RNA BIOLOGY
2022, VOL. 19, NO. 1, 44–54
https://doi.org/10.1080/15476286.2021.2012058

Taylor & Francis
Taylor & Francis Group

RESEARCH PAPER

OPEN ACCESS

sRNARFTarget: a fast machine-learning-based approach for transcriptome-wide sRNA target prediction

Kratika Naskulwar[a] and Lourdes Peña-Castillo[a,b]

[a]Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada; [b]Department of Biology, Memorial University of Newfoundland, St. John's, Canada

ABSTRACT

Bacterial small regulatory RNAs (sRNAs) are key regulators of gene expression in many processes related to adaptive responses. A multitude of sRNAs have been identified in many bacterial species; however, their function has yet to be elucidated. A key step to understand sRNAs function is to identify the mRNAs these sRNAs bind to. There are several computational methods for sRNA target prediction, and the most accurate one is CopraRNA which is based on comparative-genomics. However, species-specific sRNAs are quite common and CopraRNA cannot be used for these sRNAs. The most commonly used transcriptome-wide sRNA target prediction method and second-most-accurate method is IntaRNA. However, IntaRNA can take hours to run on a bacterial transcriptome. Here we present sRNARFTarget, a machine-learning-based method for transcriptome-wide sRNA target prediction applicable to any sRNA. We comparatively assessed the performance of sRNARFTarget, CopraRNA and IntaRNA in three bacterial species. Our results show that sRNARFTarget outperforms IntaRNA in terms of accuracy, ranking of true interacting pairs, and running time. However, CopraRNA substantially outperforms the other two programsin terms of accuracy. Thus, we suggest using CopraRNA when homolog sequences of the sRNA are available, and sRNARFTarget for transcriptome-wide prediction or for species-specific sRNAs. sRNARFTarget is available at https://github.com/BioinformaticsLabAtMUN/sRNARFTarget.

ARTICLE HISTORY
Received 6 March 2021
Revised 17 June 2021
Accepted 22 November 2021

KEYWORDS
sRNAs; sRNA target prediction; sRNA interactome; bacteria gene regulation; machine learning; comparative assessment; CopraRNA; intarna; sRNARFTarget; bioinformatics

1 Introduction

sRNAs are bacterial small regulatory RNAs, usually less than 200 nucleotides in length, involved in several biological functions, such as virulence, metabolism, and environmental stress response [1]. It is generally accepted that most bacteria have hundreds of sRNAs that regulate mRNA expression [2]. Many sRNAs exert their functions when they interact with mRNAs, and these interacting mRNAs are called the targets of the sRNAs. To understand the function and the regulatory networks of sRNAs, we first need to identify their targets.

There are several bioinformatics methods for sRNA target prediction such as CopraRNA [3], SPOT [4], TargetRNA2 [5], sTarPicker [6], and IntaRNA [7,8]. CopraRNA, the most accurate method, requires sequence conservation of both sRNA and mRNA in at least four bacterial species, and must be run one sRNA at a time. The sequence conservation requirement makes CopraRNA unsuitable for species-specific sRNAs. The number of species-specific sRNAs per bacterium varies greatly, as studies have found between one-fifth to nearly four-fifths of detected sRNAs to be species-specific [9–11]. Out of the programs that are not comparative genomic-based, IntaRNA and sTarPicker have been shown to achieve the best results in terms of the area under the ROC curve (AUROC) [6,12]. IntaRNA is also the underlying algorithm of CopraRNA [3]. However, perform ing a transcriptome-wide sRNA target prediction on a bacterial transcriptome using IntaRNA might take several hours depending on the number of sRNAs and mRNAs investigated. Here we present sRNARFTarget, the first ML-based method that predicts the probability of interaction between an sRNA-mRNA pair. sRNARFTarget is generated using a random forest [13] trained on the trinucleotide frequency difference of sRNA-mRNA pairs. As sRNARFTarget bases its predictions on sequence alone, it can be applied to any sRNA-mRNA pair (i.e. does not require sequence conservation of either sRNA or mRNA). To train sRNARFTarget we collected known sRNA–mRNA interactions including those identified using RNA sequencing (RNA-seq) [14] approaches such as MAPS [15], GRIL-seq [16], CLASH [17] and RIL-seq [18]. This generated a data set of 745 known sRNA-mRNA interacting pairs from multiple bacteria.

We comparatively assessed the performance of sRNARFTarget, CopraRNA and IntaRNA in terms of AUROC, ranking of confirmed interacting pairs, and running time using data from three bacterial species (*Escherichia coli*, *Pasteurella multocida* and *Synechocystis* sp PCC 6803). Our results show that CopraRNA is the most accurate and sRNARFTarget is the fastest of the three programs. Specifically, sRNARFTarget is on average 100 times faster than IntaRNA with the same or higher accuracy.

CONTACT Lourdes Peña-Castillo lourdes@mun.ca Memorial University of Newfoundland, of St. John's, Newfoundland A1B 3X5, Canada
Supplemental data for this article can be accessed here.

2 Materials and methods

2.1 Data collection

By searching in NCBI Pubmed, we identified studies listing confirmed sRNA–mRNA interactions including those identified by RNA-seq-based methods (Table 1). We collected all experimentally confirmed or high-confidence sRNA-mRNA pairs listed in these studies and gathered roughly 2,400 sRNA-mRNA pairs from multiple bacteria. An overview of the criteria used to select these sRNA-mRNA pairs is provided in Table S1.

The sRNA-mRNA pairs listed in the literature are in a variety of formats providing either sRNA – mRNA names, sRNA – mRNA sequences, or sRNA – mRNA genomic locations. We used the sequences directly if they were provided (e.g. sTarBase3.0 [29]). For other datasets, we created a file containing Entrez genome accession number, sRNA name and target mRNA name per sRNA-mRNA pair.

Our first data preprocessing step was to remove any duplicate pairs. To get the sRNA and mRNA sequences, we wrote two Nextflow (version 0.32.0) [30] pipelines. The first pipeline finds whether the sRNAs and mRNAs names exist in the NCBI Gene database using the esearch function of Entrez direct [31] and generates a table containing sRNA-mRNA pairs found in the NCBI Gene database. Then, our second pipeline gets the sRNA/mRNA sequences using esearch from Entrez direct, and bedtools (version 2.27.1) [32]. We then divided the collected data into training data and validation data (Tables S2 and S3). 102, 22, and 20 sRNA-mRNA pairs from *Escherichia coli* [12], *Pasteurella multocida* [33] and *Synechocystis* [34,35] respectively, were held-out for benchmarking. The remaining data was used for training the models (Table 2).

At the end of this process we have 745 interacting pairs from 37 bacterial species for training, and 144 interacting pairs from three bacterial species for validation. All Nextflow pipelines and training/validation data used are available at https://github.com/BioinformaticsLabAtMUN/sRNARFTarget.

2.2 Machine learning model generation

We generated models for sRNA target prediction using three ML methods, namely, Random Forest (RF) [13], K-nearest neighbours (KNN) [36] and gradient boosting (GB) [37] using scikit-learn [38] functions to implement these classifiers.

2.2.1 Training data

We used *k*-mer frequency difference, and secondary structure distances as features to train the machine learning models. To calculate *k*-mer frequency difference, one first has to separately compute *k*-mer frequency for both sequences (sRNA and mRNA), and then calculate for every *k*-mer *i*, $f_{i,mRNA} - f_{i,sRNA}$ where $f_{i,s}$ is the frequency of *k*-mer *i* in sequence *s*. To obtain *k*-mer frequency and then *k*-mer frequency difference, we ran another Nextflow pipeline using scikit-bio (version 0.5.5) [39] in Python (version 3.7.4). We used *k* equal to 3 and 4, which corresponds to 64 and 256 *k*-mers, respectively. We obtained predicted secondary structures of sRNAs and mRNAs using CentroidFold (version 0.0.16) [40] with default values. Then we calculated seven distances between sRNA and mRNA secondary structures using RNAdistance (version 2.4.13) [41] programwith default values and indicating with the -D parameter the distance to calculate (F, H, W, C, h, w, or c).

After processing, our training data contained 745 interacting sRNA-mRNA pairs collected from the literature (Table S2). We created negative instances by randomly swapping the sRNAs in the sRNA-mRNA pairs. Basically, negative instances are sRNA-mRNA pairs where there is no experimental evidence for interaction. The use of non-annotated sRNA-mRNA pairs as negative instances gives a conservative estimate of the performance of the models (some predictions considered false positives might in fact be true positives). In total, we had 1490 sRNA-mRNA pairs (745 positives and 745 negatives) for training the ML models.

In sum, we have a balanced training data with 1,490 instances for a binary classification task, and explore four feature sets with (a) 64 (trinucleotide frequency difference), (b) 71 (trinucleotide frequency difference plus seven distances), (c) 256 (tetra-nucleotide frequency difference), and d) 261 (tetra-nucleotide frequency difference plus seven distances) attributes.

2.2.2 Model training

We used grid-search cross-validation (CV) of scikit-learn to get the best parameters per ML method. Table 3 shows the parameter ranges used in grid-search CV. We did 10-fold stratified CV to ensure balanced class distribution in each fold and used the area under the ROC curve (AUROC) to evaluate model performance. Additionally, we used R importance function [42] based on mean decrease in

Table 1. Studies from which we collected sRNA-mRNA interacting pairs.

Bacterium	Data source
Escherichia coli	[12,17–20]
Pseudomonas aeruginosa	[16,21,22]
Burkholderia cepacia	[23]
Pasteurella multocida	[33]
Salmonella	[24,25]
Mycobacterium tuberculosis	[26]
Synechocystis	[34,35]
Multiple bacteria	[27–29]

Table 2. Training and benchmarking data characteristics.

Data	No. of species	No. of sRNAs	No. of pairs
Training	37	176	745
Benchmarking	3	25	144

Table 3. Parameters per ML method used for grid-search CV.

Method	Parameter	Values
RF	n_estimators	[500, 600, 800, 1000]
	max_features	['sqrt', 'log2']
	max_depth	range(1, 11)
GB	n_estimators	[400, 500, 700, 1000]
	max_features	['log2','sqrt']
	max_depth	range(1, 11)
KNN	n_neighbors	range(1, 50)
	weights	['distance', 'uniform']

accuracy to get the feature importance, and filtered out any feature with a mean decrease in accuracy ≤ 0.

2.2.3 Model selection

We calculated sRNA-mRNA secondary structure distances to explore whether these features will increase AUROC and added them as features together with the trinucleotide frequency difference or the tetra-nucleotide frequency difference. Thus, we trained models using either trinucleotide frequency difference (64 features), tetra-nucleotide frequency difference (256 features), trinucleotide frequency difference plus seven distances, or tetra-nucleotide frequency difference plus seven distances. For each of the four sets of features, we found the optimal parameter setting per classifier using grid search CV and compared the models' performance in terms of 10-fold CV AUROC. We selected the model with the highest AUROC, and saved this model to be used by the Nextflow pipeline implementing sRNARFTarget.

2.3 sRNARFTarget nextflow pipeline

We wrote a Nextflow pipeline that uses our best model for sRNA target prediction. The pipeline takes sRNA and mRNA sequences in FASTA format as input, creates all possible sRNA-mRNA pairs, obtains the *k*-mer frequency for both sRNA and mRNA, and calculates the *k*-mer frequency difference by subtracting sRNA frequency from mRNA frequency using pandas (version 0.25.1) [43,44] subtract function. Then, the saved best model is loaded and predictions for all pairs are generated. The final result of the pipeline is a CSV file containing predicted probabilities of sRNA–mRNA interaction sorted in descending order (see Fig. S1 for workflow of sRNARFTarget program). Additionally, a file containing the features for all sRNA and mRNA pairs is also created. This file is used by the interpretability programs.

2.4 sRNARFTarget interpretability

We wrote two Python scripts using SHAP (version 0.35.0) [45] and pyCeterisParibus (version 0.5.2) [46] packages to facilitate the interpretation of predictions generated by sRNARFTarget (Fig. S2). Both scripts use the feature file generated by sRNARFTarget to get the features for the pair of interest. sRNARFTarget_SHAP uses TreeExplainer of SHAP to create an explainer. Then, it calculates the SHAP values for a given observation and generates SHAP's decision and force plots for interpretation. sRNARFTarget_CP creates the explainer using training data and calculates ceteris paribus profiles for a chosen feature for given sRNA-mRNA pair. It then generates a plot of the calculated profiles for the selected feature.

2.5 Benchmarking

Previous comparative assessments of sRNA target prediction programs [4,6,12] reported four programs (CopraRNA, IntaRNA, SPOT and sTarPicker) as the most accurate programs, with CopraRNA been the most accurate program. SPOT is reported to be comparable to CopraRNA; however, we were unable to run SPOT locally and running SPOT through Amazon Web Services (AWS) requires payment [47]. Additionally, sTarPicker program is no longer available. Therefore, we included CopraRNA and IntaRNA in our benchmark.

The data used for independent benchmarking have 22 sRNAs and 102 confirmed interacting sRNA-mRNA pairs for *E. coli* [12], one sRNA and 22 confirmed sRNA-mRNA pairs for *P. multocida* [33], and two sRNAs and 20 pairs for *Synechocystis* bacteria [34,35]. These data were not used for training. We extracted the sequences for 22 sRNAs of *E. coli* using our Nextflow pipeline as described above. For all other sRNAs, we fetched the sequence directly from the NCBI nucleotide database based on the locations provided in the corresponding manuscript. The location of *isar1* sRNA was taken as reported in [34]. The location of *psrR1* sRNA (1,671,919–1,672,052) was confirmed by electronic communication with the author of [35]. Finally, *gcvB* sRNA location was obtained from [33]. As we wanted to perform transcriptome-wide prediction of sRNA targets, we collected genomic location for all the mRNAs belonging to each bacterium directly from NCBI. We then obtained the sequences for these mRNAs using our Nextflow pipeline. In the case of CopraRNA, if predictions for a given *E. coli* sRNA were already available in CopraRNA web server, we used the available predictions. Otherwise, to find homologs for *E. coli* sRNAs, we used GLASSgo – sRNA Homolog Finder [48]. Additionally, we used the homologs provided in [34] and [35] for *isar1* and *psrR1* sRNAs of *Synechocystis*. For *gcvB* sRNA of *P. multocida*, we retrieved homolog sRNAs from NCBI. Note that all the sRNAs in the validation data are conserved among some bacterial species. We chose conserved sRNAs so that CopraRNA could be included in the comparative assessment.

We downloaded IntaRNA (version 3.1.0.2) source code from [49], installed it locally, and executed it with default values from the command line. To obtain a total execution time for IntaRNA, we created a Nextflow pipeline to run IntaRNA's two steps: 1) getting the interaction energy and 2) calculating the p-values for the interaction energy. We ran sRNARFTarget and IntaRNA from the Linux command line (system specifications are: one processor, processor speed 2.2 GHz, 4 cores and 16 GB RAM). CopraRNA (version 2.1.2) was run from its web server (http://rna.informatik.uni-freiburg.de/CopraRNA/Input.jsp, version 4.8.2).

After running the programs, we standardized their results by assigning corresponding classes to all predictions (1 to confirmed interacting sRNA-mRNA pairs and 0 to all other sRNA-mRNA pairs) and using predicted interaction probability for all programs. CopraRNA and IntaRNA output p-values where lower p-values indicate higher predicted likelihood of interaction. Thus, we subtracted CopraRNA and IntaRNA p-values from 1 to obtain predicted interaction probability. Additionally, for all three programs we rounded the predicted interaction probability to 5 decimals. To eliminate the duplicate entries from CopraRNA result, we wrote an R (version 3.5.1) script to get the most significant p-value (lowest p-value) for each sRNA-mRNA pair, and remove all other

Table 4. Final benchmarking dataset used for all three programs. The table lists the genome accession used, the number of sRNAs, the number of mRNAs, the number of confirmed interacting pairs (P), and the number of pairs considered non-interacting (N) per bacterial species (from top to bottom: *E. coli*, *Synechocystis* and *P. multocida*).

Accession	sRNAs	mRNAs	P	N
NC_000913.3	22	4,240	101	92,348
NC_000911.1	2	3,179	20	6,324
NC_002663.1	1	1,804	22	1,781

entries. We wrote an R script to get the pairs predicted by both IntaRNA and sRNARFTarget (Table 4). For those sRNA-mRNA pairs not predicted by CopraRNA (i.e. non-conserved pairs), we included them in CopraRNA evaluation with a predicted interaction probability of zero. By doing this, all three programs were evaluated on the same number of sRNA-mRNA pairs (Table 4). On average, CopraRNA did not make a prediction for 16.7% ± 0.025 of sRNA-mRNA pairs (Table S4).

3 Results and discussion

3.1 Selection of sRNARFTarget ML model

We adopted the idea of using sequence-derived features such as *k*-mer frequency from previous studies [50–53]. As sRNAs bind mRNAs through base pairing [54], we hypothesized that *k*-mer frequency difference might capture base pairing potential between mRNA and sRNA for the classifiers to use. Thus, we created feature sets using trinucleotide and tetra-nucleotide frequency difference. We started with trinucleotide composition, and as the performance decreased with tetra-nucleotide composition, we decided not to go beyond tetra-nucleotide composition.

Table 5 shows the performance in terms of AUROC of the best model per classifier when trained using trinucleotide frequency difference and tetra-nucleotide frequency difference. AUROC achieved with trinucleotide frequency difference was higher than the AUROC achieved with tetra-nucleotide frequency difference. With trinucleotide frequency difference, the model with the best performance in terms of AUROC was the RF one with 0.67, followed by GB with 0.66, and then KNN with 0.63.

RNA secondary structures are associated with the regulation of mRNA [55]. Previous studies [53,56] used secondary structure information for prediction of sRNA–mRNA interaction and non-coding RNAs. As the secondary structure of both sRNA and mRNA affects their binding [57], we decided to assess whether including secondary structure distances as features together with the tri(tetra)-nucleotide frequency difference improved performance in terms of AUROC. However, including predicted secondary structure distances to the feature set did not increase the models' performance. When including secondary structure distances as features in addition to trinucleotide frequency difference, the AUROC was unchanged for RF (AUROC 0.67), dropped by more than half for KNN (AUROC 0.27) and went slightly up for gradient boosting (AUROC 0.67). Similarly, adding secondary structure distance features with tetra-nucleotide frequency difference features had little to no effect on model performance (Table S5). As adding distance features did not substantially improve models' performance but dramatically increased the time required to extract the features from seconds to hours (due to the prediction of RNA secondary structure using CentroidFold), we decided against using the distance features in our final model.

Table 5. 10-fold CV AUROC for the best model per classifier trained on sequence-derived features (trinucleotide frequency difference and tetra-nucleotide frequency difference) of 1490 sRNA-mRNA pairs.

	AUROC (mean ± standard deviation)	
Models	Tri nt. difference	Tetra nt. difference
RF	0.67 ± 0.03	0.31 ± 0
GB	0.66 ± 0.03	0.32 ± 0
KNN	0.63 ± 0.03	0.45 ± 0.01

RF and GB models were comparable in terms of AUROC; however, the RF model was much faster to train than GB. Thus, we decided to train our final model on the 1490 sRNA-mRNA pairs using RF and included this model in the sRNARFTarget pipeline. The parameters to create this model are 500 trees (n_estimators), log2 of features for split (max_features), and a maximum depth of the trees of 9 (max_depth). From now on, we will refer to this final RF model as sRNARFTarget. Fig. S3 shows the 10-fold CV ROC curve of sRNARFTarget and Fig. S4 shows its top 30 most important features.

3.2 Interpreting sRNARFTarget predictions

To facilitate the interpretation of sRNARFTarget predictions, we have implemented two pipelines (sRNARFTarget_SHAP and sRNARFTarget_CP) to apply interpretability programs to sRNARFTarget predictions. To illustrate the functionality of these pipelines, we discuss interpretability plots generated for *isaR1-petF* confirmed interacting pair of *Synechocystis*. SHAP's decision plot shows how the model reached its decision and suggests that the value of feature GGC lowers the probability of interaction for this pair (Fig. S5). Force plot shows that features ACC and AAT are pushing sRNARFTarget to output higher interaction probability for this pair (Fig. S6). To gain insight on how a different value for the feature GGC impacts the output of sRNARFTarget for this pair, we looked at the ceteris paribus plot for feature GGC for *isaR1-petF* pair from *Synechocystis* (Fig. S7). It shows sRNARFTarget's prediction for different values of GGC when all other feature values remain constant. These plots can help pinpoint the sequence segments (trinucleotides) that contribute more to a specific sRNA–mRNA interaction.

3.3 Benchmark on independent data set

First, we assessed the performance of sRNARFTarget, CopraRNA and IntaRNA in terms of AUROC on data from three bacterial species: *E. coli* (gammaproteobacteria), *Synechocystis* (cyanobacteria) and, *P. multocida* (gammaproteobacteria). These data were not used for training. The *E. coli* 102 confirmed sRNA-mRNA pairs were the same used in the assessment performed by Pain et al [12]. We performed transcriptome-wide predictions; i.e. the methods have to infer interaction probability for all possible sRNA-mRNA pairs. Note that this is a conservative assessment as there might be true sRNA-mRNA interacting pairs that have not been confirmed yet and are considered false positives in the evaluation. Figs. 1, 2 and 3 show the ROC curve for *E. coli*, *Synechocystis* and *P. multocida*, respectively. Table 6 shows the AUROC for the three programs per bacterium. Across the three bacterial species, CopraRNA has the highest AUROC followed by sRNARFTarget and then IntaRNA. All programs show a decrease in AUROC on *P. multocida* data. As the data used is highly unbalanced (Table 4), we also obtained the Precision-Recall curves (PRC) (Figs. S8–S10 and Table S6). As it can be seen from the PRC curves and the AUPRC achieved, there is still room for improving the precision of computational transcriptome-wide sRNA target prediction. This result is similar to that obtained by Pain et al [12].

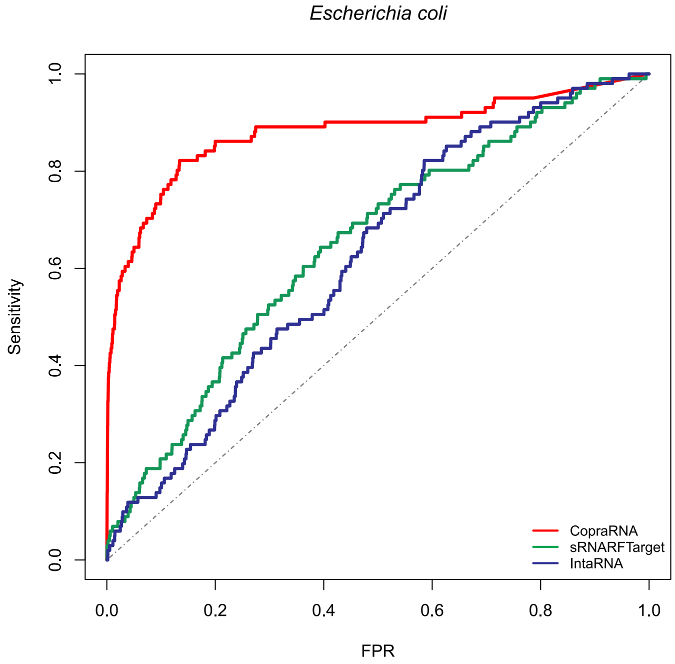

Figure 1. ROC curve for the three programs on *Escherichia coli* data. The plot shows the sensitivity (also called recall or true positive rate) as a function of the false-positive rate (FPR). The dash line indicates random classifier performance.

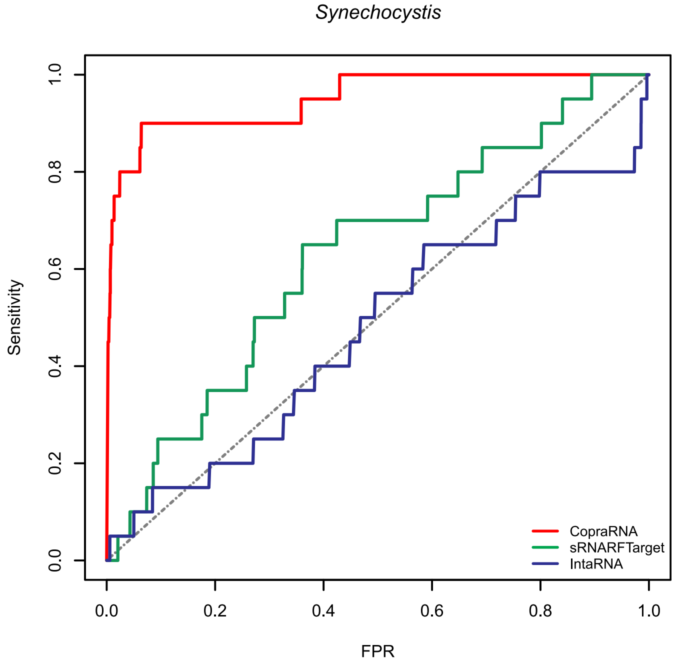

Figure 2. ROC curve for the three programs on *Synechocystis* data. The plot shows the sensitivity (also called recall or true positive rate) as a function of the false-positive rate (FPR). The dash line indicates random classifier performance.

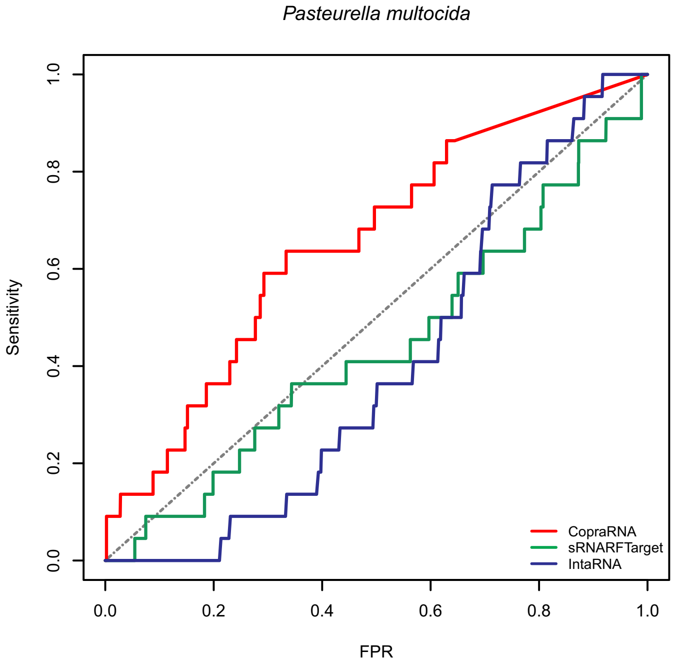

Figure 3. ROC curve for the three programs on *Pasteurella multocida* data. The plot shows the sensitivity (also called recall or true positive rate) as a function of the false-positive rate (FPR). The dash line indicates random classifier performance.

Table 6. AUROC obtained on each bacterial species included in the benchmark for all three programs assessed.

Bacterium	CopraRNA	sRNARFTarget	IntaRNA
E. coli	0.88	0.65	0.62
Synechocystis	0.95	0.63	0.48
P. multocida	0.65	0.44	0.40
Average ± sd	0.83 ± 0.16	0.57 ± 0.12	0.50 ± 0.11

Next, we looked at the rank distribution of confirmed interacting pairs per bacterium. Ideally, actual interacting pairs should have lower rank than non-interacting pairs, as a lower rank indicates that the program predicts with higher confidence that a given sRNA-mRNA pair is an actual interacting pair. To visualize program performance in terms of ranking of confirmed interacting pairs, we generated violin plots showing the rank distribution of confirmed interacting sRNA-mRNA pairs. The shape surrounding the box plots indicates the data density for different rank values. The horizontal bar inside the box shows the median rank of the confirmed interacting pairs. Fig. 4 shows the violin box plot for *E. coli*. CopraRNA has a lower median rank followed by sRNARFTarget and then IntaRNA. The shape of CopraRNA suggests that most of the confirmed interacting pairs are

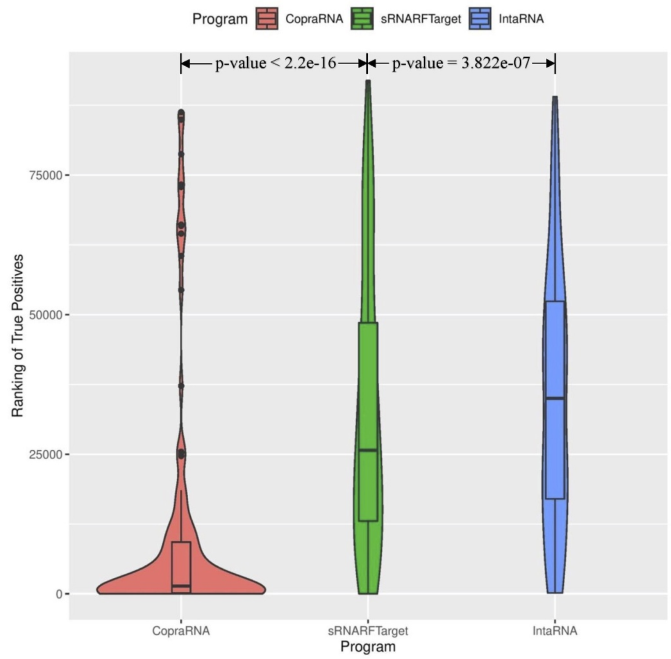

Figure 4. Rank (lower = better) distribution of 102 *Escherichia coli* confirmed interacting pairs. The violin plot for each program shows the data density for different rank values and the horizontal line inside each box indicates the median rank of confirmed interacting pairs.

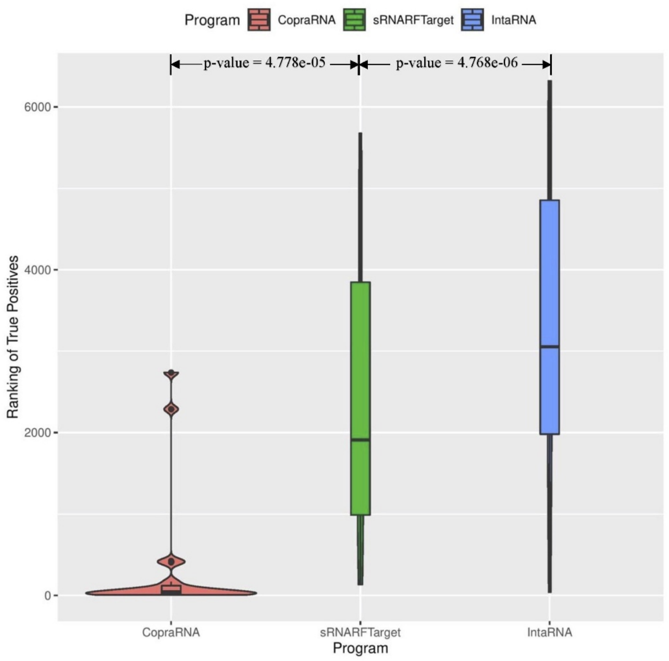

Figure 5. Rank (lower = better) distribution of 22 *Synechocystis* confirmed interacting pairs. The violin plot for each program shows the data density for different rank values and the horizontal line inside each box indicates the median rank of confirmed interacting pairs.

ranked before all other pairs. The shape of the plot for sRNARFTarget suggests that it has more confirmed interacting pairs with lower ranks than IntaRNA. We compared the rank distributions using the Mann-Whitney test (Fig. 4). The p-values obtained indicate that CopraRNA's median rank of interacting pairs is significantly lower than sRNARFTarget's median rank, and that sRNARFTarget's median rank is significantly lower than IntaRNA's median rank.

Figs. 5 and 6 show the violin plots for *Synechocystis* and *P. multocida*, respectively. For these two bacterial species as well, the median rank of confirmed interacting pairs is the lowest in CopraRNA's predictions, followed by sRNARFTarget and then IntaRNA. All three programs found it more difficult to distinguish true interacting pairs in *P. multocida* and ranked confirmed interacting pairs with higher ranks (Fig. 6) than for the other two bacteria. Nevertheless, CopraRNA still ranks confirmed interacting pairs significantly lower than sRNARFTarget (p-value = 2.15e-05), and sRNARFTarget ranks true interacting pairs lower than IntaRNA (p-value = 0.056).

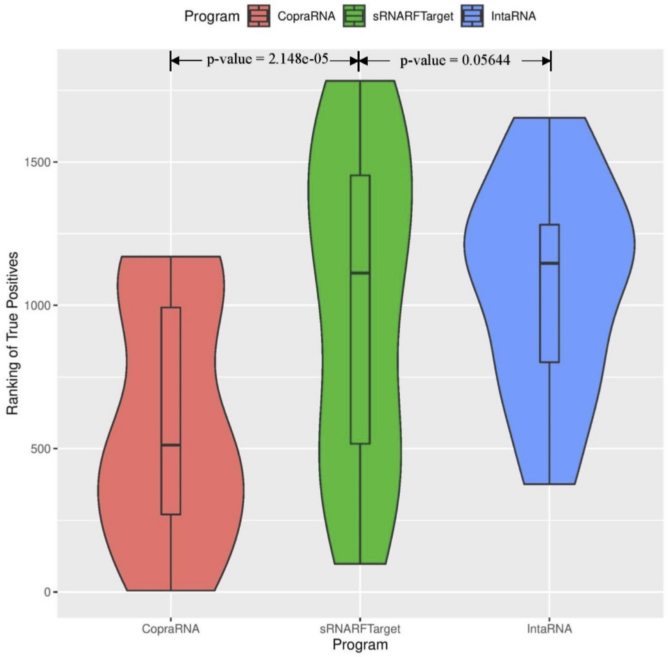

Figure 6. Rank (lower = better) distribution of 20 *Pasteurella multocida* confirmed interacting pairs. The violin plot for each program shows the data density for different rank values and the horizontal line inside each box indicates the median rank of confirmed interacting pairs.

We plotted the percentage of confirmed interacting sRNA-mRNA pairs predicted among a certain percentage of top predicted interacting pairs. To create these plots, we took the top 10% predictions for each program, counted the number of confirmed interacting pairs among these predictions, and calculated the percentage of true positives (recall) among the top 10% predictions. Then, iteratively increased the percentage of top predictions by 10% and repeated the process described above until all predictions (100%) were taken into account. We plotted the percentage of predictions on the x-axis and the percentage of confirmed interacting pairs (recall) on the y-axis. Fig. 7 shows this plot for *E. coli*. In the top 10% predictions, CopraRNA predicted 74% of confirmed interacting pairs, sRNARFTarget predicted 21% of these pairs, and IntaRNA predicted 14%. Among the top 50% predicted interacting pairs on *Synechocystis*, CopraRNA

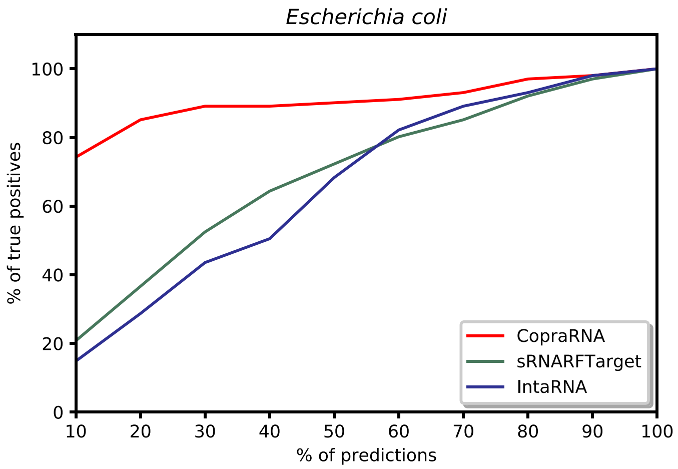

Figure 7. Percentage of *Escherichia coli* confirmed interacting sRNA-mRNA pairs (recall) as a function of percentage top predicted interacting pairs.

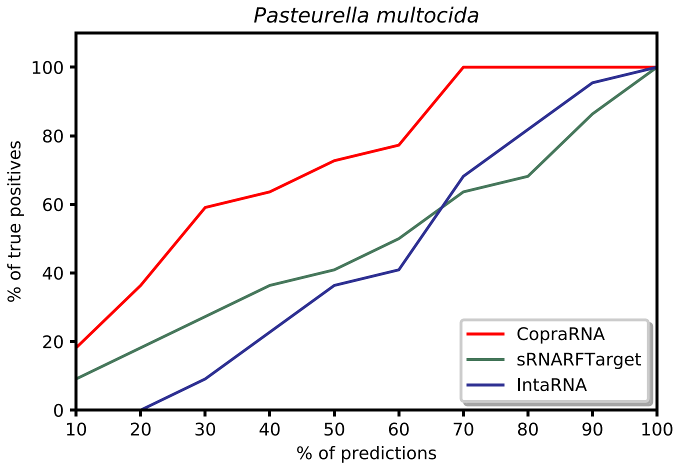

Figure 9. Percentage of *Pasteurella multocida* confirmed interacting sRNA-mRNA pairs (recall) as a function of percentage top predicted interacting pairs.

predicted 100% of the confirmed interacting pairs, sRNARFTarget predicted 70% of these pairs and IntaRNA predicted 55% (Fig. 8). In the top 20% predictions for *P. multocida*, CopraRNA predicted 18% of confirmed interacting pairs, sRNARFTarget was able to predict 10% of these pairs, and IntaRNA did not predict any confirmed interacting pair (Fig. 9). Thus, sRNARFTarget recovers more verified sRNA-mRNA interacting pairs than IntaRNA.

Finally, we looked at the amount of agreement among the three programs. To do that, for each bacterium, we took the top 10% predictions for each program and generated a Venn diagram (Fig. S11). There is low concordance among the three programs. On average only a quarter (or $24.77\% \pm 1.79$) of the top 10% predictions of each program are predicted by at least another program, and $13.86\% \pm 1.01$ of all the top 10% predictions per bacterium are supported by at least two of the programs.

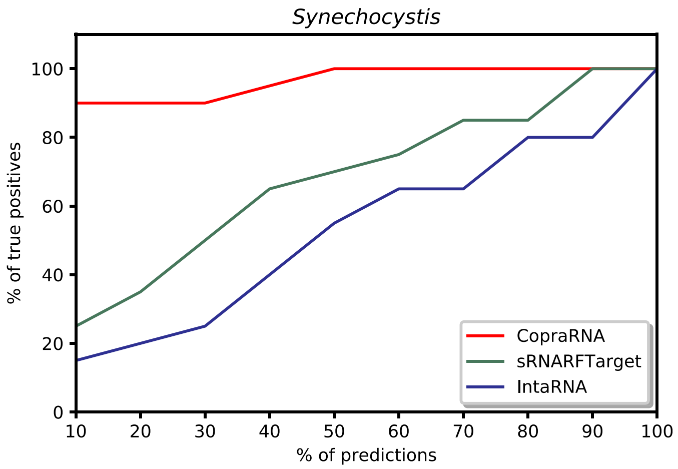

Figure 8. Percentage of *Synechocystis* confirmed interacting sRNA-mRNA pairs (recall) as a function of percentage top predicted interacting pairs.

3.4 sRNARFTarget's performance on IntaRNA 2.0's testing data [8]

We took the confirmed interacting sRNA-mRNA pairs provided by [8]. Out of 160 confirmed interacting pairs, we excluded those pairs present in our training data and used the remaining 119 interacting pairs (88 pairs of *E. coli* (NC 000913) together with 31 pairs of *Salmonella* (NC 003197)) to compare the performance of sRNARFTarget with that of IntaRNA. We ran sRNARFTarget and IntaRNA for 17 sRNAs and 4240 mRNAs of *E. coli* and, 7 sRNAs and 4450 mRNAs of *Salmonella*. The final number of pairs was 102,385 (71,427 pairs of *E. coli* and 30,958 pairs of *Salmonella*) for both programs.

Fig. 10 shows the ROC curve of sRNARFTarget and IntaRNA. AUROC of sRNARFTarget is 0.61, and IntaRNA is 0.59. sRNARFTarget's performance is comparable to that of IntaRNA in terms of AUROC. We plotted the ROC curves separately for *E. coli* and *Salmonella* for both programs to check the behaviour of the two bacteria independently. The performance for *E. coli* is comparable for both programs (AUROC 0.61 for sRNARFTarget and 0.62 for IntaRNA) (Fig. S12). For Salmonella, sRNARFTarget achieved an AUROC of 0.58 and IntaRNA achieved an AUROC of 0.51 (Fig. S13).

Fig. 11 shows the violin box plot for *E. coli* along with *Salmonella* for sRNARFTarget and IntaRNA. sRNARFTarget has a lower median rank compared to IntaRNA. P-value (Mann-Whitney test) indicates that the median rank of confirmed interacting pairs in sRNARFTarget is significantly lower than the median rank of IntaRNA.

3.5 Programs execution time

In terms of execution time, sRNARFTarget is faster than CopraRNA and IntaRNA (Tables 7 and 8). Table 8 shows the time taken by the CopraRNA web server for job completion on selected sRNAs (CopraRNA is run for one sRNA at a time). These times were calculated by taking the difference between the job submission time and the job

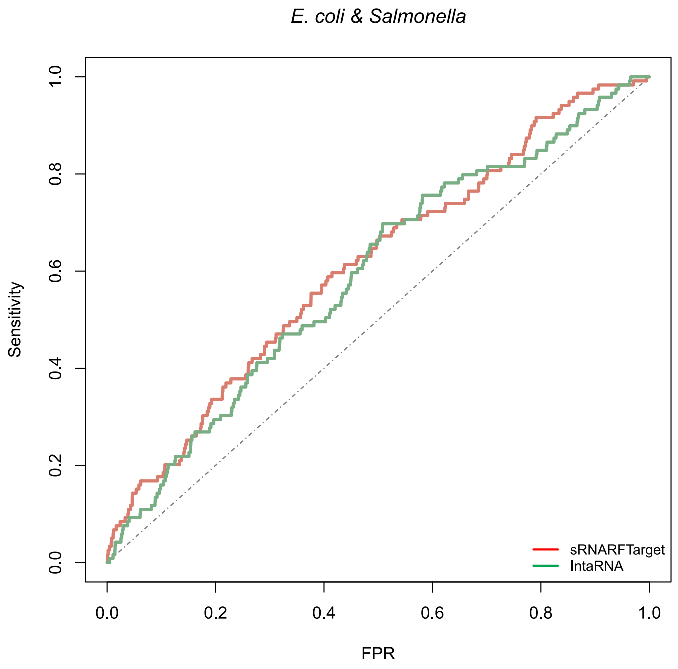

Figure 10. ROC curve for sRNARFTarget and IntaRNA on *E. coli* and *Salmonella* data. The plot shows the sensitivity (also called recall or true positive rate) as a function of the false-positive rate (FPR). The dash line indicates random classifier performance.

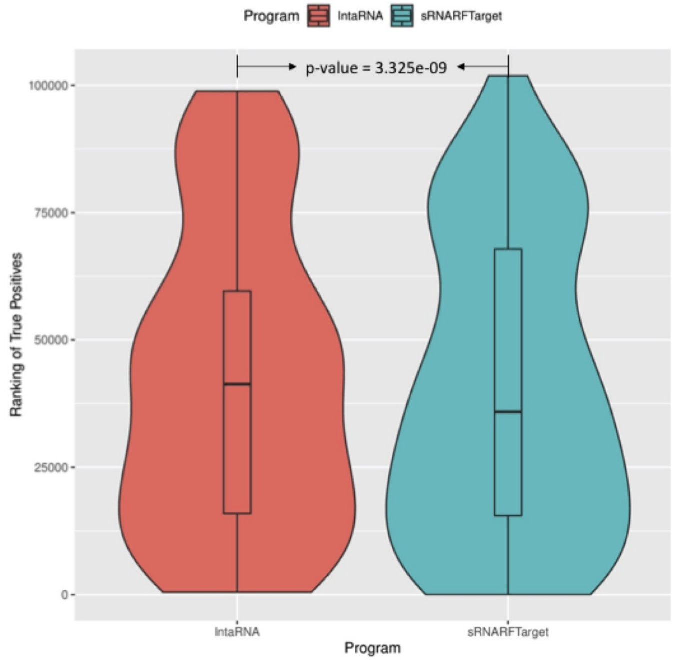

Figure 11. Rank (lower = better) distribution of 119 *E. coli* and *Salmonella* confirmed interacting pairs. The violin plot for each program shows the data density for different rank values and the horizontal line inside each box indicates the median rank of confirmed interacting pairs.

completion time (timestamp of job completion email). These times are not directly comparable to those shown in Table 7 as CopraRNA was run from the web server, and sRNARFTarget and IntaRNA were run from the Linux command line. sRNARFTarget execution time includes feature extraction (i.e. calculation of the trinucleotide frequency difference). To obtain interacting predictions for 1804 sRNA-mRNA pairs of *P. multocida*, sRNARFTarget took 31.4 seconds while IntaRNA took 6,196 seconds. To obtain interacting predictions for 93,280 sRNA-mRNA pairs (22 sRNAs and 4240 mRNAs) of *E. coli*, sRNARFTarget took 0.683% of the time taken by IntaRNA, which represents a 146-fold reduction in execution time (from more than 38 hours to 15 minutes). On average, sRNARFTarget is 100 times faster than IntaRNA with same or higher AUROC.

Table 7. Execution time for sRNARFTarget and IntaRNA on benchmarking data. Both programs were run on an Intel Core i7 (2.2 GHz) with 4 cores and 16 GB of RAM computer.

Bacterium	No. of sRNAs/ mRNAs	Execution time (HH:MM:SS)	
		sRNARFTarget	IntaRNA
P. multocida	1/1804	0:00:31	1:43:16
Synechocystis	2/3179	0:01:18	2:33:02
Salmonella	7/4450	0:05:47	6:18:16
E. coli	22/4240	0:15:56	38:52:43
Average	8/3418	0:05:53	12:21:49

Table 8. CopraRNA web server job execution time on selected sRNA for each bacterium on the benchmark data.

CopraRNA web server			
Bacteria	sRNA	No. of homologs	Execution time (HH:MM:SS)
E. coli	*arcZ*	8	08:00:00
P. multocida	*gcvB*	4	08:19:00
Synechocystis	*isrR1*	19	17:49:00

3.6 Predicting targets of sRNA RCd1 in *clostridioides (clostridium) difficile*

RCd1 is a *C. difficile* sRNA detected by RNA-seq and validated by Northern blot [58]. RCd1 is conserved only within *C. difficile* strains and bound by Hfq [59]. We used sRNARFTarget to predict RCd1 targets. To do this, we gave sRNARFTarget a FASTA file with RCd1 nucleotide sequence and a FASTA file with the mRNA sequences of *C. difficile* 630 (NC_009089.1) 3,902 mRNAs (downloaded from EnsemblBacteria release 51). FASTA files and sRNARFTarget predictions are available at https://github.com/BioinformaticsLabAtMUN/sRNARFTarget/tree/master/Data/CaseStudy.

sRNARFTarget predicted CD630_33600, a two component-response regulator, and spoVS (CD630_19350), stage V sporulation protein S, as the 6th and 19th most likely RCd1 targets, respectively. Boudry *et al.* suggested that RCd1 is involved in the control of late stages of sporulation in *C. difficile* [59]. Interestingly, in *C. difficile* sporulation is controlled by a two-component signal transduction system [60]; and, SpoVS is controlled by SigH and involved in later stages of sporulation [61].

Using DAVID Functional Annotation Tool [62], we looked at whether there was functional enrichment among sRNARFTarget top 390 (10%) predictions. There were 63 genes encoding hydrolases (FDR corrected p-value of 0.001). Hydrolases affect sporulation in *Streptomyces coelicolor* [63], another gram-positive spore-forming bacterium. Thus, one

Table 9. Sporulation-associated genes in sRNARFTarget top 10% predicted RCd1 targets. Smaller ranks indicate higher confidence of sRNARFTarget in the corresponding target prediction.

Gene	Symbol	Annotation	Rank
CD630_19350	spoVS	Stage V sporulation protein S	19
CD630_22460	cspC	Subtilisin-like serine germination related protease	71
CD630_36700		Putative cysteine desulfurase family protein	111
CD630_12140	spo0A	Stage 0 sporulation protein A	114
CD630_08560	oppD	ABC-type transport system, ATP-binding component	125
CD630_26700		ABC-type transport system, ATP-binding protein	138
CD630_15110	cotB	Spore coat protein	163
CD630_13860		Putative allophanate hydrolase subunit 1	187
CD630_24920		Two-component sensor histidine kinase, sporulation-associated spo0A	205
CD630_35690		Sporulation-specific protease	270
CD630_11950	spoIIIAD	Stage III sporulation protein AD	282
CD630_05980	cotCB	Spore-coat protein, manganese catalase	378

can speculate that some of these hydrolases might also be involved in sporulation in *C. difficile*. Additionally, we compiled a list of 78 *C. difficile* genes involved in sporulation from [61,64]. There was a slight enrichment of genes involved in sporulation among sRNARFTarget top 10% predictions (Table 9, hypergeometric p-value of 0.043), including: spo0A and CD630_24920 which encode a key regulator of sporulation and one of its associated kinases [61]. Thus, sRNARfTarget most-likely predicted RCd1/mRNA interactions are in agreement with the potential functional role of RCd1, and include potential targets worth further investigation.

4 Conclusion

There are many bacterial sRNAs without known mRNA targets. For example, for *E. coli* and *S. enterica*, two well-studied organisms, we found interactions for about 40 sRNAs for each of out of approximately 200 to 300 sRNAs expressed in these bacteria [2]. For other bacteria, the number of sRNAs without known interactions is much higher. For instance, roughly 400 putative sRNAs have been detected in *R. capsulatus* [11], but none of them has had its interactome characterized yet. Basically, the rate of sRNAs detections has outpaced the rate at which sRNAs' mRNAs targets are identified. In this study, we present a transcriptome-wide sRNA target prediction program, sRNARFTarget. We collected sRNA-mRNA pairs from the literature to create a training data set consisting of 745 confirmed interacting sRNA-mRNA pairs. As a comparison, RNAInter [65] contains 408 sRNA-mRNA interactions. We selected a Random Forest model as the final model for sRNARFTarget using the trinucleotide frequency difference between sRNA-mRNA as features.

In our benchmark, we compared sRNARFTarget with CopraRNA and IntaRNA. Our results show that the comparative genomics-based approach used by CopraRNA is the best performing approach in terms of AUROC. However, unlike CopraRNA, sRNARFTarget does not require an sRNA or mRNA sequence to be conserved among other bacteria and can generate predictions for any number of sRNA and mRNA sequences. We also show that sRNARFTarget is 100 times faster (Table 7) than the best non-comparative genomics program available, IntaRNA, with better accuracy (Table 6). Another advantage of sRNATarget is its simplicity to use, as sRNARFTarget does not require any parameter setting there is no risk to obtain a suboptimal result. On the other hand, IntaRNA has about a dozen parameters that need to be set and the setting of these parameters affects its performance [66].

As CopraRNA is the most accurate of the three programs, we suggest using CopraRNA when the homologs of the sRNA-mRNA sequences are available in at least four bacterial species. For transcriptome-wide prediction or when homolog sequences are not available, we recommend using sRNARFTarget.

Acknowledgments

This research was partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) to L. P.-C. (Grant number RGPIN: 2019-05247). K.N. was partially supported by funding from Memorial University School of Graduate Studies. This research was enabled in part by support provided by ACENET (www.ace-net.ca/) and Compute Canada (www.computecanada.ca).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) [2019-05247].

ORCID

Lourdes Peña-Castillo http://orcid.org/0000-0002-0643-2547

References

[1] Wagner EGH, Romby, P. Chapter three - small RNAs in bacteria and archaea: who they are, what they do, and how they do it. Advances in Genetics. Vol. 90. Academic Press, 2015; p. 133–208.

[2] Hör J, Gorski SA, Vogel J. Bacterial RNA biology on a genome scale. Mol Cell. 2018;70(5):785–799.

[3] Wright PR, Richter AS, Papenfort K, et al. Comparative genomics boosts target prediction for bacterial small RNA. Proc Nat Acad Sci. 2013;110(37):E3487–E3496.

[4] King AM, Vanderpool CK, Degnan PH. sRNA target prediction organizing tool (SPOT) integrates computational and experimental data to facilitate functional characterization of bacterial small RNAs. mSphere. 2019;4(1). DOI:10.1128/mSphere.00561-18

[5] Kery MB, Feldman M, Tjaden B, et al. TargetRNA2: identifying targets of small regulatory RNAs in bacteria. Nucleic Acids Res. 2014;42(W1):W124–W129.

[6] Ying X, Cao Y, Jiayao W, et al. sTarPicker: a method for efficient prediction of bacterial sRNA targets based on a two-step model for hybridization. PLOS ONE. 2011;6(7):1–12.

[7] Busch A, Richter AS, Backofen R. IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. Bioinformatics. 2008;24(24):2849–2856.

[8] Mann M, Wright PR, Backofen R. IntaRNA 2.0: enhanced and customizable prediction of RNA–RNA interactions. Nucleic Acids Res. 2017;45(W1):W435–W439.

[9] Broach WH, Weiss A, Shaw LN. Transcriptomic analysis of staphylococcal sRNAs: insights into species-specific adaption and the evolution of pathogenesis. Microb Genom. 2016;2(7):e000065.

[10] Gómez-Lozano M, Marvig RL, Molina-Santiago C, et al. Diversity of small RNAs expressed in pseudomonas species. Environ Microbiol Rep. 2015 Apr;7(2):227–236.

[11] Grüll MP, Peña-Castillo L, Mulligan ME, et al. Genome-wide identification and characterization of small RNAs in *Rhodobacter capsulatus* and identification of small RNAs affected by loss of the response regulator CtrA. RNA Biol. 2017;14 (7):914–925.

[12] Pain A, Ott A, Amine H, et al. An assessment of bacterial small RNA target prediction programs. RNA Biol. 2015;12(5):509–513.

[13] Breiman L. Random Forests. Mach Learn. 2001;45:1 5–32.

[14] Wang Z, Gerstein M, Snyder M, et al. RNA-Seq: a revolutionary tool for transcriptomics. Nature Reviews Genetics. 2009;10 (1):57–63.

[15] Lalaouna D, Carrier M-C, Semsey S, et al. A 3‘ external transcribed spacer in a tRNA transcript acts as a sponge for small RNAs to prevent transcriptional noise. Mol Cell. 2015;58 (3):393–405.

[16] Han K, Tjaden B, Lory S. GRIL-seq provides a method for identifying direct targets of bacterial small regulatory RNA by in vivo proximity ligation. Nat Microbiol. 2016;2(3):16239.

[17] Waters SA, McAteer SP, Kudla G, et al. Small RNA interactome of pathogenic *E. coli* revealed through crosslinking of RNase E. EMBO J. 2017;36(3):374–387.

[18] Melamed S, Peer A, Raya Faigenbaum-Romm YE, et al. Global mapping of small RNA-target interactions in bacteria. Mol Cell. 2016;63(5):884–897.

[19] Lalaouna D, Morissette A, Carrier M-C, et al. DsrA regulatory RNA represses both hns and rbsD mRNAs through distinct mechanisms in *Escherichia coli*. Mol Microbiol. 2015;98(2):357–369.

[20] Mihailovic MK, Vazquez-Anderson J, Li Y, et al. High-throughput in vivo mapping of RNA accessible interfaces to identify functional sRNA binding sites. Nat Commun. 2018;9(1):4084.

[21] Zhang YF, Han KS, Chandler CE, et al. Probing the sRNA regulatory landscape of *P. aeruginosa*: post-transcriptional control of determinants of pathogenicity and antibiotic susceptibility. Mol Microbiol. 2017;106(6):919–937.

[22] Pita T, Feliciano J, Leitão J. Small noncoding regulatory RNAs from *Pseudomonas aeruginosa* and *Burkholderia cepacia* complex. Int J Mol Sci. 2018 Nov;19(12):3759.

[23] Ramos CG, Da Costa PJP, Döring G, et al. The novel cis-encoded small RNA h2cR is a negative regulator of hfq2 in *Burkholderia cenocepacia*. PLoS One. 2012;7(10):e47896.

[24] Fröhlich KS, Haneke K, Papenfort K, et al. The target spectrum of SdsR small RNA in Salmonella. Nucleic Acids Res. 2016;44 (21):10406–10422.

[25] Ryan D, Mukherjee M, Suar M. The expanding targetome of small RNAs in. Salmonella Typhimurium Biochimie. 2017;137:69–77.

[26] Mai J, Rao C, Watt J, et al. *Mycobacterium tuberculosis* 6C sRNA binds multiple mRNA targets via C-rich loops independent of RNA chaperones. Nucleic Acids Res. 2019;47(8):4292–4307.

[27] Brantl S, Brückner R. Small regulatory RNAs from low-GC Gram-positive bacteria. RNA Biol. 2014;11(5):443–456.

[28] Lei L, Huang D, Cheung MK, et al. BSRD: a repository for bacterial small regulatory RNA. Nucleic Acids Res. 2013 Jan;41 (Database issue):D233–8.

[29] Wang J, Liu T, Zhao B, et al. sRNATarBase 3.0: an updated database for sRNA-target interactions in bacteria. Nucleic Acids Res. 2015;44(D1):D248–D253.

[30] Di Tommaso P, Chatzou M, Floden EW, et al. Nextflow enables reproducible computational workflows. Nat Biotechnol. 2017 Apr;35(4):316–319.

[31] Sayers E. Entrez programming utilities help [Internet]. 2008.

[32] Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26(6):841–842.

[33] Gulliver EL, Wright A, Lucas DD, et al. Determination of the small RNA GcvB regulon in the gram-negative bacterial pathogen *Pasteurella multocida* and identification of the GcvB seed binding region. RNA. 2018;24(5):704–720.

[34] Georg J, Kostova G, Vuorijoki L, et al. Acclimation of oxygenic photosynthesis to iron starvation is controlled by the sRNA IsaR1. Curr Biol. 2017;27(10):1425–1436.e7.

[35] Georg J, Dienst D, Schürgers N, et al. The small regulatory RNA SyR1/PsrR1 controls photosynthetic functions in cyanobacteria. Plant Cell. 2014;26(9):3661–3679.

[36] Mucherino A, Papajorgji PJ, Pardalos PM, et al. k-nearest neighbor classification. New York NY: Springer New York; 2009. p. 83–106.

[37] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001;29(5):1189–1232.

[38] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–2830.

[39] scikit-bio. [cited 2020 Jun 04]. Available from: http://scikit-bio.org

[40] Hamada M, Kiryu H, Sato K, et al. Prediction of RNA secondary structure using generalized centroid estimators. Bioinformatics. 2008;25(4):465–473.

[41] Lorenz R, Bernhart SH, Zu Siederdissen CH, et al. ViennaRNA Package 2.0. Algorithms Mol Biol. 2011;6(1):26.

[42] Importance function — R documentation. [cited 2020 Jun 04]. Available from: https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/importance

[43] The pandas development team. pandas-dev/pandas: Pandas; 2020.

[44] McKinney W. Data structures for statistical computing in Python. In: van der Walt S, and Millman J, editors. Proceedings of the 9th Python in science conference. 2010. p. 56–61. doi:10.25080/Majora-92bf1922-00a.

[45] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17; December 4-9; Long Beach (CA), 4768–4777. Curran Associates Inc.; 2017.

[46] Biecek P. GitHub - pbiecek/ceterisParibus: ceteris Paribus Plots (What-If plots) for explanations of a single observation. [cited 2020 Jun 25]. Available from: https://github.com/pbiecek/ceterisParibus

[47] phdegnan/SPOT: sRNA-target Prediction Organizing Tool. [cited 2020 Jun 04]. Available from: https://github.com/phdegnan/SPOT

[48] Lott SC, Schäfer RA, Mann M, et al. GLASSgo - Automated and reliable detection of sRNA homologs from a single input sequences. Front Genet. 2018;9:124.

[49] Backofenlab/intarna: efficient target prediction incorporating accessibility of interaction sites. [cited 2020 Jul 22]. Available from: https://github.com/BackofenLab/IntaRNA/#install.

[50] Yanzhen X, Zhao X, Liu S, et al. LncPred-IEL: a long non-coding RNA prediction method using iterative ensemble learning. *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*; San Diego (CA); 2019.

[51] Aimin L, Zhang J, Zhou Z. PLEK: a tool for predicting long non-coding RNAs and messenger RNAs based on an improved *k*-mer scheme. BMC Bioinformatics. 2014;15(1):311.

[52] Phan D, Nguyen NG, Lumbanraja FR, et al. Combined use of *k*-mer numerical features and position-specific categorical features in fixed-length DNA sequence classification. J Biomed Sci Eng. 2017;10(8):390–401.

[53] Chang T-H, Li-Ching W, Lin J-H, et al. Prediction of small non-coding RNA in bacterial genomes using support vector machines. Expert Syst Appl. 2010;37(8):5549–5557.

[54] Storz G, Vogel J, Wassarman KM. Regulation by small RNAs in Bacteria: expanding frontiers. Mol Cell. 2011;43(6):880–891.

[55] Fricke M, Gerst R, Ibrahim B, et al. Global importance of RNA secondary structures in protein-coding sequences. Bioinformatics. 2018;35(4):579–583.

[56] Sansen J, Thebault P, Dutour I, et al. Visualization of sRNA-mRNA interaction predictions. *2016 20th International Conference Information Visualisation (IV)*; Lisbon (Portugal); 2016.

[57] Wroblewska Z, Olejniczak M. Hfq assists small RNAs in binding to the coding sequence of ompD mRNA and in rearranging its structure. RNA. 2016;22(7):979–994.

[58] Soutourina OA, Monot M, Boudry P, et al. Genome-wide identification of regulatory RNAs in the human pathogen *Clostridium difficile*. PLoS Genet. 2013 May;9(5):e1003493.

[59] Boudry P, Piattelli E, Drouineau E, et al. Identification of RNAs bound by Hfq reveals widespread RNA partners and a sporulation regulator in the human pathogen *Clostridioides difficile*. RNA Biol. 2021;18(11) :1931–1952. doi:10.1080/15476286.2021.1882180.

[60] Underwood S, Guan S, Vijayasubhash V, et al. Characterization of the sporulation initiation pathway of *Clostridium difficile* and its role in toxin production. J Bacteriol. 2009 Dec;191 (23):7296–7305.

[61] Saujet L, Monot M, Dupuy B, et al. The key sigma factor of transition phase, SigH, controls sporulation, metabolism, and virulence factor expression in *Clostridium difficile*. J Bacteriol. 2011 Jul;193(13):3186–3196.

[62] Huang DW, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat Protoc. 2009;4(1):44–57.

[63] Haiser HJ, Yousef MR, Elliot MA. Cell wall hydrolases affect germination, vegetative growth, and sporulation in *Streptomyces coelicolor*. J Bacteriol. 2009;191(21):6501–6512.

[64] Paredes-Sabja D, Shen A, Sorg JA. *Clostridium difficile* spore biology: sporulation, germination, and spore structural proteins. Trends Microbiol. 2014 Jul;22(7):406–416.

[65] Lin Y, Liu T, Cui T, et al. RNAInter in 2020: RNA interactome repository with increased coverage and annotation. Nucleic Acids Res. 2020;48(D1):D189–D197.

[66] Raden M, Müller T, Mautner S, et al. The impact of various seed, accessibility and interaction constraints on sRNA target prediction- a systematic assessment. BMC Bioinformatics. 2020;21(1):15.