAs and we found little to no difference in community definition, correlation with lncRNA localization, or ability to predict protein-binding patterns (not shown). Gephi was used for network visualization (URLs). Community colors were automatically assigned by Gephi according to the size of each community.

We also compared communities generated with 5mers and 7mers to those generated with 6mers. We created contingency tables that compared the distribution of lncRNAs in each of the five major 6mer communities plus the null to the distribution of lncRNAs in each of the five major 5mer and 7mer communities plus their respective nulls. P-values comparing communities between the kmer lengths were all < 1E-324 (chi-squared), indicating that community definitions are largely stable when 5mers, 6mers, or 7mers are used (Supplementary Table 9 and 10). This stability, the quality of our TETRIS predictions when using 6mers (Supplementary Fig. 10), and the computational inefficiency of performing operations on matrices of 7mers or greater provided additional support for our decision to use 6mers for the bulk of our analyses.

We applied the same r-value threshold and community assignment logic that we used for human lncRNAs to define lncRNA communities using kmer length k = 6 in all other organisms.

2.4.8 Comparing lncRNA groups in hierarchical clusters to lncRNA communities found by Louvain

Clusters of lncRNAs with similar kmer content in human and mouse (from Fig 2.) were created by manually making cuts in the dendrogram of the hierarchical clusters that maximized the visual similarity of kmer profiles between lncRNAs in each cluster. Five cuts were made in the hierarchical cluster from each organism to approximate the five major communities found by the Louvain algorithm. We measured the similarity of the manually made clusters to the five major Louvain-defined communities by a creating contingency table that compared lncRNA distributions between the two methods. We then tested if the distribution of lncRNAs across the two sets of communities were significantly similar via a chi-squared test. In both human and mouse, the p-value was < 1E-324 (Supplementary Table 3 and 4).

2.4.9 LncRNA localization analysis

Localization data were downloaded from ENCODE (URLs) as fastq files and aligned to GRCh38 with STAR using default parameters [47,51]. FeatureCounts was used to tabulate the number of reads aligning to our set of lncRNAs [52]. We then filtered out all lncRNAs with <0.1 RPKM from each community, and calculated the number of reads in the nuclear fraction over the total number of reads from both the nuclear and cytosolic fractions for each lncRNA.

To determine if specific kmers were enriched in cytosolic or nuclear lncRNAs, we selected cytosolic- and nuclear-enriched subgroups of lncRNAs that were expressed in HepG2 or K562 cells. Because the subcellular distribution values for HepG2 or K562 expressed lncRNAs were not normally distributed (Fig. 3A), we needed to employ different thresholds to define cytosolic and nuclear so that the two groups would include similar numbers of lncRNAs. “Cytosolic” lncRNAs were defined as any lncRNA that was more than 50% cytosolic, which resulted in 2801 transcripts, and “nuclear” lncRNAs were defined as any lncRNA that was more than 95% nuclear, which resulted in 4576 transcripts. To determine the average difference in kmer abundance between lncRNAs in the two compartments, we calculated the mean value of the z-scores for each kmer in each group, and then used the difference between the means as the metric to calculate the nuclear-enrichment score (Supplemental Table 18). To test for significant differences between the distributions of z-scores between lncRNAs in the two compartments, we used a KS-test and calculated an adjusted p-value using a Bonferroni correction. This analysis yielded 387 kmers whose distributions differed significantly between cytosolic and nuclear lncRNAs (p-value < 0.05; Supplemental Table 18).

Using only the lncRNAs from community 3, we repeated the process of applying the Louvain algorithm to define communities and measure cellular localization in order to rule out the possibility that potential sub-communities were responsible for the cytosolic nature of community 3. The Louvain algorithm found four main sub-communities and all smaller sub-communities were grouped into a fifth community. The results of ANOVA tests indicated there was no significant differences between any of the communities for either the polyA-selected or ribosome-depleted RNA RNA-Seq data. We performed this analysis again for community 1, but no sub-communities were found to be significantly different (Supplementary Fig 11). This uniformity of cellular localization among possible sub-communities provides biological support for our original community definitions.

2.4.10 lncRNA polysome association

A recent study found 229 lncRNAs in GENCODE v22 that were polysome associated in K562 cells [36]. A chi-squared test showed these 229 lncRNAs were non-randomly distributed between the communities (p-value = 3.5E-5; Supplementary Table 17). The expected values for the chi-squared test were calculated by filtering all communities for lncRNAs expressed in K562 cells, dividing the number lncRNAs in each community by the total number of expressed lncRNAs (3277), and multiplying by the number of polysomal lncRNAs (229).

2.4.11 LncRNA protein association data

eCLIP data were downloaded from ENCODE [35,37]. For each of the 156 eCLIP experiments "bed narrowPeak" data (representing sites of protein binding that passed a ENCODE-defined threshold for enrichment over background; [35,37]) were pooled from available biological duplicates. Genomic coordinates were overlapped with lncRNA exon coordinates annotated by GENCODE. Any lncRNA which overlapped with one or more eCLIP peak was considered as having a true binding interaction with the given protein. LncRNA expression data were collected from ENCODE RNA-Seq experiments in the same cell type as that of the eCLIP experiment (HepG2 or K562).

For each protein, a vector was built for each lncRNA that encoded whether the protein-lncRNA pair did or did not interact. Next, two feature matrices (null and full) were constructed. The null matrix included the log normalized values for length and expression of each of the lncRNAs. The full matrix included log normalized length and expression, as well as an additional five columns that corresponded to each of the five lncRNA communities. Each lncRNA was assigned a value of "1" in the column representing its community.

2.4.12 Models of protein associations

To address if lncRNA communities contained information about lncRNA/protein associations, we used a machine learning model [53]. We tested if providing the model with the community data allowed it to predict interactions better than a corresponding null model that was not given the community data but still included lncRNA length and expression values as covariates. Logistic regression models were implemented with scikit-learn, using default parameters [53]. The significance of the additional community information was measured with a likelihood ratio test (LRT), where the LRT statistic, D, equaled:

$$D = 2 * [\log(full\ model\ likelihood) - \log(null\ model\ likelihood)]$$

A chi-squared distribution was used to determine the corresponding p-value for the LRT statistic. P-values were adjusted with a Bonferroni correction for the 156 comparisons.

To quantify the extent of the effect that community inclusion had on prediction of lncRNA/protein interactions, we used a Leave-One-Out-Cross-Validation approach to measure precision and recall metrics [53], defined as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

In our model, precision is the number of lncRNAs correctly predicted to bind a protein, divided by the total number of lncRNAs the model predicted to bind a protein. Recall is the number of lncRNAs the model correctly predicted to bind a protein, divided by the total number of lncRNAs found to bind a protein according to the eCLIP data. For each lncRNA, the logistic regression models were allowed to train on all other lncRNAs except the single "left out" lncRNA. After training, both models were asked to predict if the "left out" lncRNA did or did not bind the protein. This procedure was repeated for all lncRNAs in each eCLIP dataset to calculate precision and recall.

The methodology for training and testing the raw kmer models was exactly the same as described above except that the five community features were replaced by the 4096 relative kmer abundance features.

2.4.13 Calculating the abundance of motif-matching kmers in lncRNA communities

The data for the bar graph in Fig. 3C were generated by the following approach. Of the 109 proteins on which eCLIP was performed in [37], 79 showed significant association with at least one kmer community over the null (Supplementary Table 19). Of these 79 proteins, binding motifs for 17 were determined via an *in vitro* binding assay in [23]. The PWMs for each of these 17 proteins contained relative weights for each motif matching 6mer, representing the likelihood that the kmer in question would bind the protein in question. We multiplied the weight of each motif-matching 6mer by its average standardized abundance in each of the six communities, including the null, to obtain kmer abundances that were scaled by the likelihood that the kmer in question matched the binding motif in question. For each of the 17 proteins, sums of the weighted abundance for all motif-matching kmers were created for the communities in which protein binding was enriched and not enriched over the null, respectively, then divided by the number of communities in each group to obtain the average weighted abundance of motif-matching kmers in the binding-enriched and binding-not-enriched groups. These abundances are plotted in Fig. 3C. For proteins that had more than one PWM reported in [23], the average abundance shown in Fig. 3C is comprised of the weighted abundance averaged over all reported PWMs. To calculate significance, we shuffled the communities in the binding-enriched and binding-not-enriched groups 10,000 times and determined how often the difference in kmer abundance between the randomly shuffled binding-enriched and binding-not-enriched groups was greater than the difference between the real binding-enriched and binding-not-enriched groups.

2.4.14 Measuring kmer similarity surrounding motif matches in lncRNAs

The lncRNAs expressed in HepG2 and K562 cells were examined for motif matches to the 17 proteins for which eCLIP data was reported in [37] and whose PWMs were determined via a high-throughput *in vitro* assay in [23] by using FIMO at a threshold of p<0.01 (from the MEME suite, URLs; [54]; Supplementary Table 21). Each motif match was then labeled as a true positive if it overlapped an eCLIP peak, or a false positive if it did not. For each protein, the sequences surrounding the center of each true and false positive motif match (up to 150bp on either side of the center, or up to the end of the gene, whichever came first) were collected and their kmer contents were analyzed with SEEKR. Significance of the similarity between true positive regions was measured by a permutation test against randomly selected sets of false positive regions controlling for both the size of the set and the number of overlapping regions in the set (Supplementary Fig. 4).

2.4.15 Identifying motifs from eCLIP peaks

To find motifs in eCLIP peaks for the 17 proteins listed in Fig. 3C, we extracted the subset of sequences from eCLIP peaks whose CLIPper-defined p-value was <0.001 (peaks with the highest read densities relative to control; [37]). We searched these sequences for motifs using DREME at default parameter as a part of the MEME-ChIP package [55].

2.4.16 Human-to-mouse and human-to-other community similarity calculations

To evaluate the similarity between human and mouse lncRNA communities, we calculated the distribution of similarities between all pairwise combinations of lncRNAs within each human kmer community ("human-to-self"), and compared this distribution to: (1) a distribution of pairwise comparisons made between all other human lncRNAs excepting lncRNAs from the community in question ("human-to-other-human"), (2) distributions of all pairwise comparisons made between all lncRNAs in each of the five mouse lncRNA communities ("human-to-mouse"), and (3) distributions of all pairwise comparisons made between all human and mouse lncRNAs that did not fall into one of the five major communities ("human-to-null"). We then performed a permutation test to determine whether a given human community was similar enough to a mouse community to overcome its intrinsic similarity to other lncRNAs in the human genome. The expectation was that, for related communities, the human-to-mouse distribution would be more similar to the human-to-self distribution than it would be to the human-to-other-human and human-to-null distributions. Bonferroni-adjusted p-values were calculated by permutation tests where we iteratively subsampled 0.1-1% of each distribution, re-measured the mean pairwise similarities, counted number trials in which the "human-to-mouse" mean subsample was closer to the "human-to-other-human" mean than it was to the "human-to-self" mean, and finally, divided by the total number of trials performed (36,000). This bootstrapping procedure provided a statistical framework to determine if the similarities uncovered between human and mouse communities were greater than what would have been expected from random chance. For example, in each of 36,000 tests, the distribution of similarities between a randomly selected subset of lncRNAs from human community #1 and size-matched subsets of lncRNAs from mouse community #1 was always more similar to the distribution of similarities between all pairwise comparisons of the human community #1 subset than it was similar to the distribution of similarities between the human community #1 subset and size-matched subsets of non-community #1 human lncRNAs (see upper left panel in Supplementary Fig 6; “H-1 vs M-1” plot; the H-1-vs-H-1 distribution in red is nearly indistinguishable from the H-1-vs-M-1 distribution in purple).

To generate the plots in Supplementary Figs. 8 and 9, identical analyses were performed that compared human lncRNA communities to lncRNA communities from Rabbit, Dog, Opossum, Chicken, Lizard, Coelacanth, Zebrafish, Stickleback, Nile Tilapia, Elephant Shark, and Sea Urchin [10]. In these latter cases, the human *XIST* and *HOTTIP* lncRNAs were doped into the lncRNA annotation set from the organism in question to find the homologous communities that were the most *XIST*- and *HOTTIP*-like (Supplementary Fig. 7).

2.4.17 Generation of plasmids for TETRIS assays

The pTETRIS-Cargo vector was created from components of a cumate-inducible piggyBAC transposon vector (System Biosciences), pGl4.10-Luciferase (Promega), and pTRE-Tight

(Clontech). Briefly, a 567bp fragment containing a minimal mouse PGK promoter was cloned into a SacI site in pGl4.10-Luciferase to generate pGl4-PGK-Luc-pA. The reverse complement of PGK-Luc-pA was cloned into a vector containing the bovine growth hormone polyA site. The entire bGHpa-[reversePGK-Luc-pA] was cloned into NotI and SalI sites of the piggyBAC vector (System Biosciences). The cumate-inducible promoter in the piggyBAC vector was then replaced with the Tetracycline Responsive Element (TRE) from pTRE-Tight (Clontech) via Gibson assembly to generate pTETRIS-Cargo in Fig. 4A, in which the lncRNA, the luciferase gene, and a gene encoding puromycin resistance are all flanked by chicken HS4 insulator elements, and inverted terminal repeats (ITRs) recognized by the piggyBAC transposase. The rtTA-cargo vector from Fig. 4A was generated by cloning the hUbiC-rtTA3-IRES-Neo cassette from pSLIK-Neo (Addgene Plasmid #25735) into SfiI and SalI sites in a piggyBAC transposon vector (System Biosciences). The piggyBAC transposase from System Biosciences was cloned into SmaI and HindIII sites into pUC19 (NEB) to allow propagation of the transposase on ampicillin plates.

2.4.18 Generation of TETRIS-lncRNA Cargo vectors

LncRNA fragments were PCR-amplified from genomic DNA or bacterial artificial chromosomes using Phusion DNA Polymerase (NEB), or commercially synthesized (Genewiz; IDT), and cloned via Gibson assembly into the SwaI site of pTETRIS-Cargo. Insert size was verified by restriction digestion, and the 5´ and 3´ end of each insert was verified by Sanger sequencing. To generate mutant *Xist*-2kb constructs, the 2kb fragment of *Xist* was subcloned into pGEM-T-Easy, and the regions in question were deleted using site-directed mutagenesis, or by synthesis of a mutated fragment and re-cloning back into compatible sites in pGEM-*Xist*-2kb (Genewiz). Deletions were verified by Sanger sequencing and then assembled into the SwaI site of pTETRIS-Cargo. The sequence of all inserted fragments, including *Xist*-2kb mutations, are listed in Supplementary Table 22.

2.4.19 Estimation of TETRIS copy number per cell

Genomic DNA was prepared from biological triplicate derivations of TETRIS-*GFP* and TETRIS-*Xist-2kb* cell lines. qPCR signal (SsoFast, Biorad) from the genomic DNA was compared to signal from a molar standard amplified from increasing amounts of the corresponding TETRIS plasmid (Supplementary Table 23).

2.4.20 TETRIS assays

To generate stable TETRIS-lncRNA cell lines, 8x10^5 E14 embryonic stem cells were seeded in a single well of a 6-well plate, and the next day transfected with 0.5μg TETRIS cargo, 0.5μg rtTA-cargo, and 1μg of pUC19-piggyBAC transposase. Cells were subsequently selected on puromycin [2μg/ml] and G418 [200μg/ml] for 6 to 12 days. Due to the efficiency of piggyBAC cargo integration and the rapidity of puromycin selection, all observable death from drug selection occurred within ~3 days after addition of puromycin and G418 (i.e. cells with puromycin resistance were invariably resistant to G418). For luciferase assays, 1x10^5 cells per well of 24 well plate were seeded in triplicate from each biological replicate preparation of a stable TETRIS-lncRNA cell line. 24 hours post-seeding, media was changed to include doxycycline at a final concentration of 1μg/ml. After two days of growth in dox-containing media, cells were lysed with 100 ul of passive lysis buffer (Promega), and luciferase activity was measured using Bright-Glo™ Luciferase Assay reagents (Promega) on a PHERAstar FS plate reader (BMG Labtech). Luciferase activity was normalized to protein concentration in the lysates via Bradford assay (Biorad). Each lncRNA fragment was assayed in at least in triplicate from at least two independent biological replicate preparations of stable TETRIS-lncRNA cell lines.

2.4.21 Synthetic lncRNA design

Synthetic lncRNAs were designed by generating 10 million, 1650 nucleotide long lncRNAs in silico that were composed of nucleotides randomly selected based on a given input ratio. To generate synthetic lncRNAs #2 through #6, the input ratio was the mononucleotide content of the 2,016-nucleotide long fragment of *Xist* inserted into TETRIS (0.203 A: 0.262 G: 0.204 C: 0.331 T). To generate synthetic lncRNA #1, the input ratio was an equal proportion of mononucleotides (0.250 A: 0.250 G: 0.250 C: 0.250 T). Synthetic lncRNAs with the specified kmer similarity to the 2kb fragment of *Xist* were then selected and synthesized as geneBlocks (Integrated DNA Technologies) and Gibson assembled into the SwaI site in TETRIS. Similarities in kmer content to the 2kb fragment of *Xist* are relative to all other mouse GENCODE lncRNAs.

2.4.22 Visualization of *Xist* structural models

Minimum Free Energy and probability-arc structural models of *Xist*-2kb were generated using SHAPE-MaP data from [41], the visualization package VARNA [56], and a modified version of the IGV browser [57]. Predicted pseudoknots and regions of low SHAPE reactivity and low Shannon Entropy in *Xist*-2kb are from [41].

2.4.23 TETRIS predictions for kmer sizes and subsets

We measured SEEKR's ability to capture the relationship between a lncRNA's *Xist*-likeness and its repressive ability in the TETRIS assay using kmers from size one to eight. In each case, the correlation is measured using the means of all biological and technical replicates of each real and synthetic lncRNA, by normalizing kmer counts of *Xist*-2kb and the lncRNA in question in context with all mouse GENCODE lncRNAs. This process was repeated for select subsets of kmers which had the potential to increase our ability to predict repressive activity in TETRIS. Individual subsets were created by counting and normalizing kmers as normal with SEEKR then removing columns of the resulting count matrix that were not included in a given subset. Additionally, we randomly generated 100,000 kmer subsets each containing between 2 and

4095 kmers, and measured each of the subsets Pearson’s r values relative to our TETRIS data (Supplementary Fig. 10).

2.4.24 Statistical analyses

All statistics were performed in Python or R. Details of statistical analyses are described in the corresponding sections. All multiple comparison tests were adjusted using a Bonferroni correction. p-values are reported as exact values except in cases where the p-value was calculated using a permutation test, and no random samples were found to be more extreme than the observed value. In these cases, p-values are reported as (p <= 1/n), where n is the number of permutations performed.

Supp. Table 2.1. (Corresponds to Supplementary Table 2) Relationship between lncRNAs with known transcriptional regulatory function as measured by SEEKR. “Species”, GENCODE set of lncRNAs. “Function”, the literature reported regulatory role of the lncRNAs. “Count”, the number of lncRNAs curated from the literature with a given function for a given species (full lists in Supplemental Table 1). “Mean”, the average Pearson’s correlation of all pairwise comparisons of lncRNAs in the set. “p-value”, the results of a permutation test of 10,000 random, sized matched sets of lncRNAs. SEEKR predicts that the lncRNAs in each of these classes are significantly more similar to each other than would be expected, with the exception of the mouse *cis*-activators.

Species	Function	Count	Mean	p-value
human	cis-repression	9	0.079	<0.0001
human	cis-activation	6	0.060	0.0014
mouse	cis-repression	8	0.072	0.0011
mouse	cis-activation	5	0.011	0.1592

Supp. Table 2.2. (Corresponds to Supplementary Table 3) Contingency table of Louvain communities and hierarchical clusters definitions in human. Each cell represents the number of lncRNAs that are found in both the corresponding row and column labels when groups of lncRNAs are defined using either the Louvain or hierarchical method. The large values along the diagonal indicate that the group definitions are stable with respect to the particular algorithm used for detection (p < 1E-324; Chi-squared).

Human Clusters

		1	2	3	4	5	Null
Human Communities	**1**	2784	5	0	56	22	153
	2	8	1278	361	23	32	310
	3	8	94	1202	7	12	197
	4	84	37	30	796	17	133
	5	83	14	11	28	536	105
	Null	2164	295	243	121	133	4571

Supp. Table 2.3. (Corresponds to Supplementary Table 4) Contingency table of Louvain communities and hierarchical clusters definitions in mouse. Each cell represents the number of lncRNAs that are found in both the corresponding row and column labels when groups of lncRNAs are defined using either the Louvain or hierarchical method. The large values along the diagonal indicate that the group definitions are stable with respect to the particular algorithm used for detection (p < 1E-324; Chi-squared).

Mouse Clusters

		1	2	3	4	5	Null
Mouse Communities	**1**	1555	6	41	0	4	203
	2	5	751	17	0	7	499
	3	88	4	156	0	3	209
	4	0	0	1	326	0	0
	5	0	1	2	0	42	31
	Null	204	191	630	0	167	3102

Supp. Table 2.4. (Corresponds to Supplementary Table 5) Summary statistics of human lncRNA communities. “Comm.”, community assignment; number of lncRNAs in each community is in parentheses. “N”, lncRNAs not assigned to a community at the specified threshold of similarity. “Length”, average length and (standard deviation). “GC”, average GC content and (standard deviation). “CpG”, proportion of lncRNAs that overlap CpG islands. “Proteins”, proportion of lncRNAs that overlap protein-coding genes. “Exons”, average number of exons in the lncRNA and (standard deviation).

Comm.	Length	GC	CpG	Proteins	Exons
1 (3023)	**1715** (3207)	**0.37** (0.04)	**0.08**	**0.41**	**2.28** (1.88)
2 (2021)	**1629** (2245)	**0.56** (0.04)	**0.27**	**0.52**	**2.64** (2.67)
3 (1529)	**1068** (879)	**0.58** (0.06)	**0.87**	**0.61**	**2.35** (1.76)
4 (1109)	**1469** (8177)	**0.47** (0.05)	**0.18**	**0.48**	**2.70** (1.65)
5 (789)	**1316** (1670)	**0.48** (0.05)	**0.20**	**0.55**	**2.13** (1.19)
N (7545)	**755** (710)	**0.46** (0.04)	**0.15**	**0.41**	**2.79** (2.50)

Supp. Table 2.5. (Corresponds to Supplementary Table 6) Summary statistics of mouse lncRNA communities. “Comm.”, community assignment; number of lncRNAs in each community is in parentheses. “N”, lncRNAs not assigned to a community at the specified threshold of similarity. “Length”, average length and (standard deviation). “GC”, average GC content and (standard deviation). “CpG”, proportion of lncRNAs that overlap CpG islands. “Proteins”, proportion of lncRNAs that overlap protein-coding genes. “Exons”, average number of exons in the lncRNA and (standard deviation).

Comm.	**Length**	**GC**	**CpG**	**Proteins**	**Exons**
1 (1824)	**2430** (3160)	**0.39** (0.03)	**0.07**	**0.57**	**1.88** (1.64)
2 (1288)	**1610** (1282)	**0.55** (0.05)	**0.62**	**0.67**	**2.72** (3.34)
3 (463)	**1475** (1080)	**0.46** (0.04)	**0.16**	**0.51**	**2.51** (1.50)
4 (327)	**1192** (422)	**0.41** (0.01)	**0.00**	**0.01**	**3.91** (0.33)
5 (76)	**1276** (1070)	**0.49** (0.04)	**0.03**	**0.18**	**2.43** (1.76)
N (4297)	**1048** (833)	**0.47** (0.04)	**0.14**	**0.42**	**2.75** (1.68)

Supp. Table 2.6. (Corresponds to Supplementary Table 9) Contingency table comparing 5mer based human communities to 6mer based communities. Each cell represents the number of lncRNAs that are found in both the corresponding row and column labels when community detection is run using either 5mers or 6mers as a similarity measure. The large values along the diagonal indicate that the community definitions are similar to one another (p < 1E-324; Chi-squared).

		5mer Human Communities					
		1	**2**	**3**	**4**	**5**	**Null**
6mers Human Communities	**1**	2835	1	0	2	3	179
	2	1	1785	28	8	8	182
	3	1	52	1343	1	0	123
	4	107	81	23	491	3	392
	5	57	67	24	371	8	250
	Null	226	133	34	13	84	7037

Supp. Table 2.7. (Corresponds to Supplementary Table 10) Contingency table comparing 7mer based human communities to 6mer based communities. Each cell represents the number of lncRNAs that are found in both the corresponding row and column labels when community detection is run using either 7mers or 6mers as a similarity measure. The large values along the diagonal indicate that the community definitions are similar to one another (p < 1E-324; Chi-squared).

		7mer Human Communities					
		1	**2**	**3**	**4**	**5**	**Null**
6mers Human Communities	**1**	1629	50	5	70	68	1198
	2	3	53	988	53	47	868
	3	0	11	1026	32	27	424
	4	5	0	11	888	38	155
	5	1	2	0	5	671	98
	Null	84	255	76	80	74	6958

Supp. Table 2.8. (Corresponds to Supplementary Table 13) Results of HSD tests between lncRNA localization of communities using polyA-selection in HepG2 cells. “Comm1” and "Comm2", community assignment for first and second set of lncRNAs compared, respectively. “n1” and “n2”, number of lncRNAs in Comm1 and Comm2, respectively. "meandiff", the mean difference in localization values between the two communities. "lower", the lower bound of the 95% confidence interval (CI) of the "meandiff" value. "upper", the upper bound of the 95% CI. "p < 0.05", result of HSD test indicating whether or not the means of "Comm1" and "Comm2" are significantly different. The test is significant if '0' is not contained within the CI.

Comm1	**Comm2**	**n1**	**n2**	**meandiff**	**lower**	**upper**	**p < 0.05**
1	2	1719	1397	0.0128	-0.0153	0.041	
1	3	1719	1180	-0.11	-0.1395	-0.0804	Yes
1	4	1719	775	-0.0214	-0.0552	0.0123	
1	5	1719	561	-0.0252	-0.0632	0.0127	
1	null	1719	685	-0.0476	-0.0829	-0.0124	Yes
2	3	1397	1180	-0.1228	-0.1537	-0.0919	Yes
2	4	1397	775	-0.0343	-0.0693	0.0007	
2	5	1397	561	-0.0381	-0.0771	0.0009	
2	null	1397	685	-0.0605	-0.0969	-0.0241	Yes
3	4	1180	775	0.0885	0.0524	0.1246	Yes
3	5	1180	561	0.0847	0.0447	0.1247	Yes
3	null	1180	685	0.0623	0.0248	0.0998	Yes
4	5	775	561	-0.0038	-0.0471	0.0395	
4	null	775	685	-0.0262	-0.0671	0.0147	
5	null	561	685	-0.0224	-0.0668	0.0221	

Supp. Table 2.9. (Corresponds to Supplementary Table 14) Results of HSD tests between lncRNA localization of communities using ribosome-depletion in HepG2 cells. “Comm1” and "Comm2", community assignment for first and second set of lncRNAs compared, respectively. “n1” and “n2”, number of lncRNAs in Comm1 and Comm2, respectively. "meandiff", the mean difference in localization values between the two communities. "lower", the lower bound of the 95% confidence interval (CI) of the "meandiff" value. "upper", the upper bound of the 95% CI. "p < 0.05", result of HSD test indicating whether or not the means of "Comm1" and "Comm2" are significantly different. The test is significant if '0' is not contained within the CI.

Comm1	Comm2	n1	n2	meandiff	lower	upper	p < 0.05
1	2	1864	1285	-0.0904	-0.1111	-0.0698	Yes
1	3	1864	1152	-0.1464	-0.1678	-0.125	Yes
1	4	1864	786	-0.0553	-0.0796	-0.031	Yes
1	5	1864	565	-0.0449	-0.0722	-0.0175	Yes
1	null	1864	740	-0.0156	-0.0404	0.0091	
2	3	1285	1152	-0.0559	-0.0791	-0.0328	Yes
2	4	1285	786	0.0351	0.0093	0.061	Yes
2	5	1285	565	0.0456	0.0168	0.0744	Yes
2	null	1285	740	0.0748	0.0485	0.1011	Yes
3	4	1152	786	0.0911	0.0647	0.1175	Yes
3	5	1152	565	0.1015	0.0722	0.1308	Yes
3	null	1152	740	0.1307	0.1039	0.1576	Yes
4	5	786	565	0.0104	-0.021	0.0419	
4	null	786	740	0.0397	0.0105	0.0689	Yes
5	null	565	740	0.0292	-0.0026	0.0611	

Supp. Table 2.10. (Corresponds to Supplementary Table 15) Results of HSD tests between lncRNA localization of communities using polyA-selection in K562cells. “Comm1” and "Comm2", community assignment for first and second set of lncRNAs compared, respectively. “n1” and “n2”, number of lncRNAs in Comm1 and Comm2, respectively. "meandiff", the mean difference in localization values between the two communities. "lower", the lower bound of the 95% confidence interval (CI) of the "meandiff" value. "upper", the upper bound of the 95% CI. "p < 0.05", result of HSD test indicating whether or not the means of "Comm1" and "Comm2" are significantly different. The test is significant if '0' is not contained within the CI.

Comm1	Comm2	n1	n2	meandiff	lower	upper	p < 0.05
1	2	1651	1289	-0.0344	-0.0625	-0.0062	Yes
1	3	1651	1125	-0.1571	-0.1864	-0.1278	Yes
1	4	1651	758	-0.051	-0.0842	-0.0178	Yes
1	5	1651	537	-0.0466	-0.0842	-0.0089	Yes
1	null	1651	659	-0.0423	-0.0772	-0.0074	Yes
2	3	1289	1125	-0.1227	-0.1536	-0.0918	Yes
2	4	1289	758	-0.0166	-0.0513	0.018	
2	5	1289	537	-0.0122	-0.0511	0.0267	
2	null	1289	659	-0.0079	-0.0442	0.0284	
3	4	1125	758	0.1061	0.0705	0.1417	Yes
3	5	1125	537	0.1105	0.0708	0.1502	Yes
3	null	1125	659	0.1148	0.0777	0.152	Yes
4	5	758	537	0.0044	-0.0383	0.0472	
4	null	758	659	0.0087	-0.0316	0.0491	
5	null	537	659	0.0043	-0.0397	0.0483	

Supp. Table 2.11. (Corresponds to Supplementary Table 16) Results of HSD tests between lncRNA localization of communities using ribosome-depletion in K562 cells. “Comm1” and "Comm2", community assignment for first and second set of lncRNAs compared, respectively. “n1” and “n2”, number of lncRNAs in Comm1 and Comm2, respectively. "meandiff", the mean difference in localization values between the two communities. "lower", the lower bound of the 95% confidence interval (CI) of the "meandiff" value. "upper", the upper bound of the 95% CI. "p < 0.05", result of HSD test indicating whether or not the means of "Comm1" and "Comm2" are significantly different. The test is significant if '0' is not contained within the CI.

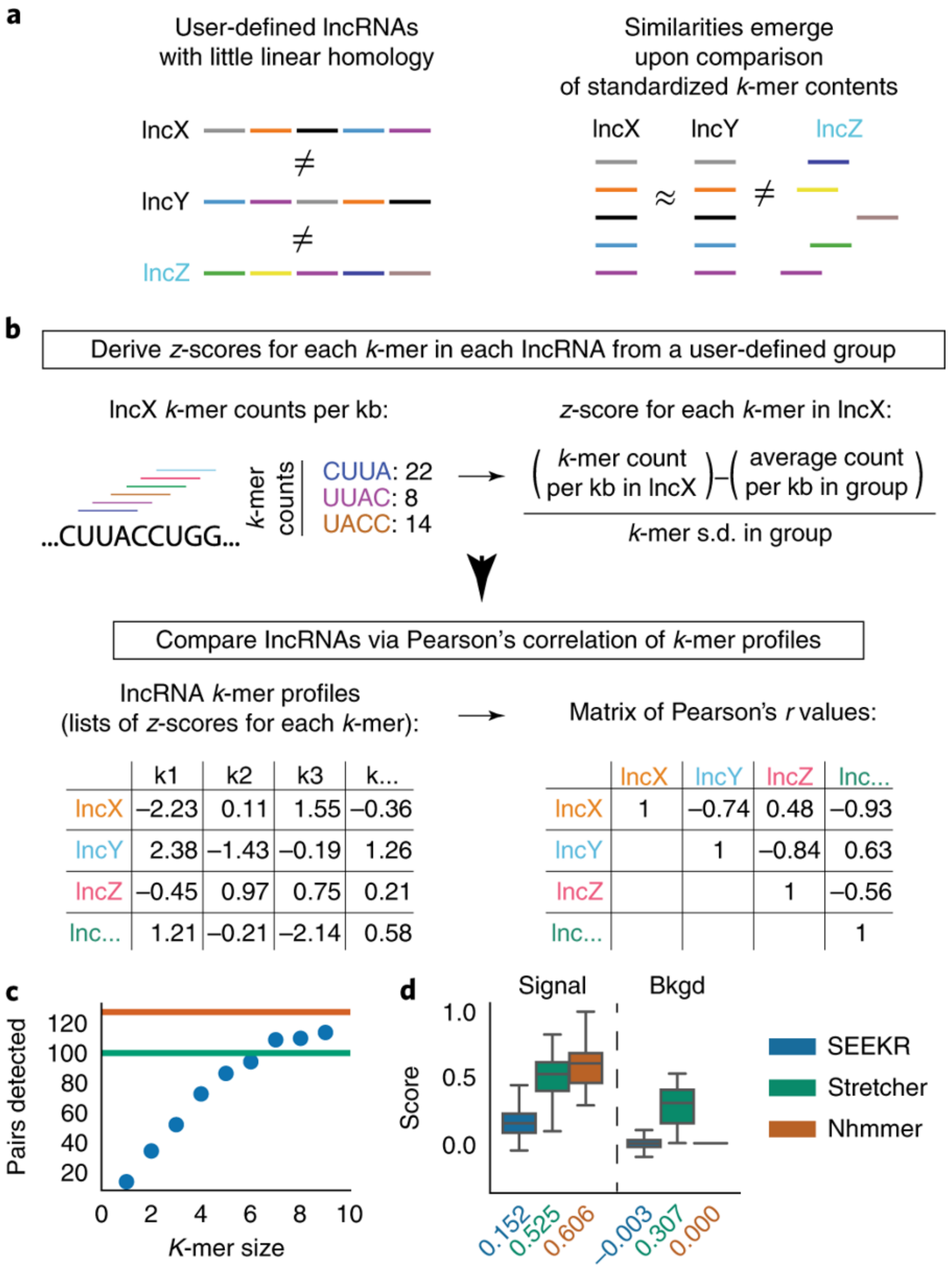
Comm1	Comm2	n1	n2	meandiff	lower	upper	p < 0.05
1	2	1636	1170	-0.1335	-0.1607	-0.1063	Yes
1	3	1636	1086	-0.1345	-0.1623	-0.1067	Yes
1	4	1636	703	-0.0929	-0.1249	-0.0608	Yes
1	5	1636	510	-0.0962	-0.1323	-0.0602	Yes
1	null	1636	621	-0.0297	-0.0632	0.0038	
2	3	1170	1086	-0.001	-0.031	0.0289	
2	4	1170	703	0.0406	0.0067	0.0745	Yes
2	5	1170	510	0.0373	-0.0004	0.075	
2	null	1170	621	0.1038	0.0685	0.1391	Yes
3	4	1086	703	0.0416	0.0072	0.076	Yes
3	5	1086	510	0.0383	0.0001	0.0764	Yes
3	null	1086	621	0.1048	0.069	0.1406	Yes
4	5	703	510	-0.0033	-0.0447	0.038	
4	null	703	621	0.0632	0.024	0.1023	Yes
5	null	510	621	0.0665	0.024	0.109	Yes

Supp. Table 2.12. (Corresponds to Supplementary Table 17) Distributions of polysome associated lncRNAs between communities. “Community”, the name of the community. “Observed”, the number of literature reported lncRNAs associated with polysomes, in a given community. “Expected”, the number of lncRNAs that would be associated with polysomes if the lncRNAs were randomly distributed between the communities. “Ratio” Observed divided by Expected. Polysomal lncRNAs are not uniformly distributed across communities (p = 3.5e-5, Chi-squared); they are most enriched in community 3 and most depleted in community 1, providing additional support for the hypothesis that kmer content provides information about lncRNA cellular localization.

Community	Observed	Expected	Ratio
1	24	51	0.47
2	32	36	0.89
3	52	39	1.33
4	25	21	1.19
5	11	17	0.65
Null	85	65	1.31

Supp. Table 2.13. (Corresponds to Supplementary Table 21) Protein binding motif counts across lncRNAs expressed in HepG2 or K562 cells. “Protein”, the name of the RNA binding protein. “True Positives”, motifs identified by FIMO that were experimentally validated by eCLIP data. “Total”, all motifs identified by FIMO. "TP%", the True Positive Rate is the number of True Positive regions divided by the Total number of regions. "(0.01)" indicates that FIMO was run at a threshold of 0.01. "(0.0001)" indicates that FIMO was run at a threshold of 0.0001. "% Diff." is the percent difference between "TP% (0.01)" and TP% (0.0001)". The True Positive percentage is low for both the 0.01 and 0.0001 threshold, and there is little to no difference between the percentages at each threshold. The 0.01 threshold was chosen for our analysis performed as part of Fig. 3D since the number of True Positive samples were multiple orders of magnitude more numerous at that threshold.

Proteins	**True Positive (0.01)**	**Total (0.01)**	**TP% (0.01)**	**True Positive (0.0001)**	**Total (0.0001)**	**TP% (0.0001)**	**% Diff.**
FXR1	399	23669	1.7	7	323	2.2	-0.5
FXR2	278	20539	1.4	12	484	2.5	-1.1
HNRNPA1	5486	64872	8.5	52	424	12.3	-3.8
HNRNPC	3474	38076	9.1	211	2611	8.1	1.0
hnRNPK	1204	22075	5.5	92	1355	6.8	-1.3
IGF2BP1	1311	35404	3.7	25	504	5.0	-1.3
IGF2BP2	340	22963	1.5	9	906	1.0	0.5
IGF2BP3	525	23434	2.2	20	906	2.2	0.0
KHDRBS1	1125	19565	5.8	54	895	6.0	-0.3
NONO	1046	26786	3.9	27	1065	2.5	1.4
PCBP2	1017	20455	5.0	188	3008	6.3	-1.3
PTBP1	2339	66007	3.5	102	2262	4.5	-1.0
QKI	1780	44428	4.0	63	498	12.7	-8.6
SFPQ	1119	40233	2.8	2	233	0.9	1.9
SRSF1	19781	238759	8.3	1072	12118	8.8	-0.6
SRSF9	3005	76606	3.9	131	2934	4.5	-0.5
TIA1	4008	77512	5.2	211	3877	5.4	-0.3


a
User-defined lncRNAs with little linear homology
lncX
≠
lncY
≠
lncZ
Similarities emerge upon comparison of standardized k-mer contents
lncX
≈
lncY
≠
lncZ
b
Derive z-scores for each k-mer in each lncRNA from a user-defined group
lncX k-mer counts per kb:
...CUUACCUGG...
k-mer counts
CUUA: 22
UUAC: 8
UACC: 14
z-score for each k-mer in lncX:
(k-mer count per kb in lncX)–(average count per kb in group)
k-mer s.d. in group
Compare lncRNAs via Pearson's correlation of k-mer profiles
lncRNA k-mer profiles (lists of z-scores for each k-mer):
Matrix of Pearson's r values:
c
Pairs detected
K-mer size
0
2
4
6
8
10
20
40
60
80
100
120
d
Signal
Bkgd
Score
0.0
0.5
1.0
SEEKR
Stretcher
Nhmmer
0.152
0.525
0.606
–0.003
0.307
0.000

	k1	k2	k3	k...
lncX	–2.23	0.11	1.55	–0.36
lncY	2.38	–1.43	–0.19	1.26
lncZ	–0.45	0.97	0.75	0.21
lnc...	1.21	–0.21	–2.14	0.58

	lncX	lncY	lncZ	lnc...
lncX	1	–0.74	0.48	–0.93
lncY		1	–0.84	0.63
lncZ			1	–0.56
lnc...				1

Figure 2.1. Overview and initial test of kmer-based sequence comparison. (A) LncRNAs of related function (names in black) may harbor similar sequence similarity in the form of motif content (colored bars) even if they lack linear homology. **(B)** In SEEKR, the abundance of all kmers of length k are counted by tiling across each lncRNA in a user-defined group in one nucleotide increments. Kmer counts are normalized for lncRNA length, and standardized across the group to derive z-scores. Similarity is evaluated by comparing lncRNA kmer profiles (lists of z-scores for each kmer in the lncRNAs) with Pearson's correlation. **(C)** Number of homologous pairs detected by SEEKR vs. kmer length in a test set of conserved lncRNAs. Green and orange lines mark the homologue number detected by Stretcher and nhmmer, respectively. **(D)** Signal to background ratios for homologue detection via the three methods. Tukey boxplots show the lower, median, and upper quartile of values, and ±1.5x the IQR (n=161 r values for signal, n=12880 r values for background); outliers are not shown.

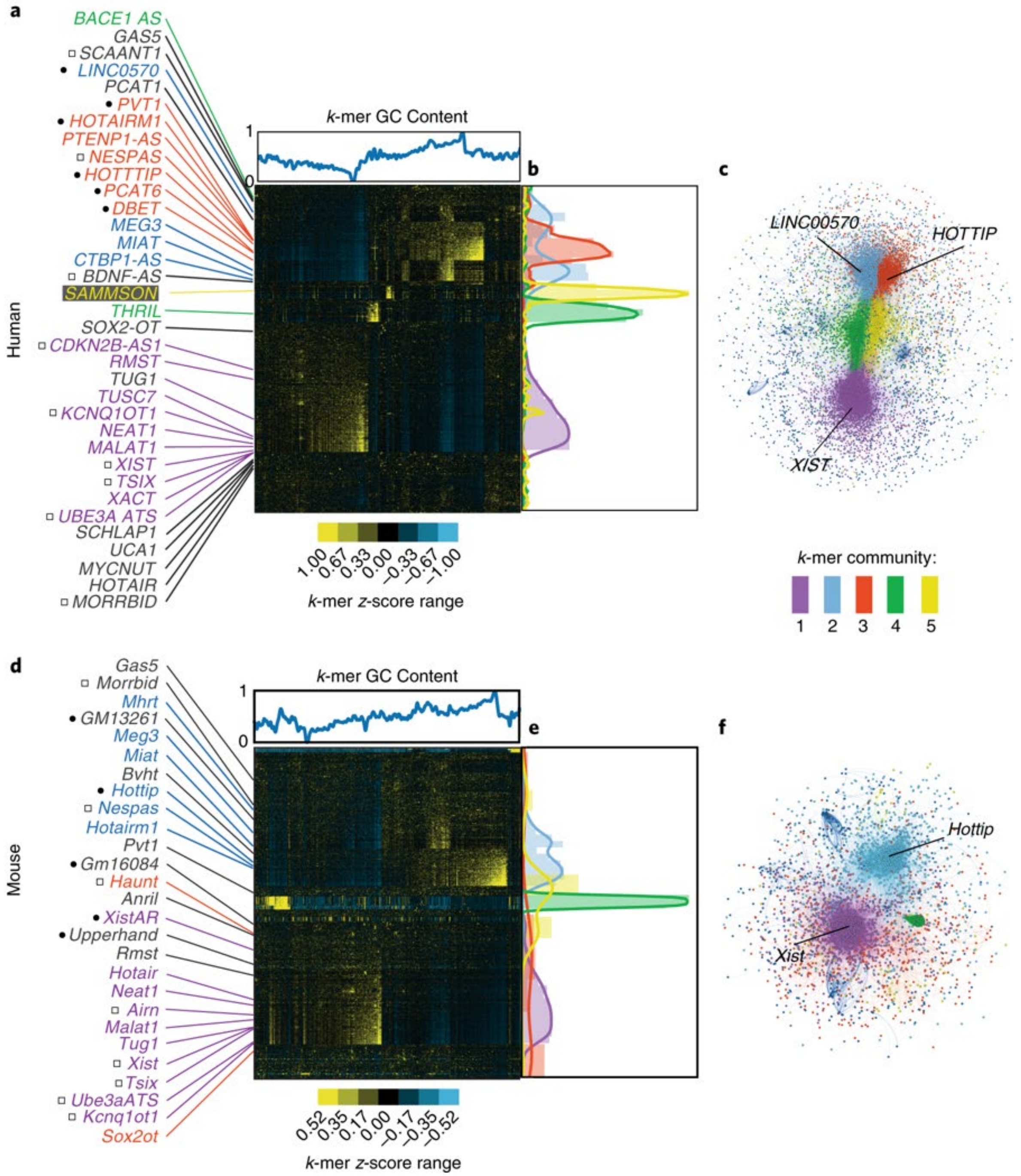
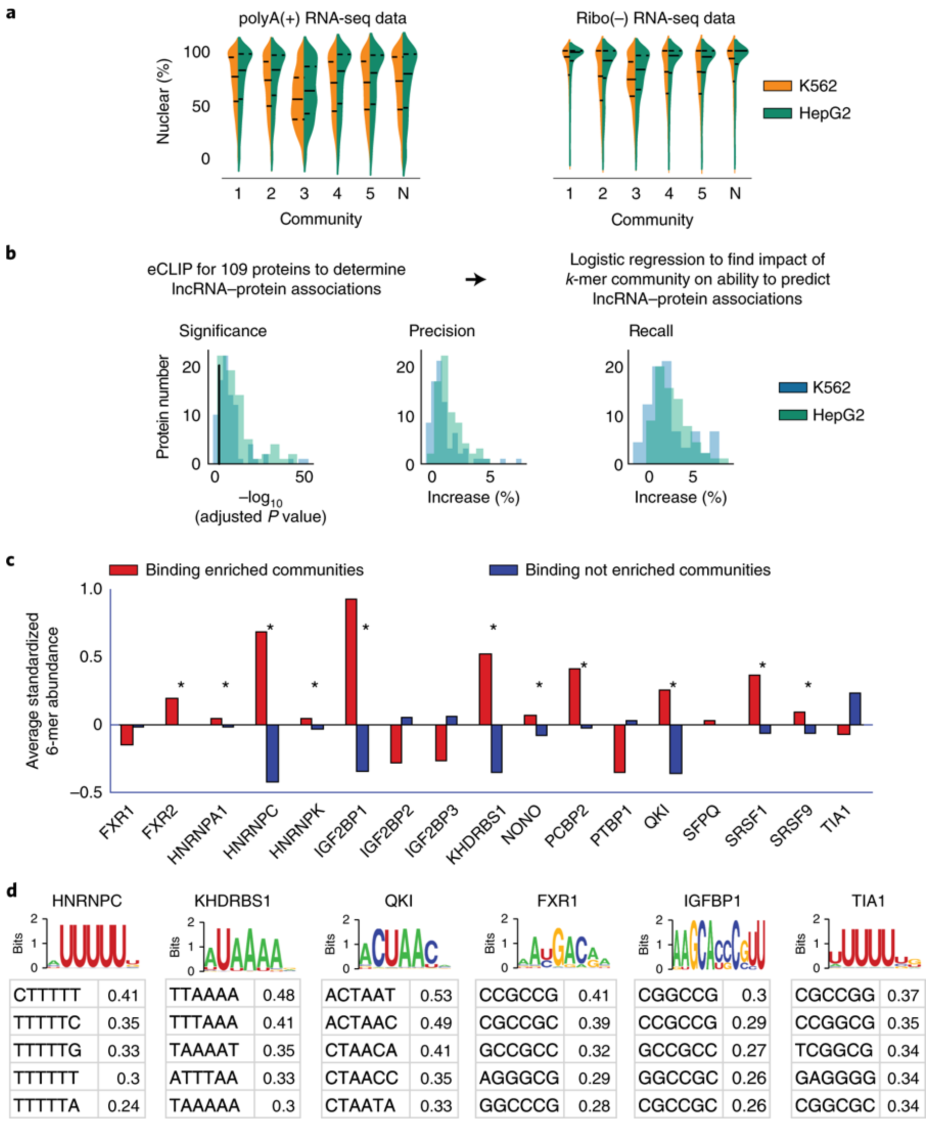


Figure 2.2. LncRNAs of related function often have related kmer contents. (A) Hierarchical cluster of all human GENCODE lncRNAs at kmer length 6, with lncRNAs and kmers on the x- and y-axes, respectively. Kmer z-scores (relative kmer abundance) range from blue (lowest) to yellow (highest). GC content of kmers is shown above the x-axis. Locations of select lncRNAs are marked. Left of lncRNA names, black circles indicate cis activators and squares indicate cis repressors. **(B)** Locations of lncRNAs assigned to communities 1 through 5 via the Louvain/network-based approach. **(C)**Network graph of Louvain-assigned lncRNA communities. LncRNA names in (A) are colored by their Louvain community assignment; lncRNAs in gray were assigned to the null. **(C, D, E)** Same as (A, B, C) but for mouse GENCODE lncRNAs.



HNRNPC		KHDRBS1		QKI		FXR1		IGFBP1		TIA1	
CTTTTT	0.41	TTAAAA	0.48	ACTAAT	0.53	CCGCCG	0.41	CGGCCG	0.3	CGCCGG	0.37
TTTTTC	0.35	TTTAAA	0.41	ACTAAC	0.49	CGCCGC	0.39	CCGCCG	0.29	CCGGCG	0.35
TTTTTG	0.33	TAAAAT	0.35	CTAACA	0.41	GCCGCC	0.32	GCCGCC	0.27	TCGGCG	0.34
TTTTTT	0.3	ATTTAA	0.33	CTAACC	0.35	AGGGCG	0.29	GGCCGC	0.26	GAGGGG	0.34
TTTTTA	0.24	TAAAAA	0.3	CTAATA	0.33	GGCCCG	0.28	CGCCGC	0.26	CGGCGC	0.34

Figure 2.3. LncRNA localization and protein binding correlate with kmer content. (A) Violin plots of lncRNA localization by kmer community in K562 (blue) and HepG2 (green) cells, as determined from RNA-Seq of polyA-selected and ribosome-depleted RNA. "N", the "null" community. Lines show the lower, median, and upper quartile of values (see Supplemental Figs. 13-16 for samples sizes)**. (B)** From left to right; Log10 significance of increase in likelihood (i), % increase in precision (ii), and % increase in recall (iii) obtained when lncRNA community information is included in a logistic regression to predict protein association. Black line in (i) corresponds to a log10(adjusted p-value) of 0.05 (n=3747 lncRNAs for HepG2, n=3278 lncRNAs for K562). **(C)** 11 of the 17 proteins with experimentally determined PWMs from [23] show significantly increased abundance of motif-matching kmers (n=4096) in lncRNA communities that are enriched for binding to the protein in question (p<0.01; permutation test; marked by *'s). **(D)** The most enriched kmers in 300 nucleotide windows surrounding motif matches in CLIP peaks do not always match the motif. PWMs from [23] are shown above average z-scores for the top 5 most enriched kmers in true positive relative to false positive binding regions for the protein in question. PWMs and top kmers are shown for all 17 proteins in Supplementary Fig. 5.

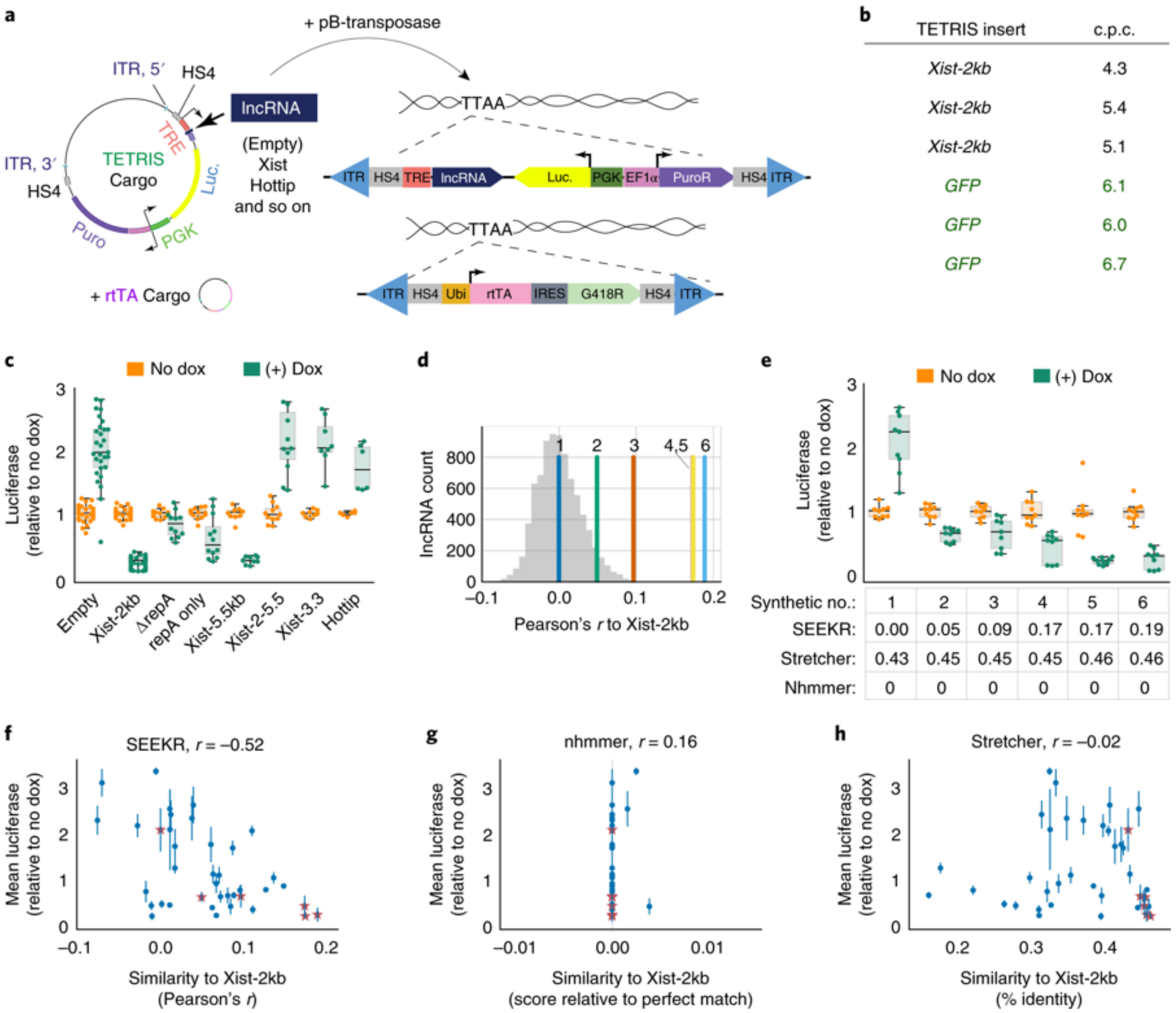


Figure 2.4. Kmer content correlates with lncRNA repressive activity. (A) Overview of vectors and concept of the TETRIS assay. **(B)** Number of TETRIS-lncRNA-cargo insertions per cell ("c.p.c.") after 10-day drug selection for two separate cargos, *Xist-2kb* and *GFP*. Each row represents copy number data from independent replicates. **(C)** Luciferase values for different TETRIS-lncRNA constructs relative to No Dox. Tukey boxplots as in Fig. 1D. Data are from at least six independent luciferase assays from at least two biological replicate derivations of TETRIS cell lines. Exact numbers of assays and replicates performed for each TETRIS lncRNA cargo are found in Supplementary Table 22. **(D)** Pearson's r similarity of kmer profiles for the six synthetic lncRNAs relative to the first 2kb of *Xist*. Histogram of similarity of *Xist*-2kb to all other GENCODE M5 lncRNAs is shown in gray. **(E)** Effect of synthetic lncRNA expression on luciferase activity. Tukey boxplots as in Fig. 1D. SEEKR, Stretcher, and nhmmer similarity for each synthetic lncRNA relative to the first 2kb of *Xist* is shown below the graph. **(F, G, H)** Pearson's correlation between repressive activity and similarities to *Xist*-2kb as defined by SEEKR, nhmmer, and Stretcher for thirty-three endogenous lncRNAs/lncRNA fragments (dots) and six synthetic lncRNAs (stars) (mean ± standard deviation). See Supplementary Table 22 for sample sizes in panels C, E, F, G, H.

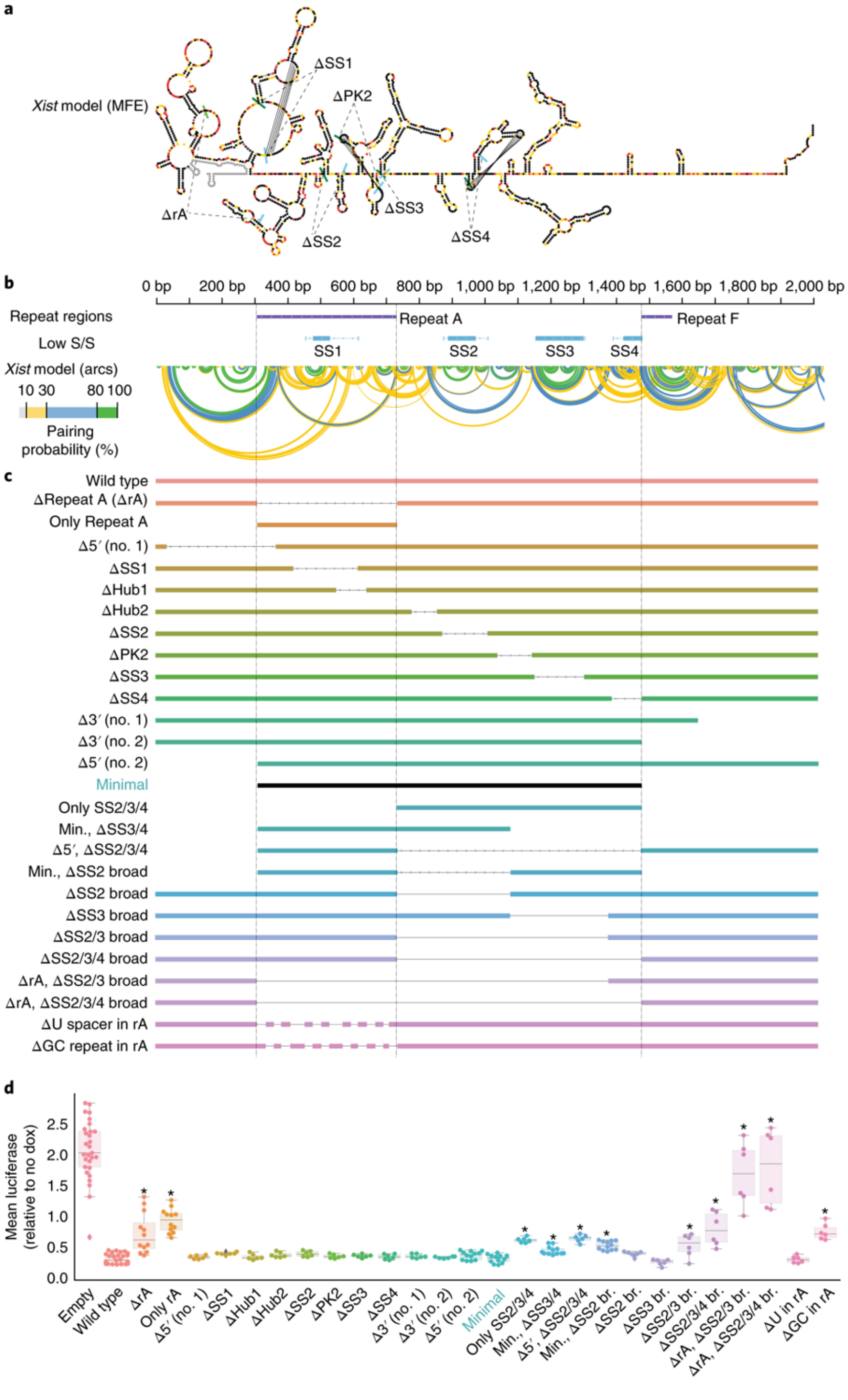
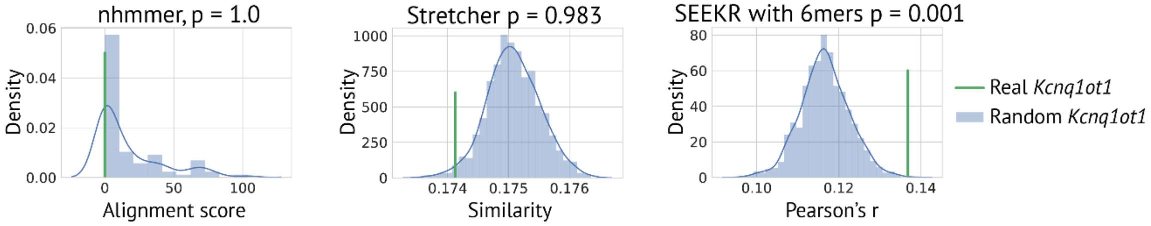
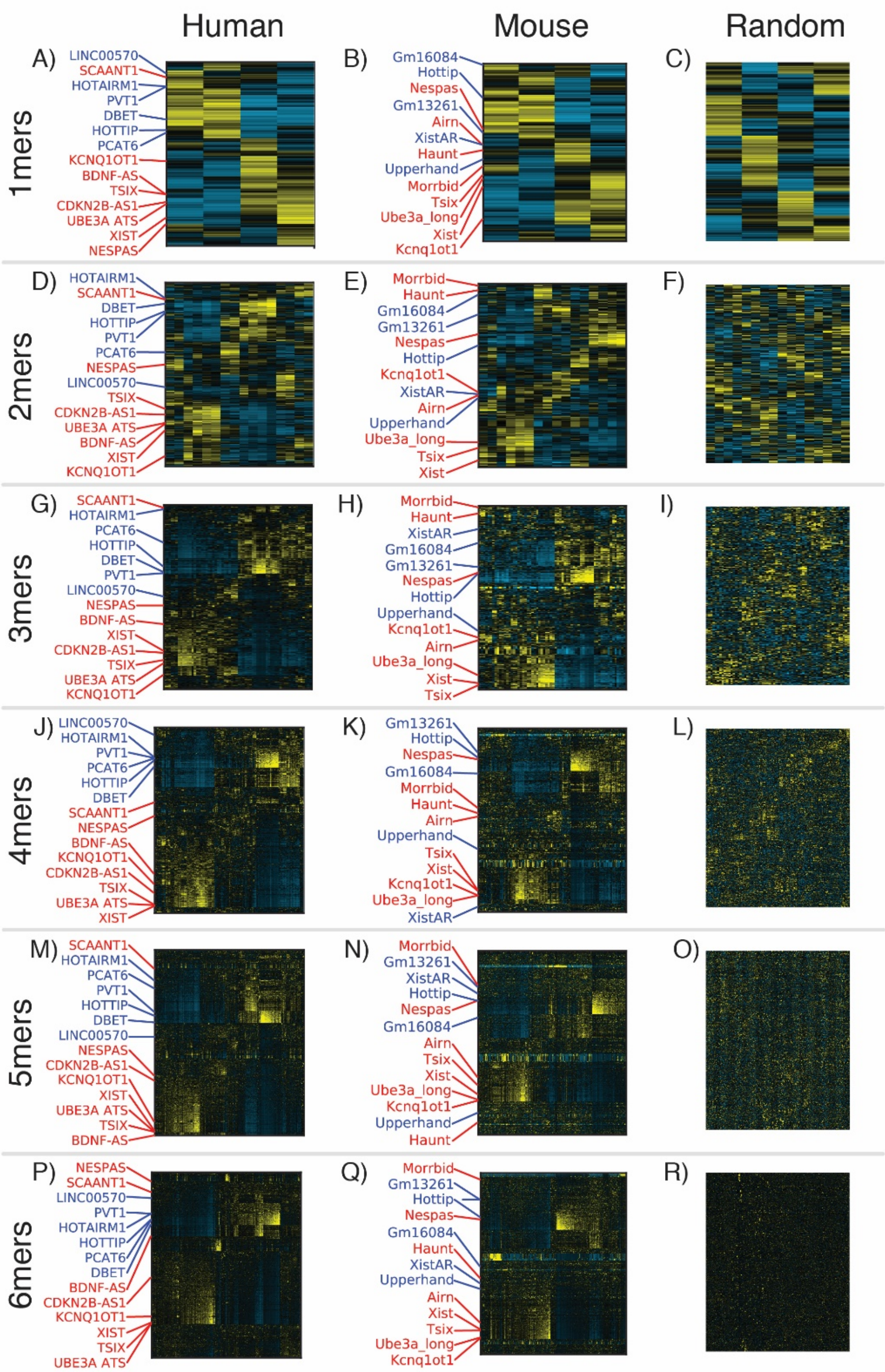


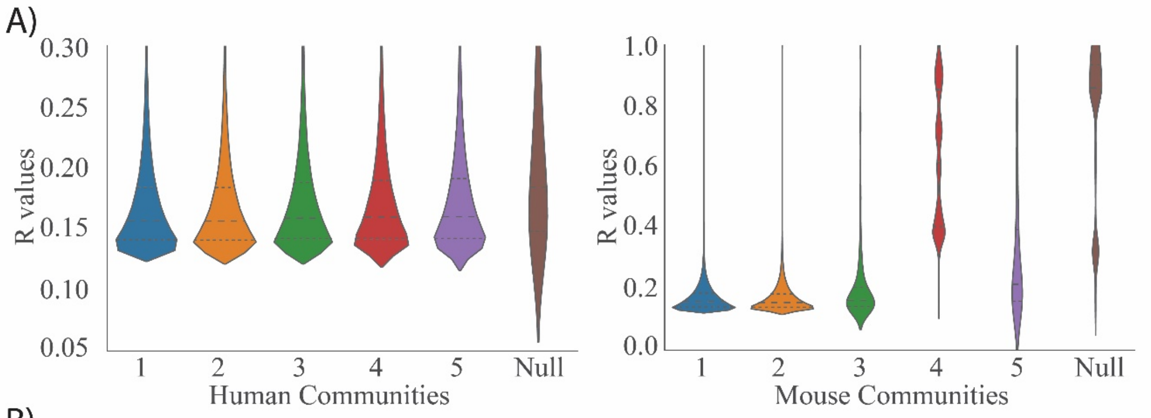
Figure 2.5. Mapping of elements required for repression by *Xist*-2kb in TETRIS. (A) Minimum Free Energy (MFE) and **(B)** arc-based structural models of the first 2kb of *Xist* from [41]; green and blue bars in (i) mark starts and stops of indicated regions; locations of *Xist* repeats [7] and predicted stable structures (low S/S, regions of low SHAPE reactivity and Shannon entropy from [41]) are also shown in (ii). **(C)** Deleted regions. **(D)** Effects on luciferase after dox addition. *, Bonferroni corrected p<0.001 relative to Wild-type/*Xist-2kb* via Student's t-test. Tukey boxplots show the lower, median, and upper quartile of values, and ±1.5x the IQR (see Supplementary Table 22 for sample sizes and exact p-values).



Supplementary Fig. 2.1. Comparison of *Xist* to *Kcnq1ot1* via nhmmer, Stretcher, and SEEKR, relative to 1,000 randomly generated lncRNAs of length/mononucleotide content identical/similar to *Kcnq1ot1*. Only SEEKR is able to detect a significant level of similarity between *Xist* and *Kcnq1ot1*.



Supplementary Fig. 2.2. Hierarchical clusters of human and mouse GENCODE lncRNAs, and sequences randomly generated using the nucleotide composition of the human set, at varying kmer lengths. Axes and label colors are the same as in Fig. 2. Locations of cis-repressing and cis-activating lncRNAs are marked in red and blue, respectively.



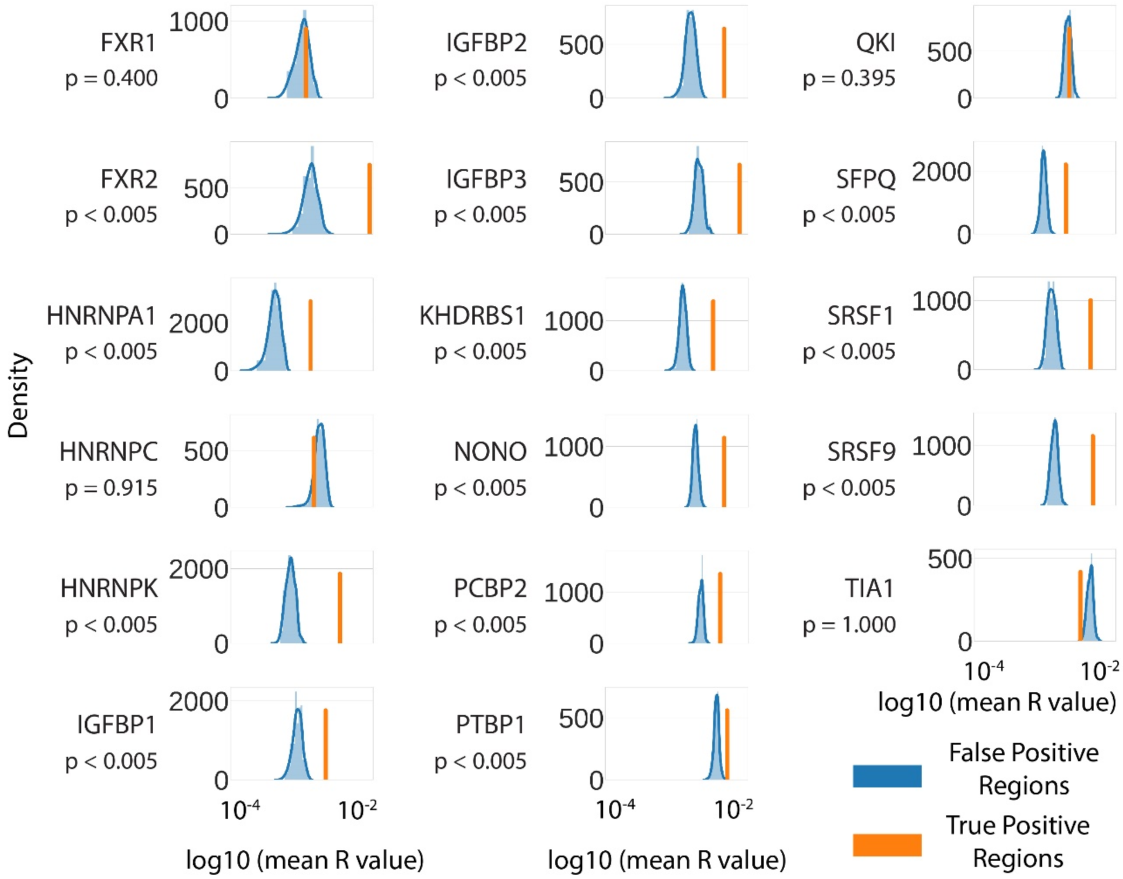
B)

Com.	Count	Mean	Std	Min	25%	50%	75%	Max
1	294126	0.169	0.041	0.130	0.141	0.156	0.184	1
2	108281	0.170	0.044	0.130	0.141	0.156	0.184	1
3	120144	0.172	0.045	0.130	0.142	0.159	0.188	1
4	36378	0.173	0.046	0.130	0.142	0.159	0.189	1
5	23268	0.175	0.053	0.130	0.142	0.160	0.191	1
Null	3841	0.251	0.190	0.130	0.148	0.184	0.252	1

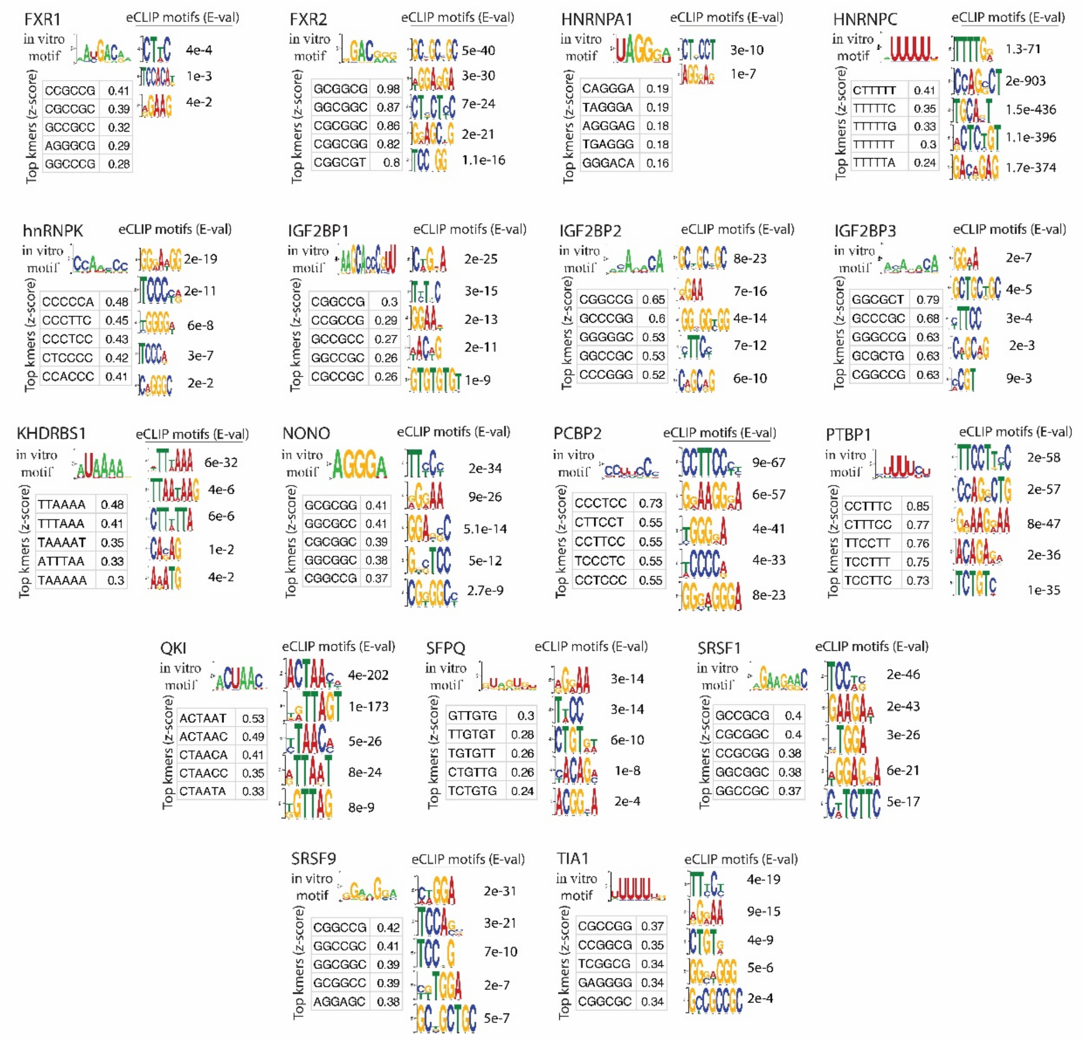
C)

Com.	Count	Mean	Std	Min	25%	50%	75%	Max
1	179939	0.171	0.041	0.130	0.142	0.160	0.188	1
2	41759	0.171	0.051	0.130	0.141	0.157	0.185	1
3	1251	0.206	0.128	0.130	0.143	0.164	0.207	1
4	52939	0.626	0.217	0.154	0.405	0.615	0.854	1
5	240	0.323	0.238	0.131	0.161	0.217	0.395	1
Null	9844	0.760	0.254	0.130	0.637	0.859	0.935	1

Supplementary Fig. 2.3. Relationships between lncRNAs in human and mouse communities. **(A)** Violin plots of the distribution of Pearson’s r values for the similarities between lncRNAs in each community. Lines show the lower, median, and upper quartile of values (see “Count” column of tables for the sample size). **(B)** Summary statistics of Pearson’s r values between human lncRNAs in each community. “Comm.”, community assignment. “Count”, number of edges (i.e. comparisons between pairs of lncRNAs) in community. “Mean”, average Pearson’s r value of edges. "Std", standard deviations. "Min", smallest Pearson’s r value. "25%", Pearson’s r value of the 25th percentile. "50%", Pearson’s r value of the 50th percentile. "75%", Pearson’s r value of the 75th percentile. "Max", largest Pearson’s r value. **(C)** Summary statistics of Pearson’s r values between mouse lncRNAs in each community. Column labels are the same as in (B).



Supplementary Fig. 2.4. The distributions of average pairwise similarities of kmer profiles from random and size matched sets of false positive binding regions compared to the average pairwise similarity of the experimentally confirmed true positive regions for each protein (n=2000 regions, p-value determined by unadjusted permutation test). Because the average pairwise similarities of true positive regions are consistently an order of magnitude or more above those for the false positive regions, the x-axes are plotted on a log scale. For 13 of 17 proteins, the true positive regions are more similar to each other than any randomly generated set of false positive regions.



Top kmers (z-score)	
CCGCCG	0.41
CGCCGC	0.39
GCCGCC	0.32
AGGGCG	0.29
GGCCCG	0.28

Top kmers (z-score)	
GCGGCG	0.98
GGCGGC	0.87
CGCGGC	0.86
CGGCGG	0.82
CGGCGT	0.8

Top kmers (z-score)	
CAGGGA	0.19
TAGGGA	0.19
AGGGAG	0.18
TGAGGG	0.18
GGGACA	0.16

Top kmers (z-score)	
CTTTTT	0.41
TTTTTC	0.35
TTTTTG	0.33
TTTTTT	0.3
TTTTTA	0.24

Top kmers (z-score)	
CCCCCA	0.48
CCCTTC	0.45
CCCTCC	0.43
CTCCCC	0.42
CCACCC	0.41

Top kmers (z-score)	
CGGCCG	0.3
CCGCCG	0.29
GCCGCC	0.27
GGCCGC	0.26
CGCCGC	0.26

Top kmers (z-score)	
CGGCCG	0.65
GCCCGG	0.6
GGGGGC	0.53
GGCCGC	0.53
CCCGGG	0.52

Top kmers (z-score)	
GGCGCT	0.79
GCCCGC	0.68
GGGCCG	0.63
GCGCTG	0.63
CGGCCG	0.63

Top kmers (z-score)	
TTAAAA	0.48
TTTAAA	0.41
TAAAAT	0.35
ATTTAA	0.33
TAAAAA	0.3

Top kmers (z-score)	
GCGCGG	0.41
GGCGCC	0.41
CGCGGC	0.39
GGCGGC	0.38
CGGCCG	0.37

Top kmers (z-score)	
CCCTCC	0.73
CTTCCT	0.55
CCTTCC	0.55
TCCCTC	0.55
CCTCCC	0.55

Top kmers (z-score)	
CCTTTC	0.85
CTTTCC	0.77
TTCCTT	0.76
TCCTTT	0.75
TCCTTC	0.73

Top kmers (z-score)	
ACTAAT	0.53
ACTAAC	0.49
CTAACA	0.41
CTAACC	0.35
CTAATA	0.33

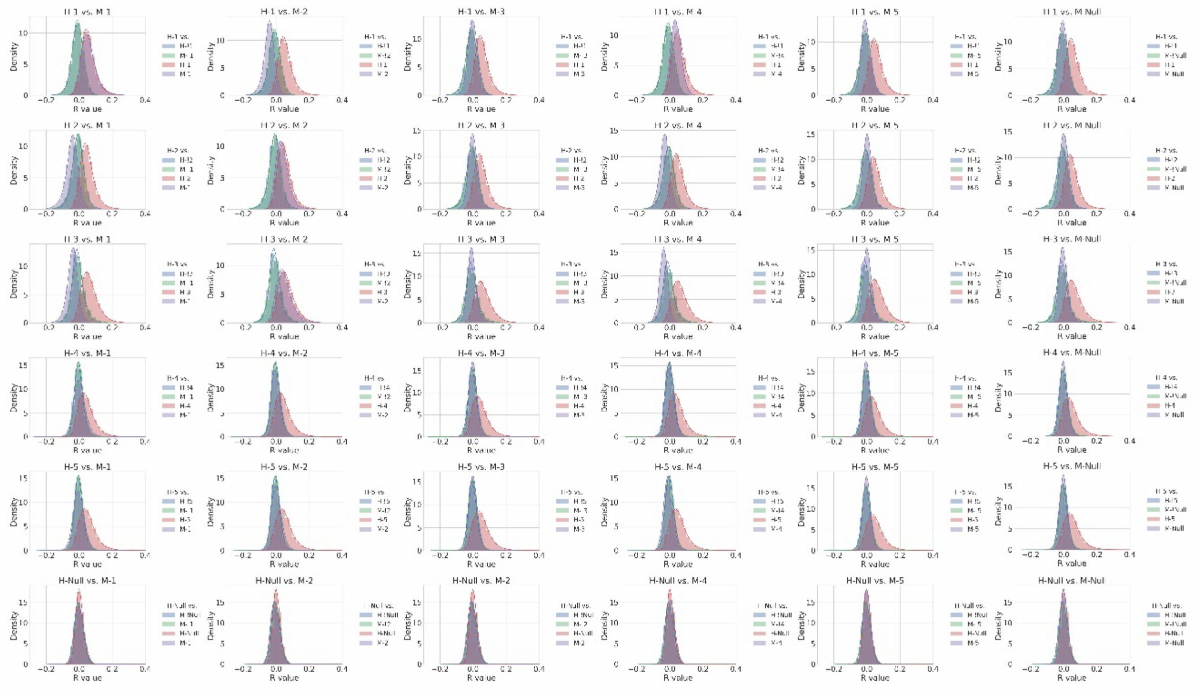
Top kmers (z-score)	
GTTGTG	0.3
TTGTGT	0.28
TGTGTT	0.26
CTGTTG	0.26
TCTGTG	0.24

Top kmers (z-score)	
GCCGCG	0.4
CGCGGC	0.4
CCGCGG	0.38
GGCGGC	0.38
GGCCGC	0.37

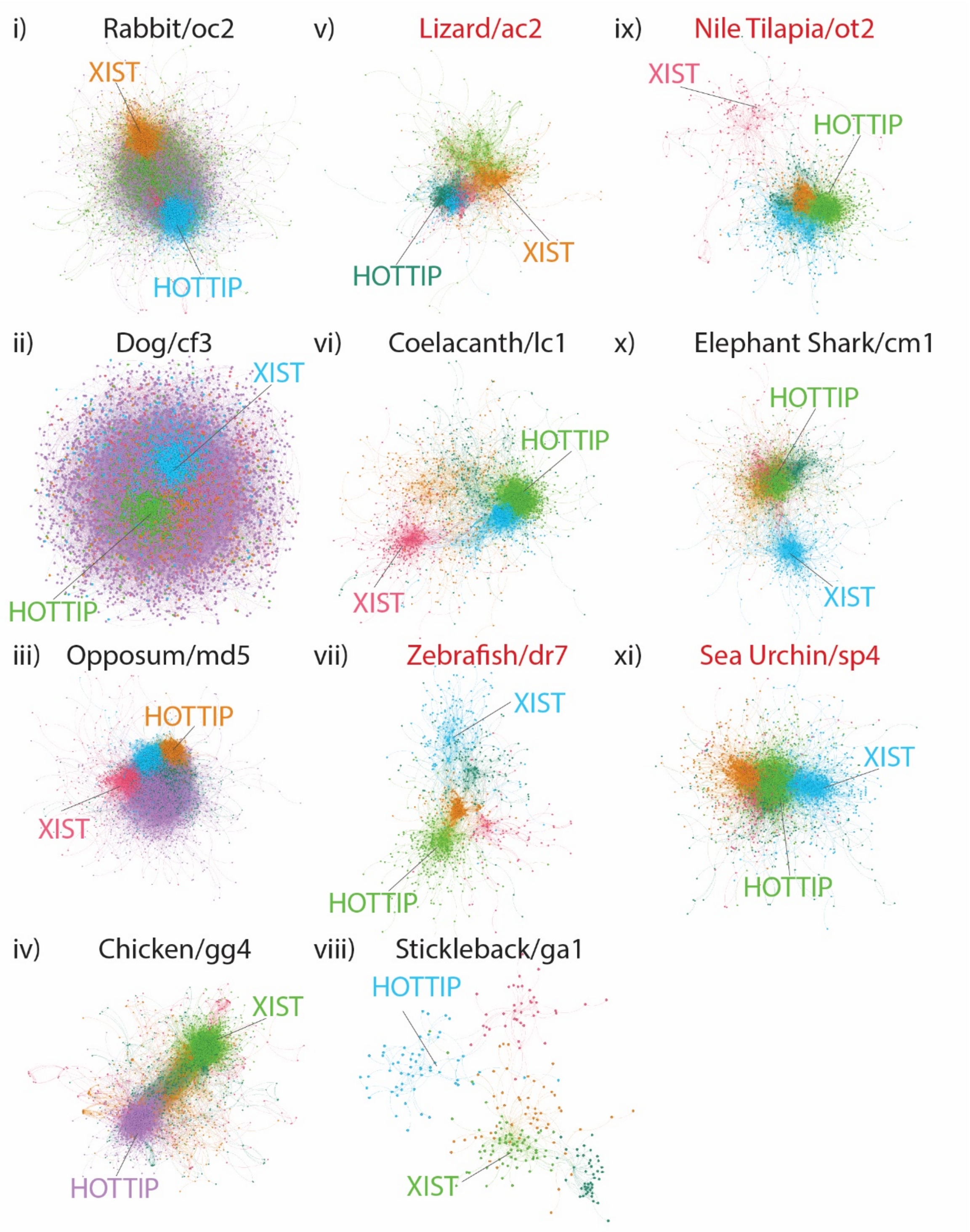
Top kmers (z-score)	
CGGCCG	0.42
GGCCGC	0.41
GGCGGC	0.39
GCGGCC	0.39
AGGAGC	0.38

Top kmers (z-score)	
CGCCGG	0.37
CCGGCG	0.35
TCGGCG	0.34
GAGGGG	0.34
CGGCGC	0.34

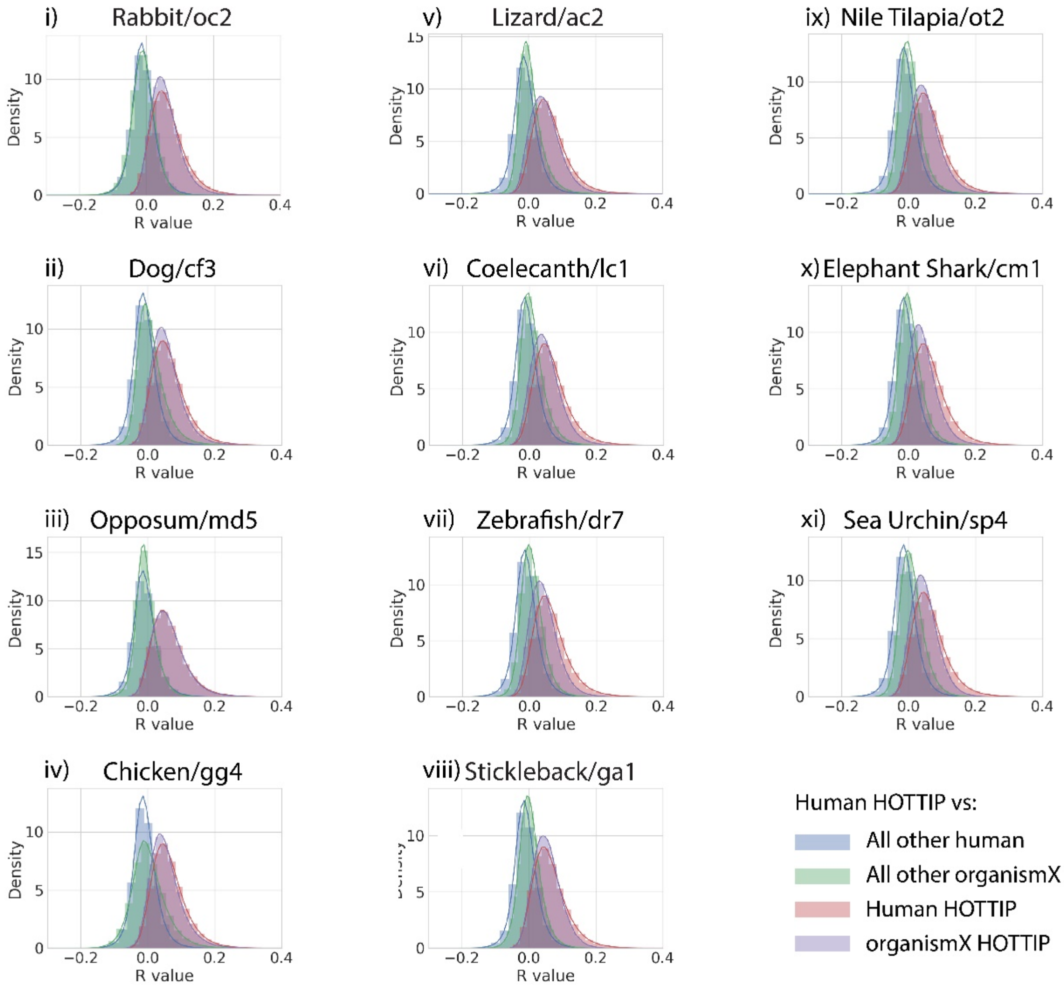
Supplementary Fig. 2.5. Biochemically measured PWMs (from (*1*); "*in vitro motifs*") for the 17 proteins with CLIP data in HepG2 and K562 cells (from (*2*)) are shown above the five kmers that were the most enriched in true positive regions (motif matches falling inside of CLIP peaks) relative to false positive regions (motif matches falling outside of CLIP peaks) for each protein in question. The z-scores associated with kmer enrichment in the true positive regions are also shown. Adjacent to that information are the top 5 motifs identified from eCLIP peaks using DREME (*3*). Only 3 and 2 motifs were identified by DREME from FXR1 and HNRNPA1 eCLIP data, respectively. For HNRNPC, the first motif listed ranked outside of the top 5 (ranked 11th by E-value), but it is shown because it is the eCLIP-derived motif that best matched the *in vitro*-derived motif. For all other proteins, the eCLIP-derived motif that in our evaluation best matched the *in vitro*-derived motif fell in the top 5. By our evaluation, *in vitro*-derived motifs, top eCLIP-derived motifs, and top enriched 6mers from SEEKR showed some level of concordance for 11 of 17 proteins (FXR2, HNRNPA1, HNRNPC, HNRNPK, KHDRBS1, NONO, PCBP2, PTBP1, QKI, SFPQ, and SRSF9), and the *in vitro*-derived motifs and top eCLIP-derived motif showed concordance for an additional protein (TIA1). For the remaining five proteins, the *in vitro*-derived motifs, top eCLIP-derived motifs, and top enriched 6mers from SEEKR showed substantial differences.



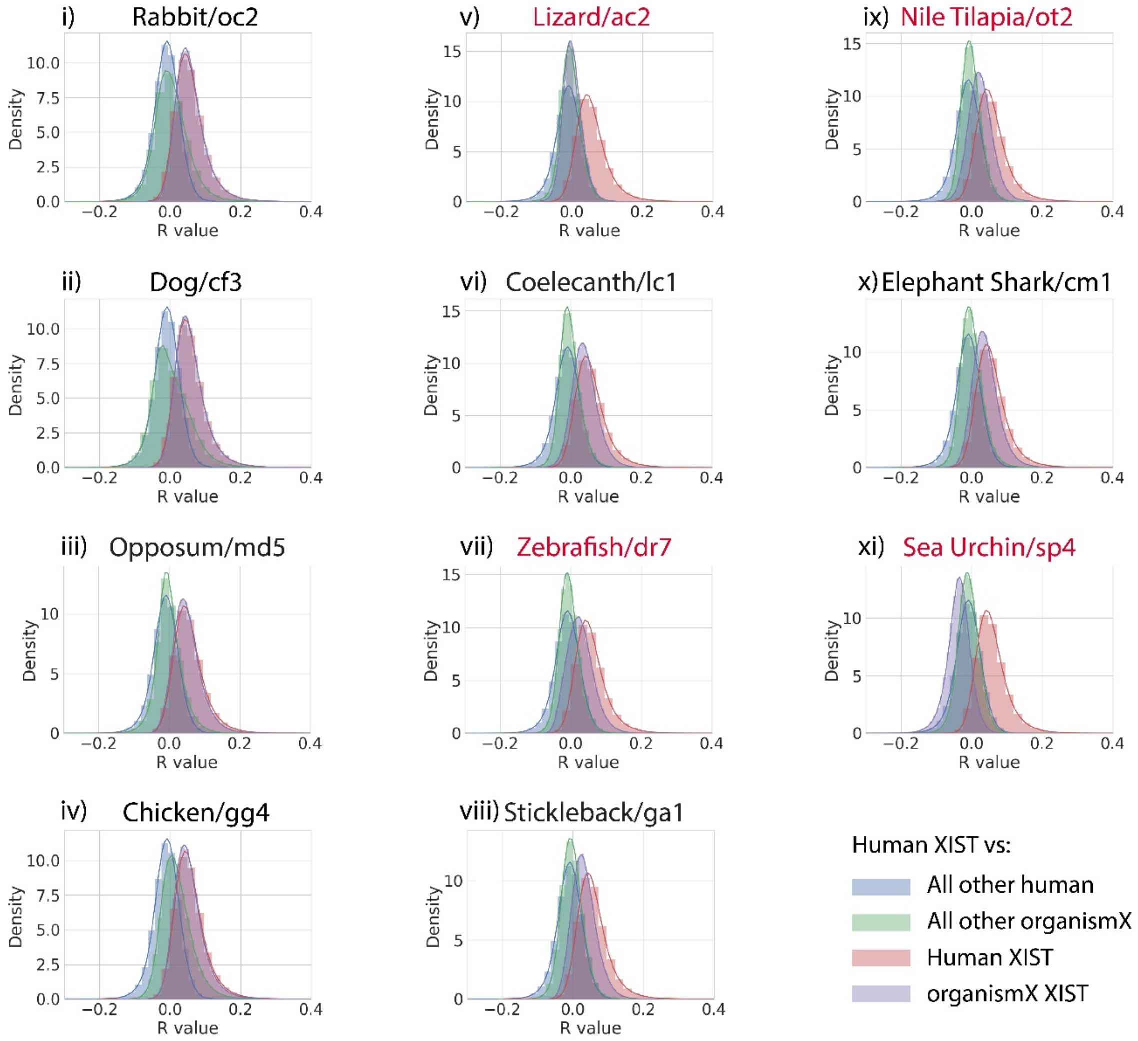
Supplementary Fig. 2.6. Similarity between human and mouse lncRNA communities. . “H-#” refers to human lncRNAs and their corresponding community number (1, 2, etc.). “H-!#” refers all human lncRNAs excepting the community number shown. “M-#” and “M-!#”, same as for human but with mouse lncRNAs. Significant similarity was observed between communities H-1 and M-1, H-1 and M-4, H-2 and M-2, and H-3 and M-2 (note the clear overlap of red and purple histograms).



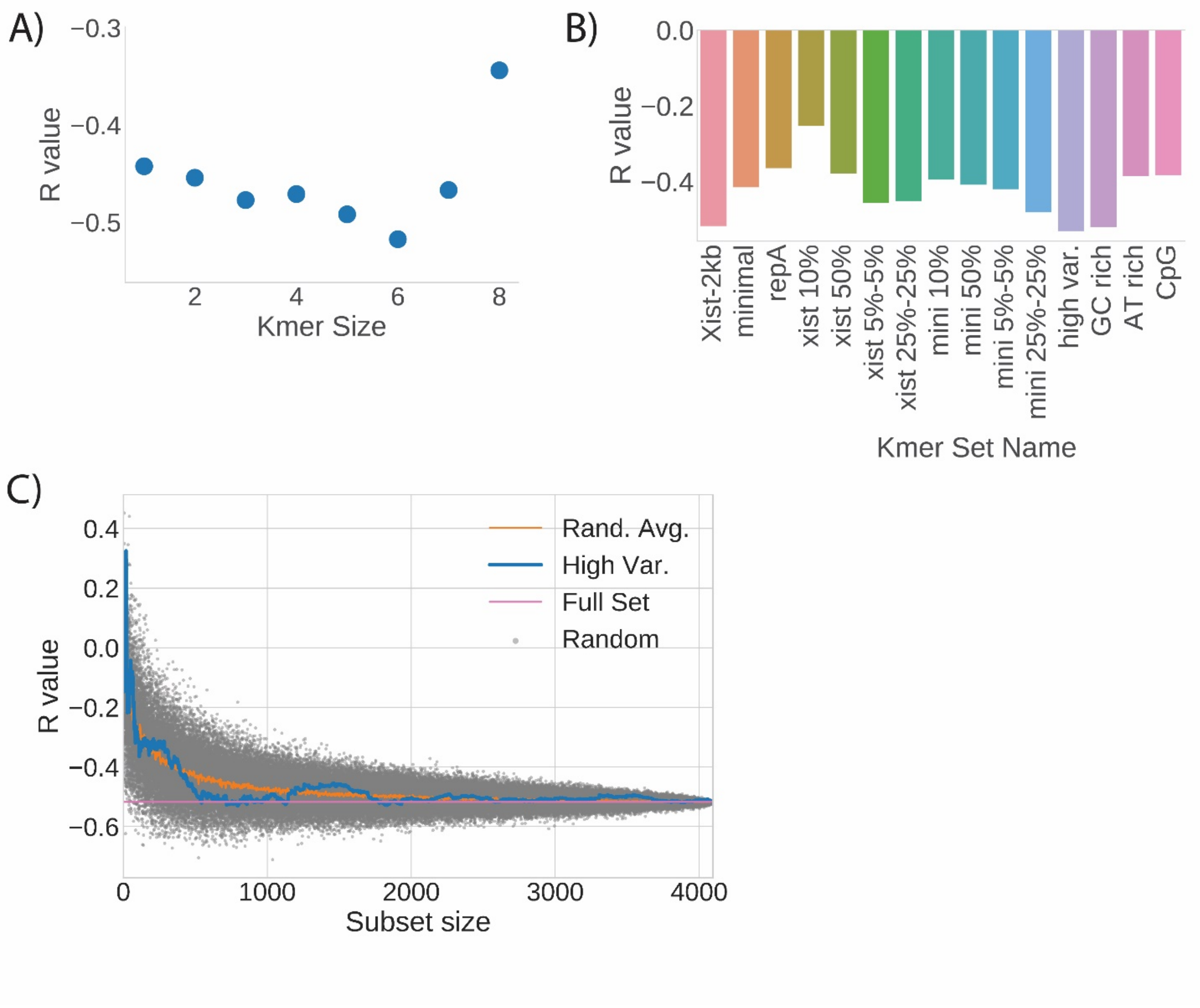
Supplementary Fig. 2.7. Louvain defined communities in other organisms. Human XIST and HOTTIP have been added to each set of lncRNAs.



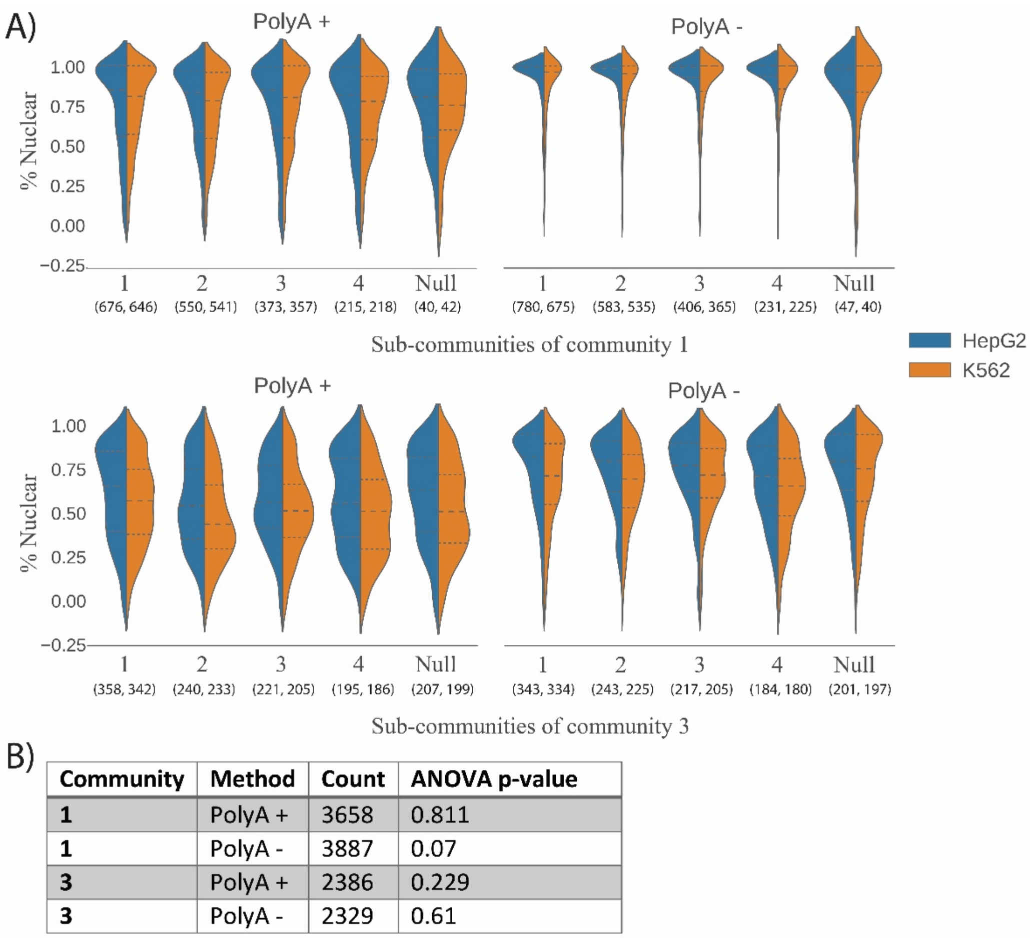
Supplementary Fig. 2.8. Similarity between the human *HOTTIP* community (community #3 in Fig. 2A) and cognate lncRNA communities in other organisms. A *HOTTIP*-like community was found all organisms examined (red and purple histograms show significant overlap).



Supplementary Fig. 2.9. Similarity between the human *XIST* community (community #1 in Fig. 2A) and cognate lncRNA communities in other organisms. An *XIST*-like community was found in seven of the ten vertebrate species examined (names in black; red and purple histograms show significant overlap).



Supplementary Fig. 2.10. Additional correlations with TETRIS data show that the full set of 4096 6mers is the kmer set that is most likely to provide the greatest predictive value in SEEKR. **(A)** The correlation between *Xist*-likeness and repressive ability in the TETRIS assay is reported (y-axis) for kmer sizes 1 through 8 when running SEEKR (x-axis). 6mers provide the best correlation (-0.52). **(B)** Subsets of 6mers were selected in an attempt to improve the correlation between *Xist*-likeness and repressive ability. “*Xist*-2kb” contains the full set of 4096 6mers, which represents the Pearson’s r value (-0.52) on which other 6mer sets could improve. “minimal” also uses the full set of 6mers, but measures each lncRNA inserted into TETRIS for its similarity to the minimal repressive fragment found in Fig. 5. Similarly, “repA” uses the full set of 6mers, but measures lncRNAs for their similarity to the repeat A region of Xist (Fig. 5). All 6mer sets to the right of the “repA” bar are subsets of 6mers that used the Xist-2kb transcript to calculate correlations between lncRNAs and TETRIS data. “*xist* 10%” is the set of 410 6mers are the most overabundant in *Xist*-2kb relative to all other mouse lncRNAs. Likewise, “*xist* 50%” is the set of 2048 6mers that have the largest z-score in *Xist*-2kb. “*xist* 5%-5%” contains the 210 6mers with the largest z-scores, plus the 210 6mers with the lowest z-scores. These low abundance z-scores were added to ensure that not all 6mers in the subset were correlated with each other. “xist 25%-25%” contains the 1024 6mers with the largest z-scores, plus another 1024 6mers with the lowest z-scores. “mini 10%”, “mini 50%”, “mini 5%-5%”, and “mini 25%-25%”, are the same 6mer subsets as their “xist” counterparts, except that the 6mers are the most over- and under-represented in the minimal fragment of *Xist*, instead of the *Xist*-2kb fragment. “high var.” 6mers are the 800 6mers with the highest standard deviations across the six communities defined in Fig. 3. “GC rich” and “AT rich” are 6mer subsets that contain at least four “GC” nucleotides or “AT” nucleotides, respectively. Each set contains 1408 kmers. “CpG” contains the 1185 6mers that contain a “CG” dinucleotide in their sequence. No rationally designed subsets of 6mers were significantly more predictive of lncRNA repressive activity than the baseline “Xist-2kb” fragment. **(C)** 100,000 subsets of randomly generated kmers, across the full range of subset sizes, are plotted relative to their Pearson’s r values for our TETRIS data (grey circles). The average Pearson’s r value at each subset size was calculated (orange line). The kmers with the largest standard deviation across lncRNA communities are also plotted for each kmer subset size (blue line). At no kmer subset size were either the average random sets or the most highly variable kmers significantly more predictive of TETRIS data than the full set of 6mers (pink line).



Community	Method	Count	ANOVA p-value
1	PolyA +	3658	0.811
1	PolyA -	3887	0.07
3	PolyA +	2386	0.229
3	PolyA -	2329	0.61

Supplementary Fig. 2.11. Lack of significant differences in subcellular localization in sub-communities of human community 1 and 3 lncRNAs. To determine if sub-communities of lncRNAs harbor significantly different biological properties within the five major lncRNA communities in human, lncRNAs from community #1 and #3 were extracted from the set of human GENCODE lncRNAs, and the Louvain algorithm run at default resolution parameter was used to identify the five most likely sub-communities within each. Because communities #1 and #3 were the most nuclear and cytoplasmic communities respectively, we examined if their respective sub-communities harbored significant differences in subcellular localization. **(A)** Violin plots of the distributions of cellular localization ratios for lncRNAs in communities 1 and 3. Lines show the lower, median, and upper quartile of values**.** The sample size of each distribution is indicated below the distribution, indicating the number of lncRNAs in the distribution for HepG2 and K562, respectively. **(B)** Results of ANOVA tests examining if the nuclear distributions amongst sub-communities was different. “Community”, the original community from which sub communities were created. “Method”, the RNA-seq method used to generate the ENCODE dataset. “Count”, number of lncRNAs in each data set and the sample size used to calculate the p-value. “ANOVA p-value”, results of the ANOVA test, where p-values < .05 indicate a significant difference between distributions. Unlike that observed for the major lncRNA communities in Fig. 3A, no significant differences in subcellular localization between sub-communities were detected.

REFERENCES

1 Iyer, M. K. *et al.* The landscape of long noncoding RNAs in the human transcriptome. *Nat Genet* **47**, 199-208 (2015).

2 Geisler, S. & Coller, J. RNA in unexpected places: long non-coding RNA functions in diverse cellular contexts. *Nat Rev Mol Cell Biol* **14**, 699-712 (2013).

3 Holoch, D. & Moazed, D. RNA-mediated epigenetic regulation of gene expression. *Nat Rev Genet* **16**, 71-84 (2015).

4 Liu, X., Hao, L., Li, D., Zhu, L. & Hu, S. Long non-coding RNAs and their biological roles in plants. *Genomics Proteomics Bioinformatics* **13**, 137-147 (2015).

5 Rinn, J. L. & Chang, H. Y. Genome regulation by long noncoding RNAs. *Annu Rev Biochem* **81**, 145-166 (2012).

6 Gutschner, T. & Diederichs, S. The hallmarks of cancer: a long non-coding RNA point of view. *RNA Biol* **9**, 703-719 (2012).

7 Lee, J. T. & Bartolomei, M. S. X-inactivation, imprinting, and long noncoding RNAs in health and disease. *Cell* **152**, 1308-1323 (2013).

8 Wu, X. & Sharp, P. A. Divergent transcription: a driving force for new gene origination? *Cell* **155**, 990-996 (2013).

9 Cech, T. R. & Steitz, J. A. The noncoding RNA revolution-trashing old rules to forge new ones. *Cell* **157**, 77-94 (2014).

10 Hezroni, H. *et al.* Principles of long noncoding RNA evolution derived from direct comparison of transcriptomes in 17 species. *Cell Rep* **11**, 1110-1122 (2015).

11 Cabili, M. N. *et al.* Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. *Genes Dev* **25**, 1915-1927 (2011).

12 Derrien, T. *et al.* The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. *Genome Res* **22**, 1775-1789 (2012).

13 Bateman, A. *et al.* UniProt: a hub for protein information. *Nucleic Acids Research* **43**, D204-D212 (2015).

14 Berman, H., Henrick, K. & Nakamura, H. Announcing the worldwide Protein Data Bank. *Nat Struct Biol* **10**, 980 (2003).

15 Ulitsky, I. & Bartel, D. P. lincRNAs: genomics, evolution, and mechanisms. *Cell* **154**, 26-46 (2013).

16 Kutter, C. *et al.* Rapid turnover of long noncoding RNAs and the evolution of gene expression. *PLoS Genet* **8**, e1002841 (2012).

17 Necsulea, A. *et al.* The evolution of lncRNA repertoires and expression patterns in tetrapods. *Nature* **505**, 635-+ (2014).

18 Eddy, S. R. Computational analysis of conserved RNA secondary structure in transcriptomes and genomes. *Annu Rev Biophys* **43**, 433-456 (2014).

19 Quinn, J. J. *et al.* Rapid evolutionary turnover underlies conserved lncRNA-genome interactions. *Genes Dev* **30**, 191-207 (2016).

20 Eddy, S. R. Homology searches for structural RNAs: from proof of principle to practical use. *RNA* **21**, 605-607 (2015).

21 Wheeler, T. J. & Eddy, S. R. nhmmer: DNA homology search with profile HMMs. *Bioinformatics* **29**, 2487-2489 (2013).

22 Rice, P., Longden, I. & Bleasby, A. EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet* **16**, 276-277 (2000).

23 Ray, D. *et al.* A compendium of RNA-binding motifs for decoding gene regulation. *Nature* **499**, 172-177 (2013).

24 Stefl, R., Skrisovska, L. & Allain, F. H. RNA sequence- and shape-dependent recognition by proteins in the ribonucleoprotein particle. *EMBO Rep* **6**, 33-38 (2005).

25 Edgar, R. C. & Batzoglou, S. Multiple sequence alignment. *Curr Opin Struc Biol* **16**, 368-373 (2006).

26 Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J Mol Biol* **215**, 403-410 (1990).

27 Pervouchine, D. D. *et al.* Enhanced transcriptome maps from multiple mouse tissues reveal evolutionary constraint in gene expression. *Nature communications* **6**, 5903 (2015).

28 Chadwick, B. P. Variation in Xi chromatin organization and correlation of the H3K27me3 chromatin territories to transcribed sequences by microarray analysis. *Chromosoma* **116**, 147-157 (2007).

29 Engreitz, J. M. *et al.* RNA-RNA interactions enable specific targeting of noncoding RNAs to nascent Pre-mRNAs and chromatin sites. *Cell* **159**, 188-199 (2014).

30 Mak, W. *et al.* Mitotically stable association of polycomb group proteins eed and enx1 with the inactive x chromosome in trophoblast stem cells. *Curr Biol* **12**, 1016-1020 (2002).

31 West, J. A. *et al.* The long noncoding RNAs NEAT1 and MALAT1 bind active chromatin sites. *Mol Cell* **55**, 791-802 (2014).

32 Clemson, C. M., McNeil, J. A., Willard, H. F. & Lawrence, J. B. XIST RNA paints the inactive X chromosome at interphase: evidence for a novel RNA involved in nuclear/chromosome structure. *J Cell Biol* **132**, 259-275 (1996).

33 Calabrese, J. M. *et al.* Site-specific silencing of regulatory elements as a mechanism of X inactivation. *Cell* **151**, 951-963 (2012).

34 Blondel, V. D., Guillaume, J. L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J Stat Mech-Theory E* (2008).

35 Dunham, I. *et al.* An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**, 57-74 (2012).

36 Carlevaro-Fita, J., Rahim, A., Guigo, R., Vardy, L. A. & Johnson, R. Cytoplasmic long noncoding RNAs are frequently bound to and degraded at ribosomes in human cells. *RNA* **22**, 867-882 (2016).

37 Van Nostrand, E. L. *et al.* Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). *Nat Methods* **13**, 508-514 (2016).

38 Hawkins, D. M. The problem of overfitting. *J Chem Inf Comput Sci* **44**, 1-12 (2004).

39 Spitale, R. C. *et al.* Structural imprints in vivo decode RNA regulatory mechanisms. *Nature* **519**, 486-+ (2015).

40 Lambert, N. *et al.* RNA Bind-n-Seq: quantitative assessment of the sequence and structural binding specificity of RNA binding proteins. *Mol Cell* **54**, 887-900 (2014).

41 Smola, M. J. *et al.* SHAPE reveals transcript-wide interactions, complex structural domains, and protein interactions across the Xist lncRNA in living cells. *Proc Natl Acad Sci U S A* **113**, 10322-10327 (2016).

42 Di Matteo, M. *et al.* PiggyBac toolbox. *Methods Mol Biol* **859**, 241-254 (2012).

43 Ding, S. *et al.* Efficient transposition of the piggyBac (PB) transposon in mammalian cells and mice. *Cell* **122**, 473-483 (2005).

44 Dowen, J. M. *et al.* Control of cell identity genes occurs in insulated neighborhoods in mammalian chromosomes. *Cell* **159**, 374-387 (2014).

45 Wutz, A., Rasmussen, T. P. & Jaenisch, R. Chromosomal silencing and localization are mediated by different domains of Xist RNA. *Nat Genet* **30**, 167-174 (2002).

46 Liu, F., Somarowthu, S. & Pyle, A. M. Visualizing the secondary and tertiary architectural domains of lncRNA RepA. *Nat Chem Biol* **13**, 282-289 (2017).

47 Tyner, C. *et al.* The UCSC Genome Browser database: 2017 update. *Nucleic Acids Res* **45**, D626-D634 (2017).

48 Team, R. C. *R: A language and environment for statistical computing.*, <https://www.R-project.org/> (2017).

49 Saldanha, A. J. Java Treeview--extensible visualization of microarray data. *Bioinformatics* **20**, 3246-3248 (2004).

50 Weir, W. H., Emmons, S., Gibson, R., Taylor, D. & Mucha, P. J. Post-Processing Partitions to Identify Domains of Modularity Optimization. *Algorithms* **10** (2017).

51 Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15-21 (2013).

52 Liao, Y., Smyth, G. K. & Shi, W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* **30**, 923-930 (2014).

53 Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J Mach Learn Res* **12**, 2825-2830 (2011).

54 Bailey, T. L. *et al.* MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Res* **37**, W202-208 (2009).

55 Machanick, P. & Bailey, T. L. MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics* **27**, 1696-1697 (2011).

56 Darty, K., Denise, A. & Ponty, Y. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics* **25**, 1974-1975 (2009).

57 Busan, S. & Weeks, K. M. Visualization of RNA structure models within the Integrative Genomics Viewer. *RNA* **23**, 1012-1018 (2017).

CHAPTER 3

SEEKR: a tool for classifying RNAs by kmer content

3.1 Introduction

Upwards of 80% of the human genome can be transcribed into RNA. Of the total number of transcribed nucleotides, approximately one half comprise pre-messenger RNAs (pre-mRNAs) that will ultimately become spliced and encode for proteins in the cytoplasm. The other half comprise long noncoding RNAs (lncRNAs), defined as RNA species that are greater than 200 nucleotides in length and have little or no potential to encode for proteins. Compared to transcripts produced from protein-coding genes, lncRNAs are, on average, less conserved, transcribed at lower levels, spliced less efficiently, and more likely to remain in the nucleus (1-6).

Nevertheless, a growing number of lncRNAs have been studied experimentally, and are now known to play important roles in health and development. Some of the most notable of these include the lncRNA *XIST*, which orchestrates transcriptional silencing during X-chromosome Inactivation (7), the lncRNAs *NEAT1* and *MALAT1*, which play roles in nuclear organization and have context-dependent functions in development and in cancer (8-14), and the lncRNA *NORAD*, which helps to maintain genome stability by promoting DNA repair (15,16). LncRNAs have also been found to play important roles in developmental transitions (17-23), in the immune system (24-26), in the brain (27-33) , and in the heart (34-37). These identified roles, coupled with the large number of lncRNAs that have yet to be studied experimentally, suggest that lncRNAs with important physiological functions remain to be discovered.

Still, identifying function in lncRNAs remains a major challenge. Many lncRNAs are thought to function as hubs that concentrate proteins, DNA, and possibly other biomolecules in particular regions of the cell, yet the sequence characteristics that give rise to these functions and the mechanisms through which they occur are poorly defined, even for the best studied lncRNAs (38-42). Moreover, relative to protein-coding genes, lncRNAs are poorly conserved, evolve rapidly, and are prone to changes in gene architecture, limiting the extent to which traditional phylogenetic analyses can be employed to identify the sequence features that are important for specifying their function (43). As an example, placental mammals express the *XIST* lncRNA to orchestrate gene silencing during X-Chromosome Inactivation (7), while marsupial mammals independently evolved their own lncRNA to orchestrate X-Chromosome Inactivation, termed *Rsx*. Remarkably, *XIST* and *Rsx* share no significant similarity by standard methods of sequence alignment (44,45). Thus, even though *Rsx* and *XIST* presumably function through analogous mechanisms, standard tools of sequence comparison are unable to detect the analogy. This problem extends to all lncRNAs. The sequence patterns that specify recurring functions in lncRNAs are largely unknown and difficult to detect computationally. Thus, to date, lncRNA functions must be determined empirically, on a case-by-case basis.

Recently, we developed a method of sequence comparison based on the notion that different lncRNAs likely encode similar functions through different spatial arrangements of related sequence motifs, and that such similarities might not be detectable by traditional methods of linear sequence alignment (46). In our method, which we termed SEEKR (sequence evaluation through k-mer representation), the sequences of any number of lncRNAs are evaluated by comparing the standardized abundance of nucleotide substrings termed “k-mers” in each lncRNA, where k specifies the length of the substring being counted, and is typically set to values of k = 4, 5, or 6. SEEKR counts k-mers independent of their position in sequences of interest, much like the “bag of words model” used by many language processing algorithms, in which sentences are classified by word abundance without regards to grammar or syntax [47]. Using SEEKR, we demonstrated that k-mer content correlates with lncRNA subcellular localization, protein-binding, and repressive function, and that evolutionarily unrelated lncRNAs with analogous functions shared significant levels of non-linear sequence similarity even when BLAST-like alignment algorithms could detect none [46].

Below, we walk users through five related applications of SEEKR that we have found to be useful. For each application, we enumerate step-by-step instructions. Where relevant, we include code to execute specific functions in python. We have deposited standalone python code to run the major applications of SEEKR in Github (https://github.com/CalabreseLab/seekr). For the simplest implementation of SEEKR, we refer users to a web portal (http://seekr.org). K-mer based classification schemes have been used in many biological contexts ([48-56] and others). Therefore, beyond lncRNAs, the methods that we describe should prove useful in the study of other nucleic acid sequences, such as 5' and 3' untranslated regions of mRNAs and DNA regulatory elements.

3.2 Materials

3.2.1 Hardware Requirements

Personal computer, preferably with a multi-core processor and at least 8GB of RAM.

3.2.2 Software Requirements

1. Python >=3.6. The easiest way to get started with Python is by downloading the Anaconda distribution: https://www.anaconda.com/download.

2. The python packages: numpy, pandas, networkx, python-igraph, louvain. All of these can be installed by running `$ pip install [name]`.

3. R, which can be installed from https://www.r-project.org/.

4. The R packages amap and ctc. amap is hosted at https://cran.r-project.org/web/packages/amap/index.html, and ctc at https://bioconductor.org/packages/release/bioc/html/ctc.html. Both can be installed by running: and can be installed by running:

```
source("http://bioconductor.org/biocLite.R")

biocLite("amap")

biocLite("ctc")
```

5. Java 1.8. See this page for help installing java: https://www.java.com/en/download/help/download_options.xml

6. Java Treeview. http://jtreeview.sourceforge.net/

7. Gephi, which can be installed from https://gephi.org/users/download/.

8. SEEKR (optional). SEEKR is hosted at pypi: https://pypi.org/project/seekr/, and can be installed by running `$ pip install seekr`. SEEKR works on Mac and Linux. As of the time of publication, there is a bug installing several dependencies of SEEKR if using Anaconda Python on MacOS. As a workaround in macOS 10.14.x, run `$ MACOSX_DEPLOYMENT_TARGET=10.14 pip install seekr`. To print the documentation associated with each SEEKR command line tool, simply type the name of the tool in the UNIX terminal (e.g. `$ seekr_download_gencode`).

3.3 Methods

3.3.1 Comparing k-mer contents between a group of lncRNAs

1. *Download lncRNA sequences*

LncRNA sequences can be downloaded from https://www.gencodegenes.org/. For this analysis, we'll use human v22:

ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_22/gencode.v22.lncRNA_transcripts.fa.gz

and mouse v5:

ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_mouse/release_M5/gencode.vM5.lncRNA_transcripts.fa.gz. Unzip these files to produce gencode.v22.lncRNA_transcripts.fa and gencode.vM5.lncRNA_transcripts.fa.gz. The following pipeline will be demonstrated using just the gencode.v22.lncRNA_transcripts.fa file. Mouse, or any other fasta file, can be substituted instead.

Downloading and unzipping can be done manually. Alternatively, if SEEKR is installed locally, you can also download the files from the command line. Use “lncRNA” to specify the biotype of transcripts file, and the “--release" flag to indicate you want a particular version of the fasta file:

```
$ seekr_download_gencode lncRNA -r 22
```

2. *Select 01 isoform*

To avoid bias that may be introduced by counting k-mers across multiple isoforms of the same transcript, we typically only select transcripts ending in 01, which in prior versions of GENCODE, represented the canonical isoform of a gene product. Using this filter, each genomic locus is only represented once.

```
fasta_path = 'v22_lncRNA.fa' #Note 1

with open(fasta_path) as infasta:
    data = [l.strip() for l in infasta]
    headers = data[::2]
    seqs = data[1::2]

fasta01_path = 'v22-01.fa'
with open(fasta01_path, 'w') as outfasta:
    for header, seq in zip(headers, seqs):
        common_name = header.split('|')[4]
        if common_name.endswith('01'): #Note 2
            outfasta.write(header+'\n')
            outfasta.write(seq+'\n')
```

To accomplish the same using the command line tool, pass `seekr_canonical_gencode` the name of the GENCODE fasta file and a path to the newly filtered fasta file:

```
$ seekr_canonical_gencode v22_lncRNA.fa v22-01.fa
```

3. Count k-mers

Next, we define a 2D matrix where each row represents one transcript, each column represents a k-mer, and each element is a normalized and standardized count of how many times a k-mer is found in a transcript. A single row of the matrix, then, defines a “k-mer profile” for a given lncRNA.

```
import pickle
import numpy as np
import pandas as pd

from collections import defaultdict
from itertools import product

# Read fasta file
fasta_path = 'v22-01.fa'
with open(fasta_path) as infasta:
    data = [l.strip() for l in infasta]
    headers = data[::2]
    seqs = data[1::2]

# Initialize data
k=6
kmers = [''.join(i) for i in product('AGTC', repeat=k)]
k_map = dict(zip(kmers, range(4**k)))
counts = np.zeros([len(seqs), 4**k], dtype=np.float32)

# Do counting
```

```
for i, seq in enumerate(seqs):

    row = counts[i]

    count_dict = defaultdict(int) #Note 3
    length = len(seq)

    increment = 1000/length

    for c in range(length-k+1): #Note 4

        kmer = seq[c:c+k]

        count_dict[kmer] += increment

    for kmer, n in count_dict.items():

        if kmer in k_map: #Note 5

            row[k_map[kmer]] = n


# Normalize

counts -= np.mean(counts, axis=0)

counts /= np.std(counts, axis=0)

counts += abs(counts.min()) + 1 #Note 6

counts = np.log2(counts)


# Save csv file

out_path = 'v22-6mers.csv'
```

```
seen = set() #Note 7

names = []

for h in headers:

    name = h.split('|')[4]

    if name in seen:

        name += 'B'

    seen.add(name)

    names.append(name)

pickle.dump(names, open('v22_names-B.pkl', 'wb'))

df = pd.DataFrame(counts, names, kmers)

df.to_csv(out_path, float_format='%.4f')
```

Using the command line tool:

```
$ seekr_kmer_counts v22-01.fa -o v22_6mers.csv
```

3.3.2 Hierarchical clustering of lncRNAs by k-mer content

1. *Cluster with amap*

The visualization tool Java Treeview allows for interactive exploration of large hierarchical clusters. Treeview parses clusters defined by a set of three plaintext files, which describe the structure of row and column clusters: .gtr, .atr, and .cdt. These files can be conveniently produced by the R packages `amap` and `ctc`, which parse a .csv file such as v22_6mers.csv. The R script `treeview_cluster.r` will create the Treeview files:

```
make_treeview <- function(csv, out_gtr, out_atr, out_cdt){
  library(amap)
  library(ctc)

  kmers <- read.csv(csv, header=TRUE, row.names=1)
  kmers <- round(scale(kmers, scale=FALSE), 6)

  # Generate distance matrix using pearson distance
  dist_mat <- Dist(kmers, method="correlation",  nbproc=4)
  dist_mat_trans <- Dist(t(kmers), method="correlation",  nbproc=4)

  # Clustering using average agglomeration method
  clust_row <- hclust(dist_mat,method="average")
  clust_col <- hclust(dist_mat_trans,method="average")

  # Exporting the gtr, atr and cdt files
  r2gtr(clust_row,file=out_gtr, distance=clust_row$dist.method,
dec='.', digits=5)
  r2atr(clust_col,file=out_atr, distance=clust_col$dist.method,
dec='.', digits=5)
  r2cdt(clust_row, clust_col, kmers, labels=FALSE, description=FALSE,
file=out_cdt, dec='.')
}


args <- commandArgs(trailingOnly = TRUE)
do.call(make_treeview, as.list(args))
```

While this script is not part of the seekr module, it can be called from the command line using:

```
$ Rscript treeview_cluster.r v22_6mers.csv v22_6mers.gtr v22_6mers.atr
v22_6mers.cdt
```

2. *Visualize in java treeview*

Launch Treeview, and give Java access to plenty of memory. If insufficient memory is allocated, Treeview will not be able to open the .cdt file. Starting Treeview with 14GB of memory can be done by:

```
$ java -Xmx14000m -jar ~/Downloads/Setups/TreeView-1.1.6r4-
bin/TreeView.jar
```

Substitute the correct path to your TreeView.jar file. Once started, open v22-6mers.cdt via "File" -> "Open". After the file is loaded, change the visualization settings by: "Settings" -> "Pixel Settings...". Find the "Global" section of the pop-up window. Click "Fill" for both "X" and "Y". In the "Contrast" section, set "Value" to 1. In the "Colors" section click "YellowBlue". Close the pop-up window.

The image can be saved by "Export" -> "Save Thumbnail Image" -> "Save".

An ordered list of all transcript names can be exported as well. To do so, you must first select all transcripts. The easiest way to do this is to click on the far left of the dendrogram, so that a portion of the transcripts are highlighted in red. Hold down the up-arrow until all transcripts are highlighted in red. Then click "Export" -> "Save List" -> "Save".

3.3.3 Identifying communities of lncRNAs with related k-mer contents

It takes several steps to convert k-mer profiles into the form needed for identifying communities. We first build an adjacency matrix, describing all pairwise relationships between all lncRNAs, then use the matrix to build a network of lncRNAs. Finally, we can use a network algorithm to assign a community label to each lncRNA.

1. *Build adjacency matrix*

First, we need to build an adjacency matrix. This matrix describes how similar each lncRNA is to all other lncRNAs, as measured by their Pearson’s r-values.

```
import pandas as pd

import numpy as np
```

```
counts = 'v22_6mers.csv'

counts = pd.read_csv(counts, index_col=0)

adjacency = np.corrcoef(counts.values)

adjacency = pd.DataFrame(adjacency, counts.index, counts.index)

adj_path = 'v22_adj.csv'

adjacency.to_csv(adj_path, float_format='%.4f')
```

To calculate our adjacency matrix, we want to compare our k-mer counts file against itself. The seekr command line tool is capable of comparing two separate counts files, so in this case, we need to pass our counts file twice:

```
$ seekr_pearson v22_6mers.csv v22_6mers.csv -o v22_adj.csv #Note 8
```

2. *Sparsify the matrix*

To decrease the runtime of community calculation, we can reduce the number of edges in the network, by sparsifying the adjacency matrix by thresholding below a limit. That is, if the Pearson's r-value between two transcripts is less than the limit, we set that element of the matrix to 0, which removes that edge from our network. However, there is no single best threshold value; it depends heavily on the specific experiment and factors such as the k-mer size used. For example, smaller k-mer sizes will likely need higher thresholds. Therefore, it may be worthwhile to test multiple thresholds. One possible guideline is the mean and standard deviation of the r-values in the adjacency matrix. Two standard deviations above the mean (i.e. the 95th percentile in a normal distribution) is one viable threshold, and easy to compute. In our original publication of SEEKR, we used 0.13 as a threshold, which is what we will use here, but we will also demonstrate how one would calculate a reasonable threshold *de novo*:

```
import pandas as pd

import numpy as np



adj = 'v22_adj.csv'

adjacency = pd.read_csv(adj, index_col=0)

print(adjacency.values.mean() + 2*adjacency.values.std())

limit = .13 #Note 9

np.fill_diagonal(adjacency.values, 0) #Note 10

adjacency[adjacency < limit] = 0

new_adj = 'v22_adj_p13.csv'

adjacency.to_csv(new_adj, float_format='%.4f')
```

In addition to calculating the mean and standard deviation, you can quickly visualize the adjacency matrix from the command line. This will create a pdf file that contains a graph of the distribution of all elements in the adjacency matrix and markings denoting the mean of the distribution as well as one and two standard deviations above the mean. Empirically, we have found that a Pearson's r value of two standard deviations above the mean provides an intuitive threshold that can be used to sparsify any adjacency matrix:

```
$ seekr_visualize_distro v22_adj.csv v22_adj.pdf
```

3. *Convert adjacency matrix to network and find communities*

Once the sparse adjacency matrix has been made, communities can be called with the Louvain algorithm. To use the Louvain algorithm, the adjacency matrix needs to be converted to a network. In this data structure, each lncRNA is represented as a “node” and each non-zero element of the adjacency matrix represents an “edge” between two nodes, describing their similarity. The Louvain algorithm attempts to find communities of nodes having significantly more edges between the nodes within a given community than edges connecting nodes between different communities. Finally, we label each node with the name of the transcript and the community it's found in before saving the graph for visualization. In addition to saving the full graph, we will also produce a two-column csv file where the first column is the name of the lncRNA and the second is the community to which the lncRNA belongs.

```
import numpy as np

import networkx

import igraph

import louvain



adj = 'v22_adj_13.csv'

adjacency = pd.read_csv(adj, index_col=0)

graph = networkx. from_pandas_dataframe(adjacency)

adjacency = None #Note 11
```

```
# Save subgraph

subgraphs = list(networkx.connected_component_subgraphs(graph))

graph_sizes = [sub.size() for sub in subgraphs]

main_sub = subgraphs[graph_sizes.index(max(graph_sizes))]

gml_path = 'v22_sub.gml'

networkx.write_gml(main_sub, gml_path) #Note 12



# Find communities with Louvain

gamma = 1 #Note 13

ig_graph = igraph.Graph.Read_GML(gml_path)

partition = louvain.find_partition(

    ig_graph, louvain.RBConfigurationVertexPartition,

    weights='weight', resolution_parameter=gamma)

n_comms = 5 #Note 14

zipped = zip(main_sub.nodes(), partition.membership)

name2group = {k:v if v <= n_comms-1 else n_comms for k, v in zipped}

networkx.set_node_attributes(

    main_sub, name='Group', values=name2group)

networkx.write_gml(main_sub, gml_path)
```

```
# Save lncRNA communities to csv
with open('communities.csv') as out_file:
    for lncRNA in graph.nodes():
        group = name2group.get(lncRNA, n_comms-1)
        out_file.write(f'{lncRNA},{group}\n')
```

Again, the thresholding value is experiment specific. For that reason, it is a required argument for the command line script. In the instance below, we also save the full gml file, with the “-g" flag, and the two-column csv file listing lncRNAs and communities, with the “-c" flag:

```
$ seekr_graph v22_adj.csv 0.13 -g v22_sub.gml -c v22_comms.csv
```

4. *Visualize in Gephi*

Gephi is open-source software that is useful for visualizing lncRNA community graphs. On launch, Gephi will provide you with a "Welcome" pop-up window. In the "New Project" section, click "Open Graph File". Select `v22_sub.gml`. If loaded correctly, you will receive an "Import report" listing the number of nodes and edges as well as other graph details. Click "OK". In the center of the main application window, you should see a small black circle. This is the default layout and coloring of the graph. Next, we'll color and properly layout the nodes. In the top left of the window, there will be an "Appearance" section. Click "Nodes" -> "Partition" -> "Choose an attribute" -> "Group" -> "Apply". After a few seconds, the nodes of the graph should be colored by group. On the bottom left, there is a section called "Layout". Click "Choose a layout" -> "Yifan Hu" -> "Run".

Running the layout will take time. Progress can be tracked in the bottom right. Once finished, save the image by clicking: "File" -> "Export" -> "SVG/PDF/PNG file" -> "Options". Set "Width" and "Height" to 4096. Click "OK". Name your file and click "Okay" again. Saving the image will also some take time.

3.3.4 SEEKR Python

The command line tools are a convenient way to use SEEKR. However, to gain additional flexibility and performance, one can also consider using SEEKR as a Python module. The code below demonstrates the same pipeline as above (from downloading a fasta file from GENCODE to producing a csv file of lncRNA communities), but runs >10x faster than the command line tools:

```
import numpy as np

import pandas as pd

from seekr import fasta, kmer_counts, graph



downloader = fasta.Downloader()

downloader.get_gencode(biotype='lncRNA', release='22')

fasta_path = 'v22_lncRNA.fa'

fasta01_path = 'v22-01.fa'

maker = fasta.Maker(fasta_path, fasta01_path)

maker.filter1()

names = fasta.Maker(fasta01_path).names
```

```
counter = kmer_counts.BasicCounter(fasta01_path, log2=False)

counter.get_counts()

adj = pd.DataFrame(np.corrcoef(counter.counts), names, names)

comms_path = 'comms.csv'

gm = graph.Maker(adj, csv_path=comms_path, threshold=0.13,
leiden=False)

gm.make_gml_csv_files()
```

3.3.5 Scaling k-mer profiles by protein-binding motifs (Positional Weight Matrices)

One of the underlying assumptions of SEEKR is that lncRNAs derive function from the proteins that they bind. Therefore, a logical step is to utilize the k-mer profile of a given sequence to predict proteins that may bind that sequence. To do this, one can scale k-mer profiles by position weight matrix probabilities (PWMs). We outline this methodology below.

Our code is written to input PWMs in the format provided by the CisBP-RNA database [57]. To download, navigate to http://cisbp-rna.ccbr.utoronto.ca/bulk.php. In ‘By Species’, select Homo_sapiens, then click ‘Download Species Archive’ and in the new page click ‘Download’. However, any PWM can be used if formatted correctly. Individual PWMs must be tab separated and saved in a .txt file. Each PWM must contain a header row with entries [Pos, A,C,G,U]. The ‘Pos’ column contains integers representing the position within the PWM. Each row must sum to 1, excluding the index column, thereby representing the probability of finding each nucleotide at each position within the motif.

This code iterates through the PWM files in `pwm_directory` and calculates the probability of observing all k-mers within each motif. The probability of observing a k-mer in a motif is calculated as the independent probability of observing each nucleotide of the k-mer at the corresponding position within the motif. The weight is then the sum of possible frames that a k-mer could occur in, for example a 5-mer could fall in two different frames in a 6bp motif. Prior to running the code below, users need to derive k-mer counts in the lncRNAs of interest, as specified in Section 3.1.

```
import pandas as pd

import numpy as np

from itertools import product

from pathlib import Path


# path to PWMs

pwm_directory = 'cisbp_pwms/pwms_all_motifs/'


pwm_directory = Path(pwm_directory)


# k-mer counts are produced by seekr_kmer_counts (See section 3.1)

counts_path = 'v22_6mers.csv'


k = 5 #Note 15

kmers = [''.join(p) for p in product('AGTC', repeat=k)] #Note 16
```

```
z_scores = pd.read_csv(counts_path, index_col=0)

score_dict = {}

for pwm_path in pwm_directory.glob('*.txt'):

    try:

        pwm = pd.read_csv(pwm_path, sep='\t')

    except pd.errors.EmptyDataError:

        print(f'The motif file {pwm_path} is empty. Skipping.')

        continue

    pwm.drop('Pos', axis=1, inplace=True)

    pwm = pwm.rename(columns={'U': 'T'}).to_dict()

    kmer2weight = dict(zip(kmers, np.zeros(4 ** k)))

    motif_len = len(pwm['A']).

    if motif_len < k: #Note 17

        kmers_within_kmer = [([kmer[i:i+4] for i in range(k-4+1)],
kmer) for kmer in kmers]

        n_kmers = motif_len - 4 + 1

        for sub_kmers, kmer in kmers_within_kmer:

            for sub_kmer in sub_kmers:

                for frame in range(n_kmers):

                    weight = 1
                    for pos, nucleotide in enumerate(sub_kmer):
                        weight *= pwm[nucleotide][pos + frame]
                    kmer2weight[kmer] += weight
    else:
        for kmer in kmers:
            n_kmers = motif_len - k + 1
            for frame in range(n_kmers):
                weight = 1
                for pos, nucleotide in enumerate(kmer):
                    weight *= pwm[nucleotide][pos+frame]
                kmer2weight[kmer] += weight
    sorted_weights = np.array([kmer2weight[k] for k in
z_scores.columns])
    weighted_z_scores = z_scores.values.copy() * sorted_weights
    scores_sums = weighted_z_scores.sum(axis=1)
    score_dict[pwm_path.name] = scores_sums


#save output
out_df = pd.DataFrame.from_dict(score_dict, orient='index',
columns=z_scores.index)
```

```
out_path = 'pwm_weighted_SEEKR.csv'

out_df.to_csv(out_path)
```

Using the command line tool (specify k-mer length if not using k = 5):

```
$ seekr_pwm cisbp_pwms/pwms_all_motifs v22_6mers.csv -k 6 –o
pwm_weighted_SEEKR.csv
```

3.3.6 Scanning lncRNAs for domains of related k-mer contents

This program is designed to scan a set of fasta sequences, or 'targets', for regions of high correlation to a set of sequences that we define as the 'query' sequences. Typical query sequences might represent functional domains in lncRNAs of interest. Targets are broken up into sliding windows with length and slide designated by the user. Correlations from each tile are then compared against a 'reference' set of sequences that are specified by the user.

This program iterates through k-mer counting three times, which we show explicitly below for completeness. The first iteration calculates the k-mer profile of a query sequence, the second iteration calculates the k-mer profiles for each tile in the target sequence, and the final iteration calculates the k-mer profile for each transcript in the reference set of sequences and correlates them with the query k-mer profiles. This last calculation yields a distribution of Pearon's correlation values from which we can derive the ranks of our targets relative to the queries.

```
import pandas as pd
import numpy as np

from itertools import product
from collections import defaultdict
from scipy.stats import pearsonr
from scipy.stats import percentileofscore

from seekr.kmer_counts import BasicCounter
from seekr.fasta_reader import Reader

# Path to a query of interest (in this example, the sequence of repeat
# B in the lncRNA Xist)
query_path = 'mm10_xist_repeatB.fa'

# This performs standard SEEKR for the query
query = Reader(query_path).get_seqs()[0]
window = 1000 #Note 18
slide = 100
k = 5
kmers = [''.join(p) for p in product('ATCG', repeat=k)]
```

```
k_map = dict(zip(kmers, range(4**k)))


mean_path, std_path = 'mean.npy', 'std.npy'

mean = np.load(mean_path)

std = np.load(std_path)


query_counter = BasicCounter(k=k, mean=mean, std=std)

query_counter.seqs = [query]

query_counter.get_counts()

query_counts = query_counter.counts


q_vs_t_rvals = []

target_path = 'mm10_kcnq1ot1.fa'

target = Reader(target_path).get_seqs()[0]

tiles = []

for i in range(0, len(target), slide):

    end = i + window

    tiles.append(target[i: end])

tiles[-1] += target[end:]
```

```
tile_counter = BasicCounter(k=k, mean=mean, std=std)

tile_counter.seqs = tiles

tile_counter.get_counts()



q_vs_t_rvals = np.array([pearsonr(query_counts[0],
tile_counter.counts[i])[0] for i in range(len(tiles))])

q_vs_ref_rvals = []

ref_path = 'v22-01.fa'

ref = Reader(ref_path).get_seqs()

ref_counter = BasicCounter(k=k, mean=mean, std=std)

ref_counter.seqs = ref

ref_counter.get_counts()

ref_counts = ref_counter.counts



q_vs_ref_rvals = np.array([pearsonr(query_counts[0], ref_counts[i])[0]
for i in range(len(ref))])

ranks = []

for tile_corr in q_vs_t_rvals:

 ranks.append(percentileofscore(q_vs_ref_rvals, tile_corr,
kind='rank'))
```

```
query_target_df = pd.DataFrame(q_vs_t_rvals)

query_target_df_out = 'query_target_pearson.csv'

query_target_df.to_csv(query_target_df_out)

ranks_df = pd.DataFrame(ranks)

ranks_df_path = 'ranks.csv'

ranks_df.to_csv(ranks_df_path)
```

2. This tool requires several pieces of data. 1) A fasta file containing one or more query sequences, 2) A second fasta file containing one or more target sequences which will be tiled into domains, 3) The mean and standard deviation vectors for normalization (e.g. appropriate output from `seekr_norm_vectors`). You can then select the locations for one or both of the possible output files with the '-r' and the '-p' flags. The '-r' flag prints a matrix of Pearson's r values describing the similarity between each query and each tile in each target, and the '-p' flag prints a corresponding matrix of the percentile rankings of the Pearson's r values relative to a reference set of sequences. If you use the '-p' flag you must also use the '-rp' flag, which specifies the reference set of sequences to be used in percentile calculations; for example, 'v22-01.fa'. Also, ensure that the '-k' flags passed to `seekr_norm_vectors` and `seekr_domain_pearson` are the same:

```
$ seekr_norm_vectors v22-01.fa –k 5

$ seekr_domain_pearson mm10_xist_repeatB.fa mm10_kcnq1ot1.fa mean.npy std.npy -rp v22-01.fa –k 5 –r r_values.csv –p percentiles.csv
```

3.4 Notes

1. If you manually download this file from GENCODE’s website, it will be called “gencode.v22.lncRNA_transcripts.fa”.
2. In human, there are a few genomic regions where the canonical isoform is not -001, but -*01 instead, usually -201. It is worth manually examining the GENCODE annotations to ensure that your lncRNA spliceform of interest is included in your analyses.
3. While it is possible to directly increment the numpy array for each k-mer, randomly accessing the array is slow when done billions of times. Instead, a dictionary is used to collect the counts for a single transcript. That way, each element of the array can be accessed only once.
4. Because we want to count overlapping k-mers we cannot use Python's built-in `count` method, and need to manually iterate over the strings ourselves.
5. k-mers that contain non-ACGT nucleotides (eg. ATCGGN) are skipped.
6. In our original publication describing SEEKR, log normalization was not used. In a limited number of tests, we have found that log normalizing k-mer counts prior to performing Pearson’s correlation mildly improves our ability to detect biological meaningful trends. In general, log normalization is an appropriate way to reduce skew in data, and k-mer counts, especially in repetitive regions of RNA, are often skewed. If log2 normalization is not desired, pass the `-nl` flag to `seekr_kmer_counts`.
7. GENCODE transcript names are not necessarily unique. To be able to use names as the index of an R DataFrame, 'B' is appended to transcript names that have already occurred.
8. This array is approximately 250 million elements (16,000 by 16,000 r-values). For a significant efficiency increase (~50x speed, 2x space), consider using the binary flags `--binary_input` and/or `--binary_output` when running `seekr_pearson`. Also note that

usage of these flags may require additional adjustment to flags in other stages of the SEEKR pipeline.

9. The 0.13 Pearson value as a threshold was chosen as a balance between computational efficiency and information retention. In GENCODE v22, the Pearson's value of 0.13 is approximately two standard deviations above the mean similarity between all pairwise lncRNA comparisons. Overall, we found little to no difference in community definition, correlation with lncRNA localization, or ability to predict protein-binding patterns over a range of limit values.
10. The diagonal of the matrix contains all '1' values, since the k-mer profile of a transcript versus itself is a perfect correlation. These edges are not useful for defining communities, so we remove them.
11. This is just done to clear some memory.
12. Writing out to disk at this point is simply used as a way to convert between a networkx graph and an igraph graph. The igraph version is needed for running louvain. There are likely better ways of doing this.
13. Gamma is the resolution parameter for the Louvain algorithm, and is used to tune how many communities are found. Gamma must be greater than 0, and the larger the value, the more communities will be created; consequentially, community sizes are smaller at larger gamma values. We chose to stay with the default resolution parameter, 1, which was supported by CHAMP [(58)]. CHAMP is an algorithm which can help provide context for which values of gamma might be most appropriate for a given graph.
14. Choosing the number of communities can be difficult. We used an estimate based on the hierarchical heatmap, in combination with the size of the communities. In our original publication of SEEKR, community 6 was significantly smaller than community 5 (relative to the ratio between, community 5 and 4, or 4 and 3, etc.; [(46)]). n_communities is defined

here so we can cap the number of communities found by the Louvain algorithm before adding values to the main subgraph below.

15. k = 5 is a reasonable default because it tends to strike a balance between decreasing sparsity in k-mer profiles while still retaining good discrimination between queries and targets.
16. Nucleotide entries in this list must be in exactly the same order as used in Section 3.1, ‘AGTC’.
17. This loop is designed to find all 4-mers within the larger k-mer if the value of k is larger than the length of the motif. For example, the 5-mer ATCGT does not exist within a 4 base pair motif, but two 4-mers within the 5-mer, ATCG and TCGT can fit within a 4 base pair motif. This loop calculates the probabilities of observing the 4-mers separately and then sums the result. No motif in CisBP-RNA database is < 4 base pairs, hence the default of k = 4.
18. The window and slide variables can be set to any positive integer. In our work, we have found that a window approximately the size of the query features, such as the tandem repeat domains of *Xist*, provides good results. In general, increasing the window size smoothens the resulting data whereas decreasing window size gives more detail but increases noise. The slide is best adjusted as a function of the size of your target dataset. If only a couple sequences are being considered, a slide of 1 may be appropriate, but if the study is over the entire transcriptome or otherwise genome-wide, then larger slides can reduce compute time and storage space exponentially.

REFERENCES

1 Cabili, M.N. *et al.* Localization and abundance analysis of human lncRNAs at single-cell and single-molecule resolution. *Genome Biology*. **16**, 20 (2015).

2 Cabili, M.N. *et al.* Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. *Genes Development* **25**, 1915-1927 (2011).

3 Derrien, T. *et al.* The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. *Genome Research* **22**, 1775-1789 (2012).

4 Mele, M. *et al.* Chromatin environment, transcriptional regulation, and splicing distinguish lincRNAs and mRNAs. *Genome Research* **27**, 27-37 (2017).

5 Mukherjee, N. *et al.* Integrative classification of human coding and noncoding genes through RNA metabolism profiles. *Nature Structural Molecular Biology* **24**, 86-96 (2017).

6 Iyer MK. *et al.* The landscape of long noncoding RNAs in the human transcriptome. *Nature Genetics* **47**, 199-208 (2015).

7 Sahakyan, A., Yang, Y. & Plath, K. The Role of Xist in X-Chromosome Dosage Compensation. *Trends in Cell Biology* (2018).

8 West, J.A. *et al.* The long noncoding RNAs NEAT1 and MALAT1 bind active chromatin sites. *Molecular Cell* **55**, 791-802 (2014).

9 Arun, G. *et al.* Differentiation of mammary tumors and reduction in metastasis upon Malat1 lncRNA loss. *Genes Development* **30**, 34-51 (2016).

10 Chakravarty, D. *et al.* The oestrogen receptor alpha-regulated lncRNA NEAT1 is a critical modulator of prostate cancer. *Nature communications* **5**, 5383 (2014).

11 Gutschner T. *et al.* The noncoding RNA MALAT1 is a critical regulator of the metastasis phenotype of lung cancer cells. *Cancer Research* **73**, 1180-1189 (2013).

12 Zhang, B. *et al.* The lncRNA Malat1 is dispensable for mouse development but its transcription plays a cis-regulatory role in the adult. *Cell Reports* **2**, 111-123 (2012).

13 Nakagawa, S. *et al.* The lncRNA Neat1 is required for corpus luteum formation and the establishment of pregnancy in a subpopulation of mice. *Development* **141**, 4618-4627 (2014).

14 Standaert L. *et al.* The long noncoding RNA Neat1 is required for mammary gland development and lactation. *RNA* **20**, 1844-1849 (2014).

15 Lee, S. *et al.* Noncoding RNA NORAD Regulates Genomic Stability by Sequestering PUMILIO Proteins. *Cell* **164**, 69-80 (2016).

16 Munschauer, M. *et al.* The NORAD lncRNA assembles a topoisomerase complex critical for genome stability. *Nature* **561**, 132-136 (2018).

17 Klattenhoff, C.A. *et al.* Braveheart, a long noncoding RNA required for cardiovascular lineage commitment. *Cell* **152**, 570-583 (2013).

18 Lin, N. *et al.* An evolutionarily conserved long noncoding RNA TUNA controls pluripotency and neural lineage commitment. *Molecular Cell* **53**, 1005-1019 (2014).

19 Luo, S. *et al.* Divergent lncRNAs Regulate Gene Expression and Lineage Differentiation in Pluripotent Cells. *Cell Stem Cell* **18**, 637-652 (2016).

20 Ng, S.Y., Johnson, R. & Stanton, L.W. Human long non-coding RNAs promote pluripotency and neuronal differentiation by association with chromatin modifiers and transcription factors. *EMBO Journal* **31**, 522-533 (2012).

21 Sheik, M.J., Gaughwin, P.M., Lim, B., Robson, P. & Lipovich, L. Conserved long noncoding RNAs transcriptionally regulated by Oct4 and Nanog modulate pluripotency in mouse embryonic stem cells. *RNA* **16**, 324-337 (2010).

22 Lai, K.M. *et al.* Diverse Phenotypes and Specific Transcription Patterns in Twenty Mouse Lines with Ablated LincRNAs. *PLoS One* **10**, e0125522 (2015).

23 Swiezewski, S., Liu, F., Magusin, A. & Dean, C. Cold-induced silencing by long antisense transcripts of an Arabidopsis Polycomb target. *Nature* **462**, 799-802 (2009).

24 Carpenter, S. *et al.* A long noncoding RNA mediates both activation and repression of immune response genes. *Science* **341**, 789-792 (2013).

25 Elling, R. *et al.* Genetic Models Reveal cis and trans Immune-Regulatory Activities for lincRNA-Cox2. *Cell Reports* **25**, 1511-1524 (2018).

26 Kotzin, J.J. *et al.* The long non-coding RNA Morrbid regulates Bim and short-lived myeloid cell lifespan. *Nature* **537**, 239 (2016).

27 Barry G. *et al.* The long non-coding RNA Gomafu is acutely regulated in response to neuronal activation and involved in schizophrenia-associated alternative splicing.

Molecular Psychiatry **19**, 486-494 (2014).

28 Goff, L.A. *et al.* Spatiotemporal expression and transcriptional perturbations by long noncoding RNAs in the mouse brain. *Proc Natl Acad Sci* **112**, 6855-6862 (2015).

29 Mercer, T.R., Dinger, M.E., Sunkin, S.M., Mehler, M.F. & Mattick, J.S. Specific expression of long noncoding RNAs in the mouse brain. *Proc Natl Acad Sci* **105**, 716-721 (2008).

30 Powell, W.T. *et al.* A Prader-Willi locus lncRNA cloud modulates diurnal genes and energy expenditure. *Human Molecular Genetics* **22**, 4318-4328 (2013).

31 Raveendra, B.L. *et al.* Long noncoding RNA GM12371 acts as a transcriptional regulator of synapse function. *Proc Natl Acad Sci* **115**, e10197-e10205 (2018).

32 Sauvageau, M. *et al.* Multiple knockout mouse models reveal lincRNAs are required for life and brain development. *eLife* **2**, e01749 (2013).

33 Sone, M. *et al.* The mRNA-like noncoding RNA Gomafu constitutes a novel nuclear domain in a subset of neurons. *Journal of Cell Science* **120**, 2498-2506 (2007).

34 Grote, P. *et al.* The tissue-specific lncRNA Fendrr is an essential regulator of heart and body wall development in the mouse. *Developmental Cell* **24**, 206-214 (2013).

35 Han, P. *et al.* A long noncoding RNA protects the heart from pathological hypertrophy. *Nature* **514**, 102-106 (2014).

36 Matkovich, S.J., Edwards, J.R., Grossenheider, T.C., de Guzman Strong, C. & Dorn, G.W., 2nd. Epigenetic coordination of embryonic heart transcription by dynamically regulated long noncoding RNAs. *Proc Natl Acad Sci* **111**, 12264-12269 (2014).

37 Wang, K. *et al.* APF lncRNA regulates autophagy and myocardial infarction by targeting miR-188-3p. *Nature communications* **6**, 6779 (2015).

38 Kopp, F. & Mendell, J.T. Functional Classification and Experimental Dissection of Long Noncoding RNAs. *Cell* **172**, 393-407 (2018).

39 Geisler, S. & Coller, J. RNA in unexpected places: long non-coding RNA functions in diverse cellular contexts. *Nature Review Molecular Cell Biology* **14**, 699-712 (2013).

40 Guttman, M. & Rinn, J.L. Modular regulatory principles of large non-coding RNAs. *Nature* **482**, 339-346 (2012).

41 Rinn, J.L. & Chang HY. Genome regulation by long noncoding RNAs. *Annual Review of Biochemistry* **81**, 145-166 (2012).

42 Kornienko, A.E., Guenzl, P.M., Barlow, D.P. & Pauler, F.M. Gene regulation by the act of long non-coding RNA transcription. *BMC Biology* **11**, 59 (2013).

43 Hezroni, H. *et al.* Principles of long noncoding RNA evolution derived from direct comparison of transcriptomes in 17 species. *Cell Reports* **11**, 1110-1122 (2015).

44 Grant, J. *et al.* Rsx is a metatherian RNA with Xist-like properties in X-chromosome inactivation. *Nature*. **487**, 254-258 (2012).

45 Johnson, R.N. *et al.* Adaptation and conservation insights from the koala genome. *Nature Genetics* **50**, 1102-1111 (2018).

46 Kirk, J.M. *et al.* Functional classification of long non-coding RNAs by k-mer content. *Nature Genetics* **50**, 1474-1482 (2018).

47 The conversational interface. New York, NY: Springer Berlin Heidelberg; 2016. pages cm p.

48 Blaisdell, B.E. Effectiveness of measures requiring and not requiring prior sequence alignment for estimating the dissimilarity of natural sequences. *J Mol Evol* **29**, 526-537 (1989).

49 Burge, C., Campbell, A.M. & Karlin, S. Over- and under-representation of short oligonucleotides in DNA sequences. *Proc Natl Acad Sci* **89**, 1358-1362 (1992).

50 Kari, L. *et al.* Mapping the space of genomic signatures. *PLoS One* **10**, e0119815 (2015).

51 Lees, J.A. *et al.* Sequence element enrichment analysis to determine the genetic basis of bacterial phenotypes. *Nature communications* **7,** 12797 (2016).

52 Pandey, P., Bender, M.A., Johnson, R., Patro, R. & Berger, B. Squeakr: an exact and approximate k-mer counting system. *Bioinformatics* **34**, 568-575 (2018).

53 Blanchette, M. & Tompa, M. Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Research* **12**, 739-748 (2002).

54 Dubinkina, V.B., Ischenko, D.S., Ulyantsev, V.I., Tyakht, A.V. & Alexeev, D.G. Assessment of k-mer spectrum applicability for metagenomic dissimilarity analysis. *BMC Bioinformatics* ***17***, 38 (2016).

55 Friedman, R.C., Farh, K.K., Burge, C.B. & Bartel, D.P. Most mammalian mRNAs are conserved targets of microRNAs. *Genome Research* **19**, 92-105 (2009).

56 Solis-Reyes S, Avino M, Poon A. & Kari L. An open-source k-mer based machine learning tool for fast and accurate subtyping of HIV-1 genomes. *PLoS One* **13**, e0206409 (2018).

57 Ray, D. *et al.* A compendium of RNA-binding motifs for decoding gene regulation. *Nature* **499**, 7457 (2013).

58 Weir, W.H., Emmons, S., Gibson, R., Taylor, D. & Mucha, P.J. Post-Processing Partitions to Identify Domains of Modularity Optimization. *Algorithms* **10**, 3 (2017).

CHAPTER 4

Finding XIST-like pre-mRNAs with repressive functions

4.1 Introduction

In our previously publish work, we introduced SEEKR, a tool for classifying lncRNAs and making predictions of their functional role, based on the transcripts k-mer content. We found that by creating k-mer profiles of lncRNAs, we could build communities of lncRNAs with similar sequence content. The lncRNA communities had similar biological properties, which allowed us to make general inferences about the functional roles of lncRNAs within a community. Furthermore, we were able to use a set of transcripts of unknown function to experimentally demonstrate a strong correlation between the transcript's k-mer profile similarity to the lncRNA XIST and the transcript's ability to repress transcription *in-cis*. The more similar a transcript was to XIST, the more likely it heavily repressed nearby transcription.

In this work, we achieve two main goals. First, we address some of the underlying assumptions and weaknesses in our original work thereby strengthening our claims surrounding SEEKR's utility. In particular, we explore our acceptance of established gene annotations and our use of a single RNA transcript per gene locus. Second, we expand SEEKR's utility by demonstrating its applicability, beyond lncRNAs, to protein coding genes.

Accurate gene annotations are critical for understanding the relationship between transcript sequence and function. If the function of an RNA is known but the sequence for the transcript truly performing that function *in vivo* is different than what has been annotated through organizations such as GENCODE[1], it is impossible to state what sequences elements contribute to the RNA's function. In our previous work, we used GENCODE annotations for all lncRNA genes. For each gene, we selected the 01 isoform, assuming it would be the most prevalent isoform, and the most likely to be relevant. However, we hypothesize that GENCODE annotations may be biased towards spliced annotation and that unspliced RNA isoforms may be functionally relevant *in vivo*.

This hypothesis that unspliced RNAs may have biological roles is not limited to lncRNAs. Indeed, we hypothesize that pre-mRNA transcripts may be able to repress transcription *in-cis*, in an XIST-like manner. If this hypothesis is true, then pre-mRNA transcripts would need to be expressed and retained at a high enough level to perform their function. In general, protein coding genes are expressed at a higher level than lncRNA genes[2], however, mammalian splicing machinery is efficient and pre-mRNA is quickly spliced into mRNA[3-5]. The number of transcripts per cell necessary for a transcript to be physiologically relevant is unknown and certainly varies by transcript and role. Other work in our lab, however, has demonstrated measurable repressive effects of lncRNAs with low copy number. Specifically, we have shown that while Xist is expressed at approximately 200 copies in TSCs, Kcnq1ot1 and Airn, two other lncRNAs known to repress transcription in *cis*, can silence genes across 5Mb and 14Mb at only eight and nine copies per cell, respectively. Given the low expression levels of these lncRNAs, even if a vast majority of a protein-coding gene's transcripts were spliced and exported immediately upon transcription, it is reasonable to hypothesis that the small minority of transcripts retained on chromatin may be enough to regulate local transcription.

Theoretically, SEEKR's sequence comparison methodology is equally applicable to protein-coding genes as it is to lncRNAs. mRNA k-mer profiles are distinctly different from lncRNAs, on the whole (data not shown), however, there is no biological reason not to use SEEKR for measuring transcript similarity of mRNAs and between lncRNAs and mRNA. Here, we show we can use SEEKR to find a set of pre-mRNA molecules which are XIST-like, and may potentially regulate local transcription.

4.2 Results

The genome browser tracks in Figure 1 illustrates a key example of a misannotation by GENCODE. The tracks show RNAseq read density for the nuclear fraction of TSCs over the Airn locus. Airn is a known repressive lncRNA which has been shown to repress multiple genes across several megabases up and downstream of the Airn locus. GENCODE annotates Airn, shown in green, as a primarily intron containing RNA, whose spliced product is approximately 1Kb. The red and blue tracks labeled “d_nuc_F” and “d_nuc_R”, forward and reverse aligned RNAseq reads, however, demonstrates reads aligning across the entire Airn locus, with little to no bias towards the GENCODE annotated bias. Other studies in our lab have confirmed that the functional Airn product is approximately 90Kb (data not shown). This data suggests that other genes may also express unspliced isoforms at a physiologically relevant level.

To test if we could find any pre-mRNA transcripts with XIST-like k-mer profiles, we first developed our own set of annotations, derived from GENCODE’s gene annotations (see Methods). First, as a baseline, we measured all lncRNA similarities relative to Xist-001, Kcnq1ot1-001, which is unspliced, and our own unspliced annotation of Airn (Figure 1A). Here, the threshold for lncRNAs with SEEKR similarities relative to XIST three standard deviations above the mean was 0.157. Only 49 lncRNAs were more similar to XIST than this threshold. Furthermore, the single most similar transcript to Xist is Gm45159, which has a similarity of 0.298. We also measured the unspliced lncRNA and mRNA classes of transcripts (Figure 1B-C). Interestingly, in both cases, the distributions were wider than the spliced lncRNA distribution we had used in our previous work. For unspliced lncRNA transcripts we found that 1953 sequences passed the threshold of 0.157 (as opposed to 49).

Finally, we performed the same calculation for pre-mRNA transcripts. For all three lncRNAs, the distributions have visibly larger variance that the corresponding spliced lncRNA distributions. For XIST, there were 7351 unspliced protein coding transcripts that were more similar to XIST than the 0.157 threshold. Perhaps even more astonishingly, there were 1898 unspliced protein-coding genes that were more similar to Xist than any lncRNA. It is worth nothing that while while spliced lncRNAs contains the smallest of annotations, population size is not the reason for the variability in these distribution of these populations. Sub-sampling unspliced pre-mRNA annotations to the size of the spliced lncRNA type produces an identical distribution to the original full sample (data not shown). From this data, we conclude that there are pre-mRNA sequences with XIST-like k-mer profiles.

Next, we tested if there was any evidence of cells transcribing and retaining pre-mRNA molecules at physiologically relevant levels. To investigate this idea, we measured the expression levels of all transcripts in our data set, using publicly available data ENCODE RNAseq for K562 and HepG2 cells (see our previous work for details regarding the data set). Previously when measuring expression levels, our simplified assumption that each lncRNA genomic locus expressed only single canonical isoform allowed us to also assume that all RNAseq reads within the exons of that canonical isoform should be assigned to that transcript.

Assigning an RNAseq read to a genomic region annotated as having multiple isoforms, as a majority of the loci in our data set do, is inherently probabilistic. Much work has gone into building complex probabilistic models capable of account for multiple biological factors when assigning reads to an isoform. These models have been packaged into several popular tools, including Salmon[6], kallisto[7], and TIGAR[8]. We initially attempted to use Salmon for isoform quantification. However, we found its output didn't match our needs. For example, despite the number of reads aligning to the annotated “introns” of Airn, and without the support of any junction overlapping reads, Salmon estimated that the abundance of the spliced Airn annotation was significantly higher than the unspliced (data not shown). Instead, we quantified isoform abundance using our own more basic approach (see Methods for details).

We plotted the distributions of all human transcript expression levels (Figure 3A-B), relative to XIST and KCNQ1OT1. We also repeated this experiment in mouse TSCs, additionally labeling unspliced Airn (Figure 3C). The qualitative relationships between the four types of transcripts are the same across all three cell types, though there is a larger separation in expression levels between lncRNAs and protein coding transcripts in the TSCs. To roughly estimate how many unspliced protein coding genes might be present at physiologically relevant levels, we counted the number of transcripts more highly expressed than Kcnq1ot1. There were 187, 559, and 886 transcripts above this threshold for HepG2, K562 and TSCs, respectively.

In order to robustly find XIST-like lncRNAs, we decided to use a network based approach to find transcripts in the same community as XIST. For this type of experiment, forming communities is beneficial in two ways. First, it may be possible that we are interested in finding transcripts that are not necessarily the most similar to XIST, but are instead similar to many of the same transcripts as XIST. Second, can provide a thresholding mechanism. There is no particular lower bound for a Pearson’s r-value that denotes an “XIST-like” transcript. Selecting only transcripts within the same community as XIST provides a way of limiting RNAs to consider for further examination. Because there were more than four times as many transcripts (or nodes) in the network than when we used only spliced lncRNAs, we began with approximately 20 times the amount of edges. Therefore, to create communities, we used a new method for retaining edges in the network, and we also integrated several other improvements that have been created/discovered since we created our communities last time.

To begin, we count k-mers and perform normalization as described in our previous work, with the addition of $\log_2$ normalizing each element. Next, we create an adjacency matrix from the counts. This adjacency matrix has over four billion edges. In order to efficiently explore the structure of the network and test multiple community definitions, it was important to remove a vast majority of these edges. In fact, we estimated that we wanted to keep less than 0.01% of edges. Simply removing all the edges below the 99.99th percentile completely disconnected a majority of the nodes in the network and was not informative (data not shown). Instead, we developed a two-pass filter that allowed us to keep almost all nodes connected to the largest connected subgraph. Next, we selected the approximate number of communities which should be included in the network partition. To some degree, this process is inherently subjective, as there is no single best value for the final number of detected communities. By default, the community detection algorithm labels dozens of communities, which is too large a number for our purposes. We used hierarchical clustering (Figure 4A) and CHAMP, run with the Leiden algorithm[9], to select our specific community definition. Finally, we visualized the communities with Gephi (Figure 4C). The relationship between the hierarchical clustering and network partition methods is also visualized (Figure 4B). See Methods for a full description of the community formation algorithm. We determined that our data set contained 11 total communities; ten detected communities ranging in size from 8950 to 4164, and an additional "Null" community containing 746 transcripts (Table 4.1). To provide more context for the contents of each community, we performed a basic characterization of each community and the transcripts within the communities (Table 4.2).

Our method does not produce communities that are solely a single class of transcript. Instead each community is a mixture of classes, though, each community is biased towards one or two classes. This supports our hypothesis that at the level of k-mers, there is some amount of similarity between certain subsets of lncRNAs and subsets of protein coding genes. As expected, given the data shown in Figure 4.2, the XIST community (#4) is biased away from spliced lncRNAs, containing only 11.6% instead of the expected 18.4% (pvalue=1.2 * 10^-264). Community #4 is also the most AT rich community, with a mean GC content of only 37% per transcript (Table 4.2).

Following the same logic presented in our previous work—specifically that cellular localization is an important factor in determining an RNA transcript's function—we measured expression levels of all transcripts in nuclear and cytosolic cell fraction samples, and calculated each transcripts nuclear ratios. Distributions of nuclear ratios were plotted for each community using a boxenplot (Figure 4.5). These plots are similar to boxplots, but more suitable for larger data sets because they visualize more quantiles of the data, and provide the user with a better sense of the shape of the distribution within the tails of the data. As expected given the export efficiency of protein coding transcripts, these communities are on average more cytosolic than the purely spliced lncRNA communities we published previously. Encouragingly, the XIST community is the second most strongly nuclear community in both K562 and HepG2 cells. The results of post-hoc Tukey-HSD tests calculating all pairwise significant differences between communities shows that 41 of 55 and 40 of 55 comparisons were significant in HepG2 cells and K562 cells, respectively. From this data, we conclude that these communities provide a significant amount of information about the cellular localization of the RNAs.

Because lncRNAs are not catalytic, their function is likely primarily determined by the set of RNA binding proteins with which they interact. The same logic also applies to pre-mRNA. Therefore, one of the most important criteria for a predicting a pre-mRNA transcript with XIST like repressive activity is a pre-mRNA transcript with an XIST like protein binding profile. To find pre-mRNA transcripts with similar protein binding profiles to XIST, we analyzed an updated version of the ENCODE eCLIP data described in our previously published work. That is, we collected the 156 eCLIP experiments used previously, as well as an additional 67 experiments which had been newly uploaded, for a total of 223 eCLIP experiments. From these experiments, we created two protein binding profiles for each transcript, one for HepG2 cells and one for K562 cells (see Methods).

In order to visualize the structure of the data within the protein binding profile matrices, we first calculated the count of all non-zero elements in each row. This provides the number of proteins with which a given transcript interacts, regardless of the strength or coverage of the interaction. We then plotted these sums as a distribution (Figure 4.6A). Broadly, transcripts either interacted with (nearly) zero proteins, or many proteins. We note that while we used ENCODE data to filter lowly expressed transcripts before creating these distributions, the RNAseq data used for filtrations was from a different data set. Some portion of the transcripts that interact with no proteins may be lowly expressed, or otherwise undetectable via eCLIP. Regardless, this data indicates that protein-RNA interactions are pervasive across the transcriptome. We also mark XIST's placement in K562's distribution, which binds 119 of 120 proteins.

XIST's high number of protein interactions prompted us to ask to what extent interactions are found across XIST's sequence. We reasoned that it may be possible that a majority of these interactions are fairly weak and only supported by small peaks at localized positions in the transcript. We plotted the distribution of values in XIST's protein binding profile, along with the distributions of all other elements in the HepG2 and K562 matrices. Surprisingly, the mean coverage ratio for a protein binding XIST was significantly higher than the background distribution of all other transcripts (Figure 4.6B). This result was exciting because it indicates that in order to find pre-mRNAs with XIST-like protein binding profiles, we must find pre-mRNAs that were detected as interacting extensively with RNA binding proteins. The opposite case—selecting pre-mRNAs with few interactions— greatly increases the likelihood of false positives.

As a simple test of likelihood of pre-mRNAs having XIST-like protein binding profiles, we selected the 289 transcripts in the 99th percentile of Pearson's r-value scores relative to XIST and asked how many of them were pre-mRNAs. Astonishingly, 253 of the 289 transcripts were pre-mRNAs. By chance, we would expect only 104 of them to be pre-mRNAs (chi squared test, p-value= 2.9e-74). Encouragingly, KCNQ1OT1, was 70th on the list of sorted XIST-like protein binding profiles, with an r-value of 0.71. Finally, a surprising number of pre-mRNAs for RNA binding proteins and other relevant proteins were present in the list. These RNAs included: SRSF4-un, HNRNPD-un, HNRNPH1-un, SRSF11-un, RBM39-un, HNRNPU-un, HNRNPA3-un, HNRNPH3-un, EZH2-un, HNRNPC-un, SFPQ-un, PCBP2-un, KHDRBS1-un, PRC1-un, HNRNPF-un, SAFB-un, HNRNPA2B1-un. The rational for why these pre-mRNAs are among the most XIST-like transcripts is unknown, but worth future exploration.

We then asked if there was a relationship between a transcript's k-mer and protein binding profiles relative to XIST, (e.g. are the transcripts with high similarity to XIST at the protein-binding level the same transcripts that are similar to XIST at the sequence level). As a baseline, we measured all the Pearson's r-values relative to XIST of all transcripts expressed in K562 cells, randomly paired the result with the r-value of one of the r-values from comparisons between XIST's protein binding profile and other transcript's protein binding profiles, and plotted the two-dimensional distribution (Figure 4.6C). We also plotted a line of best fit for the data. As expected, there is no correlation. Then, we repeated the experiment without shuffling. That is, each transcript's k-mer r-value was paired with its own protein binding r-value. We again plot the distribution and line of best fit (Figure 4.6D). Excitingly, this data shows a strong correlation (r=0.46, p-value~=0). For interpretability, we removed all transcripts with a protein binding r-value between 0.18 and 0.20, since these represent transcripts with no detected protein binding interactions. We also performed the analysis with these values included, which showed a negligible difference in results (data not shown). From this data, we conclude that a transcript's protein binding interactions are dependent on its k-mer content, and more specifically, that RNA's with Xist-like k-mer profiles likely bind similar proteins to XIST in *vivo*.

Given this data, we expected that the 289 proteins in the 99th percentile of XIST-like protein binding profiles would have higher than average k-mer similarity scores to XIST. This is indeed the case. While the mean r-value for transcripts not in the 99th percentile is 0.03, the mean for the 289 transcripts is 0.20 (Mann-Whitney U test, p-value= 5.4e-51). In turn, we also expected that, given the transcript's high Pearson's r-values, many of them would be found in the same community as XIST. Surprisingly, this was not the case. Instead, 121 of the 289 transcripts were from community #6, which is significantly higher than expected by chance. (chi squared test, p-value=3.6e-73). This observation may be explained by the overabundance of pre-mRNA transcripts in community #6. We recorded the observed and expected counts, along with the associated p-value for all communities in Table 4.3. This data shows that highly similar sequences also have highly similar protein binding profiles.

4.3 Methods

4.3.1 Creating transcript annotations

We downloaded GENCODE v26 (ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_26/gencode.v26.annotation.gtf.gz) and vM14 (ftp://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_mouse/release_M14/gencode.vM14.annotation.gtf.gz) GTF files for human and mouse, respectively. From these files, we extracted genes, along with their associated transcript and exon features, if they were labeled as either "protein coding" or "lncRNA". Next, each gene feature was checked for an unspliced transcript. An unspliced transcript was defined as a transcript annotated with a single exon whose start and stop positions were the same as the transcript's start and stop positions. If there was no annotated unspliced transcript, one was inserted. Then, the gene was checked for additional transcript isoforms. If a spliced 01 isoform was present, it was kept; all other spliced isoforms were discarded. After processing, the new GTF files had all "protein coding" and "lncRNA" genes, each of which had an unspliced transcript annotation and, potentially, a single canonical spliced transcript annotation.

4.3.2 Measuring expression levels between unspliced and spliced RNA isoforms

All RNAseq reads are aligned to GRCh38_p10 with STAR using default setting with the exception of the `--sjdbGTFfile`, used to pass a custom GTF file containing all isoforms annotated by GENCODE, plus the additional annotations of an unspliced isoform for each gene which did not previously have an unspliced annotation. Next, FeatureCounts is performed twice, using seperate flags and GTF files. The first run quantifies the unspliced isoforms. FeatureCounts is passed the same GTF file used for aligning reads with STAR. FeatureCounts is then run using `--fracOverlap 1` flag and `-g transcript_id` (which allows aggregation of exon counts per transcript instead of per gene). By default, FeatureCounts only assigns reads that uniquely overlap a single genomic region. Given that the GTF file described above has an exon feature spanning the entire gene locus for each gene, this configuration allows us to map all reads which fall within a gene but do not share any overlap with *any* spliced exonic region (to be conservative, we refuse to assign any read overlapping any exon, not just exons from the canonically spliced isoform, to the unspliced transcript). Each read falls entirely into an intronic regions. Thus, we can be confident that each read assigned during this run of FeatureCounts should belong to the unspliced isoform. Given the number of reads uniquely assigned to introns, and the intronic length, we calculate an RPKM value for unspliced expression by assuming the reads-per-base-pair (RPBP) across the whole unspliced transcript is uniform, both for the introns and potential exons.

The second run quantifies the spliced isoforms of each gene. Here, FeatureCounts is passed a different GTF file containing only the unspliced and potentially the canonical spliced annotations, if it exists, for each gene. FeatureCounts is then run using the `-O` flag, which allows for each read to be assigned to more than one matched exon, and `-g transcript_id` again. For each read assigned to the splicing isoform, there is some probability that that read truly belongs to the unspliced isoform instead. To subtract out this background, we calculate the RPBP of the spliced trancript by dividing the FeatureCounts value by the length of the transcript. From this spliced RPBP value we subtract the unspliced RPBP calculated previously. If this result is less than 0, we set the spliced RPBP to 0 and make the assumption that the spliced isoform is not expressed. Next, we convert back to raw reads per region by multiplying by the length of the spliced transcript. Finally the data set is normalized to RPKM values across all transcripts.

4.3.2 Building communities

Communities of transcripts were formed by counting all overlapping k-mers for each transcript of interest to create a matrix of k-mer counts. As in our previous work, the raw counts were then row normalized by the length of each transcript. Columns were normalized by subtracting column means and dividing the column standard deviation from each column element. Additionally, each element was $\log_2$ normalized by adding one plus the minimum element of the k-mer counts matrix to each element (so that the smallest element in any k-mer counts matrix is 1), then applying a $\log_2$ transform elementwise.

A completely connected adjacency matrix was then created by calculating all pairwise Pearson r-values between rows in the k-mer count matrix. We then performed a two-pass filtering algorithm. In the first path, all edges below 0.1 were removed. This was a less stringent threshold than used in our previous method, but it allowed us to create a final network where fewer of the nodes are completely disconnected. For the second pass, the r-value of the 100th strongest edge was calculated for each node in the network. If a node had fewer than 100 edges, the r-value of the weakest edge was retained. This value was considered the “limit” for a given node. Then, for each edge, the edge was removed from the graph if the edge’s r-value was lower than the previously calculated “limit” for both of the nodes to which the edge is connected. Note that, because the edge must fail to be above both limits, many nodes can have many more than 100 edges.

Next, we selected an approximate number of communities the network should have. Our choice was informed by two things. First, we visualized the k-mer count profiles using hierarchical clustering and TreeView. TreeView crashed with a segfault while attempting to load the entire normalized count matrix. Therefore, we randomly sub-sampled 45,000 rows, and only visualized them. By eye, we manually counted the number of clusters visible in TreeView, and used that as an estimate of how many communities to expect in the network. Second, we ran CHAMP for 100 iterations over a range of zero to five for the resolution parameter. While running CHAMP, we used the newly developed Leiden algorithm to find partitions instead of the classical Louvain method.

For each CHAMP approved partition, we calculated the size of each community. Then, for each community, we calculated the ratio of its size to the size of the next largest community. When gamma was small, there was a large variance in community sizes, with a few major communities and many minor communities. The largest of the minor communities was several times smaller than the smallest of the major communities. When gamma is large, the community sizes become more uniform with each community having approximately the same size as the next largest or smallest community. When selecting a CHAMP approved partition, we chose to use a partition where the number of major communities was approximately the same as the number of communities we estimated by eye using the TreeView heatmap.

All transcripts that were part of a minor communities were then aggregated into a "Null" community. In our previous work, between a third and a half of transcripts fell into the "Null" community, making it suitable for baseline comparisons against the other communities. Using this current algorithm, the "Null" community was a small portion of the total number of nodes, and may not be useful for comparisons.

4.3.4 Visualizing communities in Gephi

Our use of Gephi to visualize the network has also changed from our previous method. Instead of running Yufan Hu as a layout algorithm, we used OpenOrd. Yufan Hu provided a better visualization when many of nodes were completely disconnected, but OpenOrd provided an equally high-quality layout with the newer well-connected network. However, OpenOrd is significantly faster than Yufan Hu. After OpenOrd has finished running, filter edges below .5. Note that this much stricter threshold is for visualization purposes only, and should not be used to effect the definition of the network or the communities outside of Gephi.

An additional complication was Gephi's random node coloring, which made it difficult to consistently label all communities in a visually distinct manner. This made it harder to manually make a decision about the quality of a community definition, since two separate communities looked like one, due to their colors similarity. The randomness also made it more difficult to keep track of communities between multiple runs of the Leiden algorithm and subsequent visualization. For more control over node colors, we downloaded Gephi's "Scripting Plugin" in order to programmatically (using Python) set the color of each community using a color visually distinctive colors, Matplotlib's Tab20 palatte[10].

4.3.5 Plotting nuclear ratios

This analysis uses the same ENCODE cellular fractionation RNAseq data sets used in our previous work. We used the same method for calculating isoform abundance as we did for the whole cell RNAseq. Once RPKM values were calculated for each transcript, nuclear ratios were found by dividing the nuclear RPKM value by the sum of nuclear and cytosolic RPKM values for each transcript. The distributions nuclear ratios were then plotted for each community using Seaborn's "boxenplot" function.

4.3.6 Calculating protein binding profiles

For each of the 223 eCLIP experiment available on ENCODE, we used the experiment's narrowPeaks files to calculate the ratio of each RNA covered by each RNA-binding protein via the bedtools coverage tool. This output provides the total number of base pairs covered by all peaks across all replicates for each transcript. We then divide the total base pairs covered by the length of the transcript. These coverage values were then normalized using the same methodology used for normalizing k-mer counts. That is, each column is mean centered and divided by its standard deviation. An elementwise $\log_2$ transform was also applied. We term the list of a transcript's normalized coverage ratios across all eCLIP experiment in a single cell type its "protein profile", for that cell type.

Table 4.1 Counts of the number of each type of RNA transcript present in each community for human transcripts. Totals are also provide in the final row and column.

Transcript Type

Community	Spliced lncRNA	Unspliced lncRNA	Spliced mRNA	Unspliced pre-mRNA	Total
0	1259	934	5622	1135	8950
1	1443	1308	3076	2248	8075
2	2697	1354	2881	349	7281
3	1147	1767	585	3287	6786
4	738	2256	1940	1441	6375
5	1753	1537	1031	1845	6166
6	490	1636	373	3653	6152
7	1052	2497	816	1506	5871
8	334	899	1118	2762	5113
9	903	1311	394	1556	4164
10	249	100	362	35	746
Total	12065	15599	18198	19817	65679

Table 4.2 Basic characterization of RNA communities. Size) the number of transcripts in each community. lncRNA) the number of lncRNA transcripts. Spliced) the total number of spliced transcripts (summing lncRNAs and mRNAs). mRNA) the number of mRNA transcripts. Unspliced) the total number of unspliced transcripts. GC Content) the mean ratio of GC nucleotides to transcript length for RNAs in the community. Length) Mean sequence length of RNAs in the community.

Community	Size	lncRNA	Spliced	mRNA	Unspliced	GC Content	Length
0	8950	2193	6881	6757	2069	0.567	1964
1	8075	2751	4519	5324	3556	0.581	2681
2	7281	4051	5578	3230	1703	0.424	1717
3	6786	2914	1732	3872	5054	0.502	7890
4	6375	2994	2678	3381	3697	0.372	5391
5	6166	3290	2784	2876	3382	0.512	2442
6	6152	2126	863	4026	5289	0.429	24610
7	5871	3549	1868	2322	4003	0.407	7063
8	5113	1233	1452	3880	3661	0.394	16520
9	4164	2214	1297	1950	2867	0.457	4069
10	746	349	611	397	135	0.455	2186

Table 4.3 Chi-squared tests of the number of observed transcripts in the set of 289 99th percentile protein binding profile similarities relative to XIST. Observed) the number of transcripts from a given community found in the set of 289 transcripts. Expected) the expected number of transcripts that should be included from a given community, based on the size of the community, and assuming a random uniform distribution of all communities within the set of 289 transcripts. Outgroup observed) the number of transcripts from a given community not found in the set of 289 transcripts. Outgroup expected) the number of transcripts from a given community expected to not to be included in the set of 289 transcripts. p-value) the significance of the difference between the observed and expected columns, as evaluated by a chi-squared test.

Community	Observed	Expected	Outgroup observed	Outgroup expected	p-value
0	0	39	8950	8911	3.88E-10
1	1	35	8074	8040	8.43E-09
2	3	32	7278	7249	2.78E-07
3	35	29	6751	6757	0.26
4	19	28	6356	6347	0.09
5	7	27	6159	6139	0.0001
6	121	27	6031	6125	1.84E-73
7	12	25	5859	5846	0.009
8	79	22	5034	5091	4.04E-34
9	12	18	4152	4146	0.16
10	0	3	746	743	0.08

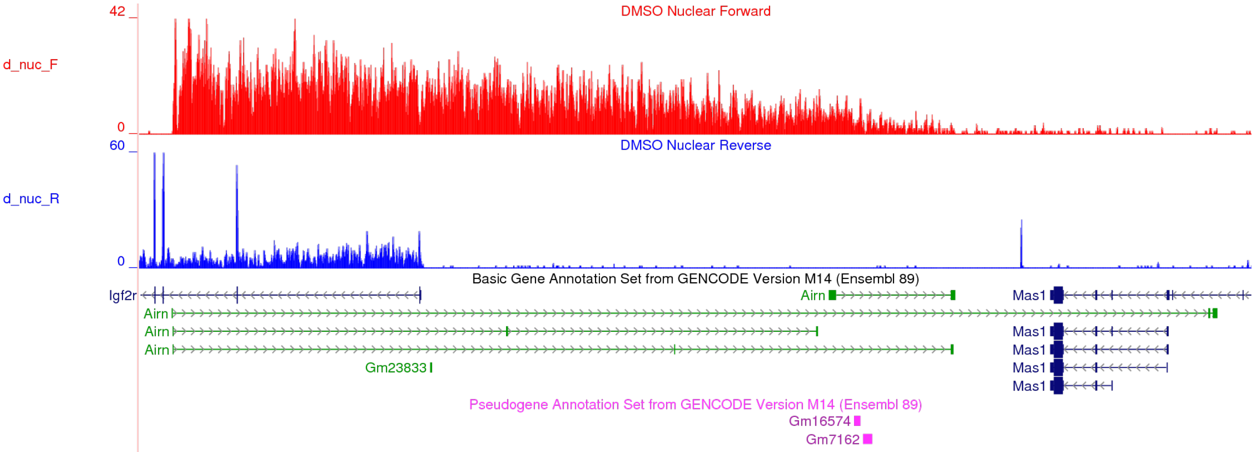


Figure 4.1 Airn provides an example of a misannotated unspliced transcript. Genome browser track of the mouse Airn, approximately at 13Mb on chromosome 17. Multiple GENCODE annotations of Airn are provided in green. All of them report Airn as a primarily intron containing RNA, whose spliced product is approximately 1Kb. The red and blue tracks labeled "d_nuc_F" and "d_nuc_R" are the forward and reverse aligned RNAseq reads for the nuclear extract of TSCs, respectively. The "d_nuc_F" clearly demonstrates reads aligning across the entire Airn locus, with little to no bias towards the GENCODE annotated bias. The spliced RNA Igf2r, ohe left side of the corresponding "d_nuc_R" track, provides an example of an accurately annotated spliced transcript.

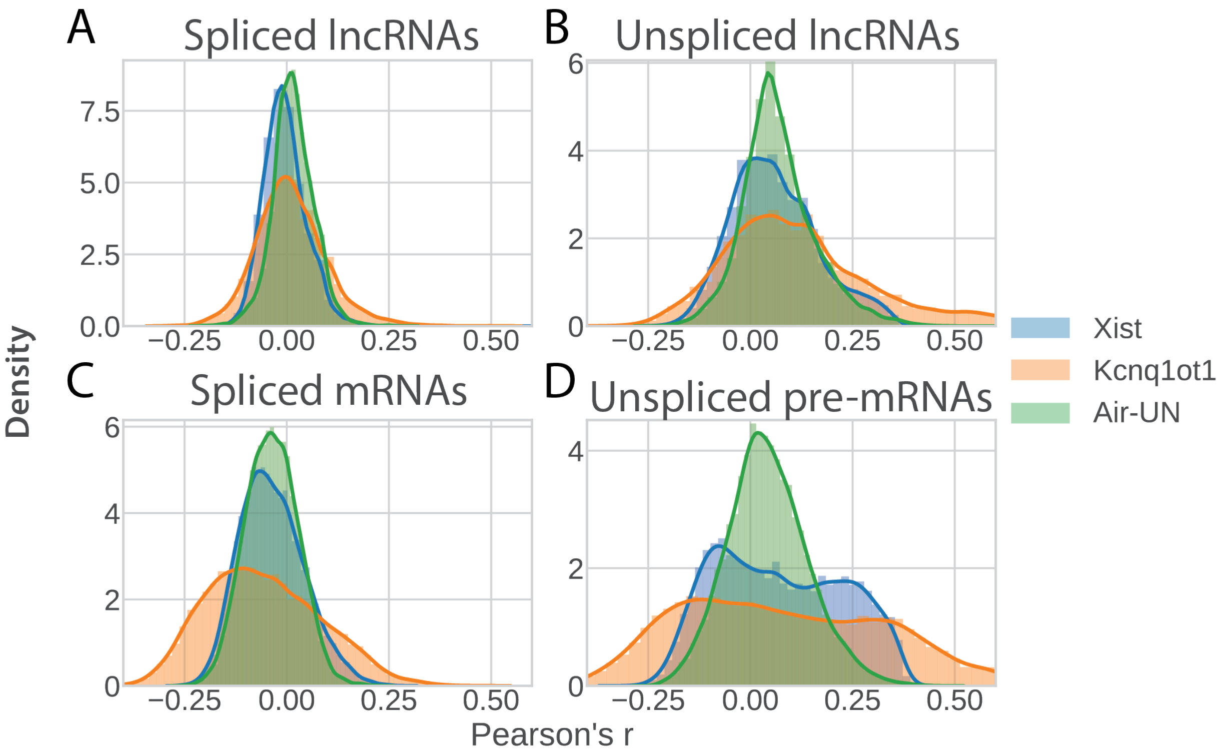


Figure 4.2 Repressive lncRNA SEEKR similarity to classes of RNA transcripts. (A) Distributions of Pearson's r-values between all mouse spliced lncRNAs and either Xist, Kcnq1ot1, or unspliced Airn are shown as normalized histograms. **(B-D)** The same as (A), but comparing the repressive lncRNAs to all unspliced lncRNAs, spliced mRNAs, and unspliced pre-mRNAs, respectively.

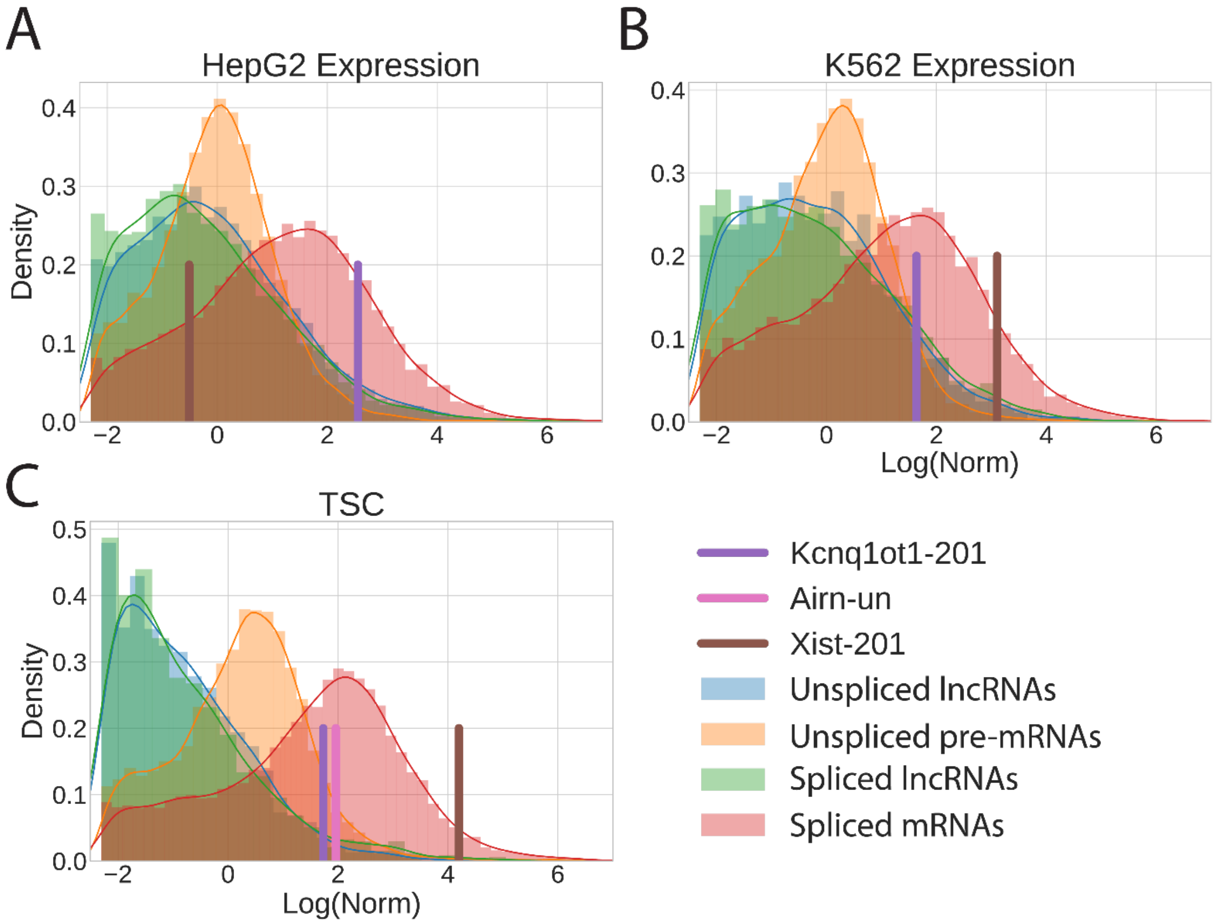


Figure 4.3 Expression levels of classes of RNAs. (A) Log normalized RPKM values of whole cell ENCODE RNAseq data in HepG2. Separate classes of RNAs are color coded as different distributions. As expected, spliced mRNA are the most abundant class of transcripts. Kcnq1ot1 and Xist RPKM values have been labeled. **(B)** The same as (A), but using RNAseq data in K562 cells. **(C)** The same as (A, B), but using RNAseq data in mouse TSCs from experiments performed by our lab. Additionally, unspliced Airn is labeled.

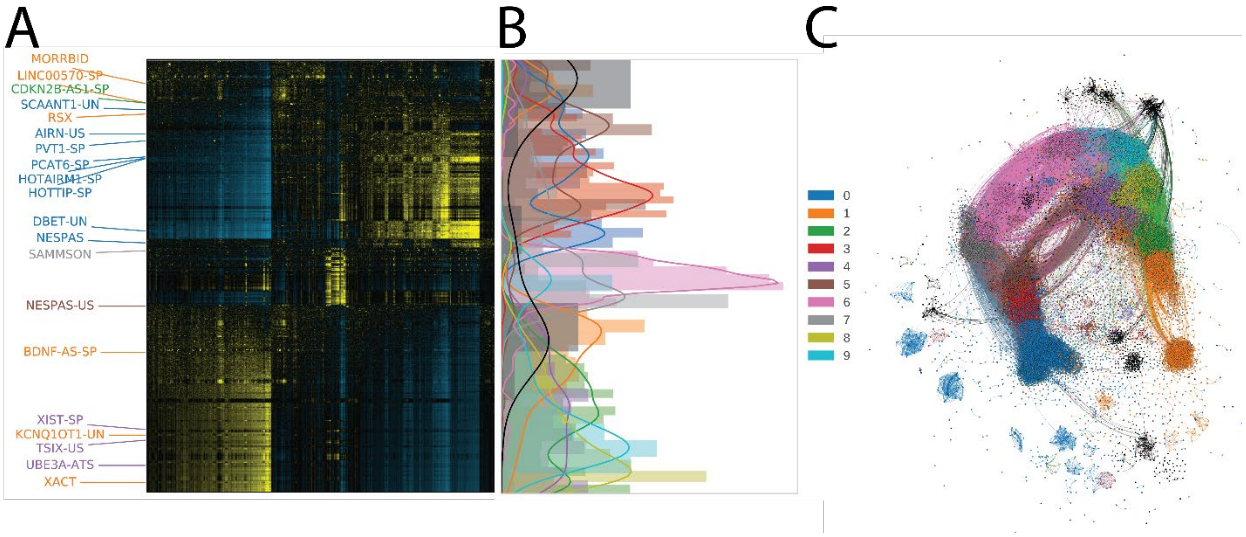


Figure 4.4 Communities and clusters of RNAs. (A) Hierarchical clustering of a random subsample of 45,000 transcripts. Positions of functional lncRNAs of interest have been labeled by their community color (see panel C). XIST is found in the AT-rich community #4. **(B)** Distributions of transcripts within each network defined community (panel C), relative to its y-axis position in the hierarchical cluster (panel A). **(C)** Network graph of Leiden defined communities. While all edges above the thresholds set by two-pass filter algorithm are used to calculate the layout, only edges above 0.5 are visualized.

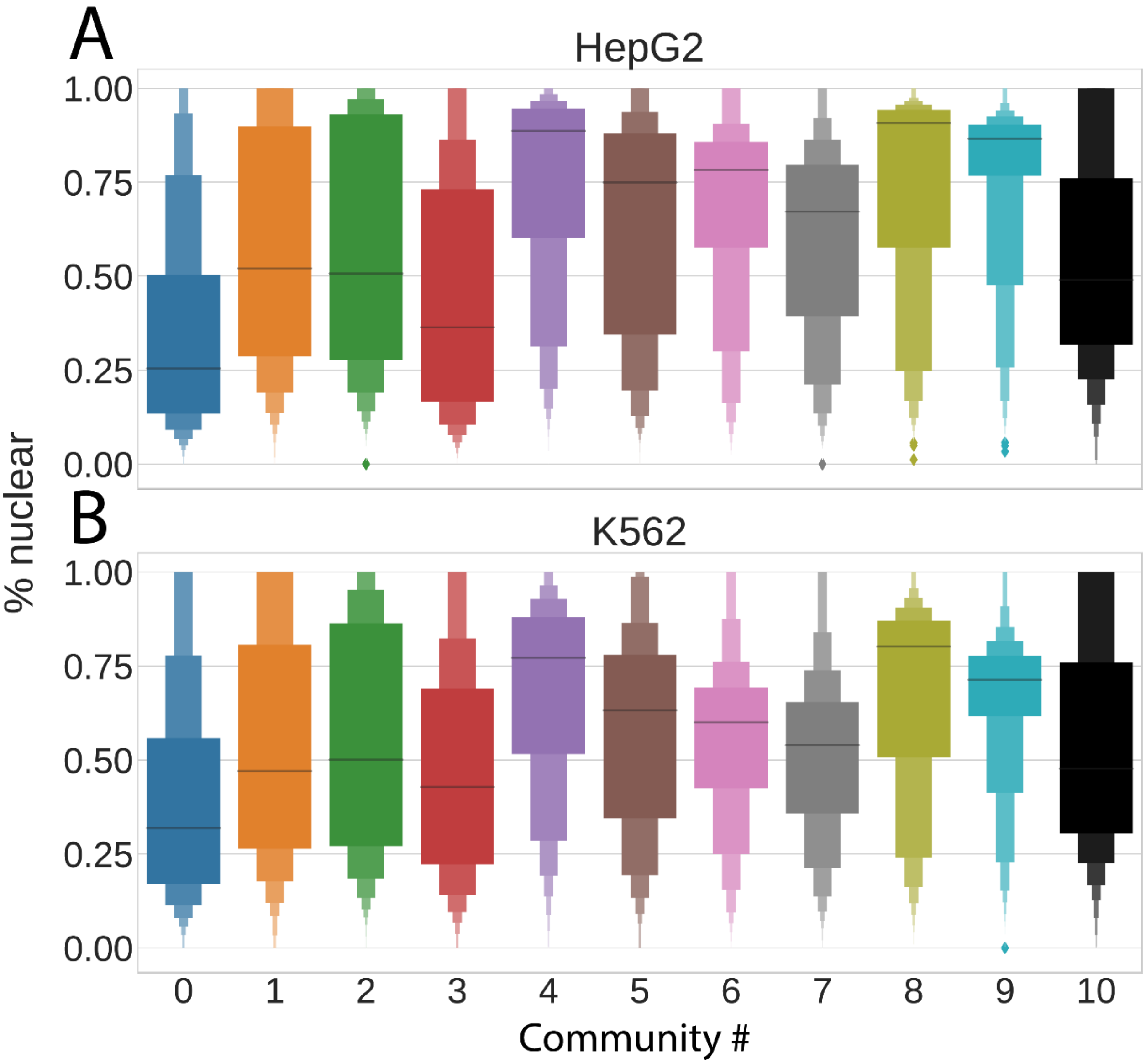


Figure 4.5 Cellular localization of RNAs. (A) Distributions of nuclear ratios for all transcripts within a given community for HepG2 cells. Distributions are visualized using Seaborn's "boxenplot" function. Any portion of the distribution below a y-axis "% nuclear" value indicates a transcript that is primarily cytosolic. **(B)** the same as (A), but using RNAseq data from K562 cells.

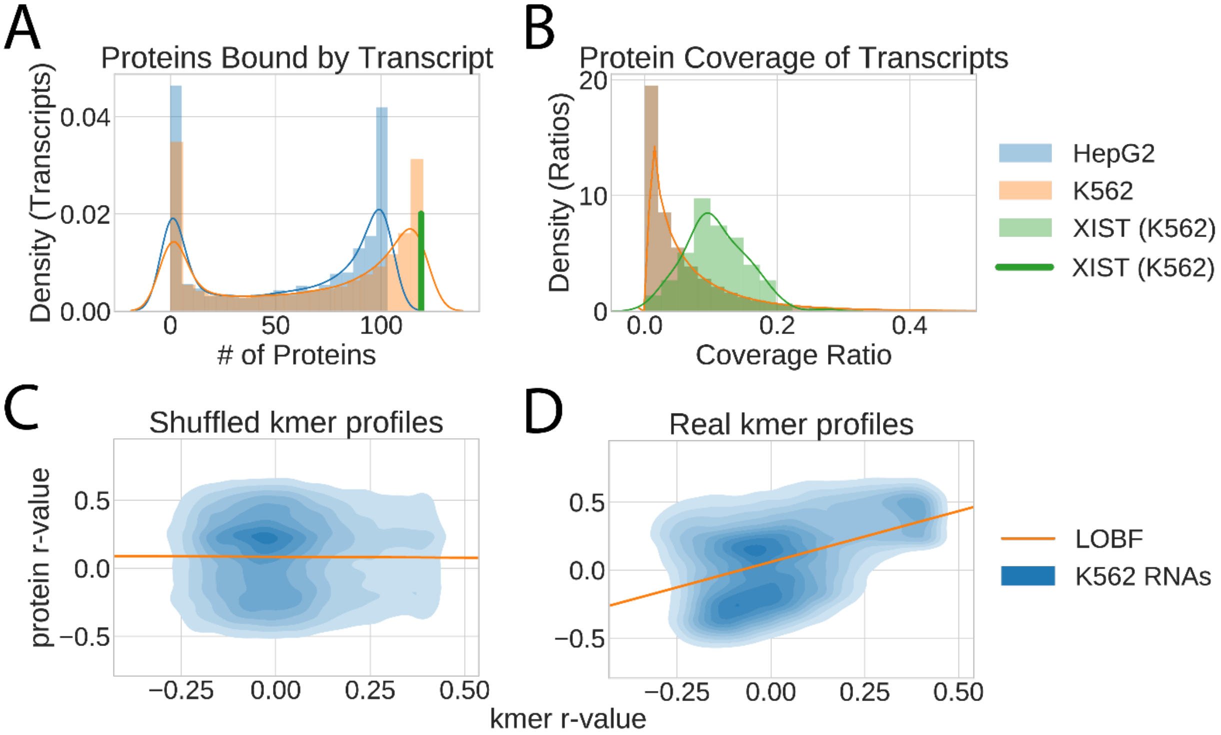


Figure 4.6 Protein binding profiles. (A) The number of proteins bound by each RNA in HepG2 and K562 cells**.** Most RNAs either bind many proteins, or (almost) none. XIST, labeled in green, binds 119 of 120 proteins in K562 cells. The right edge of the HepG2 distribution is farther to the left than the K562 distributions since there are fewer eCLIP data sets in HepG2. **(B)** Distributions of all coverage ratios (pre-normalization values) in HepG2 cells, K562 cells, and just within XIST. On average, a protein that binds XIST can bind a larger percentage of XIST than other RNA-protein interactions. HepG2's distribution is not clearly visible because it is nearly the same as the K562 distribution. **(C)** The correlation between each expressed K562 transcript's k-mer profile similarity to XIST's k-mer profile, and a random transcript's protein profile similarity to XIST's protein profile. As expected, there is no correlation. The blue is a 2 dimensional density estimate of all K562 transcripts. The orange line is the line of best fit. **(D)** The same as (C), but the k-mer profile r-values not been scrambled and are paired with their transcript's true protein binding profile r-value.

REFERENCES

1 Harrow, J. *et al.* GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Research* **22**, 1760–1774 (2012).

2 Derrien, T. *et al.* The GENCODE v7 catalog of human long noncoding RNAs: Analysis of their gene structure, evolution, and expression. *Genome Res.* **22**, 1775–1789 (2012).

3 Alpert, T., Herzel, L. & Neugebauer, K. M. Perfect timing: splicing and transcription rates in living cells. *Wiley Interdisciplinary Reviews: RNA* **8**, e1401 (2017).

4 Krämer, A. THE STRUCTURE AND FUNCTION OF PROTEINS INVOLVED IN MAMMALIAN PRE-mRNA SPLICING. *Annual Review of Biochemistry* **65**, 367–409 (1996).

5 Padgett, R. A., Grabowski, P. J., Konarska, M. M., Seiler, S. & Sharp, P. A. Splicing of Messenger Rna Precursors. *Annual Review of Biochemistry* **55**, 1119–1150 (1986).

6 Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods; New York* **14**, 417–419 (2017).

7 Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology* **34**, 525–527 (2016).

8 Nariai, N., Hirose, O., Kojima, K. & Nagasaki, M. TIGAR: transcript isoform abundance estimation method with gapped alignment of RNA-Seq data by variational Bayesian inference. *Bioinformatics* **29**, 2292–2299 (2013).

9 Traag, V. A., Waltman, L. & Eck, N. J. van. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports* **9**, 5233 (2019).

10 Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* **9**, 90–95 (2007).

CHAPTER 5

Conclusion and Future Directions

SEEKR[1] is a simple yet surprisingly effective method for predicting the biological properties of lncRNAs. It is fast: analyses are possible that would be either cumbersome or impossible with more sophisticated alignment algorithms. It's intuitive: each element in a kmer profile is directly tied to an understandable biological interpretation. Likewise, similarities between two transcripts are simply defined by the correlation between their kmer profiles. It's extensible: because of its other properties, modifying or building additional analyses off of the primary algorithm is straightforward. Together, these properties make SEEKR an extremely effective tool for studying lncRNAs, particularly in the context of tying sequence to function.

One of the difficulties—but also one of the most exciting opportunities—of this work is the novelty of the lncRNA field. Many possible analyses considered during the course of this work were curtailed by a lack of sufficiently diverse experimental data. In the future, it should be possible to use machine learning techniques to gain a deeper understanding of the relationship between sequence and function (similar to the logistic regression experiments detailed in chapter 2)[2,3]. Supervised techniques learning, however, needs labels which are time-consuming and expensive to gather. Despite the costs, the amount and types of publicly available data will continue to grow[4,5]. As data becomes available, there are a number of analyses which should be considered.

Chromatin modifications are an important hallmark of transcriptional regulation. In theory, it should be possible to uncover relationships between a lncRNA's sequence and its effects on nearby chromatin modifications such as H3K27me3. We attempted several related analyses; for example, we investigated if we could demonstrate a correlation between XISTlikeness and increased H3K27me3 levels in *cis*. However, we were unable to yield any conclusive results. Additional CHIP data, along with a larger set of conclusively known repressive lncRNAs, may be sufficient to link sequence elements to chromatin modifications.

lncRNAs are known to be cell-type specific[6]. It is reasonable to hypothesize that they are also cell specific and that their expression levels vary from cell to cell, even within a fairly homogenous cell population. If this is the case, single cell sequencing data will be an important future source of discoveries in the lncRNA field.

Beyond exploring new data, future directions of this work should include the development of SEEKR 2.0, a new algorithm to address some of the underlying assumptions of the original SEEKR model. In particular, the assumption that individual kmer function is completely independent of linear position or sequence context is clearly an over-simplification. We attempted to address this concern by developing what we called “context kmers”. When counting kmers in the original sequence, the local GC content just up- and downstream of the kmer was also computed. Each kmer was then grouped into one of four categories, based on the local GC content. Unfortunately, “context kmers” did not appear to provide additional predictive power in our analyses. Another potential avenue of interest is to create an algorithm that mixes an HMM algorithm like nhmmer with kmers. Details of such a model have not been worked out, but it is certainly important to begin to capture spatial relationships between kmers to understand how these interactions affect the overall biological properties of a lncRNA.

Providing evidence for the claims surrounding pre-mRNA regulator function presented in chapter 4 would be an exciting discovery for the entire genomics community. Despite a half century of extensive study of RNA biology, we were unable to find any published hypothesis, much less evidence, for the concept that the introns of mRNA play an important role in regulating the transcription of their neighboring genes. Conclusively demonstrating this effect in an endogenous setting may prove difficult, however. Attempting to overexpress the unspliced isoform must be decoupled from simultaneous overexpression of the spliced transcript. Treating cells with splicing inhibitors provides an avenue for modulating the ratio of spliced to unspliced product, but this treatment currently cannot be applied to a single gene of interest. On the other hand, while caveats would remain about the effect of the spliced isoform, demonstrating preliminary evidence for this effect should be a relatively straightforward set of experiments. To begin, an experimenter could select one (or more) of the genes corresponding to the unspliced pre-mRNA transcripts discussed in chapter 4, which have both XIST-like kmer and protein binding profiles. This transcript could be endogenously overexpressed, as well as repressed, perhaps using a system like a dCAS9. The expression levels of several nearby, expressed genes of interest could then be measured for downregulation and upregulation, respectively.

REFERENCES

1 Kirk, J.M. *et al.* Functional classification of long non-coding RNAs by k-mer content. *Nature Genetics* **50**, 1474-1482 (2018).

2 Yang, C. *et al.* LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning. *Bioinformatics* **34**, 3825–3834 (2018).

3 Gudenas, B. L. & Wang, L. Prediction of LncRNA Subcellular Localization with Deep Learning from Sequence Features. *Scientific Reports* **8**, 16385 (2018).

4 Lakiotaki, K., Vorniotakis, N., Tsagris, M., Georgakopoulos, G. & Tsamardinos, I. BioDataome: a collection of uniformly preprocessed and automatically annotated datasets for data-driven biology. *Database (Oxford)* **2018**, (2018).

5 Caswell, J. *et al.* Defending our public biological databases as a global critical infrastructure. *Frontiers in Bioengineering and Biotechnology* (2019).

6 Mattioli, K. *et al.* High-throughput functional analysis of lncRNA core promoters elucidates rules governing tissue specificity. *Genome Res.* **29**, 344–355 (2019).