Single Sequence based Feature Engineering for Convolutional Neural Networks towards RNA Contact Map Prediction

1st Mahmood A Rashid
Institute for Integrated and Intelligent Systems
Griffith University, Nathan
Queensland, Australia
mahmood.rashid@griffith.edu.au

2nd Kuldip K Paliwal
Institute for Integrated and Intelligent Systems
Griffith University, Nathan
Queensland, Australia
k.paliwal@griffith.edu.au

***Abstract*—Features are crucial for deep learning models because they help understand complex data, learn patterns, and make accurate predictions. Feature engineering can sometimes be computationally expensive, but it can also speed up the training and inference phases of deep learning models. By providing a more concise and informative representation, it reduces the number of parameters and computations required in the downstream operations. Well-chosen features can enhance a model's ability to represent data in a structured and meaningful way. They help learn hierarchical dependencies in data, reduce the dimensionality of the input data, and generalizing from the training data to unseen data to make the model robust. In this work, we present a self-supervised learning model for feature generation from RNA sequences towards applying in deep learning models for RNA contact map prediction. We test the efficacy of our extracted features by comparing the prediction performance with the prediction performance obtained by the features extracted using the state-of-the-art language foundation model, RNA-FM. We found our approach promising.**

***Index Terms*—feature extraction, feature engineering, RNA structure, RNA contact map, deep learning**

I. Introduction and Background

Ribonucleic acids (RNAs) play critical roles in various biological processes governed by their structures. RNAs need to be folded into three-dimensional (3D) structures to perform their specific tasks. Therefore, to study the biological functions of non-coding RNAs we need to know their structures. The experimental methods, the X-ray crystallography, nuclear magnetic resonance (NMR), and cryogenic electron microscopy (cryo-EM) are highly accurate in determining RNA structures, however, these methods are expensive, time consuming, and in case of X-ray crystallography, the RNA sample need to be crystallized which itself is challenging [1], [2]. In reality, the number of non-coding RNA sequences are increasing in RNACentral database [3] almost in an exponential rate and the number of known 3D (also known as tertiary structure) structures are lagging far behind in protein data bank (PDB) repository [4]. Hence, an active research domain towards developing computational methods has evolved to predict RNA secondary and tertiary structures aiming to bridge the gap [5]–[9]. For better prediction, machine learning models demand for better features and better features could be achieved through feature engineering.

A. Features in Machine Learning

Features, also known as attributes or variables are the measurable properties or characteristics of data that are used as input for training a machine learning model [10]. Features represent the information that a model learns from to make future predictions, to classify data, or to perform a specific task. Features can be thought of as the dimensions of the data space that the machine learning algorithm explores to identify patterns and relationships. Features are equally important as the learning algorithms for deep learning models. Effective feature engineering and selection of features are often essential for achieving the best performance in deep learning applications. The better the features are, the better the model will be able to learn and make accurate predictions. Based on the nature of the data, features could be of different types such as, numerical features, categorical features, textual features, image features, time-series features, and special domain specific features as well.

B. Feature Engineering

Usually data does not receive in a direct usable format for the machine learning models. Hence, preprocessing of data is a vital and one of the most time consuming steps in developing a machine learning model [11]. Feature engineering is the process of transforming raw data into features that can be used by machine learning models [12], [13]. This process involves selecting the most relevant features from the data, cleaning and preprocessing the data, creating new features, and eliminating unnecessary features. In short, feature engineering is the process of preparing feature-set that are more informative and predictive of the target variable. Well-engineered features can help models to learn more efficiently and accurately, and they can also make models more interpretable and robust.

Here are some common feature engineering techniques:

- **Feature selection:** This involves selecting the most relevant features from the data [13]. This can be done using statistical methods, such as correlation analysis and information gain, or by using domain knowledge to identify the features that are most likely to be predictive of the target variable.

- **Data cleaning and preprocessing:** This involves removing any errors or inconsistencies from the data, and transforming the data into a format that is compatible with the machine learning algorithm. This may involve scaling the data, converting categorical features to numerical features, and handling missing values.
- **Feature creation:** This involves creating new features from the data that are more informative and predictive of the target variable. This can be done by combining existing features, transforming existing features, or creating entirely new features based on domain knowledge.

Feature engineering is a complex and iterative process. It is important to experiment with different feature engineering techniques and to evaluate the performance of the model on a held-out validation set to ensure that the features are improving the model's performance.

C. *RNA Features*

In single sequence RNA structure prediction, the features are extracted solely based on the nucleotide compositions of the sequences [14]. The features are enhanced by including the physicochemical properties of nucleotide, the level of free energy, functional motif, base-pairing, Position-specific scoring matrix (PSSM) [15], and functional and statistical information. Some of the widely used feature engineering methods are as follows:

- **One-hot vector:** a sequence of length L is represented by a binary *one-hot* vector of shape (L, 4), here 1 corresponds to the presence of any base types (A, U, C, G) and 0 elsewhere, i.,e., $A : [1, 0, 0, 0], U : [0, 1, 0, 0], C : [0, 0, 1, 0], G : [0, 0, 0, 1]$. A missing or invalid nucleotide is represented by -1 value, i.e., $N : [-1, 0, 0, 0], X : [0, -1, 0, 0]$ and so on. Many successful machine learning models applied one-hot representation of the sequence data [16]–[18].
- **Nucleotide composition:** this approach computes the frequencies of nucleotide (A, U, G, C) or di-nucleotide (occurrences of adjacent nucleotides) in the sequence and used as the features [19].
- **K-mers:** this approach computes the frequencies of all possible sub-sequences of length k in the RNA sequence e.g., 3-mers, 4-mers. K-mers capture the local sequence patterns [20].
- **Base-pairing:** this approach computes the frequency of complementary nucleotide pairs also known as canonical base pairs or Watson-Crick base pairs - AU , GC. Non-canonical also known as wobble base pairs may also be considered in feature extraction but with lower weights in comparison to the canonical base pairs [21].
- **Free energy:** this approach computes the free energy associated with the secondary structure. The structures with lower free energies consider as the more stable structures [15].
- **Embedding:** this is a machine learning based approach widely applied in language translation models to generate word embedding [22]. Transformer [23] based models learn contextualized representations of RNA sequences that capture positional dependencies of nuleotides in the sequences.

Feature engineering is an essential skill for machine learning practitioners. By mastering the art of feature engineering, better models can be developed to make better predictions.

II. Related work

A. *One-hot encoding based $L \times L \times 8$ features:*

One of the simplest but effective feature engineering techniques where a sequence of length L is represented by a binary *one-hot* vector of size $L \times 4$ as described in Section IC. In SPOT-RNA [24], SPOT-RNA2 [17], SPOT-RNA-2D [18], this $L \times 4$ *one-hot* encoding was transformed into a feature tensor of shape $(L, L, 8)$ applying an outer concatenation method [25] in the data prepossessing phase to generate the input data for the convolutional neural network. In another work [16], x $(L \times 4)$ *one-hot* encoded sequence has been transformed into a feature tensor of shape $(L, L, 16)$ through a Kronecker product between x and itself to apply in convolutional neural network based on UNet architecture.

B. *Basepairing based $L \times L \times 8$ features:*

This feature engineering approach was introduced by Booy *etal.* in [21] where a sequence of length L has been represented as an $L \times L \times 8$ tensor which can be considered as an $L \times L$ 2D map with 8 channels corresponds to 8 *one-hot* representations as below:

- Six channels correspond to the pairs (A, U), (U, A), (U, G), (G, U), (G, C), (C, G)
- One channel corresponds to pairs (A,A), (U, U), (C, C) and (G, G), where a base is paired with itself that represents the diagonal of the map.
- One channel represents the non-valid combination of bases due to the distance (too short for pairing) or any other constraints.

C. *Language model based features:*

The RNA foundation model (RNA-FN) [22] is based on the BERT [26] language model architecture and is built on 12 transformer-based encoder blocks. RNA-FM model generates $L \times 640$ embedding matrix for an RNA of length L which is further applied in the downstream model training for RNA structure prediction. An end-to-end deep learning model, called E2Efold [27] also applied the transformers [23] with convolutional neural network for feature extraction and structure prediction, respectively.

Among other feature extraction methods, an $L \times L$ LinearPartition [28], an $L \times 4$ PSSM [29], and an $L \times L$ Direct Coupling Analysis (DCA) [29] are also used in single sequence RNA structure prediction.

III. OUR APPROACHES

Although the *one-hot* encoding is easy to implement and produce competitive results, it does not capture the implicit relations between the bases in the sequence [30]. Keeking this in mind, we presented a self-supervised learning (SSL) based feature extraction approach. We developed and trained a self-supervised machine learning model for feature generation to use as an upstream transfer learning model with a downstream deep learning models as shown in Figure 1.

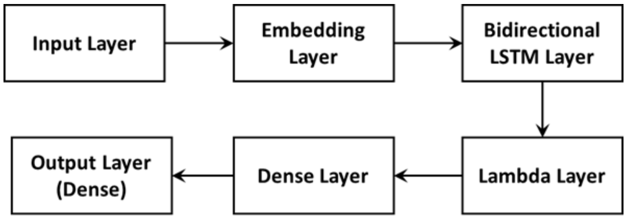


Fig. 1. **Self-supervised learning model**, extracts features for applying in deep learning models.

The SSL model consists of six layers having more than two million (2m) trainable parameters. The layers are as follows:

- **One input layer:** This layer takes input as a sequence. We convert the RNA sequences into *tokens* to extract the vocabularies (A, U, G, C) so that the next layer can generate embedding.
- **One embedding layer:** This layer generates embedding from the series of tokens to apply to the Long Short-Term Memory (LSTM) layers in the next of the network.
- **One Bidirectional layer:** This layer consists to two LSTM layers, one for forward pass and another for backward pass.
- **One lambda layer:** This layer is applied for feature reduction using TensorFlow reduce mean functionality that computes the mean of elements across dimensions of a tensor.
- **Two dense layers:** These are fully connected layers of the network applied for dimensionality reduction.

We consider Bidirectional LSTM because this algorithm captures the intrinsic relation among the constituent elements (nucleotide) for sequential data. The SSL model generates a feature set of size $L \times 640$ for an RNA sequence of L bases.

IV. RESULTS AND DISCUSSION

A. Datasets:

We trained the SSL model using the STRAlign datasets [31] for feature generation which is a widely used benchmark RNA dataset consists of 37149 sequences. We used SPOT-RNA-2D [18] datasets (*TR - training, VL - validation, TS1 - test 1, TS2 - test 2*) to evaluate the performance of our feature extraction model. Towards this process, we first generate features for SPOT-RNA-2D datasets using using RNA-FM [22] and the earlier trained SSL model. We then, train the downstream deep neural network separately with these two diffident sets of features.

B. Deep network for feature evaluation:

Our convolutional neural network is based on ResNet [32] architecture as shown in Figure 2. The ResNet model consists of 40 convolution (Conv2d) layers of identical parameters (filters= 40, kernels= $(3, 3)$, strides= $(1, 1)$) with one fully connected output layer. Each Conv2d layer is accompanied by layer-normalization and dropout operations. The model uses the $ReLU$ activation function and $Adam$ optimizer. The $sigmoid$ function is applied to the fully connected output layer to predict the probabilities of the basepair contacts.

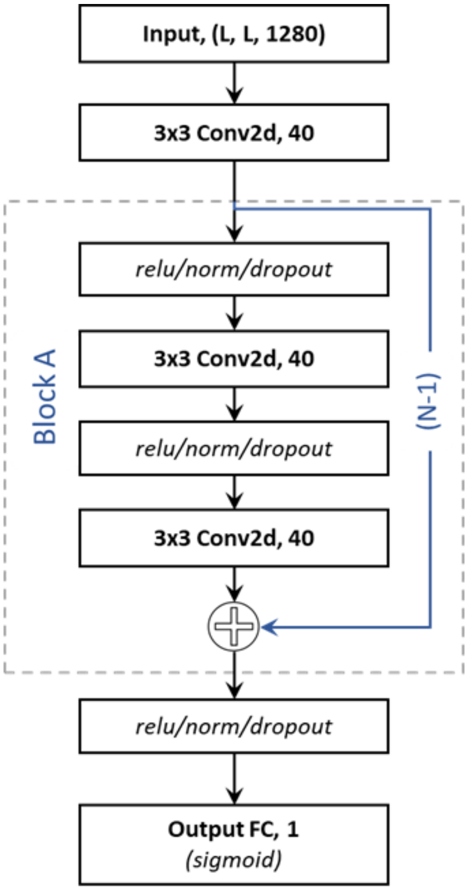


Fig. 2. **ResNet architecture**, a deep convolutional neural network applied for RNA distance based contact map prediction. This network is trained separately with both RNA-FM and SSL based features to evaluate the performance of the feature sets.

Note that for both feature generation approach, we generate $L \times 640$ features for an RNA sequence of length L. Then this $L \times 640$ feature vector has been transformed to a tensor of shape $L \times L \times 1280$ using an outer concatenation function to apply as the input of the 2D convolutional neural networks.

We present the performance measures of both the models in terms of precision $(TP/(TP+FP))$, sensitivity $(TP/(TP+FN))$, accuracy $((TP+TN)/(TP+TN+FN+FP))$, and F1-score $(2(precision \times sensitivity)/(precision + sensitivity))$, where TP, TN, FP, and FN are true positives, true negatives, false positives and false negatives, respectively.

TABLE I
PERFORMANCE METRICS OF THE EXTRACTED FEATURES WHEN APPLYING IN THE DOWNSTREAM DEEP LEARNING MODEL.

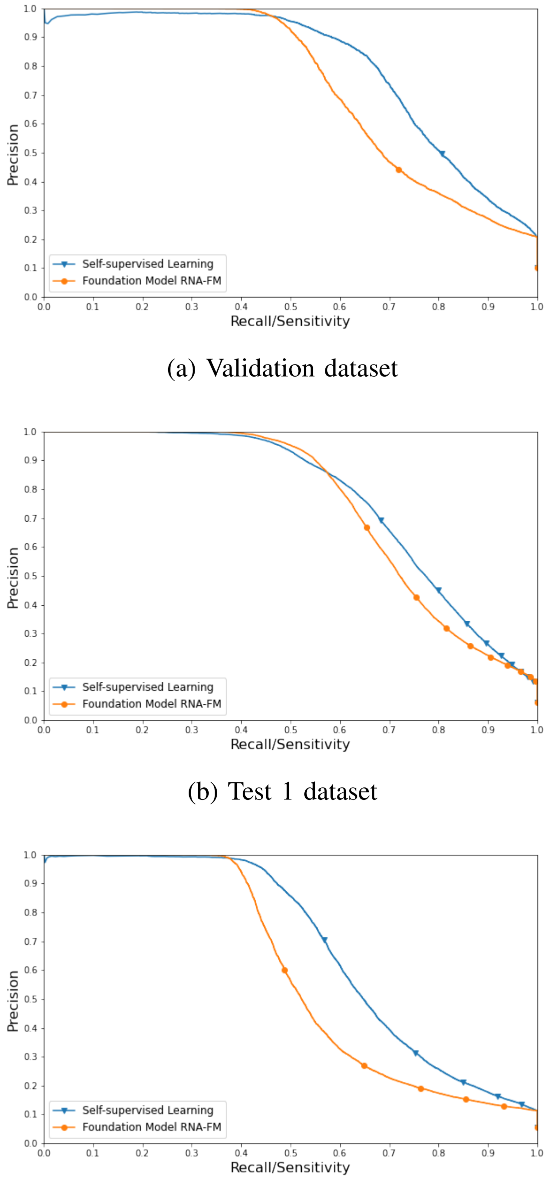
(A) VALIDATION DATASET				
Feature extraction model	**F1-score**	**Precision**	**Recall**	**Accuracy**
Foundation Model, RNA-FM	0.659	**0.866**	0.532	0.945
Self-supervised BiLSTM	**0.734**	0.836	**0.654**	**0.952**
(B) TEST DATASET 1				
Feature extraction model	**F1-score**	**Precision**	**Recall**	**Accuracy**
Foundation Model, RNA-FM	0.688	**0.833**	0.587	**0.968**
Self-supervised BiLSTM	**0.702**	0.792	**0.630**	0.967
(C) TEST DATASET 2				
Feature extraction model	**F1-score**	**Precision**	**Recall**	**Accuracy**
Foundation Model, RNA-FM	0.568	**0.869**	0.422	0.964
Self-supervised BiLSTM	**0.638**	0.814	**0.525**	**0.967**

The data in Table I shows that our SSL based features produces better *F1-score* and *Recall* for all three datasets, better *Accuracy* for validation and test-2 datasets, and better *Precision* for only validation dataset. The *bold-faced* values in Table I represent the winners of the corresponding measures.

The performance of different features are further illustrated through the precision-recall (PR) curve on validation and test datasets as shown in Figure 3(a, b and c). PR-curve is very significant performance measure for this type of binary classification problem (0-unpaired, 1-paired) where datasets are highly imbalanced i.e., one class (0-unpaired) is much more prevalent that others. PR curves help visualize the trade-off between precision and recall. The PR curve of SSL is higher and to the right of RNA-FM model which demonstrates that SSL generated feature based deep learning model achieves higher precision and recall at the same time. The SSL generated feature based model shows better performance on both validation and test datasets.

The similar performance trends are also observed for receiver operating characteristics (ROC) curves as presented in Figure 4(a, b, and c) for both validation and test sets. The ROC distinguishes between the two classes (positive and negative) across various threshold values. ROC curves help visualize the trade-off between the false positive rate (TPR) and true positive rate (FPR) of a model which are valuable for comparing different models' performances. The ROC curve of SSL generated feature based model is higher and more to the left of the RNA-FM generated feature based model which demonstrates that our SSL based model achieves higher TPR and lower FPR at the same time, hence is performing better.

We also listed the Area Under Curve (AUC) values for both PR-curve and ROC-curve in Table II. For both AUC-PR and AUC-ROC the values of AUC are well-above 0.5 which suggest that the model is outperforming random guessing at distinguishing between the classes and is doing better in positive prediction minimising the false positives.



(a) Validation dataset

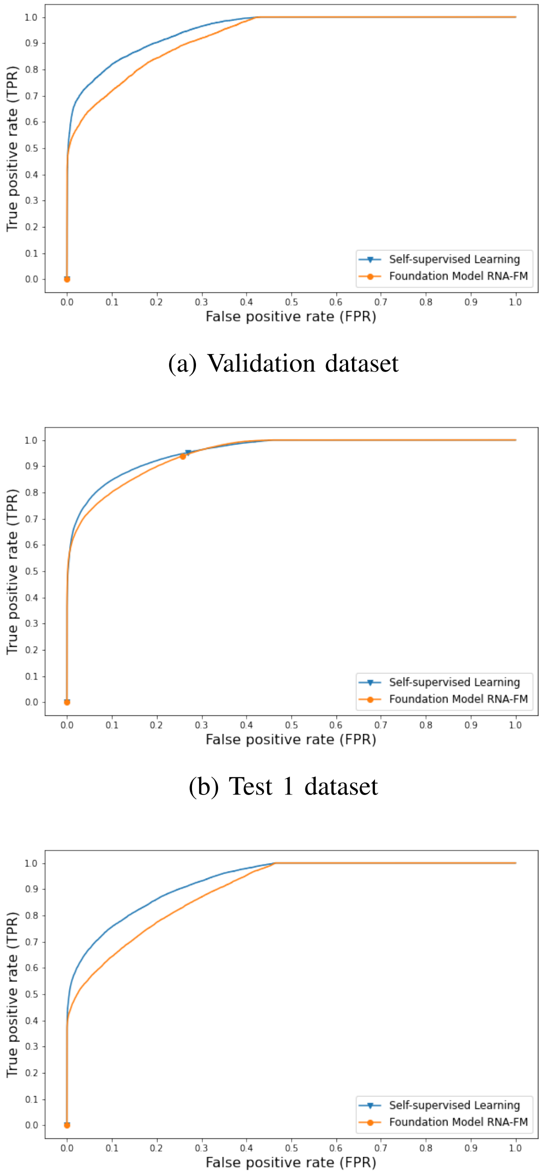
(b) Test 1 dataset

(c) Test 2 dataset

Fig. 3. **Precision-recall curves** present the performance of both feature sets. (a), (b), and (c) represent the PR-curve for validation, test 1 and test 2 datasets. The figures show that the features generated using SSL is performing better than the feature generated by RNA-FM.

CONCLUSION

Feature engineering is a crucial step in the machine learning pipeline as it can have a significant impact on the performance of the model. Feature engineering can help improve the model accuracy, reduce the overfitting issues, reduce the feature dimensionality, improve the model's robustness, and improve the model interpretability. In this work, we demonstrate a Bidirectional LSTM based self-supervised feature engineering



(a) Validation dataset

(b) Test 1 dataset

(c) Test 2 dataset

Fig. 4. **ROC curve** present the performance of both feature sets. (a), (b), and (c) represent the ROC-curve for validation, test 1 and test 2 datasets. The figures show that the features generated using SSL is performing better than the feature generated by RNA-FM.

TABLE II
Values of Area Under Curve (AUC) for both PR-curve and ROC-curve.

	PR-AUC			**ROC-AUC**		
Model	**VL**	**TS1**	**TS2**	**VL**	**TS1**	**TS2**
RNA-FM based	0.731	0.744	0.592	0.925	0.947	0.898
SSL based	**0.796**	**0.768**	**0.683**	**0.95**	**0.955**	**0.932**

model that trained on a large RNA dataset and used to generate features to apply to train a convolutinal neural network based deep learning model. We also trained the same network with the features generated by the state-of-the-art transformer based language model, RNA-FM to verify our approach. We found that our feature engineering model is comparable and in most of the indicators perform better in comparison to the RNA-FM based feature generating model. Feature engineering is a complex and challenging task, however, it is essential for building successful machine learning models.

Acknowledgment

We would like to acknowledge the *Signal Processing Lab* and the *High-Performance Computing clusters* of Griffith University, Queensland, Australia.

References

[1] S. E. Lietzke, C. L. Barnes, and C. E. Kundrot, "Crystallization and structure determination of RNA," *Current Opinion in Structural Biology*, vol. 5, no. 5, pp. 645–649, 1995.

[2] A. Marchanka, B. Simon, G. Althoff-Ospelt, and T. Carlomagno, "RNA structure determination by solid-state nmr spectroscopy," *Nature communications*, vol. 6, no. 1, p. 7024, 2015.

[3] RNACentral, "RNAcentral: a hub of information for non-coding RNA sequences," *Nucleic Acids Research*, vol. 47, no. D1, pp. D221–D229, 2019.

[4] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

[5] M. Solayman, T. Litfin, J. Singh, K. Paliwal, Y. Zhou, and J. Zhan, "Probing RNA structures and functions by solvent accessibility: an overview from experimental and computational perspectives," *Briefings in Bioinformatics*, vol. 23, no. 3, p. bbac112, 2022.

[6] Q.-J. Luo, J. Zhang, P. Li, Q. Wang, Y. Zhang, B. Roy-Chaudhuri, J. Xu, M. A. Kay, and Q. C. Zhang, "RNA structure probing reveals the structural basis of dicer binding and cleavage," *Nature communications*, vol. 12, no. 1, pp. 1–12, 2021.

[7] Z. Cai, C. Cao, L. Ji, R. Ye, D. Wang, C. Xia, S. Wang, Z. Du, N. Hu, X. Yu *et al.*, "Ric-seq for global in situ profiling of RNA-RNA spatial interactions," *Nature*, vol. 582, no. 7812, pp. 432–437, 2020.

[8] S. Janssen and R. Giegerich, "The RNA shapes studio," *Bioinformatics*, vol. 31, no. 3, pp. 423–425, 2015.

[9] J. S. Reuter and D. H. Mathews, "RNAstructure: software for RNA secondary structure prediction and analysis," *BMC bioinformatics*, vol. 11, no. 1, pp. 1–9, 2010.

[10] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[11] F. Nargesian, H. Samulowitz, U. Khurana, E. B. Khalil, and D. S. Turaga, "Learning feature engineering for classification." in *Ijcai*, vol. 17, 2017, pp. 2529–2535.

[12] G. Dong and H. Liu, *Feature engineering for machine learning and data analytics*. CRC press, 2018.

[13] M. Kuhn and K. Johnson, *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC, 2019.

[14] M. Mathur, S. Patiyal, A. Dhall, S. Jain, R. Tomer, A. Arora, and G. P. Raghava, "Nfeature: A platform for computing features of nucleotide sequences," *BioRxiv*, pp. 2021–12, 2021.

[15] A. R. Gruber, R. Lorenz, S. H. Bernhart, R. Neuböck, and I. L. Hofacker, "The vienna RNA websuite," *Nucleic acids research*, vol. 36, no. suppl_2, pp. W70–W74, 2008.

[16] L. Fu, Y. Cao, J. Wu, Q. Peng, Q. Nie, and X. Xie, "Ufold: fast and accurate RNA secondary structure prediction with deep learning," *Nucleic acids research*, vol. 50, no. 3, pp. e14–e14, 2022.

[17] J. Singh, K. Paliwal, T. Zhang, J. Singh, T. Litfin, and Y. Zhou, "Improved RNA secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning," *Bioinformatics*, vol. 37, no. 17, pp. 2589–2600, 2021.

[18] J. Singh, K. Paliwal, T. Litfin, J. Singh, and Y. Zhou, “Predicting RNA distance-based contact maps by integrated deep learning on physics-inferred secondary structure and evolutionary-derived mutational coupling,” *Bioinformatics*, vol. 38, no. 16, pp. 3900–3910, 2022.

[19] C.-C. Chen and Y.-M. Chan, “Redfold: accurate rna secondary structure prediction using residual encoder-decoder network,” *BMC bioinformatics*, vol. 24, no. 1, pp. 1–13, 2023.

[20] A. Li, J. Zhang, and Z. Zhou, “Plek: a tool for predicting long non-coding rnas and messenger rnas based on an improved k-mer scheme,” *BMC bioinformatics*, vol. 15, pp. 1–10, 2014.

[21] M. Saman Booy, A. Ilin, and P. Orponen, “RNA secondary structure prediction with convolutional neural networks,” *BMC bioinformatics*, vol. 23, no. 1, pp. 1–15, 2022.

[22] J. Chen, Z. Hu, S. Sun, Q. Tan, Y. Wang, Q. Yu, L. Zong, L. Hong, J. Xiao, T. Shen *et al.*, “Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions,” *bioRxiv*, 2022.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[24] J. Singh, J. Hanson, K. Paliwal, and Y. Zhou, “RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning,” *Nature communications*, vol. 10, no. 1, pp. 1–13, 2019.

[25] S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, “Accurate de novo prediction of protein contact map by ultra-deep learning model,” *PLoS computational biology*, vol. 13, no. 1, p. e1005324, 2017.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[27] X. Chen, Y. Li, R. Umarov, X. Gao, and L. Song, “RNA secondary structure prediction by learning unrolled algorithms,” *arXiv preprint arXiv:2002.05810*, 2020.

[28] H. Zhang, L. Zhang, D. H. Mathews, and L. Huang, “Linearpartition: linear-time approximation of RNA folding partition function and base-pairing probabilities,” *Bioinformatics*, vol. 36, no. Supplement_1, pp. i258–i267, 2020.

[29] Z. Zhang, P. Xiong, T. Zhang, J. Wang, J. Zhan, and Y. Zhou, “Accurate inference of the full base-pairing structure of RNA by deep mutational scanning and covariation-induced deviation of activity,” *Nucleic acids research*, vol. 48, no. 3, pp. 1451–1465, 2020.

[30] H. Zhang, C. Zhang, Z. Li, C. Li, X. Wei, B. Zhang, and Y. Liu, “A new method of RNA secondary structure prediction based on convolutional neural network and dynamic programming,” *Frontiers in genetics*, vol. 10, p. 467, 2019.

[31] Z. Tan, Y. Fu, G. Sharma, and D. H. Mathews, “Turbofold ii: Rna structural alignment and secondary structure prediction informed by multiple homologs,” *Nucleic acids research*, vol. 45, no. 20, pp. 11 570–11 581, 2017.

[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.