Perturbation Analysis applied to an Artificial Neural Network Classifier of Long Non-Coding RNA

Robyn Ayscue[1], Jason Miller[1]
[1]Lane Engineering, West Virginia University

ABSTRACT

Classification of RNA sequence as protein coding RNA or long non-coding RNA is an important and vexing problem in biology. Various machine learning techniques have been applied to the problem, including artificial neural networks. Although neural networks achieve accuracy, their internal reasoning is difficult to characterize, especially for networks trained solely on raw sequence. Perturbation analysis was previously used to infer the features critical to one such classifier. To explore whether the approach can be applied more generally, perturbation analysis was applied to a different classifier whose features were known *a priori*. In this case study, perturbation analysis failed to confirm the *a priori* understanding of classifier internals and results were sensitive to the type of perturbation used. These results indicate that perturbation design is critical and that perturbation interpretation can be challenging.

KEYWORDS

LncRNA; neural network; perturbation analysis; process verification

INTRODUCTION

In 2001 (Venter *et al*. 2001; Lander *et al*. 2001), two papers announced resolution of the human genome sequence and the tens of thousands of protein-coding genes it encompassed. Shortly thereafter, bioinformatician Sean Eddy published a review calling for new tools for discovering and characterizing non-coding genes and non-coding RNA in the human genome (Eddy 2001). Many software tools are now capable of detecting and characterizing microRNA (Saçar Demirci *et al*. 2017), a form of non-coding RNA whose small size is dictated by its Dicer-mediated activity (Kurzynska-Kokorniak *et al*. 2015). Long non-coding RNAs, defined as having length>200 and abbreviated LncRNA, presumably act by different and varied mechanisms. Many human LncRNA genes were identified and characterized under the umbrella of the GenCode project (Jia *et al*. 2010; Derrien et al. 2012; Frankish *et al*. 2019). The current decade has seen the introduction of software tools for detection of LncRNA, including several based on artificial neural networks trained on the GenCode exemplars; mRNN (Hill *et al*. 2018) and LncADeep (Yang *et al*. 2018) are but two examples. Nevertheless, the classification of gene or RNA sequence as protein-coding or long non-coding remains an unsolved problem, even within the extensively studied human genome (Abascal *et al*. 2018).

The recently introduced mRNN (Hill *et al*. 2018) classifier uses a recurrent neural network trained on RNA sequences and their GenCode labels of coding or non-coding. The classifier does not incorporate any predefined features based on expert knowledge. In fact, it was created specifically to avoid bias that might be baked in to expert-based feature selection. The classifier performed reasonably well, *e.g.* above 90% accuracy, in three tests. Because the neural network was essentially a black box, its creators sought to infer the rules that the model had learned from the data. The authors applied perturbation analysis *i.e.* the systematic alteration and re-classification of inputs with the goal of identifying sequence elements that were required for the original classification. The authors devised three types of perturbation: interval shuffling, single point mutation, and pairwise point mutation. The study was limited to RNA sequences of 2000 letters or shorter. The point mutations included all three possible substitutions (RNA sequence is drawn from a 4-letter alphabet) but excluded insertion and deletion mutations. One conclusion of the study was that mRNN relied heavily on the CDS *i.e.* the central part of protein-coding RNA that includes a statistically unlikely sequence structure called an open reading frame (ORF).

The success in Hill *et al*. suggests that perturbation analysis might be an approach for deciphering the mechanisms inside a wide range of sequence classifiers. To inspire confidence in the method, it should be put on a theoretical foundation and validated. Theory could help practitioners identify the one permutation type, from the many possible, that is most appropriate for a particular scientific question. Theory could also help reduce the computational burden by narrowing all possible permutations to an informative subset of what are called adversarial permutations (Fawzi *et al.* 2017). And certainly, the approach should be validated on RNA sequence classifiers.

LncADeep (Yang *et al.* 2018) is one tool that could be used for validation of perturbation analysis. LncADeep incorporates separate models trained on human and mouse sequences. LncADeep bundles three types of model: binary classifiers of full-length RNA sequences; binary classifiers of partial RNA sequence; and multiclass classifiers of LncRNA function. The binary classifier of human full-length RNA uses a form of artificial network called a deep belief network (DBN). It operates, not on the raw RNA sequence, but on features extracted from the sequence. Its features encompass various measures of the protein coding potential of an RNA sequence, including especially recognition of CDS. LncADeep outputs a score that ranges from 1 for coding to 0 for non-coding.

In order to validate perturbation analysis as an approach for deciphering the internal mechanisms of neural network trained on RNA sequence, we attempted to reverse-engineer the LncADeep binary classifier via perturbation. We exploited our *a priori* knowledge of the features on which LncADeep was trained. We asked whether perturbation would be capable of recovering those features if those features had not been known in advance. We tested and compared two RNA sequence permutation types: systematic one-letter substitution and systematic one-letter deletion. We hypothesized that perturbation analysis would confirm that LncADeep classification relies on the CDS of protein-coding RNA.

METHODS

Reference sequences and annotation were obtained from GenCode (Frankish *et al*. 2019). Version 32 files were obtained by FTP. Two files were analyzed: protein-coding RNA sequence in FASTA format (gencode.v32.pc_transcripts.fa) and

RNA sequence annotation in GFF3 format (gencode.v32.basic.annotation.gff3).

The sequence data was modified to remove internal newlines. The sequence data was filtered to enforce sequence length constraints of 200 ≤ length ≤ 2500. The annotation data was filtered to discard comments and irrelevant fields and to use transcript rather than genome coordinates (Script: *Annotations.py*).

The LncADeep software, version 1.0, was downloaded from the public source code repository (https://github.com/cyang235/LncADeep). The software was configured and installed following instructions given in the repository, including installation of dependencies via the package manager *pip*. The software was installed on the Spruce Knob high-performance computing cluster at West Virginia University. LncADeep was run with Python version 2.7 as required, using the python/2.7.15_gcc82 module on Spruce Knob. The LncADeep command line was 'python LncADeep.py --MODE lncRNA --fasta $A --species human --model full --out $B' where variables $A and $B indicated the input and output files, respectively. Every LncADeep job operated on one FASTA file containing one original sequence plus mutants derived from it. All jobs were submitted to the PBS job control system on Spruce Knob using task ID as a parameter to identify each job's input FASTA file. (Script: *grid_classify.sh*). Primary outputs were written to the scratch file system but filtered outputs were copied to the user's home directory.

The perturbation process read original sequence from the RNA sequence file. For each input sequence, it generated one FASTA file containing the original sequence plus all mutants derived from it. The process was repeated for substitutions and deletions in different unix directories. Each substitution was generated per letter using the function **f**(x) = {**f**(A)=C, **f**(C)=G, **f**(G)=T, **f**(T)=A}. (Script: *Perturbation.py*). The pipeline was tested on the first input 21 sequences then run on the first 2001 sequences.

The primary outputs of LncADeep were processed to retain only information about the critical mutant i.e. the mutant with maximal score divergence from the original. For each input sequence, a script recorded the score and classification of the original sequence, plus the position, score, and classification of the critical mutation. (Script: *CriticalPosition.py*). In order to incorporate biological meaning, each critical position was labeled with every GenCode annotation that overlapped it. (Script: *Intersect.py*). Counts for **Table 3** and **Table 4** were accumulated using unix sort.

RESULTS: PIPELINE CONSTRUCTION

Over 100,000 protein-coding RNA sequences were downloaded from GenCode; see **Table 1**. The GenCode long non-coding RNA (LncRNA) sequences were not used in this study. The sequences were filtered to remove sequences shorter than the minimum length required to be classified as LncRNA, namely 200. The sequences were also filtered to remove sequences longer than 2500. The cutoff was based on preliminary analysis that indicated that longer sequences would consistently require run times in excess of 15 minutes (wall clock) per input sequence; see **Fig S1**. Finally, sequences with length 200-2500 were analyzed under a computational limit of 20 minutes each (wall clock).

Over 1.7 million lines of annotation were downloaded from GenCode. The annotations were filtered and translated from DNA to RNA coordinates. The annotation included labeled intervals that partitioned every transcript into exons, and every exon into coding (CDS) or non-coding (UTR); see **Fig 1**.

The LncADeep manuscript and source code were analyzed to make predictions about what kinds of mutations would have the most effect on its binary classification. Most of the LncADeep feature dimensions are formulated in terms of open reading frames; see **Table 2**. ORFs are intervals that begin with a start codon (ATG) and end with a stop codon (TAG, TGA, or TAA) between which there is an integer number of non-overlapping consecutive 3-mers called codons, none of which is a stop codon. ORFs span multiple exons if they are interrupted by genomic introns. In the GenCode annotation, the ORF portion of each exon is labeled CDS.

Since most of the LncADeep feature dimensions are consumed by ORF-related scores, we speculated that LncADeep would be more sensitive to mutations to coding sequence (CDS) than to mutations in the non-coding regions (UTR) of protein-coding transcripts. Bolstering the hypothesis, we observed that LncADeep's two UTR-related features are based on length not sequence. These are UTR length and UTR length as a portion of the transcript. Furthermore, LncADeep’s two remaining features, applied to the full-length transcripts, likely derive their signal from the CDS mostly. These are hexamer count and Fickett score which both rely on the notion of codon, which is only meaningful within CDS.

We next considered which types of perturbation to use. One-letter substitutions would have the chance to alter single codons within CDS. One-letter deletions would have the chance to disrupt the period-3 codon structure, potentially altering many codons and possibly truncating the CDS. Both of these permutation types were selected for this study.

Based on the observations above, we formulated these hypotheses.

Hypothesis 1:
Deletions will alter more LncADeep classifications than substitutions will.

Hypothesis 2:
Mutations to CDS will alter LncADeep scores more than mutations to UTR.

To perform this experiment, a perturbation pipeline was developed to mutate each sequence by one letter at a time; see **Fig 2**. The pipeline was run two times independently. One run applied a one-letter substitution at each position. The other run applied a one-letter deletion at each position. Every mutant was re-scored by LncADeep. The mutant whose score differed most from that of the original sequence was termed the critical mutant. The altered position of the critical mutant was called the critical position. Every critical position was mapped to either CDS or UTR using the GenCode reference annotation.

RESULTS: PERMUTATION ANALYSIS

Though the maximum score change was close to 1, most score changes were small; see **Fig S2**. All score changes that crossed the 0.5 boundary resulted in a flip from coding to non-coding or non-coding to coding.

Table 3 presents the number of critical mutants whose binary classification had flipped after the mutation. Regardless of direction of the flip, substitutions caused more classification flips than deletions. Thus, hypothesis 1 is not supported by these data.

Table 3 includes a surprisingly large number of flips from non-coding to coding. In nature, mutations that transform a non-coding sequence into coding are unlikely. Visual inspection of these cases indicated that most scores were close to the 0.5 boundary before and after the mutation. Furthermore, in all these cases, the original classification by LncADeep was contrary to the GenCode reference classification. Thus, these mutations had altered the sequence in a way that allowed LncADeep to improve its own marginally incorrect score.

Table 4 presents the number of critical positions that were in CDS, UTR, and other annotation categories. As a result of the substitution study, CDS mutations were overrepresented with respect to chance. In contrast, as a result of the deletion study, CDS mutations were underrepresented with respect to chance. The observed sensitivity of LncADeep to one-letter deletions of 3'UTR was not expected. The results of the substitution and deletion studies are mutually contradictory and so do not lend support to Hypothesis 2.

CONCLUSIONS

Perturbation had been shown to be informative when applied to the LncRNA classifier neural network mRNN (Hill *et al.* 2018). Perturbation analysis depends on the assumption that a mutation that alters a classification indicates a letter or position that was important to the classifier. We sought to validate the perturbation approach by applying it to a different classifier, LncADeep (Yang *et al.* 2018), for which *a priori* knowledge was available. LncADeep had been trained on 45 specific sequence features that it extracts from RNA sequence. Since the vast majority of input features related to CDS, and since the UTR-based features measured length only, we hypothesized that perturbation of CDS would be more disruptive to LncADeep than perturbations to UTR. Further, since CDS requires a period-3 codon structure, we hypothesized that one-letter deletions would be more likely to disrupt CDS and have more effect on LncADeep, compared to one-letter substitutions.

Our results did not support, and even contraindicated, our hypotheses. We conclude that perturbation did not provide the expected insight to the nature of the neural network under study. More generally, we also conclude that interpretation of perturbation results can be challenging. Our perturbation analysis failed to confirm what we already knew (or thought we knew) about a particular classifier. It is possible that our understanding of the classifier was flawed. It is possible that our study was confounded by its small sample size. Nevertheless, the results indicate that perturbation analysis can be difficult to interpret and may not be informative in all situations.

In our study, CDS substitutions were more disruptive than UTR substitutions. However, mutations implemented as deletions generated opposite results. Had we not not performed the deletion analysis, we might have concluded that our permutation analysis by substitutions had confirmed the importance of CDS to LncADeep. Thus, our perturbation analysis was sensitive to the type of perturbation employed, and this sensitivity might generalize. It is notable that the Hill *et al*. study used substitutions only. Since many types of perturbation are possible, theoretical work is warranted to guide the proper selection of perturbation type for future perturbation studies.

In retrospect, we can state several reasons to be suspicious of the perturbation approach. Perturbation alters the raw data. Perturbation generates artificial sequences for which no classifier has been trained. It is unclear what question is being framed when a natural-sequence classifier is forced to classify artificial sequences. If deletion mutants can be considered more unnatural and substitution mutants, this may explain the unexpected outcome of deletion mutagenesis in our study.

LIMITATIONS AND FUTURE WORK

Our study incorporated several limitations. Our study of substitutions explored only one of three possibilities per letter, given the 4-letter alphabet of RNA. Our study did not explore insertions. Our study only examined one-letter mutations. Our study was terminated after processing 2000 sequences (2% of the data), so small sample size could be a factor. Our study analyzed only one classifier, LncADeep. Our evaluations relied on the CDS/UTR boundary provided by GenCode; it is possible that LncADeep computed different boundaries.

Our study was slightly compromised by the fact that the reference annotations file did not contain annotation for every given sequence. The fact was realized too late to be incorporated into the experimental design. In future studies, we would limit the test set to protein-coding sequences for which annotation is at hand.

Although LncADeep was designed to distinguish coding from non-coding RNA, our study only tested the coding variety because little is known about the structure or function on LncRNA. LncRNA is probably a heterogeneous category deserving of subcategorization. Machine learning may play an important role there, and perturbation analysis may help decipher those models.

The findings and limitations of our study suggest directions for future work on analysis of neural networks. Gentler approaches could replace permutation of the data. For example, a black box classifier could be presented with different categories of natural sequence. To explore the hypothesis that long CDS is critical to the classifier, the input sequences could be partitioned by length of CDS. With large enough samples, and unless other features are dependent on CDS length, the classifier's reliance on CDS length would be elucidated by comparison of its average scores of the various partitions.

CODE AVAILABILITY

Python and bash scripts written for this study are available online (*https://github.com/JasonRafeMiller/LncRNA*).

TABLES AND FIGURES

File	Contents	# Sequences	Metrics
pcRNA.fasta	Protein coding	100,291	2186 avg len
max2500.min200.pc	Our size filters	69,738	1124 avg len
basic.annotation.gff3	Labeled Intervals	108,610	16.3 avg #labels

Table 1. Input data characteristics. Reference data were obtained from GenCode (black text). The sequence data was filtered (blue text) to enforce sequence length constraints of 200 ≤ length ≤ 2500. The annotation data was filtered to retain relevant fields and modified to include transcript coordinates as well as given genomic coordinates (not shown).

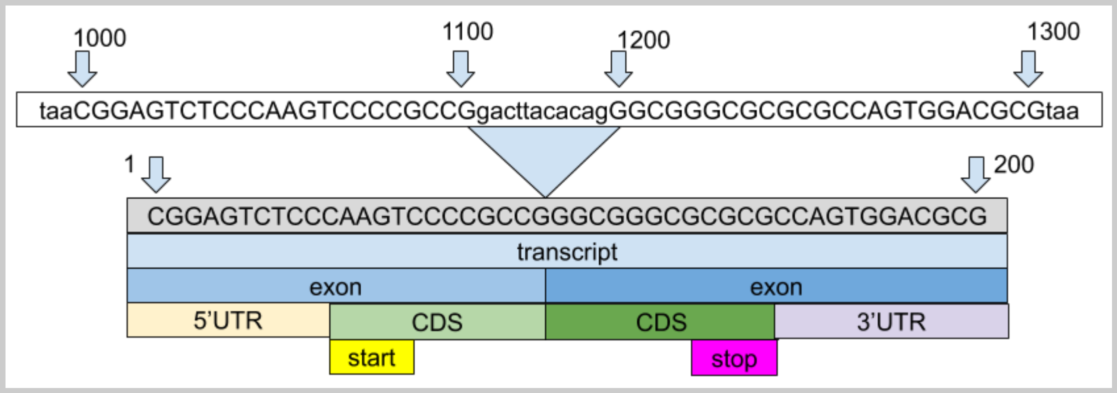
1000
1100
1200
1300
taaCGGAGTCTCCCAAGTCCCCGCCGgacttacacagGGCGGGCGCGCGCCAGTGGACGCGtaa
1
200
CGGAGTCTCCCAAGTCCCCGCCGGGCGGGCGCGCGCCAGTGGACGCG
transcript
exon
exon
5'UTR
CDS
CDS
3'UTR
start
stop

Fig 1. Schematic of the input GenCode annotation. The GFF file from GenCode provided a hierarchy of intervals (colored boxes). The given chromosome coordinates (white box) span genomic sequence (lower case) that is not included in the given transcript sequence (grey box). Therefore, the annotation coordinates were translated to transcript coordinates as slown. Each transcript has a central coding region (green) that spans one to many exons (blue). Each coding sequence (CDS) includes a start and stop codon. Each CDS has flanking untranslated sequence (UTR) that is non-coding. When present, CDS and UTR provide complete, non-overlapping coverage of the transcript. The hypothetical transcript depicted here derives from a forward-strand gene. The relative positions of 5' and 3' UTR, and of the start and stop codons, would be reversed on reverse-strand genes.

Feature type	#	Interpretation	Category
ORF len, ORF coverage	2	CDS length and % of transcript	CDS
EDP of ORF	36	Entropy of protein coding K-mer frequencies	CDS
HMMer score	3	Similarity to model of any protein family	CDS
UTR len, UTR coverage	2	Non-coding portion of transcript	UTR
Mean hexamer score	1	K-mer frequency	both
Fickett score	1	Letter frequency per codon position	both

Table 2. Analysis of LncADeep features. The six feature types used by the LncADeep model for binary classification of full-length transcripts [Yang *et al.*, supplement page 9] are listed by type and dimensionality (columns 1 and 2). An interpretation and category were assigned to each type (columns 3 and 4). According to this analysis, three feature types that operate exclusively on the inferred coding sequence (CDS) account for 41 dimensions of the feature space, while the one feature type that operates

exclusively on the inferred non-coding sequence (UTR) contributes only 2 dimensions. Furthermore, the two remaining dimensions belong to features that apply to both CDS and UTR though their signal strength is likely derived from the CDS. This analysis inspired the hypothesis that LncADeep classification would be more sensitive to mutations in CDS than in UTR.

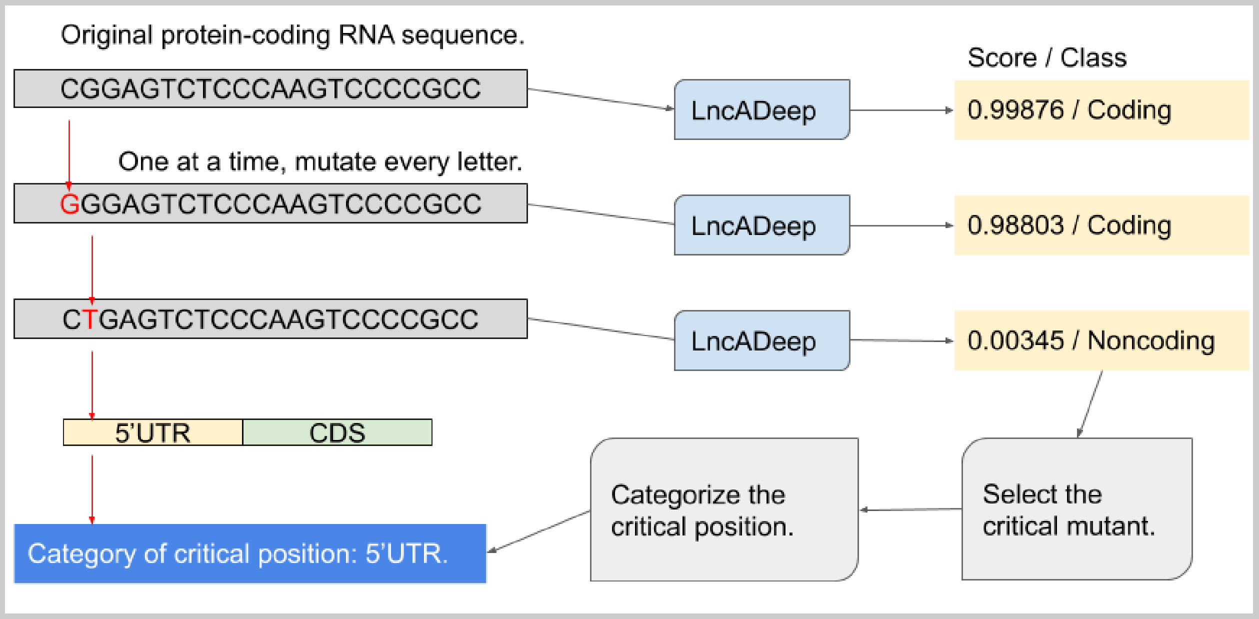
Original protein-coding RNA sequence.
CGGAGTCTCCCAAGTCCCCGCC
Score / Class
LncADeep
0.99876 / Coding
One at a time, mutate every letter.
GGGAGTCTCCCAAGTCCCCGCC
LncADeep
0.98803 / Coding
CTGAGTCTCCCAAGTCCCCGCC
LncADeep
0.00345 / Noncoding
5'UTR
CDS
Categorize the critical position.
Select the critical mutant.
Category of critical position: 5'UTR.

Fig 2. Schematic of the perturbation analysis pipeline. Every input sequence was altered by one-letter substitution. Every altered sequence was analyzed by the LncADeep artificial neural network to yield a score and classification. For the alteration yielding the maximal score, the altered position was characterized according to functional labels provided by GenCode. All sequences and scores in the figure are hypothetical, and for clarity, only two sequence alterations are depicted. A similar pipeline that used one-letter deletions was run independently (not shown).

Source:	Reference	Classifier	Classifier
Classification:	coding	coding→noncoding	noncoding→coding
substitution	1989	688 (35%)	472 (24%)
deletion	1990	53 (3%)	2 (0%)

Table 3. Effect of substitution and deletion on mutant classification. The first 2000 filtered protein coding sequences were subjected to perturbation and reclassification. These were classified by the LncADeep classifier before and after two forms of alteration: substitution and deletion. For each original sequence, the critical mutant was determined as the altered version that yielded the maximum score change. Critical mutants whose class had flipped were binned according to the direction of the flip: mutant-of-coding was non-coding (3rd column), or mutant-of-non-coding was coding (4th column). These results indicate that the classifier was more sensitive to one-letter substitutions than to one-letter deletions.

Annotation	Sequences With Critical Substitutions	% of exon	Sequences With Critical Deletions	% of exon
None	490		489	
transcript & exon	499		502	
CDS	445	89.2	234	46.6
start codon	8	1.6	1	0.2
stop codon	24	4.8	19	3.8
5’ UTR	4	0.8	9	1.8
3’ UTR	29	5.8	259	51.6
total UTR	33	6.6	268	53.4

Table 4. Effect of substitutions and deletions on CDS vs UTR. The first 2000 filtered protein coding sequences were subjected to perturbation and reclassification. For every input sequence, the critical position was determined as the position whose mutation yielded the maximum score change. Each critical position was characterized by its annotation interval, if one existed. By their definitions, transcript and exon counts were equal to each other and to the sum of the CDS plus total UTR. As reported above, the CDS:UTR ratio differed by mutation type. Both observed ratios, 89:7 for substitutions and 47:53 for deletions, differ from the expected but in opposite directions. The expected ratio was 59:41 based on the sums of CDS and UTR positions in the sequences analyzed. Both deviations from expectation are statistically significant by the binomial formula; the probability of seeing either observed ratio, or a more extreme one, by chance is 10^{-6}.

APPENDIX - SUPPLEMENTAL FIGURES

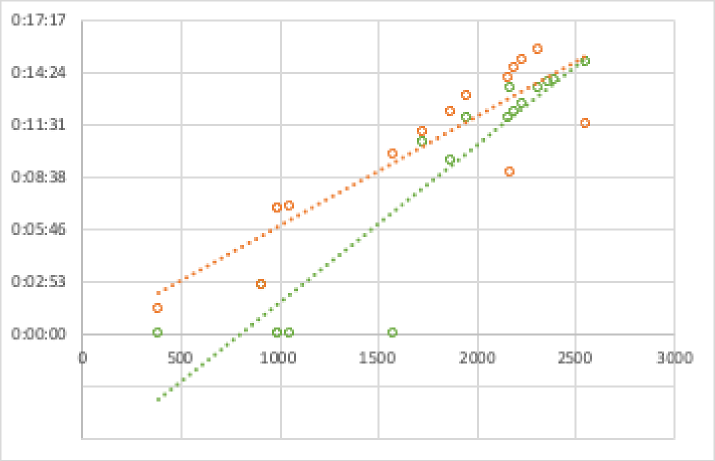

Fig S1. Effect of sequence length on mutant classification runtime. The classifier LncADeep was run as a job array on Spruce Knob. The scatter plot places each job according to the length of its original sequence (horizontal axis, number of letters) and run time (vertical axis, h:mm:ss). Each job classified one original sequence plus its mutants created by substitution (green) or deletion (orange). Linear regression was employed (dotted lines) based on the expectation that runtime would scale linearly with the number of mutants to be classified, which would scale linearly with the original sequence length. Based on this preliminary analysis of 21 sequences, the input date were filtered to remove sequences with length > 2500 and computations were restricted runs to 20 min wall clock per original sequence.

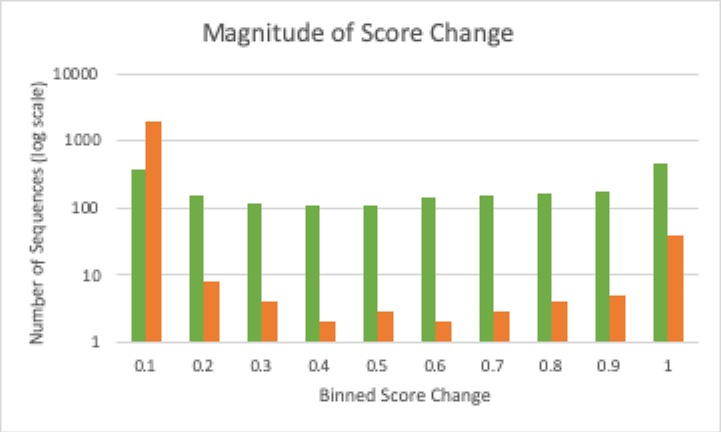

Fig S2. Histogram of score change. The classifier LncADeep re-scored every mutant of every input sequence. The scores range from 0 to 1. For each input sequence, max(abs(mutantScore-originalScore)) was accumulated in 10 bins. Counts of substitution (green) and deletion (orange) perturbations were plotted on a log scale. The majority of sequences experienced a maximum score change of 0.1 or less.

Bibliography

Abascal, F., Juan, D., Jungreis, I., et al. 2018. Corrigendum: Loose ends: almost one in five human genes still have unresolved coding status. *Nucleic Acids Research* 46(22), p. 12194.

Derrien, T., Johnson, R., Bussotti, G., et al. 2012. The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. *Genome Research* 22(9), pp. 1775–1789.

Eddy, S.R. 2001. Non-coding RNA genes and the modern RNA world. *Nature Reviews. Genetics* 2(12), pp. 919–929.

Fawzi, A., Fawzi, O. and Frossard, P. 2017. Analysis of classifiers' robustness to adversarial perturbations. *Machine learning* 107(3), pp. 1–28.

Frankish, A., Diekhans, M., Ferreira, A.-M., et al. 2019. GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Research* 47(D1), pp. D766–D773.

Hill, S.T., Kuintzle, R., Teegarden, A., Merrill, E., Danaee, P. and Hendrix, D.A. 2018. A deep recurrent neural network discovers complex biological rules to decipher RNA protein-coding potential. *Nucleic Acids Research* 46(16), pp. 8105–8113.

Jia, H., Osak, M., Bogu, G.K., Stanton, L.W., Johnson, R. and Lipovich, L. 2010. Genome-wide computational identification and manual annotation of human long noncoding RNA genes. *RNA (New York)* 16(8), pp. 1478–1487.

Kurzynska-Kokorniak, A., Koralewska, N., Pokornowska, M., et al. 2015. The many faces of Dicer: the complexity of the mechanisms regulating Dicer gene expression and enzyme activities. *Nucleic Acids Research* 43(9), pp. 4365–4380.

Lander, E.S., Linton, L.M., Birren, B., et al. 2001. Initial sequencing and analysis of the human genome. *Nature* 409(6822), pp. 860–921.

Saçar Demirci, M.D., Baumbach, J. and Allmer, J. 2017. On the performance of pre-microRNA detection algorithms. *Nature Communications* 8(1), p. 330.

Venter, J.C., Adams, M.D., Myers, E.W., et al. 2001. The sequence of the human genome. *Science* 291(5507), pp. 1304–1351.

Yang, C., Yang, L., Zhou, M., et al. 2018. LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning. *Bioinformatics* 34(22), pp. 3825–3834.