Utilizing RNNs and Ensemble Learning for Enhanced Bacterial sRNA Classification

Honours Dissertation

April 2019

Moustafa Elsisy
Supervisor: Dr. Lourdes Peña-Castillo
Winter 2019

Abstract

Bacterial small RNAs (sRNAs) are involved in the regulation of gene expression within the majority of bacterial species. These RNA molecules are traditionally validated using laboratory-based techniques. Unfortunately, there is a significant amount of potential sRNAs in the literature, which makes it prohibitive to validate all of them using such methods. While a number of models have recently been proposed to computationally identify sRNAs, they do not achieve high magnitudes of both precision and recall when evaluated over real subsequences of the genome. Here we propose a machine-learning model that utilizes sequence-based features of RNA molecules along with features relating to their genomic context in order to distinguish sRNAs from random genomic sequences. We used a dataset consisting of five different bacterial species, through which we composed a feature space based on the tetranucleotide composition, free energy of predicted secondary structure, distances to the closest predicted promoter/terminator sites, distances to the closest left/right ORF, and whether the sRNA is transcribed on the same strand as each ORF. The proposed model exhibits an Average Precision score of 0.75 over real genomic subsequences, thus improving state-of-the-art performance in sRNA identification. Our results indicate that our feature space is conserved across bacterial species, and that combining sequence-based features with genomic context ones yields a model that outperforms preceding models.

Acknowledgements

This work would certainly have not neared its potential if it was not for the continuous support and encouragement I have received from my supervisor. I would like to wholeheartedly thank Dr. Lourdes Peña-Castillo for giving me the chance to endeavour on a journey to the limits of human accomplishments as they stand today, for entrusting me on an opportunity to further that boundary a bit further, and for the invaluable insights I have had the pleasure to acquire on such a journey.

— Moustafa Elsisy

Contents

1 Introduction

Bacterial small RNAs (sRNAs) are RNA molecules produced by the majority of bacteria species, whose purpose is to regulate the expression of genes by means of binding to RNAs/proteins and interacting with ribosome-binding sites [1]. As a result, sRNAs are known to be involved in the control of multiple responses exhibited by bacteria towards their environment, and in the regulation of genes that code for proteins [2].

Traditionally, sRNAs have been identified using laboratory-based techniques [3] [4]. However, the significant number of presumed newly discovered sRNAs makes it prohibitive to perform a comprehensive validation of all these candidates while relying on such methods. Indeed, the use of computational methods, ranging from comparative genomics [5] to (most recently) machine learning models -including estimators, such as Support Vector Machines (SVMs) [6] and ensemble learners [7] [8]- are recently being investigated as means to overcome this obstacle. The goal is to get as close as possible to a perfect sRNA classifier, in order for the model to act as a reliable source of prioritization for the most promising sRNA candidates, which in turn will be verified in the lab. As a result, the success criteria for solving this problem is to create a model, that does not only exhibit high levels of precision, but also at a high level of recall. To our understanding, this success criteria has not been sufficiently met yet.

Earlier comparative genomics methods seem to have drawn inspiration from tools that are quite popular in computational linguistics, such as Hidden Markov Models (HMMs) and Context-Free Grammars (CFGs) [5] [9]. Even recent machine learning models, such as the SVM proposed by Barman et al. [6] and the models proposed by Tang et al. [8], make significant use of nucleotide k-mer profiles, which are analogous to the notion of n-grams in computational linguistics.

These methods are now known to be quite classical in the linguistics field, while newer methods such as Recurrent Neural Networks (RNNs) have shown, given a sufficient amount of data, to be considerably superior [10]. It is worth noting that there seems to be a significant amount of similarity between the nature of sequence-based features of RNAs and that of features in linguistics. This realization has been pretty fundamental to the approach we opted for in our proposed model; indeed, RNNs have shown great success in the field of bioinformatics, for instance in predicting transcription factor binding sites [11].

The performance of the existing machine learning models for the task of sRNA classification, as reported in the literature, seems very promising. Barman et al.'s SVM has been reported to exhibit an Area under the Receiver Operating Characteristic Curve (AUC) of ~0.94 [6], while Tang et al.'s Neural Network Ensemble Method (NNEM) has subsequently exhibited an AUC of ~0.96 [8]. However, as pointed out by Eppenhof et al.,

Barman et al.'s SVM performance degrades drastically when the negative instances of the dataset are (real) random non-sRNA subsequences of the genome (as opposed to shuffled/artificial genomic sequences), exhibiting an Area under the Precision-Recall Curve (AUPRC) of only ~0.18 [7]. Tang et al. have also relied on shuffling genomic sequences to generate negative instances, hence we think it is likely that their proposed model is susceptible to the same issue.

On the other hand, Eppenhof et al. have reported an AUPRC of ~0.65, as exhibited by their genomic context random forest. Genomic context refers to the use of attributes describing the location of potential sRNAs with respect to genomic features such as promoters, terminators and open reading frames (ORFs). This performance has been achieved while using non-sRNA subsequences of the genome as negative instances. [7]

Our proposed model is an ensemble of the genomic context model proposed by Eppenhof et al., and a novel RNN tuned for extracting sequence-based features out of sRNAs, which provides a prediction score to be used along side the rest of the ensemble. Our proposed model has exhibited an Average Precision (AP) score of ~0.75 on the Eppenhof et al. dataset, enhancing the performance of what has been reported in the literature so far, as evaluated over real subsequences of the genome as negative instances. In this thesis, additionally to our proposed model, we will also discuss a number of other models that we have investigated prior to the proposed model - some of which have also enhanced over the performance reported in the literature.

2 Related Works

The search for methods to identify different kinds of RNA molecules, in a relatively inexpensive manner when compared to laboratory methods, has been ongoing for at least a couple of decades [5]. Some of the earlier proposed methods for accomplishing this task rely heavily on comparative genomics [5] [12] and/or on free energy methods [13] [14], while some older [15] and a significant amount of modern methods [16] [6] [7] [8] choose to take a machine-learning approach to the problem.

2.1 Comparative Genomics Methods

Comparative genomics identifies sRNAs by primarily performing sequence alignment between the candidate molecule and known sRNAs from different bacteria, and subsequently quantifying sequence and structural homology between the candidate molecule and the known sRNAs [17]. QRNA utilized pair-HMM and pair-stochastic-CFGs (pair-SCFGs) to construct three models that are capable of identifying coding

genes, structural RNAs or "something else" respectively, by observing the pattern of substitutions in a pairwise alignment of two sequences [5]. Klein et al. were able to predict thirty new non-coding RNA genes (ncRNAs) and verify nine of them to be bona-fide, by using QRNA in conjunction with a screening for regions rich in the dinucleotide "GC", within organisms that are "AT" rich. Klein et al. do note that their proposed dinucleotide screening method is meant to work on hyperthermophilic organisms that exhibit such a DNA composition bias [12].

Lu et al. have analyzed the performance of four major comparative genomics methods used for sRNA classification: QRNA [5], RNAz [18] [19], sRNAPredict3/SIPHT [20] [21] and NAPP [22]. They have observed sensitivities in the range of 2% - 71%, with corresponding precisions of 0% - 24%. They have noted that the existing methods seem to significantly trade-off precision for sensitivity, or vice-versa (as opposed to exhibiting a high magnitude of both sensitivity and precision). Their results also indicate that comparative genomics models exhibit significantly degraded performance as the evolutionary distance between the taxa of bacteria used increases, and attribute that issue to the lack of conservation of sRNA sequences across taxa, especially distantly-related ones [23].

2.2 Free Energy Methods

Free energy methods that are available in the literature seem to utilize free energy as a feature in different ways. Washietl et al. composed an energy score out of free energy over alignments of RNA molecules and a covariation term, and utilized it for comparative genomics in order to detect functional RNAs (fRNAs). This method has yielded a significantly higher recall than that of single sequence predictions [13]. Uzilov et al. have shown that predicting common secondary structures between two RNA molecules, by minimizing the folding free energy change and scoring structural alignment, can be used effectively for classifying ncRNA [14]. The program used for this purpose is called Dynalign [24]. Over tRNAs, and as reported by Uzilov et al., their approach has exhibited precision of 92.7 ± 14.3 at a recall of 92.9 ± 12.6. [14]

However, in a study by Vockenhuber et al. to identify new sRNAs in *Streptomyces coelicolor* by means of deep-sequencing, Dynalign was tested, and out of the 639 potential sRNAs predicted by Dynalign, only 46 have overlapped with the 276 sRNAs predicted by the sequencing (estimated ~7.2% precision at ~16.7% recall). This study does note that the sRNAs identified in *Streptomyces coelicolor* have only been found to be conserved within the *Streptomyces* genus, and suggested that such non-conserved sRNAs are ill-suited for such comparative genomics programs [3].

2.3 Machine Learning (ML) Methods

While comparative genomics methods were very popular, Carter et al. proposed a machine-learning approach using Neural Networks (NNs) and SVMs to predict fRNAs. They used 1-spectrum (1-mer) and 2-spectrum (2-mer) profiles, along with known RNA sequence motifs and calculated free energy of folding as features for their models. Their model has shown some potential to predict RNA sequences across different organisms from related species [15]. Unfortunately, that claim has been justified using accuracy as a metric, and while that does not necessarily nullify the validity of that potential, it is susceptible to bias introduced by class-imbalance in the testing dataset.

Noting the need for a classifier of sRNAs that achieves both high precision and recall, Arnedo et al. composed an ensemble of comparative genomics and free energy tools to identify sRNAs (including QRNA [5], Alifoldz [13] and Dynalign [24], among others [25] [19] [26]). They used an evolutionary algorithm to optimize the aggregation of the predictions generated by the different predictors. Their model exhibited 78% specificity at 67% sensitivity on the *Salmonella enterica* serovar Typhimurium LT2 (SLT2) dataset [16]. The SLT2 dataset has become a quite common benchmark dataset in subsequent papers on the topic of sRNA classification [6] [7] [8].

In an attempt to further improve the standard of sRNA classification, Barman et al. have proposed an SVM that uses the trinucleotide composition (i.e. 3-spectrum profile) of the RNA molecule to predict whether it is an sRNA molecule or not. While Arnedo et al.'s model has shown an accuracy of 72.50% on the SLT2 dataset, Barman et al.'s model has surpassed that by exhibiting an accuracy of 88.35% [6] (it is worth noting that using accuracy as a metric for comparing the relative performance of these two models is reasonable, since they have been tested on the same dataset).

While a sensitivity-specificity pair can change for a given model by varying its classification threshold, Barman et al.'s model seems to indeed exhibit a high level of precision *and* recall (91.58% precision at 85.11% recall), compared to the performance shown by Arnedo et al.'s model (~78% precision at ~67% recall), further reinforcing the observation made from the accuracy metric. According to Barman et al.'s results, most of the comparative genomics methods preceding Arnedo et al.'s model and Barman et al.'s predominantly favour precision significantly over recall. It is also worth noting that Barman et al.'s results have shown considerable potential for operating across different species of bacteria: compared to an AUC of 0.937 on the development dataset, the SVM has shown an AUC of 0.901 and 0.926 when tested on *E. coli K-12* and *S. Typhi Ty2* respectively, even though the SVM has never been trained on either of them. [6]

Subsequently, Eppenhof et al. have shown that Barman et al.'s model exhibits significantly degraded performance when the negative instances used are real, randomly

selected non-sRNA subsequences of the genome (as opposed to shuffled genomic sequences, which was the approach used by Barman et al.'s tests). The AUPRC score of Barman et al.'s model dropped from 0.811 on SLT2 using shuffled genomic sequences as negative instances, to 0.182 on the same dataset using real subsequences of the genome. Eppenhof et al. have instead proposed a Random Forest (RF) that makes use of the RNA's genomic context as a feature space. This model has shown an AUPRC score of 0.656 while using non-sRNA subsequences of the genome as negative instances. Over their development dataset, which consists of different species of bacteria, the RF has shown an AUPRC of ~0.43, which seemed to indicate a noticeable degree of feature conservation across taxa of bacteria. [7]

Seeing the results from the aforementioned scholarly articles, specifically Vockenhuber et al.'s and Barman et al.'s, it seems that feature conservation in methods of comparative genomics is indeed very limited across species, but n-spectrum-profile (k-mer) features seem to be a lot more promising in this aspect than their comparative genomics counterparts. Hence, we have hypothesized that the n-spectrum-profile is worth investigation alongside the genomic context as a feature space.

Tang et al. have proposed a couple of ensemble ML models for the task of sRNA classification: Weighted Average Ensemble Method (WAEM) and Neural Network Ensemble Method (NNEM). They have used a wide range of sequence-derived features, such as the *n-spectrum-profile* (frequencies of nucleotide subsequences of size *n*), the *(m,n)-mismatch-profile* (equivalent to the n-spectrum-profile but allowing up to *m* mismatches in each subsequence), *k-RevcKmer* (same as the n-spectrum-profile, but with the reverse complement k-length adjacent subsequences removed before calculating occurrence frequencies) and *Pseudo nucleotide composition features* (equivalent to the n-spectrum-profile but also includes the physicochemical properties of the subsequences [27]). The WAEM is a weighted average over multiple RF models, with weights optimized using a genetic algorithm. The NNEM is a two-step neural network, with the first step representing a number of different neural networks each learning over features of the same type (for example, learning over different settings of *n* for the n-spectrum-profiles), and the second step learning over the outputs of the first layer. On the balanced dataset, the NNEM model produced the highest AUC of 0.958 compared to Barman et al.'s 0.938 on the same dataset. [8]

Unfortunately, Tang et al. have also used shuffled genomic sequences as negative instances, which considering how close the NNEM is in performance to Barman et al.'s SVM, raises concern as to whether the NNEM will suffer degraded performance when tested on real non-sRNA subsequences as negative instances. We were not able to find any open source repository for Tang et al.'s work, and thus were not able to ascertain the performance of the NNEM on Eppenhof et al.'s dataset.

3 Methods

3.1 Dataset

We have used the same development dataset as that used by Eppenhof et al.; the dataset consists of genuine sRNAs from *R. capsulatus*, *M. tuberculosis*, *S. pyogenes*, *S. enterica* and *E. coli* to populate the positive instances, while the negative instances consist of randomly selected subsequences of the genomes of the aforementioned species, that possess no experimental evidence for containing sRNAs. The negative instances have been selected to match the length and strand of the positive instances. [7]

It is worth noting that this dataset contains a variety of bacterial species, and thus the performance achieved by a classifier evaluated over this dataset is not only indicative of the reliability of sRNA prediction within a given taxon, but across different species as well. In total, the dataset consists of 348 positive instances and 3469 negative instances, yielding a positive:negative ratio of approximately 1:10. We explain how our chosen evaluation metrics are resilient against this class imbalance in section 3.2, and we further investigate the impact of class balancing methods (or lack thereof) on some of our models in section 3.3.3.

Subsequently, we evaluated our models on a multi-species dataset compiled by Lu et al. [23], and preprocessed for the inclusion of genomic context features by Eppenhof et al. [7]. We used this dataset as an additional source of benchmarking that has not been involved in the training/validation process.

3.2 Performance Metrics

For evaluating models, we utilized the Area under the Precision-Recall Curve (AUPRC) and the Average Precision score (AP) as evaluation metrics. To define these metrics and justify why they have been preferred for evaluation, we need to define other common metrics first. Let the number of True Positives, True Negatives, False Positives and False Negatives be denoted by *TP*, *TN*, *FP* and *FN* respectively, then we can define the following metrics:

Recall (Sensitivity, True Positive Rate [TPR]) [28]

Intuitively, recall is the percentage of all the actual positives that the model has also predicted to be positive instances. Recall is given by:

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

Precision (Positive Predictive Value[PPV]) [28]

Precision is the percentage of all the predicted positives that turned out to be actual positives. It is given by the following equation:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Specificity [28]

Specificity is similar to recall, but applied over the negative instances, i.e. it is the percentage of all the actual negatives that the model has also predicted to be negative instances.

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

Another metric that is very closely tied to specificity is the **False Positive Rate** [**FPR**], which is given by $FPR = 1 - Specificity$.

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

[28]

Something worth noting about accuracy is that it is very sensitive to the ratio of positive:negative instances in the dataset. For example, consider the Eppenhof et al. dataset at hand, which as we have outlined has a ratio of positive:negative = 1:10; on such a dataset, a naive model that simply predicts the negative class for any input it gets will exhibit an accuracy of $10/11 \approx 91\%$, and it is important to note that this becomes the baseline accuracy for the dataset.

Area under Receiver Operating Characteristic (ROC) Curve [**AUC**] [29]

The ROC curve plots the TPR (recall) on the y-axis and the FPR on the x-axis. The AUC represents the probability that a model will rank a randomly selected positive instance higher than a randomly selected negative instance.

———

We can now define AUPRC and AP, and explain why we find them preferable to the aforementioned metrics:

Area under the Precision-Recall Curve [**AUPRC**]

The Precision-Recall curve plots the Precision on the y-axis and the Recall on the x-axis. The reason why we have a range of values for the precisions and recalls is because such a curve is generated by changing the classification threshold operating over the

predicted positive class probability. [30] Increasing the threshold makes it harder for an observation to be predicted as a positive instance, thus reducing recall and, generally, increasing precision.

It has been shown that optimizing for AUC does not necessarily optimize the AUPRC [30], and that for datasets exhibiting significant class imbalance towards the negative class, the AUC can be a deceptive metric due to the influence of this bias on specificity [31] [30]. On the other hand, the AUPRC can be a reliable performance metric despite such a class imbalance, since it incorporates precision over the minority of positive instances [31]. To avoid the influence of class imbalance on our model-selection process and performance evaluation, we decided to include the AUPRC as one of our two main metrics.

Unlike in a ROC curve, where the objective is to get the curve as close as possible to the top-left corner of the graph, an ideal binary classifier is one that exhibits a Precision-Recall curve that is as close as possible to the top-right corner of the graph [30].

Average Precision Score [AP]

It has been shown that linearly interpolating a Precision-Recall curve should not be done, since it tends to yield an optimistic value for the actual area under the curve [30]. In the python package *scikit-learn*, calculating the area under the curve is generally done using the trapezoidal rule, which is a form of linear interpolation.

The AP score voids the need for interpolation by approximating the true AUPRC using the mean precision weighted by the increase in recall from the last sampled threshold:

$$AP = \sum_n (R_n - R_{n-1}) P_n \tag{5}$$

[32]

Given a sufficiently granular sampling of the classification thresholds for a classifier, the interpolated AUPRC can closely approximate the true AUPRC, and in the process we get to also have enough classification thresholds to generate an accurate Precision-Recall curve, which shows detailed information about the precision of the classifier across different magnitudes of recall. However, it is not always plausible to perform such a granular sampling, especially with computationally intensive classifiers (we have indeed found the AP to be a lot more consistent and sensible when used for evaluating our RNN compared to the interpolated AUPRC); hence, we opt for the AP score in such cases where we want an accurate approximation of the true AUPRC at a relatively low computational cost.

3.3 Models

In order to establish a performance baseline for our potential models, we first constructed a genomic context model that is very similar to Eppenhof et al.'s. We then constructed sequence-based models that aim to improve over the performance of sequence-based methods available in the literature, and finally we tested our hypothesis for the lack of covariance between those two sets of features, by investigating models that combine sequence-based features with genomic context ones.

3.3.1 Sequence-based Models

We define sequence-based models as those that operate purely over sequence-derived features. Examples of those features are the *n-spectrum-profile*, and other features that have been used by Tang et al. to train their ensemble models (Section 2.3).

Random Forest Models

According to Tang et al.'s results for determining feature importance, *n=4* is the optimal setting for the *n-spectrum-profile*[8] (we also refer to the *4-spectrum-profile* as tetranucleotide frequencies). Hence, we first developed a python module that can generate the *n-spectrum-profile* for a given *n* over a set of sequences, and used that to generate the *4-spectrum-profile* on the Eppenhof et al. dataset. We then constructed a Random Forest that uses the tetranucleotide frequencies to predict whether the observation is an sRNA or not.

We noticed that an issue with the use of the *4-spectrum-profile* as a feature space is the resulting dimensionality of it: given that each nucleotide can take one of 4 values (A,C,G,T), then for subsequences of length 4, there are $4^4 = 256$ possibilities (and hence features) to operate on. This results in an increase in difficulty when it comes to tuning the model, and can be quite prohibitive for extending the feature space with other potential features. It is also worth noting that it is unlikely for all tetranucleotides to be significant features for this classification task. Thus, we decided that it is going to be preferable to perform feature selection and distill down the size of the feature space, and use this space in the construction of some of our combined models.

To perform this process, we first set the number of tress in the RF (we will refer to the larger RF as RF_f) to a high value (400), and ran a few cross-validation (CV) runs at higher values of this parameter and compared it to this setting, to ensure that it is no longer a limiting factor to the model's performance. We then ranked the features in descending order of the RF's feature importance score. The implementation of *scikit-learn* for this score is based on *gini importance* (also known as *mean decrease impurity*), which measures the decrease in node impurity, weighted by the probability

of reaching each node, and averaged over the trees of the ensemble [33]. We then selected the best features, cut-off by their cumulative feature importance, and evaluated the performance of a smaller RF (RF_s) on the selected features at different selection threshold levels. The threshold levels have been increased from 10% to 90% of the total feature importance, at intervals of 10%. We then determined the optimal threshold that maximizes the AUPRC for the smaller RF, and used the selected set of features as the new feature space. We found that a selection threshold of 50% has maximized the AUPRC for the RF_s, which corresponded to a reduction of the size of the feature space from 256 to 83.

The aforementioned models have been parameterized as follows:

Model	Ensemble size	Class weighting
RF_f	400	Balanced
RF_s	40	Balanced

A balanced class weighting is one that weighs classes in a manner inversely proportional to their relative frequencies (i.e. a minority class is weighted higher than a majority class), in order to avoid issues that might result from class imbalance (for example, yielding an estimator that simply always predicts the majority class).

Recurrent Neural Network Models

Evidently, limiting the feature space to just the *4-spectrum-profile* could be voiding other sequence-based features that are not covariant with the *4-spectrum-profile*, and hence can contribute positively to the performance of an sRNA classifier. One approach is to extract more sequence-based features, in a similar manner to what Tang et al. have proposed. However, we realized that there is a considerable similarity between the classification task at hand, and common tasks in Natural Language Processing (NLP): for example, semantic analysis and text classification are very common tasks in this field which involve taking a sequence as an input and assigning a class to it [34].

Traditional methods of text classification rely on statistical features such as n-grams (which are analogous to the *n-spectrum-profiles* within our domain) and bag-of-words to construct their feature space; however, an inherent issue of these features is that they do not incorporate the context of each element of the sequence, and thus are unable to capture more high-level features such as semantics [34]. More recently, the utilization of other methods such as RNNs and word embeddings (continuous vector representations of the elements within a sequence that encode their relationships [35] [36]) which are able to incorporate contextual information, have seen a great rise in popularity within the field of NLP due to their significantly superior performance [34] [37] [38].

Realizing the developments in the field of NLP, and noticing the influence of RNNs

within the field of bioinformatics [11] [39] [40], we decided to test out the performance of RNNs as a sequence-based model for sRNA classification.

We first split the RNA sequences into mononucleotides, and right-pad them with some null identifier so that all input sequences are of the same length. We then generate the tetranucleotides from these sequences with a stride of 1 nucleotide. Subsequently, we construct a word-embedding using Word2Vec [35], and utilize it to initialize an embedding layer within the RNN. The RNN architecture (shown in Figure 1) incorporates an embedding layer (producing vectors of length 100), a recurrent layer of 150 Gated-Recurrent Unit (GRU) cells, a dense layer (implementing leaky ReLU as an activation followed by batch normalization), and a sigmoid output. We opted for GRU cells in our implementation instead of Long-Short Term Memory (LSTM) cells, since GRU cells are less computationally intensive and usually have comparable performance with that of LSTMs [41].

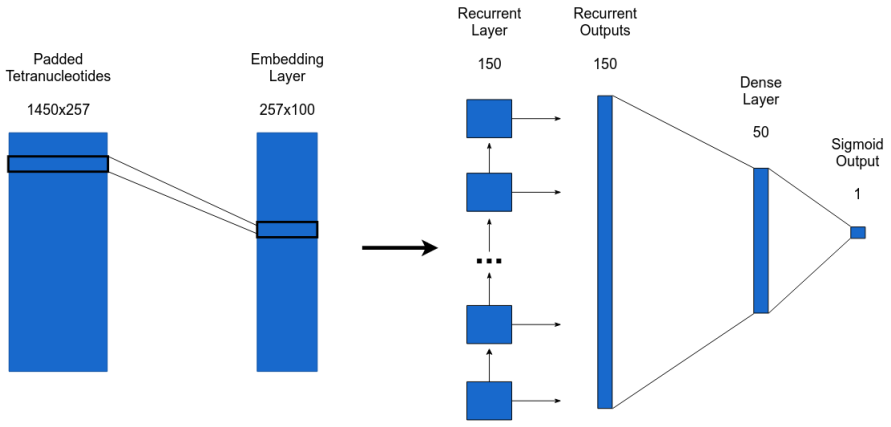

Figure 1: The architecture of the sRNA RNN. The padded tetranucleotides are of a fixed length of 1450, with each tetranucleotide taking on one of 257 values ($4^4 + 1$ with the additional 1 representing the null identifier). The tetranucleotides are mapped into their continuous vector representations through the embedding layer, which in turns feeds the vector representations to the recurrent layer. Each GRU cell outputs a value to generate a vector of length 150, that is subsequently fed into a dense layer to learn any interactions among the cell outputs, and finally a sigmoid activation is used to generate a classification probability.

3.3.2 Genomic Context Model

This model almost follows the same architecture proposed by Eppenhof et al.; the Genomic-Context Model is an RF that consists of 100 trees (as recommended by the original paper [7]), but does not set a max depth and uses class weight balancing.

We used Eppenhof et al.'s preprocessing package, sRNACharP [7], to get the genomic context features for the Eppenhof et al. dataset, and trained this model on the resulting feature space.

3.3.3 Combined Models

Genomic Context + *n-spectrum-profile*

For this model, we combined the genomic context features produced by the sRNACharP pipeline, and the selected tetranucleotide frequencies produced from our feature selection process discussed for the RF models in Section 3.3.1. We performed hyperparameter tuning over the size of the ensemble ([50, 400] at a step size of 50), the max depth of the decision trees ({None} $\cup$ [15, 25] at a step size of 1), and the method for class weighting ({None, balanced, balanced per tree}). We found no significant difference in performance to be induced by either the tree max depth or the class weighting method; thus we opted for the default on those two hyperparameters (no max depth and no class weight balancing). As for the size of the ensemble, we found no significant increase in performance for any size > 200 trees, and thus we opted for 200 to be the size of the ensemble.

Genomic Context + RNN

This model makes use of the tetranucleotide sequences as described in the RNN portion of Section 3.3.1 combined with the genomic context features generated by sRNACharP. First, the sequence-based features are selected from the feature space and provided to the RNN described in Section 3.3.1 for learning (during training) and prediction (during both training and inference). The positive class probabilities provided by the RNN are then concatenated with the genomic context features, and passed to an RF of the same hyperparameters as those tuned for the *Genomic Context + n-spectrum-profile* model. The output of the RF is the final output of the complete model.

We also experimented with exposing the layer before last (the outputs of the deep layer) in the RNN to the RF, as opposed to passing the sigmoid output, but we ended up observing a slightly smaller value for the AP score with this approach (~0.04 less), thus we kept the sigmoid output instead. We hypothesize that the decrease in performance was due to the fact that the outputs of the deep layer are tuned by weights that were in turn tuned by weights on the sigmoid layer, and the absence of those weights at prediction time would result in values that are less correlated with the target variable; the RF might have subsequently managed to recover a portion of that relationship, but the result was not quite as optimal as that produced by the full RNN.

Analyzing the feature importance scores reported by the RF shows that the predictions provided by the RNN account for 30% of the sum of importance scores (which always adds up to 1.0). That happens to be the highest individual feature importance among the feature space, at 1.7 times the value of the second highest feature importance score (which corresponds to the distance to the nearest right Open Reading Frame [ORF]). However, these observations also imply that, in aggregate, the genomic context features account for 70% of the feature importance scores.

3.4 Evaluation

For the RF_f, RF_s, *Genomic Context*, and the *Genomic Context + n-spectrum-profile* models, we have implemented repeated and stratified cross-validation (CV) at 10 folds and 5 repetitions. Stratification ensures that the relative class frequencies exhibited by the full dataset are also exhibited in the separate folds, in order to increase the consistency of results across the folds [42], and to incorporate any impact resulting from the class distribution seen in production. All these models have been evaluated based on their AUPRC at a very granular threshold sampling, which was confirmed to closely follow their AP score. A summary of these Random Forest models is given below:

Model	No. of Features	Ensemble Size	Class Weighting	Max Depth
RF_f	256	400	Balanced	No Limit
RF_s	83	40	Balanced	No Limit
Genomic Context	7	100	Balanced	No Limit
Genomic Context + n-spectrum-profile	90 (83+7)	200	None	No Limit

For the stand-alone RNN and the *Genomic Context + RNN* models we have relied on repeated and stratified 9:1 train:test splits, with 5 repetitions, and computed the AP score of the classifier after each run. We opted for this approach as opposed to K-fold CV due to the computational complexity of training the RNN. In each run, the Word2Vec model has been trained on the training data only, and then used to transform all the data into embedding indices and then further into vectors at the embedding layer. Both of the RNN models have been trained (per evaluation run) for 60 iterations at a mini-batch size of 100 observations. Between each evaluation run the Word2Vec, RNN and RF models have been reset and then retrained from scratch. We summarize these models below:

Model	Type	No. of Features	RF size
RNN	Word2Vec → RNN	1450 x 257 (Pre-embedding)	-
Genomic Context + RNN	Word2Vec → RNN → RF	1450 x 257 (Pre-embedding) + 7 x 1	200

4 Results and Discussion

4.1 Comparison of Sequence-based RF models

As discussed in Section 3.3.1, we constructed a complete RF that operates over all the tetranucleotide frequencies (RF_f) , and a smaller RF that was solely used for the purpose of feature selection (RF_s), which in turn allowed us to construct combined models that have a reasonably sized feature space. Figure 2 shows the observed distribution of the AUPRC for these two models over their respective feature spaces. The RF_f has a mean AUPRC of 0.43 with a standard deviation of 0.07, while the RF_s exhibits a mean of 0.41 with a standard deviation of 0.08.

Figure 2 and the AUPRC statistics above show that despite the RF_s feature space being only about 1/3rd the size of the full space, and the RF_s itself being a full order of magnitude smaller in size than the RF_f , the performance of the RF_s still manages to be quite comparable to that of the larger ensemble. Albeit showing a slight increase in spread and a minor decrease in average AUPRC, we still favoured to use the more selective feature space for generating the *Genomic Context + n-spectrum-profile* model. We hypothesized that the slight decrease in performance exhibited by the RF_s is caused by the smaller ensemble size rather than the feature space itself. Figure 3 further shows how closely the precision-recall curves for both models tread to each other.

4.2 Genomic Context + *n-spectrum-profile*

To highlight the performance improvement resulting from this model, we have compared the Precision-Recall curves of this model, the RF_f and the *Genomic Context* model as shown in Figure 4. The RF_f has yielded (as expected) a mean AUPRC of 0.435 with a standard deviation of 0.065, while the *Genomic Context* model has a mean AUPRC of 0.67 with a standard deviation of 0.065. The performance of the *Genomic Context* model can be seen to be quite representative of the proposed model by Eppenhof et al.; the *Genomic Context + n-spectrum-profile* model further improves over that performance, exhibiting a mean AUPRC of 0.725 with a standard deviation of 0.07. It is also worth noting that this model retains the highest precision at low recalls

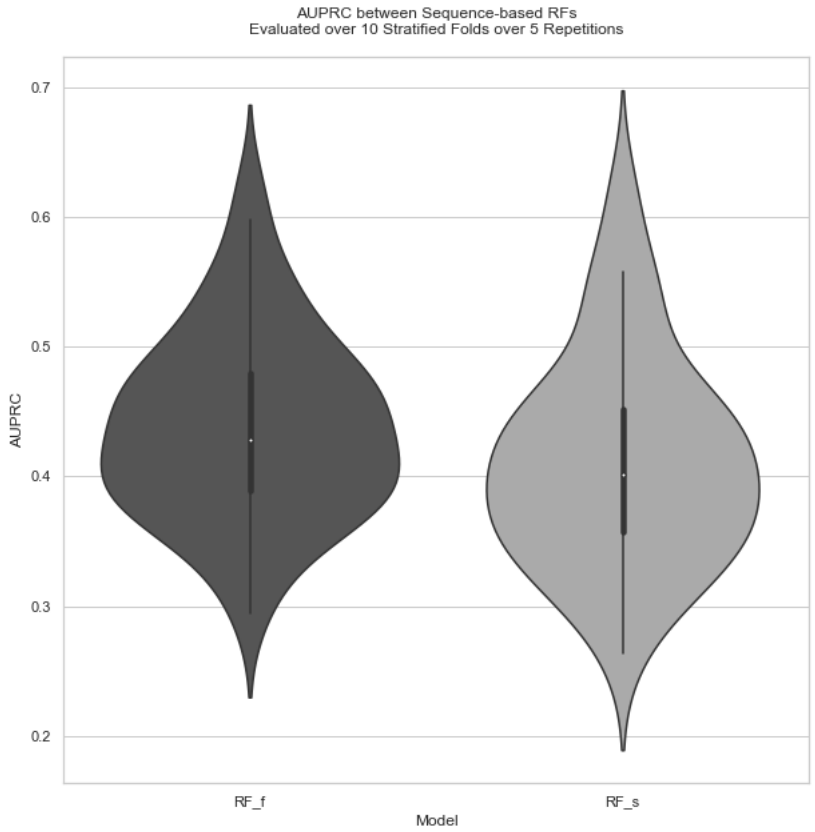

Figure 2: A violin plot showing the distribution of the observed AUPRC for the RF_f and the RF_s over their respective feature spaces.

(~98% precision at 10% recall). This verifies our hypothesis that there is a degree of lack of covariance between sequence-based and genomic context features, which can be used to construct highly performing models that operate over both sets of features.

4.3 RNN

The aim of testing out an RNN is to follow the same progression that NLP has went through when moving from statistical methods to Neural Networks, and examining whether such an improvement will have a similar impact on performance when it comes to sRNA classification. Indeed, we noticed a significant increase in mean AP score by ~0.06 over the RF_f. The table below summarizes the performance of the RNN

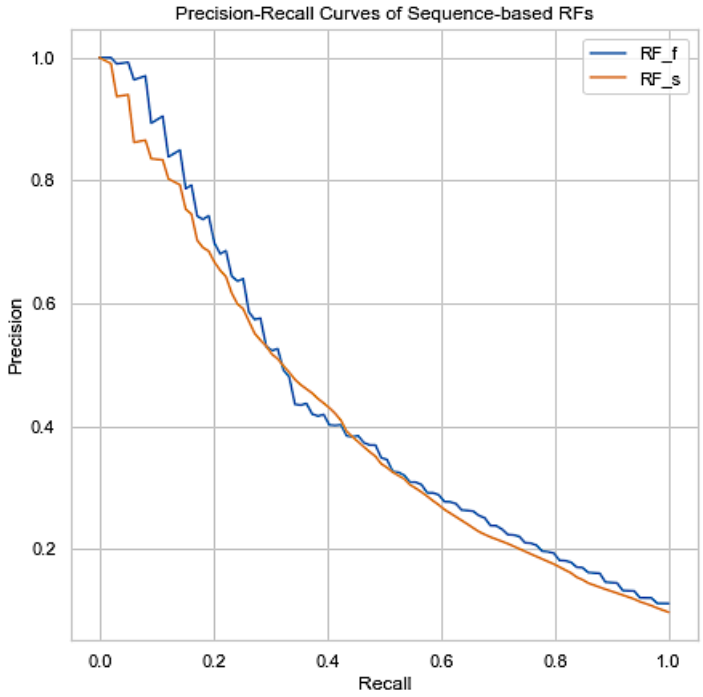

Figure 3: The Precision-Recall curves for the RF_f and RF_s

when compared to the RF_f :

Model	Mean AUPRC/AP	Std. Dev.
RNN	0.49	0.05
RF_f	0.43	0.07

4.4 Genomic Context + RNN

We were particularly interested in testing whether the *Genomic Context + RNN* is going to be significantly better than the *Genomic Context + n-spectrum-profile* model. Unfortunately, we only observed an improvement of ~0.025 in mean AP score, which is quite less than the improvement exhibited by the stand-alone RNN over the RF_f. We hypothesize that the reason for this behaviour is that the sequence-based features learnt by the RNN are in fact covariant with the features of the Genomic Context to a greater degree than that of tetranucleotide frequencies with the Genomic Context. While that is an exciting observation for the stand-alone RNN, it also implies that the

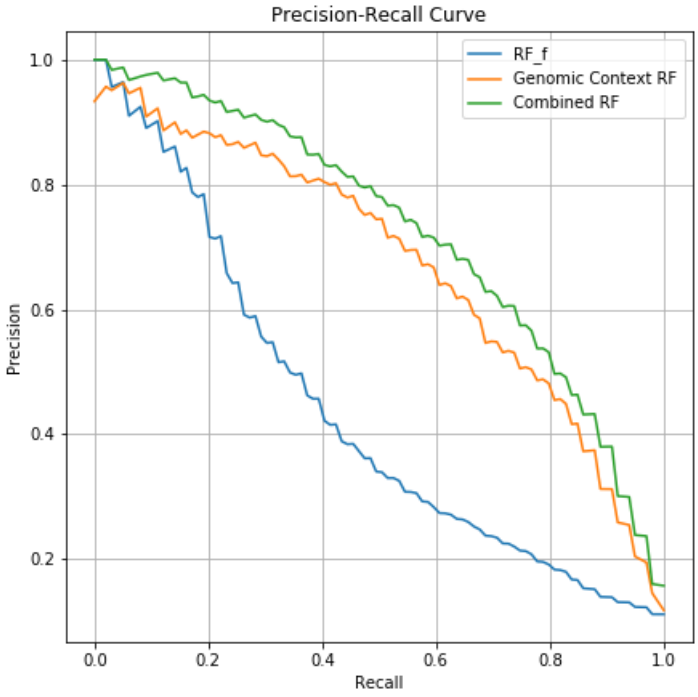

Figure 4: The Precision-Recall curves for the RF_f , the *Genomic Context* and the *Genomic Context + n-spectrum-profile* (combined) models.

performance of the *Genomic Context + RNN* model would not exhibit as much of an improvement as we would hope. The table below compares the performance of the *Genomic Context + RNN* with that of the *Genomic Context + n-spectrum-profile* model:

Model	Mean AUPRC/AP	Std. Dev.
Genomic Context + RNN	0.750	0.05
Genomic Context + n-spectrum-profile	0.725	0.07

4.5 Benchmarks on the Lu et al. Dataset

In order to verify our results, we investigated the performance of our models against Barman et al.'s SVM and the genomic context RF replica of Eppenhof et al., on the Lu et al. multi-species dataset as preprocessed by Eppenhof et al. for the inclusion of genomic context features [7]. In addition, we removed 4 observations that included the

"R" nucleotide within them, since our models do not support that form of nucleotide encoding.

As shown in Figure 5, the results obtained on the Lu et al. dataset ended up depicting the performance of our models in a different light; primarily, the RNN model ended up performing worse than both the RF_s and RF_f models, despite the fact that the RNN has exhibited superior performance to these models on the Eppenhof et al. dataset. In fact, we repeated the evaluation run for the RNN on the Lu et al. dataset (RNN_2) to verify our observations. This seems to be a significant indication of overfitting to the development set; such a discrepancy implies that the distributions of the Eppenhof et al. dataset and that of the Lu et al. dataset differ over the feature space which the RNN operates on. It is worth noting that the RNN model has good AP scores on the development train/test splits, with a standard deviation comparable to the remaining models, which leads us to believe that this form of overfitting is not due to lack of regularization.

Another noteworthy observation is the similarity between the AUPRC scores exhibited by all of the *Genomic Context + RNN* (RF_RNN), the *Genomic Context* and the *Genomic Context + n-spectrum-profile* models. Specifically, it seems like the sequence-based features added to either of the combined models do not add any significant improvement in performance over that of the standalone genomic context RF. This is especially interesting with respect to the *Genomic Context + n-spectrum-profile* model, since the RF_f performs quite well on the Lu et al. dataset, indicating an alignment of the distributions of both the development and benchmark datasets over the 4-spectrum-profile.

Undoubtedly, the results we have observed on the Lu et al. dataset warrant further investigation, but we still find the results observed on the Eppenhof et al. dataset to be promising, since it also is a multi-species dataset. Subsequent investigation of the roots of these differences in performance will help shed some light on the nature of these distributions, and on the form of features that the RNNs use to perform their classification of sRNAs.

4.6 Feature Importance

We inspected the feature importance scores given by the RF in the *Genomic Context + RNN* model (shown in Figure 6), in order to get an idea of the contribution of the RNN probabilities to the complete model, and to see how the genomic context features compare to the RNN probabilities and to each other. We found that the RNN outputs contribute to almost 30% of the Gini importance scores measured by the RF. Interestingly, we found a different ranking of genomic context features compared to

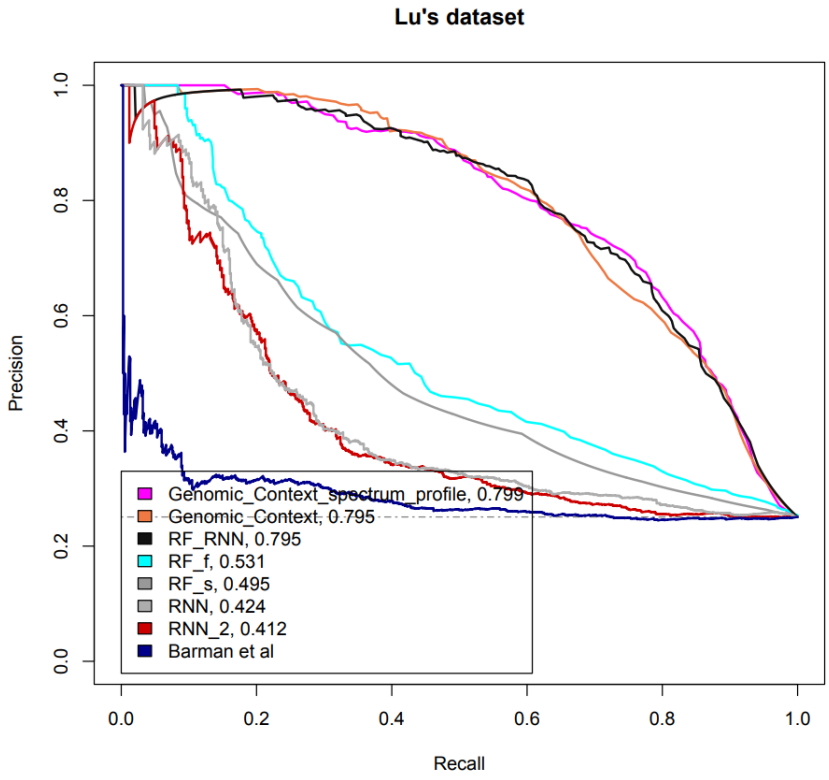

Figure 5: The performance of Barman et al.'s SVM, the Genomic Context RF replica of Eppenhof et al., and our investigated models as observed on the Lu et al. multi-species dataset

that reported by Eppenhof et al.; specifically, we found the free energy of the predicted secondary structure and the distance to the nearest -10 promoter site to be more important to the model than the distance to the closest Rho-independent terminator, while this case was reversed in the observations reported by Eppenhof et al. [7].

This observation might be due to the fact that Eppenhof et al. have evaluated their feature importance scores based on mean decrease in accuracy [7], while we are using the Gini importance scores. However, it is also possible that this observation might further explain the limited increase in AP exhibited by the *Genomic Context + RNN* over the *Genomic Context + n-spectrum-profile* model; we speculate that these observations indicate that the features learnt by the RNN are rather covariant with the distance to the nearest Rho-independent terminator, and hence render this distance less important in the model.

The remaining features of the model agree in importance with the observations reported by Eppenhof et al., nonetheless, with the orders of the Left/Right ORF distance

flipped, and the same goes to the Left/Right ORF same strand flags; the difference in importance between each left/right feature is minimal however, and thus we do not deem these order differences significant.

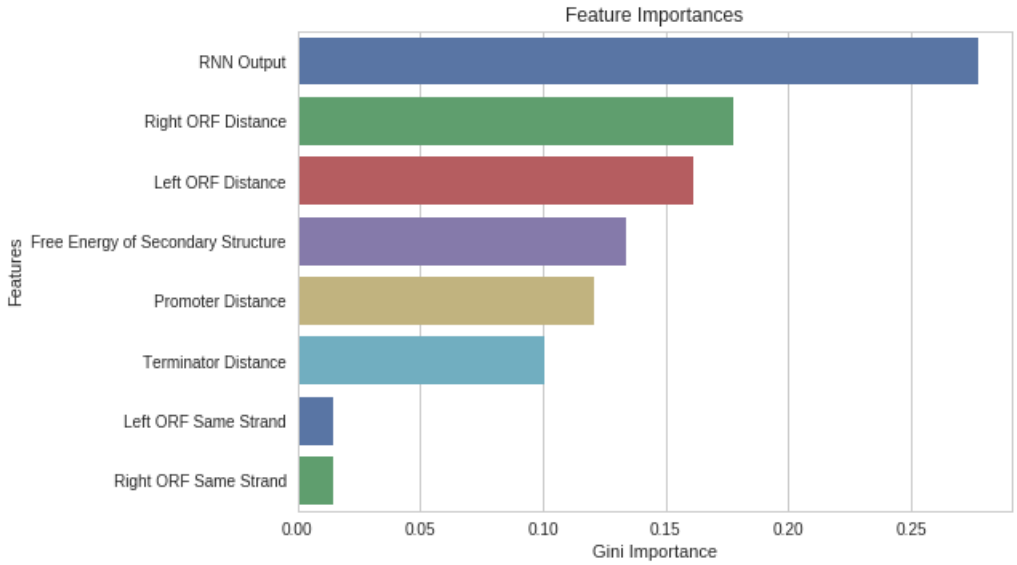

Figure 6: Gini importance scores determined by the RF in the *Genomic Context + RNN*

5 Conclusions

The problem of sRNA classification is one that has been tackled a lot in the literature over a number of years, yet it seems that the performance reported for the majority of previous computational methods turned out to be quite optimistic. The methods that did report a more conservative and accurate performance score indicate that there is a significant room for improvement left for researchers to investigate. We showed that datasets where there is a degree of lack of covariance between sequence-based and genomic context features do exist. We also showed that modern approaches in NLP can be ported over to the problem of sRNA classification in order to maximize the utility of sequence-based features.

We have also seen that despite enhancements to the sequence-based aspect of our model, the aggregate of genomic context features continued to dominate the feature importance scores. As pointed out by Eppenhof et al., the preprocessing packages used for the extraction of the genomic context features at hand do not have perfect recall; specifically, the package used for the detection of promoter sites has a recall rate

between 49% and 59%, and the package used for terminator detection does not detect *cis*-acting terminators [7]. Eppenhof et al. have thus suggested that improving the performance of these detectors would play a key role in enhancing the performance of their genomic context model, and we believe the same applies to our combined models, considering that the feature importance scores of promoter/terminator distances are lagging behind the remaining significant features.

6 Code Availability

Python packages containing the proposed models, preprocessing scripts and Jupyter/Colaboratory notebooks detailing our research results, can be found on our GitHub repository at *https://github.com/BioinformaticsLabAtMUN/sRNARankingRNN*. The published models implement the *scikit-learn* interface for generating predictions.

References

[1] T. Dutta and S. Srivastava, "Small RNA-mediated regulation in bacteria: a growing palette of diverse mechanisms," *Gene*, vol. 656, pp. 60–72, 2018.

[2] L. S. Waters and G. Storz, "Regulatory RNAs in bacteria," *Cell*, vol. 136, no. 4, pp. 615–628, 2009.

[3] M.-P. Vockenhuber, C. M. Sharma, M. G. Statt, D. Schmidt, Z. Xu, S. Dietrich, H. Liesegang, D. H. Mathews, and B. Suess, "Deep sequencing-based identification of small non-coding RNAs in *Streptomyces coelicolor*," *RNA biology*, vol. 8, no. 3, pp. 468–477, 2011.

[4] J. M. Liu, J. Livny, M. S. Lawrence, M. D. Kimball, M. K. Waldor, and A. Camilli, "Experimental discovery of sRNAs in *Vibrio cholerae* by direct cloning, 5s/tRNA depletion and parallel sequencing," *Nucleic acids research*, vol. 37, no. 6, pp. e46–e46, 2009.

[5] E. Rivas and S. R. Eddy, "Noncoding RNA gene detection using comparative sequence analysis," *BMC bioinformatics*, vol. 2, no. 1, p. 8, 2001.

[6] R. K. Barman, A. Mukhopadhyay, and S. Das, "An improved method for identification of small non-coding RNAs in bacteria using support vector machine," *Scientific reports*, vol. 7, p. 46070, 2017.

[7] E. J. Eppenhof and L. Peña-Castillo, "Prioritizing bona fide bacterial small RNAs with machine learning classifiers," *PeerJ*, vol. 7, p. e6304, 2019.

[8] G. Tang, J. Shi, W. Wu, X. Yue, and W. Zhang, "Sequence-based bacterial small RNAs prediction using ensemble learning strategies," *BMC bioinformatics*, vol. 19, no. 20, p. 503, 2018.

[9] A. Stolcke and J. Segal, "Precise n-gram probabilities from stochastic context-free grammars," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pp. 74–79, Association for Computational Linguistics, 1994.

[10] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[11] Z. Shen, W. Bao, and D.-S. Huang, "Recurrent neural network for predicting transcription factor binding sites," *Scientific reports*, vol. 8, no. 1, p. 15270, 2018.

[12] R. J. Klein, Z. Misulovin, and S. R. Eddy, "Noncoding RNA genes identified in

AT-rich hyperthermophiles," *Proceedings of the National Academy of Sciences*, vol. 99, no. 11, pp. 7542–7547, 2002.

[13] S. Washietl and I. L. Hofacker, "Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics," *Journal of molecular biology*, vol. 342, no. 1, pp. 19–30, 2004.

[14] A. V. Uzilov, J. M. Keegan, and D. H. Mathews, "Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change," *BMC bioinformatics*, vol. 7, no. 1, p. 173, 2006.

[15] R. J. Carter, I. Dubchak, and S. R. Holbrook, "A computational approach to identify genes for functional RNAs in genomic sequences," *Nucleic acids research*, vol. 29, no. 19, pp. 3928–3938, 2001.

[16] J. Arnedo, R. Romero-Zaliz, I. Zwir, and C. del Val, "A multiobjective method for robust identification of bacterial small non-coding RNAs," *Bioinformatics*, vol. 30, no. 20, pp. 2875–2882, 2014.

[17] X. Xia, "What is comparative genomics?," in *Comparative Genomics*, pp. 1–20, Springer, 2013.

[18] S. Washietl, I. L. Hofacker, and P. F. Stadler, "Fast and reliable prediction of non-coding RNAs," *Proceedings of the National Academy of Sciences*, vol. 102, no. 7, pp. 2454–2459, 2005.

[19] A. R. Gruber, S. Findeiß, S. Washietl, I. L. Hofacker, and P. F. Stadler, "RNAz 2.0: improved noncoding RNA detection," in *Biocomputing 2010*, pp. 69–79, World Scientific, 2010.

[20] J. Livny, A. Brencic, S. Lory, and M. K. Waldor, "Identification of 17 *Pseudomonas aeruginosa* sRNAs and prediction of sRNA-encoding genes in 10 diverse pathogens using the bioinformatic tool sRNAPredict2," *Nucleic acids research*, vol. 34, no. 12, pp. 3484–3493, 2006.

[21] J. Livny, H. Teonadi, M. Livny, and M. K. Waldor, "High-throughput, kingdom-wide prediction and annotation of bacterial non-coding RNAs," *PloS one*, vol. 3, no. 9, p. e3197, 2008.

[22] A. Marchais, M. Naville, C. Bohn, P. Bouloc, and D. Gautheret, "Single-pass classification of all noncoding sequences in a bacterial genome using phylogenetic profiles," *Genome research*, vol. 19, no. 6, pp. 1084–1092, 2009.

[23] X. Lu, H. Goodrich-Blair, and B. Tjaden, "Assessing computational tools for the discovery of small RNA genes in bacteria," *RNA*, vol. 17, no. 9, pp. 1635–1647, 2011.

[24] D. H. Mathews and D. H. Turner, "Dynalign: an algorithm for finding the secondary structure common to two RNA sequences," *Journal of molecular biology*, vol. 317, no. 2, pp. 191–203, 2002.

[25] A. Coventry, D. J. Kleitman, and B. Berger, "MSARI: multiple sequence alignments for statistical detection of RNA secondary structure," *Proceedings of the National Academy of Sciences*, vol. 101, no. 33, pp. 12102–12107, 2004.

[26] W. K. Dawson, K. Fujiwara, and G. Kawai, "Prediction of RNA pseudoknots using heuristic modeling with mapping and sequential folding," *PloS one*, vol. 2, no. 9, p. e905, 2007.

[27] B. Liu, F. Liu, L. Fang, X. Wang, and K.-C. Chou, "repDNA: a python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects," *Bioinformatics*, vol. 31, no. 8, pp. 1307–1309, 2014.

[28] D. L. Olson and D. Delen, *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[29] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[30] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, ACM, 2006.

[31] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, p. e0118432, 2015.

[32] M. Zhu, "Recall, precision and average precision," *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, vol. 2, p. 30, 2004.

[33] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[34] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[37] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1422–1432, 2015.

[38] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, pp. 649–657, 2015.

[39] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles," *Proteins: Structure, Function, and Bioinformatics*, vol. 47, no. 2, pp. 228–235, 2002.

[40] D. Quang and X. Xie, "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences," *Nucleic acids research*, vol. 44, no. 11, pp. e107–e107, 2016.

[41] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[42] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.