Analytical Biochemistry 654 (2022) 114802

Contents lists available at ScienceDirect

Analytical Biochemistry

journal homepage: www.elsevier.com/locate/yabio

Predicting RNA solvent accessibility from multi-scale context feature via multi-shot neural network

Xue-Qiang Fan [a], Jun Hu [a,*], Yu-Xuan Tang [a], Ning-Xin Jia [a], Dong-Jun Yu [b,**], Gui-Jun Zhang [a,***]

[a] *College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310023, China*
[b] *School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolingwei, Nanjing, 210094, China*

ARTICLE INFO

Keywords:
RNA solvent Accessibility prediction
Bioinformatics
Sequence-based feature
Multi-scale context feature learning
Deep residual attention network

ABSTRACT

Knowledge of RNA solvent accessibility has recently become attractive due to the increasing awareness of its importance for key biological process. Accurately predicting the solvent accessibility of RNA is crucial for understanding its **3D** structure and biological function. In this study, we develop a novel computational method, termed M^{2}pred, for accurately predicting the solvent accessibility of RNA from sequence-based multi-scale context feature. In M^{2}pred, three single-view features, **i.e.,** base-pairing probabilities, position-specific frequency matrix, and a binary one-hot encoding, are first generated as three feature sources, and immediately **concatenated to engender a super feature**. Secondly, for the **super feature, the matrix-format features of each nucleotide are extracted** using an **initialized sliding window technique, and** regularly stacked into a cube-format feature**. Then, using** multi-scale context feature extraction strategy, a pyramid feature constructed of contextual feature of four scales related to target nucleotides is extracted from the cube-format feature. Finally, a customized multi-shot neural network framework, which is equipped with four different scales of receptive fields mainly integrating several residual attention blocks, is designed to dig discrimination information from the contextual pyramid feature. Experimental results demonstrate that the proposed M^{2}pred achieve a high prediction performance and outperforms existing state-of-the-art prediction methods of RNA solvent accessibility.

1. Introduction

Knowledge of the RNA solvent accessibility plays an important role in various key biological processes. e.g., RNA-ligand interactions [1,2], protein-RNA recognition studies [3], RNA fold recognition [4], and identification of structural signature in RNA thermal adaptation [5]. Furthermore, it can provide essential clues for the RNA structure prediction, which will speed up the progress of RNA function detection and understanding. Hence, accurate determination of the solvent accessibility of RNA molecule is critical for understanding its tertiary structure and biological function, especially in the post-genome era where a large volume of non-coding RNA sequences without being structurally determined is rapidly accumulated [6–8]. Nevertheless, the traditional wet-laboratory methods, e.g., Cryo-electron microscopy [9], X-ray crystallography [10], nuclear magnetic resonance [11], and hydroxyl radical footprint [12,13], for predicting the solvent accessibility of RNA are expensive and time-consuming. In view of this situation, it is highly desirable to develop cost-effective computational methods for high-throughput and accurate RNA solvent accessibility prediction.

Early-stage methods of predicting RNA solvent accessibility are developed mainly based on traditional machine learning (ML) algorithms. For example, RNAsnap, to the best of our knowledge, it is the first report on solving the problem of RNA solvent accessibility prediction. The RNAsnap presented in this report includes two separate ML-based methods, i.e., RNAsnap-seq [14] and RNAsnap-prof [14], which use the query RNA sequence alone and the evolutionary information from multiple sequence alignment as the input into support-vector-machine (SVM) algorithms, respectively, to predict RNA solvent accessibility. However, applying the ensemble of sequence-based features and traditional ML classifiers widely to process the RNA biology knowledge recognition [14–17] inevitably suffers from certain disadvantages. For instance, with the continuous increase and

[*] Corresponding author.
[**] Corresponding author.
[***] Corresponding author.
E-mail addresses: hujunum@zjut.edu.cn (J. Hu), njyudj@njust.edu.cn (D.-J. Yu), zgj@zjut.edu.cn (G.-J. Zhang).

https://doi.org/10.1016/j.ab.2022.114802
Received 21 February 2022; Received in revised form 11 June 2022; Accepted 28 June 2022
Available online 7 July 2022

rapid accumulation of RNA sequence data, traditional ML algorithms cannot be effectively mine the hidden information in a multitude of sequences, making it difficult for the ML-based methods to be considered the optimal option for training classifiers.

To overcome the drawback of conventional shallow ML algorithms that cannot effectively mine large amounts of data, a few deep learning (DL) techniques that utilize multi-layered artificial neural networks to learn tasks, have been successfully applied to solve many bioinformatics and computational biology problems, including RNA solvent accessibility prediction. Sun et al. used unidirectional long short-term memory recurrent neural networks (ULSTM) [18] to dig out the evolutionary information from improved sequence profiles based on the covariance models [19,20]. ULSTM-based algorithms could improve the accuracy of RNA solvent accessibility prediction; however, due to the limits of the ULSTM, it cannot handle long-range information dependency well. To address this problem, Hanumanthappa et al. proposed two separate deep learning-based methods, i.e., RNAsnap2 [20] and RNAsnap2(SingleSeq) [21], which successfully combined the dilated convolutional neural networks [22,23] with one fusion feature, for predicting RNA solvent accessibility. Nevertheless, despite the efficiency and accuracy achieved, the existing methods still have several following critical deficiencies.

First, to develop powerful computational models for RNA solvent accessibility prediction, a critical step is to extract sufficient discriminative features to construct more accurate models. By revisiting existing RNA solvent accessibility prediction methods, it was found that all of them employ limited scale one-shot feature related to target nucleotides to capture discriminative information. Despite preserving a certain degree of discriminative information, they still lose the remote low-level feature information that is likely to aid in predicting target nucleotides solvent accessibility. Second, a high-performance deep-learning framework, which is able to learn high-level representation knowledge from low-level feature information based on the raw nucleotide sequence, should possess multiple receptive field that accepts multi-scale contextual information, instead of existing ML- and DL-based methods using a single receptive field to capture only an inherent contextual information. Therefore, there remains an urgent need for new and high-performance prediction methods of RNA solvent accessibility.

To address the important issues mentioned above, in this study, we propose a novel deep learning-based method, called M^{2}pred, to further improve the performance of RNA solvent accessibility prediction. Specifically, we first extract three single-view feature sources, i.e., base-pairing probabilities, position-specific frequency matrix, and a binary one-hot encoding, from primary sequences, and immediately concatenate them to engender a super feature. Secondly, for the super feature, the matrix-format features of each nucleotide are extracted using an initialized sliding window technique, and regularly stacked into a cube-format feature. Then, to extract more effective information, we design one multi-scale context feature extraction strategy (refer to the section of "Multi-scale Context Feature Extraction" for detail) to generate a pyramid feature constructed of contextual features of four scales associated with target nucleotides, from the cube-format feature. Finally, a well-designed multi-shot neural network framework, which is equipped with four different scales of receptive fields mainly integrating several residual attention blocks, is designed to capture more discriminative knowledge hidden in the contextual pyramid feature. Benchmarking results and comparisons demonstrate that the proposed M^{2}pred outperforms existing state-of-the-art ML-based methods as well as DL-based methods, and is a suitable DL-based method for predicting RNA solvent accessibility. Furthermore, based on the proposed M^{2}pred, we implement a new standalone-version predictor for predicting RNA solvent accessibility, which is freely available at https://github.com/XueQiangFan/M2pred/for academic use.

2. Materials and methods

2.1. Benchmark datasets

One comprehensive dataset of benchmark RNAs, which is utilized to evaluate the DL-based methods, i.e., RNAsol [20], RNAsnap2 [23], and RNAsnap2(SingleSeq) [21], is collected to fairly examine the effectiveness of the proposed M^{2}pred in this study. This dataset contains three subsets: one training dataset called TR119, two independent testing datasets, named TS45 and TS31, respectively. All RNA sequences in the benchmark dataset, which have >32 nucleotides and < 4 Å X-ray resolution, are non-redundant from each other through CD-HIT-EST [24] and BLASTclust program [25] with identity cut-off of 80% and 30%, respectively. TR119, TS45, and TS31 consist of 119 (119 effectively as 1 RNA appeared twice), 45, and 31 high-resolution RNA chains, respectively. In addition to TS45 and TS31, we further prepare a new independent test set TEST36 (36 RNAs) by downloading all protein-free and protein-complex RNAs (204 chains) which are submitted to PDB after January 2020, the previous date for obtaining TR119, TS45, and TS31. These 204 chains are then filtered using CD-HIT-EST and BLASTclust program with 80% and 30% identity cut-off, respectively, so that the new set is non-redundant from the training (TR119) and test (TS45 and TS31) sets and between each other. As a result, 44 chains were retained. Subsequently, we further exclude potential family RNA chains by searching of these 44 chains against the training and testing data sets with a large E-value cut-off 10 and 0.1 using BLAST-N and Infernal tools [19,26,27], respectively. The final remaining 36 chains constitute the independent testing set, called TEST36. We also statistically analyze the maximum, minimum, and average sequence lengths of each RNA in the four subsets, i.e., TR119, TS45, TS31, and TEST36, and whether it is RNA-protein complexes or not (see Supplementary Table S1). Furthermore, as shown in Supplementary Table S2, for these three subsets, the distribution of the number of Adenine (A), Cytosine (C), Uracil (U), and Guanine (G) nucleotides varies between 22% and 26%, 22% and 27%, 18% and 22%, 29% and 33%, respectively.

The ground-truth labels of relative solvent accessible surface area (RSA) for each RNA of TR119, TS45, TS31, and TEST36 are derived from their tertiary structure using POPS [14,20,21,28]. Concretely, the tertiary structure of the individual chain is extracted by using Biopython [29] from the tertiary structure of multiple chains or RNA-protein complexes. Subsequently, generating the ground-truth ASA values for every RNA chain by POPS [28], a fast tool for solvent accessible surface area (ASA) at atomic and residue level. Finally, the ground-truth labels of RSA in one RNA sequence are normalized values from 0 to 1 by dividing the ASA by the maximum ASA value of the corresponding nucleotide, i.e., A, G = 400 Å2 and U, C = 350 Å2.

2.2. Performance measurement

Two widely used evaluation indexes, i.e., mean absolute error (MAE) and Pearson Correlation Coefficient (PCC), are employed to evaluate the performance of RSA prediction. MAE is used to quantitatively measure the average deviation between the predicted and experimental RSA values of each RNA. PCC is employed to quantify the relationship between the predicted and experimental RSA values of each RNA, and its value is between −1 and 1. The two indexes can be calculated by following equations: $MAE = \sum_i^N |\lambda_i - \mu_i|/N$, $PCC = \sum_i^N (\lambda_i - \overline{\lambda}) \times (\mu_i - \overline{\mu})/\sqrt{\sum_i^N (\lambda_i - \overline{\lambda})^2 \times \sum_i^N (\mu_i - \overline{\mu})^2}$. Where N is the length of the query RNA chain; λ_i and μ_i are the predicted and experimental RSA values of the ith nucleotide in the query RNA, and $\overline{\lambda}$ and $\overline{\mu}$ are the corresponding average values of the entire query RNA, respectively. Furthermore, to show the statistical significance of improvement by M^{2}pred over other state-of-the-art methods, a paired Student's *t*-test [30] and Wilcoxon rank-sum test [31] are used to PCC and MAE and predicted RSA values to obtain *p*-value, respectively. The smaller the *p*-value is, the more significant the difference is between the two methods.

2.3. Feature representation

In this study, the input to M^{2}pred is an RNA sequence of length *L*. Each nucleotide (A, C, G, or U) is encoded into three single-view features using trainable embedding functions, i.e., base-pairing probabilities, position-specific frequency matrix, and a binary one-hot encoding.

(1) *Base-pairing Probabilities (BPP).* Inspired by the fact that protein secondary structure has a critical impact on the protein solvent accessibility [32–34], RNA secondary structure information, i.e., base-pairing probabilities, is employed to predict RNA solvent accessibility as one feature source. To avoid the possible over-fitting, the LinearPartition tool [35] with the thermodynamic parameters, i.e., LinearPartition-V version, which can be downloaded at http://linearfold.org/partition, is utilized to generate the base-pairing probabilities of RNA secondary structure instead of the default version where the parameters are learned using the ML-based algorithm. In detail, given each RNA sequence, LinearPartition-V precisely predicts its base-pairing probabilities matrix (*L* rows and one column, where *L* is the length of the RNA sequence), which reveals the pairing probability of A-U, G-C or G-U base pairs in the RNA sequence.

(2) *Position-Specific Frequency Matrix (PSFM).* Position-Specific Frequency Matrix (PSFM) is employed to dig out the RNA evolutionary information for reflecting the nucleotide conservation score at specific positions and improving the performance of RNA RSA prediction. To obtain the PSFM profile, given a target sequence with *L* nucleotides, first, RNAfold program [36] is used to annotate the secondary structure information of this RNA sequence. Second, Infernal [19], a fast and accurate RNA sequence alignment tool, is utilized to search through the NCBI's nucleotide sequence database (**available at** https://ftp.ncbi.nih.gov/) with default parameters 10.0 as the *E*-value cutoff for constructing an multiple sequence alignment profile (MSA). Then, the MSA is filtered to guarantee that the sequence identity between each two aligned sequences is less than 90% and no aligned sequence includes more than 50% gaps. Finally, based on the filtered precise MSA profile (PMSA), the corresponding PSFM profile with size of $L \times 4$ is calculated as follows:

$$PSFM_{i,j} = \frac{\sum_{m=1}^{M} \psi\left(PMSA_i^m,\ \Phi_j\right) + \beta\left(T_i, \Phi_j\right)}{\sum_{n=1}^{4}\left(\sum_{m=1}^{M} \psi\left(PMSA_i^m,\ \Phi_n\right) + \beta(T_i, \Phi_n)\right)} \tag{1}$$

Where $PSFM_{i,j}$ is the *i*th row and *j*th column element of PSFM profile; $PMSA_i^m$ stands for the nucleotide type at the *i*th position of the *m*th aligned sequence in the PMSA profile, $i = 1,2, \ldots, L$, and $m = 1,2, \ldots, M$; T_i is the nucleotide type at the *i*th position in the target RNA sequence; Φ_j and Φ_n are the nucleotide type of the *j*th and *n*th element of the set of four naturally-occurring nucleotide types, i.e., A, U, C, and G, respectively, $j, n = 1,2,3,4$; $\psi(x, y) = 1$ if *x* is same as *y*, otherwise, $\psi(x,y) = 0$; $\beta(x, y) = 9$ if *x* same as *y*, or else $\beta(x,y) = 0.3$.

(3) *One-hot Encoding (OHE).* A binary one-hot vector of dimension $L \times 4$ is used to represent an RNA sequence, where *L* is the length of the RNA sequence, and four corresponds to the number of nucleotide types, **i.e.,** A, U, C, and G. In detail, in one-hot encoding, each nucleotide in a sequence is denoted as one of four one-hot vectors [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], and [0, 0, 0, 1], the value of 1 corresponding to the nucleotide at that position and 0 elsewhere [20,21,37–39].

2.4. Deep multi-scale context feature learning architecture

Deep multi-scale context feature learning architecture consists of two-stage process of multi-scale context feature extraction and deep multi-shot neural network construction, where the flowchart is depicted in Fig. 1.

2.5. Multi-scale context feature extraction

Deep neural network can automatically learn high-level representation from sequence-based features, such as nucleotide composition and evolutionary information. Nevertheless, only the observed a single scale one-shot feature or feature information of limited spatial dimensions related to target nucleotides is not sufficient for training a high-performance model since exposure or burial of nucleotide should be affected by different factors. Recognizing this, in this study, a multi-scale sliding window technique and clipping algorithms are utilized to design a multi-scale context feature extraction strategy, which provides more discrimination contextual features at multiple spatial scales for target nucleotides. Fig. 1A illustrates the multi-scale context feature extraction strategy.

As shown in Fig. 1A, given an RNA sequence with *L* nucleotides, M^{2}pred generates the three different discriminative features, i.e., BPP ($L \times 1$), PSFM ($L \times 4$), and OHE ($L \times 4$), by calling the corresponding programs (refer to the section of "Feature Representation" for detail) and obtains a super feature, i.e., BPP + PSFM + OHE ($L \times 9$), by jointing these three types of features serially. Based on the super feature, a 2D initialization sliding window of size $\mathbb{R}^{\theta\times 9}$ (2D-θ-SW), is first used to transform 2D matrix-format BPP + PSFM + OHE into a cube-format feature of size $\mathbb{R}^{L\times\theta\times 9}$, called 3D-$\mathbb{R}^{L\times\theta\times 9}$. Subsequently, a 2D clipping algorithm of size $\mathbb{R}^{\theta\times 9}$ (2D-θ-C) and a 2D clipping algorithm of size $\mathbb{R}^{(\sqrt{\tau}\times\theta)\times(\sqrt{\tau}\times 9)}$ (2D-τ-C) are applied to 3D-$\mathbb{R}^{L\times\theta\times 9}$ and extract contextual features of two scales, i.e., a 2D minimal feature of size $\mathbb{R}^{\theta\times 9}$ and a 2D maximal feature of size $\mathbb{R}^{(\sqrt{\tau}\times\theta)\times(\sqrt{\tau}\times 9)}$, respectively, called 2D-$\mathbb{R}^{\theta\times 9}$ and 2D-$\mathbb{R}^{(\sqrt{\tau}\times\theta)\times(\sqrt{\tau}\times 9)}$. Furthermore, a 3D minimal sliding window of size $\mathbb{R}^{\varphi\times\theta\times 9}$ (3D-φ-SW) and a 3D maximal sliding window of size $\mathbb{R}^{\omega\times\theta\times 9}$ (3D-ω-SW) are also employed to extract contextual features of two scales, respectively, called 3D-$\mathbb{R}^{\varphi\times\theta\times 9}$ and 3D-$\mathbb{R}^{\omega\times\theta\times 9}$. Finally, the contextual feature of four scales mentioned above are regularly stacked into a contextual pyramid feature as input source of deep multi-shot neural network framework.

To search the optimal local 2D initialization sliding window hyper-parameter θ, 2D clipping window size hyper-parameter τ, 3D minimal sliding window hyper-parameter φ, and 3D maximal sliding window hyper-parameter ω, we use the strategy of grid search and adjust the hyper-parameter θ, τ, φ, and ω by observing a DL model performance based on a baseline single-scale contextual feature learning architecture (abbreviated as SSCFL) shown in Supplemental Fig. S1, on training dataset TR119 over five-fold cross-validation tests. Supplemental Fig. S2 demonstrates the performance variation curve of PCC versus sliding window size. By visiting Fig. S2, it is easy to find that, SSCFL gains better performance at θ of 25, τ of 9, φ of 9, and ω of 31, respectively (refer to the Supplemental **Text S1** and Figs. S1 and S2 for detail). Hence, the following values for the above hyper-parameters $\theta = 25$, $\tau = 9$, $\varphi = 9$, and $\omega = 31$ are adopted in this study.

2.6. Residual attention block

Capturing long-range dependencies knowledge of target nucleotides is central importance in computational biology. Although through repeating convolutional and recurrent operations can process long-range dependencies, repeatedly adding these operations may result in computationally inefficient and vanishing gradients. In this regard, the modified residual network block [40] embedded an attention module

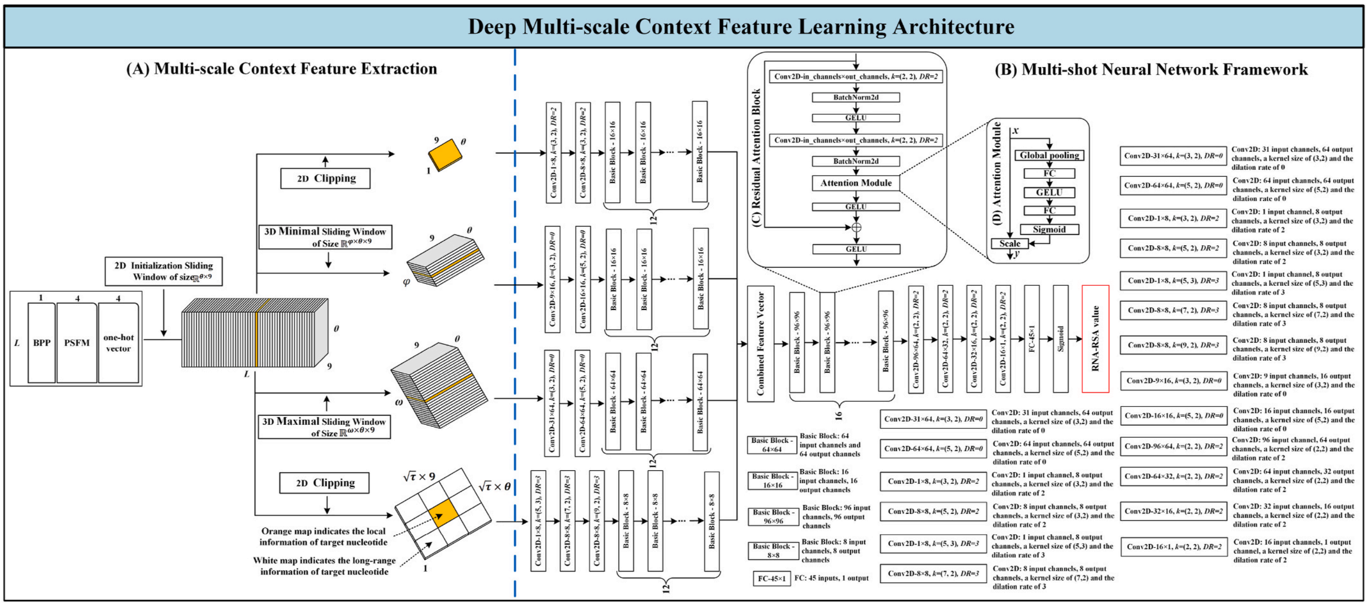
Deep Multi-scale Context Feature Learning Architecture
(A) Multi-scale Context Feature Extraction
L
BPP
PSFM
one-hot vector
1
4
4
2D Initialization Sliding Window of size $\mathbb{R}^{\theta\times 9}$
L
θ
9
2D Clipping
3D Minimal Sliding Window of Size $\mathbb{R}^{\varphi\times\theta\times 9}$
3D Maximal Sliding Window of Size $\mathbb{R}^{\omega\times\theta\times 9}$
2D Clipping
φ
ω
$\sqrt{\tau}\times 9$
$\sqrt{\tau}\times\theta$
Orange map indicates the local information of target nucleotide
White map indicates the long-range information of target nucleotide
Conv2D-1×8, k=(3, 2), DR=2
Conv2D-8×8, k=(3, 2), DR=2
Basic Block - 16×16
Conv2D-9×16, k=(3, 2), DR=0
Conv2D-16×16, k=(5, 2), DR=0
Basic Block - 16×16
Conv2D-31×64, k=(3, 2), DR=0
Conv2D-64×64, k=(5, 2), DR=0
Basic Block - 64×64
Conv2D-1×8, k=(5, 3), DR=3
Conv2D-8×8, k=(7, 2), DR=3
Conv2D-8×8, k=(9, 2), DR=3
Basic Block - 8×8
12
(C) Residual Attention Block
Conv2D-in_channels×out_channels, k=(2, 2), DR=2
BatchNorm2d
GELU
Conv2D-in_channels×out_channels, k=(2, 2), DR=2
BatchNorm2d
Attention Module
GELU
GELU
(D) Attention Module
x
Global pooling
FC
GELU
FC
Sigmoid
Scale
y
(B) Multi-shot Neural Network Framework
Combined Feature Vector
Basic Block - 96×96
16
Conv2D-96×64, k=(2, 2), DR=2
Conv2D-64×32, k=(2, 2), DR=2
Conv2D-32×16, k=(2, 2), DR=2
Conv2D-16×1, k=(2, 2), DR=2
FC-45×1
Sigmoid
RNA-RSA value
Basic Block - 64×64 | Basic Block: 64 input channels and 64 output channels
Basic Block - 16×16 | Basic Block: 16 input channels, 16 output channels
Basic Block - 96×96 | Basic Block: 96 input channels, 96 output channels
Basic Block - 8×8 | Basic Block: 8 input channels, 8 output channels
FC-45×1 | FC: 45 inputs, 1 output
Conv2D-31×64, k=(3, 2), DR=0 | Conv2D: 31 input channels, 64 output channels, a kernel size of (3,2) and the dilation rate of 0
Conv2D-64×64, k=(5, 2), DR=0 | Conv2D: 64 input channels, 64 output channels, a kernel size of (5,2) and the dilation rate of 0
Conv2D-1×8, k=(3, 2), DR=2 | Conv2D: 1 input channel, 8 output channels, a kernel size of (3,2) and the dilation rate of 2
Conv2D-8×8, k=(5, 2), DR=2 | Conv2D: 8 input channels, 8 output channels, a kernel size of (3,2) and the dilation rate of 2
Conv2D-1×8, k=(5, 3), DR=3 | Conv2D: 1 input channel, 8 output channels, a kernel size of (5,3) and the dilation rate of 3
Conv2D-8×8, k=(7, 2), DR=3 | Conv2D: 8 input channels, 8 output channels, a kernel size of (7,2) and the dilation rate of 3
Conv2D-31×64, k=(3, 2), DR=0 | Conv2D: 31 input channels, 64 output channels, a kernel size of (3,2) and the dilation rate of 0
Conv2D-64×64, k=(5, 2), DR=0 | Conv2D: 64 input channels, 64 output channels, a kernel size of (5,2) and the dilation rate of 0
Conv2D-1×8, k=(3, 2), DR=2 | Conv2D: 1 input channel, 8 output channels, a kernel size of (3,2) and the dilation rate of 2
Conv2D-8×8, k=(5, 2), DR=2 | Conv2D: 8 input channels, 8 output channels, a kernel size of (3,2) and the dilation rate of 2
Conv2D-1×8, k=(5, 3), DR=3 | Conv2D: 1 input channel, 8 output channels, a kernel size of (5,3) and the dilation rate of 3
Conv2D-8×8, k=(7, 2), DR=3 | Conv2D: 8 input channels, 8 output channels, a kernel size of (7,2) and the dilation rate of 3
Conv2D-8×8, k=(9, 2), DR=3 | Conv2D: 8 input channels, 8 output channels, a kernel size of (9,2) and the dilation rate of 3
Conv2D-9×16, k=(3, 2), DR=0 | Conv2D: 9 input channels, 16 output channels, a kernel size of (3,2) and the dilation rate of 0
Conv2D-16×16, k=(5, 2), DR=0 | Conv2D: 16 input channels, 16 output channels, a kernel size of (5,2) and the dilation rate of 0
Conv2D-96×64, k=(2, 2), DR=2 | Conv2D: 96 input channel, 64 output channels, a kernel size of (2,2) and the dilation rate of 2
Conv2D-64×32, k=(2, 2), DR=2 | Conv2D: 64 input channels, 32 output channels, a kernel size of (2,2) and the dilation rate of 2
Conv2D-32×16, k=(2, 2), DR=2 | Conv2D: 32 input channels, 16 output channels, a kernel size of (2,2) and the dilation rate of 2
Conv2D-16×1, k=(2, 2), DR=2 | Conv2D: 16 input channels, 1 output channel, a kernel size of (2,2) and the dilation rate of 2

Fig. 1. Architecture of deep multi-scale context feature learning. (A) Multi-scale Context Feature Extraction; (B) Multi-shot Neural Network Framework; (C) Residual Attention Block; (D) Attention Module.

[41], called "residual attention network" [42], which not only focuses on digging both long-distance and local intra-sequence dependencies, but also guarantees the extraction of the key position knowledge of the target nucleotides, is employed to dig out more discriminative information. Moreover, we use the dilated convolutional neural network [22, 23] having kernel size of (2, 2) and the dilation rate (*DR*) of 2 instead of the plain convolutional in the traditional residual network [40] to expand the perception of the network. Each dilated convolutional layer is followed by the batch normalization layer [43], the gaussian error linear units (GELU) activation function, and a dropout strategy with a ratio of 10% [44]. It is shown in Fig. 1C and D that implementation of the *l*th residual attention block and attention module can be expressed as follow, respectively:

$$u_{l+1} = f(u_l + \mathscr{F}(u_l, \mathscr{W}_l)) \tag{2}$$

$$y_c = \sigma\left[\mathscr{W}_2 \times f\left(\mathscr{W}_1 \times \frac{1}{W\times H}\sum_{i=1}^{W}\times\sum_{j=1}^{H} x_c(i,j)\right)\right] \times x_c\left(i,j\right),\ x_c\left(i,j\right) \in \mathbb{R}^{W\times H\times C} \tag{3}$$

where u_l and u_{l+1} denote the input and output of the *l*th residual attention block, respectively; $\mathscr{W}_l$ is a set of weights in the *l*th residual attention block, which contains the weights of two dilated convolutional layers and two batch normalization layers. *F* is the activation function GELU, while $\mathscr{F}$ stands for the residual function. Where $x_c(i,j)$ and y_c denote the input and output of the attention module, respectively; $\mathscr{W}_1$ and $\mathscr{W}_2$ is the weights of two fully-connected hidden layers (FC) [45], respectively; σ is the sigmoid function [46].

2.7. Deep multi-shot neural network

In this study, the solvent accessibility prediction of RNA is taken as a regression problem. Leveraging the power of the residual attention network and multi-scale context feature extraction scheme, we design and implement the custom-made deep multi-shot neural network framework (MSNN) to solve this problem. As shown in Fig. 1B, the proposed MSNN framework, which is equipped with four receptive fields corresponding to four scales spatial feature of the contextual pyramid feature, i.e., 2D-$\mathbb{R}^{\theta\times 9}$, 2D-$\mathbb{R}^{(\sqrt{\varphi}\times\theta)\times(\sqrt{\varphi}\times 9)}$, 3D-$\mathbb{R}^{\varphi\times\theta\times 9}$, and 3D-$\mathbb{R}^{\omega\times\theta\times 9}$, is constructed with initial dilated convolutional layers followed by residual attention blocks, fully-connected hidden layers, and a sigmoid activation function.

In MSNN, the input of each receptive field is one single-scale contextual feature map in contextual pyramid feature backbone. To extract the discriminative contextual knowledge, for each receptive field, N_1 initial dilated convolutional layers, which are used before a group of N_2 residual attention basic blocks, are first employed to transform the input feature maps into a spatial vector with a larger signal channel. Each initial dilated convolutional layer, which is followed by the batch normalization layer, the GELU activation function, and the max-poling layer for further down-sampling, is configured a larger size convolution *K* to preserver as much of the original input information as possible. Besides, a dropout strategy with a ratio of *d*% is utilized to reduce network overfitting during training. Subsequently, the spatial vector is fed into a group of N_2 residual attention basic blocks. Then, to capture the fusion information of the contextual pyramid feature, four transformed features of four receptive fields are coalesced by using scale-transfer operations. The fusion feature maps are entered into a group of N_3 residual attention basic block and N_4 dilated convolutional layers. Here, the batch normalization layer, the GELU activation function, and a dropout ratio of *d*% are again utilized. Finally, the RSA value of each nucleotide is calculated by N_5-layer fully-connected hidden layers with *h* hidden units and a sigmoid activation function.

MSNN framework, which is implemented using Pytorch software (version 1.3.1) [47], is trained on one NVIDIA TITAN X graphics processing unit (GPU) to speed up training. In the model training process, we use the mean squared error function to calculate the loss and optimized the model by the Adam algorithm [48] with a learning rate of *lr* and a batch size of *bs*. In this study, we use the strategy of grid search and adjust the network's hyper-parameters, i.e., N_1, N_2, N_3, N_4, N_5, *d*, *K*, and *h*, by observing the model performance on the training dataset TR119 over 5-fold cross-validation tests. Finally, according the best performance of MSNN model, we use the following values for the above hyper-parameters: $N_1 = 2$ and 3, $N_2 = 12$, $N_3 = 16$, $N_4 = 4$, $N_5 = 1$, $d = 10$, $K = \{(3, 2), (5, 2), (5, 3), (7, 2), (9, 2)\}$, $h = 45$, $rl = 0.001$, and $bs = 2000$.

3. Results and discussion

3.1. Performance comparison between different features

This section examines to what extent the three sequence-based features and their combined features can help to predict RNA solvent accessibility. Specifically, for efficient and time-saving selection of optimal feature combination, three sequence-based features, BPP, PSFM, and OHE, and two separate serial feature combinations, BPP + PSFM and BPP + PSFM + OHE, are used as the inputs to the multilayer perceptron (MLP) algorithm [49] (see Supplemental Fig. S3), and the performance of each feature is investigated. "+" means simple serial combination of different sequence-based features. Furthermore, inspired by the work of RNAsol [20], each nucleotide is extracted more discriminative feature from each feature using a sliding window of size 10. Each feature is evaluated by performing five-fold cross-validation on the training dataset TR119. Table 1 summarizes the discriminative performance comparison between the five features on TR119 over five-fold cross-validation.

From Table 1, we observe that the BPP + PSFM + OHE feature consistently outperforms other four features in terms of two evaluation indexes, i.e., PCC and MAE. Concretely, the PCC and MAE of BPP + PSFM + OHE are 0.35 and 38.76, which are improvement of 9.3% and 0.9%, respectively, over the second-best feature, i.e., BPP + PSFM. These experimental results demonstrate that the three single-view features contain complementary information.

3.2. Comparison to state-of-the-art methods

The categories of the methods mentioned in the introduction section can be generally categorized into two major groups, i.e., ML-based methods and DL-based methods. The purpose of this section is to experimentally demonstrate the efficacy of the proposed M^{2}pred by comparing it with both ML-based methods (i.e., RNAsnap-seq and RNAsnap-prof) and DL-based methods (i.e., RNAsol, RNAsnap2, and RNAsnap2(SingleSeq)).

3.3. Comparison to ML-based models

This study compared the performance of the two ML-based methods, i.e., RNAsnap-seq [14] and RNAsnap-prof [14], which are trained on one dataset contained 89 non-redundant protein-bound RNAs (TR89). Note that, to make the comparison as fair as possible, in this section, our proposed M^{2}pred learns the prediction model with the same training dataset of RNAsnap-seq and RNAsnap-prof, i.e., TR89. Subsequently, the model is assessed on the independent testing datasets of RNAsnap-seq and RNAsnap-prof, i.e., TS44, CN48, and TEST36. Here, the strategy of a grid search is again used to adjust the network's hyper-parameter by observing the model performance on the training dataset TR89 over five-fold cross-validation tests. To obtain the prediction results on TS44, CN48, and TEST36, standalone packages of RNAsnap-seq and RNAsnap-prof are downloaded at https://servers.sparks-lab.org/downloads/RNAsnap.tgz.

Table 1
Performance comparison of different features on TR119 over five-fold cross-validation tests using the SVM algorithm.

Feature Type	PCC		MAE	
	value	*p*-value	value	*p*-value
BPP	0.12	3.1×10^{-3}	46.32	8.3×10^{-4}
PSFM	0.21	2.4×10^{-2}	42.57	5.7×10^{-2}
OHE	0.24	6.8×10^{-2}	43.65	1.8×10^{-3}
BPP + PSFM	0.32	7.1×10^{-1}	39.12	1.1×10^{-1}
BPP + PSFM + OHE	0.35		38.76	

The *p*-values in Student's *t*-test are calculated for the differences between BPP + PSFM + OHE and other features.

Table 2 demonstrates the performance comparison of M^{2}pred, RNAsnap-seq, and RNAsnap-prof on the datasets, i.e., TS44, CN48, and TEST36, over independent validation tests. **Supplementary** Table S3 lists the *p*-values in Wilcoxon rank-sum test for the differences in predicted RSA values between the three methods on test datasets, i.e., TS44, CN48, and TEST36. From Table 2, it is clear that compared with the ML-based models, the proposed M^{2}pred performs best in terms of the PCC and MAE values on TS44, CN48, and TEST36. Specifically, M^{2}pred achieves 74.3% and 11.9% average improvement in PCC and MAE on the three testing datasets, compared with the better performer of RNAsnap-seq. Taking results on CN48 as an example, the PCC and MAE of M^{2}pred are 0.50 and 30.62, which are 108.3% and 16.5%, 117.3% and 15.7% higher, respectively, than RNAsnap-seq and RNAsnap-prof, with p-values $<10^{-5}$.

Fig. 2 illustrates the head-to-head comparisons between M^{2}pred and the two ML-based methods based on PCC and MAE values on the union dataset of TS44, CN48, and TEST36, which contains 128 independent test RNA targets. Out of the 128 targets, there are 107 and 98, 115 and 101 cases where M^{2}pred has better PCC and MAE values than RNAsnap-seq and RNAsnap-prof, respectively. As expect, a low Pearson's correlation coefficient (PCC$^+$) is observed between the PCC values of M^{2}pred and those of all comparison methods on the union dataset of TS44, CN48, and TEST36. This indicates that there is significant difference between these three methods.

3.4. Comparison to DL-based models

The compared DL-based methods of three control methods include RNAsol [20], RNAsnap2 [21], and RNAsnap2(SingleSeq) [21], and all the prediction models are learned on the training dataset TR119. Note that, in RNAsnap2 [21] and RNAsnap2(SingleSeq) [21], there are 24 RNAs are randomly selected from TR119 to tune the hyper-parameters. In this section, the proposed M^{2}pred uses the same training dataset as RNAsol, RNAsnap2, and RNAsnap2(SingleSeq), i.e., TR119, to learn prediction model. Here, we use the strategy of a grid search and adjust the network's hyper-parameter by observing the model performance on the training dataset TR119 over 5-fold cross-validation tests. For an objective and fair comparison, the standalone packages of RNAsol, RNAsnap2, and RNAsnap2(SingleSeq) are first downloaded from https://yanglab.nankai.edu.cn/RNAsol/and https://github.com/jaswindersingh2/RNAsnap2/, and installed locally. All RNA sequences in TS45, TS31, and TEST36 are then fed into the standalone program of these methods to calculate the solvent accessibility of RNA. Note that, M^{2}pred, RNAsol, and RNAsnap2 utilize the same NCBI's reference database to generate evolutionary features. Table 3 provides a comparison of the predictive performance between the proposed M^{2}pred, RNAsol, RNAsnap2, and RNAsnap2(SingleSeq) on the independent test

Table 2
Performance comparison between M^{2}pred and other state-of-the-art ML-based RSA prediction methods on three independent test datasets, i.e., TS44, CN48, and TEST36.

Dataset	Method	PCC		MAE	
		value	*p*-value	value	*p*-value
TS44	RNAsnap-seq	0.36	1.8×10^{-3}	39.04	1.6×10^{-1}
	RNAsnap-prof	0.38	4.2×10^{-4}	40.22	3.3×10^{-1}
	M^{2}pred	0.51		36.15	
CN48	RNAsnap-seq	0.24	1.6×10^{-12}	36.70	7.3×10^{-5}
	RNAsnap-prof	0.23	5.7×10^{-13}	36.29	5.4×10^{-5}
	M^{2}pred	0.50		30.62	
TEST36	RNAsnap-seq	0.26	7.1×10^{-6}	42.23	3.7×10^{-2}
	RNAsnap-prof	0.25	1.3×10^{-6}	41.64	5.7×10^{-2}
	M^{2}pred	0.45		37.21	

The *p*-values in Student's *t*-test are calculated for the differences between M^{2}pred and other control methods.

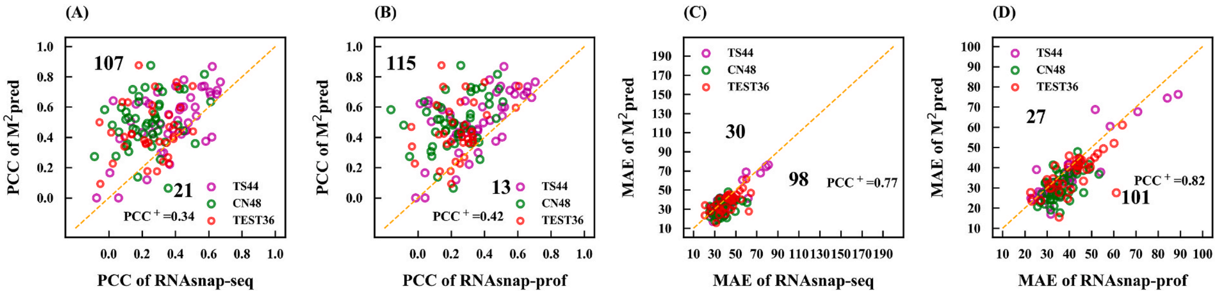

Fig. 2. Head-to-head comparisons of PCC and MAE between M^{2}pred and other ML-based methods on the union set of TS44, CN48, and TEST36. PCC$^+$ is the Pearson's correlation coefficient between the PCC or MAE values of the two compared methods. Each purple, green, and red circle mean one RNA in the TS44, CN48, and TEST36, respectively. The numbers in each panel represent the number of points in the upper and lower triangles, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 3
Performance comparison between M^{2}pred and other state-of-the-art DL-based RSA prediction methods on TS45, TS31, and TEST36.

Dataset	Method	PCC		MAE	
		value	*p*-value	value	*p*-value
TS45	RNAsol	0.48	6.9×10^{-2}	35.49	2.6×10^{-2}
	RNAsnap2	0.54	3.4×10^{-1}	33.37	3.7×10^{-1}
	RNAsnap2(SingleSeq)	0.50	7.3×10^{-2}	33.91	1.7×10^{-1}
	M^{2}pred	0.58		31.07	
TS31	RNAsol	0.41	4.1×10^{-1}	36.36	1.7×10^{-1}
	RNAsnap2	0.51	9.6×10^{-1}	32.65	4.8×10^{-1}
	RNAsnap2(SingleSeq)	0.48	6.6×10^{-1}	33.53	3.2×10^{-1}
	M^{2}pred	0.52		31.42	
TEST36	RNAsol	0.44	4.9×10^{-1}	38.62	3.1×10^{-1}
	RNAsnap2	0.48	9.1×10^{-1}	36.11	9.4×10^{-1}
	RNAsnap2(SingleSeq)	0.47	9.3×10^{-1}	36.70	8.4×10^{-1}
	M^{2}pred	0.48		36.26	

The *p*-values in Student's *t*-test are calculated for the differences between M^{2}pred and other control methods.

datasets, i.e., TS45, TS31, and TEST36. **Supplementary** Table S4 summarizes the *p*-values of the Wilcoxon rank-sum test for differences in predicted RSA values between the four methods on TS45, TS31, and TEST36.

As shown in Table 3, according to the PCC and MAE average, which are two overall measurements of the quality of prediction performance, we can find that the M^{2}pred acts as the best performer followed by RNAsnap2, RNAsnap2(SingleSeq), and RNAsol on three independent test datasets. Taking results on TS45 as an example, the PCC and MAE values of M^{2}pred are 0.58 and 31.07, which are 7.4% and 5.3%, 16.0% and 8.4%, and 28.9% and 14.3% higher than those of RNAsnap2, RNAsnap2(SingleSeq), and RNAsol, respectively. In particular, M^{2}pred achieves the highest PCC values and it is the sole method with PCC values larger than 0.52 on TS45 and TS31. By carefully observing Table 3, it is found that the differences between M^{2}pred and other existing DL-based methods in the PCC and MAE values are not consistently statistically significant. Nonetheless, from the viewpoint of the *p*-values on Wilcoxon rank-sum test for differences in predicted RSA values between four methods listed in **Supplementary** Table S4, the *p*-values smaller than 10^{-58} indicate the distribution of predicted RSA values between M^{2}pred and other DL-based methods are significantly different.

By visiting Fig. 3, it is easy to find that, among the 112 target RNAs in the union dataset of TS45, TS31, and TEST36, M^{2}pred has 71 and 78, 58 and 63, and 61 and 67 cases with better PCC and MAE values, respectively, than RNAsol, RNAsnap2, RNAsnap2(SingleSeq). It is worthwhile mentioning that, there is a low Pearson's correlation coefficient (PCC$^+$) between the proposed M^{2}pred and RNAsol (0.56), RNAsnap2 (0.66), and RNAsnap2(SingleSeq) (0.63) on the PCC evaluation index. However, M^{2}pred has a high correlation with RNAsnap2 (0.79) and RNAsnap2 (SingleSeq) (0.77) on the MAE evaluation index. We speculate that due to the three DL-based methods use similar feature representation (i.e., one-hot encoding) and the same training dataset (i.e., TR119).

To understand the influence of the quality of the MSA on the prediction performance of M^{2}pred, RNAsnap2(SingleSeq), and RNAsnap2, we have employed the overall evaluation index, i.e., PCC, to measure the prediction performance of each RNA in the union dataset of TS45, TS31, and TEST36. **Supplementary** Text 3 demonstrates the relationship between the values of PCC and the number of effective homologous sequences (N_{eff}) [50]. We presume that the performance of M^{2}pred, RNAsnap2, RNAsnap2(SingleSeq) will be greatly affected if the amount of information in MSA is insufficient; when the amount of MSA information reaches or falls below a certain level, the dependence between the increase in MSA information and the performance improvement of these methods becomes less.

3.5. Performance comparison on both protein-bound and protein-free RNAs

To further investigate the highlights of the proposed M^{2}pred, Tables 2 and 4**,** and Supplementary Tables S3 and S5 demonstrate the performance comparison between M^{2}pred and other existing state-of-the-art ML-based methods (i.e., RNAsnap-seq and RNAsnap-prof) and DL-based methods (i.e., RNAsol, RNAsnap2, and RNAsnap2(SingleSeq)), on the protein-bound and protein-free RNAs, respectively. There are 37 protein-bound RNAs and 8 protein-free RNAs in TS45, 31 protein-free RNA in TS31, 44 protein-bound RNAs in TS44, and 48 protein-free RNAs in CN48, 5 protein-bound RNAs and 31 protein-free RNAs in TEST36. For the convenience of comparison with DL-based methods on the protein-bound and protein-free RNAs, the above 42 protein-bound and 70 protein-free RNAs in the union set of TS45, TS31, and TEST36 are separately collected to composite a protein-bound independent test dataset, called PB42 and a protein-free independent test dataset, called PF70.

From Tables 2 and 4, S3, and S5, we can observe that M^{2}pred is consistently superior to both ML-based methods and DL-based methods concerning the two evaluation indexes, i.e., PCC and MAE, on protein-bound and protein-free datasets, i.e., TS44, CN48, PB42, and PF70. Taking results in DL-based methods as an example, the PCC and MAE values of M^{2}pred are greater than 0.50 and smaller than 33.10, respectively, for both two datasets, which outperform other three DL-based methods, i.e., RNAsol, RNAsnap2, and RNAsnap2(SingleSeq). More specifically, compared with the second-best method, M^{2}pred achieves 7.5% and 0.6% improvements in PCC and MAE on protein-bound dataset, i.e., PB42, respectively. Moreover, by revisiting Tables S3 and S5, it should be pointed out that, although the *p*-values between M^{2}pred and other methods in PCC and MAE evaluation indexes are both larger than 0.05 on PB42 and PF70, the *p*-values on Wilcoxon rank-sum test for the difference in predicted RSA values between the four methods on PB42 and PF70 are both less than 10^{-87}, which indicate

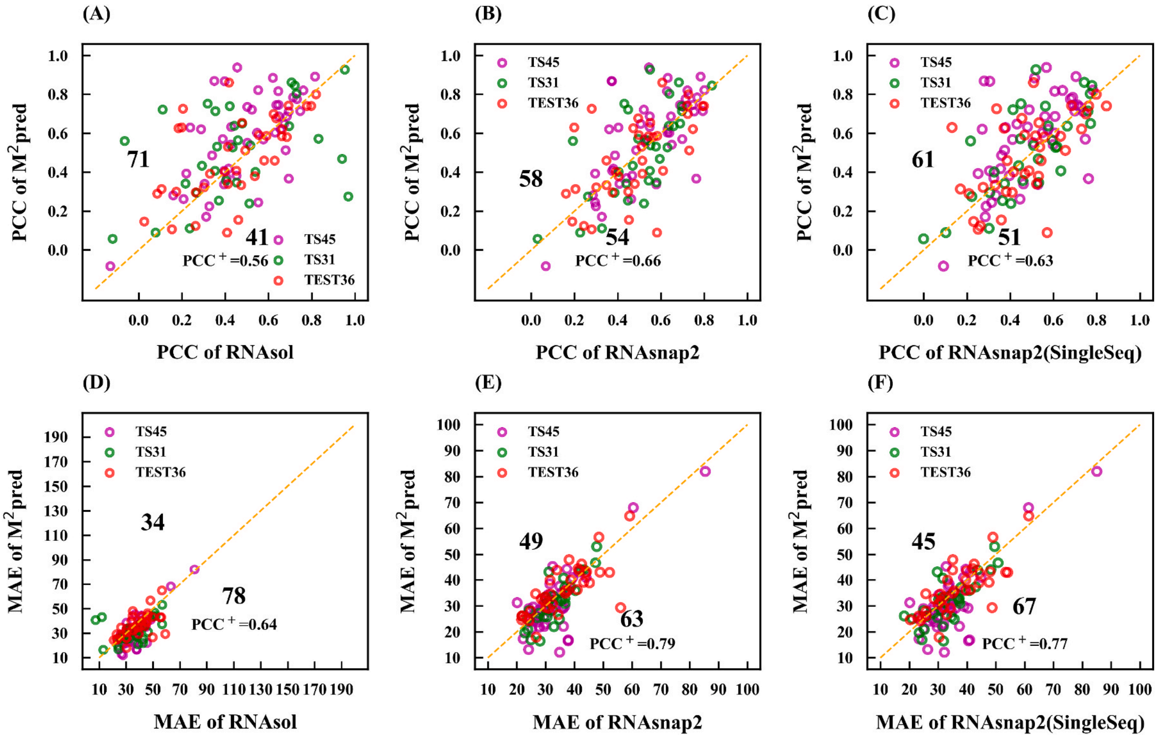

Fig. 3. Head-to-head comparisons of PCC and MAE between M^{2}pred and other DL-based methods on the union set of TS45, TS31, and TEST36. PCC$^+$ is the Pearson's correlation coefficient between the PCC or MAE values of the two compared methods. Each purple, green, and red circle mean one RNA in the TS45, TS31, and TEST36, respectively. The numbers in each panel represent the number of points in the upper and lower triangles, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 4
Performance comparison between M^{2}pred and other state-of-the-art DL-based solvent accessibility prediction methods on protein-bound and protein-free datasets, i.e., PB42 and PF70.

Dataset	Method	PCC		MAE	
		value	*p*-value	value	*p*-value
PB42	RNAsol	0.50	8.5×10^{-2}	37.60	9.6×10^{-2}
	RNAsnap2	0.53	2.8×10^{-1}	35.07	4.7×10^{-1}
	RNAsnap2(SingleSeq)	0.49	5.2×10^{-2}	36.46	2.2×10^{-1}
	M^{2}pred	0.57		33.10	
PF70	RNAsol	0.45	2.4×10^{-1}	36.26	3.0×10^{-2}
	RNAsnap2	0.50	8.1×10^{-1}	33.45	5.7×10^{-1}
	RNAsnap2(SingleSeq)	0.48	7.9×10^{-1}	34.01	3.4×10^{-1}
	M^{2}pred	0.50		32.67	

The *p*-values in Student's *t*-test are calculated for the differences between M^{2}pred and other control methods.

the distribution of predicted RSA values between M^{2}pred and other methods are significant statistically.

Although M^{2}pred use fused multi-view features to represent the information contained in the RNA sequence, in most of the cases it introduces redundant or irrelevant information inevitably that will seriously reduce the efficiency of RSA prediction model. Hence, eliminating noise in the feature is also an important step in the process of RSA identification. Furthermore, the influence of RNA sequence features on RSA prediction is not fully elucidated. It is still improved in RSA prediction by extracting features based on RNA sequences.

3.6. Case studies

Two RNA sequences, 3k0j_E and 6p2h_A, selected from the two independent test sets, i.e., TS44 and TS31, respectively, are used for case studies. Here, 3k0j_E is employed to compare with traditional ML-based methods, while 6p2h_A is used for comparison with DL-based methods. The detailed comparisons between predicted versus actual RSA values on 3k0j_E and 6p2h_A are separately shown in **Supplementary** Figs. S4A and B. The actual RSA values calculated by **POPS** program are fitted based on the experimental 3D structures of the two RNAs.

By visiting Figs. S4A and B, it is easily found that M^{2}pred consistently outperforms other existing state-or-the-art methods, i.e., RNAsnap-seq, RNAsnap-prof, RNAsol, RNAsnap2, and RNAsnap2(SingleSeq), on both two cases. Concretely, on 3k0j_E and 6p2h_A, M^{2}pred achieves the highest PCC values (i.e., 0.74 and 0.75) and the best MAE values (i.e., 25.76 and 24.88). From Fig. S4, we can also find that, compared with other state-of-the-art methods, the predicted values of M^{2}pred are more similar to the actual values of 3k0j_E and 6p2h_A. In addition, Table 5 demonstrates the performance and running time comparison between M^{2}pred and other state-of-the-art solvent accessibility prediction methods on 3k0j_E and 6p2h_A. Note that, in order to make the comparison as fair as possible, the running time of M2pred and the control methods are evaluated on the same computational device (Intel(R) Core (TM) i9-10920X CPU @3.50 GHz, 64.0 GB of RAM, NVIDIA GeForce RTX3090 24.0 GB). From Table 5, compared with traditional machine

Table 5
The performance and inference time comparison between M^{2}pred and other state-of-the-art solvent accessibility prediction methods on 3k0j_E and 6p2h_A.

RNA name	Method	PCC	MAE	Inference time (minute)
3k0j_E (87 bases)	RNAsnap-seq	0.40	37.21	0.021
	RNAsnap-prof	0.50	36.06	1.25
	M2pred	0.74	25.76	53.30
6p2h_A (69 bases)	RNAsol	0.32	42.80	115.39
	RNAsnap2	0.43	34.75	35.91
	RNAsnap2 (SingleSeq)	0.45	35.61	0.063
	M^{2}pred	0.75	24.88	41.58

learning-based methods, i.e., RNAsnap-seq and RNAsnap-prof, the performance of our proposed M^{2}pred is significantly improved, despite the longer running time of M^{2}pred. Compared with three deep learning-based methods, i.e., RNAsol, RNAsnap2, and RNAsnap2(SingleSeq), the running time of M^{2}pred is slightly higher than that of RNAsnap2, but M^{2}pred achieves 74.4% and 28.4% average improvements of PCC and MAE, respectively, on 6p2h_A. We also count the number of model parameters of RNAsol, RNAsnap2, and M2pred. The parameter number of the proposed M2pred is 1,771,337, which is less than that of RNAsol, but larger than that of RNAsnap2 (refer to the Supplemental **Text S2** for detail). We tried to count the network parameters of RNAsnap, but the SVM-based RNAsnap uses the Kernel technique, and it is difficult to compute its network parameters directly. Thus, we did not compare with RNAsnap.

4. Conclusions

Accurate prediction of RNA solvent accessibility is one of the most important tasks in the annotation of RNA functions. In order to enhance the prediction performance of RNA solvent accessibility, in this study, we have designed and implemented a novel deep learning-based approach, named M2pred. Benchmarking experiments show that the performance of M2pred is superior to other existing state-of-the-art ML- and DL-based methods, i.e., RNAsnap-seq, RNAsnap-prof, RNAsol, RNAsnap2, and RNAsnap2(SingleSeq). The characteristics of this approach are summarized as follows: First, a new multi-scale contextual feature extraction strategy is designed to provide more discriminative feature of target nucleotides. Second, a modified residual attention network is used to effectively dig both long-distance and local intra-sequence dependencies, but also guarantees the extraction of high-level knowledge related to the target nucleotides. Furthermore, the proposed method M2pred is trained by the designed deep learning-based pipeline, which effectively learns the solvent accessibility knowledge buried in sequence-based features. For easy to use, the standalone package of M2pred could be also downloaded at https://github.com/XueQiangFan/M2pred/.

Although M2pred has achieved a good performance in predicting RNA solvent accessibility, it has room for further improvement. There are some other important aspects that may be improved, which include: (1) developing more accurate prediction models to predict the related feature source information, such as effective evolutionary information based on multiple sequence alignment, RNA secondary structure, and protein-RNA recognition; (2) employing the powerful deep learning algorithm, to obtain the available information extracted from the original feature representation; (3) developing a more accurate method by combining M2pred and other state-of-the-art RNA solvent accessibility prediction methods. Finally, we believe that M2pred will be exploited as a useful tool to speed up the progress of RNA function detection and understanding.

Author contributions

X.Q.F, J.H., D.J.Y. and G.Z. designed research; X.Q.F. and J.H. performed research; Y.X.T and N.X.J. analyzed data; and X.Q.F. and J.H. wrote the paper.

Declaration of competing interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61902352, 62072243, 61772273, and 61773346), the Natural Science Foundation of Zhejiang (No. aLY21F020025 nd LZ20F030002), the Natural Science Foundation of Jiangsu (No. BK20201304), the Foundation of National Defense Key Laboratory of Science and Technology (JZX7Y202001SY000901), and the Fundamental Research Funds for the Provincial Universities of Zhejiang (No. RF-A20200012). J Hu, D.J. Yu, and G.J. Zhang are the corresponding authors for this paper.

Appendix A. Supplementary data

Supplementary data related to this article can be found online at https://doi.org/10.1016/j.ab.2022.114802.

References

[1] P. Ramaswamy, S.A. Woodson, S16 throws a conformational switch during assembly of 30S 5' domain, Nat. Struct. Mol. Biol. 16 (4) (Apr, 2009) 438–445, https://doi.org/10.1038/nsmb.1585.
[2] P.D. Carlson, M.E. Evans, A.M. Yu, E.J. Strobel, J.B. Lucks, SnapShot: RNA structure probing technologies, Cell 175 (2) (Oct 4, 2018), https://doi.org/10.1016/j.cell.2018.09.024, 600-600.e1.
[3] S. Mukherjee, R.P. Bahadur, An account of solvent accessibility in protein-RNA recognition, Sci. Rep. 8 (1) (Jul 12, 2018) 10546, https://doi.org/10.1038/s41598-018-28373-2.
[4] C. Hartlmüller, J.C. Günther, A.C. Wolter, J. Wöhnert, M. Sattler, T. Madl, RNA structure refinement using NMR solvent accessibility data, Sci. Rep. 7 (1) (Jul 14, 2017) 5393, https://doi.org/10.1038/s41598-017-05821-z.
[5] C. Jegousse, Y. Yang, J. Zhan, J. Wang, Y. Zhou, Structural signatures of thermal adaptation of bacterial ribosomal RNA, transfer RNA, and messenger RNA, PLoS One 12 (9) (2017), e0184722, https://doi.org/10.1371/journal.pone.0184722.
[6] Y. Wan, M. Kertesz, R.C. Spitale, E. Segal, H.Y. Chang, Understanding the transcriptome through RNA structure, Nat. Rev. Genet. 12 (9) (Aug 18, 2011) 641–655, https://doi.org/10.1038/nrg3049.
[7] S.A. Mortimer, M.A. Kidwell, J.A. Doudna, Insights into RNA structure and function from genome-wide studies, Nat. Rev. Genet. 15 (7) (Jul, 2014) 469–479, https://doi.org/10.1038/nrg3681.
[8] C. Feng, D. Chan, J. Joseph, M. Muuronen, W.H. Coldren, N. Dai, I.R. Corrêa Jr., F. Furche, C.M. Hadad, R.C. Spitale, Light-activated chemical probing of nucleobase solvent accessibility inside cells, Nat. Chem. Biol. 14 (3) (Feb 14, 2018) 325, https://doi.org/10.1038/nchembio0318-325.
[9] B. Felden, RNA structure: experimental analysis, Curr. Opin. Microbiol. 10 (3) (Jun, 2007) 286–291, https://doi.org/10.1016/j.mib.2007.05.001.
[10] B.M. Muñoz-Flores, R. Santillán, N. Farfán, V. Álvarez-Venicio, V.M. Jiménez-Pérez, M. Rodríguez, O.G. Morales-Saavedra, P.G. Lacroix, C. Lepetit, K. Nakatani, Synthesis, X-ray diffraction analysis and nonlinear optical properties of hexacoordinated organotin compounds derived from Schiff bases, J. Organomet. Chem. 769 (2014) 64–71.
[11] L.G. Scott, M. Hennig, RNA structure determination by NMR, Methods Mol. Biol. 452 (2008) 29–61, https://doi.org/10.1007/978-1-60327-159-2_2.
[12] J.A. Latham, T.R. Cech, Defining the inside and outside of a catalytic RNA molecule, Science 245 (4915) (Jul 21, 1989) 276–282, https://doi.org/10.1126/science.2501870.
[13] L.J. Kielpinski, J. Vinther, Massive parallel-sequencing-based hydroxyl radical probing of RNA accessibility, Nucleic Acids Res. 42 (8) (Apr, 2014), https://doi.org/10.1093/nar/gku167 e70.
[14] Y. Yang, X. Li, H. Zhao, J. Zhan, J. Wang, Y. Zhou, Genome-scale characterization of RNA tertiary structures and their functional impact by RNA solvent accessibility prediction, RNA 23 (1) (2017) 14–22.
[15] A.H. Muhammad Rafid, M. Toufikuzzaman, M.S. Rahman, M.S. Rahman, CRISPRpred(SEQ): a sequence-based method for sgRNA on target activity prediction using traditional machine learning, BMC Bioinf. 21 (1) (Jun 1, 2020) 223, https://doi.org/10.1186/s12859-020-3531-9.
[16] H. Wei, B. Wang, J. Yang, J. Gao, RNA flexibility prediction with sequence profile and predicted solvent accessibility, IEEE ACM Trans. Comput. Biol. Bioinf (Nov 28, 2019), https://doi.org/10.1109/TCBB.2019.2956496.
[17] S. Yin, X. Tian, J. Zhang, P. Sun, G. Li, PCirc: random forest-based plant circRNA identification software, BMC Bioinf. 22 (1) (Jan 6, 2021) 10, https://doi.org/10.1186/s12859-020-03944-1.
[18] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (Nov 15, 1997), https://doi.org/10.1162/neco.1997.9.8.1735, 1735-80.
[19] E.P. Nawrocki, S.R. Eddy, Infernal 1.1: 100-fold faster RNA homology searches, Bioinformatics 29 (22) (2013) 2933–2935.
[20] S. Sun, Q. Wu, Z. Peng, J. Yang, Enhanced prediction of RNA solvent accessibility with long short-term memory neural networks and improved sequence profiles, Bioinformatics 35 (10) (2019) 1686–1691.
[21] A.K. Hanumanthappa, J. Singh, K. Paliwal, J. Singh, Y. Zhou, Single-sequence and profile-based prediction of RNA solvent accessibility using dilated convolutional neural network, Bioinformatics (Oct 27, 2020), https://doi.org/10.1093/bioinformatics/btaa652.
[22] R.S. Roy, F. Quadir, E. Soltanikazemi, J. Cheng, A deep dilated convolutional residual network for predicting interchain contacts of protein homodimers, Bioinformatics 38 (7) (Feb 4, 2022), https://doi.org/10.1093/bioinformatics/btac063, 1904-10.

[23] A.K. Hanumanthappa, J. Singh, K. Paliwal, J. Singh, Y. Zhou, Single-sequence and profile-based prediction of RNA solvent accessibility using dilated convolutional neural network, Bioinformatics 36 (21) (2021) 5169–5176, Jan 29, https://doi.org/10.1093/bioinformatics/btaa652.

[24] W. Li, A. Godzik, Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences, Bioinformatics 22 (13) (Jul 1, 2006), https://doi.org/10.1093/bioinformatics/btl158, 1658-9.

[25] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, Basic local alignment search tool, J. Mol. Biol. 215 (3) (Oct 5, 1990) 403–410, https://doi.org/10.1016/s0022-2836(05)80360-2.

[26] J. Singh, K. Paliwal, T. Zhang, J. Singh, T. Litfin, Y. Zhou, Improved RNA secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning, Bioinformatics (Mar 11, 2021), https://doi.org/10.1093/bioinformatics/btab165.

[27] S.F. Altschul, T.L. Madden, A.A. Schäffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (17) (Sep 1, 1997) 3389–3402, https://doi.org/10.1093/nar/25.17.3389.

[28] L. Cavallo, J. Kleinjung, F. Fraternali, POPS: a fast algorithm for solvent accessible surface areas at atomic and residue level, Nucleic Acids Res. 31 (13) (Jul 1, 2003), https://doi.org/10.1093/nar/gkg601, 3364-6.

[29] P.J. Cock, T. Antao, J.T. Chang, B.A. Chapman, C.J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, M.J. de Hoon, Biopython: freely available Python tools for computational molecular biology and bioinformatics, Bioinformatics 25 (11) (Jun 1, 2009), https://doi.org/10.1093/bioinformatics/btp163, 1422-3.

[30] G.D. Ruxton, The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test, Behav. Ecol. 17 (4) (2006) 688–690.

[31] J. Cuzick, A Wilcoxon-type test for trend, Stat. Med. 4 (1) (1985) 87–90.

[32] R. Adamczak, A. Porollo, J. Meller, Combining prediction of secondary structure and solvent accessibility in proteins, Proteins 59 (3) (May 15, 2005) 467–475, https://doi.org/10.1002/prot.20441.

[33] N. Goldman, J.L. Thorne, D.T. Jones, Assessing the impact of secondary structure and solvent accessibility on protein evolution, Genetics 149 (1) (May, 1998) 445–458, https://doi.org/10.1093/genetics/149.1.445.

[34] A. Garg, H. Kaur, G.P. Raghava, Real value prediction of solvent accessibility in proteins using multiple sequence alignment and secondary structure, Proteins 61 (2) (Nov 1, 2005) 318–324, https://doi.org/10.1002/prot.20630.

[35] H. Zhang, L. Zhang, D.H. Mathews, L. Huang, LinearPartition: linear-time approximation of RNA folding partition function and base-pairing probabilities, Bioinformatics 36 (Suppl_1) (Jul 1, 2020) i258–i267, https://doi.org/10.1093/bioinformatics/btaa460.

[36] R. Lorenz, S.H. Bernhart, C. Höner Zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, I.L. Hofacker, ViennaRNA package 2.0, Algorithm Mol. Biol. 6 (Nov 24, 2011) 26, https://doi.org/10.1186/1748-7188-6-26.

[37] Y. Zhang, Y. Liu, J. Xu, X. Wang, X. Peng, J. Song, D.J. Yu, Leveraging the attention mechanism to improve the identification of DNA N6-methyladenine sites, Briefings Bioinf. 22 (6) (Nov 5, 2021), https://doi.org/10.1093/bib/bbab351.

[38] J. Singh, J. Hanson, K. Paliwal, Y. Zhou, RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning, Nat. Commun. 10 (1) (Nov 27, 2019) 5407, https://doi.org/10.1038/s41467-019-13395-9.

[39] S.T. Hill, R. Kuintzle, A. Teegarden, E. Merrill 3rd, P. Danaee, D.A. Hendrix, A deep recurrent neural network discovers complex biological rules to decipher RNA protein-coding potential, Nucleic Acids Res. 46 (16) (2018) 8105–8113, Sep 19, https://doi.org/10.1093/nar/gky567.

[40] F. He, T. Liu, D. Tao, Why ResNet works? Residuals generalize, IEEE Transact. Neural Networks Learn. Syst. 31 (12) (2020) 5349–5362, Dec, https://doi.org/10.1109/tnnls.2020.2966319.

[41] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-Excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (8) (2020) 2011–2023, Aug, https://doi.org/10.1109/tpami.2019.2913372.

[42] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual Attention Network for Image Classification." pp. 3156-3164.

[43] S. Wu, G. Li, L. Deng, L. Liu, D. Wu, Y. Xie, L. Shi, L1 -norm batch normalization for efficient training of deep neural networks, IEEE Transact. Neural Networks Learn. Syst. 30 (7) (2019) 2043–2051, Jul, https://doi.org/10.1109/tnnls.2018.2876179.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[45] M. Iliadis, L. Spinoulas, A.K. Katsaggelos, Deep fully-connected networks for video compressive sensing, Digit. Signal Process. 72 (2018) 9–18.

[46] J. Han, and C. Moraga, "The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning." pp. 195-201.

[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, Pytorch: an Imperative Style, High-Performance Deep Learning Library, 2019 *arXiv preprint arXiv:1912.01703*.

[48] D. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, Computer Science, 2014.

[49] Q. Shi, W. Chen, S. Huang, Y. Wang, Z. Xue, Deep learning for mining protein data, Briefings Bioinf. 22 (1) (Jan 18, 2021) 194–218, https://doi.org/10.1093/bib/bbz156.

[50] C. Zhang, W. Zheng, S.M. Mortuza, Y. Li, Y. Zhang, DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins, Bioinformatics 36 (7) (2020) 2105–2112, Apr 1, https://doi.org/10.1093/bioinformatics/btz863.