A multi-level approach using deep learning and transfer learning for classifying non-coding RNAs

Mirele C. S. F. Costa[1] **, Celia G. Ralha**[1]**, Marcelo M. Brigido**[2] **,**
André C. P. L. F. Carvalho[3]**, Peter F. Stadler**[4] **, Maria Emilia M. T. Walter**[1]

[1]Departament of Computer Science - University of Brasilia
Brasilia, Brazil

[2]Institute o Biology - University of Brasília
Brasilia, Brazil

[3]Institute of Mathematics and Computer Sciences - University of São Paulo - São Carlos
São Carlos, Brazil

[4]Interdisciplinary Center for Bioinformatics - University of Leipzig
Leipzig, Germany

mirele.costa@aluno.unb.br,{ghedini,brigido,mariaemilia}@unb.br,
andre@icmc.usp.br, studla@bioinf.uni-leipzig.de

Abstract. *In this article, we propose a new approach to classify non-coding RNAs (ncRNAs) combining deep learning (DL) algorithms with transfer learning (TL) in a multi-level approach. DL was used to pre-train a model with data from seven ncRNA classes: CD-box, HACA-box, scaRNA, miRNA, tRNA, 5S rRNA, and 5.8S rRNA. As a case study, TL was used to classify riboswitches. The training and testing datasets were carefully chosen by searching for sequences in taxonomy-based species trees to maximize diversity. The proposed approach was compared with other methods found in the literature, and the results obtained outperformed others for small datasets. In addition, it can be used for other ncRNA classes.*

Resumo. *Neste artigo, apresentamos uma nova abordagem para classificar RNAs não-codificadores (ncRNAs), combinando deep learning (DL) com transfer learning (TL) em uma abordagem multinível. No pré-treinamento, DL foi usado com dados de sete classes de ncRNAs, CD-box, HACA-box, scaRNA, miRNA, tRNA, 5S rRNA e 5.8S rRNA. No estudo de caso, TL foi usado para classificar riboswitches. Os dados de treinamento e teste foram cuidadosamente escolhidos, buscando sequências em árvores de espécies para maximizar a diversidade taxonômica. Esta abordagem foi comparada com outros métodos da literatura e os nossos resultados foram melhores para conjuntos de dados pequenos. Além disso, pode ser aplicado a outras classes de ncRNAs.*

1. Introduction

Each family of non-coding RNAs (ncRNAs) performs specific cellular functions that usually cannot be inferred using only its bases (primary structure), as occurs in protein homology inference. In addition to sequence composition and length, the prediction of the

ncRNA family depends on its spatial structure (secondary structure). The structured RNA relevant for this study is not well conserved on long evolutionary timescales.

Many different approaches based on machine learning (ML) have been proposed to predict ncRNAs. Some of them can identify different classes of ncRNAs [Zhang, X. et al. 2022, Oliveira et al. 2016]. Regarding classification, it should be noted that some methods aim to distinguish ncRNAs from proteins [Ammunét, T. et al. 2022, Asim, M. N. et al. 2021, Liu, J. et al. 2006]. Some others have the objective of classifying different classes of ncRNAs [Chen, K. et al. 2023, Chantsalnyam, T. et al. 2021, Asim, M. N. et al. 2020, Fiannaca, A. et al. 2017]. A common assumption of these traditional ML methods is that the training and testing data belong to the same domain, resulting in a direct dependence on these data. A widely adopted ML method is deep learning (DL) [LeCun, Y. et al. 2015], characterized by the use of deep neural networks (DNNs) [Zhan, Z. et al. 2022]. DNNs comprise multiple layers of interconnected artificial neurons enabling them to learn complex patterns and representations from data, model intricate non-linear relationships, achieving high accuracy in real world tasks. Other non-traditional ML approaches, for example transfer learning (TL) [Weiss, K. et al. 2016, Torrey and Shavlik 2010], allow the application of pre-trained models in related domains, also enabling a reduction in the costs associated with obtaining and training large volumes of data. TL models have proven to be efficient solutions for various biological problems [Bansal, S. et al. 2024, Singh, J. et al. 2021], especially in cases where datasets present a low number of sequences, which affects the obtaining of robust ML methods.

Therefore, considering the complexity and diversity of ncRNA families, some of them presenting small number of sequences not sufficient to be used in ML methods, we believe that the use of models based on a multi-level approach, with DL for pre-training and TL for training, can be more effective for classifying specific ncRNA families. In this article, we present a multi-level method that combines a type of DL model designed to process graph-structured data, Graph Convolutional Network (GCN) [Kipf and Welling 2016] with TL to classify a special ncRNA class, riboswitches [McCown, P. J. et al. 2017]. For such, we used a TL to fine-tune a DL pre-trained model on various ncRNA classes to classify riboswitches.

Riboswitches can regulate genes involved in various biological processes, including the absorption of nutrients and the biosynthesis of essential molecules. They act as on-/off switches for the genes they regulate, modulating the production of proteins in response to the cell's metabolic needs. Although found in all three domains of life (Bacteria, Archaea, and Eukarya), they are particularly abundant and widespread in Bacteria, including many human pathogens, making them an attractive target for the development of antimicrobials [Stagno and Wang 2024]. Classifying riboswitches can be challenging, as they come in over 55 distinct classes and selectively sense small molecules or elemental ions [Olenginski, L. T. et al. 2024, Kavita and Breaker 2023]. Some riboswitches are complex and rival protein factors in their structural and functional sophistication [Breaker 2011]. Computational methods, such as deep learning frameworks, are being used to address this challenge [Premkumar, K. A. R. et al. 2020].

The method uses pre-trained models on seven specific ncRNA classes, which are microRNAs (miRNAs), three small nucleolar RNAs (snoRNAs) - CD-box, H/ACA-box

and scaRNA, ribosomal RNA (rRNA) including 5.8S-rRNA and 5S-rRNA, and transfer RNA (tRNA). For training and testing, we carefully constructed the datasets through a search of sequences in taxonomy-based species trees to maximize diversity. For this, we propose the Sequence Selection in Taxonomic Species Trees (SSTST) problem, designed to choose sequences present in many species taxonomic trees to maximize the number of distinct families to which these species belong.

This article is divided as follows. In Section 2, we devise the algorithm to ensure the diversity of taxonomic data. In Section 3, the multi-level approach method is detailed. In Section 4, the results obtained from a variety of experiments are discussed. Finally, in Section 5, we conclude and suggest future work.

2. Optimizing Taxonomic Data Diversity

In evolutionary biology, classification problems are almost always plagued by the phylogenetic dependence of the data. More precisely, all sequences within an RNA or protein family are homologous by common descent and thus carry patterns that are family-specific. Since the number of unrelated sequence families is usually very small, it is usually impossible to avoid phylogenetic dependencies since multiple members of each of the few families must be utilized for training ML models. To alleviate this problem, it is imperative to choose datasets that are as diverse and uniform as possible.

The problem of selecting data that maximize diversity given a phylogenic or taxonomic tree and/or measures of (dis)similarity has been intensively studied in the literature. [Leinster and Meckes 2016] present a mathematical approach to maximize taxonomic diversity (TD). Furthermore, taxonomic databases such as Taxallnomy [Sakamoto, T. et al. 2021] highlight the role of taxonomy not only as a classification system, but also as a framework to represent evolutionary scenarios and organism diversity, reinforcing the use of taxonomy-based species trees in computational approaches. [Pardi and Goldman 2005] propose strategies for selecting species in comparative genomics, focusing on how these choices influence the effectiveness of genome analysis. The authors argue that a heuristic approach to choosing a large number of species, even if not all of them are directly related, can result in a better discovery of information about genomic evolution and the functional diversity of genes. The study suggests that, rather than limiting the choices to very closely related species, expanding the range to include a greater diversity of organisms can increase the chances of revealing significant evolutionary patterns and insights into universal and specific gene functions.

In this work, we consider a particular variant of the class of TD problems that naturally arises when working with large, fine-grained taxonomies. In this setting, it is desirable to select taxa from the taxonomic tree alone, without further computation of a (dis)similarity measure. Instead, we strive to use the defining features of a taxonomy, namely the well-defined levels of classification.

The SSTST problem aims to select a set of sequences that maximize the diversity of families and, within families, the diversity of species. In practice, this problem considers two taxonomic levels, one that can be exhaustively covered, and the next lower one, from which a choice is to be retrieved. In our specific case, the task is to select *species* in such a way that *families* are represented as fairly as possible. However, when looking at

the Rfam families[1] not only members of the same Rfam family are correlated. There are also biases, e.g., in sequence composition, that introduce dependencies between members of different Rfam families in the same sequence.

When choosing sequences from such data, we should also avoid choosing data from the same and closely related species. The practical problem is further aggravated by the fact that for each Rfam family, data is available for different subsets of species. The input data are therefore a set $\{T_1, T_2, \ldots, T_n\}$ of species trees, each of which is a "subtree" of a common taxonomy T^* obtained by retaining only the taxa present in T_i. For practical calculations, we use the taxonomic species trees in phyloXML format extracted from the Rfam database [Ontiveros-Palacios, N. et al. 2024] since we also use Rfam as the source for our ncRNA sequence data. These trees have been derived from the NCBI taxonomy [Federhen 2012]. In contrast to phylo-genetic trees, in which most inner vertices are binary, taxonomic trees have large degrees and small depth. The depth is determined by a fixed set of hierarchical taxonomic levels (e.g. domain, kingdom, phylum, class, order, family, genus and species).

Problem (SSTST) *Given a set $T = \{T_1, T_2, \ldots, T_n\}$ of taxonomic species trees, find a set of sequences S of size M associated with the leaves of T_i and a fixed taxonomic level, such that (i) all members of a given taxonomic level are represented as evenly as possible, and (ii) the number of sequences taken from the same species is minimized.*

To solve SSTST, we propose a heuristic algorithm. The basic idea is to first choose a single species for each *family*, the taxonomic level of interest here. When moving on to trees $T_2, \ldots, T_n$, the families not used are considered first. When this set of *families* is exhausted, the choice begins with an iteration over all the *families*, but avoiding species that have already been chosen. In case the process exhausts the set of species, these are free again for further choice. The procedure is stopped if the desired total size M of the sequences is reached or if there are no more sequences to choose. For the algorithm, the number of sequences of all species in the trees must be greater than or equal to M. A more detailed description of the algorithm is presented in the pseudocode below.

The algorithm is executed in rounds, where in each round T_i, $i = 1, \ldots, n$, is searched. In the algorithm, $selected\_families$ is a data structure that stores the families selected from T. Species in T_i can have more than one sequence, and the same species can occur in more than one tree in T, so we created the data structure $available\_species$ to store the sequences of each species available to be chosen in T. To initialize $available\_species$, we first call the function $initialize\_available\_species$. The function iterates over $T_i \in T$, $i = 1, \ldots, n$, and, for each species in T_i, retrieves the corresponding sequences. After execution, the function returns $available\_species$ with all species not chosen in T and their corresponding sequences.

In the first round, T_1 is selected, and one sequence of one species from each family is added to S. In the corresponding species of $available\_species$, this sequence is removed, and family is inserted in $selected\_families$. Next, T_i, $i = 2, \ldots, n$, the algorithm searches for families that have not yet chosen species, ensuring that the same family is not selected again. If $|S| < MAX$, then the algorithm goes to the next rounds,

[1]Note that Rfam "families" of ncRNAs and the taxonomic level of "family" are independent concepts that must not be confused!

Algorithm: Heuristic algorithm for solving SSTST

Input:
$T = \{T_1, T_2, \ldots, T_n\}$: set of species trees
M: maximum number of sequences to be selected
Output:
S, $|S| = M$: set of sequences maximizing the number of families and species in T

$available\_species \leftarrow initialize\_available\_species(T)$
$S \leftarrow \emptyset$; $selected\_families \leftarrow \emptyset$; $round \leftarrow 1$
/* Round 1: select one species per family in T_1 */
foreach *family in* T_1 **do**
    find a not chosen *species* in $family$
    $S = S \cup \{sequence\}$; remove *sequence* from $available\_species$
    $selected\_families = selected\_families \cup \{family\}$
    if $|S| = M$ **then**
        exit
$tree\_index \leftarrow 2$
while $|S| < M$ **do**
    foreach T_i, $i = tree\_index$ *to* n **do**
        foreach $family$ *in* T_i *not in* $selected\_families$ **do**
            find a not chosen *species* in $family$
            $S = S \cup \{sequence\}$; remove *sequence* from $available\_species$
            $selected\_families = selected\_families \cup \{family\}$
            if $|S| = M$ **then**
                exit
    if *there are no species to be chosen in* $available\_species$ **then**
        make available for choice the *species* with at least one *sequence* to choose from
    $selected\_families \leftarrow \emptyset$; $tree\_index \leftarrow 1$; $round \leftarrow round + 1$

performing the same steps. When a new round starts, $selected\_families$ is reset, allowing all families to be re-examined, but ensuring that previously selected species are not chosen again. This procedure continues until $|S| = MAX$.

However, if all species in all trees have already been selected, but $|S| < MAX$, there are no more distinct species to be included. In this case, $available\_species$ is reset, allowing all species to be reconsidered. If some species in $available\_species$ still have sequences, which means that the list of sequences is not empty, these species are then reconsidered for selection. The algorithm proceeds, selecting different sequences from these species as previously described, which means that more than one sequence is chosen per species, until $|S| = MAX$ or there are no more sequences available. The output is S, which contains sequences of species selected by the algorithm, maximizing family diversity, and within each family, species diversity.

3. The multi-level approach method

In this section, we start by describing the datasets and then detail the method.

3.1. Data

The classes of ncRNA data were extracted from Rfam 15.0 [Ontiveros-Palacios, N. et al. 2024] to build positive datasets, while the NCBI

data were collected to build negative datasets, for the DL and TL models. In Figure 1, we show the pre-processing steps for constructing the datasets[2].

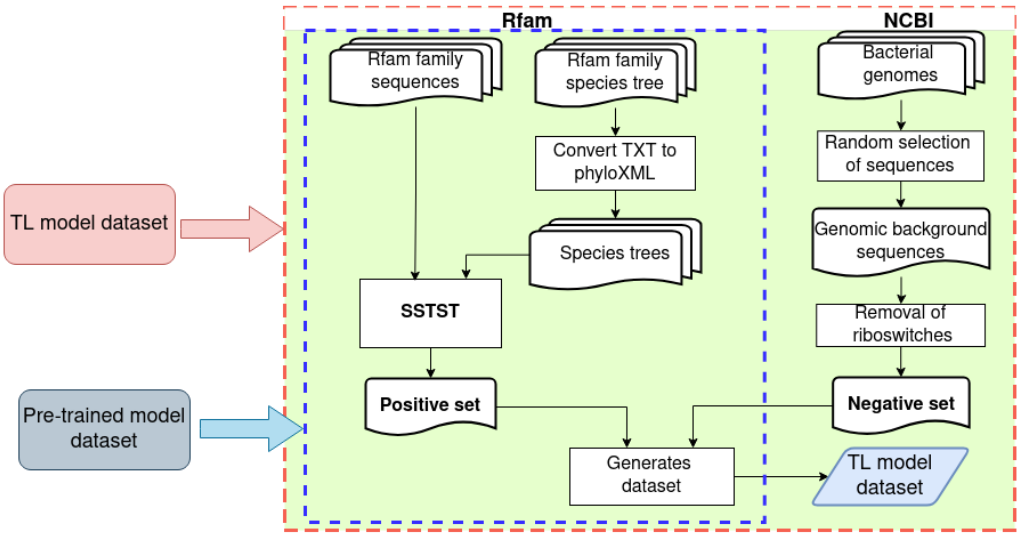

Figure 1. Data pre-processing steps for the construction of the datasets. The positive set was obtained from Rfam and the negative set from NCBI genomes.

For the positive set, we used the heuristic algorithm to select the sequences, to maximize diversity. Sequences from eight classes of ncRNAs were used to cover various families cataloged in Rfam 15.0 [Ontiveros-Palacios, N. et al. 2024]. Seven of these classes were used for constructing the pre-training datasets: miRNA, CD-box, H/ACA-box, scaRNA, 5.8S-rRNA, 5S-rRNA, and tRNA. For the positive dataset, we used riboswitches, the class of interest in our classification task. It was not included in the pre-training dataset. For each ncRNA class, the FASTA files for their corresponding Rfam families were selected, identified by unique codes (for example, RF01704). Each selected Rfam family contained at least 20 sequences. In addition, the corresponding species tree, available in TXT format, was also collected. The trees were converted from TXT to phyloXML format and then sorted according to their number of sequences, from the smallest to the largest. This ordering ensured that the set T used by the heuristic algorithm was structured to include the sequences of species that appear in smaller trees. This approach prevents families present in smaller trees from being preemptively selected by larger trees. Therefore, we used the FASTA files and previously sorted species trees in phyloXML format as input for the heuristic algorithm.

The negative set was obtained from the genomes of four bacterial species from the NCBI [Geer, L. Y. et al. 2009], since riboswitches are mainly spread in bacteria: *Escherichia coli*, *Staphylococcus aureus*, *Pseudomonas*, and *Mycobacterium tuberculosis*. From these genomes, we selected distinct random sequences, ranging in length from 100 to 300 nucleotides (riboswitch length), stored in FASTA format, referred to as "genomic background". In addition, considering the class of ncRNA of interest, we used the Infernal [Nawrocki, E. P. et al. 2009] tool to identify possible riboswitch sequences in the

[2]The datasets are available for public use at `https://github.com/Mirele7/multi-level-approach-DL-TL` our GitHub repository

FASTA files of positive and negative datasets. All detected riboswitches were removed and replaced with new randomly chosen sequences.

The TL model dataset comprises riboswitches as the ncRNA class of interest. A total of 20 Rfam families was collected. The positive and negative datasets were constructed with the same number of sequences. First, we used RNAFold, from the ViennaRNA package [Lorenz, R. et al. 2011], to obtain the secondary structure of the sequences. Therefore, we constructed datasets, each consisting of three types of information for each sequence: (i) a graph representing the secondary structure, (ii) the MFE (Minimum Free Energy) value, and (iii) the primary structure.

For the pre-trained model dataset, we performed the same steps described previously for each of the seven classes of ncRNAs: CD-box (20 Rfam families), H/ACA-box (20 Rfam families), miRNA (20 Rfam families), scaRNA (18 Rfam families), tRNA (2 Rfam families), 5.8S-rRNA (1 Rfam family) and 5S-rRNA (1 Rfam family). Given the large volume of sequences present in the tRNA 5.8S-rRNA and 5S-rRNA families, it was not possible to obtain the TXT files containing the species trees. To cope with this limitation, we adopted a random approach to selecting the sequences per species from the FASTA file, until MAX was reached. In this way, we ensured that different sequences from different species were chosen. Therefore, we obtained seven datasets, one for each ncRNA class. For these sequences, we used the RNAFold tool to obtain the secondary structure. As in the TL model, we obtained a dataset consisting of three types of information for each sequence: (i) a graph representing the secondary structure, (ii) the MFE value, and (iii) the primary structure.

3.2. Building the TL Model

To classify riboswitch sequences, our method combines a pre-trained GCN classifier with TL (see Figure 2).

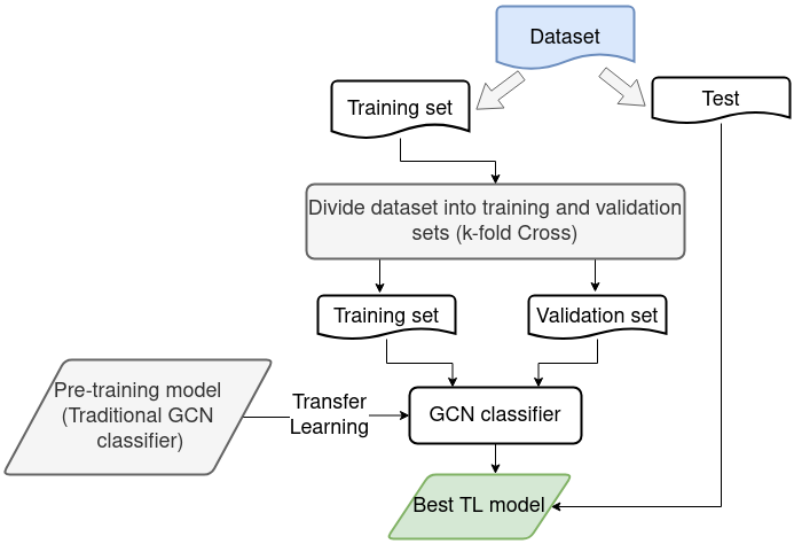

Figure 2. The multi-level approach method to classify riboswitches using DL and TL.

For the pre-trained model, we used the datasets of the seven ncRNA classes, as described in the previous section, and the architecture of the GCN classifier proposed

by [Rossi, E. et al. 2019], which was implemented using PyTorch and PyTorch Geometric libraries. This architecture was adapted to the k-fold cross-validation, $k = 5$, to ensure a robust and reliable evaluation of the model performance. The classifier architecture consisted of an embedding layer, followed by five convolutional layers of the NNConv type Message Passing Neural Network (MPNN), each executed with the BatchNorm batch normalization, LeakyReLU activation, and dropout application. After the convolutional stage, the graph representation was globally aggregated using the Set2Set pooling technique, whose output was then passed through a Fully Connected (FC) Layer with a log softmax activation function. The hyperparameters used were: embedding dimension 20, hidden dimension 80, batch size 64, training for up to 100 epochs with early stop (30).

The dataset from the TL model was divided into two subsets, training and testing, with a ratio of 70% and 30%, respectively. Therefore, to train the TL model, we used the GCN classifier, in which the weights previously learned by the pre-trained model were reused and refined for the new riboswitch classification task. The weights of the embedding layer, NNConv convolutional layers, BatchNorm layers, and the Set2Set pooling module were loaded from the pre-trained model, while the original FC layer was discarded and replaced with a new FC layer with two output units, compatible with the binary task. Initially, all inherited layers were frozen, allowing only the new FC layer to be trained. From the first epoch, convolutional layers 3 and 4 were unfrozen, and from the fifth epoch, all the convolutional layers were progressively unfrozen for fine-tuning.

The classifier architecture remained consistent with the original embedding model, five NNConv layers, Set2Set pooling, and a final FC layer, but with weights adapted to the new domain through fine-tuning. This progressive unfreezing strategy preserved useful representations learned during pre-training while selectively adapting the model to the specific characteristics of the new task. For each division carried out in cross-validation, we selected the best TL model based on the performance obtained. In the end, the TL model with the best performance in the validation subsets was chosen.

4. Results and discussion

In this article, we performed the classification of riboswitches using a multilevel approach. To this end, four datasets were generated for the TL model, containing 200, 400, 1,000 and 2,000 sequences, respectively, the same number of sequences used for the positive (riboswitch) and negative (genomic background) sets. For the TL model training, the pre-trained model was used to take advantage of acquired knowledge about ncRNAs in general and improve performance in the specific task of classifying riboswitches.

4.1. Pre-trained model

The pre-trained model was used to classify seven classes of ncRNAs (see Table 1). These results show that the model achieved high performance rates in all classes, with precision, recall and F1 score values above 96% in all metrics. The precision and recall of 1.0 for 5S RNA and tRNA, respectively, can be explained by each being chosen from a single Rfam family on the eukaryote class.

4.2. TL model

For comparison purposes, we tested the model with and without the use of TL. Table 2 summarizes the performance of the models, with and without TL, on datasets presenting

Table 1. Performance of the pre-trained model.

Class	Precision	Recall	F1-score
5S rRNA	1.000	0.990	0.995
5.8S rRNA	0.996	0.966	0.981
CD-box	0.955	0.990	0.972
HACA-box	0.951	0.983	0.967
miRNA	0.993	0.990	0.991
scaRNA	0.979	0.943	0.961
tRNA	0.993	1.000	0.996

different sizes. The results indicate that for smaller datasets (200 and 400 sequences), the models with TL outperformed those without TL, with improvements in the evaluated metrics. However, for larger datasets (1,000 and 2,000 sequences), models without TL performed slightly better. For sets with 1,000 sequences, the results were similar, while for sets with 2,000 sequences, the differences were less than 3%. Overall, these results suggest that the use of TL can be advantageous when working with limited training data, while its advantage decreases as the size of the dataset increases.

Table 2. Performance of the model with and without TL. In blue (yellow), it is shown the better (worst) results when comparing the models with/without TL.

Model with/without TL	Accuracy	Precision	Recall	F1-score
200 (with TL)	91.67%	92.09%	91.67%	91.65%
200 (without TL)	90.00%	90.00%	90.00%	90.00%
400 (with TL)	85.00%	86.46%	85.00%	84.85%
400 (without TL)	84.17%	86.44%	84.17%	83.92%
1,000 (with TL)	92.33%	92.34%	92.33%	92.33%
1,000 (without TL)	92.33%	92.38%	92.33%	92.33%
2,000 (with TL)	91.50%	91.58%	91.50%	91.50%
2,000 (without TL)	94.17%	94.18%	94.17%	94.17%

Figure 3 shows the ROC curves obtained for cases using datasets with 200, 400, 1,000 and 2,000 sequences, respectively. For 200 and 400 sequences, it can be seen that the models with TL performed better than the models without TL. In the set with 200 sequences, the TL model obtained an improvement of approximately 3% in the Area Under the ROC Curve (AUC) compared to the model without TL. Similarly, in the set with 400 sequences, the improvement was around 2%. However, for 1,000 and 2,000 sequences, we observed that models without TL achieved higher AUC values. Specifically, the model without TL outperformed the TL model by approximately 1% for the dataset with 1,000 sequences and by approximately 2% for the dataset with 2,000 sequences.

There are other computational methods in the literature for classifying riboswitches. Methods such as those proposed by [Beyene, S. S. et al. 2020] and [Premkumar, K. A. R. et al. 2020] have the objectives of differentiating Rfam families of riboswitches. These methods achieved high accuracy, $\sim$99%. [Premkumar, K. A. R. et al. 2020] used a set of about 35,000 riboswitch sequences belonging to 32 Rfam families, while [Beyene, S. S. et al. 2020] used 4,767 sequences from 16 families. Since the methods were specifically designed for classifying riboswitch fami-

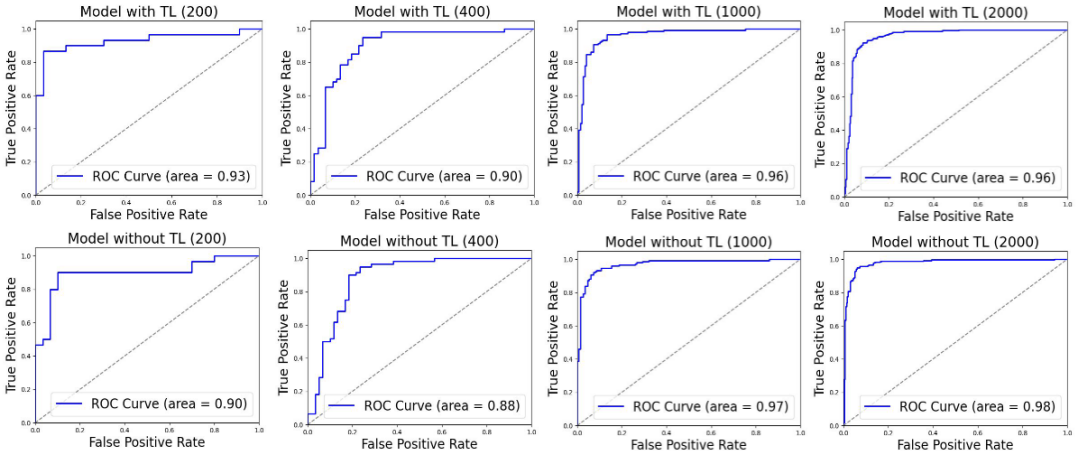

Figure 3. ROC curves for models with (top row) and without TL (bottom row) using the dataset with 200, 400, 1000 and 2000 sequences

lies, their applicability to broader scenarios is limited. Although we achieved lower classification accuracy compared to these riboswitch-specific models, our method was designed to be broadly applicable and is not limited to models for specific riboswitches.

[Chen, K. et al. 2023, Chantsalnyam, T. et al. 2021, Wang, L. et al. 2021, Chantsalnyam, T. et al. 2020, Wang, L. et al. 2020, Fiannaca, A. et al. 2017] presented supervised ML models to classify 13 classes of ncRNAs, with 500 sequences each, including riboswitches. The best results are from [Chantsalnyam, T. et al. 2021]. Regarding riboswitches, we compared the results of our TL method with 1,000 and 2,000 sequences. Their results for 500 sequences report recall, precision and F1-score, with 90.40, 91.70, and 90.90, respectively, while our results were 92.67, 92.05, and 92.36, for 1,000 sequences, and 93.67, 89.78, and 91.68, for 2,000 sequences. Our method exhibits comparable values, noting that our recall is better, the precision for 1,000 was higher and with 2,000 lower, and the F1-score is similar to or better.

5. Conclusion

In this article, we presented a multi-level method that combines a graph convolutional network (GCN) with transfer learning (TL) to classify riboswitches. The method uses pre-trained models on seven specific ncRNA classes, microRNAs (miRNAs), CD-box, H/ACA-box and scaRNA, 5.8S-rRNA and 5S-rRNA, and transfer RNA (tRNA). For training and testing, we carefully constructed the datasets through a search of sequences in species trees to maximize taxonomic data diversity. For this, we propose the SSTST problem, to select sequences present in many species trees in order to maximize the number of distinct families and species. Our results indicated that the use of TL can be advantageous when working with limited training data, while its advantage decreases as the size of the dataset increases. Furthermore, we aim to: study the complexity of each algorithm used in our method; use ablation test to quantify the impact of TL, GCN, and the SSTST in the results; and do an extensive comparison with uptodate methods. Moreover, we plan to use the same method for other classes of ncRNAs, as IRES and Leader sequences.

References

Ammunét, T. et al. (2022). Deep learning tools are top performers in long non-coding rna prediction. *Briefings in Functional Genomics*, 21(3):230–241.

Asim, M. N. et al. (2021). Advances in computational methodologies for classification and sub-cellular locality prediction of non-coding rnas. *International Journal of Molecular Sciences*, 22:1–43.

Asim, M. N. et al. (2020). A robust and precise convnet for small non-coding RNA classification (RPC-snRC). *IEEE Access*, 9:19379–19390.

Bansal, S. et al. (2024). Exploration of deep learning and transfer learning techniques in bioinformatics. In *Applying Machine Learning Techniques to Bioinformatics: Few-Shot and Zero-Shot Methods*, pages 238–257. IGI Global.

Beyene, S. S. et al. (2020). A novel riboswitch classification based on imbalanced sequences achieved by machine learning. *PLoS computational biology*, 16(7):e1007760.

Breaker, R. R. (2011). Prospects for riboswitch discovery and analysis. *Molecular Cell*, 43(6):867—-879.

Chantsalnyam, T. et al. (2020). ncRDeep: non-coding RNA classification with convolutional neural network. *Computational Biology and Chemistry*, 88:107364.

Chantsalnyam, T. et al. (2021). ncRDense: a novel computational approach for classification of non-coding RNA family by deep learning. *Genomics*, 113(5):3030–3038.

Chen, K. et al. (2023). ncDENSE: a novel computational method based on a deep learning framework for non-coding RNAs family prediction. *BMC Bioinformatics*, 24(1):68.

Federhen, S. (2012). The NCBI taxonomy database. *Nucleic acids research*, 40(D1):D136–D143.

Fiannaca, A. et al. (2017). nRC: non-coding RNA classifier based on structural features. *BioData Mining*, 10(1):1–18.

Geer, L. Y. et al. (2009). The ncbi biosystems database. *Nucleic Acids Research*, 38(suppl_1):D492–D496.

Kavita, K. and Breaker, R. R. (2023). Discovering riboswitches: the past and the future. *Trends in Biochemical Sciences*, 48(2):119–141.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

LeCun, Y. et al. (2015). Deep learning. *nature*, 521(7553):436.

Leinster, T. and Meckes, M. W. (2016). Maximizing diversity in biology and beyond. *Entropy*, 18(3):88.

Liu, J. et al. (2006). Distinguishing protein-coding from non-coding RNAs through support vector machines. *PLoS genetics*, 2(4):e29.

Lorenz, R. et al. (2011). ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6:1–14.

McCown, P. J. et al. (2017). Riboswitch diversity and distribution. *RNA*, 23(7):995–1011.

Nawrocki, E. P. et al. (2009). Infernal 1.0: inference of RNA alignments. *Bioinformatics*, 25(10):1335–1337.

Olenginski, L. T. et al. (2024). Flipping the script: Understanding riboswitches from an alternative perspective. *Journal of Biological Chemistry*, 300(3):105730.

Oliveira, J., Costa, F., and Backofen, R. e. a. (2016). SnoReport 2.0: new features and a refined support vector machine to improve snoRNA identification. *BMC Bioinformatics*, 17(18):73–86.

Ontiveros-Palacios, N. et al. (2024). Rfam 15: RNA families database in 2025. *Nucleic Acids Research*, 53(D1):D258–D267.

Pardi, F. and Goldman, N. (2005). Species choice for comparative genomics: being greedy works. *PLoS Genetics*, 1(6):e71.

Premkumar, K. A. R. et al. (2020). Riboflow: Using deep learning to classify riboswitches with 99 accuracy. *Frontiers in Bioengineering and Biotechnology*, 8:808.

Rossi, E. et al. (2019). ncRNA classification with graph convolutional networks. *arXiv preprint arXiv:1905.06515*.

Sakamoto, T. et al. (2021). Taxallnomy: an extension of ncbi taxonomy that produces a hierarchically complete taxonomic tree. *BMC bioinformatics*, 22:1–23.

Singh, J. et al. (2021). Improved rna secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning. *Bioinformatics*, 37(17):2589–2600.

Stagno, J. R. and Wang, Y.-X. (2024). Riboswitch mechanisms for regulation of p1 helix stability. *International Journal of Molecular Sciences*, 25(19):10682.

Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

Wang, L. et al. (2020). ncRFP: a novel end-to-end method for non-coding RNAs family prediction based on deep learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(2):784–789.

Wang, L. et al. (2021). ncDLRES: a novel method for non-coding RNAs family prediction based on dynamic LSTM and ResNet. *BMC Bioinformatics*, 22:1–14.

Weiss, K. et al. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

Zhan, Z. et al. (2022). Evolutionary deep learning: a survey. *Neurocomputing*, 483:42–58.

Zhang, X. et al. (2022). Pinc: a tool for non-coding RNA identification in plants based on an automated machine learning framework. *International Journal of Molecular Sciences*, 23(19):11825.