Briefings in Bioinformatics, 22(6), 2021, 1–18

https://doi.org/10.1093/bib/bbab285
Review

A systematic evaluation of the computational tools for lncRNA identification

Hansi Zheng†, Amlan Talukder†, Xiaoman Li and Haiyan Hu

Corresponding authors: Xiaoman Li, Burnett School of Biomedical Science, College of Medicine, University of Central Florida, Orlando, FL, USA. E-mail: xiaoman@mail.ucf.edu; Haiyan Hu, Department of Computer Science, University of Central Florida, Orlando, FL, USA. E-mail: haihu@cs.ucf.edu
†These authors contributed equally to this work.

Abstract

The computational identification of long non-coding RNAs (lncRNAs) is important to study lncRNAs and their functions. Despite the existence of many computation tools for lncRNA identification, to our knowledge, there is no systematic evaluation of these tools on common datasets and no consensus regarding their performance and the importance of the features used. To fill this gap, in this study, we assessed the performance of 17 tools on several common datasets. We also investigated the importance of the features used by the tools. We found that the deep learning-based tools have the best performance in terms of identifying lncRNAs, and the peptide features do not contribute much to the tool accuracy. Moreover, when the transcripts in a cell type were considered, the performance of all tools significantly dropped, and the deep learning-based tools were no longer as good as other tools. Our study will serve as an excellent starting point for selecting tools and features for lncRNA identification.

Key words: lncRNAs; lncRNA identification; lncRNA features; lncRNA prediction tools

Introduction

Only about 1.5% of the human genome is transcribed into messenger RNAs (mRNAs), which can further be translated into proteins. A vast proportion (>90%) of the genome is transcribed into non-coding RNAs (ncRNAs) and do not possess protein-coding ability [1–5]. For a long period, the non-coding part of the genome is usually regarded as the 'dark matter' of the genome and the resulting ncRNAs are ignored from further studies. With the development of high-throughput technologies, a variety of functions of the ncRNAs, especially the long ncRNAs (lncRNAs), are now being revealed in numerous biological processes, such as gene regulation, gene silencing and RNA modification [6–13].

To further understand the properties and functions of lncRNAs, efficient identification of lncRNA transcripts is essential [14]. The most common definition of lncRNAs so far is the ncRNA transcripts longer than 200 nucleotides (nt). Because of their longer sizes, which are similar to those of mRNAs, distinguishing lncRNA transcripts from mRNAs efficiently has remained a challenging task. LncRNAs are just like mRNAs, as both are transcribed by RNA polymerase II from the genomic loci with similar chromatin states [15]. However, they can have delicate differences. For instance, lncRNAs tend to be shorter than mRNAs; lncRNAs may also have fewer but longer exons, lower level of expression, shorter open reading frames (ORFs), etc. Note that all characteristics mentioned above have exceptions [16, 17]. For instance, the lncRNA NEAT1, which plays an important role in various biological processes, has a single exon of 227 000 nt in length, longer than most mRNAs [18, 19]. Also, some well-studied lncRNAs, including HOTAIR, MALAT1, ANRIL and NEAT1,

Hansi Zheng is a graduate student from Department of Computer Science, University of Central Florida. He mainly works on non-coding RNAs and epigenomics.
Amlan Talukder is a graduate student from Department of Computer Science, University of Central Florida. He mainly works on miRNAs and epigenomics.
Xiaoman Li is an associate professor from Burnett School of Biomedical Science, University of Central Florida. He works on chromatin interactions and metagenomics.
Haiyan Hu is an associate professor from Department of Computer Science, University of Central Florida. She works on miRNAs, epigenomics and gene transcriptional regulation.
Submitted: 12 May 2021; **Received (in revised form):** 21 June 2021

are found to be highly expressed in different types of cancer cells [20–28].

The first discovered eukaryotic lncRNA, H19, was identified in mouse in 1984 [22]. Later, the discovery of the pervasive transcription phenomenon revealed thousands of lncRNAs such as Xist, Airn, MALAT1 and HOTAIR in animals [29, 30]. Studies found that lncRNAs play roles in gene expression regulation during both developmental and differentiation processes. Also, developmentally complex organisms tend to have a higher number of lncRNAs, indicating the importance of lncRNAs in multicellular development processes [30]. LncRNAs were found to act both in *cis* by regulating the expression of neighboring coding genes [31] and in *trans* by regulating the expression of distant genes [32, 33]. Based on their functionalities, lncRNAs are also categorized into three major groups, where they regulate chromatin states and gene expression at regions distant from their transcription sites, influence nuclear structure and organization and regulate the behavior of proteins and/or other RNA molecules [34].

A variety of experimental approaches have been in practice to identify lncRNAs. Microarrays and RNA-seq can perform high-throughput analysis of lncRNA expression [35, 36]. Northern blots, reverse transcription polymerase chain reaction (RT-PCR), fluorescence in situ hybridization (FISH) and RNA interference (RNAi) are used to verify the authenticity of high-throughput data [37–39]. RNA pull-down assay, RIP-chip/seq and CLIP are used to identify lncRNA–protein interactions [40]. Conducting such experimental methods in diverse cell-specific conditions can be time-consuming and expensive. Hence, in recent years, various computational methods have been introduced that take advantage of these technologies and the increased computational power to identify lncRNAs. Efficient identification of lncRNAs by computational methods minimizes the need for experimental identification in many cases.

Dozens of lncRNA prediction tools are available that are built on the machine learning and deep learning technologies and utilize a variety of features. Although these computational methods have yielded encouraging results to identify lncRNAs, to our knowledge, they have not been systematically evaluated, especially the recently developed methods and tools [41–52]. To date, there is no consensus regarding the performance of these tools. To fill this void, we evaluated the following 17 tools published after 2012: CPAT, CNCI, PLEK, FEELnc, CPC2, lncRNAnet, CPPred, LGC, lncFinder, lncRNA_Mdeep, CNIT, CREMA, LncADeep, Lncident and PredLnc-GFStack [53–69]. We compared their efficiency and accuracy and studied the importance of the features used by these tools for lncRNA identification. Our study shed new light on the study of lncRNAs and their functions.

Features commonly used in lncRNA identification

The computational problem to identify lncRNAs is to essentially distinguish lncRNA transcripts apart from the mRNA transcripts, given the corresponding sequences. The features that can be derived directly from the sequences have thus been popular from the beginning. Many efforts have been made so far to engineer the features from sequences to allow a better understanding of the lncRNA transcripts and thus improve the lncRNA prediction performance [57, 62, 63].

Since ORFs have a strong correlation with the coding potential, several features were introduced based on ORFs, such as the ORF size, the ORF integrity and the ORF coverage [53, 54, 56–58, 60, 61, 63–67, 69]. Deep learning models were also used to model ORFs, which serve as an ORF indicator to improve the overall performance [53]. The *k*-mer (*k* nt long DNA or RNA segment) based features can also be directly derived from sequences [62, 63, 65, 70–72]. Along with the standard *k*-mer profiles, several other features were engineered from the *k*-mers that can better explain the coding potential of a sequence, such as the Fickett TESTCODE score, the hexamer score, the adjoining nucleotide triplet (ANT) frequencies, the multi *k*-mer frequencies, and the composition, transition and distribution (CTD) features [41, 54–56, 59–63, 65–67, 69]. There were some attempts to model the relationship between the ORF size and the GC content as well [60, 61, 64, 69]. While investigating the lncRNA functions, studies revealed that certain lncRNAs with short ORFs (ORF <300 nt) can still code for short peptides, which have certain key biological functions [60, 73–84]. These studies led to the features that focus on the formation and stability of the resulted peptides, such as the isoelectric point, instability index, grand average hydropathicity and electron–ion interaction pseudo-potential (EIIP) [57, 58, 60, 63, 69]. The isoelectric point denotes the pH at which a peptide molecule does not carry any electrical charge [64]. The instability index feature measures the stability of a peptide, which can be calculated for a simulated test tube atmosphere. The grand average hydropathicity tells about the hydrophobicity of the protein molecule. The emerging powerful deep learning models can capture more complicated patterns directly from sequences; hence the sequences themselves are also encoded as input features [53, 54]. In the following, we discuss the features used in the published tools under three categories: ORF features, *k*-mer features and peptide-level features.

ORF features

An ORF consists of a set of consecutive non-overlapping codons that can be translated into a protein. An ORF is bounded by a start codon (AUG) and a stop codon (UAA, UAG, UGA). The translation starts from the start codon and stops after the stop codon. The high correlation between ORFs and the ability of protein translation makes ORFs one of the popular features that can separate lncRNAs from mRNAs. The ORF size is another fundamental feature that is commonly used to distinguish lncRNAs from mRNAs. Sometimes the ORF size is used indirectly, such as the ORF coverage, which is defined as the ratio of the ORF size to the total transcript length. Note that, considering the ORF size feature as a sole ORF feature may misclassify lncRNAs containing large ORFs into mRNAs [16]. Hence the ORF integrity is introduced as a stand-alone feature to make up the deficiency of the ORF size. The ORF integrity is a Boolean value feature, which tells where the ORF starts and ends. Sometimes an ORF contains non-canonical (non-AUG) start codons, which makes it difficult to correctly identify the exact start position of the ORF. Therefore, instead of considering the start and stop codons to define an ORF, some studies use the stop-to-stop ORFs to model lncRNAs [53] (Table 1).

K-mer features

The composition of the DNA sequences is fundamental to distinguish lncRNAs from mRNAs. Other than the ORF features that can be extracted from the DNA sequences, there are some efforts to extract more useful composition information from the sequences (Table 1). Most of these involve the frequency profiles of *k*-mers with different values of *k*.

Several studies applied *k*-mer frequencies as the direct features [59, 66], while others used these features to design more

Table 1. Machine learning models and features used by the tools

Tools	Classifier	ORF size	Peptide features	*k*-mer profile	Other features
CPAT	LR	ORF size, ORF coverage		Fickett score, hexamer score	
CNCI	SVM			ANT frequency matrix	
PLEK	SVM			Frequency of *k*-mer ($k = 1$–5) patterns	
FEELnc	RF	ORF coverage		Multi *k*-mer frequencies	Sequence length
CPC2	SVM	ORF size, ORF integrity	Isoelectric point	Fickett score	
lncRNAnet	CNN + RNN	ORF size, ORF coverage, ORF indicator			Sequence profile
CPPred	SVM	ORF size, ORF coverage, ORF integrity	Isoelectric point, instability index, Gravy	Fickett score, hexamer score, CTD	
LGC	MLE	ORF size			GC content
lncFinder	SVM	ORF size, ORF coverage	EIIP	Distance between hexamer frequencies of a new transcript and lncRNA transcripts	RNA secondary structure features
lncRNA_Mdeep	DNN + CNN	ORF size, ORF coverage		Fickett score, hexamer score, *k*-mer frequency	Sequence profile
CNIT	XGBoost			ANT frequency	
CREMA	LR stacked on the GB models	ORF size		Fickett score, hexamer score	GC content, transcript length, alignment identity, alignment length, alignment length:transcript length, alignment length:ORF length
LncADeep	DBN	ORF size, ORF coverage, EDP of ORF		Fickett score, hexamer score, EDP of 3-mer from 7 amino acid (codons) groups, EDP of LCDS	HMMER index
Lncident	SVM	ORF size, ORF coverage		Adjoining *k* nucleotide frequency in ORF	
PredLnc-GFStack	Ensemble of RF	ORF size, ORF coverage, ORF integrity, EDP of ORF	Isoelectric point, instability index, Gravy	Fickett score, hexamer score, EDP of transcript, CTD	GC content
BASiNET	Decision tree			Adjacency of *k*-mers	
NCResNet	ResNet	ORF size, ORF coverage, ORF integrity	Isoelectric point, instability index, Gravy, molecular weight, EIIP	Fickett score, hexamer score, CTD, codon number, codon ratio	GC content, GC variance

Here, Gravy is the short form of grand average hydropathicity.

sophisticated features [53, 57, 62, 63, 65]. Incorporating the idea of codon and *k*-mer frequencies, CNCI uses the ANT profiles to generate the 'most-like' coding domain sequence region feature, which represents the sub-sequences that have the most ability to code [62]. CPAT and CPPred use the hexamer score, which bears almost the same idea as the ANT profiles except that they score a test sequence by using a pre-calculated log-likelihood ratio of the hexamer scores from the coding and non-coding training sequences. The positive hexamer score represents a mRNA, and the negative score corresponds to an lncRNA [63, 65]. LncFinder considers the Euclidean distances between the hexamer score of a query transcript with the hexamer scores of the lncRNAs and the mRNAs in the training data, then uses the ratio between these two distances to identify the lncRNAs [57]. CPPred, PredLnc-GFStack and NCResNet use a set of features named CTD that record the individual frequency of the 4 nt (Composition), the number of transitions from one nucleotide type to another (Transition) and the occurrence locations of

0%, 25%, 50%, 75% and 100% of every nucleotide types in the sequence (Distribution) [60, 63, 69]. In 1982, Fickett introduced the Fickett TESTCODE score that could be calculated by combining the position and composition values (nucleotide content) of the four types of nt in a DNA sequence [85]. This score is a simple linguistic feature to distinguish mRNAs from ncRNAs [54, 58, 63, 65]. Although this score is commonly used, it is computed differently in different tools. In CPAT, it is calculated on the longest ORF region, while CPC2 calculates it on the full length of the sequences [58, 65]. Apart from these manually designed sequence features, the deep learning-based tools lncRNAnet and lncRNA_Mdeep learn the abstract features from the sequence patterns of the transcripts using the recurrent neural network (RNN) and convolution neural network (CNN), respectively, as the deep learning models are more facilitated than the vanilla machine learning methods to deal with the sequence data [53, 54].

Peptide features

Proteins or peptides are formed with multiple amino acid molecules linked by peptide bonds. A peptide bond is formed when the amino group of one amino acid molecule is linked to the carboxyl bond of another amino acid molecule through a covalent bond. Features regarding the formation and stability of a peptide bond are thus used by several studies. For example, CPC2 inspects several peptide-level features at the feature selection stage and selects the isoelectric point as one of its features. As pointed out above, the isoelectric point denotes the pH at which a peptide molecule does not carry any electrical charge [58]. Besides CPC2, isoelectric point is used by several other tools such as CPPred, PredLnc-GFStack and NCResNet [60, 63, 69]. These three tools also use the instability index and the grand average hydropathicity scores as primary peptide features [60, 63, 69]. LncFinder and NCResNet use a feature named EIIP, which is an improved version of isoelectric point and can be directly applied to RNA sequences [57, 69] (Table 1).

A few studies employ other features that do not belong to the above three categories. For example, LncFinder uses several RNA secondary structure features calculated by the minimum free energy-based algorithms [57]. LGC considers a relationship between the ORF size and the GC content in the given sequence to model the coding potential score [64]. NCResNet is another tool that considers the GC content and the GC variance [69]. LncADeep and PredLnc-GFStack use the entropy density profile (EDP) in ORFs or the whole transcript regions to estimate the entropy of the k-mer or amino acid compositions [60, 67]. LncADeep and CREMA use features from the alignment information of a transcript to a protein database [61, 67]. Although a variety of features have been utilized by different studies, in most cases, the commonly used features are slightly updated to adapt with the newly published datasets and emerging technologies (Table 1).

Computational methods for lncRNA identification

We surveyed 17 recently published tools on lncRNA identification (Table 1). A tool published after 2012 was chosen if (i) the tool has publicly available source codes or a ready-to-use package and does not depend a third-party library or database; (ii) the tool takes only sequences as input and (iii) the tool provides the trained human models or species-independent models. The classifiers used by these tools include traditional statistical method like maximum likelihood estimation (MLE); popular machine learning methods such as logistic regression (LR), support vector machines (SVM), random forests (RF), gradient boosting (GB) and extreme gradient boosting (XGBoost); and the latest deep learning-based methods like deep neural network (DNN), CNN, RNN, deep belief network (DBN) and residual neural network (ResNet) [52, 55, 61, 67, 69, 86–104]. Some of these tools apply similar classifiers with different feature sets or different classifiers with similar feature sets. Therefore, side-by-side comparison of these tools can provide insights about the classifiers or feature sets that contribute the most to identify lncRNAs, which can further assist the study of lncRNAs and their functions. It is also interesting to see whether the performance of deep learning-based methods can justify the higher demand for computational resources.

Coding Potential Assessment Tool

Coding Potential Assessment Tool (CPAT), published in 2013, uses a LR model to classify the lncRNA sequences based on the ORF size, ORF coverage, Fickett score and hexamer score features [65]. CPAT takes the FASTA formatted sequence data as input and outputs the four feature scores, the coding probability and the predicted class (coding or non-coding) for the input sequence. CPAT claims a higher performance [area under the receiver operating characteristic curve (AUROC) 0.99] than the contemporary tools like CPC, PhyloCSF and PORTRAIT [65]. CPAT also provides a web service, which is more user-friendly than its local installation version.

Coding-Non-Coding Index

Coding-Non-Coding Index (CNCI) was introduced in 2013, which classifies lncRNAs and mRNAs by profiling the ANT feature using SVM [62]. CNCI takes the FASTA formatted sequence data as input. Given a transcript sequence, CNCI identifies the most-like coding region among the six reading frames and calculates the S-score, ORF coverage, score-distance and codon-bias features in this most-like coding region using the profile of the ANT feature. It trains the SVM model with the four features from the training sequences and outputs a coding potential score between -1 and 1 with a predicted class label indicating non-coding or coding for each input testing sequence. CNCI reported a high accuracy (97.3%).

PLEK

PLEK employs an SVM model that learns from the frequency profiles of k-mer patterns ($k = 1$–5) [59]. It claimed to achieve up to 95.6% accuracy on human RefSeq mRNAs and GENCODE lncRNAs. PLEK introduced an improved k-mer scheme according to the length k to increase its performance. PLEK takes the FASTA formatted sequence data as input and provides a score for each sequence, where a score ≤ 0 indicates that the input sequence is an lncRNA sequence.

CPC2

CPC2, published in 2017, is an updated version of the Coding Potential Calculator (CPC) algorithm [58, 105]. CPC2, reportedly, is 1000 times faster than CPC. CPC2 applies an SVM model that uses the ORF size, ORF integrity, Fickett TESTCODE score and isoelectric point as the input features. CPC2 claims over 95% accuracy on the human dataset and 93.7–99.1% accuracies

for mouse, zebrafish, fly, nematode and *Arabidopsis*. For each FASTA formatted sequence input, CPC2 generates the features and trains the SVM classifier to predict the coding or non-coding label for the sequence, along with a coding probability. The CPC2 model is species-neutral, which comes in handy for ever-growing non-model organism transcriptomes, especially those without the genome assembly.

FEELnc

FEELnc was also published in 2017, which uses the RF model and considers the ORF coverage and multiple *k*-mer schemes ($k = 1$–12) as the training features [66]. FEELnc achieved a 95.6% accuracy with 10-fold cross-validation on the human training datasets. The authors claimed that FEELnc had higher performance on the reads generated by PacBio and 454 platforms due to longer read length. FEELnc is the first tool that allows users to annotate conservative sets of lncRNAs and mRNAs by automatically fixing their own specificity thresholds. The authors also developed a score named '*k*-mer in short' to combine the multiple *k*-mer frequencies in a faster way. FEELnc takes the FASTA formatted sequence data as input, calculates the feature scores and uses the trained RF model to predict a coding or non-coding label along with a coding potential score.

LncRNAnet

lncRNAnet is one of the earliest published tools that use deep learning techniques to identify lncRNAs [66]. Instead of finding the ORF region in a traditional way, which may not include non-canonical start codons, it considers the stop-to-stop codon frames as the ORF indicator. It uses the sequence within the ORF indicator frames to train a network of 1-D convolutional layers to predict the ORF indicator. For a given sequence, first, it predicts the ORF indicator values across the sequence as a vector of the same length as the sequence. Then, it uses both the sequences and the ORF indicator vectors to train an RNN to predict the coding probability score. The score >0.5 indicates that the provided sequence is an lncRNA. LncRNAnet claims a 91.79% accuracy and an AUROC of 0.97 on the human dataset.

CPPred

CPPred, published in 2019, is another SVM-based tool [63]. The perk of this tool is that it also focuses on distinguishing the 'small' coding RNAs and 'small' ncRNAs along with the regular coding RNAs and lncRNAs. It uses the same ORF features as CPC2 with two additional peptide-level features, the stability and grand average hydropathicity; *k*-mer features such as hexamer score and 30 CTD features. The CTD features include the four frequencies of A, T, C, G; six transitions between A and T, A and C, A and G, T and C, T and G and C and G; 20 distribution features containing the positions of occurrences of the first 25%, 50%, 75% and 100% of A, T, C and G. CPPred is the first tool that uses the CTD features to predict coding potential in eukaryotes. They show that the CTD features are important to predict the sequences with small ORFs. CPPred claims a 96.23% accuracy and a 0.99 AUROC with its human test set. CPPred extracts all the features from the input FASTA formatted sequences and uses the trained SVM model to predict the coding or non-coding label along with the coding potential.

LGC

LGC is the first tool to consider the GC content to differentially characterize lncRNAs and mRNAs [64]. For each FASTA formatted sequence, LGC calculates the length and the GC content of the longest ORF, along with coding potential score and coding label. LGC models the relationship between the ORF size and the GC content with four parameters and uses the MLE to estimate the values of the four parameters. LGC reported a 94.5% accuracy on the human test data and a higher accuracy than CPC, CNCI, CPAT and PLEK to identify lncRNAs in a cross-species manner without the need for species-specific adjustments.

LncFinder

Another SVM-based tool, LncFinder, considers carefully curated feature sets including the ORF size, ORF coverage, EIIP, the distance between hexamer frequencies of a new transcript and lncRNA transcripts, and features extracted from the secondary folding structure of functions. LncFinder performs feature selection to rule out redundant features and model selection to find the optimized model. The reported highest accuracy is 97% on the human test data. LncFinder can be installed as an R package or can be accessed directly via the web server. It takes FASTA formatted sequences as input and outputs coding or non-coding label for each sequence. Other than lncRNA identification, the stand-alone version of LncFinder also provides sequence inputs with varying lengths.

LncRNA_Mdeep

Another deep learning-based tool lncRNA_Mdeep uses a CNN to learn from the sequence and two separate DNNs to learn from the *k*-mer frequency profile and other features (ORF size and coverage, Fickett score and hexamer score) [54]. The final hidden layers of the three neural networks are concatenated together to predict coding probability scores. lncRNA_Mdeep takes FASTA formatted sequences as input and outputs the coding or non-coding label along with coding potential score for each sequence. It reported a 98.73% accuracy on the human validation dataset.

CNIT

CNIT, an updated version of CNCI, was published in 2019. It uses the same ANT features used by CNCI with a more powerful ensemble machine model XGBoost [55, 62]. CNIT reported higher AUROC scores than CNCI, CPC2, CPAT and PLEK on a variety of species from both animal and plant kingdoms. Similar to CNCI, it also takes the input sequence in the FASTA format and outputs a value from -1 to 1, where a value <0 indicates an lncRNA.

CREMA

CREMA is another ensemble method-based tool that was published in 2018, which applies LR as a stacking generalizer of an ensemble of gradient boosting models [61]. Eight models were trained using negative data sets from eight different combinations of species. The prediction results of the eight models were then used to train a LR classifier. The prediction of the LR classifier is considered as the final output. CREMA uses a recursive feature elimination strategy to remove a set of features from its initial set. On an independent training dataset, CREMA showed an 88.3% AUROC and a 99.4% specificity. The model takes transcript sequences in the FASTA format and outputs a

value between 0 and 1, where a score ≥0.5 indicates an lncRNA prediction.

LncADeep

LncADeep is a deep learning-based tool that employs DBN as the classifier [67]. LncADeep is designed to handle partial annotated transcripts when making a prediction. From an initial dataset containing both partial-length and full-length mRNA transcripts, 21 DBN classifiers were trained with datasets consisting of partial- and full-length mRNAs. The final output of LncADeep was decided by the majority votes of the 21 trained classifiers. It uses the following features to classify transcripts: ORF length and coverage, EDP of ORF, mean hexamer score, Fickett score, HMMER index, UTR length and GC content. LncADeep showed a higher harmonic mean of sensitivity and specificity than other tools on datasets containing 100% full-length, 100% partial-length and mixed length transcripts.

Lncident

Lncident applies an SVM model to classify the lncRNA transcripts [57]. For a given transcript, it uses the length and coverage of the longest ORF and 5-mer usage within the ORF region as features. Compared with other tools like CPAT, CNCI, PLEK and CPC, it showed a higher F1-score on both human and mouse datasets. It takes the FASTA formatted transcript sequences as input and outputs a score between 0 and 1, where a score <0.5 indicates an lncRNA.

PredLnc-GFStack

Another ensemble-based tool named PredLnc-GFStack was published in 2019 [60]. It uses a genetic algorithm as its feature selection strategy. The best feature subsets are employed to train separate RF models. The average prediction scores of the RF models are considered as the final score. The common selected features of the high performing RF models for human and mouse include the GC content, ORF integrity, EDP of transcripts and ORFs, *k*-mer profile, isoelectric point, etc. PredLnc-GFStack showed higher AUROC scores than several well-known tools on both human and mouse datasets.

BASiNET

BASiNET, published in 2018, applies a graph theory to extract features from the input transcript sequences to classify ncRNAs and mRNAs [68]. It takes a transcript sequence in the FASTA format as input and creates an adjacency graph between all *k*-mer pairs in the sequence with a provided step size. Each node of the graph represents a unique *k*-mer and the edge between two nodes represents the number of times the two *k*-mers are adjacent in the sequence. From the adjacency graph of each sequence, it calculates 10 topological properties that include 'assortativity', average, minimum and maximum degrees, 'average betweenness centrality', etc. It then trains the decision tree classifiers with the 0–1 normalized values of the topological properties to classify ncRNA and mRNA transcripts. In comparison with CNCI, PLEK and CPC2, BASiNET showed higher accuracies in six different species.

NCResNet

NCResNet is published in 2020 [69]. It extracts 57 features from the input transcripts that convey sequence, protein, RNA structure and physicochemical property information. It then trains a deep ResNet-based model containing four modules: input, feature enhancement, deep feature learning and prediction. The input model takes sequences as input and calculates the 57 features. The feature enhancement module enhances the feature information using repeated layers and combines the information with a flatten layer. The deep feature learning module contains 6 units, where 3 of them contain the residual units designed to capture the high-level features. Finally, the prediction module integrates the learned features to predict the output. Compared with five other tools on seven species datasets, NCResNet showed higher accuracy scores in five of the seven species for long transcripts and in all seven species for short transcripts.

Most of the 17 tools consider ORF-based features and/or some type of *k*-mer-based features with traditional machine learning models such as SVM and RF. A few of the recently published tools also consider the peptide level features and deep learning models of varied flavors and structures. Several tools used feature selection strategy like feature elimination, feature selection by genetic algorithm, feature expansion by deep learning layers [57, 60, 61, 69]. While all tools reported high performance in terms of different performance metrics on their own test datasets, a comprehensive comparison among them on a unified dataset is necessary to find the best performing tools in terms of both accuracy and efficiency. Alongside, a proper investigation of the feature patterns used by the tools is needed to identify the role of the feature set used in the lncRNA identification problem. In the following, we thus evaluate these tools and the features on unified datasets.

Testing data and comparison criteria

Datasets

The aforementioned 17 tools were tested using different test data sets as reported in the corresponding studies. To obtain an unbiased conclusion on the superiority of the tools in the identification of lncRNAs, we here present several benchmark datasets on which we evaluate these tools in terms of the runtime, memory requirements and accuracy.

The first dataset is the protein-coding and lncRNA transcript sequences in human from GENCODE Release 32 (GRCH38.p13), which we denote as the 'HA1 dataset' (Supplementary Table S1 available online at https://academic.oup.com/bib). The HA1 dataset contains 100 291 mRNA and 48 351 lncRNA transcript sequences, downloaded directly in the FASTA format, which is the required input format of all tools we surveyed here. Since some tools (such as CPAT) have alternative models for mouse, for a more comprehensive comparison, we also used the annotated protein-coding and lncRNA transcript sequences in mouse from GENCODE Release M24 (Supplementary Table S1 available online at https://academic.oup.com/bib). The mouse dataset consists of the FASTA formatted 67 056 mRNA and 18 800 lncRNA transcript sequences.

Because GENCODE contains genome-wide annotations for the transcript sequences independent of experimental conditions, only a small portion of the annotated transcripts are expressed in a given cell type. To produce a reliable context-specific human annotation dataset, we considered the total RNA-seq data in human T cells differentiated under TH1

condition (SRR1817386, TH1 Primary_2695 data) [106]. This RNA-seq dataset was also used by lncRNAnet for the performance evaluation [53]. We generated the new dataset by overlapping the GENCODE protein-coding and lncRNA transcript annotations with the expressed transcripts from these RNA-seq data. We denote this dataset as the 'HA2 dataset', which consists of 9728 mRNA and 1170 lncRNA sequences (Supplementary Table S1 available online at https://academic.oup.com/bib). Note that context-specific mRNAs and lncRNAs in this study mean these transcripts are expressed in a specific cell type and do not mean that they are not expressed in other human cell types or cell lines.

We used the HA1 dataset to compare the accuracy of the 17 tools. This dataset is relatively large, and the number of sequences within each length interval is also different. In order to distinguish the performance of the tools in terms of runtime and memory requirement and finally establish relationships between runtime and memory requirements with respect to the length of the input sequences, we needed a dataset that is rich with the consistent number of sequences of different lengths of our choice. To serve this purpose, we designed an 'HM dataset' consisting of 80 000 mRNA and 80 000 lncRNA transcript sequences chosen from the HA1 dataset that fall evenly under the four following length ranges: (0, 1000], (1000, 2000], (2000, 3000] and (3000, 100 000]. In other words, each length range contains 20 000 mRNA transcript sequences and 20 000 lncRNA transcript sequences.

To evaluate how the tools perform on more species, we created the 'HA3 dataset' comprising transcripts from 30 species. This dataset was generated from the different species dataset published by Duan *et al*. [107]. The original dataset contains transcripts from 33 species, which are categorized into 18 representative core species and 15 peripheral species by their phylogenetic relationships. To create the HA3 dataset, we considered all 18 representative core species and 12 of the 15 peripheral species and excluded the 3 peripheral species (*Homo sapiens*, *Macaca fascicularis* and *Gorilla gorilla*) that contained the NONCODE defined lncRNA transcripts (Supplementary Table S2 available online at https://academic.oup.com/bib). NONCODE was built on the positive prediction of CNIT, which is one of the tools we evaluate here. Using the NONCODE transcripts from the three species would thus create bias towards CNIT and be unfair to other tools [55]. For each core species, we randomly chose 2000 mRNAs sequences and 2000 lncRNAs sequences. We only considered sequences with length between 200 nt and 5000 nt. For each peripheral species, 500 mRNA and 500 lncRNA sequences with length between 200 nt and 5000 nt were selected. Because the number of lncRNA sequences required is not available for some species, the HA3 dataset consists of 42 000 mRNA and 33 155 lncRNAs. We evaluated all provided models of each tool on this dataset.

Feature interpretation

To assess the importance of the features used by the 17 tools, we applied three popular feature interpretation techniques: analysis of variance (ANOVA) *F*-value [108], feature ranks by recursive feature elimination (RFE) [109] and information gain using decision trees [110], to measure the importance of the features in terms of identifying lncRNAs.

ANOVA *F*-value selects n most significant features by applying an ANOVA test on the dataset. Here, n is the desired number of features provided as an input. RFE needs a classifier to fit the dataset with the feature scores, such as RF and LR. It ranks the features in the order of their importance scores calculated by the classifier. Then, it removes the least important feature from the dataset and fits the classifier again with the remaining features. It iterates the above steps until the desired number of features is obtained. Information gain, also known as Kullback–Leibler divergence [111], is a characteristic used by the decision tree classifiers to decide which feature and feature value should be used as the threshold to bifurcate the dataset. In a decision tree structure, every feature value divides the dataset into two groups. The 'entropy' of a group is decided by the following equation:

$$H(g) = -\sum_{i \in C} p_i \log_2 p_i$$

Here, $C = \{0, 1\}$ represents the two classes of a binary classification task and p_i denotes the percentage of data in the class i. Information gain can be calculated for each feature value. If a feature value f divides the parent group g into two children groups g_1 and g_2, IG(f) is calculated by the difference between the entropies of the parent and children groups,

$$\mathrm{IG}(f) = H(g) - (H(g_1) + H(g_2))$$

The maximum information gain for all values of a feature F on a dataset is considered the weight or importance of the feature.

Running time and memory usage

The runtime and memory usage of the 17 tools were obtained under the following conditions on a dedicated machine. The testing platform we used was an AMD Ryzen 2700X (8 cores @3.7GHz) with 80 gigabytes memory, with Ubuntu 18.04 Long Term Support. For the tools that allow users to provide a maximum computational resource (e.g. number of CPU threads), we used the maximum available resource of our machine (e.g., 16 threads). The runtime of a tool was calculated by obtaining the difference between the timestamps when the tool started and when it finished the running. We recorded the memory usage of our system using the 'free' command in Ubuntu. We obtained the memory usage of a tool by subtracting the memory usage recorded before the tool was executed from the maximum used memory during the tool was running. All other activities were suspended during the execution of a tool. Since the memory of our testing platform was enough for all tools, no virtual memory or swap memory was used by the tools.

Comparison of the 17 tools

Prediction accuracy

The most important criteria to evaluate the reliability of a tool is to investigate whether its predictions are consistent with the ground truths. To evaluate the performance of these tools, we used the HA1 and mouse datasets and divided each of the datasets into different length intervals. Based on the results on the two datasets, we investigated the changes in the performances of the tools in terms of the length of the input sequence and the features that might contribute to the changes. Due to the unbalanced size of the HA1 and mouse datasets, AUROC, the area under the precision recall curve (AUPR) and F1-score were also calculated along with accuracy, precision, sensitivity and specificity scores (Table 2, Supplementary Table S3 available online at https://academic.oup.com/bib).

Table 2. Performance of the tools on HA1 dataset

Tool	Pos	Neg	AUROC	AUPR	F1	MCC	Accuracy	Precision	**Recall**	**Specificity**
CPAT	48 351	100 291	0.9482	0.8530	0.8419	0.7630	0.8859	0.7664	0.9340	0.8627
CNCI	48 351	100 291	0.8647	0.6282	0.8098	0.7207	0.8508	0.6919	0.9761	0.7904
PLEK	48 060	99 917	0.8851	0.7542	0.7104	0.5633	0.7515	0.5715	0.9385	0.6616
FEELnc	48 342	100 227	0.9293	0.8119	0.7761	0.6713	0.8165	0.6436	0.9775	0.7389
CPC2	48 351	100 291	0.8899	0.7519	0.7194	0.5777	0.7610	0.5819	0.9419	0.6738
LncRNAnet	48 350	100 282	0.9798	0.9433	0.9020	0.8549	0.9311	0.8397	0.9742	0.9103
CPPred	48 351	100 291	0.7923	0.5029	0.7108	0.5696	0.7440	0.5618	0.9673	0.6363
LGC	48 351	100 291	0.8421	0.6622	0.6833	0.5158	0.7219	0.5427	0.9220	0.6255
LncFinder	48 351	100 291	0.9273	0.7924	0.8145	0.7242	0.8586	0.7103	0.9543	0.8124
LncRNA_Mdeep	48 351	100 291	0.9744	0.9158	0.8984	0.8499	0.9281	0.8317	0.9767	0.9047
CNIT	48 351	100 291	0.9383	0.8172	0.8324	0.7538	0.8719	0.7249	0.9772	0.8212
CREMA	48 351	100 291	0.7977	0.7245	0.6910	0.6473	0.8445	**0.9771**	0.5345	**0.9939**
LncADeep	48 351	100 291	**0.9872**	**0.9676**	0.9297	0.8956	0.9520	0.8881	0.9753	0.9408
Lncident	48 351	100 291	0.8269	0.5480	0.7379	0.6151	0.7732	0.5912	**0.9815**	0.6728
PredLnc-GFStack	48 351	100 291	0.9618	0.915	**0.948**	**0.9225**	**0.9659**	0.9455	0.9501	0.9736
BASiNET	48 351	100 291	0.56	0.358	0.406	0.1201	0.6137	0.4063	0.4065	0.7136
NCResNet	48 351	100 291	0.5116	0.325	0.405	0.0209	0.5111	0.3352	0.5112	0.5111

The best scores are highlighted in bold. PLEK filtered out 291 lncRNAs and 374 protein-coding sequences. FEELnc filtered out 9 lncRNAs and 64 protein-coding sequences. LncRNAnet filtered out 1 lncRNAs and 9 protein-coding sequences.

By having a quick look at the performance results of the tools, we can safely declare that the three of the four deep learning-based tools lncRNAnet, lncRNA_Mdeep and LncADeep were the best in terms of all the performance metrics (Table 2). In terms of AUROC and AUPR performance metric, LncADeep was the best performing tool. In every performance metric, LncRNAnet and lncRNA_Mdeep were very close to LncADeep. It is also noteworthy that, using a relatively simple feature set that includes a unique ORF indicator feature, lncRNAnet beat most of the other tools. This shows the power of the deep learning models to pick up intricate features directly from sequence inputs, which helps the models keep the number of false positives down. But using complicated deep learning layers could not ensure high performance, as another deep learning-based tool NCResNet, which considered almost all features and a very deep model with a lot of layers, surprisingly underperformed and was the worst performing tool on both the HA1 and mouse datasets. Apart from NCResNet, another poorly performing tool was BASiNET, which used the topological properties of the adjacency graph of unique k-mers in a sequence to train decision tree models. Among other machine learning tools, PredLnc-GFStack performed the best on both HA1 and mouse datasets. It outperformed all other tools including the deep learning tools in terms of F1, MCC and accuracy scores (Table 2). The specialty of PredLnc-GFStack is that it uses feature selection by genetic algorithm, trains multiple RF models with the best performing feature subset and finally considers the maximum voting prediction of the models as the final prediction.

We also noticed that all tools showed steady improvement rates in their performances with the growth of sequence length up to 2000 nt (Figure 1, Supplementary Figure S1 available online at https://academic.oup.com/bib). When the sequence length went beyond 2000 nt, the specificity scores of the tools kept improving, while the recall scores degraded for most tools. FEELnc showed consistently high recall scores for different lengths of input sequences. When the sequence length was over 10 000 nt, the AUPR, F1-score and Precision metrics dropped significantly for most tools.

CPAT and CPC2 use the ORF size and the Fickett score as common features. As additional features, CPAT uses the ORF coverage and the hexamer score, while CPC2 uses the ORF integrity and three peptide features. CPPred considers all features from CPAT and CPC2 together with the additional 30 CTD features (Table 1). Despite using a simpler feature set and simpler machine learning model (Table 1), CPAT outperformed CPC2 and CPPred in terms of all prediction metrics except recall. The only features, that both CPC2 and CPPred use but CPAT does not, are the peptide features, which help CPC2 and CPPred have a little higher recall than CPAT but ultimately generated many more false positives.

Too many features may sometimes hurt the performance. For example, PLEK only uses the k-mer profiles with different k values as its feature set to train an SVM model. CPPred also trains an SVM model, but with a much larger feature set (Table 1). In terms of performance, PLEK was on par with CPPred, if not better, showing the futility of having too many features. Another example, NCResNet considers 57 features of different kinds, while still shows a poorer performance compared with tools using fewer features.

Some tools such as CNCI, PLEK and LGC do not consider sequences smaller than 200 nt during the model training. According to our benchmark datasets, none of the tools we tested can accurately identify the protein-coding sequences shorter than 200 nt, resulting in a huge number of false positives in that length range (Figure 1, Supplementary Figure S1 available online at https://academic.oup.com/bib). The reason may be some of these tools tend to classify shorter (<200 nt) transcript sequences as 'non-coding'. PLEK had comparatively higher performance in the (0, 200] region on the surface, but its specificity was as poor as the other tools on the HA1 and mouse datasets. On a closer look we found that, out of the 375 mRNAs and 298 lncRNA sequences in the HA1 dataset, PLEK generated results for only one protein-coding and seven non-coding sequences. Similarly, in the mouse data set, only 2 of the 248 lncRNAs and 1 of the 290 mRNAs were recorded in the PLEK prediction results. The few recorded transcripts by PLEK in the (0, 200] range were exactly 200 nt long. The other transcripts were filtered out due to their smaller lengths (<200 nt).

There were 16 lncRNAs from the HA1 dataset that could not be predicted as lncRNAs by any tool. After visualizing the scores

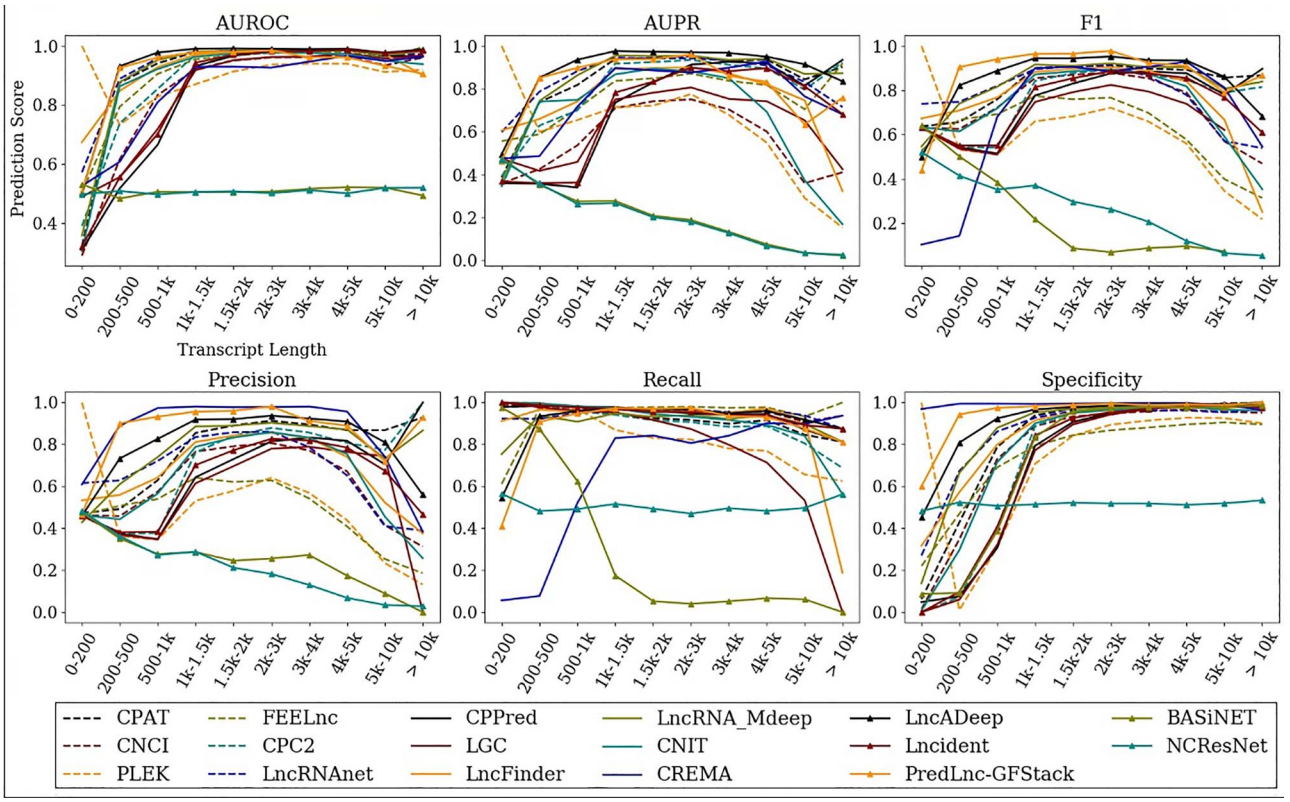

Figure 1. Comparison of prediction performance of the tools on the HA1 dataset. Six performance metrics are shown for different input sequence lengths. The *X* and *Y* axis labels are shown for the first subfigure only as they are the same for all subfigures.

of the 17 tools for these 16 lncRNAs, it appeared that CNCI and PLEK came close to predict them. We found that 13 out of the 16 lncRNAs were overlapped by at least one protein-coding RNA, which could be a reason for the tools to mistake them for mRNAs (Figure 2). On average, around 50% of the misclassified lncRNAs by each of the 17 tools were found to overlap with a protein-coding mRNA. These lncRNAs overlapped with protein-coding gene transcripts were more challenging for the tools to classify correctly. The remaining three lncRNAs had a much larger ORF length (1107 nt, 951 nt and 1653 nt) than the median ORF length (423 nt) of the lncRNAs in the HA1 dataset, which might explain their misclassification.

Analyses on the RNA-seq and 30 species data

To reflect the performance of the tools on predicting context-specific lncRNAs, we compared their performance using the RNA-seq dataset HA2 (Table 3). The HA2 dataset is a subset of the HA1 dataset. It was used to produce results that focused on context-specific transcripts. All the mRNA transcripts in this dataset are longer than 200 nt. CNCI could not handle the HA2 mRNA transcripts with length over 10 000 nt in 400 h. So, the tool was not executed on those transcripts. Due to the filtering rules of PLEK, lncRNAs smaller than 200 nt and mRNAs more than 10 000 nt could not be included to evaluate this tool.

The overall performances of all tools were much lower on the HA2 dataset, compared with the HA1 dataset. In terms of recall, FEELnc was still the best on the HA2 dataset, as it was on the HA1 dataset. PLEK performed better than other tools in terms of AUPR and precision. CNCI performed better than others in terms of F1-score and precision. But due to the missing mRNA transcripts, the negative dataset of PLEK and CNCI was smaller than other tools. Keeping the consistency in the number of positive and negative datasets in mind, LGC performed better than others in terms of the AUROC, accuracy and specificity. Note that, unlike other tools, LGC models the relationship between the ORF size and the GC content in a sequence to decide its class by MLE. Although the deep learning-based tools lncRNAnet and lncRNA_Mdeep were the best performing tools on HA1, both of them were among the lowest performing tools in terms of specificity. The overall poor performance of the tools on the HA2 dataset suggests that the annotation datasets, these tools were trained on, might be significantly different from the context-specific annotations, and the models used by the tools are most likely overfitted on the transcripts that are not expressed under the context-specific conditions. Note that one cannot simply filter the predicted lncRNAs based on their expression under a given experimental condition to define the context-specific lncRNAs under this condition, because at least more than 6.5% of lncRNAs (highest recall 93.5%) cannot be correctly predicted by the available tools in this case (Table 3), indicating that the context-specific lncRNAs in at least certain cell types may not be represented well enough in the GENCODE annotation.

When we evaluated the tools on the 30-species HA3 dataset, we found that Lncident performed the best in terms of the F1-score and AUROC (Table 4). Lncident uses a simple SVM model with only ORF features and ANT features of *k*-mers. LncADeep was the second best tool in this regard. Overall, all tools except CREMA, BASiNET and NCResNet showed high AUROC and AUPR on this dataset.

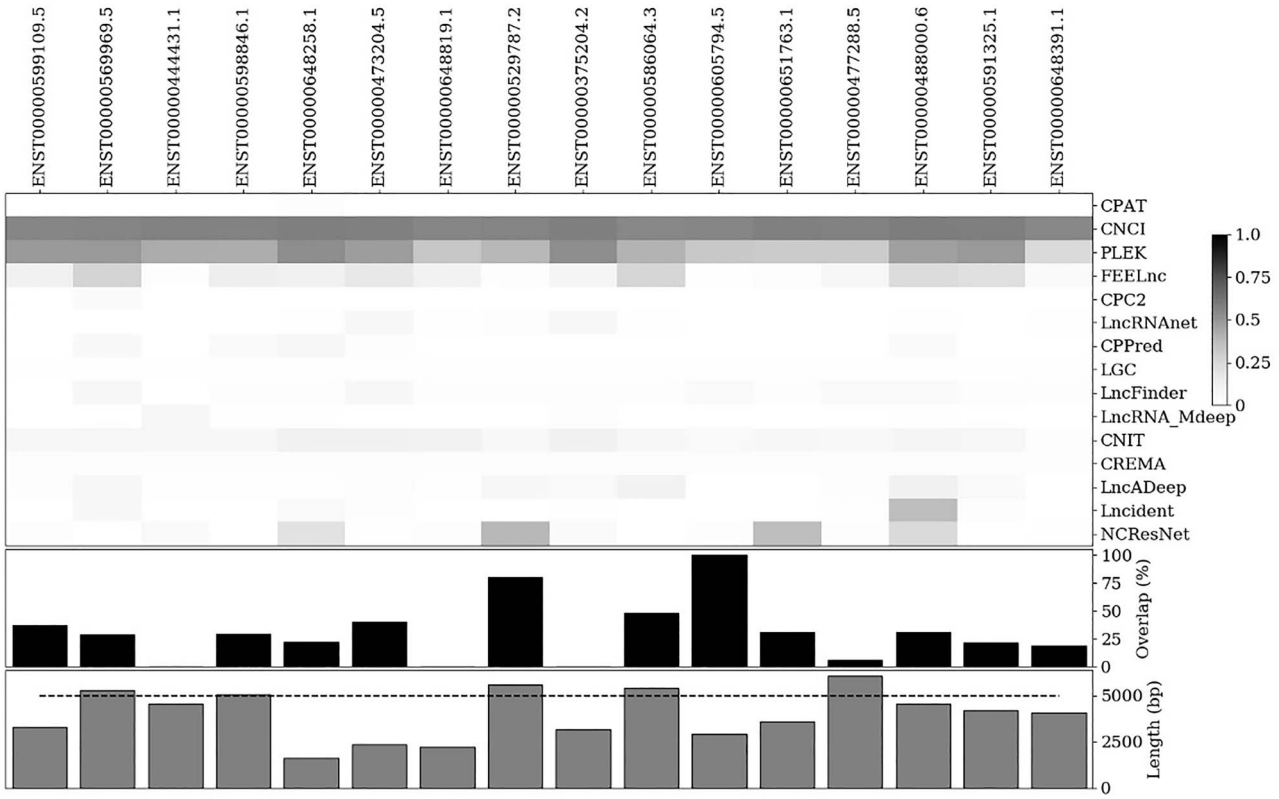

Figure 2. The heatmap (top) shows the scores of the 10 tools for 16 lncRNAs unidentified by any of the tools. The bar plot (middle) shows the maximum percentage of overlapping area of the lncRNAs with an mRNA. The bar plot (bottom) shows the length of the 16 lncRNAs.

Table 3. Performance of the tools on HA2 dataset

Tool	Pos	Neg	AUROC	AUPR	F1	MCC	Accuracy	Precision	Recall	Specificity
CPAT	1170	9728	0.5833	0.1212	0.2287	0.1097	0.4448	0.1344	0.7667	0.4060
CNCI	1170	1157	0.4657	0.4937	**0.6528**	0.0887	0.5346	0.5223	0.8701	0.1953
PLEK	1153	2388	0.7041	**0.5576**	0.5452	0.2438	0.5609	0.4113	0.8083	0.4414
FEELnc	1170	9728	0.5593	0.1593	0.1877	-0.0494	0.1312	0.1043	**0.9350**	0.0345
CPC2	1170	9728	0.6927	0.3296	0.2468	0.1418	0.5139	0.1480	0.7419	0.4865
LncRNAnet	1105	8411	0.5592	0.1353	0.2119	0.0326	0.2732	0.1212	0.8416	0.1985
CPPred	1170	9728	0.7166	0.3380	0.2393	0.1375	0.4406	0.1401	0.8197	0.3950
LGC	1170	9728	0.7367	0.2901	0.4119	**0.3354**	**0.8297**	0.3273	0.5556	0.8627
LncFinder	1170	9728	**0.7375**	0.4409	0.3209	0.2417	0.6920	0.2102	0.6778	0.6937
LncRNA_Mdeep	1170	9728	0.6182	0.2190	0.1990	0.0354	0.2790	0.1130	0.8342	0.2122
CNIT	1170	9726	0.7361	0.4352	0.2750	0.1849	0.5910	0.1698	0.7222	0.5753
CREMA	818	2388	0.4422	0.2323	0.3119	-0.0712	0.4414	0.2275	0.4963	0.4225
LncADeep	1170	9728	0.5353	0.1192	0.1943	0.0184	0.2731	0.1103	0.8162	0.2078
Lncident	1172	9730	0.6648	0.2024	0.2079	0.0677	0.2765	0.1178	0.8831	0.2034
PredLnc-GFStack	818	2388	0.5581	0.2800	0.4198	0.1101	0.4482	0.2869	0.7824	0.3338
BASiNET	1172	2388	0.5804	0.4049	0.3150	0.2574	0.7093	**0.7021**	0.2031	**0.9577**
NCResNet	1170	9728	0.5054	0.1070	0.1789	0.0071	0.5223	0.1097	0.4846	0.5268

The best scores are highlighted in bold. All the tools except Lncident are missing some of the lncRNA and mRNA transcripts in HA2 dataset.

Feature analysis

We studied most of the feature scores calculated by the tools on the HA1 and mouse datasets. In our analysis, we considered the ORF size, Fickett score and hexamer score features from CPAT; the ORF coverage, grand average hydropathicity and instability score features from CPPred; the ORF integrity and isoelectric point features from CPC2; and 1, 2, 3, 6, 9 and 12-mer features from FEELnc prediction results. We also inspected the effectiveness of the features to classify the input sequences of different lengths (Figures 3–5, Supplementary Figures S2–S4 available online at https://academic.oup.com/bib).

The ORF features are among the most important features used by the tools we studied (Figure 3, Supplementary Figure S2 available online at https://academic.oup.com/bib). Five of the 17 tools (CPAT, CPC2, lncRNAnet, LGC and CPPred) consider the ORF size as one of the input features. This feature may be less significant for the shorter non-coding and protein-coding sequences (<1000 nt) than for the longer ones (≥1000 nt). But it

Table 4. Performance of the tools on the 30 species HA3 dataset

Tools	Pos	Neg	AUROC	AUPR	F1	MCC	Accuracy	Precision	Recall	Specificity
CPAT (human)	33 155	41 966	0.9207	0.8960	0.9130	0.8549	0.9273	0.9670	0.8648	0.9767
CPAT (mouse)	33 155	41 966	0.9269	0.8980	0.9197	0.8626	0.9318	0.9573	0.8850	0.9688
CNCI (vertebrates)	33 155	41 534	0.8982	0.8305	0.8868	0.7942	0.8979	0.8728	0.9013	0.8952
CNCI (plants)	33 155	41 534	0.8728	0.8212	0.8568	0.7578	0.8799	0.9100	0.8094	0.9361
PLEK	33 155	41 999	0.8733	0.7789	0.8599	0.7414	0.8682	0.8095	0.9169	0.8297
FEELnc	33 141	41 999	0.8753	0.7768	0.8623	0.7457	0.8683	0.8003	**0.9347**	0.8159
CPC2	33 155	42 000	0.9239	0.8887	0.9160	0.8543	0.9280	0.9446	0.8890	0.9589
CPPred	33 155	42 000	0.9299	0.8787	0.9212	0.8579	0.9297	0.9105	0.9322	0.9276
LncRNAnet	33 155	41 964	0.9064	0.8701	0.8961	0.8244	0.9127	0.9443	0.8525	0.9603
LGC	33 155	42 000	0.9094	0.8582	0.8990	0.8214	0.9121	0.9115	0.8867	0.9320
LncFinder (human)	33 155	42 000	0.9272	0.8962	0.9199	0.8621	0.9317	0.9529	0.8892	0.9653
LncFinder (mouse)	33 155	42 000	0.9266	0.8925	0.9190	0.8594	0.9306	0.9465	0.8931	0.9601
LncFinder (wheat)	33 155	42 000	0.9081	0.8393	0.8969	0.8124	0.9064	0.8730	0.9221	0.8941
LncRNA_Mdeep	33 155	42 000	0.9105	0.8835	0.9011	0.8369	0.9181	0.9641	0.8458	0.9751
Lncident	33 155	42 000	**0.9367**	0.8976	**0.9295**	**0.8745**	**0.9382**	0.9351	0.9240	0.9494
LncADeep	33 155	42 000	0.9251	**0.9028**	0.9182	0.8635	0.9316	0.9715	0.8703	0.9799
CREMA	33 155	41 966	0.6458	0.6028	0.4530	0.4294	0.6871	**0.9912**	0.2936	**0.9980**
PredLNC-GFStack (human)	33 155	42 000	0.8732	0.8476	0.8553	0.7796	0.8863	0.9745	0.7621	0.9843
PredLNC-GFStack (mouse)	33 155	42 000	0.8384	0.8096	0.8088	0.7246	0.8557	0.9739	0.6915	0.9854
CNIT (vertebrates)	32 398	41 531	0.9272	0.8804	0.9183	0.8549	0.9286	0.9212	0.9154	0.9389
CNIT (plants)	32 398	41 531	0.8533	0.8213	0.8294	0.7455	0.8689	0.9651	0.7271	0.9795
BASiNET	33 155	42 000	0.7450	0.6638	0.6897	0.5195	0.7622	0.8128	0.5990	0.8911
NCResNet	33 155	42 000	0.4956	0.4390	0.4606	-0.0088	0.4965	0.4367	0.4873	0.5039

The best scores are highlighted in bold.

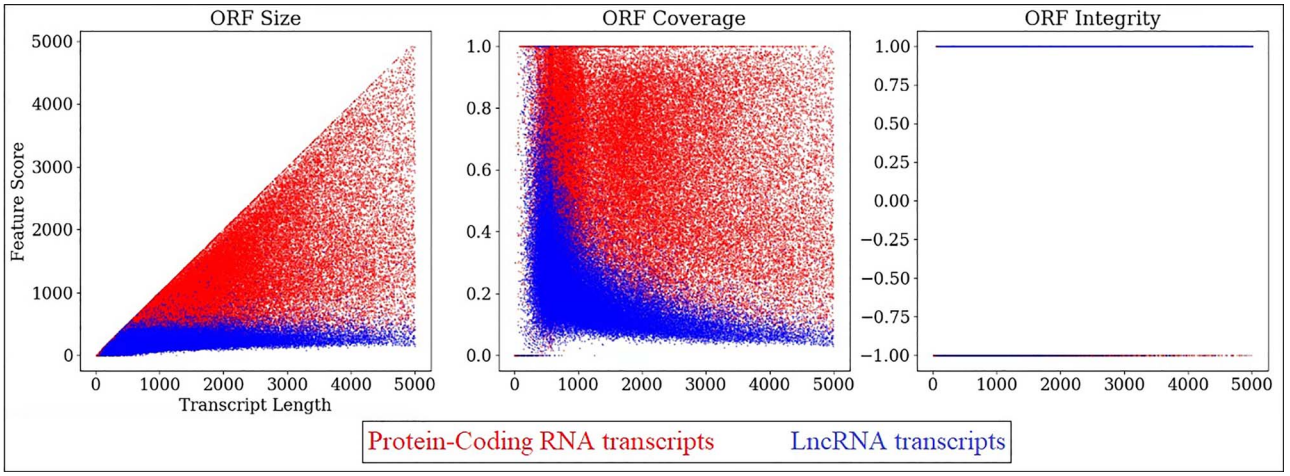

Figure 3. ORF feature values with respect to input sequence lengths. Three ORF features from CPAT are shown for different sequence lengths up to 5000 nt. The X and Y axis labels are shown for the first subfigure only as they are the same for all subfigures.

is still one of the most significant features to classify sequences, as most protein-coding transcripts usually have longer ORFs and thus the sequences with longer ORFs could easily be classified as 'protein coding' (Figure 3). Other than the ORF size, several other ORF features, such as the ORF coverage and the ORF integrity, are also used by some tools. CPAT, FEELnc, lncRNAnet and CPPred use the ORF coverage as one of the features. The ORF coverage considers not only the size of the ORFs but also the size of the input sequences. For up to a certain length (~1000 nt) of the input sequences, the protein-coding and non-coding sequences could not be completely separated into two distinct clusters. But for sequences longer than 1500 nt, the two groups showed a significant difference, where the higher ORF coverage could be confidently classified as 'protein coding' (Figure 3). The ORF integrity scores a sequence based on whether it has a start codon and a stop codon. This feature score did not help much in the classification task, which may be due to the fact that either the ORFs of a lot of lncRNA sequences have start and stop codons or a lot of protein-coding sequences miss the canonical start or stop codons (Figure 3).

The *k*-mer features also play an important part in lncRNA identification (Figure 4, Supplementary Figure S3 available online at https://academic.oup.com/bib). Among these features, the Fickett TESTCODE scores and the hexamer scores have the most discriminative potentials. For input sequences shorter than 1000 nt, the protein-coding group and the non-coding

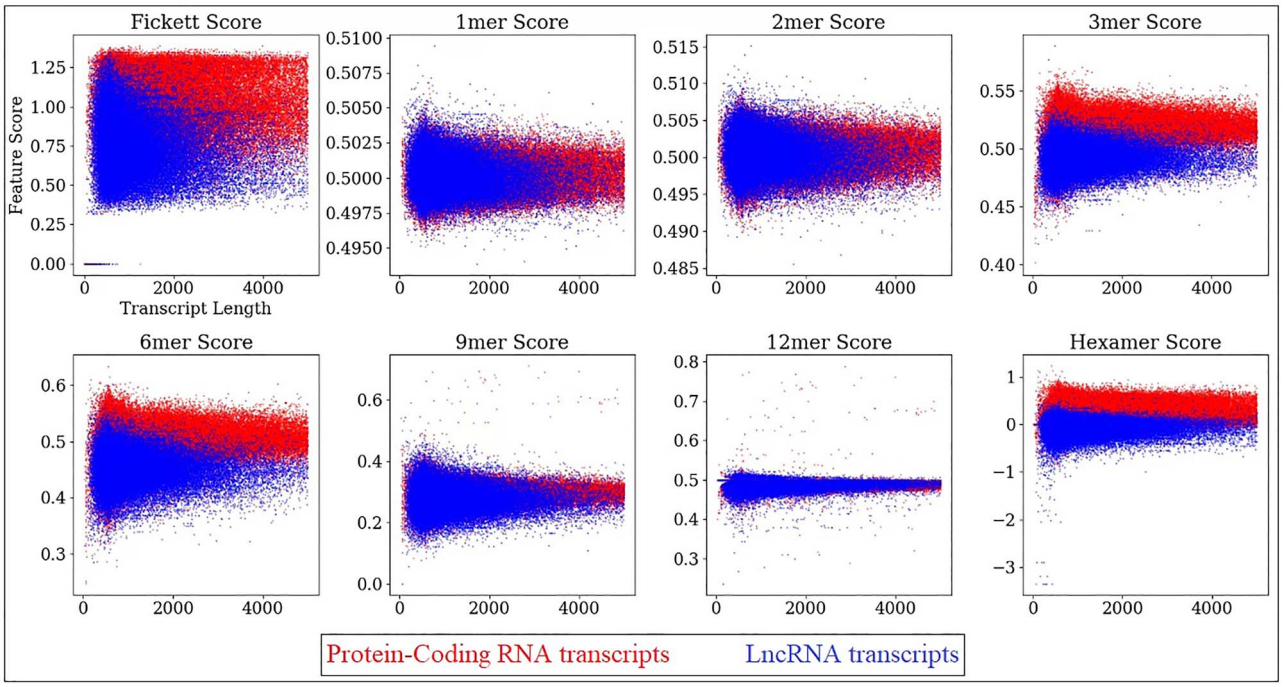

Figure 4. *K*-mer feature values with respect to input sequence lengths. The Fickett score and hexamer score features from CPAT and 1-, 2-, 3-, 6-, 9- and 12-mer features from FEELnc are shown for different sequence lengths up to 5000 nt. The *X* and *Y* axis labels are shown for the first subfigure only as they are the same for all subfigures.

group form visually indistinguishable clusters in terms of the Fickett score. As the sequence becomes longer, the non-coding group appears to obtain a lower Fickett score that can be visually distinguished from the protein-coding group (Figure 4). The hexamer scores on the other hand show a consistent discrimination between the protein-coding and non-coding sequences, where relatively the higher hexamer scores (>0.5) can be confidently classified as protein coding (Figure 4). Among the *k*-mer scores, 3-mer and 6-mer scores were the best features. The distributions of the 3-mer and 6-mer scores are separable for the two groups irrespective of the sequence length, though the classification confidence increases with the longer input sequences. In case of the other *k*-mer scores, although the lncRNA sequence scores tend to be lower than the mRNA sequence scores with the increment of the sequence length, the scores for the two groups overlap for the most part (Figure 4).

Only two of the tools use the peptide features (isoelectric point, grand average hydropathicity and instability index). LncFinder uses an EIIP as an improved version of the isoelectric point. None of these feature scores seems useful in separating mRNA sequences from lncRNA sequences for each sequence length (Figure 5, Supplementary Figure S4 available online at https://academic.oup.com/bib). This confirms the fact that since a protein molecule has both acidic (-COOH) and basic (-NH2) groups, it cannot be totally neutral (Figure 5). We also inspected the distribution peptide lengths from the output of CPC2, which shows the similar distribution of the ORF lengths (Figures 3 and 5). This shows when the ORF length is included as a feature, considering the peptide length is redundant, since the ORF length is synonymous with the peptide length [58].

To measure the importance of the 15 features mentioned above, the three feature interpretation methods; ANOVA *F*-value, RFE and Information Gain were applied (Table 5). To calculate RFE, a RF classifier with 100 estimators was used with the 'Gini-index' criterion to calculate the feature importance [109]. In case of information gain, a RF classifier with 100 estimators and the 'entropy' criterion were provided [110]. Four *k*-mer features (3-mer score, hexamer score, 6-mer score and Fickett score) and two ORF features (ORF coverage and ORF size) were ranked among the top six features by all three methods. The 3-mer score was considered the best feature based on ANOVA *F*-value and RFE, while the ORF size was considered the best feature in terms of the information gain.

Runtime and memory analysis

With the availability of powerful computational resources, there is usually more concern about how a tool performs in terms of accuracy than in terms of processing time and computational power. But in reality, the latter still needs to be taken into consideration to fully assess the efficiency of a tool, especially when evaluating on a large dataset. High maximum memory requirements might delay the progress if users do not have the adequate resources to perform such a task. Tools with a higher runtime consume more computational power along with longer time to produce the results. To present a comprehensive comparison among the 17 tools, we thus also benchmarked the tools in terms of the runtime and memory consumption on the curated HM dataset.

Although all tools take the same FASTA formatted sequences as inputs, a group of the tools showed significantly different performance than others in terms of runtime on the HM dataset (Figure 6A and B). Based on their runtime, the 17 tools were divided into two groups. The first group had a runtime less than 400 seconds when dealing with 20 000 sequences with the length within (3000, 100 000] interval (Figure 6A). The runtime of a tool is not only determined by the method it uses but also the programming language it is designed on and the expense of its feature calculation. This explains that although PLEK, CPC2,

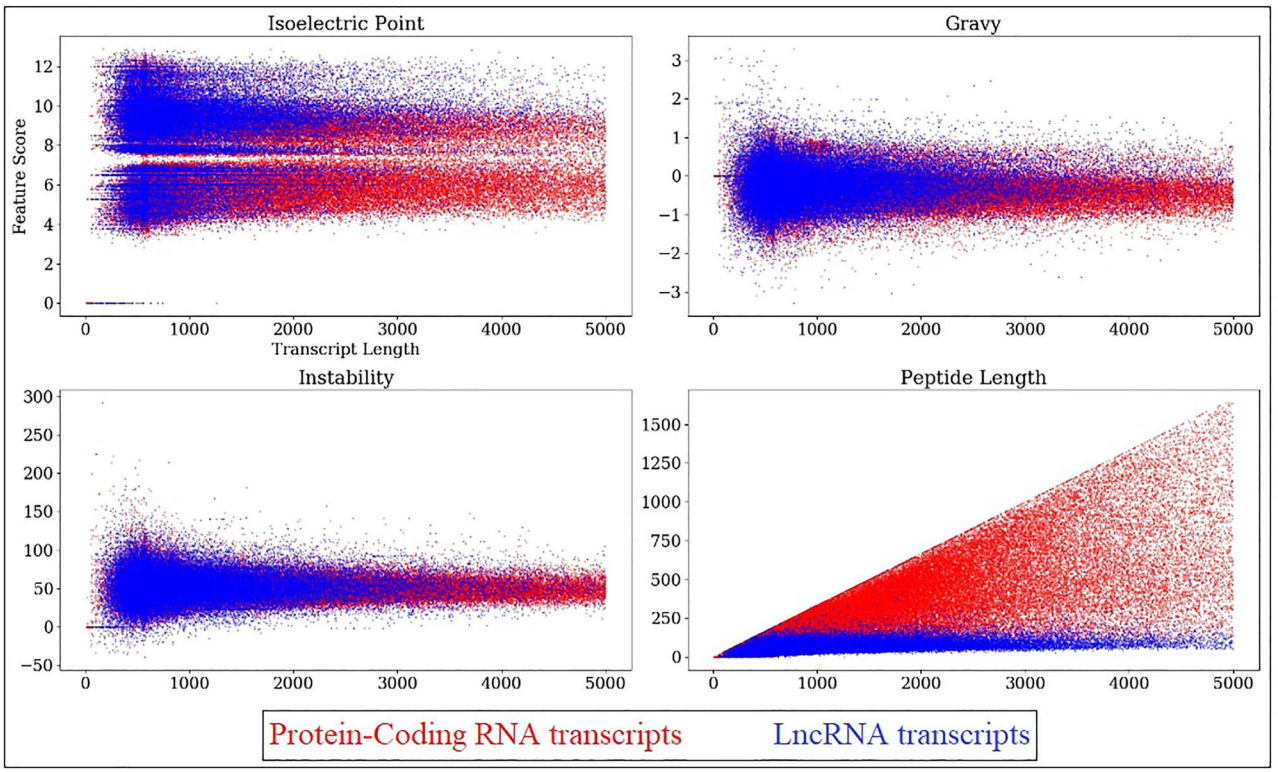

Figure 5. Peptide feature values with respect to input sequence lengths. Isoelectric point feature from CPC2 and grand average hydropathicity (Gravy) and instability index features from CPPred are shown for different sequence lengths up to 5000 nt. The X and Y axis labels are shown for the first subfigure only as they are the same for all subfigures.

Table 5. Ranks and scores of the 15 features by 3 feature interpretation methods.

	log(ANOVA F-value)	RFE-rank	Information gain
ORF size	**4.2637**	**2**	**0.1766**
ORF coverage	**4.9382**	**4**	**0.1446**
ORF integrity	3.7169	15	0.0128
Fickett score	**4.7451**	**5**	**0.0574**
1-mer score	1.4897	10	0.0264
2-mer score	1.9358	8	0.0271
3-mer score	**4.9675**	**1**	**0.1735**
6-mer score	**4.8387**	**6**	**0.0800**
9-mer score	3.9711	9	0.0298
12-mer score	0.1770	11	0.0265
Hexamer score	**4.9601**	**3**	**0.1396**
GC content	3.4784	7	0.0303
Isoelectric point	4.0300	13	0.0277
Grand average hydropathicity	3.4190	12	0.0242
Instability	3.1834	14	0.0236

The top six features are marked in bold.

CNCI, CPPred, and lncFinder all use SVM as their classification methods, the runtimes of these tools have notable divergence.

The deep learning-based tools lncRNA_Mdeep and lncRNAnet had the highest runtime on average among all tools (Figure 6B). Although the run time of lncRNAnet was lower for shorter sequences, lncRNA_Mdeep showed an equally high run time for input sequences with different lengths. lncRNA_Mdeep's workflow includes three different neural networks trained on different categories of features engineered from the input sequences, which may be the reason behind this consistently high runtime. For input sequences with longer lengths (>3000 nt), the runtime of lncRNAnet and CNCI were almost two times of lncRNA_Mdeep. lncRNAnet uses a CNN module to calculate the ORF indicator feature and an RNN module as its primary classifier, which is why despite using 'Bucketing' to minimize unnecessary cost, the time required by RNN eventually grows linearly with the increasing length of sequences.

Even though all tools were targeting the same problem, their maximum memory usage statistics (Figure 6C and D) were also

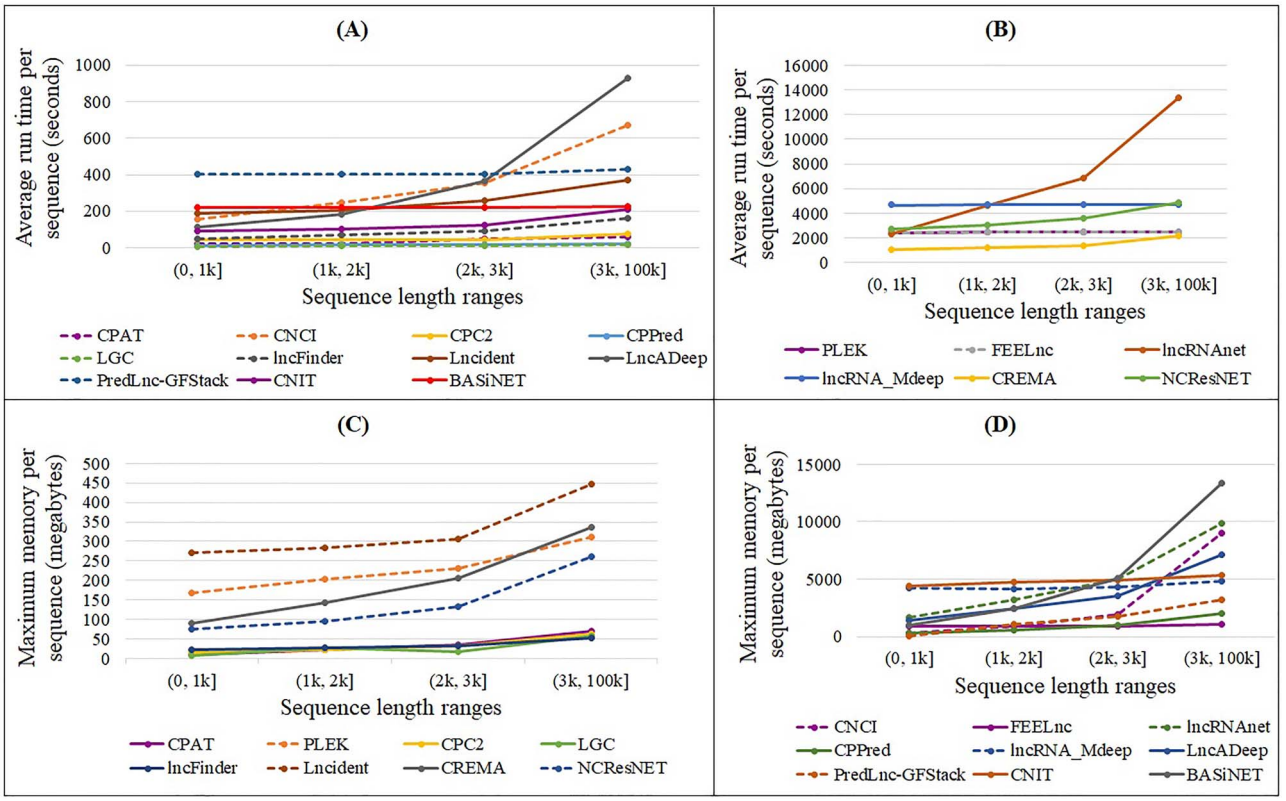

Figure 6. The run time and memory consumption of the tools on the HM dataset. The tools with lower (**A**) and higher (**B**) run times and lower (**C**) and higher (**D**) memory consumptions with different length sequences as inputs are shown.

on very different levels. LGC, due to its straightforward model, had indisputably the lowest maximum memory usage. Although BASiNET showed a low run time, its memory consumption grew faster than others with the increment of the input sequence length. BASiNET builds separate adjacency graphs for different input sequences and each graph can grow in size for larger sequence occupying a large chunk of memory. The CPC2 and CPPred, both using the SVM method with the same libsvm package, had almost identical memory footprints. The maximum memory usage of all tools except lncRNAnet and CNCI had only trivial differences when the length of the input sequence increased. Having a steady memory requirement with different sequence lengths would make these tools more suitable for tasks involving longer input sequences, such as the pri-miRNA classification. LncRNAnet and CNCI, on the other hand, consumed a large amount of memory when dealing with the longer sequences. FEELnc showed the highest memory consumption for the shorter input sequences (<2000 nt). This is the only RF classifier-based tool that calculates a novel multi *k*-mer frequency feature that might contribute to this high memory consumption.

CPAT, CPC2, LGC and lncFinder showed a consistently low runtime and memory consumption than other tools. CPAT and LGC use the LR classifier and MLE, respectively, as their base models, while CPC2 and lncFinder use the SVM classifiers. LGC was the best in terms of both runtime and memory consumption among the 17 tools. LGC uses the most simplistic model to tackle the coding potential prediction problem with the ORF size and CG content features, which improves the efficiency. Among the four tools, the memory consumption of lncFinder was higher than the other three tools. CPAT, CPC2 and lncFinder use *k*-mer features along with the ORF features. Both CPC2 and lncFinder additionally use the peptide-level features. Therefore, the only lncFinder features that are different from the other three tools are the secondary structure features, which likely took the extra memory load, especially for the longer input sequences. The two deep learning tools were the worst in terms of runtime and memory use.

Conclusions and outlook

We studied 17 published popular tools to predict lncRNA. We provided a side-by-side comparison scenario among the 17 tools in terms of their accuracy, run time, memory consumption and the features used by these tools. We also dissected the distribution of the most popular feature scores with respect to different lengths of the input sequences to find the impact of these features on lncRNA identification.

In terms of accuracy and other prediction metrics, three of the four deep learning-based tools, lncRNAnet, LncADeep and LncRNA_Mdeep, outperformed other tools. But two of the three tools suffered from the highest run time and all three tools suffered from high memory consumption, due to the nature of the deep learning-based systems. The performance of the tools increases with the increment of the input sequence length to certain range and eventually decreases when the input length exceeds the thresholds. These thresholds where the performance shifts are different for different tools. The tools with relatively less features (e.g. PLEK) may work better than the ones that use the same model but a larger feature set (e.g., CPPred). Overall, the tools do not perform well when the input sequences are too short (<200 nt) or too long (>5000 nt).

We discovered that the direct use of the encoded sequences as inputs of the deep learning-based tools can significantly contribute to their performance. These tools use CNN or RNN models that can capture the intricate sequence features, which are otherwise uninterpretable by other algorithms. Several visualization strategies [112] can be applied to interpret the learned patterns, which may provide insights on the lncRNAs identification features from different perspectives in the future.

Using too many features do not necessarily go well for better performance, rather in some case, hinder the performance of the models. Traditional machine learning models sometimes require feature selection steps to remove redundant features and improve performance. The deep learning-based models, due to their higher fitting abilities, suffer less from the redundant features but can also introduce the additional risk of overfitting. Our comparison shows that even with a similar feature set used by traditional machine learning-based methods, the deep learning-based methods are more likely to have better performance.

Among the ORF features, we could confirm the usefulness of the ORF size feature based on the distribution of the feature scores in terms of different input lengths. Among the popular *k*-mer features, 3-mer and 6-mer scores showed the most classifiable distribution with respect to different input lengths. The effectiveness of the ORF size, 3-mer and 6-mer score features was more clearly visible for longer input sequences ($\geq$1000 nt). The 3-mer score and ORF size features were also the top-ranked features based on our feature evaluation techniques. The peptide features did help the tools to improve their recall but had the risk of increasing the false positives. The lncRNA classification task becomes harder for those that are most likely to overlap a protein-coding gene transcript. The performance of all tools surprisingly dropped when the context-specific transcripts were used as inputs (the HA2 dataset), which showed that the training data of these tools might not have the context-specific support and thus were likely to be overfitted on the unexpressed annotated transcript data.

Recent findings reported a plethora of lncRNAs coding for small peptides. To observe the feature score distribution of the lncRNA transcripts that are experimentally validated to code small peptides, we downloaded and studied human lncRNA genes reported to code small peptides [113, 114]. There were in total 467 transcripts in the HA1 dataset associated with these lncRNA genes. For all features, the feature score distribution for the 467 transcripts overlapped with the feature score distributions of other lncRNAs and mRNAs (Supplementary Figures S5–S7 available online at https://academic.oup.com/bib), suggesting the challenges in distinguishing the 467 special lncRNAs from other lncRNAs. By inspecting the performance of the tools on the 467 transcripts, we noticed that all tools except CREMA, BASiNET and NCResNet showed a very high recall (>90%) to classify this set of lncRNAs (Supplementary Table S5 available online at https://academic.oup.com/bib), indicating that these lncRNAs are different from mRNAs. In the future, we should explore new features to distinguish them from other lncRNAs.

Based on our analysis, deep learning-based tools showed a higher performance than other tools. Although these tools utilize GPU for training, they provide utilities that can be run without GPU. However, the tools still require higher memory and run time, which are more likely to become the bottleneck when the user is trying to run them on a large dataset. With the availability of large-scale computational resources, this may not remain a problem anymore. Still, in order to efficiently utilize the memory, users may need to split the data and run the tools on each split separately.

If the user focuses on human and mouse datasets, lncRNAnet, LncADeep and PredLnc-GFStack can be the top choice, according to their high overall performance and medium level run time. CPAT can serve as an alternative, especially when the user lacks the required computational resource (CPU core count and memory) or deals with a very large dataset, since CPAT can achieve top-tier performance with a low consumption of computational resources. CPAT also provides a web server that is convenient to test small-sized datasets. If the input data mostly contain very short sequences (<200 nt), none of the 17 tools analyzed in this study can provide reliable performance, as the short sequences are highly likely to be misclassified as the non-coding class. For species with no ready-to-use model, Lncident, LncADeep, CPAT, LncFinder, CPC2 and CPPred can be good choices (Supplementary Table S4 available online at https://academic.oup.com/bib). The availability links and the installation and execution steps of the 17 tools are also provided in the supplementary file to help the users (Supplementary Section 2 available online at https://academic.oup.com/bib).

Key Points

- A systematic evaluation of computational tools on lncRNA identification is necessary.
- Seventeen tools on lncRNA identification are assessed on a set of unified data in human and mouse in terms of accuracy, run time and memory consumption.
- Common features used by the popular tools are analyzed based on the feature score distribution and the importance of the features on lncRNA identification.
- The peptide features do not contribute much to the accuracy of the tools in lncRNA identification.
- Current tools do not perform well on context-specific lncRNA identification.

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bib.

Authors' contributions

H.H. and X.L. conceived the idea. H.Z. and A.T. implemented the idea and generated results. H.Z., A.T., X.L. and H.H. analyzed the results and wrote the manuscript. All authors reviewed the manuscript.

Conflict of Interest

We declare that there is no conflict of interest regarding the publication of this article.

Funding

National Science Foundation (1661414, 2015838).

References

1. Dunham I, Consortium EP. An integrated encyclopedia of DNA elements in the human genome. *Nature* 2012;**489**:57–74.

2. Djebali S, Davis CA, Merkel A, *et al*. Landscape of transcription in human cells. *Nature* 2012;**489**:101–8.
3. Venter JC, Adams MD, Myers EW, *et al*. The sequence of the human genome. *Science* 2001;**291**:1304–51.
4. Kapranov P, Cheng J, Dike S, *et al*. RNA maps reveal new RNA classes and a possible function for pervasive transcription. *Science* 2007;**316**:1484–8.
5. Xu Z, Wei W, Gagneur J, *et al*. Bidirectional promoters generate pervasive transcription in yeast. *Nature* 2009;**457**:1033–7.
6. Brockdorff N, Ashworth A, Kay GF, *et al*. The product of the mouse Xist gene is a 15 kb inactive X-specific transcript containing no conserved ORF and located in the nucleus. *Cell* 1992;**71**(3):515–26.
7. Brown CJ, Lafreniere RG, Powers VE, *et al*. Localization of the X inactivation centre on the human X chromosome in Xq13. *Nature* 1991;**349**:82–4.
8. Hung T, Chang HY. Long noncoding RNA in genome regulation: prospects and mechanisms. *RNA Biol* 2010;**7**: 582–5.
9. Johnson JM, Edwards S, Shoemaker D, *et al*. Dark matter in the genome: evidence of widespread transcription detected by microarray tiling experiments. *Trends Genet* 2005;**21**:93–102.
10. Kornienko AE, Guenzl PM, Barlow DP, *et al*. Gene regulation by the act of long non-coding RNA transcription. *BMC Biol* 2013;**11**:59.
11. Malecová B, Morris KV. Transcriptional gene silencing through epigenetic changes mediated by non-coding RNAs. *Curr Opin Mol Ther* 2010;**12**:214–22.
12. Zhou KI, Parisien M, Dai Q, *et al*. N(6)-Methyladenosine modification in a long noncoding RNA hairpin predisposes its conformation to protein binding. *J Mol Biol* 2016;**428**:822–33.
13. Szczesniak MW, Makalowska I. lncRNA-RNA interactions across the human transcriptome. *PLoS One* 2016;**11**:e0150353.
14. Wapinski O, Chang HY. Long noncoding RNAs and human disease. *Trends Cell Biol* 2011;**21**:354–61.
15. Guttman M, Amit I, Garber M, *et al*. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature* 2009;**458**:223–7.
16. Cabili M, Trapnell C, Goff L, *et al*. Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. *Genes Dev* 2011;**25**:1915–27.
17. Derrien T, Johnson R, Bussotti G, *et al*. The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. *Genome Res* 2012;**22**:1775–89.
18. Sunwoo H, Dinger ME, Wilusz JE, *et al*. MEN epsilon/beta nuclear-retained non-coding RNAs are up-regulated upon muscle differentiation and are essential components of paraspeckles. *Genome Res* 2009;**19**:347–59.
19. Standaert L, Adriaens C, Radaelli E, *et al*. The long noncoding RNA Neat1 is required for mammary gland development and lactation. *RNA* 2014;**20**:1844–9.
20. Wang Z, Katsaros D, Biglia N, *et al*. High expression of long non-coding RNA MALAT1 in breast cancer is associated with poor relapse-free survival. *Breast Cancer Res Treat* 2018;**171**:261–71.
21. Tang Q, Hann SS. HOTAIR: an oncogenic long noncoding RNA in human cancer. *Cell Physiol Biochem* 2018;**47**: 893–913.
22. Pisignano G, Pavlaki I, Murrell A. Being in a loop: how long non-coding RNAs organise genome architecture. *Essays Biochem* 2019;**63**:177–86.
23. Xie ZY, Wang P, Wu YF, *et al*. Long non-coding RNA: the functional regulator of mesenchymal stem cells. *World J Stem Cells* 2019;**11**:167–79.
24. Sulaiman SA, Muhsin NIA, Jamal R. Regulatory non-coding RNAs network in non-alcoholic fatty liver disease. *Front Physiol* 2019;**10**:279.
25. Duenas A, Exposito A, Aranega A, *et al*. The role of non-coding RNA in congenital heart diseases. *J Cardiovasc Dev Dis* 2019;**6**:15.
26. Zhou F, Chen W, Jiang Y, *et al*. Regulation of long non-coding RNAs and circular RNAs in spermatogonial stem cells. *Reproduction* 2019;**158**:R15–25.
27. Ghafouri-Fard S, Taheri M. Nuclear Enriched Abundant Transcript 1 (NEAT1): a long non-coding RNA with diverse functions in tumorigenesis. *Biomed Pharmacother* 2019;**111**:51–9.
28. Li ZX, Zhu QN, Zhang HB, *et al*. MALAT1: a potential biomarker in cancer. *Cancer Manag Res* 2018;**10**: 6757–68.
29. Dinger ME, Amaral PP, Mercer TR, *et al*. Pervasive transcription of the eukaryotic genome: functional indices and conceptual implications. *Brief Funct Genomic Proteomic* 2009;**8**:407–23.
30. Fatica A, Bozzoni I. Long non-coding RNAs: new players in cell differentiation and development. *Nat Rev Genet* 2014;**15**:7–21.
31. Villegas VE, Zaphiropoulos PG. Neighboring gene regulation by antisense long non-coding RNAs. *Int J Mol Sci* 2015;**16**:3251–66.
32. Ulitsky I, Bartel DP. lincRNAs: genomics, evolution, and mechanisms. *Cell* 2013;**154**:26–46.
33. Yan P, Luo S, Lu JY, *et al*. Cis- and trans-acting lncRNAs in pluripotency and reprogramming. *Curr Opin Genet Dev* 2017;**46**:170–8.
34. Kopp F, Mendell JT. Functional classification and experimental dissection of long noncoding RNAs. *Cell* 2018;**172**:393–407.
35. Dinger ME, Amaral PP, Mercer TR, *et al*. Long noncoding RNAs in mouse embryonic stem cell pluripotency and differentiation. *Genome Res* 2008;**18**:1433–45.
36. Mortazavi A, Williams BA, McCue K, *et al*. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods* 2008;**5**:621–8.
37. Furuno M, Pang KC, Ninomiya N, *et al*. Clusters of internally primed transcripts reveal novel long noncoding RNAs. *PLoS Genet* 2006;**2**:e37.
38. Rudkin GT, Stollar B. High resolution detection of DNA–RNA hybrids in situ by indirect immunofluorescence. *Nature* 1977;**265**:472–3.
39. Siomi H, Siomi MC. On the road to reading the RNA-interference code. *Nature* 2009;**457**:396–404.
40. Zhu JJ, Fu HJ, Wu YG, *et al*. Function of lncRNAs and approaches to lncRNA-protein interactions. *Sci China Life Sci* 2013;**56**:876–85.
41. Han S, Liang Y, Li Y, *et al*. Long noncoding RNA identification: comparing machine learning based tools for long noncoding transcripts discrimination. *Biomed Res Int* 2016;**2016**:8496165.
42. Iwakiri J, Hamada M, Asai K. Bioinformatics tools for lncRNA research. *Biochim Biophys Acta* 2016;**1859**: 23–30.

43. Pinkney HR, Wright BM, Diermeier SD. The lncRNA toolkit: databases and in silico tools for lncRNA analysis. *Noncoding RNA* 2020;**6**:49.
44. Statello L, Guo CJ, Chen LL, *et al*. Gene regulation by long non-coding RNAs and its biological functions. *Nat Rev Mol Cell Biol* 2021;**22**:96–118.
45. Fritah S, Niclou SP, Azuaje F. Databases for lncRNAs: a comparative evaluation of emerging tools. *RNA* 2014; **20**:1655–65.
46. Veneziano D, Marceca GP, Di Bella S, *et al*. Investigating miRNA-lncRNA interactions: computational tools and resources. *Methods Mol Biol* 2019;**1970**:251–77.
47. Choudhari R, Sedano MJ, Harrison AL, *et al*. Long noncoding RNAs in cancer: from discovery to therapeutic targets. *Adv Clin Chem* 2020;**95**:105–47.
48. Prensner JR, Chinnaiyan AM. The emergence of lncRNAs in cancer biology. *Cancer Discov* 2011;**1**:391.
49. Yan C, Zhang Z, Bao S, *et al*. Computational methods and applications for identifying disease-associated lncRNAs as potential biomarkers and therapeutic targets. *Mol Ther Nucleic Acids* 2020;**21**:156–71.
50. Ma L, Bajic VB, Zhang Z. On the classification of long noncoding RNAs. *RNA Biol* 2013;**10**:925–33.
51. Dahariya S, Paddibhatla I, Kumar S, *et al*. Long non-coding RNA: classification, biogenesis and functions in blood cells. *Mol Immunol* 2019;**112**:82–92.
52. Tang B, Pan Z, Yin K, *et al*. Recent advances of deep learning in bioinformatics and computational biology. *Front Genet* 2019;**10**:214.
53. Baek J, Lee B, Kwon S, *et al*. LncRNAnet: long noncoding RNA identification using deep learning. *Bioinformatics* 2018;**34**:3889–97.
54. Fan XN, Zhang SW, Zhang SY, *et al*. Lncrna_mdeep: an alignment-free predictor for distinguishing long noncoding RNAs from protein-coding transcripts by multimodal deep learning. *Int J Mol Sci* 2020;**21**:1–11.
55. Guo JC, Fang SS, Wu Y, *et al*. CNIT: a fast and accurate web tool for identifying protein-coding and long noncoding transcripts based on intrinsic sequence composition. *Nucleic Acids Res* 2019;**47**:W516–22.
56. Han S, Liang Y, Li Y, *et al*. Lncident: a tool for rapid identification of long noncoding RNAs utilizing sequence intrinsic composition and open reading frame information. *Int J Genom* 2016;**2016**:1–11.
57. Han S, Liang Y, Ma Q, *et al*. LncFinder: an integrated platform for long non-coding RNA identification utilizing sequence intrinsic composition, structural information and physicochemical property. *Brief Bioinform* 2019; **20**:2009–27.
58. Kang YJ, Yang DC, Kong L, *et al*. CPC2: a fast and accurate coding potential calculator based on sequence intrinsic features. *Nucleic Acids Res* 2017;**45**:W12–6.
59. Li A, Zhang J, Zhou Z. PLEK: a tool for predicting long noncoding RNAs and messenger RNAs based on an improved k-mer scheme. *BMC Bioinformatics* 2014;**15**:1–10.
60. Liu S, Zhao X, Zhang G, *et al*. Predlnc-gfstack: a global sequence feature based on a stacked ensemble learning method for predicting lncRNAs from transcripts. *Genes* 2019;**10**:672.
61. Simopoulos CMA, Weretilnyk EA, Golding GB. Prediction of plant lncRNA by ensemble machine learning classifiers. *BMC Genomics* 2018;**19**:316.
62. Sun L, Luo H, Bu D, *et al*. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. *Nucleic Acids Res* 2013;**41**:e166.
63. Tong X, Liu S. CPPred: coding potential prediction based on the global description of RNA sequence. *Nucleic Acids Res* 2019;**47**:e43.
64. Wang G, Yin H, Li B, *et al*. Characterization and identification of long non-coding RNAs based on feature relationship. *Bioinformatics* 2019;**35**:2949–56.
65. Wang L, Park HJ, Dasari S, *et al*. CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model. *Nucleic Acids Res* 2013;**41**:1–7.
66. Wucher V, Legeai F, Rizk G, *et al*. FEELnc: a tool for long non-coding RNA annotation and its application to the dog transcriptome. *Nucleic Acids Res* 2017;**45**:1–12.
67. Yang C, Yang L, Zhou M, *et al*. LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning. *Bioinformatics* 2018;**34**:3825–34.
68. Ito EA, Katahira I, Vicente FFR, *et al*. BASiNET-BiologicAl Sequences NETwork: a case study on coding and noncoding RNAs identification. *Nucleic Acids Res* 2018;**46**:e96–6.
69. Yang S, Wang Y, Zhang S, *et al*. NCResNet: noncoding ribonucleic acid prediction based on a deep resident network of ribonucleic acid sequences. *Front Genet* 2020;**11**:90.
70. Ding J, Cai X, Wang Y, *et al*. ChIPModule: systematic discovery of transcription factors and their cofactors from ChIP-seq data. *Pac Symp Biocomput* Hawaii, USA: Kohala Coast, 2013;320–31.
71. Ding J, Dhillon V, Li X, *et al*. Systematic discovery of cofactor motifs from ChIP-seq data by SIOMICS. *Methods* 2015;**79-80**:47–51.
72. Ding J, Li X, Hu H. Systematic prediction of cis-regulatory elements in the *Chlamydomonas reinhardtii* genome using comparative genomics. *Plant Physiol* 2012;**160**:613–23.
73. Anderson DM, Anderson KM, Chang CL, *et al*. A micropeptide encoded by a putative long noncoding RNA regulates muscle performance. *Cell* 2015;**160**:595–606.
74. Bi P, Ramirez-Martinez A, Li H, *et al*. Control of muscle formation by the fusogenic micropeptide myomixer. *Science* 2017;**356**:323–7.
75. D'Lima NG, Ma J, Winkler L, *et al*. A human microprotein that interacts with the mRNA decapping complex. *Nat Chem Biol* 2017;**13**:174–80.
76. Hanyu-Nakamura K, Sonobe-Nojima H, Tanigawa A, *et al*. Drosophila Pgc protein inhibits P-TEFb recruitment to chromatin in primordial germ cells. *Nature* 2008;**451**:730–3.
77. Huang JZ, Chen M, Chen, *et al*. A peptide encoded by a putative lncRNA HOXB-AS3 suppresses colon cancer growth. *Mol Cell* 2017;**68**:171–184 e176.
78. Kondo T, Hashimoto Y, Kato K, *et al*. Small peptide regulators of actin-based cell morphogenesis encoded by a polycistronic mRNA. *Nat Cell Biol* 2007;**9**:660–5.
79. Magny EG, Pueyo JI, Pearl FM, *et al*. Conserved regulation of cardiac calcium uptake by peptides encoded in small open reading frames. *Science* 2013;**341**:1116–20.
80. Matsumoto A, Pasut A, Matsumoto M, *et al*. mTORC1 and muscle regeneration are regulated by the LINC00961-encoded SPAR polypeptide. *Nature* 2017;**541**:228–32.
81. Nelson BR, Makarewich CA, Anderson DM, *et al*. A peptide encoded by a transcript annotated as long noncoding RNA enhances SERCA activity in muscle. *Science* 2016;**351**: 271–5.
82. Pauli A, Norris ML, Valen E, *et al*. Toddler: an embryonic signal that promotes cell movement via Apelin receptors. *Science* 2014;**343**:1248636.
83. Röhrig H, Schmidt J, Miklashevichs E, *et al*. Soybean ENOD40 encodes two peptides that bind to sucrose synthase. *Proc Natl Acad Sci* 2002;**99**:1915–20.

84. Zhang Q, Vashisht AA, O'Rourke J, *et al*. The microprotein minion controls cell fusion and muscle formation. *Nat Commun* 2017;**8**:15664.
85. Fickett JW, Tung CS. Assessment of protein coding measures. *Nucleic Acids Res* 1992;**20**:6441–50.
86. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Comput* 2006;**18**:1527–54.
87. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science* 2006;**313**:504–7.
88. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;**521**:436–44.
89. Makantasis K, Karantzalos K, Doulamis A, *et al*. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. Milan, Italy: IEEE, 2015, 4959–62.
90. Geirshick R. Fast R-CNN. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, 2015, 1440–8.
91. He K, Gkioxari G, Dollár P, *et al*. Mask R-CNN. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy: IEEE, 2017, 2980–8.
92. LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. In: *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA: MIT Press, 1998, 255–8.
93. Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. *Neural Inf Process Syst* 2012;**25**:1097–105.
94. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. *IEEE Trans Signal Process* 1997;**45**:2673–81.
95. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;**9**:1735–80.
96. Wang Y, Goodison S, Li X, *et al*. Prognostic cancer gene signatures share common regulatory motifs. *Sci Rep* 2017;**7**:4750.
97. Talukder A, Saadat S, Li X, *et al*. EPIP: a novel approach for condition-specific enhancer–promoter interaction prediction. *Bioinformatics* 2019;**35**:3877–83.
98. Barham C, Cha M, Li X, *et al*. Application of deep learning models to microRNA transcription start site identification. In: *2019 IEEE 7th International Conference on Bioinformatics and Computational Biology (ICBCB)*. Zhejiang University, Hangzhou, China: IEEE, 2019, 22–8.
99. Cha M, Zheng H, Talukder A, *et al*. A two-stream convolutional neural network for microRNA transcription start site feature integration and identification. *Sci Rep* 2021; **11**:5625.
100. Dey R, Salem FM. Gate-variants of Gated Recurrent Unit (GRU) neural networks. In: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. Boston, MA, USA: IEEE, 2017, 1597–600.
101. Chung J, Gulcehre C, Cho K *et al*. Empirical evaluation of gated recurrent neural networks on sequence modeling. In: *In NIPS 2014 Workshop on Deep Learning*. Montreal, Quebec, Canada: NIPS, 2014.
102. Berrar D, Dubitzky W. Deep learning in bioinformatics and biomedicine. *Brief Bioinform* 2021;**22**:1513–4.
103. Karim MR, Beyan O, Zappa A, *et al*. Deep learning-based clustering approaches for bioinformatics. *Brief Bioinform* 2021;**22**:393–415.
104. Zheng H, Li X, Hu H. Deep Learning to Identify Transcription Start Sites from CAGE Data. In: *International Conference on Bioinformatics and Biomedicine, BIBM 2020, Virtual Event, South Korea, 2020*. IEEE.
105. Kong L, Zhang Y, Ye Z-Q, *et al*. CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Res* 2007;**35**:W345–9.
106. Spurlock CF, Tossberg JT, Guo Y, *et al*. Expression and functions of long noncoding RNAs during human T helper cell differentiation. *Nat Commun* 2015;**6**:6932.
107. Duan Y, Zhang W, Cheng Y, *et al*. A systematic evaluation of bioinformatics tools for identification of long noncoding RNAs. *RNA* 2021;**27**:80–98.
108. Heiman GW. *Understanding Research Methods and Statistics: an integrated introduction for psychology*. Houghton: Mifflin and Company, 2001.
109. Guyon I, Weston J, Barnhill S, *et al*. Gene selection for cancer classification using support vector machines. *Mach Learn* 2002;**46**:389–422.
110. Quinlan JR. Induction of decision trees. *Mach Learn* 1986;**1**:81–106.
111. Kullback S, Leibler RA. On information and sufficiency. *Ann Math Stat* 1951;**22**:79–86.
112. Talukder A, Barham C, Li X, *et al*. Interpretation of deep learning in genomics and epigenomics. *Brief Bioinform* 2020;**2020**:1–16.
113. Choi S-W, Kim H-W, Nam J-W. The small peptide world in long noncoding RNAs. *Brief Bioinform* 2019;**20**:1853–64.
114. Dragomir MP, Manyam GC, Ott LF, *et al*. FuncPEP: a database of functional peptides encoded by non-coding RNAs. *Non-coding RNA* 2020;**6**:41.