doi: 10.1093/nargab/lqz024

RNAsamba: neural network-based assessment of the protein-coding potential of RNA sequences

Antonio P. Camargo[1], Vsevolod Sourkov[2], Gonçalo A. G. Pereira[1] and Marcelo F. Carazzolle[1,*]

[1]Department of Genetics, Evolution, Microbiology and Immunology, Institute of Biology, University of Campinas, Campinas, SP, 13083-862, Brazil and [2]Department of Computer Science, ReDNA Labs, Pattaya, Chonburi, 20150, Thailand

Received August 09, 2019; Revised November 15, 2019; Editorial Decision December 10, 2019; Accepted December 17, 2019

ABSTRACT

The advent of high-throughput sequencing technologies made it possible to obtain large volumes of genetic information, quickly and inexpensively. Thus, many efforts are devoted to unveiling the biological roles of genomic elements, being the distinction between protein-coding and long non-coding RNAs one of the most important tasks. We describe RNAsamba, a tool to predict the coding potential of RNA molecules from sequence information using a neural network-based that models both the whole sequence and the ORF to identify patterns that distinguish coding from non-coding transcripts. We evaluated RNAsamba's classification performance using transcripts coming from humans and several other model organisms and show that it recurrently outperforms other state-of-the-art methods. Our results also show that RNAsamba can identify coding signals in partial-length ORFs and UTR sequences, evidencing that its algorithm is not dependent on complete transcript sequences. Furthermore, RNAsamba can also predict small ORFs, traditionally identified with ribosome profiling experiments. We believe that RNAsamba will enable faster and more accurate biological findings from genomic data of species that are being sequenced for the first time. A user-friendly web interface, the documentation containing instructions for local installation and usage, and the source code of RNAsamba can be found at https://rnasamba.lge.ibi.unicamp.br/.

INTRODUCTION

High-throughput sequencing technology has enabled the sequencing of genomes and transcriptomes of a myriad of species, yielding large quantities of genetic information (1). Hence, great effort is dedicated to characterize the obtained data, mainly by the identification of functional genomic elements such as messenger RNAs (mRNAs) and long non-coding RNAs (lncRNAs).

Due to their role of carriers of protein synthesis information, mRNAs have been studied for several decades and are well represented in genetic databases. In contrast, lncRNAs, which are defined as transcripts >200 nucleotides that are not translated into proteins (2), have been known for much less time and only recently their role as regulators of gene expression and their link to genetic diseases has been unveiled.

One of the main goals of the functional annotation of genomes and transcriptomes is the identification of mRNAs and lncRNAs. Over the last two decades, a massive effort was conducted by the ENCODE and GENCODE projects to identify and characterize all functional elements of the human and mouse genomes, including mRNAs and lncRNAs, using a range of different sequencing data and manual curation procedures (3,4). For non-model organisms, however, the annotation of such elements usually depends solely on computational inferences.

In the vast majority of genome annotation projects, the characterization of genomic elements relies on the comparison of sequences or structures with databases of biological sequences, which is very time-consuming (5) and poses limitations for both the annotation of mRNAs and lncRNAs. As only a fraction of the genetic diversity existing in nature is known and available in databases, many new protein-coding genes are not identified because their protein product is not found among existing data (6). On the other side, as lncRNAs are not under the same evolutionary constraints as mRNAs, they display lower sequence conservation than protein-coding transcripts (7,8), resulting in failure to find homologous sequences in database searches (9,10).

Even though mRNAs and lncRNAs usually share many molecular features (11,12), they display contrasting sequence properties that can be used to create statistical models capable of computing the coding potential of any given

*To whom correspondence should be addressed. Tel: +55 19 3521 6651; Email: mcarazzo@unicamp.br

transcript without the limitations of database-based annotation pipelines. Most of these approaches employ machine learning algorithms to differentiate coding and non-coding transcripts based on a series of human-designed sequence features such as ORF length and integrity (13,14), GC-content (11), 3-base periodicity (15), *k*-mer frequencies (16,17) and hexamer usage bias (18). However, the usage of these features may introduce bias to the classification, causing, for instance, the models to misclassify lncRNAs possessing long ORFs and coding transcripts containing short or truncated ORFs.

The power of multi-layered neural networks to identify deep patterns has made them the *de facto* standard in many machine learning applications, such as image and text analysis, and have been extensively employed in bioinformatics to provide new biological insights (19). Contrasting to conventional machine learning algorithms, deep learning approaches do not necessarily depend on human-designed features and can be used to capture concealed sequence signals that are fundamentally different between mRNAs and lncRNAs.

Here we describe RNAsamba, a tool that uses a novel neural network architecture to tackle the mRNA/lncRNA classification problem relying solely on sequence information. We show that our method outperforms previous tools in a variety of metrics, can be used to classify transcripts from a range of different species and is robust to limitations commonly found in real world data, such as truncated ORFs.

BACKGROUND

Sequence modeling with neural networks

Recurrent neural networks (RNNs) are a type of neural network in which each node takes the output of a previous node as input, forming a directed graph. This architecture confers RNNs the property of remembering previous states, making them ideal to deal with sequential data such as nucleotide sequences (19). One well documented drawback of traditional RNNs is the issue of long-range dependencies, which hinders the training of networks with sequences longer than a few hundred elements and makes it difficult to train RNNs with long sequences (20). To tackle this problem, the recently introduced IGLOO (21) architecture looks at sequences as a whole rather than sequentially like in the recurrent paradigm. To do so, IGLOO creates representations of sequences via the multiplication of sequence patches by learnable weights (Figure 1A).

In an IGLOO layer, input sequences are of shape (L, M), where L is the length of the sequence and M is feature size, i.e. the size of the representation of the element at a given position. IGLOO uses an initial 1-D convolutional layer and max pooling to transform the input into a (L, M*)-shaped array, which can be scaled to accommodate for the overall size of the network. Then, IGLOO iteratively collects K patches, each containing 4 random matrix slices, which are multiplied by a matrix of learnable weights, resulting in a K-sized representation of the sequence. Intuitively, the weight learns relationships between non-necessarily contiguous slices of the sequence representation. Using K of those weights allows the network to find a new sequence representation composed of K different non-local relationships. This representation can then be fed to a dense layer for classification.

By taking global snapshots of the sequence, IGLOO networks can be used to process very long sequences, making them particularly interesting for nucleotide sequence data. Furthermore, IGLOO layers can be easily parallelized and run significantly faster than RNN variants, such as GRUs and LSTMs, for a similar number of trainable parameters.

Coding potential computation approaches

Current coding potential assessment tools fall into one of two categories: the ones that depend on information other than the transcript sequence alone and the ones that only use sequence-derived information (22).

The methods that fall into the first category use external data along with sequence-derived information to classify transcripts into mRNAs or lncRNAs. For instance, COME (23) uses a variety of sequence conservation, genomic context and experiment-based features; lncScore and lncRScan-SVM (24) use splicing information; CPC (25) and LncADeep (22) search the translated transcript sequences in protein databases to identify conserved domains. Even though the usage of external information may improve the detection of coding signals, it introduces dependencies on reliable annotations, which are usually not available for non-model organisms, and on a time-consuming database searches.

Methods within the second category, on the other hand, only use intrinsic sequence information to distinguish between mRNAs and lncRNAs. They can be further divided into two groups, depending on the approach used to assess transcript features. The first group comprises algorithms that extract explicitly defined features from nucleotide sequences and feed them to classic machine learning algorithms. Examples of methods belonging to this group include CPAT (26), CPC2 (27) and FEELnc (28). The second group encompasses tools, such as lncRNAnet (29) and mRNN (30), which model transcript sequences using neural networks, and thus are not strictly dependent on human-engineered features.

ALGORITHM

Starting from the initial nucleotide sequence, RNAsamba computes the coding potential of a given transcript by combining information coming from two different sources (Figure 1B): the Whole Sequence Branch (B1) and the Longest ORF Branch (B2). B1 contains whole sequence representations of the transcript and can capture protein-coding signatures irrespective of the identification of the ORF. In contrast, B2 carries information extracted from the longest identified ORF and the putative protein translated from it. By taking into account these two sources of sequence information, RNAsamba builds a thorough representation of the transcript, improving the classification performance of the algorithm.

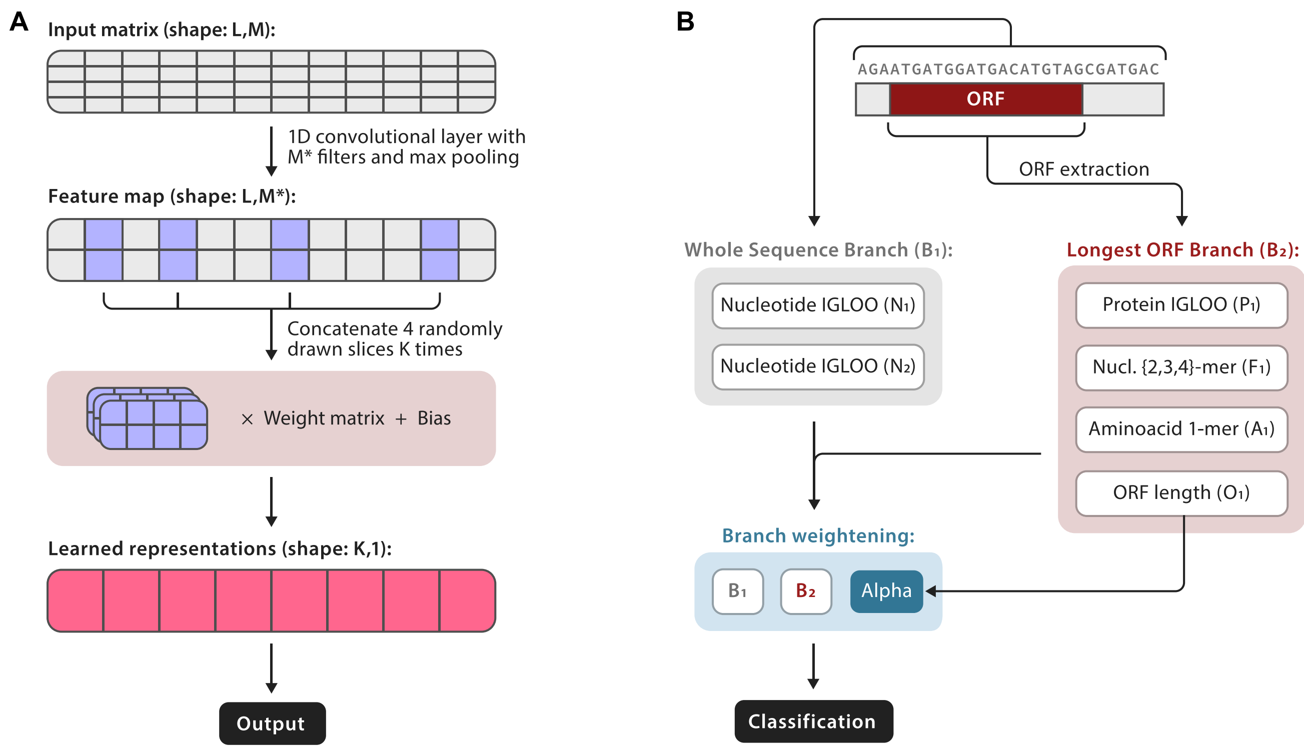

Figure 1. (**A**) In an IGLOO layer, the input sequence is initially processed by an 1D convolutional layer and down-sampled using the max pooling approach. From the resulting matrix, K patches consisting of four random slices are drawn from the matrix and then multiplied by matrix of K learnable weights, producing a high-level representation of the sequence input. (**B**) From the RNA sequence RNAsamba derives two branches. In the Whole Sequence Branch (B1), the whole transcript nucleotide sequence is fed to two IGLOO layers to create high-level representations of the transcript (N1 and N2). In the Longest ORF Branch (B2), four layers are derived from the extracted ORF sequence: an IGLOO representation of the putative protein (P1), nucleotide *k*-mer frequencies (F1), amino acid frequencies (A1) and the ORF length (O1). The two branches are weighted by the α parameter and then used to compute the final classification of the transcript.

ORF extraction

To obtain ORF information to feed to B2, RNAsamba scans each of the three reading frames looking for fragments that initiate with a start codon (ATG) and finish either with a stop codon (TAG, TAA or TGA) or at the end of the transcript. The longest fragment among the ones found in all reading frames is then extracted, regardless of finishing with a stop codon or not. In case no start codon is found, B2 will carry no information and the classification will solely depend on B1.

Sequence pre-processing and encoding

RNAsamba generates high-level representations of both nucleotide and amino acid sequences using IGLOO units. As these units require fixed length sequences as input, transcript and protein sequences are truncated to a maximum length of 3000 nucleotides and 1000 amino acids, respectively. Even though these thresholds were arbitrarily chosen, we observed that, while using them, the algorithm exhibits faster training times and can capture enough information to correctly classify very long transcripts (Supplementary Table S1). We believe that this is because the region that contributes the most to classification is located right after the start codon (30). The sequences are then converted into numeric representations as follows:

$$\text{Nucleotide}: \; ATGACT\cdots \rightarrow (1, 2, 4, 1, 3, 2, \cdots)$$

$$\text{Aminoacid}: \; MTGQLV\cdots \rightarrow (19, 10, 5, 11, 1, 7, \cdots)$$

Finally, nucleotide and protein sequences shorter than the maximum length threshold are then zero-padded to 3000 and 1000 elements, respectively.

Whole Sequence Branch (B1)

To obtain high-level representations of the transcript, the whole nucleotide sequence is inputted into two independent stacked IGLOO units, N1 and N2, with K1 (K1 = 900) patches and distinct kernel sizes in their initial convolutional layers. The outputs of these units are then concatenated and fed to a dense layer resulting in B1.

Longest ORF Branch (B2)

B2 is the result of the combination of four different layers that carry different properties of the ORF sequence. Layer P1 contains a representation of the protein sequence and is obtained by inputting the amino acid sequence of the putative protein into an stacked IGLOO layer with K2 (K2 =

600) patches; layer F1 is comprised of the relative frequencies of nucleotide k-mers ($k \subset \{2, 3, 4\}$) in the ORF; layer A1 contains the relative amino acid frequency of the translated ORF; layer O1 consists of the length of the longest identified ORF. B2 is obtained by feeding P1, F1, A1 and O1 to four independent dense layers, concatenating the outputs into a single matrix that is then fed a final dense layer.

Branch weightening

The branches B1 and B2 gather different information from the transcript: while B1 captures patterns from the whole transcript sequence, B2 picks up information specific to the ORF. Therefore, we include an attention mechanism, the α parameter, to weight information coming from these two branches. This mechanism is important, for instance, to correctly classify transcripts with unusual ORF length, such as non-coding transcripts with long ORFs or truncated protein-coding RNAs.

$$\alpha = \text{softmax}\left(\mathrm{O}_1 \cdot \mathrm{W}_1 + \mathrm{W}_2\right)$$

$$Y = \alpha \cdot B_1 + (1 - \alpha) \cdot B_2$$

Where W1 and W2 are trainable matrices and α is a matrix that is used to weight B1 and B2 in the final layer (Y). While training the algorithm end-to-end, the weights in W1 and W2 are optimized to maximize classification accuracy.

To obtain the coding score, Y is fed to a dense layer with a softmax activation that computes the probabilities for each class (31). Training is performed by minimizing the categorical cross-entropy using the Adam optimizer (32).

Training and classification routines

During training, sequences are pre-processed into numerical information and propagated through the neural network in batches, gradually adjusting the learnable weights to improve the model's classification performance. For the inference, input transcripts go through the same pre-processing steps but, instead of being used to update the network, their numerical representation is processed by the trained model to compute their coding potential.

IMPLEMENTATION

RNAsamba is written in Python and Rust and uses popular state-of-the-art deep learning libraries, TensorFlow (33) and Keras. We provide an installation and execution manual to make the process of training new models and classifying transcripts easy for the end user. For training new models, RNAsamba supports changing the number of epochs (the number of times each sample is visited) and batch size (the number of samples that are propagated through the network in each iteration). It also allows the user to enable early stopping, which is useful to avoid overfitting. For inference, our implementation allows the input of multiple weights files that are combined in an ensemble classification, also helping to reduce model variance.

RESULTS

RNAsamba can accurately distinguish mRNAs from lncRNAs in several datasets

To evaluate the ability of RNAsamba's algorithm to learn how to discriminate coding sequences from non-coding ones, we compared it with five state-of-the-art coding potential predictors that solely rely on intrinsic sequence information: CPAT, CPC2, FEELnc, lncRNAnet and mRNN. To keep the comparison as unbiased as possible, the benchmark was performed using four independent datasets consisting of coding and non-coding human transcripts previously used in the literature. These datasets exhibit differing characteristics regarding gene composition, balance, and transcript and ORF length distributions (Supplementary Table S2 and Supplementary Figure S1). In this evaluation, we found that RNAsamba largely outperforms the other predictors in almost every metric across all the datasets (Figure 2A and Supplementary Table S3).

As the comparison was performed with built-in models trained with different data, it cannot be used to evaluate the performance of models trained with the same set of mRNAs and lncRNAs. Therefore, we performed a second benchmark in which we trained new models using the train set corresponding to each the test dataset. In this evaluation, RNAsamba also displayed superior classification quality (Supplementary Table S4), being only outperformed by mRNN in the mRNN-Challenge dataset.

RNAsamba's model generalizes to different species

As human genes have been carefully annotated throughout the years, we believe RNAsamba's main value is the accurate identification of mRNAs and lncRNAs in novel genomes. Thus, in order to evaluate if RNAsamba's built-in model, trained with human RNA sequences, generalizes well to other species, we evaluated its performance in multiple test datasets, each containing both mRNAs and ncRNAs from one of five different species: *Mus musculus*, *Danio rerio*, *Drosophila melanogaster*, *Caenorhabditis elegans* and *Arabidopsis thaliana* (Supplementary Table S5 and Supplementary Figure S2). We also compared the performance of RNAsamba to five other algorithms pre-trained with human transcripts.

RNAsamba exhibits good classification performance in every species, irrespective of the evolutionary distance to humans, showing that a model learned from human sequence data can be generalized to different organisms. When compared to other software, RNAsamba recurrently is placed among the best tools, showing slightly worse results only in *D. rerio* and *D. melanogaster*, where it displays a reduction in precision (Figure 2B and Supplementary Table S6). Notably, mRNN exhibits a significant decrease in classification performance when compared to its results in human data, evidencing that its algorithm may not handle well RNA sequences from different species.

In a second benchmark, where every tool was trained with the same train data (CPC2's train dataset), RNAsamba also exhibited excellent classification performance (Supplementary Table S7). In this evaluation, we observed that RNAsamba's performance in the *D. rerio* and *D.*

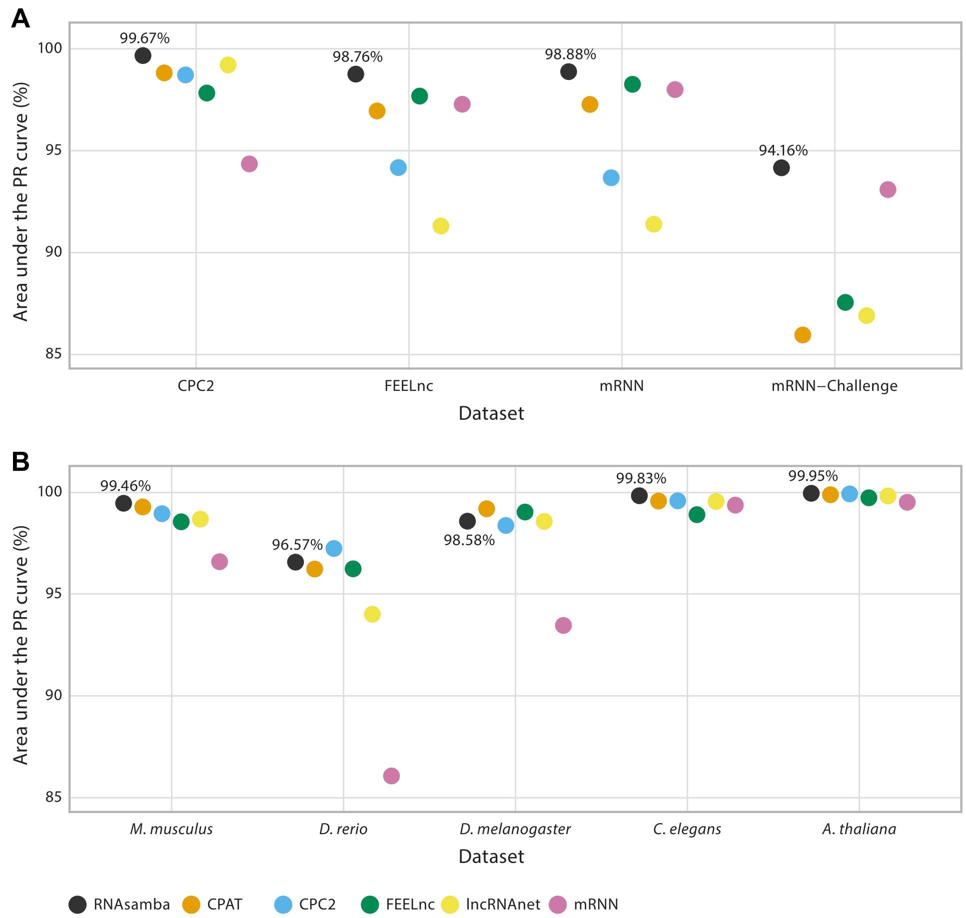

Figure 2. Classification benchmark of six different coding potential calculators. (**A**) Classifiers performance in four independent test datasets containing human transcripts. CPC2 is outside of the displayed range in the mRNN-Challenge test dataset (75.35%). (**B**) Classifiers performance in five different species. Values correspond to the area under the precision-recall curve. Pre-trained models provided by the authors of each tool were used.

melanogaster test data improved, showing that its model is not inherently worse for these species.

As the precision of every software was worse in the *D. rerio* and *D melanogaster* data when compared to the data of other species (Supplementary Tables S6 and S7), we hypothesized that this discrepancy could be partly due to proteins being misannotated as lncRNAs in the datasets. To test this hypothesis, we reannotated lncRNAs that RNAsamba classified as protein-coding (401 and 1185 transcripts for *D. rerio* and *D. melanogaster*, respectively) by comparing these sequences to established protein databases (UniRef90, CDD and Pfam). Surprisingly, a large fraction (83.5% and 67.8% for *D. rerio* and *D. melanogaster*, respectively) of these lncRNAs had significant hits to protein sequences or protein domains in at least one of databases (Supplementary Data). For *D. rerio*, we identified several complete and highly conserved kinases and immunoglobulin domains. For *D. melanogaster*, several putative casein kinases, important to signal transduction regulation, were identified. Even though our annotation process lacks careful validation and is not sufficient to confidently assign protein-coding properties to these transcripts, our results suggest that RNAsamba can be used as a line of evidence to identify misannotations in published genomes.

RNAsamba can identify truncated coding sequences

Since it constitutes the coding portion of the RNA, the ORF is generally used as the main source of information to detect potential protein-coding transcripts. Because of that, most mRNA/lncRNA classifiers use human-engineered features extracted from the coding portion of the transcript, such as the ORF length and coverage. This dependence on a detectable in-frame ORF to identify coding sequences impairs the function of these algorithms to annotate the majority of transcriptome datasets, which contain a large fraction of partial-length transcripts (9,34,35).

As the B1 branch of RNAsamba captures sequence information that is independent of the ORF, it can detect protein-coding signatures even in the absence of a start codon. Thus, we tested the algorithm's performance in the identification of truncated mouse and *A. thaliana* mRNA transcripts, in which both the start and stop codon are absent. To avoid biases caused by the detection of a fragment of the true ORF, we also evaluated RNAsamba's performance in separate sets of truncated mouse and *A. thaliana* transcripts that possess no in-frame start codon inside the ORF, meaning that the model would have to capture ORF-independent coding marks to identify mRNAs. For this test,

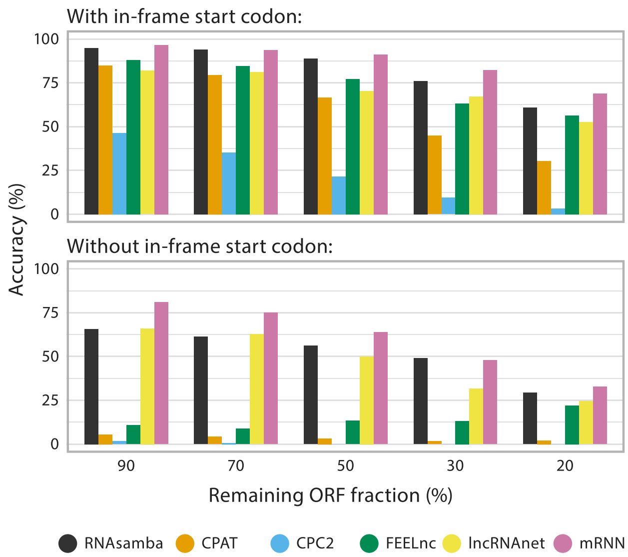

Figure 3. Evaluation of the ability of different tools to detect the coding potential of mouse ORFs with varying degrees of fragmentation.

we trained RNAsamba with both complete and truncated human sequences, aiming to provide users with model that is more capable of identifying mRNAs by looking at the whole sequence context.

Inspection of the fraction of identified mRNAs obtained from each stratum of truncated ORFs revealed that RNAsamba can identify a substantial fraction of the mRNAs even when most of the ORF is absent (Figure 3 and Supplementary Figure S3). We also noted a negative association between the amount of available ORF information and the median value of the α parameter, showing that RNAsamba favors B1 as ORF-derived data becomes sparse (Supplementary Figure S4).

When contrasted to three ORF-dependent algorithms, CPAT, CPC2 and FEELnc, RNAsamba displayed much better performance at identifying partial coding sequences. The discrepancy between RNAsamba and these algorithms is much more pronounced in the case of the truncated transcripts without in-frame start codons, as CPAT, CPC2 and FEELnc are incapable of finding fragments of the true ORF, making their predictions mostly unreliable. Even though FEELnc depends on the detection of the transcript ORF, it uses a relaxed definition of ORFs that makes it more capable of finding coding sequences when the canonical signals—start and stop codons—are absent (28). When compared to other algorithms that don't strictly rely on ORF sequences, RNAsamba displays better classification performance than lncRNAnet, but generally worse than the mRNN model. We suspect that mRNN's good performance in this specific kind of data is possibly due to the use of artificially introduced reading frame shifts during the data augmentation process (30).

RNAsamba can detect a translation-related sequence residing outside of the ORF

The Kozak consensus sequence, which spawns from the −6 to the +4 positions of mRNAs, is a recurring sequence in coding transcripts (36) and plays a major role in the initiation of the translation process (37), evidencing that portions of untranslated regions can affect translation efficiency. As RNAsamba uses whole-sequence information to process RNA sequence data, we investigated whether its algorithm is sensitive to changes in the Kozak sequence region.

Thus, for each of 1000 randomly chosen mouse mRNAs, we derived two sets containing 100 computer-generated transcripts each. In the control set, new sequences were created by replacing the Kozak sequence region of the mRNA by fragments generated by sampling nucleotides from a uniform probability distribution. In contrast, nucleotides of the computer-generated fragments of the second set were sampled according to the probability distribution of the Kozak consensus sequence (Supplementary Figure S5).

We found that Kozak-derived sequences lead to an overall increase of transcripts' coding score. In the majority of the tested transcripts (77.71%), this score was significantly larger (FDR-adjusted P-value $\leq$ 0.05) in fragments generated from the Kozak consensus probability distribution, indicating that RNAsamba is able to detect an important signal that contributes to mRNA translation even though it mostly resides outside of the ORF. Accordingly, we observed that there is a significant (P-value $\approx$ 0.01) negative correlation between the coding score of a given sequence and the Hamming distance between its computer-generated portion and the Kozak sequence consensus.

We also investigated whether the effect of the Kozak sequence on the coding score is diminished in longer sequences, since they intrinsically carry larger amounts of information to be processed by the RNAsamba algorithm. We noticed that for transcripts longer than a well-defined threshold, around 3160 base pairs (bp), there is no detectable variation among the coding scores of the control and the Kozak-derived groups (Supplementary Figure S6), suggesting that the effect of this short signal is no longer detectable as the algorithm processes larger chunks of information.

RNAsamba can be used to identify mRNAs that encode micropeptides

In recent years, the advent of ribosome profiling (Ribo-Seq) has greatly contributed to our understanding of the translation dynamics in the cell. Unlike traditional RNA-Seq technologies, which generate sequencing data from RNA molecules irrespective of their coding potential, Ribo-Seq targets transcripts that are being actively translated (38). Therefore, this technology has made it possible to uncover micropeptides that are translated from transcripts that were previously thought to be non-coding (39). These peptides are much shorter than most known proteins, being translated from small ORFs (sORFs) containing 100 or less codons, but play important functional roles in multiple organisms (40,41).

Assuming a random codon distribution, the probability of a stop codon appearing within 100 codons of a start codon is approximately 99.2%, meaning that the genome contains a large quantity of short ORFs that arise by pure chance and carry no biological meaning. Because the ORF information of micropeptide transcripts is limited, the distinction between true sORFs and random short ORFs is challenging for computational methods. As RNAsamba uses whole-sequence information and is able to identify coding signatures outside of the ORF (as shown the the section above), we evaluated whether it can be used to identify coding transcripts with sORFs.

The assessment of the classification performance of RNAsamba and other classifiers in sORF data was conducted using test datasets containing both sORF mRNAs and lncRNAs with untranslated ORFs from five different species (human, *M. musculus*, *D. rerio*, *D. melanogaster* and *Saccharomyces cerevisiae*) (Supplementary Table S8). We observed that RNAsamba can reliably distinguish true sORFs from lncRNAs, outperforming other tools in most datasets (Figure 4 and Supplementary Table S9). In agreement with our previous results, RNAsamba's performance is slightly worse in the *D. rerio* and *D. melanogaster* datasets, which we suspect is partly due to protein-coding transcripts being misannotated as lncRNAs.

These results led us to believe that RNAsamba can be used to validate Ribo-Seq results or to identify sORFs in the absence of ribosome profiling data. This second application is especially useful for the annotation of novel genomes, where more specialized sequencing data is usually scarce.

RNAsamba is faster than neural network-based alternatives

Neural networks models are becoming increasingly popular due to their ability to learn non-intuitive patterns, which would otherwise be ignored by humans, from large quantities of data. This learning power is, however, accompanied by an enormous increase in the number of trainable parameters when compared to traditional machine learning techniques, greatly increasing training time (19). We felt that the available neural network-based coding-potential calculators impose a barrier for most users, as they do not possess GPU hardware to increase performance. By using modern libraries and IGLOO layers we sought to develop an algorithm that makes it feasible to train new models even with traditional CPUs. Using the FEELnc dataset, we compared RNAsamba to lncRNAnet and mRNN with respect to memory usage and wall time during inference and training.

Regarding peak memory usage, we found that during inference RNAsamba uses less resources than lncRNAnet and slightly more than mRNN. While training, due to its larger number of trainable parameters, RNAsamba uses twice as much memory as mRNN (Figure 5A and Supplementary Table S10). In both situations RNAsamba's peak memory usage did not exceed reasonable amounts and could be executed in regular notebooks.

LncRNAnet and mRNN employ traditional RNN variations—LSTM in lncRNAnet and GRU in mRNN—that were previously shown to be slower than IGLOO (21). Indeed, we found that RNAsamba's inference in a CPU is, on average, 32.0 and 11.1 times faster than lncRNAnet and mRNN, respectively. Regarding training, RNAsamba is 41.7 faster than mRNN in a CPU. RNAsamba's speed improvements are also significant when using a GPU (Figure 5B and Supplementary Table S10). Jointly, these results show that RNAsamba is much faster than other neural network-based alternatives, making it more accessible to most users.

We also compared RNAsamba's inference performance in the FEELnc dataset to that of tools that employ traditional machine learning algorithms and compute a reduced number of features. We observed that RNAsamba (31.02 s in CPU and 26.03 s in GPU) is slightly slower than both CPAT (19.90 s) and CPC2 (23.09 s) but is much faster than FEELnc (1664.05 s). However, it is important to note that FEELnc trains a new model before each inference, increasing the time it takes to classify a given set of sequences.

To evaluate whether RNAsamba is scalable to very large datasets, we tested its memory usage and speed in the GENCODE 32 human genome annotation (4), which contains 100 291 mRNAs and 48 351 lncRNAs. We found that, at this scale RNAsamba's inference is fast, showing that it can be used to promptly predict mRNAs and lncRNAs even in very large datasets, as long as the computer has enough memory to store the sequence features (Supplementary Table S11). As for training, even though it is feasible to use a whole transcriptome to train a new model in a powerful computer, we note that using a smaller subset of sequences is enough to train very accurate models and filtering sequences by length, for instance, can improve the model classification performance (30).

Ablation studies

RNAsamba's hyperparameters (number and size of hidden layers, number of IGLOO patches etc.) and features (*k*-mer and amino acid frequencies, ORF length etc.) were tuned through extensive manual and automated search. We investigated the effect of altering some of the properties of the RNAsamba algorithm to its overall performance.

Changing the maximum sequence length. As IGLOO layers require fixed-length inputs, we arbitrarily chose to truncate nucleotide and amino acid sequences at the positions 3000 and 1000, respectively. To check whether this choice negatively affected RNAsamba's classification performance by not providing it with important sequence information, we developed two alternative versions of the model that truncate nucleotide and amino acid sequences at 4500/1500 and 6000/2000. We verified that raising the input sequences maximum length increased both the train and test times, without improving the model's accuracy. Reducing the maximum lengths to 2400/800 resulted in a slight drop in classification performance (Table 1).

Removing the B1 branch. The removal of B1 reduces forces RNAsamba to rely only on information that is extracted from the identified ORF. We found that this ablation reduces RNAsamba's classification performance by a small amount in the FEELnc dataset, showing that B2 carries enough information to correctly classify full-length transcripts. However, we note that B1 is crucial for the classification of truncated transcripts, which may have partial or no ORF at all (Supplementary Figure S4).

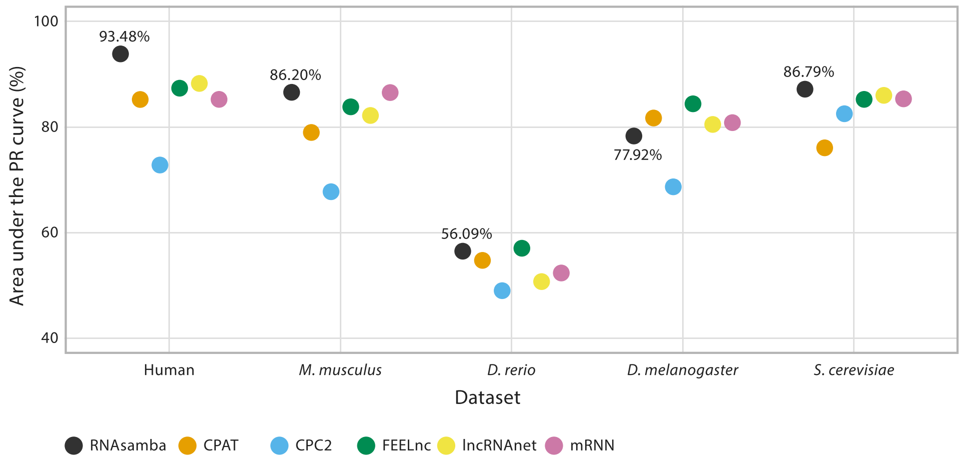

Figure 4. Classification benchmark of six different coding potential calculators in short ORF (sORF) datasets from five different species. Values correspond to the area under the precision-recall curve.

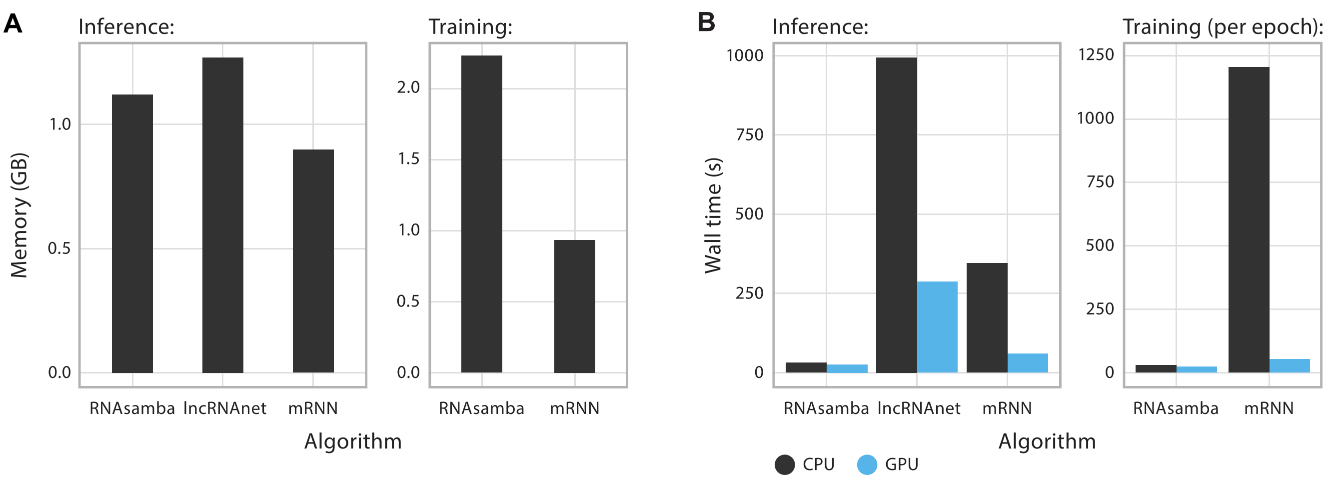

Figure 5. Computational performance of RNAsamba, lncRNAnet and mRNN in the FEELnc dataset. (**A**) Peak memory usage during inference and training. (**B**) Average inference and training wall time of five independent executions of each algorithm. LncRNAnet does not provide an interface to train new models, thus its training times were not measured. CPU computations were performed with two Intel® Xeon® E5-2420 v2 CPUs and GPU computations were performed with a NVIDIA® Tesla® K80. Inference execution time was measured with the hyperfine tool.

Table 1. Ablation studies of the RNAsamba model

Architecture	Maximum length (nt/aa)	Branches	Training (s)	Inference (s)	Accuracy
IGLOO	**3000/1000**	**B1, B2**	**394.04**	**30.11**	**0.9325**
IGLOO	2400/800	B1, B2	358.85	28.92	0.9300
IGLOO	4500/1500	B1, B2	493.41	37.20	0.9318
IGLOO	6000/2000	B1, B2	518.54	42.36	0.9321
IGLOO	3000/1000	B1	154.09	17.95	0.7901
IGLOO	3000/1000	B2	264.70	23.20	0.9237
IGLOO	3000/1000	B1, B2 (−P1)	251.44	24.39	0.9315
IGLOO	3000/1000	B1, B2 (−F1)	361.08	24.87	0.9210
IGLOO	3000/1000	B1, B2 (−A1)	390.67	28.78	0.9200
IGLOO	3000/1000	B1, B2 (−O1)	398.09	31.01	0.9322
GRU	3000/1000	B1, B2	4659.29	214.12	0.9054
LSTM	3000/1000	B1, B2	5197.71	479.05	0.9061

Default parameters are highlighted in bold. Reported train times, test times and accuracy values correspond to the average of five independent runs. Computations were performed with two Intel® Xeon® E5-2420 v2 CPUs. Execution time was measured with the hyperfine tool.

Removing the B2 branch. By removing the B2 branch, we deprived RNAsamba's algorithm of ORF-derived features, forcing it to leverage whole-sequence information to distinguish between mRNAs and lncRNAs. We observed that this ablation reduced the accuracy of the network by 15.79% (Table 1), leading us to the conclusion that features the algorithm derives from the ORF contain key information that is not extracted from the nucleotide sequence by the IGLOO layer alone. We also observed that removing each of the layers of B2 (P1, F1, A1 and O1) individually led to small drops in accuracy, evidencing that there is partly redundant information among them.

Replacing IGLOO with GRU and LSTM. The Gated Recurrent Unit (GRU) (42) and the Long Short-Term Memory (LSTM) (43) are established RNN architectures, commonly used in deep-learning tasks that deal with sequences. Recently, IGLOO has been shown to outperform both GRU and LSTM in terms of run time and accuracy on some standard benchmark problems such as the copy-memory and the addition tasks (21). To evaluate whether this holds true in the mRNA/lncRNA classification paradigm, we developed alternative versions of our algorithm in which IGLOO was substituted by GRU or LSTM layers with 256 units. We found that the model using IGLOO is more accurate and significantly faster, for both training and classification, than the GRU and LSTM variants (Table 1). We note, however, that as IGLOO creates sequences representation from slices taken from random locations of the transcript, sequence fragments cannot be mapped to specific network weights. Therefore, RNAsamba cannot be used to evaluate the contribution of individual regions to the overall coding score in an unsupervised manner.

CONCLUSION

In this study, we presented RNAsamba, a new deep learning-based tool to predict the coding potential of RNA transcripts relying solely in sequence information. Compared to other algorithms, RNAsamba exhibits better classification performance in multiple human datasets and generalizes very well to other species, without relying on computationally expensive data augmentation.

We believe that RNAsamba's algorithm introduces two major contributions: (1) the usage of the IGLOO architecture to learn from sequence data and (2) the integration of whole transcript and ORF-derived information into a single coding score. By using IGLOO layers, RNAsamba can learn non-intuitive coding patterns, as we demonstrated with the Kozak consensus, without relying on biased human-designed features. This architecture also makes RNAsamba significantly faster than RNN-based algorithms, making it more appealing to most users. Through the usage of its two branches, RNAsamba can identify mRNAs with short or incomplete ORFs, which usually are misclassified by most algorithms. Based on this, we believe that RNAsamba is a useful tool for the annotation of novel genomes and transcriptomes, improving the quality of coding and non-coding gene prediction.

With RNAsamba, we sought to offer a fast and easy-to-use tool to most researchers, developed using modern and well documented libraries. Also, we provide a Docker image, a convenient web server (https://rnasamba.lge.ibi.unicamp.br/) and an intuitive command-line interface to promptly execute training and inference tasks. By doing so, we believe that RNAsamba provides researchers with a state-of-the-art coding potential calculator that allows fast and accurate predictions of mRNAs and lncRNAs, enabling more precise biological insights from the genomes of newly sequenced species.

MATERIALS AND METHODS

Full-length transcripts datasets

To keep the comparisons unbiased, the train and test datasets used in the benchmarks were all obtained from previous publications (27,28,30). Links for download of the datasets used in these benchmarks can be found in the Supplementary Data.

CPC2's train set consists of a set of mRNAs with high-quality coding sequences annotated by the CCDS project (44) selected from the RefSeq database (45) and lncRNAs randomly selected from GENCODE. CPC2 includes test sets for several species (human, *Mus musculus*, *Danio rerio*, *Drosophila melanogaster*, *Caenorhabditis elegans* and *Arabidopsis thaliana*) and they were built using non-redundant mRNA sequences obtained from RefSeq and lncRNAs retrieved from Ensembl and Ensembl Plants. Sequences that were present in the training set were excluded from the test sets.

For the FEELnc dataset, human transcripts were obtained from GENCODE 24 and the gene biotypes were used to select mRNAs ('protein_coding') and lncRNAs ('lincRNA' and 'antisense'). To avoid biases, a single transcript was selected per locus.

mRNN's dataset is comprised of a subset of human transcripts obtained from GENCODE 25. In addition to the regular test set, mRNN dataset includes a challenge test (mRNN-Challenge) set that contains mRNAs with short ORFs ($\leq$50 codons in the in GENCODE annotation) and lncRNAs with long untranslated ORFs ($\geq$50 codons). Transcripts associated with loci present in any of the test sets were excluded from the training data.

Finally, sORF datasets were built from mRNAs and lncRNAs from five different species (human, *Mus musculus*, *Danio rerio*, *Drosophila melanogaster*, *Caenorhabditis elegans* and *Arabidopsis thaliana*). Sequences were retrieved from RefSeq and Ensembl and mRNAs and lncRNAs that were filtered to retain only the ORF fragments that were $<$303 nucleotides.

Classification performance evaluation

We assessed the performance of RNAsamba and five other sequence-dependent classification software: CPAT (1.2.4), CPC2, FEELnc (version 0.1.1), lncRNAnet and mRNN. We calculated the performance metrics with the scikit-learn Python package (46). For these computations, mRNAs were considered the positive class and ncRNAs the negative class.

For the performance evaluation in the human, *M. musculus*, *D. rerio*, *D. melanogaster*, *C. elegans* and *A. thaliana* datasets we took two approaches: (i) classification with pre-trained models and (ii) classification with models trained with the train dataset that corresponds with the test data. For the first approach, we used the models that are built-in with each tool. RNAsamba was trained with a non-redundant set containing the sequences of all four human train datasets; mRNN was loaded with weights provided in the w14u3.pkl file. For the second approach, new models were trained for each tool using default parameters. CPAT's hexamer frequency tables were generated from each train set using the companion make_hexamer_tab.py script. LncRNAscore was excluded from benchmarks that used newly trained models because it does not provide an interface for training. The classification benchmark in sORF datasets from human, *M. musculus*, *D. rerio*, *D. melanogaster* and S. *cerevisiae* was performed using built-in models.

Annotation of new putative protein-coding transcripts

To evaluate whether *D. rerio* and *D. melanogaster* lncRNAs classified as protein-coding by RNAsamba can be putatively assigned to protein families, we compared these sequences to a series of protein-related databases. Transcript nucleotide sequences were matched to proteins in the UniRef90 database (release 2019_10) (47) using the easy-search command from MMseqs2 (version 10.6d92c) (48). Nucleotide sequences were also used as queries to search for conserved protein domains in the CDD database (version 3.17) (49) using RPS-BLAST (version 2.9.0) (50). Finally, translated ORFs were compared to the Pfam protein family database (version 32.0) (51) using the hmmscan command from the HMMER suite (version 3.2.1) (52). For all searches, only hits with E-value ≤ 0.001 were considered.

Truncated ORFs dataset

To generate the test for the analysis of truncated transcripts, mouse and *A. thaliana* ORF sequences were retrieved from Ensembl (release 94) and Ensembl Plants (release 43) (53), respectively, and sequences <300 nucleotides were discarded. Next, ORFs that exhibited an in-frame start codon and the ones that didn't were separated into different sets. The start and stop codons were removed from the sequences of both sets, guaranteeing that the true beginning and end of the ORFs would not be detected by the classifiers. Subsequently, each set was used to generate five subsets consisting of 1000 randomly sampled sequences. Finally, the sequences from each dataset were sliced at random positions to generate sets of fragmented ORFs with fixed relative lengths (20%, 30%, 50%, 70% and 90% of the total ORF length).

For the performance evaluation, we used a RNAsamba model trained with a set containing the CPC2, FEELnc and mRNN human train and test sets as well as fragmented ORFs extracted from 50 000 of those sequences. We used the easy-search command from MMseqs2 to identify and remove sequences from the train set that displayed >90% identity and covered >90% of the length of any of the test sequences. CPAT, CPC2 FEELnc, lncRNAnet and mRNN were executed using pre-trained models. mRNN was loaded with weights provided in the w14u3.pkl file.

Kozak sequence analysis

The 100 different 10 bp fragments in the Kozak sequence set and the control set were generated, respectively, from the Kozak sequence probability distribution (Supplementary Figure S5A) and a uniform distribution, in which all four nucleotides are equally probable to be drawn in each position (except for the start codon). The distance between the generated fragments and the Kozak sequence was obtained by computing their Hamming distances to two sequences derived from the Kozak consensus (GCC[AG]CCATGG) and choosing the lowest value.

We randomly selected 1000 sequences among mouse mRNAs, retrieved from Ensembl (release 94), whose 5′ UTR contained at least 6 nucleotides. Then, the region spawning the positions −6 to +1 of each mRNA was replaced by the 10 bp fragments of the Kozak sequence set and control set, producing two sets of hybrid transcripts containing both biological and computer-generated sequences (Supplementary Figure S5B).

For each mRNA, we used one-tailed Mann–Whitney U tests to evaluate differences between the coding scores of sequences in the two sets. We used the Benjamini–Hochberg procedure to compute the false discovery rates (FDR). Kendall's tau coefficient was used to measure the degree of association between coding scores and Hamming distance to the Kozak sequence consensus.

Ablation studies

The models generated in the ablation studies were trained for 10 epochs using the FEELnc human train set and all the benchmarks were performed on the FEELnc human test set.

DATA AVAILABILITY

The source code for RNAsamba is available in an online repository (https://github.com/apcamargo/RNAsamba). Train and test sequences generated for the truncated ORF analysis, as well as computer-generated Kozak fragments, were uploaded to Open Science Framework (https://doi.org/10.17605/OSF.IO/MD56Y).

SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

ACKNOWLEDGMENTS

We thank Adrielle A. Vasconcelos for helpful advices to the writing of the manuscript and the anonymous reviewers for helpful suggestions and insights.

FUNDING

São Paulo Research Foundation (FAPESP) [FAPESP/CEP ID #2013/08293-7, #2018/04240-0 (to A.P.C.)].
Conflict of interest statement. None declared.

REFERENCES

1. Wang,Z., Gerstein,M. and Snyder,M. (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, **10**, 57–63.
2. Wang,K.C. and Chang,H.Y. (2011) Molecular mechanisms of long noncoding RNAs. *Mol. Cell*, **43**, 904–914.
3. Consortium,E.P., Dunham,I., Kundaje,A., Aldred,S.F., Collins,P.J., Davis,C.a., Doyle,F., Epstein,C.B., Frietze,S., Harrow,J. *et al.* (2013) An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**, 57–74.
4. Frankish,A., Diekhans,M., Ferreira,A.M., Johnson,R., Jungreis,I., Loveland,J., Mudge,J.M., Sisu,C., Wright,J., Armstrong,J. *et al.* (2019) GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res.*, **47**, D766–D773.
5. Iwakiri,J., Hamada,M. and Asai,K. (2016) Bioinformatics tools for lncRNA research. *Biochim. Biophys. Acta - Gene Regul. Mech.*, **1859**, 23–30.
6. Gollery,M., Harper,J., Cushman,J., Mittler,T., Girke,T., Zhu,J.K., Bailey-Serres,J. and Mittler,R. (2006) What makes species unique? The contribution of proteins with obscure features. *Genome Biol.*, **7**, R57.
7. Guttman,M., Amit,I., Garber,M., French,C., Lin,M.F., Feldser,D., Huarte,M., Zuk,O., Carey,B.W., Cassady,J.P. *et al.* (2009) Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature*, **458**, 223–227.
8. Ulitsky,I. (2016) Evolution to the rescue: Using comparative genomics to understand long non-coding RNAs. *Nat. Rev. Genet.*, **17**, 601–614.
9. Zhao,J., Song,X. and Wang,K. (2016) LncScore: Alignment-free identification of long noncoding RNA from assembled novel transcripts. *Sci. Rep.*, **6**, 34838.
10. Noviello,T.M.R., Di Liddo,A., Ventola,G.M., Spagnuolo,A., D'Aniello,S., Ceccarelli,M. and Cerulo,L. (2018) Detection of long non-coding RNA homology, a comparative study on alignment and alignment-free metrics. *BMC Bioinformatics*, **19**, 407.
11. Haerty,W. and Ponting,C.P. (2015) Unexpected selection to retain high GC content and splicing enhancers within exons of multiexonic IncRNA loci. *RNA*, **21**, 320–332.
12. Quinn,J.J. and Chang,H.Y. (2016) Unique features of long non-coding RNA biogenesis and function. *Nat. Rev. Genet.*, **17**, 47–62.
13. Dinger,M.E., Pang,K.C., Mercer,T.R., Mattick,J.S., Frith,M.C., Bailey,T.L., Kasukawa,T., Mignone,F., Kummerfeld,S.K., Madera,M. *et al.* (2006) Discrimination of non-protein-coding transcripts from protein-coding mRNA. *RNA Biol.*, **4**, 40–48.
14. Dinger,M.E., Pang,K.C., Mercer,T.R. and Mattick,J.S. (2008) Differentiating protein-coding and noncoding RNA: Challenges and ambiguities. *PLoS Comput. Biol.*, **4**, e1000176.
15. Yin,C. and Yau,S.S.T. (2007) Prediction of protein coding regions by the 3-base periodicity analysis of a DNA sequence. *J. Theor. Biol.*, **247**, 687–694.
16. Li,A., Zhang,J. and Zhou,Z. (2014) PLEK: A tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme. *BMC Bioinformatics*, **15**, 311.
17. Pian,C., Zhang,G., Chen,Z., Chen,Y., Zhang,J., Yang,T. and Zhang,L. (2016) LncRNApred: Classification of long non-coding RNAs and protein-coding transcripts by the ensemble algorithm with a new hybrid feature. *PLoS One*, **11**, e0154567.
18. Fickett,J.W. and Tung,C.-S. (1992) Assessment of protein coding measures. *Nucleic Acids Res.*, **20**, 6441–6450.
19. Min,S., Lee,B. and Yoon,S. (2017) Deep learning in bioinformatics. *Brief. Bioinform.*, **18**, 851–869.
20. Lipton,Z.C., Berkowitz,J. and Elkan,C. (2015) A critical review of recurrent neural networks for sequence learning. arXiv doi: https://arxiv.org/abs/1506.00019, 29 May 2015, preprint: not peer reviewed.
21. Sourkov,V. (2018) IGLOO: Slicing the features space to represent long sequences. arXiv doi: https://arxiv.org/abs/1807.03402, 09 July 2018, preprint: not peer reviewed.
22. Zhou,M., Xie,H., Wang,M.D., Yang,L., Zhu,H., Yang,C. and Zhang,C. (2018) LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning. *Bioinformatics*, **34**, 3825–3834.
23. Hu,L., Xu,Z., Hu,B. and Lu,Z.J. (2017) COME: A robust coding potential calculation tool for lncRNA identification and characterization based on multiple features. *Nucleic Acids Res.*, **45**, e2.
24. Sun,L., Liu,H., Zhang,L. and Meng,J. (2015) lncRScan-SVM: A tool for predicting long non-coding RNAs using support vector machine. *PLoS One*, **10**, e0139654.
25. Gao,G., Kong,L., Wei,L., Zhao,S.-Q., Ye,Z.-Q., Liu,X.-Q. and Zhang,Y. (2007) CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Res.*, **35**, W345–W349.
26. Wang,L., Park,H.J., Dasari,S., Wang,S., Kocher,J.P. and Li,W. (2013) CPAT: Coding-potential assessment tool using an alignment-free logistic regression model. *Nucleic Acids Res.*, **41**, e74.
27. Kang,Y.J., Yang,D.C., Kong,L., Hou,M., Meng,Y.Q., Wei,L. and Gao,G. (2017) CPC2: A fast and accurate coding potential calculator based on sequence intrinsic features. *Nucleic Acids Res.*, **45**, W12–W16.
28. Lohi,H., Le Béguec,C., Lagoutte,L., Leeb,T., Fredholm,M., Derrien,T., André,C., Lindblad-Toh,K., Legeai,F., Rizk,G. *et al.* (2017) FEELnc: a tool for long non-coding RNA annotation and its application to the dog transcriptome. *Nucleic Acids Res.*, **45**, e57.
29. Baek,J., Lee,B., Kwon,S. and Yoon,S. (2018) LncRNAnet: Long non-coding RNA identification using deep learning. *Bioinformatics*, **34**, 3889–3897.
30. Kuintzle,R., Hendrix,D.A., Danaee,P., Teegarden,A., Hill,S.T. and Merrill,E. (2018) A deep recurrent neural network discovers complex biological rules to decipher RNA protein-coding potential. *Nucleic Acids Res.*, **46**, 8105–8113.
31. Bishop,C.M. (2006) *Pattern Recognition and Machine Learning*, Springer, NY.
32. Kingma,D.P. and Ba,J. (2014) Adam: A Method for Stochastic Optimization. arXiv doi: https://arxiv.org/abs/1412.6980, 22 December 2014, preprint: not peer reviewed.
33. Abadi,M., Barham,P., Chen,J., Chen,Z., Davis,A., Dean,J., Devin,M., Ghemawat,S., Irving,G., Isard,M. *et al.* (2016) TensorFlow: A system for large-scale machine learning. arXiv doi: https://arxiv.org/abs/1603.04467, 14 March 2016, preprint: not peer reviewed.
34. Steijger,T., Abril,J.F., Engström,P.G., Kokocinski,F., Akerman,M., Alioto,T., Ambrosini,G., Antonarakis,S.E., Behr,J., Bertone,P. *et al.* (2013) Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods*, **10**, 1177–1184.
35. Iyer,M.K., Niknafs,Y.S., Malik,R., Singhal,U., Sahu,A., Hosono,Y., Barrette,T.R., Prensner,J.R., Evans,J.R., Zhao,S. *et al.* (2015) The landscape of long noncoding RNAs in the human transcriptome. *Nat. Genet.*, **47**, 199–208.
36. Kozak,M. (1987) An analysis of 5′-noncoding sequences from 699 vertebrate messenger rNAS. *Nucleic Acids Res.*, **15**, 8125–8148.
37. De Angioletti,M., Lacerra,G., Sabato,V. and Carestia,C. (2004) β+45 G → C: A novel silent β-thalassaemia mutation, the first in the Kozak sequence. *Br. J. Haematol.*, **124**, 224–231.
38. Ingolia,N.T. (2014) Ribosome profiling: New views of translation, from single codons to genome scale. *Nat. Rev. Genet.*, **15**, 205–213.
39. Ruiz-Orera,J. and Albà,M.M. (2019) Translation of small open reading frames: roles in regulation and evolutionary innovation. *Trends Genet.*, **35**, 186–198.
40. Pauli,A., Norris,M.L., Valen,E., Chew,G.L., Gagnon,J.A., Zimmerman,S., Mitchell,A., Ma,J., Dubrulle,J., Reyon,D. *et al.* (2014) Toddler: An embryonic signal that promotes cell movement via apelin receptors. *Science*, **343**, 1248636.
41. Herberg,S., Gert,K.R., Schleiffer,A. and Pauli,A. (2018) The Ly6/uPAR protein Bouncer is necessary and sufficient for species-specific fertilization. *Science*, **361**, 1029–1033.
42. Cho,K., van Merrienboer,B., Gulcehre,C., Bahdanau,D., Bougares,F., Schwenk,H. and Bengio,Y. (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv doi: https://arxiv.org/abs/1406.1078, 03 June 2014, preprint: not peer reviewed.
43. Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.
44. Pruitt,K.D., Harrow,J., Harte,R.A., Wallin,C., Diekhans,M., Maglott,D.R., Searle,S., Farrell,C.M., Loveland,J.E., Ruef,B.J. *et al.* (2009) The consensus coding sequence (CCDS) project: Identifying a

common protein-coding gene set for the human and mouse genomes. *Genome Res.*, **19**, 1316–1323.
45. O'Leary,N.A., Wright,M.W., Brister,J.R., Ciufo,S., Haddad,D., McVeigh,R., Rajput,B., Robbertse,B., Smith-White,B., Ako-Adjei,D. *et al.* (2016) Reference sequence (RefSeq) database at NCBI: Current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.*, **44**, D733–D745.
46. Pedregosa,F., Varoquaux,G., Gramfort,A., Michel,V., Thirion,B., Grisel,O., Blondel,M., Prettenhofer,P., Weiss,R., Dubourg,V. *et al.* (2011) Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.
47. Suzek,B.E., Wang,Y., Huang,H., McGarvey,P.B. and Wu,C.H. (2015) UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, **31**, 926–932.
48. Steinegger,M. and Söding,J. (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, **35**, 1026–1028.
49. Marchler-Bauer,A., Bo,Y., Han,L., He,J., Lanczycki,C.J., Lu,S., Chitsaz,F., Derbyshire,M.K., Geer,R.C., Gonzales,N.R. *et al.* (2017) CDD/SPARCLE: Functional classification of proteins via subfamily domain architectures. *Nucleic Acids Res.*, **45**, D200–D203.
50. Boratyn,G.M., Schäffer,A.A., Agarwala,R., Altschul,S.F., Lipman,D.J. and Madden,T.L. (2012) Domain enhanced lookup time accelerated BLAST. *Biol. Direct*, **7**, 12.
51. El-Gebali,S., Mistry,J., Bateman,A., Eddy,S.R., Luciani,A., Potter,S.C., Qureshi,M., Richardson,L.J., Salazar,G.A., Smart,A. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.
52. S R Eddy (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
53. Yates,A.D., Achuthan,P., Akanni,W., Allen,J., Allen,J., Alvarez-Jarreta,J., Amode,M.R., Armean,I.M., Azov,A.G., Bennett,R. *et al.* (2020) Ensembl 2020. *Nucleic Acids Res.*, **48**, D682–D688.