SOFTWARE TOOL ARTICLE

REVISED RNAmining: A machine learning stand-alone and web server tool for RNA coding potential prediction [version 2; peer review: 2 approved]

Thaís A.R. Ramos[1-3], Nilbson R.O. Galindo[2], Raúl Arias-Carrasco[3], Cecília F. da Silva[2], Vinicius Maracaja-Coutinho[1,3,4], Thaís G. do Rêgo[1,2]

[1]Programa de Pós-Graduação em Bioinformática, Bioinformatics Multidisciplinary Environment (BioME), Instituto Metrópole Digital, Universidade Federal do Rio Grande do Norte, Natal, Brazil
[2]Departamento de Informática, Centro de Informática, Universidade Federal da Paraíba, João Pessoa, Brazil
[3]Advanced Center for Chronic Diseases (ACCDiS), Facultad de Ciencias Químicas y Farmacéuticas, Universidad de Chile, Santiago, Chile
[4]Instituto Vandique, João Pessoa, Brazil

v2 **First published:** 26 Apr 2021, **10**:323
https://doi.org/10.12688/f1000research.52350.1
Latest published: 08 Jun 2021, **10**:323
https://doi.org/10.12688/f1000research.52350.2

Abstract

Non-coding RNAs (ncRNAs) are important players in the cellular regulation of organisms from different kingdoms. One of the key steps in ncRNAs research is the ability to distinguish coding/non-coding sequences. We applied seven machine learning algorithms (Naive Bayes, Support Vector Machine, K-Nearest Neighbors, Random Forest, Extreme Gradient Boosting, Neural Networks and Deep Learning) through model organisms from different evolutionary branches to create a stand-alone and web server tool (RNAmining) to distinguish coding and non-coding sequences. Firstly, we used coding/non-coding sequences downloaded from Ensembl (April 14th, 2020). Then, coding/non-coding sequences were balanced, had their trinucleotides count analysed (64 features) and we performed a normalization by the sequence length, resulting in total of 180 models. The machine learning algorithms validations were performed using 10-fold cross-validation and we selected the algorithm with the best results (eXtreme Gradient Boosting) to implement at RNAmining. Best F1-scores ranged from 97.56% to 99.57% depending on the organism. Moreover, we produced a benchmarking with other tools already in literature (CPAT, CPC2, RNAcon and TransDecoder) and our results outperformed them. Both stand-alone and web server versions of RNAmining are freely available at https://rnamining.integrativebioinformatics.me/.

Keywords

Machine Learning, non-coding RNA, benchmarking, coding potential prediction

Open Peer Review

Reviewer Status ✓ ✓

	Invited Reviewers	
	1	**2**
version 2 (revision) 08 Jun 2021	✓ report	✓ report
version 1 26 Apr 2021	? report	? report

1. **Gilderlanio Araújo**, Universidade Federal do Pará, Belém, Brazil

2. **Andre Yoshiaki Kashiwabara**, Federal University of Technology - Parana, Curitiba, Brazil

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding authors: Vinicius Maracaja-Coutinho (vinicius.maracaja@uchile.cl), Thaís G. do Rêgo (gaudenciothais@gmail.com)

Author roles: Ramos TAR: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Galindo NRO**: Data Curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Arias-Carrasco R**: Formal Analysis, Methodology, Software, Validation; **da Silva CF**: Formal Analysis, Methodology, Software, Validation; **Maracaja-Coutinho V**: Conceptualization, Data Curation, Investigation, Methodology, Project Administration, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **do Rêgo TG**: Conceptualization, Data Curation, Investigation, Methodology, Project Administration, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was funded in part by grants from ANID-FONDECYT (11161020 and 1211731), ANID-PAI (PAI79170021) and ANID-FONDAP (15130011) to VMC. TARR received a Master and a PhD fellowship from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil. RAC received a post-doctoral fellowship from ACCDiS.

How to cite this article: Ramos TAR, Galindo NRO, Arias-Carrasco R *et al.* **RNAmining: A machine learning stand-alone and web server tool for RNA coding potential prediction [version 2; peer review: 2 approved]** F1000Research 2021, **10**:323 https://doi.org/10.12688/f1000research.52350.2

First published: 26 Apr 2021, **10**:323 https://doi.org/10.12688/f1000research.52350.1

REVISED **Amendments from Version 1**

Here, we present the revised update manuscript. In brief, the minor changes as below;

We updated the abstract

We updated the Introduction section with reviewer's suggestion: 1- We included the citations for BASiNET and CoDaN; 2- We added the sentence "Next, RNAmining was evaluated in another 9 phylogenetically related and unrelated organisms that were not used in our training, demonstrating the efficiency of the tool even when applied in species phylogenetically distant from those used in training."

We restructured the second paragraph of "Machine learning classifier algorithms selection" section and the first paragraph of "Training and testing datasets, model building and quality measuring for coding potential evaluation" section.

We added a new key point in conclusion "RNAmining was evaluated using other phylogenetically related and unrelated organisms that were not used in our training, demonstrating the efficiency of the tool even when applied in species phylogenetically distant from those used in training."

We updated Figure 2 and the source code of RNAmining (including the classification probabilities in the output) as suggested by the reviewers.

Any further responses from the reviewers can be found at the end of the article

Introduction

Non-coding RNAs (ncRNAs) are key functional players on different biological processes in organisms from all domains of life[1,2]. Its investigation is already routine in almost every transcriptome or genome project. Dysregulations in these molecules may lead to different types of human disease, including cancers[3], neurological disorders[4] and cardiovascular infirmities[5].

The genome of eukaryotic[6] organisms is, in general, majority composed of non-coding transcripts, with complex organisms estimated to transcribe more than 75% of their genomes[7]. Besides strong evidence associating these ncRNAs to key functions in the cell, most of them are not yet associated with a functional mechanism. In a transcriptome project there exists an important step in the computational identification of ncRNAs, which is the evaluation of their potential to be translated into proteins using different bioinformatics approaches[8,9]. To computationally evaluate the coding potential of a set of transcripts, available tools or algorithms normally analyse specific characteristics available in primary sequences (*e.g.* nucleotides counts, the existence of a trustful open reading frame).

For instance, RNAcon implements a Support Vector Machine (SVM)-based model for the discrimination between coding and non-coding sequences[10]. Coding Potential Assessment Tool (CPAT)[11] assesses the coding potential through an alignment-free method, which uses a logistic regression model built based on different characteristics of the sequence open reading frame (ORF), which includes length, coverage and nucleotides compositional bias. TransDecoder identifies candidate coding transcripts based on other distinctive features from predicted ORFs (*e.g.* a minimum length ORF, a log-likelihood score, encapsulated ORF)[12]. CPC2[13] trained a SVM model using Fickett TESTCODE score, ORF length, ORF integrity and isoelectric point as features. The LIBSVM[14] package was employed by training a SVM model using the standard radial basis function kernel (RBF kernel) with the training dataset containing 17,984 high-confident human protein-coding transcripts and 10,452 non-coding transcripts[11]. CoDaN uses Generalized Hidden Marvov to generate probabilistic models based on the GC content of nucleotide sequences in order to estimate the coding regions and both 5' and 3' untranslated regions of transcripts[15]. BASiNET performs feature selection to transform nucleotide sequences as complex networks, then it generates topological measures to build a feature vector used to classify the sequences[16].

Here, we applied and benchmarked seven different machine learning algorithms (Random Forest, eXtreme Gradient Boosting (XGBoost), Naive Bayes, K-Nearest Neighbors (K-NN), SVM, Artificial Neural Network (ANN) and Deep Learning (DL)) through 15 organisms from different evolutionary branches, in order to evaluate their performance in distinguishing coding and non-coding RNA sequences. Next, we developed a stand-alone and web server tool, called RNAmining (http://rnamining.integrativebioinformatics.me/), by selecting and implementing the algorithm with the best performance in all organisms (XGBoost). Next, RNAmining was evaluated in another 9 phylogenetically related and unrelated organisms that were not used in our training, demonstrating the efficiency of the tool even when applied in species phylogenetically distant from those used in training. In total, it was evaluated through 24 organisms from the eukaryotic tree of life and its results outperformed publicly available tools commonly used for that purpose.

Methods

Machine learning classifier algorithms selection

In the classification process there is a division related to the learning paradigm, with classification algorithms divided into: (i) *Symbolic*, which seeks to learn by constructing symbolic representations of a concept through the analysis of examples and counterexamples (e.g. Decision Trees and Rule-based System); (ii) *Statistical*, which looks for statistical methods and use models to find a good approximation of the induced concept (*e.g.* Bayesian learning); (iii) *Based on Examples* (lazy systems), which aims to classify examples never seen using similar known examples, assuming that the new example will belong to the same class as the similar example (*e.g.* K-Nearest Neighbor); (iv) *Based on Optimization*, which consists of maximizing (or minimizing) an objective function or finding an optimal hyperplane that best divides two classes (*e.g.* SVM and Neural Networks); (v) *Connectionist Representation*, which represents simplified mathematical constructions inspired by the biological model of the nervous system (e.g. Neural Networks). In this benchmarking, we decided to evaluate the performance of selected algorithms from each paradigm

type in the coding potential prediction of RNA sequences: Random Forest, XGBoost, Naive Bayes, K-NN, SVM and Neural Networks (ANN and Convolutional Neural Networks (CNN)).

All the machine learning methods were executed using scikit-learn (Version 0.21.3)[17], except for Neural Network and DL models which were implemented using Keras API with Tensorflow as backend (Version 2.3.0) and XGBoost algorithm which was executed using XGBoost Library (version 1.2.0)[18] in Python Language (Version 3.8). XGBoost, K-NN and Naive Bayes models were trained with the default values. The Random Forest and SVM parameters were obtained through grid search method. The Random Forest and SVM parameters were obtained through grid search method, the best results using Random Forest resulted in a model generated with the default parameters, with the exception of the number of trees used (150 estimators) and the criterion parameter setted to 'entropy' for information gain. For SVM, the resulting model was trained with the Radial Basis Function (RBF) kernel, with the Regularization parameter (C) and Kernel coefficient (Gamma) defined in 1000 and 0.8, respectively. ANN and DL were performed with different architectures according to grid search and empirical tests. The first ANN experiment was composed of three hidden layers consisting of 32-16-8 neurons, respectively; the second ANN experiment was performed with 64-32-16-8 neurons; and the third experiment was executed with 32-32-16-8 neurons. Next, we produced four experiments with DL using 2 CNN layers, followed by 2 fully connected (dense) layers: the first experiment had 512(CNN)-512(CNN) filters and 28(Dense)-1(Dense) neurons; the second was created with 64(CNN)-64(CNN) filters and 128(Dense)-1(Dense) neurons; the third was performed with 32(CNN)-32(CNN)-128(Dense)-1(Dense) neurons; and the last was built with 128(CNN)-128(CNN)-128(Dense)-1(Dense) neurons. These layers received as input the total number of attributes (*i.e.* combination of trinucleotides counts, described in the next topics). The hyperparameters used to execute the DL and ANN approaches are made available in *Extended data:* Supplementary File S1[19].

Datasets selection and filtering criteria

We compared the algorithms performances using different sets of coding and non-coding RNA sequences from Ensembl (April 14th 2020)[20] database, covering 15 organisms of distinct representative Chordata clades (Figure 1A): *Anolis carolinensis* (Sauria, Squamata), *Chrysemys picta bellii* (Sauria, Testudines), *Crocodylus porosus* (Archosauria, Pseudosuchia), *Danio rerio* (Actinopterygii, Teleostei), *Eptatretus burgeri* (Agnatha, Myxinidae), *Gallus gallus* (Archosauria, Theropoda), *Homo sapiens* (Placentalia), *Latimeria chalumnae* (Sarcopterygii, Coelacanth), *Monodelphis domestica* (Marsupialia), *Mus musculus* (Placentalia), *Notechis scutatus* (Sauria, Squamata), *Ornithorhynchus anatinus* (Monotremata), *Petromyzon marinus* (Agnatha, Petromyzontiformes), *Sphenodon punctatus* (Sauria, Rhynchocephalia), *Xenopus tropicalis* (Amphibia). All non-coding RNA sequences for each organism were downloaded from Ensembl transcripts. In order to obtain a balanced set of sequences (*i.e.* equal number of coding and non-coding), the group of coding RNAs were randomly selected in order to obtain the same number of ncRNAs for each species. Moreover, before generating the models, the sequences were normalized through their length (*i.e.* each trinucleotide count was divided by the total size of the given sequence). All sequences in FASTA format with their respective Ensembl identifiers can be retrieved at RNAmining website (https://rnamining.integrativebioinformatics.me/download).

Training and testing datasets, model building and quality measuring for coding potential evaluation

The cross-validation approach was applied in the grid search method, using the training dataset to validate the hyperparameters and obtain the best set of parameters to be used. In addition, this partition method validates the hyperparameter's results through different validation sets. Therefore, it proves that our model is working and generalizing the problem. Thus, sequences were randomly divided into training and testing datasets, using 80% of the data for training and 20% for testing. The connectionist methods (e.g. Artificial Neural Networks and Convolutional Neural Networks) demand a validation dataset to adjust the model, because of the weights optimization stage and its hyperparameters. Thus, for experiments with ANN and CNN, 20% were used for validation, 60% for training (defined as 80% for the other algorithms) and 20% for testing. The testing dataset was the same used in all machine learning algorithms. The number of sequences used for each organism for the training and test sets can be observed in Table 1. Next, we generated 180 models (*i.e.* one per algorithm for each organism, whereas three experiments for ANN models and four experiments for CNN models), which were further evaluated in this work.

After selection of the best model, it was applied and evaluated in other nine organisms (Figure 1A), different from the one used in the training process, including five related Chordata and other four phylogenetically distant species. Among the chordates, the models were tested in *Carassius auratus* (Actinopterygii, Teleostei), *Gorilla gorilla gorilla* (Placentalia), *Pseudonaja textilis* (Sauria, Squamata), *Rattus norvegicus (*Placentalia*)* and *Terrapene carolina triunguis* (Sauria, Testudines). Within non-chordates species, we evaluated the model in *Arabidopsis thaliana* (Plantae, Eudicots), *Caenorhabditis elegans* (Nematoda), *Drosophila melanogaster* (Insecta, Diptera) and *Saccharomyces cerevisiae* (Fungi, Ascomycota). Finally, it was evaluated using artificial sequences containing the same nucleotides composition of the ncRNAs for each species of the testing dataset (Table 1). Ten sets of random sequences containing the same number of ncRNAs per species were generated using MEME suite Version 5.1.1 with default parameters[21]. All sequences in FASTA format with their respective Ensemble identifiers can be retrieved at RNAmining website (https://rnamining.integrativebioinformatics.me/download).

Comparisons with publicly available tools

The performance of all algorithms in the coding potential evaluation was compared with publicly available tools commonly employed for this purpose (RNAcon[10], CPAT[11], TransDecoder[12] and CPC2[13]), using default parameters. It is worth noting that CPAT only made available models for *H. sapiens* with a coding

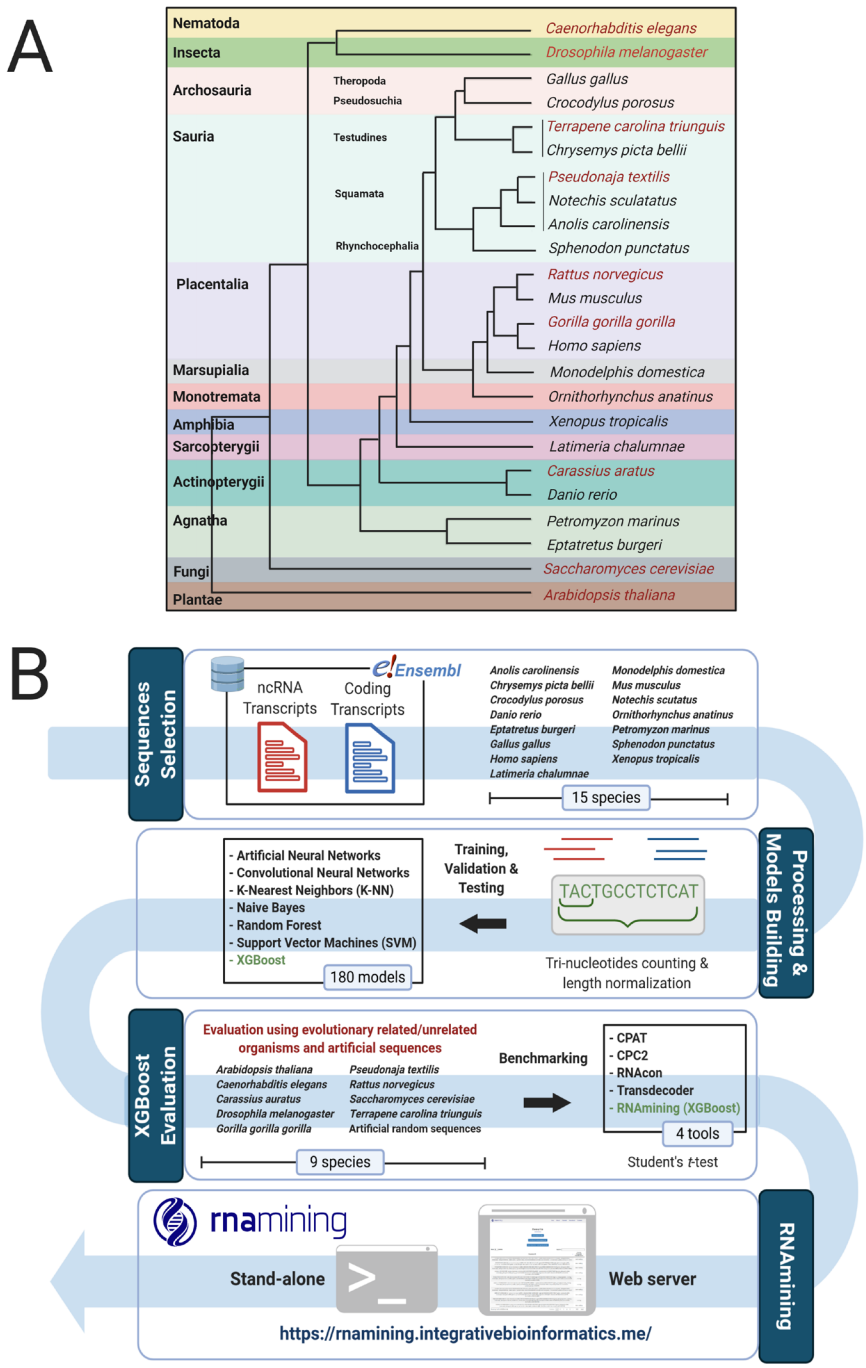

Figure 1. A. Taxonomic tree according to the used organisms for the models building (black color) and validation (red color). **B**. Pipeline used to perform the benchmarking and create the tool. Firstly, we download the coding and non-coding sequences from Ensembl; Next, we performed the trinucleotides counts and sequence normalization. After this, we created a machine learning benchmarking within the 7 algorithms and selected the one with the best performance to be implemented in the RNAmining tool (XGBoost algorithm), which was again evaluated using sequences from 9 other different species and sets of artificially generated ones. Finally, we performed a novel benchmarking with RNAmining against the public available tools for coding potential prediction.

probability (CP) cutoff of 0.364 (*i.e.* CP >=0.364 indicates coding sequence); *M. musculus* with a CP cutoff of 0.44; *D. melanogaster* with a CP cutoff of 0.39; and *D. rerio* with a CP cutoff of 0.38. Therefore, for the other organisms we built new models using our training sets and we used the statistical method provided by the authors to calculate the cutoffs probability for coding prediction: *A. carolinensis* (0.4); *C. picta bellii* (0.57); *C. porosus* (0.38); *E. burgeri* (0.35); *G. gallus* (0.42); *L. chalumnae* (0.365); *M. domestica* (0.51); *N. scutatus* (0.15); *O. anatinus* (0.28); *P. marinus* (0.34); *S. punctatus* (0.18);

Table 1. Set of sequences used in the training and testing datasets. List of organisms and the total number of sequences used for testing and training both coding and non-coding RNAs. The numbers are separated into training/testing values. All sequences can be retrieved at RNAmining website (https://rnamining.integrativebioinformatics.me/download).

Species	Total	Coding	ncRNAs
Models Generation (training / testing):			
Anolis carolinensis	12,542 / 3,136	6,243 / 1,596	6,299 / 1,540
Chrysemys picta bellii	11,260 / 2,816	5,626 / 1,412	5,634 / 1,404
Crocodylus porosus	7,388 / 1,848	3,700 / 918	3,688 / 930
Danio rerio	12,984 / 3,246	6,527 / 1,588	6,457 / 1,658
Eptatretus burgeri	1,742 / 436	867 / 222	875 / 214
Gallus gallus	16,851 / 4,213	8,426 / 2,106	8,425 / 2107
Homo sapiens	92,844 / 23,212	46,575 / 11,453	46,269 / 11,759
Latimeria chalumnae	4,668 / 1,168	2,344 / 574	2,324 / 594
Monodelphis domestica	34,336 / 8,584	17,113 / 4,347	17,223 / 4,237
Mus musculus	35,272 / 8,818	17,668 / 4,377	17,604 / 4,441
Notechis scutatus	2,705 / 677	1,351 / 340	1,354 / 337
Ornithorhynchus anatinus	12,604 / 3,152	6,280 / 1,598	6,324 / 1,554
Petromyzon marinus	4,243 / 1,061	2,107 / 545	2,136 / 516
Sphenodon punctatus	1,456 / 364	723 / 187	733 / 177
Xenopus tropicalis	2,224 / 556	1,120 /270	1,104 / 286
RNAmining Evaluation:			
Arabidopsis thaliana	11,308	5,654	5,654
Caenorhabditis elegans	50,558	25,279	25,279
Carassius auratus	15,004	7,502	7,502
Drosophila melanogaster	31,808	15,904	15,904
Gorilla gorilla gorilla	15,978	7,989	7,989
Pseudonaja textilis	1,486	743	743
Rattus norvegicus	18,662	9,331	9,331
Saccharomyces cerevisiae	848	424	424
Terrapene carolina triunguis	2,054	1,027	1,027

X. tropicalis (0.25). The whole workflow of RNAmining development can be visualized in Figure 1B.

RNAmining tool implementation and availability

The XGBoost method was implemented using XGBoost Library (version 1.2.0) in Python Language (Version 3.8) and the models for each species were saved using pickle Python's library. The web server interface was developed using HTML and CSS. The connection within the front and back-end was implemented through JavaScript. The control of files and the connection with Python's scripts was performed through PHP language. RNAmining user friendly tool and its stand-alone version can be accessed at https://rnamining.integrativebioinformatics.me/. Instructions on how to use it and a whole documentation are made available. Its source code with a Docker platform can be freely obtained at https://gitlab.com/integrativebioinformatics/RNAmining.

Results

Using machine learning algorithms to improve the coding potential prediction of RNA sequences

It is known that the algorithm performance in predictive analysis is influenced by particularities available in the genomes sequences of the organisms used in the training set[22], and it

should be taken into account when developing novel tools for nucleotides coding prediction. Thus, it is necessary to test several methods to observe which ones can have a good prediction for specific species from evolutionary branches. Similar to Panwar *et al.*[10], we used the trinucleotides count to distinguish coding and non-coding sequences. We evaluated the performance of seven machine learning algorithms using representative organisms from different branches of the Chordata clade. For that, we used a training and testing set composed by sequences from the same species. The algorithm with best performance within all evaluated organisms, according to F1-scores metric, was XGBoost, as one can see in the following: *A. carolinensis* (98.79); *C. picta bellii* (98.00); *C. porosus* (98.15); *D. rerio* (97.98); *E. burgeri* (97.56); *G. gallus* (99.24); *H. sapiens* (98.50); *L. chalumnae* (99.57); *M. domestica* (98.84); *M. musculus* (97.73); *N. scutatus* (96.51); *O. anatinus* (97.61); *P. marinus* (99.42); *S. punctatus* (99.20); *X. tropicalis* (99.13) (Table 2). As observed, XGBoost algorithm presented F-score values above 97.00, with the worst performance obtained for *Eptatretus burgeri* with a F-score of 97.56. The best performance was obtained for *Petromyzon marinus* with 99.42. All detailed performances with sensitivity, specificity, precision, accuracy, F1-score and the confusion matrix from each algorithm is listed in Supplementary File S2[19]. Based on these results, XGBoost was selected to be implemented in a novel web server and stand-alone tool for RNA coding potential prediction called RNAmining.

Using RNAmining in evolutionary related and unrelated organisms

To demonstrate the generalization of the model built in our tool, we evaluated its performance using the following nine Chordata and non-Chordata organisms that were not used in our training step: *A. thaliana*; *C. elegans*; *C. auratus*; *D. melanogaster*; *G. gorilla gorilla*; *P. textilis*; *R. norvegicus*; *S. cerevisiae*; *Terrapene carolina triunguis*. In the training set described in the previous topic, we used sequences from representative species from amphibians, birds, mammals, fishes and reptiles. In this new experiment we executed tests using other chordates, but covering other evolutionary groups such as plants, fungi, insects and nematodes. The F1-score obtained values varying from 86.25 to 98.10. The worst performance was when we used the training set from *L. chalumnae* (Sarcopterygii, Coelacanth) to predict the coding potential of known coding genes and ncRNAs from *D. melanogaster* (Insecta, Diptera). However, the best performance was obtained when we applied the training set from *C. picta bellii* (Sauria, Testudines) in coding and ncRNA sequences from *Terrapene carolina triunguis* (Sauria, Testudines). The F1-score for each organism, together with the respective training set evaluated, can be found in Table 3, meanwhile the confusion matrix and the other metrics can be visualized in *Extended data:* Supplementary File S3[19].

Even without using any plant in the original training set, we applied the different models to predict the coding potential of

Table 2. Benchmarking machine learning methods for coding potential prediction based on trinucleotides count. F1-score for each one of the 15 species in which the algorithms were tested. Other metrics (sensitivity, specificity, precision, accuracy and the confusion matrix) used for the comparison of the algorithm's performance were made available at the *Extended data:* Supplementary File S2[19].

Species	ANN	CNN	K-NN	NAIVE BAYES	RANDOM FOREST	SVM	XGBoost
Anolis carolinensis	98.47	98.31	93.55	95.50	98.30	98.03	**98.79**
Chrysemys picta bellii	96.54	96.02	93.54	93.13	96.89	96.04	**98.00**
Crocodylus porosus	96.74	96.48	93.67	93.93	97.26	96.35	**98.15**
Danio rerio	97.54	97.77	95.44	94.55	97.56	97.27	**97.98**
Eptatretus burgeri	94.88	95.69	92.24	94.57	97.35	95.82	**97.56**
Gallus gallus	98.47	98.27	96.87	95.11	98.91	98.06	**99.24**
Homo sapiens	98.01	97.66	96.63	86.00	98.30	96.83	**98.50**
Latimeria chalumnae	99.05	98.72	91.61	98.23	99.56	99.24	**99.57**
Monodelphis domestica	98.39	98.09	97.11	95.31	98.67	98.01	**98.84**
Mus musculus	96.67	96.96	95.95	91.56	97.66	96.10	**97.73**
Notechis scutatus	95.90	94.10	87.77	89,81	94.94	95.73	**96.51**
Ornithorhynchus anatinus	97.23	96.59	93.59	91.45	96.99	96.38	**97.61**
Petromyzon marinus	98.40	98.26	88.10	95.99	98.79	97.49	**99.42**
Sphenodon punctatus	97.83	96.97	78.41	96.70	96.46	95.29	**99.20**
Xenopus tropicalis	98.28	98.81	85.53	97.14	98.88	97.20	**99.13**

Table 3. Evaluation (F1-score) of the models generated by XGBoost, the method implemented in RNAmining, according to evolutionary related and unrelated organisms. Each line comprises the model for each one of the trained species, meanwhile the columns represent the set of 9 evolutionary related and unrelated organisms in which the method was evaluated. Other metrics (sensitivity, specificity, precision, accuracy and the confusion matrix) used for the comparisons were made available at the *Extended data:* Supplementary File S3[19].

Testing *Training*	Arabidopsis thaliana	Caenorhabditis elegans	Carassius auratus	Drosophila melanogaster	Gorilla gorilla	Pseudonaja textilis	Rattus norvegicus	Saccharomyces cerevisiae	Terrapene carolina triunguis
Anolis carolinensis	95.35	89.97	94.77	97.16	95.17	96.56	96.74	93.07	95.83
Chrysemys picta bellii	97.24	97.79	95.97	98.13	97.01	97.73	97.15	96.09	98.10
Crocodylus porosus	96.19	96.76	95.73	97.87	97.01	96.90	97.25	95.07	97.56
Danio rerio	96.64	90.50	95.29	97.96	97.24	96.89	96.42	93.96	96.62
Eptatretus burgeri	94.90	95.57	94.80	96.73	95.34	95.43	95.76	91.49	95.51
Gallus gallus	97.60	97.89	95.76	98.02	97.93	97.79	97.59	96.48	97.69
Homo sapiens	95.71	81.25	92.19	96.44	97.73	96.24	94.60	93.57	95.65
Latimeria chalumnae	93.71	96.78	91.63	86.25	96.30	93.39	94.37	95.47	95.63
Monodelphis domestica	97.40	97.91	95.69	98.04	97.90	97.53	97.46	93.54	97.31
Mus musculus	96.44	87.68	94.66	97.17	97.57	97.31	96.67	94.32	96.30
Notechis scutatus	97.16	97.54	95.22	97.46	97.35	97.37	96.79	94.96	97.22
Ornithorhynchus anatinus	97.39	97.48	95.39	87.74	97.32	97.86	97.29	94.67	97.53
Petromyzon marinus	93.31	94.48	92.07	87.74	95.81	93.47	94.72	92.48	95.56
Sphenodon punctatus	94.00	97.07	91.94	86.89	96.60	93.95	94.12	95.02	95.81
Xenopus tropicalis	93.46	96.65	91.53	84.86	95.51	93.68	93.16	94.42	95.02

F1000Research 2021, 10:323 Last updated: 11 JUN 2021

known coding and ncRNA sequences from *A. thaliana* (Plantae, Eudicots). The lowest F1-score that RNAmining obtained was 93.31 using a fish model (*Petromyzon marinus*, Agnatha, Petromyzontiformes). The best F1-score was obtained with a marsupial model (*M. domestica*, Marsupialia) that reached 97.40. Thus, this experiment demonstrated the efficiency of the method and the models created even when applied in organisms phylogenetically distant from those used in training.

Finally, in order to show that the results obtained were not by chance, we created 10 datasets of artificial sequences containing the same number, length and nucleotides composition of the coding and ncRNA sequences from the 15 species used in our testing shown in Table 1. The F1-score mean, minimum and maximum values of the 10 datasets from each organism can be visualized in Table 5. The confusion matrix and all the other metrics (accuracy, specificity, sensitivity and precision) can be found in *Extended data:* Supplementary File S4[19]. As we can visualize, the F1 measurement mean remained below 38.00 for all artificial sequences created for the tested organisms, with the exception of *P. marinus* (F1-score equals to 64.13), which still had a F1-score below to the values obtained with the other organisms tested for the coding potential prediction (Table 4).

Comparing RNAmining performance with publicly available tools

Next, we compared RNAmining performance with other four tools commonly used for nucleotides coding potential prediction: CPAT, CPC2, RNAcon and TransDecoder. We used as input all coding and ncRNA sequences from the testing dataset used in the 15 species listed in Table 1. According to the F1-score metric, RNAmining outperformed all the tools in all organisms with the exception of CPAT for *L. chalumnae*, in which both tools presented an equal F1-score of 99.57. The comparative performance of all tools can be observed in Table 5. The detailed results regarding their accuracy, sensitivity, specificity, precision, F1-score and the confusion matrix can be found in Supplementary File S2[19]. Finally, we used the t-student test to compare the results from RNAmining and the other tools, revealing that our software presented significantly better results in performing coding potential predictions based on known coding genes and ncRNAs. The p-values obtained in these comparisons were: 0.0026 (*vs* CPAT); 1.57e-05 (*vs* CPC2); 2.69e-05 (*vs* RNAcon); and 2.89e-05 (*vs* TransDecoder).

RNAmining stand-alone and web server tool

RNAmining tool was made available in both stand-alone and web server versions. The tools only require the nucleotide

Table 4. Evaluation of RNAmining performance according to different sets of artificial sequences from each trained model. F1-score metrics for 10 datasets of artificial sequences randomly generated for each species. The mean, minimum and maximum values are displayed separated by organism. Other metrics (sensitivity, specificity, precision, accuracy and the confusion matrix) used for the comparisons were made available at the *Extended data:* Supplementary File S4[19].

Species	MEAN	MINIMUM	MAXIMUM
Anolis carolinensis	1.66	0.86	2.44
Chrysemys picta bellii	1.08	0.70	1.40
Crocodylus porosus	0.95	0.43	1.72
Danio rerio	1.25	0.12	2.21
Eptatretus burgeri	2.31	0.90	3.51
Gallus gallus	2.48	1.88	2.89
Homo sapiens	11.15	10.53	11.52
Latimeria chalumnae	24.86	21.95	27.03
Monodelphis domestica	1.34	1.00	1.18
Mus musculus	6.64	5.74	7.58
Notechis scutatus	1.80	0.58	3.99
Ornithorhynchus anatinus	3.62	2.67	5.04
Petromyzon marinus	64.13	62.99	65.76
Sphenodon punctatus	37.43	31.72	41.84
Xenopus tropicalis	23.26	17.65	28.21

Table 5. Benchmarking results from RNAmining and the other tools already described in the literature according to organisms from different evolutionary branches. F1-score metric for CPAT, CPC2, RNAcon, TransDecorder and RNAmining, based on the predictions using models provided by each tool or generated according to their instructions. The bold numbers are the best values regarding F1-score metric. The results for other metrics were made available at the *Extended data:* Supplementary File S2[19].

Species	CPAT	CPC2	RNAcon	TransDecoder	RNAmining
Anolis carolinensis	94.55	86.87	83.03	88.26	**98.79**
Chrysemys picta bellii	92.56	89.01	82.36	84.80	**98.00**
Crocodylus porosus	94.07	92.48	84.32	87.63	**98.15**
Danio rerio	94.64	87.17	80.97	87.74	**97.98**
Eptatretus burgeri	95.59	78.82	75.84	76.26	**97.56**
Gallus gallus	96.95	90.69	75.81	83.50	**99.24**
Homo sapiens	95.20	75.85	71.73	76.02	**98.50**
Latimeria chalumnae	**99.57**	91.60	97.45	98.86	**99.57**
Monodelphis domestica	96.24	91.44	80.90	85.22	**98.84**
Mus musculus	95.48	81.40	76.78	80.80	**97.73**
Notechis scutatus	85.19	86.29	84.83	83.44	**96.51**
Ornithorhynchus anatinus	87.47	72.04	84.73	84.63	**97.61**
Petromyzon marinus	96.59	75.14	95.11	96.68	**99.42**
Sphenodon punctatus	97.61	91.91	97.86	95.24	**99.20**
Xenopus tropicalis	99.07	97.92	98.70	97.77	**99.13**

sequences of the RNAs in which the user intends to perform the coding potential prediction in FASTA format, together with the species name in a standardized format related to the model to be used. Besides our tool presented good results even when using phylogenetically distant organisms, we recommend to always use the most closely related species to the one the user wants to perform the predictions. Furthermore, RNAmining documentation presents all the guidelines on how to generate a model for a particular set of sequences and organisms of interest. The web interface of RNAmining tool was developed to allow users to quickly perform the coding potential prediction without the need of installing any specific program and using only a generic internet browser. The only requirement for running the tool is a FASTA file containing the nucleotide sequences and the organism model that the user wants to use, which can be selected in a drop-down menu containing all 15 organisms used in the training step (Figure 2A). There is no limit of the number of sequences, but the web server supports files up to 20Mb. For files bigger than that, we recommend using the stand-alone RNAmining tool. RNAmining will automatically classify the FASTA sequences used as input and identify which of them are coding or non-coding RNAs. Finally, as a result it offers a table with the sequences' IDs, its classification as coding or non-coding and the classification probabilities, which can also be downloaded in tabular format, together with two separate FASTA files containing both the coding and non-coding sequences separately (Figure 2B).

Discussion

The coding potential prediction of nucleotides is a key step in the definition of the repertoire of non-coding RNAs in a genome or transcriptome project, especially when dealing with non-model organisms. Sometimes, predictive tools for the computational characterization of RNA molecules in analyses like the prediction of specific RNA families[22] or the estimation of a network of RNA-RNA[23] or protein-RNA interactions[24], have their performance affected according to the training organism, increasing the number of false positives when applied in evolutionarily distant species. In this work, we evaluated the performances of seven different supervised machine learning algorithms, using eukaryotic species from a variety of evolutionary clades, revealing their potential to be used in the development of novel and improved computational tool for the coding potential prediction of RNA sequences. Artificial intelligence has been widely used in computational biology[25,26], but its application to characterize ncRNAs has been limited.

In this benchmarking, we opted to analyze the trinucleotides count as the main feature to be evaluated for the coding potential prediction, followed by a normalization considering the

Figure 2. RNAmining web server overview. A. Job launcher screen (Run tab). The user only needs to upload the nucleotide sequences in FASTA format and select the model to be used based on the evolutionary close related species. **B**. Results web page screen. General report containing the list of coding and non-coding sequences in a dynamic table, in which the user can search for a particular sequence or filter only those coding or non-coding RNAs by using a free text form that will filter the results in the table dynamically. The user can download the complete table in tabular format and two FASTA files containing the set of coding and non-coding RNAs separately.

sequences length (*i.e.* each trinucleotides count was divided by the total size of the given sequence). Panwar *et al.*[10] used nucleotides counting successfully for this purpose. They considered 40,905 non-coding RNAs from Rfam release 10.0 database and 62,473 coding RNA sequences from Human RefSeq database, divided into 50% of training and 50% of test (*i.e.* the training and test sets were composed of 20,453 non-coding and 31,237 coding sequences). They used the counts of mono-, di-, tri-, tetra- and penta-nucleotides and a combination of all counts using the SVM method, and showed that using trinucleotides count is enough to predict the coding potential of ncRNAs with better accuracies. Our comparisons of the machine learning algorithms revealed XGBoost as the algorithm with better performance, presenting efficiency in predicting the coding potential of RNA sequences even when using the models of distantly related organisms. This latter shows the usefulness of this approach for performing coding predictions in non-model organisms.

We implemented XGBoost in RNAmining, a stand-alone and web server tool flexible to be used in genome or transcriptome projects focused in both model and non-model eukaryotic organisms. Our tool outperformed similar approaches, such as CPAT[11], CPC2[13], RNAcon[10] and TransDecoder[12]. Both versions of the software are easy to use, with the web version providing a simple report and FASTA format files that can be used in downstream analysis. It provides 15 models generated from eukaryotic from different evolutionary clades. Other models can be generated by the user using the stand-alone version, which can be used with simple command line operations. These features facilitate its usage for experienced users and, especially, for those without any programming experience, which can easily perform large-scale predictions of the coding potential of nucleotide sequences in both genome or transcriptome initiatives.

Conclusions

- We used pattern recognition approaches to investigate the coding potential prediction of RNAs, using 64 features (all combinations of trinucleotides count).
- We performed a benchmarking from seven machine learning algorithms (Naive Bayes, SVM, K-NN, Random Forest, XGBoost, ANN and DL), through 15 model organisms from different evolutionary branches and implemented the best one (XGBoost) in a novel tool (RNAmining).
- RNAmining is a user-friendly coding potential prediction web tool that performs XGBoost algorithm to predict the coding potential of RNA sequences.
- RNAmining was evaluated using other phylogenetically related and unrelated organisms that were not used in our training, demonstrating the efficiency of the tool even when applied in species phylogenetically distant from those used in training.
- A comprehensive analysis using data from 15 organisms revealed that RNAmining outperformed other tools available in literature (CPAT, CPC2, RNAcon and TransDecoder).

Data availability

Underlying data

Ensembl is an open access genome browser for vertebrate genomes in the Ensembl website (https://www.ensembl.org/index.html).

RNAmining is a tool for coding potential prediction which is freely available at (https://rnamining.integrativebioinformatics.me/download).

Extended data

Zenodo: RNAmining Software Supplementary Material, http://doi.org/10.5281/zenodo.4699571[19]

This project contains the following extended data:

- Supplementary File S1: ANN and DL parameters
- Supplementary File S2: All metrics used for the comparison of the algorithm's performance from the 15 model organisms.
- Supplementary File S3: All metrics used for the XGBoost algorithm's performance from the 9 evolutionary related and unrelated organisms in which the method was evaluated.
- Supplementary File S4: All metrics used for the XGBoost algorithm's performance from the artificial sequences created for the tested organisms.

Software availability

RNAmining is available from: https://rnamining.integrativebioinformatics.me/

Source code available from: https://gitlab.com/integrativebioinformatics/RNAmining/-/tree/master/volumes/rnamining-front/assets/scripts/ and https://github.com/thaisratis/RNAmining

Archived source code as at time of publication: https://doi.org/10.5281/zenodo.4891914[27]

License: MIT

Acknowledgements

The authors would like to thank Dr. Savio Torres de Farias for the helpful discussions during the preparation of this manuscript.

A previous version of this article can be found on bioRXiv: https://doi.org/10.1101/2020.10.26.354357

References

1. Mattick JS: **The central role of RNA in the genetic programming of complex organisms.** *An Acad Bras Cienc.* 2010; **82**(4): 933–939.
PubMed Abstract | Publisher Full Text

2. Gelsinger DR, DiRuggiero J: **The Non-Coding Regulatory RNA Revolution in Archaea.** *Genes (Basel).* 2018; **9**(3): 141.
PubMed Abstract | Publisher Full Text | Free Full Text

3. Croce CM: **Causes and consequences of microRNA dysregulation in cancer.** *Nat Rev Genet.* 2009; **10**(10): 704–714.
PubMed Abstract | Publisher Full Text | Free Full Text

4. Schaefer A, O'Carroll D, Tan CL, *et al.*: **Cerebellar neurodegeneration in the absence of microRNAs.** *J Exp Med.* 2007; **204**(7): 1553–1558.
PubMed Abstract | Publisher Full Text | Free Full Text

5. Zhao Y, Ransom JF, Li A, *et al.*: **Dysregulation of cardiogenesis, cardiac conduction, and cell cycle in mice lacking miRNA-1-2.** *Cell.* 2007; **129**(2): 303–317.
PubMed Abstract | Publisher Full Text

6. Djebali S, Davis CA, Merkel A, *et al.*: **Landscape of transcription in human cells.** *Nature.* 2012; **489**(7414): 101–108.
PubMed Abstract | Publisher Full Text | Free Full Text

7. Kellis M, Wold B, Snyder MP, *et al.*: **Defining functional DNA elements in the human genome.** *Proc Natl Acad Sci U S A.* 2014; **111**(17): 6131–6138.
PubMed Abstract | Publisher Full Text | Free Full Text

8. Arias-Carrasco R, Vásquez-Morán Y, Nakaya HI, *et al.*: **StructRNAfinder: an automated pipeline and web server for RNA families prediction.** *BMC Bioinformatics.* 2018; **19**(1): 55.
PubMed Abstract | Publisher Full Text | Free Full Text

9. Torres F, Arias-Carrasco R, Caris-Maldonado JC, *et al.*: **LeishDB: a database of coding gene annotation and non-coding RNAs in *Leishmania braziliensis*.** *Database (Oxford).* 2017; **2017**: bax047.
PubMed Abstract | Publisher Full Text | Free Full Text

10. Panwar B, Arora A, Raghava GPS: **Prediction and classification of ncRNAs using structural information.** *BMC Genomics.* 2014; **15**: 127.
PubMed Abstract | Publisher Full Text | Free Full Text

11. 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**(6): e74.
PubMed Abstract | Publisher Full Text | Free Full Text

12. Haas BJ, Papanicolaou A, Yassour M, *et al.*: ***De novo* transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis.** *Nat Protoc.* 2013; **8**(8): 1494–1512.
PubMed Abstract | Publisher Full Text | Free Full Text

13. 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**(W1): W12–W16.
PubMed Abstract | Publisher Full Text | Free Full Text

14. Chang CC, Lin CJ: **LIBSVM: A library for support vector machines.** *ACM Trans Intell Syst Technol.* 2011; **2**(3): 1–27.
Publisher Full Text

15. Nachtigall PG, Kashiwabara AY, Durham AM: **CodAn: predictive models for precise identification of coding regions in eukaryotic transcripts.** *Brief Bioinform.* 2021; **22**(3): bbaa045.
PubMed Abstract | Publisher Full Text | Free Full Text

16. Ito E, Katahira I, Vicente F, *et al.*: **BASiNET—BiologicAl Sequences NETwork: a case study on coding and non-coding RNAs identification.** *Nucleic Acids Res.* 2018; **46**(16): e96.
PubMed Abstract | Publisher Full Text | Free Full Text

17. Pedregosa F, Varoquaux G, Gramfort A, *et al.*: **Scikit-learn: Machine Learning in Python.** *J Mach Learn Res.* 2011; **12**: 2825–2830.
Reference Source

18. **Python API Reference — xgboost 1.3.0-SNAPSHOT documentation**. [cited 14 Oct 2020].
Reference Source

19. Ratis T, Galindo N: **RNAmining Software Supplementary Material [Data set].** *Zenodo.* 2021.

20. Zerbino DR, Achuthan P, Akanni W, *et al.*: **Ensembl 2018.** *Nucleic Acids Res.* 2018; **46**(D1): D754–D761.
PubMed Abstract | Publisher Full Text | Free Full Text

21. Bailey TL, Boden M, Buske FA, *et al.*: **MEME SUITE: tools for motif discovery and searching.** *Nucleic Acids Res.* 2009; **37**(Web Server issue): W202–8.
PubMed Abstract | Publisher Full Text | Free Full Text

22. Aguiar RR, Ambrosio LA, Sepúlveda-Hermosilla G, *et al.*: **miRQuest: integration of tools on a Web server for microRNA research.** *Genet Mol Res.* 2016; **15**(1).
PubMed Abstract | Publisher Full Text

23. Umu SU, Gardner PP: **A comprehensive benchmark of RNA-RNA interaction prediction tools for all domains of life.** *Bioinformatics.* 2017; **33**(7): 988–996.
PubMed Abstract | Publisher Full Text | Free Full Text

24. Nithin C, Mukherjee S, Bahadur RP: **A non-redundant protein-RNA docking benchmark version 2.0.** *Proteins.* 2017; **85**(2): 256–267.
PubMed Abstract | Publisher Full Text

25. de Brito DM, Maracaja-Coutinho V, de Farias ST, *et al.*: **A Novel Method to Predict Genomic Islands Based on Mean Shift Clustering Algorithm.** *PLoS One.* 2016; **11**(1): e0146352.
PubMed Abstract | Publisher Full Text | Free Full Text

26. Ramos TAR, Maracaja-Coutinho V, Ortega JM, *et al.*: **CORAZON: a web server for data normalization and unsupervised clustering based on expression profiles.** *BMC Res Notes.* 2020; **13**(1): 338.
PubMed Abstract | Publisher Full Text | Free Full Text

27. Ratis T, Galindo N: **thaisratis/RNAmining: RNAmining (Version v1.0.4).** *Zenodo.* 2021.

Open Peer Review

Current Peer Review Status: ✔ ✔

Version 2

Reviewer Report 11 June 2021

https://doi.org/10.5256/f1000research.57466.r87006

✔ **Gilderlanio Araújo**
Laboratory of Human and Medical Genetics, Postgraduate Program in Genetics and Molecular Biology, Universidade Federal do Pará, Belém, Brazil

The authors made the corrections. The authors clarified the text and added prediction probabilities to the software.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics. Machine Learning. Transcriptome Analysis. Population Genomics.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 09 June 2021

https://doi.org/10.5256/f1000research.57466.r87007

✔ **Andre Yoshiaki Kashiwabara**
Department of Computer Science, Bioinformatics Graduate Program, Federal University of Technology - Parana, Curitiba, Brazil

The authors have addressed all my concerns.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, Computational Biology, Machine Learning.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 25 May 2021

https://doi.org/10.5256/f1000research.55616.r84936

?

Andre Yoshiaki Kashiwabara
Department of Computer Science, Bioinformatics Graduate Program, Federal University of Technology - Parana, Curitiba, Brazil

This paper presents RNAmining to predict the protein-coding potential of transcripts. The authors have compared many algorithms using cross-validation and selected XGBoost. The tool has the potential to be very useful. It is available online, and it is easy to use.

1. In recent studies, some small ORF in annotated ncRNA has validated protein-coding potential [1]. How does RNAmining behave when these annotated ncRNAs that contain such small ORF challenge it?

2. It is important to cite RNAploc [2], BASINET [3], and CoDaN [4].

3. I suggest plotting ROC curves when comparing classification methods.

4. In the abstract, the authors have described the use of cross-validation to assess the accuracy of each classification method. However, in methodology, the author explained that 80% is the training set and the other 20% is the validation set. And for CNN, and ANN this number is also different (60%/20%). It isn't clear. It will help create a figure showing the "workflow" of the Training/Validation/Testing part.

5. The standard deviation of the cross-validation can be helpful to show the stability of each classification method.

References

1. Xing J, Liu H, Jiang W, Wang L: LncRNA-Encoded Peptide: Functions and Predicting Methods. *Front Oncol*. 2020; **10**: 622294 PubMed Abstract | Publisher Full Text

2. Paschoal AR, Maracaja-Coutinho V, Setubal JC, Simões ZL, et al.: Non-coding transcription characterization and annotation: a guide and web resource for non-coding RNA databases.*RNA Biol*. 2012; **9** (3): 274-82 PubMed Abstract | Publisher Full Text
3. Ito E, Katahira I, Vicente F, Pereira L, et al.: BASiNET—BiologicAl Sequences NETwork: a case study on coding and non-coding RNAs identification. *Nucleic Acids Research*. 2018; **46** (16). Publisher Full Text
4. Nachtigall PG, Kashiwabara AY, Durham AM: CodAn: predictive models for precise identification of coding regions in eukaryotic transcripts.*Brief Bioinform*. 2021; **22** (3). PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: I am the author of CoDaN (https://academic.oup.com/bib/article/22/3/bbaa045/5847603)

Reviewer Expertise: Bioinformatics, Computational Biology, Machine Learning.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 02 Jun 2021

Thaís A. R. Ramos, Universidade Federal do Rio Grande do Norte, Natal, Brazil

1- RNAmining was trained using coding genes and ncRNAs from the Ensembl database. It evaluates the patterns in the tri-nucleotide counts in any RNA sequence (which could be an ORF or not) and, according to this, it classifies into coding and non-coding sequences. The RNAmining training was not performed with the proposal of generating a specific model for ORFs or some specific type of sequence, it works independently of this.

2- Thank you for your suggestion. We included the citations for BASiNET and CoDaN in the Introduction section of the manuscript. We were not able to find the manuscript describing RNAploc and it was not included.

3- We believe that it is possible to visualize the performance of the methods from the tables presented along the main text of the paper, as well as made available as supplementary material, since the measures used for the ROC curves construction are the same presented there. In addition, we consider that when we have too similar numbers it is easier to see the difference in a table.

4- The cross-validation method was used in the grid search method using the training dataset to validate the hyperparameters, choosing the best set of parameters. In addition, this partition method validates the hyperparameter's results through different validation sets. Therefore, it proves that our model is working and generalizing the problem. Thus, when we had the best parameters we used the test dataset (20%) to generate the final models and to calculate the metrics. The connectionist methods (e.g. Artificial Neural Networks and Convolutional Neural Networks) demand a validation dataset to adjust the model, because of the weights optimization stage and its hyperparameters. Thus, for experiments with ANN and CNN, 20% was used for validation, 60% for training (defined as 80% for the other algorithms) and 20% for testing, in all the cases. In addition, due to the complexity of these 2 algorithms, it is more common in literature to use the holdout (training/validation/test) partition method instead of cross-validation. Thereby, we modified the sentence in the abstract: "All the machine learning algorithms tests were performed using 10-folds cross-validation..." to "The machine learning algorithms validations were performed using 10-fold cross-validation...". In addition, in the section "Training and testing datasets, model building and quality measuring for coding potential evaluation" we changed the following sentence: "Sequences were randomly divided into training and testing datasets, using 80% of the data for training and 20% for testing. For ANN and CNN experiments, sequences were split into 60% of the data for training and 20% for validation. The testing dataset was the same used in the other machine learning algorithms." to the following text: "The cross-validation approach was applied in the grid search method, using the training dataset to validate the hyperparameters and obtain the best set of this to be used. In addition, this partition method validates the hyperparameter's results through different validation sets. Therefore, it proves that our model is working and generalizing the problem. Thus, sequences were randomly divided into training and testing datasets, using 80% of the data for training and 20% for testing. The connectionist methods (e.g. Artificial Neural Networks and Convolutional Neural Networks) demand a validation dataset to adjust the model, because of the weights optimization stage and its hyperparameters. Thus, for experiments with ANN and CNN, 20% were used for validation, 60% for training (defined as 80% for the other algorithms) and 20% for testing.The testing dataset was the same used in all machine learning algorithms."

5- The results shown in this paper are not obtained using cross-validation. The cross-validation method was used in the grid search method to validate the hyperparameters, choosing the best set of parameters. Thus, we can validate the hyperparameter's results through different validation sets. Therefore, it proves that our model is working and generalizing the problem. Thus, to generate the final models and to calculate the metrics,

we used the test dataset (20%) with the best parameters.

Competing Interests: No competing interests were disclosed.

Reviewer Report 10 May 2021

https://doi.org/10.5256/f1000research.55616.r83973

?

Gilderlanio Araújo
Laboratory of Human and Medical Genetics, Postgraduate Program in Genetics and Molecular Biology, Universidade Federal do Pará, Belém, Brazil

1. Are there two different datasets of model organisms?
   On the abstract "...**15 organisms** from different evolutionary branches..."
   On the main text "RNAmining was evaluated through **24 organisms** from the eukaryotic tree of life and its results outperformed publicly available tools commonly used for that purpose."

2. Why fine-tuning SVM was performed with a grid search strategy and not Random Forest too? Provide some reasoning.

3. Why sequences were divided using different proportions? Provide some reasoning.
   "Sequences were randomly divided into training and testing datasets, using **80% of the data for training and 20%** for testing. For ANN and CNN experiments, sequences were split into **60% of the data for training and 20%** for validation."

4. On the web tool, you should provide a column for prediction probability for coding and non-coding variants. The new column will improve user analysis, such as filtering for those predictions with XGboost >= 0.9.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets

and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics. Machine Learning. Transcriptome Analysis. Population Genomics.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 02 Jun 2021

Thaís A. R. Ramos, Universidade Federal do Rio Grande do Norte, Natal, Brazil

1- The connectionist methods (e.g. Artificial Neural Networks and Convolutional Neural Networks) demand a validation dataset to adjust the model, because of the weights optimization stage and its hyperparameters. Thus, for experiments with ANN and CNN, 20% were used for validation, 60% for training (defined as 80% for the other algorithms) and another 20% for testing, in all the cases.

2- Yes, in fact we have the dataset 1 composed of 15 model organisms which were used to build the models, and the dataset 2 composed of other 9 phylogenetically related and unrelated organisms that were not used in our training, demonstrating the efficiency of the tool even when applied in species phylogenetically distant from those used in training. On the main text, we changed this sentence in the “Introduction” section to: "Next, RNAmining was evaluated in another 9 phylogenetically related and unrelated organisms that were not used in our training, demonstrating the efficiency of the tool even when applied in species phylogenetically distant from those used in training. In total, it was evaluated through 24 organisms from the eukaryotic tree of life and its results outperformed publicly available tools commonly used for that purpose.”

3- In fact all the methods were executed with the grid search method. We made a mistake in the writing. It was modified by replacing the sentences: "The Random Forest model was implemented using empirical tests and the best result was selected for training the model. We considered the default parameters with the exception of the number of trees used (150 estimators) and the criterion parameter setted to 'entropy' for information gain. KNN and Naive Bayes models were trained with the default values. The SVM parameters were obtained through grid search method and the resulting model was trained with the Radial Basis Function (RBF) kernel, with the Regularization parameter (C) and Kernel coefficient (Gamma) defined in 1000 and 0.8, respectively. ANN and DL were performed with different architectures according to grid search and empirical tests" to the following: "XGBoost, K-NN

and Naive Bayes models were trained with the default values. The Random Forest and SVM parameters were obtained through grid search method, the best results using Random Forest resulted in a model generated with the default parameters, with the exception of the number of trees used (150 estimators) and the criterion parameter setted to 'entropy' for information gain. For SVM, the resulting model was trained with the Radial Basis Function (RBF) kernel, with the Regularization parameter (C) and Kernel coefficient (Gamma) defined in 1000 and 0.8, respectively. ANN and DL were performed with different architectures according to grid search and empirical tests."

4- Thank you for your suggestion. We considered it and provided a new column in the output file (Classification probabilities) in both web server and stand-alone versions.

Competing Interests: No competing interests were disclosed.