INTEGRATED PRETRAINING WITH EVOLUTIONARY INFORMATION TO IMPROVE RNA SECONDARY STRUCTURE PREDICTION

A PREPRINT

Zichao Yan
School of Computer Science
McGill University
zichao.yan@mail.mcgill.ca

William Hamilton
School of Computer Science
McGill University
wlh@cs.mcgill.ca

Mathieu Blanchette*
School of Computer Science
McGill University
blanchem@cs.mcgill.ca

ABSTRACT

RNA secondary structure prediction is a fundamental task in computational and molecular biology. While machine learning approaches in this area have been shown to improve upon traditional RNA folding algorithms, performance remains limited for several reasons such as the small number of experimentally determined RNA structures and suboptimal use of evolutionary information. To address these challenges, we introduce a practical and effective pretraining strategy that enables learning from a larger set of RNA sequences with computationally predicted structures and in the meantime, tapping into the rich evolutionary information available in databases such as Rfam. Coupled with a flexible and scalable neural architecture that can navigate different learning scenarios while providing ease of integrating evolutionary information, our approach significantly improves upon state-of-the-art across a range of benchmarks, including both single sequence and alignment based structure prediction tasks, with particularly notable benefits on new, less well-studied RNA families.

1 Introduction

RNA secondary structure prediction is a fundamental task in computational and molecular biology. Given a sequence S of RNA nucleotides from $\{A, C, G, U\}$, the goal is to predict which pairs of nucleotides will form interactions (called base pairs) in the 3D structure S will fold into. Traditional approaches in this area have extensively relied on existing biological priors such as the decomposition of RNA secondary structures into nested substructures, as formulated in thermodynamic nearest neighbour models (TINOCO *et al.*, 1973; Pipas and McMahon, 1975) and stochastic context free grammars (SCFG) (Durbin *et al.*, 1998; Do *et al.*, 2006) based probabilistic models, which are amenable to exact dynamic programming algorithms (McCaskill, 1990; Zuker, 2003; Lorenz *et al.*, 2011). More recently, methods that leverage the power of representation learning have emerged, with learnt parameterized function that predicts RNA contact maps (Singh *et al.*, 2019; Chen *et al.*, 2020).

Throughout these lines of works, accuracy of the structure prediction pipeline has been been consistently improved, especially with the aid of more powerful deep learning techniques (Singh *et al.*, 2019), the combination of physics-based regularization with supervised learning (Sato *et al.*, 2021), and the use of evolutionary information extracted from multiple sequence alignment (MSA) (Singh *et al.*, 2021). However, several core challenges remain.

The first remarkable challenge is the limited amount of RNA secondary structure data. Compared to its protein counterpart, the task of predicting RNA secondary structures is systemically hindered by the lack of available training

*Corresponding author.

data. Among many standardized and publicly available RNA secondary structure benchmark datasets, the bprna dataset curated from Danaee *et al.* (2018) is by far the largest one with 10,814 RNAs in the training set. Importantly, it carefully controls for sequence similarity within the training, validation and test sets. In comparison, another more recent dataset (Chen *et al.*, 2020), contains a significant amount of redundant sequences between validation and test set, and hence cannot be used to measure model performance. [2]

Evolutionary information is poorly used by deep learning models. While using multiple alignment of homologous sequences has long been known to increase predictive power (Sankoff and Blanchette, 1998), the best performing deep learning approach (Singh *et al.*, 2021) so far incorporating this type of information relies on external bioinformatics tools such as gremlin (Kamisetty *et al.*, 2013) to extract direct coupling signals. While such feature engineering strategy translates well on the benchmark, an end-to-end deep learning system operating directly on raw evolutionary data may improve performance and robustness.

In order to address the challenges outlined above, we introduce a flexible and scalable neural network architecture, referred as RNA Secondary Structure Model (**RSSM**), which we show achieves state-of-the-art performance across a range of benchmark tasks, including single-sequence structure prediction, evolutionary information aided structure prediction, and transfer learning from large-scale pretrained models using sequences and evolutionary information curated from the Rfam database (Kalvari *et al.*, 2020) along with computational predicted structures. An illustration of these benchmark tasks is shown Figure 1, with more detailed description provided in Suppl. section A.

Our contribution in this paper also includes outlining a general and practical strategy for improving RSSM generalization capability in unseen RNA families without any known structures, and adapting RSSM to considerably longer RNA sequences with increased efficiency and accuracy through a sliding window based local structure prediction approach. We thoroughly analyze factors that can influence model performance, such as model ensembling and the impact of spurious homologs on alignment based RSSMs. We also investigate the robustness of alignment based RSSMs and improve its overall performance with the use of MSA transformer Rao *et al.* (2021).

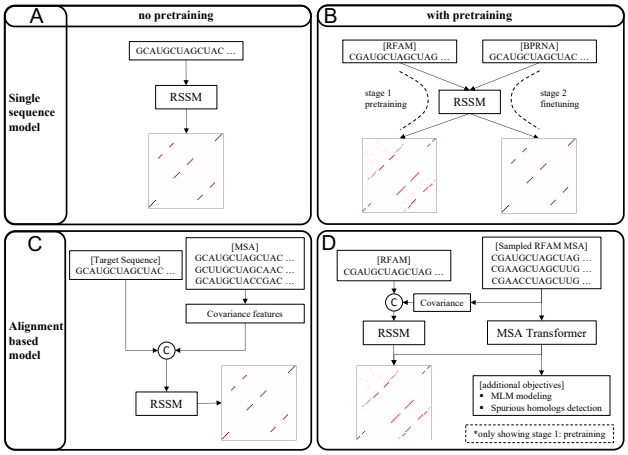

Figure 1: An illustration of the four benchmark tasks framed in our study. (A) Single sequence based RSSM learns to predict RNA secondary structure on a per RNA basis in the bprna dataset. (B) RSSM is pretrained using RNAs selected from Rfam database with computationally predicted structures, then finetuned in the bprna dataset. (C) Evolutionary information in the form of covariance matrix can be incorporated into RSSM via concatenation with the initial RNA contact map representation. (D) Rfam database also provided manually curated alignments which we seek to integrate into the pretraining process, either in its extracted covariance feature form or through a dedicated MSA transformer.

2 Related work

RNA secondary structure prediction algorithms have seen a long line of development from the earlier physics-aided thermodynamic models to more recent deep learning based approaches. Traditional RNA folding algorithms such as Mfold (Zuker, 2003), RNAstructures (Reuter and Mathews, 2010) and RNAfold Lorenz *et al.* (2011) are derived from the well-known Zucker's algorithm (Zuker and Stiegler, 1981), which is based on dynamic programming that identify the most stable RNA folding via minimizing an approximation to the molecule's global free energy (Lu *et al.*, 2006; Turner and Mathews, 2010). These approaches generally ignore pseudoknots, a type of crucial basepairing interactions, although various heuristics have been proposed to circumvent this issue, most notably by exploiting the Boltzmann ensemble basepairing probabilities estimated from methods such as LinearPartition (Zhang *et al.*, 2020) and McCaskill's algorithm (McCaskill, 1990). Representative approaches in this direction are IPknot (Sato and Kato, 2021) and ThreshKnot (Zhang *et al.*, 2019) that directly sample basepairs with high probabilities from a dense RNA contact map.

Learning-based methods represent another promising direction supported by the emergence of more RNA structure data. Earlier methods in this category are CONTRAfold (Do *et al.*, 2006; Andronescu *et al.*, 2007), ContextFold (Zakov *et al.*, 2011) and Tornado (Rivas *et al.*, 2012) which combined the strength of traditional RNA folding algorithms

[2] Limiting sequence similarity at 80% reduces the dataset to approximately 3000 sequences.

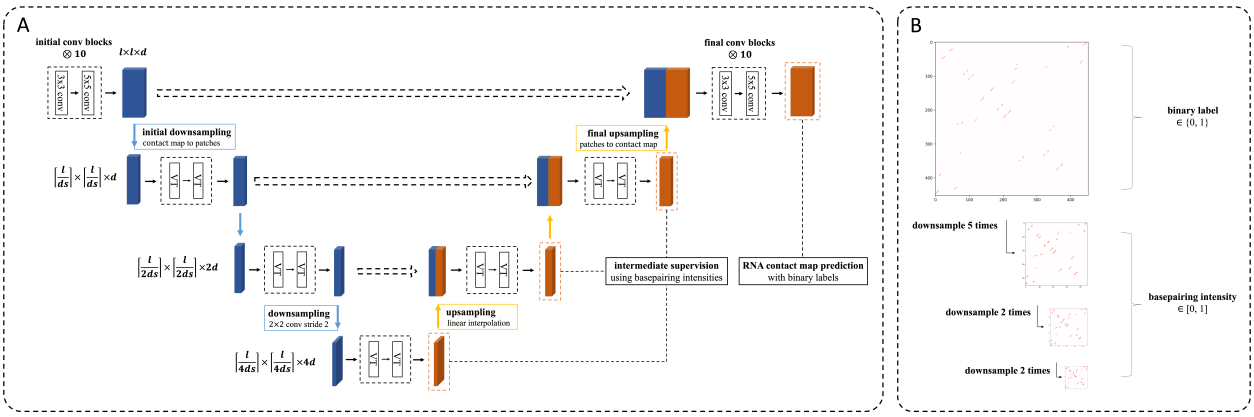

Figure 2: (A) Our model is designed with the principle to balance its focus on local and long-range RNA basepairing interactions. For this purpose, RSSM adopts a u-net architecture, with a convolutional backbone at the top dedicated to local RNA basepairs and vision transformers operating on spatially downsampled RNA feature maps for the extraction of longer range interactions. (B) RSSM can be trained with intermediate supervision in the form of basepairing intensity scores. Refer to section B for more details.

with machine learning. Inductive bias such as substructure decomposition can be built into deep learning models, as demonstrated in MXFold2 (Sato *et al.*, 2021), although they come at the expense of higher computational complexity as well as pseudoknots not being supported. Other deep learning based methods such as SPOT-RNA (Singh *et al.*, 2019) and E2Efold (Chen *et al.*, 2020) directly predict RNA contact maps (matrix of basepairing probabilities) that naturally supports pseudoknots and any non-canonical interactions, although generalization to new RNA family that lies outside the scope of their training data can be problematic (Sato and Kato, 2021).

Multiple sequence alignment (MSA) of evolutionarily related homologs can reveal co-variation patterns that are informative about interacting nucleotides and are used by approaches such as RNAalifold (Bernhart *et al.*, 2008) and CentroidAlifold (Hamada *et al.*, 2011). Direct coupling analysis (Morcos *et al.*, 2011; Kamisetty *et al.*, 2013) (DCA) has also been used to identify potential interactions by capturing co-variation between MSA columns (De Leonardis *et al.*, 2015; Pucci *et al.*, 2020). Such features are integrated into deep learning approaches such as SPOT-RNA2 (Singh *et al.*, 2021).

Recently, methods such as AlphaFold2 (Jumper *et al.*, 2021; Baek *et al.*, 2021) for protein structure prediction have demonstrated that learning to extract and integrate evolutionary information can yield a dramatic increase in performance compared to their predecessors such as AlphaFold1 (Senior *et al.*, 2020). MSA-Transformer (Rao *et al.*, 2021) also demonstrates the superiority of learnt evolutionary features over DCA.

3 Methods

The overall architecture of RSSM is shown in Figure 2 A. RSSM reflects our design principle to balance the focus on short and long range RNA secondary structure basepairs, which motivates the use of a hybrid architecture consisting of dedicated local and global modules. Indeed, we have observed that transformer-only (Vaswani *et al.*, 2017) architectures can be subject to severe overfitting (Suppl. Figure 12).

The input to our RSSMs are outer-concatenated RNA sequence embedding yielding an $\mathbb{R}^{l \times l \times d}$ representation. In alignment based RSSMs, outer-concatenated one-hot encoding of the RNA sequence are used instead, and upon concatenation with evolutionary information, undergoes a 3×3 convolution layer to obtain the initial RNA contact map representation.

3.1 RSSM convolutional backbone

Based on the above consideration, our model resorts to a u-net architecture (Ronneberger *et al.*, 2015), with convolutional modules (LeCun *et al.*, 1989) at the top operating on the full RNA contact map representation. We arrange our convolutional modules into blocks, each consisting of a 3×3 convolution followed by a 5×5 convolution. Each convolutional layer is preceded by layer normalization (Ba *et al.*, 2016), activation function and dropout (Srivastava *et al.*, 2014) along with residual connection (He *et al.*, 2016).

Our RSSM contains 20 such blocks and they are considered the backbone of our model, as feature maps generated by the first half of these convolutions are used as input to the later self-attention modules, while the latter half produce the final RNA contact map representation.

3.2 Self-attention on spatially downsampled RNA contact map representation

For a given RNA contact map representation $M \in \mathbb{R}^{l \times l \times d}$, an ordinary self-attention proceeds by first flattening the map along its spatial dimensions into $\hat{M} \in \mathbb{R}^{l^2 \times d}$. Then, self-attention proceeds as follows:

$$\text{SA}(\hat{M}) = \text{Softmax}(\frac{W_q(\hat{M})W_k(\hat{M})^T}{\sqrt{d}} + B)W_v(\hat{M}) \tag{1}$$

where W_q, W_k and W_v respectively map query, key and value vectors from $\hat{M}$, and B represents a bias matrix which can be used to encode positional information in M.

Despite its straightforwardness, the computational complexity in Eq. 1 is $\mathcal{O}(l^4)$, which poses considerable optimization challenge for RNAs of a few hundred nucleotides or more. In order to facilitate such computation, we implement two strategies: (i) downsampling the original spatial dimensionality of M, and (ii) limiting the scope of self-attention.

Below the convolutional backbone, our model looks at spatially downsampled RNA feature maps, with dimension being $\lceil \frac{l}{ds} \rceil \times \lceil \frac{l}{ds} \rceil$, $\lceil \frac{l}{2ds} \rceil \times \lceil \frac{l}{2ds} \rceil$ and $\lceil \frac{l}{4ds} \rceil \times \lceil \frac{l}{4ds} \rceil$. This follows the approach of a vision transformers (Dosovitskiy *et al.*, 2021) (denoted by VT). These downsampled maps are obtained using convolution layers with strides of ds, 2 and 2, which merge spatially contiguous regions while doubling the number of feature channels. Self-attention is only applied to these downsampled feature maps, which incurs less computational overhead while simultaneously facilitating the discovery of longer range interactions.

These downsampled feature maps are connected to the original RNA contact map representation through the horizontal shortcut and expansive path in the u-net architecture, which allows spatially condensed information from lower levels to be gradually merged into the RSSM backbone. In each upsampling step, we concatenate linearly interpolated lower level features with a corresponding map from the shortcut, which then undergo a 3×3 convolution layer followed by another two vision transformer modules. Along the contractive path, we insert paddings whenever necessary. Similarly, linear interpolated feature maps are cropped to align the corresponding downsampled map from the contractive path.

To limit the scope of self-attention, we followed the local window approach in swin-transformer (Liu *et al.*, 2021), which confines the computation of self-attention within non-overlapping local windows. Assuming a window size of w, RNA contact map representation M is partitioned into non-overlapping patches of size $w \times w$ which leads to $M_p \in \mathbb{R}^{\lceil \frac{l}{w} \rceil^2 \times w \times w \times d}$, and self-attention simply computes within each patch. Using this localized mode of self-attention, the computational complexity can be brought down to $\mathcal{O}(w^2 l^2)$ which is more practical with longer RNAs. In addition, we apply shifted local windows for each consecutive transformer module as well as relative positional embeddings in Eq. 1.

3.3 Pretraining strategies

The core objective of pretraining is to overcome the limited size of our labeled training set, which otherwise poses considerable challenge to the generalization capability of our RSSMs in new RNA families. An illustration of our pretraining strategies is shown in Figure 1 B and D.

For single sequence pretraining, we leverage predicted basepairing probabilities from LinearPartition (Zhang *et al.*, 2020) as learning targets in place of the ordinary ground truth binary labels (Eq. 4). Such supervised pretraining strategy is applicable to any RNA sequences, without requiring knowledge of their actual structures. Here, we use non-coding RNA sequences from Rfam database as they are mostly associated with biologically meaningful regulatory functions that are useful to other studies.The Rfam database also provides manually curated alignments that can be used later for our evolution-based pretraining.

We also consider another auxiliary objective which is to classify non-coding RNA superfamilies based on the taxonomy in RNArchitecture (Boccaletto *et al.*, 2017). A total of 91 superfamilies are selected, each represented by at least 100 RNA sequences in our curated pretraining dataset. Sequences from less well-represented RNA superfamilies are excluded. We use feature maps at the bottom of RSSM to perform this type of classification via global averaging and a dedicated classification head.

When MSAs are available for evolutionary information integrated pretraining, we consider either directly incorporating such information into RSSM via extraction of a covariance matrix followed by concatenation with the

initial RNA contact map representation, or leveraging a more suited representation learning technique known as MSA transformer (Rao *et al.*, 2021).

During the course of pretraining, for each RNA sequence we sample a subset of its homologs from our curated pretraining data, and align them into a MSA with the corresponding Rfam covariance model (Figure 6 A). We also add noise to these subsampled Rfam MSAs that can be interpreted as spurious homologs, in the hope of accommodating the potential quality discrepancies between Rfam's expert annotated MSAs and ad-hoc MSAs obtained with the RNAcmap (Zhang *et al.*, 2021) automatic pipeline for RNA sequences in other dataset. In particular, 10% of sequences in Rfam subsampled MSAs are manually contaminated with noise, with half of these sequences randomly replacing 10% of their nucleotides with $\{A, C, G, U, -\}$, and the second half having 20% of their k-mers replaced by contents in other RNA families (k being a uniformly sampled integer from 5 to 15).

As illustrated in Figure 1 D, we also aim to fully represent these subsampled noisy Rfam MSAs with a dedicated MSA transformer, from which evolutionary information may be more informatively and robustly integrated into our RSSM. Similar to Rao *et al.* (2021), our MSA transformer is trained with masked language modeling (MLM) objective (Devlin *et al.*, 2019), with 15% of tokens randomly masked. In addition, our MSA transformer performs spurious homolog detection and basepairing probability prediction. Given a learnt MSA $h \in \mathbb{R}^{m,l,d}$ where m and l are the depth and length of the subsampled MSA, and all row-wise attention weights $a \in \mathbb{R}^{l,l,d'}$, the overall objective can be expressed as follows:

$$\begin{aligned}\mathcal{J}_{\text{MSA}} = &\sum_{i,j \in \text{mask}} \log p_\theta(y_{ij}^{mlm} | h_{ij}) \\ &+ \sum_{i=1\ldots m} \log p_\theta(y_i^{noise} | \frac{1}{l} \sum_{j=1\ldots l} h_{ij}) \\ &+ \sum_{i,j \in \text{triu_indices}(l \times l)} \log p_\theta(y_{ij}^{prob} | [H_{tt}; a])\end{aligned} \tag{2}$$

H_{tt} is the outer concatenation of $h_{t\cdot}$, representation of the target RNA sequence with its index in the MSA denoted by t. Concerning our computational budget, we keep the maximum depth of randomly subsampled MSA at $m = 2^{15}/l$ during the course of pretraining. We make sure that the target sequence is always included in subsampled MSAs.

4 Results

Information on hyperparameters and training details is in Suppl. section B. Here, we only present our core results.

4.1 Performance of single-sequence based models

We present the performance for our single-sequence based RSSM in Table 1, along with a variety of baselines on the bprna test set. All our models are trained 300 epochs using the bprna training set and we are able to pinpoint several factors that benefit RSSM performance. For each base dimension and transfer learning setting, we train five RSSM replicates with random weight initialization and register the mean and standard deviation of their performance under single models.

First of all, doubling the base dimension of RSSM from 64 to 128 yields a consistent 4% F1 improvement across most training setting. These gains come at a cost in terms of computational demand, but this remains tractable.

Pretraining using Rfam significantly improves performance across various settings. It helps maintaining a reasonable computational overhead with $d = 64$, while achieving nearly comparable performance to RSSM with higher hidden dimension $d = 128$. Furthermore, utilizing our curated pretraining data allows RSSM to achieve better generalization capability in new RNA families absent from the bprna finetuning dataset, which we will elucidate later in section 4.2 and Suppl. section C.

Finally, model ensembling by simply averaging the predicted contact map basepairing probabilities across the five replicates also substantially improves performance. Although all replicates share the same neural architecture, random initialization and shuffled training help each model learn different hypotheses, and ensemble averaging serves as a mean of trimming outlier predictions while retaining confident basepairs that all replicates agree on. We look into this phenomenon in Suppl. section G. Indeed, this simple ensembling strategy has proven to be essential later in our attempt to boost RSSM generalization capability in new RNA families (section 4.4) and in our end-to-end alignment based RSSM (section J).

Table 1: Performance of our single sequence based RSSMs and a variety of baselines, on the bprna dataset. Subscripts indicate the base dimension (d64 or d128) of RSSM and use of transfer learning (T).

		Test set		
Mode	Model	F1	Precision	Recall
Singe models	RSSM_{d64}	0.656±0.002	0.662±0.006	0.674±0.006
	$\text{RSSM}_{\text{d64,T}}$	0.671±0.003	0.689±0.005	0.676±0.005
	$\text{RSSM}_{\text{d128}}$	0.679±0.003	0.700±0.005	0.682±0.008
	$\text{RSSM}_{\text{d128,T}}$	0.702±0.001	0.728±0.011	0.698±0.008
Ensemble models	RSSM_{d64}	0.672	0.668	0.704
	$\text{RSSM}_{\text{d64,T}}$	0.683	0.715	0.683
	$\text{RSSM}_{\text{d128}}$	0.700	0.737	0.690
	$\text{RSSM}_{\text{d128,T}}$	0.720	0.758	0.708
Baselines	SPOT-RNA	0.626	0.709	0.560
	RNAfold	0.491	0.449	0.577
	CONTRAfold	0.522	0.484	0.598
	IPknot			
	LinearPartition-C	0.525	0.514	0.569
	LinearPartition-V	0.500	0.455	0.591
	ThreshKnot			
	LinearPartition-C	0.523	0.542	0.542
	LinearPartition-V	0.499	0.453	0.589

4.2 Generalization to unseen RNA families

All our RSSM variants in Table 1 significantly outperform the baselines as they have leveraged the power of representation learning. Traditional RNA folding algorithms such as RNAfold, CONTRAfold and LinearPartition based methods have built-in inductive bias such as the decomposition formula for large RNA secondary structures. In comparison, deep learning models such as SPOT-RNA and RSSMs may fail to generalize to new RNA families outside the scope of training data.

In order to comprehensively evaluate the generalization capability of our RSSM series, we obtain two external datasets (Sato and Kato, 2021): the bprna-1m dataset and an Rfam-derived dataset consisting of RNA families not included in the bprna training set. Here we only give a succinct summary of our findings; see Suppl. section C and Table 4, 5 and 6 for more details on these external datasets and our experiments.

On the Rfam-derived dataset featuring new RNA families, RSSM_{d64} and $\text{RSSM}_{\text{d64,T}}$ have limited generalization capability compared to traditional approaches. Nevertheless, comparable or even superior performance is achieved by a novel RSSM ensemble referred as $\text{RSSM}_{\text{d64,mix}}$, which combines a diverse selection of model hypotheses that aim to more thoroughly leverage computationally predicted RNA secondary structures. Models in this mixed ensemble are either trained on bprna dataset as usual in section 4.1, or finetuned with lower learning rates and optionally augmented with sequences sampled from our curated Rfam pretraining data, with labels being predicted basepairing probabilities rather than real RNA structures.

Including these computationally predicted structures into finetuning slightly compromises the performance of $\text{RSSM}_{\text{d64,mix}}$ on real RNA structures in the bprna-1m dataset. However, a more balanced performance can now be achieved between old and new RNA families in the bprna-1m and Rfam-new-families dataset. In fact, $\text{RSSM}_{\text{d64,mix}}$ has caught up with or even surpassed state-of-the-art traditional approaches on short and medium RNAs in the Rfam derived dataset.

It is worth emphasizing that the improvement of $\text{RSSM}_{\text{d64,mix}}$ is achieved without having access to any real RNA structures in newer RNA families, which indicates that our RSSMs and presumably other learning based methods can adapt to new RNA families and domains of structures even if experimentally determined RNA structures are not available for those families. This is achieved by leveraging a larger pool of RNAs with computationally predicted structures produced by LinearPartition (Zhang *et al.*, 2020).

We also evaluate our RSSMs on datasets of high resolution RNA secondary structures in section E, where we observe similar performance gains of $\text{RSSM}_{\text{d64,mix}}$. Different contact map discretization techniques potentially have a large

impact on the final performance, which we analyze in section F. We find out that our RSSMs are amenable to a wide variety of discretization strategies at low probability cutoffs.

4.3 Performance of RSSM on long RNA sequences

The length of RNA sequences included for pretraining or finetuning is either capped at 1000 or 500 nts respectively for the sake of computational efficiency. Therefore, prediction accuracy on longer RNA sequences (up to 4381 nts) needs to be assessed. As results in Table 6 show, our most sophisticated ensemble model $\text{RSSM}_{\text{d64,mix}}$ underperforms compared to the best performing traditional algorithms (usually LinearPartition).

Without soliciting additional supervision from these much longer sequences, we seek a simple workaround by adding LinearPartition into our ensemble model in order to leverage its built-in inductive bias on larger RNA structure decomposition. This leads to a hybrid model referred as $\text{RSSM}_{\text{d64,mix,LPC}}$ which comprehensively surpasses $\text{RSSM}_{\text{d64,mix}}$ on long RNA sequences in both dataset.

Furthermore, we adopt a sliding window based local structure prediction approach, considering the fact that RNA basepair spans are effectively capped during training, which gives rise to a reasonable conjecture that RSSM cannot reliably predict longer basepairs. Therefore, in the hope of trading the discovery of possible longer range interactions for more accurate local RNA structures, we average basepairing probabilities predicted within many overlapped local windows (Equation 6), as originally proposed in RNAplfold (Bernhart *et al.*, 2006).

Combining the sliding window approach with our RSSMs tangibly improves their performance on long RNA sequences in both bprna-1m and Rfam-new-families, as shown in section D and Table 7.

Table 2: Performance of our alignment based models on the bprna test set. Subscript also indicates the type of consensus secondary structure used to annotate MSAs (rnafold or dl).

		Test set		
Mode	Model	F1	Precision	Recall
Singe models	$\text{RSSM}_{\text{d64,rnafold}}$	0.733±0.003	0.742±0.002	0.742±0.004
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.740±0.004	0.755±0.009	0.745±0.005
	$\text{RSSM}_{\text{d64,dl}}$	0.742±0.001	0.764±0.002	0.738±0.002
	$\text{RSSM}_{\text{d64,T,dl}}$	0.747±0.002	0.769±0.002	0.743±0.004
Ensemble models	$\text{RSSM}_{\text{d64,rnafold}}$	0.746	0.766	0.747
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.752	0.779	0.749
	$\text{RSSM}_{\text{d64,dl}}$	0.749	0.779	0.741
	$\text{RSSM}_{\text{d64,T,dl}}$	0.754	0.782	0.747
End-to-end model	$\text{MSA-RSSM}_{\text{d64,T,rnafold}}$	0.777	0.793	0.776
Baselines	SPOT-RNA2-IL0	0.727	0.731	0.723
	SPOT-RNA2-IL1	0.734	0.780	0.693
	SPOT-RNA2-IL2	0.726	0.716	0.737
	RNAalifold	0.552	0.695	0.498
	CentroidAlifold	0.619	0.611	0.679
	IPknot	0.627	0.680	0.623

4.4 Performance of alignment based models

Performance of our alignment based RSSMs is shown in Table 2, while full result is contained in supplementary section J and details about the role played by consensus secondary structures annotation inside the RNAcmap pipeline (Zhang *et al.*, 2021), which is a crucial step for obtaining MSAs, are in supplementary section A.

Compared to single sequence based RSSMs, the gain from simply adding covariance features extracted from MSAs is significant (see Figure 3 A), irrespective of the type of consensus secondary structure annotated within these MSAs (indicated by the subscript, e.g. dl stands for our own deep learning model $\text{RSSM}_{\text{d64,T}}$). Performance gains are significant and reliable especially when the number of effective sequences in the MSA is sufficiently large ($N_{\text{eff}} > 50$) as shown in Figure 3 B and C. Among other contributing factors, finetuning on top of evolutionarily pretrained RSSMs and ensemble modeling have also tangibly contributed to the performance.

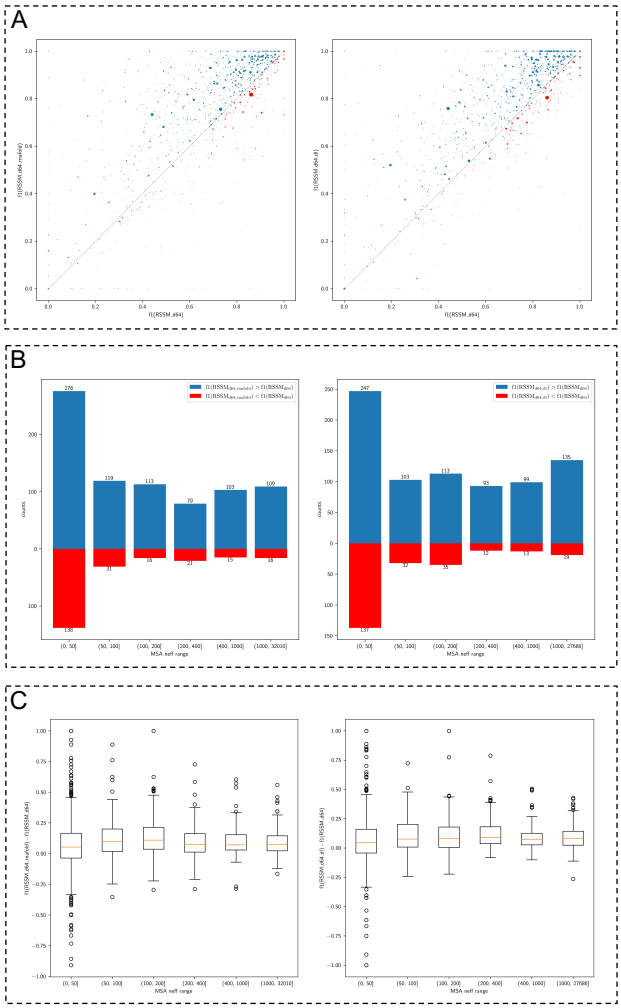

Figure 3: (A) Performance of alignment based $\text{RSSM}_{\text{d64,rnafold}}$ and $\text{RSSM}_{\text{d64,dl}}$ compared to single sequence based RSSM_{d64}. Point size indicates N_{eff} of MSA. (B), (C) The effect of N_{eff} on model performance illustrated in bar plots and box plots.

The type of consensus secondary structure lays a large impact on MSAs and consequently influences the performance (see supplementary section A and Figure 6 for a better understanding). In particular, consensus secondary structures annotated by our own deep learning tool i.e. one $\text{RSSM}_{\text{d64,T}}$ replicate, appear to entail more powerful MSAs, which is indicated by the performance gain of $\text{RSSM}_{\text{d64,dl}}$ over $\text{RSSM}_{\text{d64,rnafold}}$. In section I and Figure 13, we directly extract co-conservation signals through DCA from MSAs as a mean to predict RNA basepairs. Once again, our results suggest the superiority of deep learning annotated MSAs.

However, further analysis on the robustness of alignment based RSSMs (Supp. section K.1) reveals that RNAfold-annotated MSAs should be used instead for training purposes, as $\text{RSSM}_{\text{d64,T}}$ annotated MSAs can induce severe overfitting and cause instability of trained RSSMs on the RNAfold annotated MSAs (Table 10). On the other hand, RSSMs trained with RNAfold annotated MSAs are more robust and in fact, perform better on $\text{RSSM}_{\text{d64,T}}$ annotated MSAs. We seek to better understand this phenomenon in section K.1, and point out a possible source of perturbation in SPOT-RNA2 intermediate features that can undermine its prediction accuracy.

To fully integrate evolutionary information from raw MSAs, we combine MSA transformer (Rao *et al.*, 2021) with RSSM which leads to an end-to-end model referred as MSA-RSSM. More details on this model and a dedicated novel ensembling strategy through repeated random MSA subsampling can be found in Suppl. section J. We pretrain the MSA transformer using Rfam MSAs as part of our pretraining strategies (section 3.3), before the whole MSA-RSSM is finetuned on RNAfold-annotated MSAs. Coupled with the novel ensembling strategy, MSA-RSSM produces a new state-of-the-art on the benchmark in Table 2 and 9.

Furthermore, we investigate the robustness of our RSSMs on shallow MSAs (see Suppl. section K.2 and Figure 14). Both $\text{RSSM}_{\text{d64,T,rnafold}}$ and $\text{MSA-RSSM}_{\text{d64,T,rnafold}}$ are relatively robust to shallow MSAs as they outperform single sequence $\text{RSSM}_{\text{d64,T}}$ as long as a minimal amount of $N_{\text{eff}} = 5$ homologous sequences are retained for each MSA under the low setting, or $N_{\text{eff}} = 10$ sequences in the high setting. In addition, $\text{MSA-RSSM}_{\text{d64,T,rnafold}}$ has been shown to better utilize evolutionary information than the covariance feature based RSSM.

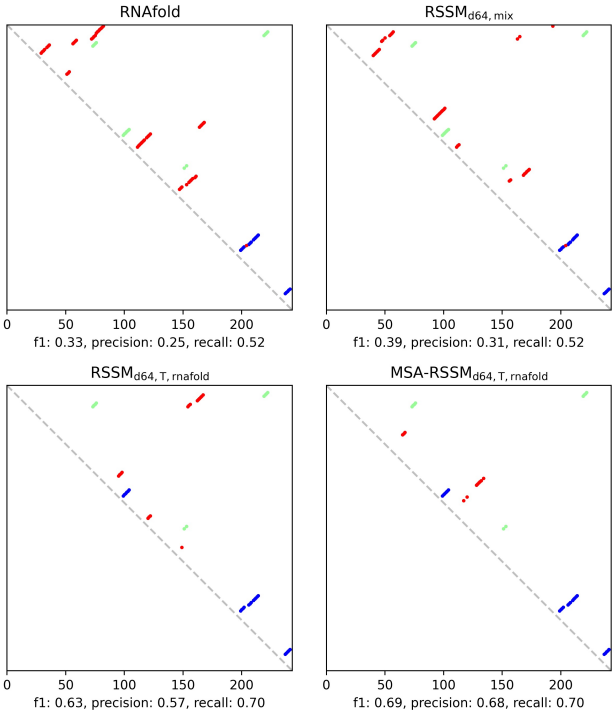

Figure 4: Leveraging evolutionary information improves prediction accuracy of many RNA secondary structures in the RFAM derived dataset. Here is an example of a small bacterial non-coding RNA (Rfamid: RF02673, seqid: ABYA01000314.1/2329-2086), with 330 non-redundant homologs in the MSA. Blue: true positive; red: false positive; green: false negative.

4.5 Generalization of alignment based RSSMs

We also evaluate our alignment based RSSMs on new RNA families, as in sections 4.2 and C, except that we focus on Rfam-derived dataset and only consider medium to long RNA sequences. Full results are presented in Suppl. section L, Table 11 and Table 12.

Using RNAfold or $\text{RSSM}_{\text{d64,mix}}$ annotated MSAs, on medium RNA sequences and long RNA sequences with pseudoknots, our $\text{RSSM}_{\text{d64,rnafold}}$, $\text{RSSM}_{\text{d64,T,rnafold}}$ and $\text{MSA-RSSM}_{\text{d64,T,rnafold}}$ significantly outperform all single sequence as well as alignment based baselines such as traditional RNA folding algorithms and DCA. In particular, our end-to-end $\text{MSA-RSSM}_{\text{d64,T,rnafold}}$ dominates the benchmark on medium RNA sequences, and its performance is most impressive on RNAs with pseudoknots where we observe gains of 8% over the best covariance based RSSM, and 22% over the best single sequence model. An example of such performance gain is shown in Figure 4.

Overall, comparing alignment based to single sequence based RSSMs reveals that the leveraging evolutionary information substantially improves performance on new RNA families. Our alignment based RSSMs are also more effective at incorporating evolutionary information than traditional methods.

4.6 Effect of spurious homologs on prediction

Adding evolutionary information to RSSM in the form of extracted covariance features measurably improves its overall performance. However the scale of such improvement varies depending on factors such as the quality of the alignment. In particular, MSAs often include non-homologous sequences that hampered covariance-based analyses. Detecting and removing such sequences would likely improve RSSM's performance.

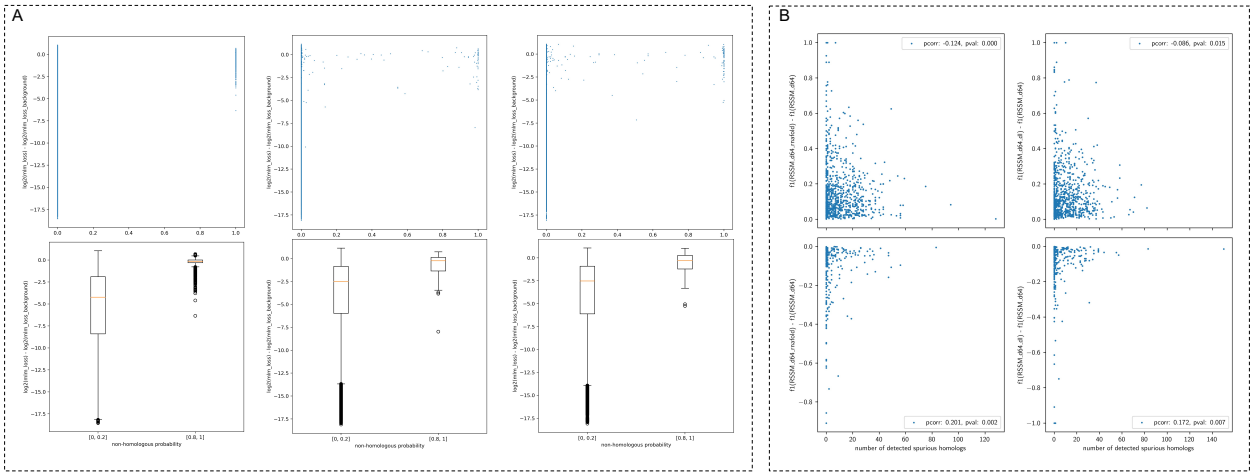

Figure 5: (A) Distribution of MLM loss fold change on homologs and detected non-homologs. (B) The effect of spurious homologs discovered in MSA on model performance.

Intuitively, a spurious homolog is inherently incompatible with the evolutionary information presented by a MSA, therefore reconstructing masked tokens should be difficult for those sequences. First, we first try to establish the relationship between spurious homologs that are detected by our pretrained MSA transformer and the loss of predicting masked tokens therein. However, considering the fact that reconstructing masked tokens does not always necessitate the knowledge of other homologs, where a single sequence would suffice as shown in single sequence based language models such as ESM and ProtTrans (Rives *et al.*, 2021; Elnaggar *et al.*, 2021), we turn to measure the fold decrease of masked token prediction loss to the background loss of using only the sequence itself. Given a masked MSA input $x \in \mathbb{R}^{m,l}$, the fold change of a specific masked position indexed by i, j is defined as:

$$\log \frac{\mathcal{L}_{mlm}(\text{MSA-T}_\theta(x)_{ij}, y_{ij})}{\mathcal{L}_{mlm}(\text{MSA-T}_\theta(x_{i\cdot})_j, y_{ij})} \tag{3}$$

MSA-T$_\theta$ is pretrained using Rfam evolutionary data which are subsampled high quality MSA except manually contaminated noise. We discard masked tokens that can be easily inferred without other homologs by setting a threshold on the background loss, i.e. $\mathcal{L}_{mlm}(\text{MSA-T}_\theta(x_{i\cdot})_j, y_{ij}) > 5.0$, and record how much fold decrease can be achieved via utilizing homologous information. We discard MSAs with less than 1000 homologs.

As shown in Figure 5 A, there is a striking difference between the fold change values obtained on masked tokens from actual homologs versus those from spurious homologs. This observation is relatively consistent across various sources of MSA. In particular, masked tokens on detected spurious homologs tend to only exhibit smaller fold decrease, which means incorporating evolutionary information from other sequences in the MSA cannot improve their prediction, hence suggesting its incompatibility with the MSA.

In Figure 5 B, we take a closer look at how potential non-homologs induce performance change. Each point is an RNA in bprna validation set and a general trend is reflected that as the number of spurious homologs increases the impact of extracted covariance features diminishes.

Overall, we are able to show a negative correlation between the amount of potential noise in a MSA and its influence on performance change. This indicates that noise within MSAs unavoidably undermine the utility of the evolutionary information it represents thereof.

5 Conclusion

With evolutionary information integrated pretraining, our deep learning models obtain state-of-the-art performance across a range of benchmark tasks that span single sequence and alignment based structure prediction, and possess superior generalization capability rivaling or surpassing the state-of-the-art traditional RNA folding algorithms in new RNA families. This opens the door to future research as more RNA structure data can be made available to support deep learning methods, accompanied with dedicated modelling techniques and inference strategies to better accommodate evolutionary information.

References

Andronescu, M. *et al.* (2007). Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics*, **23**(13), i19–i28.

Ba, J. L. *et al.* (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Baek, M. *et al.* (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*.

Bernhart, S. H. *et al.* (2006). Local RNA base pairing probabilities in large sequences. *Bioinformatics*, **22**(5), 614–5.

Bernhart, S. H. *et al.* (2008). RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, **9**(1), 474.

Boccaletto, P. *et al.* (2017). RNArchitecture: a database and a classification system of RNA families, with a focus on structural information. *Nucleic Acids Research*, **46**(D1), D202–D205.

Chen, X. *et al.* (2020). RNA secondary structure prediction by learning unrolled algorithms. In *International Conference on Learning Representations (ICLR)*.

Danaee, P. *et al.* (2018). bpRNA: large-scale automated annotation and analysis of RNA secondary structure. *Nucleic Acids Research*, **46**(11), 5381–5394.

De Leonardis, E. *et al.* (2015). Direct-coupling analysis of nucleotide coevolution facilitates RNA secondary and tertiary structure prediction. *Nucleic Acids Research*, page gkv932.

Devlin, J. *et al.* (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019*.

Do, C. B. *et al.* (2006). CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, **22**(14), e90–e98.

Dosovitskiy, A. *et al.* (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*.

Durbin, R. *et al.* (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Elnaggar, A. *et al.* (2021). ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. *arXiv:2007.06225 [cs, stat]*.

Fu, L. *et al.* (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**(23), 3150–3152.

Hamada, M. *et al.* (2011). Improving the accuracy of predicting secondary structure for aligned rna sequences. *Nucleic acids research*, **39**(2), 393–402.

He, K. M. *et al.* (2016). Identity Mappings in Deep Residual Networks. *Computer Vision - Eccv 2016, Pt Iv*, **9908**, 630–645.

Jumper, J. *et al.* (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*.

Kalvari, I. *et al.* (2020). Rfam 14: expanded coverage of metagenomic, viral and microRNA families. *Nucleic Acids Research*, **49**(D1), D192–D200.

Kamisetty, H. *et al.* (2013). Assessing the utility of coevolution-based residue-residue contact predictions in a sequence- and structure-rich era. *Proceedings of the National Academy of Sciences*, **110**(39), 15674–15679.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR*.

LeCun, Y. *et al.* (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, **1**(4), 541–551.

Liu, Z. *et al.* (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, **abs/2103.14030**.

Lorenz, R. *et al.* (2011). ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, **6**, 26.

Lu, Z. J. *et al.* (2006). A set of nearest neighbor parameters for predicting the enthalpy change of RNA secondary structure formation. *Nucleic Acids Research*, **34**(17), 4912–4924.

Madden, T. (2013). The BLAST sequence analysis tool. *The NCBI handbook*, **2**, 425–436.

McCaskill, J. S. (1990). The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**(6-7), 1105–19.

Morcos, F. *et al.* (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, **108**(49), E1293–E1301.

Nawrocki, E. P. and Eddy, S. R. (2013). Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*, **29**(22), 2933–2935.

Pipas, J. M. and McMahon, J. E. (1975). Method for predicting rna secondary structure. *Proceedings of the National Academy of Sciences*, **72**(6), 2017–2021.

Pucci, F. *et al.* (2020). Evaluating DCA-based method performances for RNA contact prediction by a well-curated dataset. *RNA*, page rna.073809.119.

Rao, R. M. *et al.* (2021). MSA Transformer. In *International Conference on Machine Learning (ICML), 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR.

Reuter, J. S. and Mathews, D. H. (2010). RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, **11**(1), 129.

Rivas, E. *et al.* (2012). A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. *RNA (New York, N.Y.)*, **18**(2), 193–212.

Rives, A. *et al.* (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, **118**(15).

Ronneberger, O. *et al.* (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer.

Sankoff, D. and Blanchette, M. (1998). Multiple genome rearrangement and breakpoint phylogeny. *Journal of computational biology*, **5**(3), 555–570.

Sato, K. and Kato, Y. (2021). Prediction of RNA secondary structure including pseudoknots for long sequences. *Briefings in Bioinformatics*.

Sato, K. *et al.* (2021). RNA secondary structure prediction using deep learning with thermodynamic integration. *Nature Communications*, **12**(1), 941.

Senior, A. W. *et al.* (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, **577**(7792), 706–710.

Singh, J. *et al.* (2019). RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature Communications*, **10**(1), 5407.

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.

Srivastava, N. *et al.* (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, **15**, 1929–1958.

TINOCO, I. *et al.* (1973). Improved Estimation of Secondary Structure in Ribonucleic Acids. *Nature New Biology*, **246**(150), 40–41.

Turner, D. H. and Mathews, D. H. (2010). NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic acids research*, **38**(Database issue), D280–D282.

Vaswani, A. *et al.* (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (Neurips), 2017*, pages 5998–6008.

Zakov, S. *et al.* (2011). Rich parameterization improves RNA structure prediction. *Journal of Computational Biology*, **18**(11), 1525–1542.

Zhang, H. *et al.* (2020). LinearPartition: linear-time approximation of RNA folding partition function and base-pairing probabilities. *Bioinformatics*, **36**(Supplement_1), i258–i267.

Zhang, L. *et al.* (2019). Threshknot: Thresholded probknot for improved rna secondary structure prediction. *arXiv preprint arXiv:1912.12796*.

Zhang, T. *et al.* (2021). RNAcmap: a fully automatic pipeline for predicting contact maps of RNAs by evolutionary coupling analysis. *Bioinformatics*, **37**(20), 3494–3500.

Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, **31**(13), 3406–3415.

Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic acids research*, **9**(1), 133–148.

A Task description

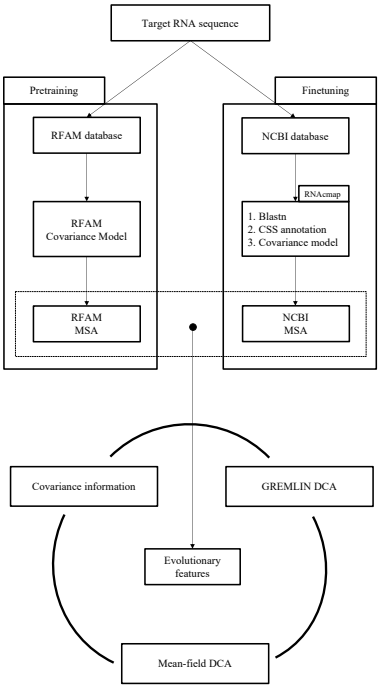

Figure 6: Our study is concerned with two ways of obtaining multiple sequence alignment. Approach shown on the left makes use of an existing Rfam covariance model to align a target RNA sequence with other known sequences from the same RNA family. It should be noted that this approach is not widely used in practice for two reasons: (1) an RNA may come from an unknown RNA family, and (2) Rfam covariance models already contain built-in consensus secondary structures which dissolve the need for another predictive system. Nevertheless, this method can be used to obtain Rfam MSAs for the purpose of evolutionary information integrated pretraining. The other approach shown on the right, having the RNAcmap tool at its core, is able to construct useful MSAs for RNAs from unknown RNA families. The RNAcmap pipeline has three procedures: (1) first round of blastn search in NCBI to obtain seed alignments, (2) annotate consensus secondary structure, (3) build customized covariance model and initiate the secondary round search using Infernal in NCBI to obtain the final MSA.

Training single sequence based models. While adding evolutionary information usually improves a model's prediction, acquiring such information from a gigantic nucleotide database and applying them to the model in a desirable format often generate an incredible amount of computational pressure that undermines the efficiency of the whole pipeline (Zhang *et al.*, 2021; Singh *et al.*, 2021), not to mention the possibility of unable to find any useful evolutionary information in the first place. Therefore, it is still worth considering single sequence based predictors even if they come with limited accuracy, which leads to the first task considered in this study as shown in Figure 1 A. For this task, we train RSSM using the same bprna dataset as in Singh *et al.* (2019), with 10,814 sequences in the training set, 1300 sequences in the validation set and 1305 sequences in the test set. Then, we acquire external dataset from Sato and Kato (2021) to evaluate the generalization capability of RSSMs.

Training alignment based models. Here, we consider adding evolutionary information into RSSM. For each sequence in bprna dataset, we use the RNAcmap tool (Zhang *et al.*, 2021) to obtain a MSA from the NCBI nucleotide database (Figure 6 right). For the annotation of consensus secondary structure inside the RNAcmap pipeline, we use either one of our own deep learning models or RNAfold (Lorenz *et al.*, 2011), considering that thermodynamic parameters in RNAfold are independent of RNA families and the tool has been shown to generalize better to new Rfam families (Sato and Kato, 2021). Consensus secondary structure of a set of related RNA homologs is meant to reveal evolutionarily co-conserved RNA basepairs, and having an accurate consensus secondary structure annotation may lead to the discovery of more evolutionarily related RNA homologs that can be leveraged as cues to the ground truth RNA structures. Combining the seed alignment found by blastn (Madden, 2013) and annotated consensus secondary structure, RNAcmap builds a covariance model (Durbin *et al.*, 1998) and uses Infernal (Nawrocki and Eddy, 2013) to obtain the final MSA in the hope of discovering more evolutionarily related RNA homologs.

MSAs are the sources of evolutionary information, from which we extract covariance features as additional inputs to RSSM (Figure 1 C) and only covariance features, as it is analogous to the one-hot encoding commonly used to represent single sequences, although much information from the MSA may be lost. For more details about the construction of covariance features which involves gathering single MSA column and pairwise columns statistics under a sequence re-weighting scheme meant to correct for phylogeny sampling bias , please refer to Morcos *et al.* (2011) Eq.1, Eq.2 and Eq.10. It is worth pointing out that covariance matrix of MSA has formed the basis of various direct coupling analysis pipelines including mean-field DCA (Morcos *et al.*, 2011).

Transfer learning from Rfam pretrained models. To address the issue of limited amount of training data, we seek to pretrain our RSSM with sequences from the Rfam database (Kalvari *et al.*, 2020) (v14.5). As shown in Figure 1 B and D, we consider either pretraining single sequence based models or integrating evolutionary information from Rfam MSAs into the pretraining process. For both purposes, we need a basic sequence library and we simply combine sequences shorter than 1000 nts from all Rfam families, followed by procedures for removing redundant sequence. In particular, cd-hit-est (Fu *et al.*, 2012) is used to control the sequence similarity within the Rfam sequence library at 80%, and cd-hit-est-2d is used to remove sequences which share at least 80% similarity with sequences in the bprna test set. We finally obtain a set of 184,226 unique sequences.

Afterwards, for single sequence pretraining, we obtain LinearPartition (Zhang *et al.*, 2020) ensemble basepairing probabilities for each sequence, which are employed as supervised learning targets. Additionally, we consider an auxiliary pretraining objective that is non-coding RNA classification, with the class labels being RNA superfamilies according to RNArchitecture (Boccaletto *et al.*, 2017). A total of 91 superfamilies are selected, each represented by at least 100 RNA sequences in our curated pretraining dataset, with some of the larger superfamilies being transfer RNA (tRNA), C/D-box RNA, large subunit ribosomal RNA (LSU rRNA) and Cobalamin riboswitch, each containing more than 7000 sequences. Sequences from less well-represented RNA superfamilies are excluded from this type of supervision. The single sequence pretrained model is then used to finetune in bprna dataset, as shown in Figure 1 B.

For evolutionary information integrated pretraining, sequences belonging to the same RNA family are aligned together using Rfam covariance models (Nawrocki and Eddy, 2013) (Figure 6 left), and a subsampled noisy MSA is obtained for each sequence in the library. For each subsampled noisy MSA, we either manually extract covariance features to pretrain our RSSM (Figure 1 D left), or employ a MSA-transformer (Rao *et al.*, 2021) to more rigorously represent a MSA and extract its evolutionary information (Figure 1 D right). Detailed pretraining strategies for these scenarios is elaborated in section 3.3.

B Training details

B.1 Supervision on intermediate feature maps

In addition to the final RNA contact map representation, RSSM generates a set of downsampled feature maps which have condensed information about potential long range basepairs within or between large patches of merged contact map areas. These features are eventually merged into the final RNA contact map representation through the horizontal shortcut and expansive path. Nevertheless, we provide auxiliary supervision to these intermediate feature maps in the hope of better extracting and integrating these information to improve the final RNA contact map prediction. Our auxiliary targets are in the form of basepairing intensity scores that can be obtained from the ground truth binary RNA contact map, as illustrated in Figure 2 B. The loss function can be expressed as follow:

$$\mathcal{L} = -\sum_{s=l,\lceil l/ds\rceil,\lceil l/2ds\rceil,\lceil l/4ds\rceil} w_s \sum_{i,j\in\text{triu_indices}(s\times s)} y_{ij}^{s\times s}\log M_{ij}^{s\times s} + (1-y_{ij}^{s\times s})\log(1-M_{ij}^{s\times s}) \tag{4}$$

where s denotes the spatial dimensionality of the full RNA contact map or downsampled feature map. and w_s are weights to balance the loss from different maps. When the map is in its full resolution, i.e. $s = l$, $y_{ij}^{s\times s}$ are binary labels, with $y_{ij}^{s\times s} = \mathbb{1}(\text{i forms a basepair with j})$. In its downsampled versions, for example $s = \lceil \frac{l}{ds} \rceil$,

$$y_{ij}^{s\times s} = \frac{1}{ds}\sum_{\substack{i'\in[i\cdot ds,(i+1)\cdot ds)\\ j'\in[j\cdot ds,(j+1)\cdot ds)}} y_{i'j'}^{l\times l} \tag{5}$$

considering that at most ds basepairs could be formed between two stretches of distant RNA segments of length ds, thereby rendering $y_{ij}^{s\times s}$ a measurement of basepairs concentration within a merged patch in the ground truth RNA contact map.

$M^{s\times s}$ contains basepairing probabilities or intensities predicted from the corresponding RNA contact map representation or downsampled feature maps. Each $M^{s\times s}$ employs a separate classification head implemented as a three-layer multi-layer perceptron (MLP), followed by sigmoid activation. Each fully-connected layer in the MLP is preceded by layer normalization, nonlinear activation and dropout.

B.2 Hyperparameters and training configurations

Table 3: Below are hyperparameters and training configurations for RSSM and MSA transformer. For the sake of training efficiency, we group RNA sequences into different ranges of length indicated in the "group sequence length" entry. For example, "180, 280, 380" indicates four length ranges that are $(0, 180]$, $(180, 280]$,$(280, 380]$ and $(380, 500]$. Each length group has a different batch size indicated by "group batch size".

RSSM	
base dimension (number of feature channels in the convolutional backbone)	64
initial downsampling patch size (ds)	20×20
number of attention heads in transformer modules	4
vision transformer local window size	20×20
activation function (used everywhere)	ReLU
MSA transformer	
hidden dimension	64
number of attention heads	4
number of row- and col-wise self attention layers	12
activation function (used everywhere)	ReLU
Common training configuration	
dropout ratio (used everywhere within RSSM and MSA transformer)	0.25
dropout ratio (used in classification head MLPs)	0.5
optimizer	Adam (Kingma and Ba, 2015)
w_l	0.7
$w_{\lceil l/ds \rceil}$, $w_{\lceil l/2ds \rceil}$,$w_{\lceil l/4ds \rceil}$	0.1
RSSM pretraining	
number of epochs	100
learning rate	1e-5
group sequence length	100, 200, 300, 400, 500
group batch size	32, 16, 8, 4, 2, 1
gradient accumulation size	128
learning rate warmup	160,000 steps
MSA transformer pretraining	
number of epochs	100
learning rate	1e-3
group sequence length	200, 300, 600
group batch size	4, 3, 2, 1
gradient accumulation size	128
learning rate warmup	160,000 steps
maximal subsampled MSA depth	$2^{15}/l$
RSSM finetuning	
number of epochs (single sequence based)	300
number of epochs (alignment based)	500
learning rate for the convolutional backbone	1e-3
learning rate for other components (e.g. vision transformers along the contractive and expansive path)	1e-5
group sequence length	180, 280, 380
group batch size	16, 8, 4, 1
learning rate warmup	320,000 steps
MSA-RSSM finetuning	
number of epochs	500
learning rate for the convolutional backbone	1e-3
learning rate for other components (e.g. vision transformers and MSA transformer)	1e-5
group sequence length	200, 300, 400
group batch size	4, 3, 2, 1
learning rate warmup	320,000 steps
maximal subsampled MSA depth	$2^{13}/l$

C RSSM generalization performance in external dataset

Table 4: Performance of RSSMs and baselines including SPOT-RNA and a variety of traditional RNA folding algorithms on short RNA sequences between 12 to 150 nts with or without pseudoknots in bprna-1m dataset and Rfam derived dataset.

	bprna-1m dataset (12-150 nts)						Rfam derived dataset (12-150 nts)					
	without pseudoknot (size: 1971)			with pseudoknot (size: 125)			without pseudoknot (size: 5656)			with pseudoknot (size: 1542)		
Model	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
IPknot												
LinearPartition-C	0.552	0.539	0.603	0.529	0.601	0.494	**0.726**	0.704	0.774	0.554	0.596	0.536
LinearPartition-V	0.535	0.487	0.638	0.466	0.479	0.471	**0.716**	0.656	0.813	0.502	0.492	0.524
ThreshKnot												
LinearPartition-C	0.553	0.563	0.581	0.503	0.631	0.445	**0.726**	0.725	0.756	0.500	0.616	0.444
LinearPartition-V	0.534	0.485	0.637	0.453	0.467	0.457	**0.716**	0.654	0.814	0.487	0.483	0.501
RNAfold	0.531	0.486	0.628	0.431	0.446	0.433	**0.716**	0.657	0.807	0.475	0.478	0.484
CONTRAfold	0.553	0.523	0.621	0.510	0.564	0.486	**0.728**	0.683	0.802	0.520	0.558	0.502
SPOT-RNA	0.640	0.604	0.714	**0.683**	0.687	0.695	0.664	0.649	0.705	**0.616**	0.649	0.602
RSSM_{d64}	**0.708**	0.685	0.755	**0.675**	0.705	0.669	0.664	0.639	0.713	0.578	0.601	0.571
$\text{RSSM}_{\text{d64,T}}$	**0.709**	0.686	0.756	0.659	0.705	0.642	0.676	0.653	0.722	0.578	0.607	0.566
$\text{RSSM}_{\text{d64,mix}}$	**0.701**	0.676	0.753	0.654	0.693	0.640	**0.718**	0.679	0.786	**0.617**	0.629	0.618
$\text{RSSM}_{\text{d64,mix,LPC}}$	**0.696**	0.673	0.746	0.653	0.697	0.634	**0.730**	0.689	0.800	**0.620**	0.634	0.619

We measure the generalization capability of our RSSMs on two recent dataset curated by Sato and Kato (2021).

- Bprna-1m dataset is extracted from the same online database (Danaee *et al.*, 2018) that was used earlier in Singh *et al.* (2019) to compile bprna training, validation and test sets. Nevertheless, having removed RNA sequences that share more than 80% similarity to those in the existing bprna dataset, the newer bprna-1m dataset can be used to measure the performance of our RSSMs on new RNA sequences from the existing RNA families.
- Rfam derived dataset is curated from the latest Rfam database (version 14.5) using consensus secondary structures as ground truth, and it only contains sequences from new RNA families that are absent in the bprna dataset. Therefore, Rfam derived dataset is employed to evaluate the generalization capability of our RSSMs in unseen RNA families, and to see if a balanced performance can be achieved between existing and new RNA families.

We follow the same evaluation scheme in Sato and Kato (2021) and compare our RSSMs to baselines including SPOT-RNA and a variety of RNA folding algorithms. RNAs in these two dataset are grouped into different types according to their length and the existence of pseudoknots, as shown in Table 4, Table 5 and Table 6. We include fours types of RSSM into the comparison, which are RSSM_{d64}, $\text{RSSM}_{\text{d64,T}}$, $\text{RSSM}_{\text{d64,mix}}$ and $\text{RSSM}_{\text{d64,mix,LPC}}$. Each one is a five-model ensemble:

- RSSM_{d64} is composed by directly trained models using bprna training set (as in in Table 1).
- $\text{RSSM}_{\text{d64,T}}$ is composed by finetuned models following pretraining (as in in Table 1).
- $\text{RSSM}_{\text{d64,mix}}$ is a mixture of directly trained and finetuned models, with its formula carefully chosen to balance its generalization performance in existing and new RNA families. We will elucidate its exact composition later in greater details.
- $\text{RSSM}_{\text{d64,mix,LPC}}$ is essentially the same $\text{RSSM}_{\text{d64,mix}}$, but with additional LinearPartition-V basepairing probabilities incorporated into our ensemble model as a crude form of leveraging biological priors.

For each RNA type in Table 4, Table 5 and Table 6, the best performing method or those within 2% difference of the best are marked with boldface. For the simplicity of analysis, we compare their performance by the F1 score. Across all RSSM series, a threshold of 0.3 is used for all RNA types in the bprna-1m dataset, and a threshold of 0.2 is used for all RNA types in the Rfam derived dataset.

RSSM_{d64} performs well on sequences shorter than 500 nts in the bprna-1m dataset, but generalize poorly on longer RNA sequences and new RNA families in the Rfam derived dataset.

Compared to a wide selection of baselines, RSSM_{d64} always emits better or comparable performance on sequences below 500 nts in the bprna-1m dataset, which represent the group of RNA sequences similar to its training set. On the other hand, in Rfam derived dataset and on longer RNAs, RSSM_{d64} exhibits extremely limited accuracy, especially compared to RNAfold which is a traditional thermodynamic model that is built on the decomposition of nested RNA

Table 5: Performance of RSSMs and baselines including SPOT-RNA and a variety of traditional RNA folding algorithms on medium RNA sequences between 151 to 500 nts with or without pseudoknots in bprna-1m dataset and Rfam derived dataset.

	bprna-1m dataset (151-500 nts)						Rfam derived dataset (151-500 nts)					
	without pseudoknot (size: 514)			with pseudoknot (size: 162)			without pseudoknot (size: 656)			with pseudoknot (size: 401)		
Model	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
IPknot												
LinearPartition-C	0.449	0.437	0.489	0.415	0.437	0.406	0.526	0.505	0.579	0.509	0.504	0.530
LinearPartition-V	0.423	0.375	0.511	0.437	0.410	0.479	0.521	0.451	0.645	0.470	0.418	0.547
ThreshKnot												
LinearPartition-C	0.450	0.473	0.463	0.413	0.478	0.375	**0.528**	0.557	0.543	0.500	0.546	0.482
LinearPartition-V	0.424	0.376	0.511	0.434	0.412	0.469	0.526	0.459	0.645	0.464	0.416	0.535
RNAfold	0.412	0.365	0.497	0.417	0.393	0.455	0.522	0.456	0.638	0.452	0.405	0.520
CONTRAfold	0.449	0.422	0.509	0.460	0.457	0.473	**0.539**	0.497	0.620	0.513	0.487	0.555
SPOT-RNA	0.464	0.484	0.481	0.485	0.556	0.440	0.460	0.495	0.456	0.476	0.534	0.441
RSSM_{d64}	0.543	0.543	0.570	**0.654**	0.670	0.651	0.460	0.446	0.497	0.492	0.493	0.503
$\text{RSSM}_{\text{d64,T}}$	**0.573**	0.558	0.618	**0.667**	0.666	0.679	0.487	0.459	0.546	0.488	0.478	0.508
$\text{RSSM}_{\text{d64,mix}}$	0.558	0.551	0.597	**0.658**	0.664	0.664	0.517	0.480	0.586	**0.537**	0.516	0.570
$\text{RSSM}_{\text{d64,mix,LPC}}$	0.557	0.553	0.593	**0.655**	0.665	0.657	**0.532**	0.495	0.602	**0.543**	0.523	0.576

secondary structures. RSSM_{d64} also falls short of CONTRAfold which similarly encodes strong inductive bias of RNA secondary structures through a generalized form of stochastic context free grammar. Both approaches rely on strong biological priors which cannot fully explain observed RNA secondary structure data, but they do provide a set of biologically well-grounded principles for RNA folding which can be extended to any unseen RNA families.

IPknot and ThreshKnot have also demonstrated superior performance over RSSM_{d64}, and these approaches can be understood as discretization techniques on top of basepairing probabilities in a predicted RNA contact map. We only apply IPknot and ThreshKnot on basepairing probabilities predicted by LinearPartition, which usually leads to better performance than other alternatives such as McCaskill's algorithm or CONTRAfold's native scoring system (Sato and Kato, 2021).

Compared to SPOT-RNA, our RSSM_{d64} has shown better performance on medium and long RNAs with pseudoknots in the Rfam derived dataset, while maintaining largely identical performance elsewhere, with the exception of short pseudoknotted structures.

In general, models with biological priors have consistently outperformed deep learning selections that are SPOT-RNA and RSSM_{d64}. This indicates that given limited amount of training data, existing deep learning approaches that are trained from scratch would not outperform traditional approaches in unseen RNA families or on longer sequences that are absent in the training set, unless more data is made available to properly represent these newer domains of RNA structures.

Transfer learning improves generalization capability across all dataset and RNA types.

Adding $\text{RSSM}_{\text{d64,T}}$ to the comparison, we observe that performance is improved in almost every setting, especially for long RNA sequences and unseen RNA families in the Rfam derived dataset. In particular, $\text{RSSM}_{\text{d64,T}}$ outperforms its direct learning counterpart RSSM_{d64} as well as the deep learning baseline SPOT-RNA on almost every RNA type in unseen RNA families, with the exception of short pseudoknotted sequences.

Nevertheless, the gap between $\text{RSSM}_{\text{d64,T}}$ and the variety of traditional RNA folding algorithms still exists. In an attempt to fully tap into the power of pretraining, and for the purpose of balancing the performance of our model between old and new RNA families, between shorter and longer RNAs, we compose a novel RSSM ensemble model with experts addressing different aspects of these generalization tasks, as opposed to having a simple ensemble of replicate models.

The idea of composing an ensemble with different hypotheses stems from a crucial observation on SPOT-RNA, which is originally an ensemble of five different deep learning architectures. While the whole SPOT-RNA ensemble has shown an impressive performance on the these generalization tasks, we find out that each single model performs much worse, which seems to suggest that ensembling diverse model hypotheses may lead to a more dramatic increase in the overall performance.

Therefore, we propose to build an ensemble with mixed hypotheses referred as $\text{RSSM}_{\text{d64,mix}}$. While we make no attempt at searching for new model architectures, we simply train our basic RSSM with different learning schedules. Our $\text{RSSM}_{\text{d64,mix}}$ is an ensemble of the following models, all trained to 300 epochs:

Table 6: Performance of RSSMs and baselines including SPOT-RNA and a variety of traditional RNA folding algorithms on long RNA sequences from 501 to a maximum of 4381 nts with or without pseudoknots in bprna-1m dataset and Rfam derived dataset.

	bprna-1m dataset (501-4381 nts)						Rfam derived dataset (501-951 nts)					
	without pseudoknot (size: 420)			with pseudoknot (size: 245)			without pseudoknot (size: 9)			with pseudoknot (size: 142)		
Model	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
IPknot												
LinearPartition-C	**0.437**	0.441	0.446	0.408	0.423	0.425	0.214	0.273	0.180	**0.463**	0.454	0.483
LinearPartition-V	0.382	0.348	0.439	0.347	0.312	0.421	0.302	0.305	0.301	0.411	0.344	0.513
ThreshKnot												
LinearPartition-C	**0.444**	0.480	0.425	**0.412**	0.470	0.396	0.197	0.321	0.148	**0.464**	0.501	0.445
LinearPartition-V	0.385	0.350	0.443	0.348	0.314	0.418	0.288	0.295	0.282	0.414	0.348	0.513
RNAfold	0.361	0.325	0.418	0.321	0.287	0.392	**0.393**	0.392	0.395	0.392	0.328	0.489
CONTRAfold	0.429	0.411	0.463	0.399	0.389	0.442	0.245	0.292	0.214	0.443	0.394	0.511
SPOT-RNA	0.283	0.318	0.263	0.278	0.320	0.265	0.168	0.244	0.129	0.360	0.384	0.344
RSSM_{d64}	0.259	0.376	0.209	0.281	0.384	0.246	0.168	0.206	0.143	0.380	0.358	0.409
$\text{RSSM}_{\text{d64,T}}$	0.358	0.384	0.345	0.371	0.395	0.377	0.196	0.235	0.168	0.419	0.383	0.465
$\text{RSSM}_{\text{d64,mix}}$	0.393	0.466	0.351	0.397	0.470	0.371	0.212	0.250	0.185	0.444	0.403	0.498
$\text{RSSM}_{\text{d64,mix,LPC}}$	0.423	0.505	0.375	**0.418**	0.502	0.386	0.214	0.259	0.183	**0.455**	0.417	0.505

- One directly trained model from RSSM_{d64}.
- One finetuned model from $\text{RSSM}_{\text{d64,T}}$.
- Two finetuned models with a smaller base learning rate at 1e-5, in order to help RSSM to retain more knowledge of the diverse RNA secondary structures it has seen during pretraining.
- One finetuned model but with subsampled Rfam pretraining data directly complementing each epoch.

The mixture in our $\text{RSSM}_{\text{d64,mix}}$ is selected with the purpose of balancing the information beneficial to the existing RNA families in bprna dataset, or to the newer RNA families in Rfam derived dataset. In particular, finetuned models with lower learning rates tend to retain more information from its pretraining, while complementing the finetuning process directly with subsampled Rfam pretraining data also boosts the generalization capability of our RSSMs on other RNA sequences in newer RNA families.

Ensembling diverse hypotheses improves generalization in new RNA families and helps to obtain more balanced performance between existing and new RNA families.

Our mixed ensemble $\text{RSSM}_{\text{d64,mix}}$ is able to maintain a more balanced performance between bprna-1m and Rfam derived dataset. In particular, improvement in Rfam derived dataset is the most substantial, with $\text{RSSM}_{\text{d64,mix}}$ outperforming all aforementioned deep learning selections across all RNA types by a large margin. Compared to the state-of-the-art traditional RNA folding algorithms, $\text{RSSM}_{\text{d64,mix}}$ demonstrates superior performance on short and medium pseudoknotted RNA sequences, which can be understood from the fact that RSSM naturally supports pseudoknot predictions, while RNAfold, CONTRAfold and LinearPartition cannot.

On the other hand, while $\text{RSSM}_{\text{d64,mix}}$ emits largely comparable or slightly lower performance on short and medium RNA sequences without pseudoknots, it falls short on longer RNA sequences not only in the Rfam derived dataset but also in the bprna-1m dataset, which is not unexpected given the fact that the maximal length of RNA sequences included throughout pretraining or finetuning is capped at 1000 or 500 nts respectively.

In bprna-1m dataset, $\text{RSSM}_{\text{d64,mix}}$ has largely comparable or slightly lower accuracy compared to the deep learning selections RSSM_{d64} or $\text{RSSM}_{\text{d64,T}}$. Since our Rfam pretraining data only contain predicted basepairing probabilities that cannot reflect pseudoknots or non-canonical basepairs in actual RNA secondary structures, introducing these computationally predicted structures into finetuning has slightly compromised the performance of $\text{RSSM}_{\text{d64,mix}}$ on real RNA secondary structures in the bprna and bprna-1m dataset. Nevertheless, $\text{RSSM}_{\text{d64,mix}}$ is still able to outperform all traditional RNA folding algorithms in both datasets.

Overall, given the balanced performance of $\text{RSSM}_{\text{d64,mix}}$ in the more challenging Rfam derived dataset, reaching or surpassing the state-of-the-art traditional RNA folding algorithms on short and medium RNA sequences, in addition to the fact that $\text{RSSM}_{\text{d64,mix}}$ has outperformed all other deep learning models on long RNA sequences by a large margin, it is fair to say $\text{RSSM}_{\text{d64,mix}}$ is the most useful RNA secondary structure prediction tool we have built so far.

Improving the prediction for longer RNAs by incorporating LinearPartition basepairing probabilities.

Throughout Table 6, our $\text{RSSM}_{\text{d64,mix}}$ has fallen short on longer RNA sequences compared to traditional RNA folding algorithms. While adding biological priors to our RSSM in the form of RNA secondary structure decomposition

rules represents a promising direction for improvement, here we propose a simple workaround to add LinearPartition predicted basepairing probabilities to our $\text{RSSM}_{\text{d64,mix}}$, considering the benevolent effect of model ensemble averaging which is discussed in section G.

We refer to this LinearPartition augmented approach as $\text{RSSM}_{\text{d64,mix,LPC}}$ and it has a systemically improved generalization capability in Rfam derived dataset, obtaining comparable or state-of-the-art performance across all RNA types compared to the traditional RNA folding algorithms, with the exception of long nonpseudoknotted RNA sequences which however only contains 9 examples. The improvement is particularly impressive for longer RNA sequences, for which $\text{RSSM}_{\text{d64,mix,LPC}}$ has obtained comparable or only slightly lower performance in bprna-1m dataset as well as Rfam derived dataset.

Combining LinearPartition with RSSM in fact slightly reduces its performance on short and medium RNA sequences in bprna-1m dataset, possibly due to the fact that bprna-1m dataset contains a high proportion of non-canonical basepairs which cannot be predicted by LinearPartition. This leads us to another important discussion later which is to highlight the differences between bprna-1m dataset and Rfam derived dataset, and their impact on traditional RNA folding algorithms and deep learning approaches.

Difference between bprna-1m and Rfam derived datasets.

The most remarkable difference between bprna-1m and Rfam derived dataset lies in their proportions of non-canonical basepairs, as shown in Figure 7. Bprna and bprna-1m datasets consistently feature up to 10% of non-canonical interactions, whereas Rfam derived dataset exclusively contains Watson-Crick and Wobble basepairs.

Another important observation can be made by noticing the performance difference between RNAfold in these two dataset. Since RNAfold is a traditional energy based model with experimentally measured thermodynamic parameters, it is rather unexpected to observe such a dramatic performance gap between RNA families in bprna-1m and Rfam derived dataset. In particular, from Rfam derived dataset to bprna-1m dataset, the performance of RNAfold consistently drops more than 20% on short and medium non-pseudoknotted RNA sequences. Similar observations can be made to CONTRAfold and two LinearPartition supported methods, with respect to all RNA types. In fact, none of these methods support non-canonical interactions. It should clarified, however, CONTRAfold is by design capable of accommodating non-canonical interactions but its default mode of prediction does not support non-canonical basepairs, otherwise its overall accuracy would suffer [3].

Overall, these observations seem to demonstrate that non-canonical interactions pose a major challenge to existing traditional RNA folding algorithms, which may suffice to explain their large performance gap between bprna-1m and Rfam derived dataset. Deep learning approaches such as SPOT-RNA and RSSM series, in comparison, naturally support all types of basepairing interactions and such behaviour may be unfairly penalized in Rfam derived dataset. Considering that RNAs ultimately fold into tertiary structures that fulfill their biological functions in the three dimensional cellular space, RNA secondary structures as mere intermediate representation without modeling these crucial non-canonical interactions may not be able to fully capture their underlying biological properties.

[3] See CONTRAfold manual "–noncomplementary"

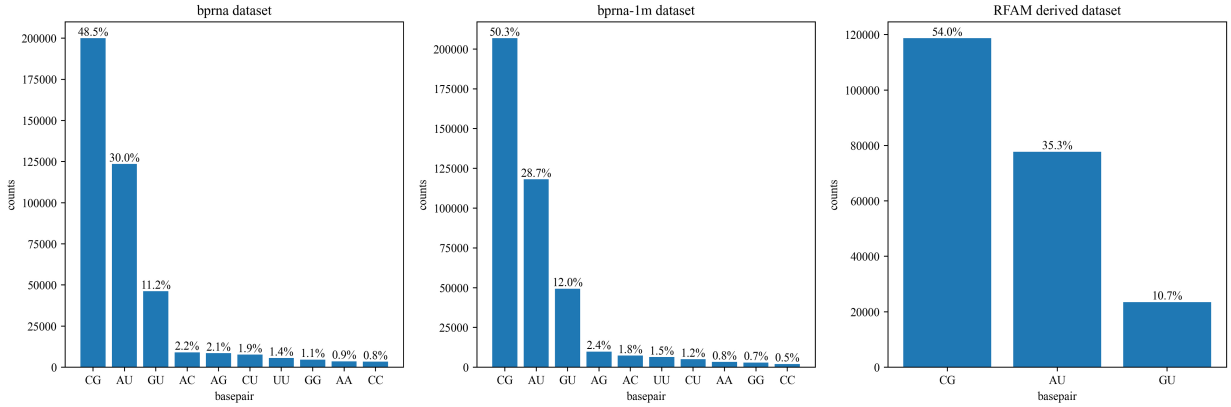

Figure 7: RNA secondary structures in bprna and bprna-1m dataset contain a high proportion of non-canonical basepairs which are up to 10%. Rfam derived dataset, one the other hand, does not contain any non-canonical basepairs at all.

D Sliding window based RNA secondary structure prediction for long RNA sequences

Consider an arbitrary basepair indexed by (i, j) and a window size of $L + 1$ and a step size. Assuming each time we slide the window to the 3' end by an offset of 1, then the basepairing probability averaged within all encompassed local windows can be expressed by:

$$M^p_{ij} = \begin{cases} \frac{\sum_{s=j-L}^{i} \text{RSSM}(x_{[s,s+L]})_{i-s,j-s}}{L-(j-i)+1} & \text{, if } L \geq j-i \\ 0 & \text{, otherwise} \end{cases} \tag{6}$$

while in practice for the sake of efficiency we use larger offsets as in Table 7.

Table 7: For bprna-1m dataset, we used a local window of size 1000 and a step size of 200. For Rfam derived dataset, we used a local window of size 500 and a step size of 100.

		bprna-1m dataset (501-4381 nts)						Rfam derived dataset (501-951 nts)					
		without pseudoknot (size: 420)			with pseudoknot (size: 245)			without pseudoknot (size: 9)			with pseudoknot (size: 142)		
Mode	Model	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Baselines	IPknot												
	LinearPartition-C	**0.437**	0.441	0.446	0.408	0.423	0.425	0.214	0.273	0.180	**0.463**	0.454	0.483
	LinearPartition-V	0.382	0.348	0.439	0.347	0.312	0.421	0.302	0.305	0.301	0.411	0.344	0.513
	ThreshKnot												
	LinearPartition-C	**0.444**	0.480	0.425	0.412	0.470	0.396	0.197	0.321	0.148	**0.464**	0.501	0.445
	LinearPartition-V	0.385	0.350	0.443	0.348	0.314	0.418	0.288	0.295	0.282	0.414	0.348	0.513
	RNAfold	0.361	0.325	0.418	0.321	0.287	0.392	**0.393**	0.392	0.395	0.392	0.328	0.489
Without sliding window	$\text{RSSM}_{\text{d64,mix}}$	0.393	0.466	0.351	0.397	0.470	0.371	0.212	0.250	0.185	0.444	0.403	0.498
	$\text{RSSM}_{\text{d64,mix,LPC}}$	0.423	0.505	0.375	**0.418**	0.502	0.386	0.214	0.259	0.183	**0.455**	0.417	0.505
With sliding window	$\text{RSSM}_{\text{d64,mix}}$	0.415	0.463	0.387	0.412	0.465	0.398	0.225	0.262	0.198	0.451	0.414	0.500
	$\text{RSSM}_{\text{d64,mix,LPC}}$	**0.438**	0.494	0.405	**0.427**	0.489	0.408	0.227	0.267	0.198	**0.463**	0.430	0.506

E RSSM performance on high resolution RNA secondary structures

Our RSSMs have shown comparable or superior performance compared to a diverse selection of baselines in bprna-1m and Rfam derived dataset. Here, we aim to offer a glimpse of their performance on a smaller set of high resolution RNA structures obtained with techniques such as X-ray crystallography and nuclear magnetic resonance (NMR).

These high resolution RNA secondary structures are originally curated by Singh *et al.* (2019), where the proposed SPOT-RNA is initially trained on bprna training set, then finetuned on PDB TR1 set [4], validated on PDB VL1 set and finally tested on PDB TS1 and NMR TS2 set. Here, we skip the step of finetuning our RSSMs on the high resolution PDB TR1 and PDB VL1 set. Instead, we directly measure the performance of our RSSMs in PDB TS1 and NMR TS2, as shown in Table 8.

[4] PDB standards for the Protein Data Bank

Table 8: Performance of RSSM series on high resolution RNA structures. Results of baselines are taken from SPOT-RNA paper, except IPknot and ThreshKnot which are new methods introduced to these dataset.

	PDB TS1 dataset (size: 67)			NMR TS2 dataset (size: 39)		
Model	F1	Precision	Recall	F1	Precision	Recall
IPknot						
LinearPartition-C	0.612	0.769	0.524	0.759	0.918	0.665
LinearPartition-V	0.610	0.722	0.538	0.763	0.913	0.674
ThreshKnot						
LinearPartition-C	0.604	0.808	0.499	0.755	0.926	0.658
LinearPartition-V	0.596	0.711	0.523	0.770	0.913	0.682
RNAfold	0.585	0.724	0.491	0.781	0.919	0.679
CONTRAfold	0.611	0.765	0.508	0.771	0.896	0.677
Knotty	0.603	0.742	0.508	0.783	0.913	0.686
ContextFold	0.621	0.811	0.503	0.739	0.910	0.623
SPOT-RNA	0.687	0.888	0.562	0.802	0.925	0.708
RSSM_{d64}	0.656	0.843	0.550	0.707	0.853	0.614
$\text{RSSM}_{\text{d64,T}}$	0.661	0.852	0.552	0.703	0.865	0.608
$\text{RSSM}_{\text{d64,mix}}$	0.694	0.857	0.595	0.746	0.913	0.649
$\text{RSSM}_{\text{d64,mix,LPC}}$	0.700	0.861	0.602	0.751	0.913	0.655

In both dataset of high resolution RNA structures, once again we observe benefits of pretraining RSSM with Rfam curated data, mixing diverse model hypotheses and integrating LinearPartition prediction in the ensemble, which are consistent with our observation made earlier in bprna-1m and Rfam derived dataset. In particular, having certain members in our RSSM ensemble to more thoroughly leverage the large quantity of computationally predicted RNA secondary structures also tangibly improves its overall performance on high resolution RNA secondary structures.

Compared to baselines in PDB TS1 dataset, our best performing RSSM variants such as $\text{RSSM}_{\text{d64,mix}}$ and $\text{RSSM}_{\text{d64,mix,LPC}}$ have outperformed SPOT-RNA without soliciting any additional supervision from high resolution RNA secondary structures. On the other hand, in NMR TS2 dataset our RSSMs are outperformed by SPOT-RNA and traditional RNA folding algorithms such as RNAfold and CONTRAfold, while such drawback does not hold statistical significance given the small amount of high resolution RNA structures in this dataset.

F Comparison of discretization strategies

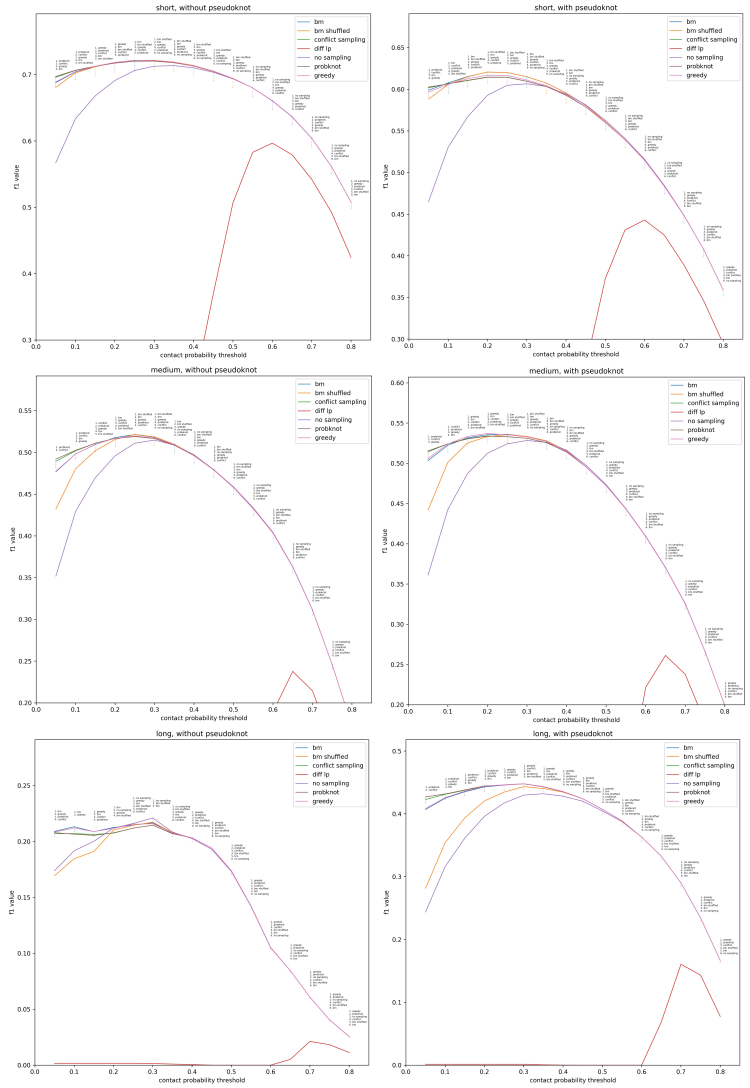

Figure 8: bm and bm shuffled stand for two beam search based discretization methods — the first one iterates basepairing probabilities sorted from highest to lowest, while the second one proceeds by randomly shuffled basepairing probabilities. diff lp stands for differential linear programming that is used by E2Efold. We compare all these methods across different RNA types in the Rfam derived dataset with varying thresholds. Methods that have less than 2% performance difference from the best at each threshold are ranked and marked. Zoom in on these figures is suggested.

Given a symmetric RNA contact map $M^p \in \mathbb{R}^{l \times l}$ densely populated by predicted basepairing probabilities, one last procedure is to convert this continuous representation to a discrete RNA secondary structure $Y \in \{0,1\}^{l \times l}$. Note that Y by definition is required to be symmetrical, but in practice this can be relaxed by only computing the upper triangular portion of Y. The discretization process proceeds by maximizing the cumulative basepairing probabilities above a certain threshold t, i.e. $\sum_{i=1\ldots l, j=i+1\ldots l}(M^p_{ij} - t)Y_{ij}$, while observing a set of constraints that governs the space of admissible RNA secondary structures, which usually are:

- no overlapping of basepairs (no multiplets), i.e. $\sum_{j=i+1\ldots l} Y_{i\cdot} \in \{0,1\}, \quad \forall i = 1 \ldots l$
- no sharp hairpin loops, i.e. $\forall\, Y_{ij} = 1, \quad |i-j| \geq 4$
- only canonical pairs or wobble pairs, i.e. $\forall\, Y_{ij} = 1, \quad x_i x_j \in \{AU, UA, CG, GC, GU, UG\}$

In this study, we only consider the first two types of constraints, which allows our RSSM to enrich an RNA secondary structure with any non-canonical interactions. Our default discretization strategy, which is used to report all performance in this paper unless otherwise specified, is a greedy approach that iterates M^p_{ij} above the threshold from highest to lowest and tries to form as many basepairs as possible without violating the constraints. Our default threshold is set to 0.5. We have also considered alternative strategies based on beam search which keep multiple trajectories of sampled pairs that are no longer greedily selected.

Throughout the lines of existing work, we identified some popular discretization methods, such as the one used by SPOT-RNA and SPOT-RNA2, which we refer to as conflict sampling since it essentially reduces multiplet clusters to nonoverlapping pairs. We also include the postprocessing technique known as differentiable linear programming which is employed by E2Efold [5], and the ProbKnot approach from ThreshKnot. Finally, we include a naive approach referred as no sampling, which basically pairs any two bases above the threshold while disregarding the constraints. We compare all these approaches together in Figure 8 using our $\text{RSSM}_{\text{d64,mix}}$ on the Rfam derived dataset and with varying thresholds.

As shown in figure 8, ProbKnot and conflict sampling perform better when we use a threshold below 0.1. Beam search based methods perform better on short and medium RNAs when the threshold varies between 0.2 and 0.35, while the difference between these two are not significant and the gain over our default greedy method is fairly marginal. The global maximum of F1 performance is usually attained at a threshold of 0.2 or 0.25.

Shuffled beam search and the naive no sampling methods are in particular susceptible to low threshold values. However, as the threshold grows larger than 0.6, pairing as many basepairs while overlooking the constraints as in the no sampling approach begins emitting marginally better performance, since fewer positive predictions remain after filtering through a larger threshold. At this point, F1 performance is mainly influenced by small variation in recall, since precision has become high but largely static. For longer RNAs without pseudoknots, the global maximum is markedly obtained by the naive approach, due to the large quantity of low basepairing probabilities predicted by RSSM as it is uncertain on these examples. In this case, variations in the recall also influence more on the harmonized F1 value.

Overall, we have shown that basepairing probabilities predicted by RSSM are amenable to a wide variety of discretization techniques, some of which are implemented in this work while others come from previous studies. In Rfam derived dataset, most discretization methods have performed comparably well on low threshold values, with the best overall performance usually obtained at a threshold of 0.2 or 0.25.

[5] While the method contains a larger amount of hyperparameters such as step sizes lr_{min}, lr_{max}, number of iterations, l_1 coefficient ρ and etc., we simply adopted their original implementation on github (link).

G Effect of model ensembling

Figure 9 and 10 show the perplexity of basepairing probabilities averaged within the upper triangular portion of predicted RNA contact maps, either from the ensemble model or one of its five replicates, versus length of the RNA. Each point indicates an RNA sequence in the Rfam derived dataset.

In every case, there is positive correlation between the length of RNA and the averaged perplexity, which means determining basepairs becomes more difficult as the length of RNA increases. However, such positive correlation in our ensemble model is much smaller compared to any of its replicate, which indicates that ensemble averaged RNA contact maps tend to ease the difficulty of sampling basepairs.

Together with our observation made in Figure 11, the benefit of model ensembling may possibly be explained by the trimming of outlier predictions that have high probabilities in only a few replicate models, while retaining more confident basepairs that are agreed upon by most replicates inside the ensemble.

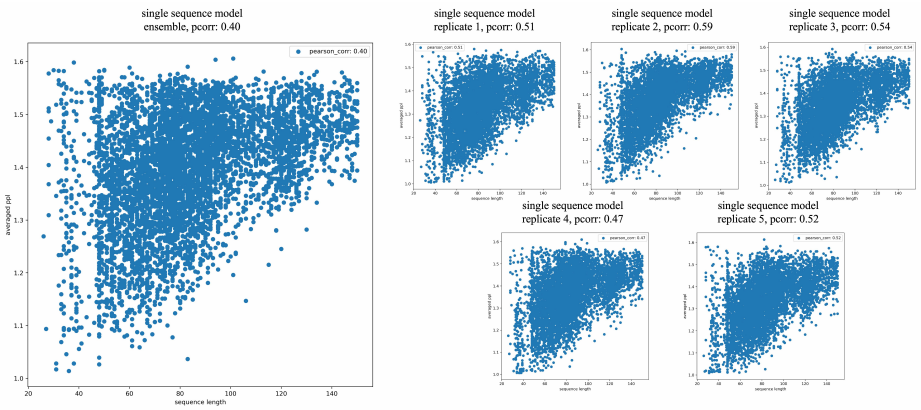

Figure 9: Using short RNA sequences without pseudoknot in the Rfam derived dataset. Each point indicates an RNA sequence.

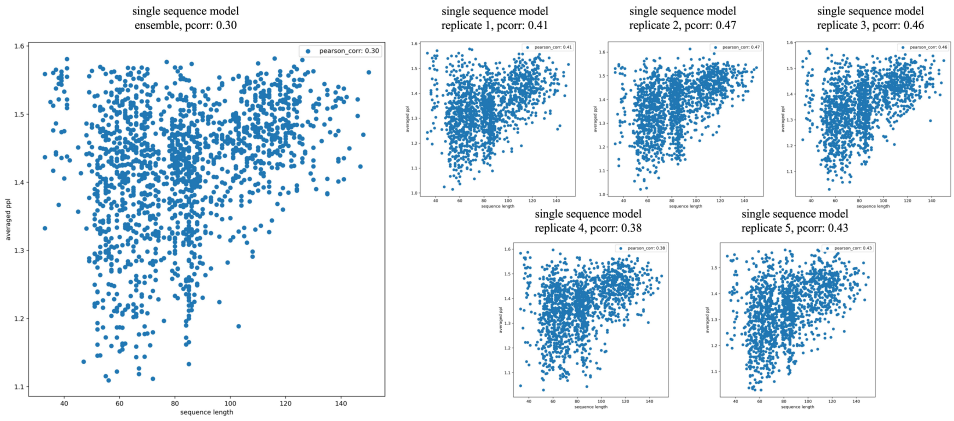

Figure 10: Using short RNA sequences with pseudoknots in the Rfam derived dataset. Each point indicates an RNA sequence.

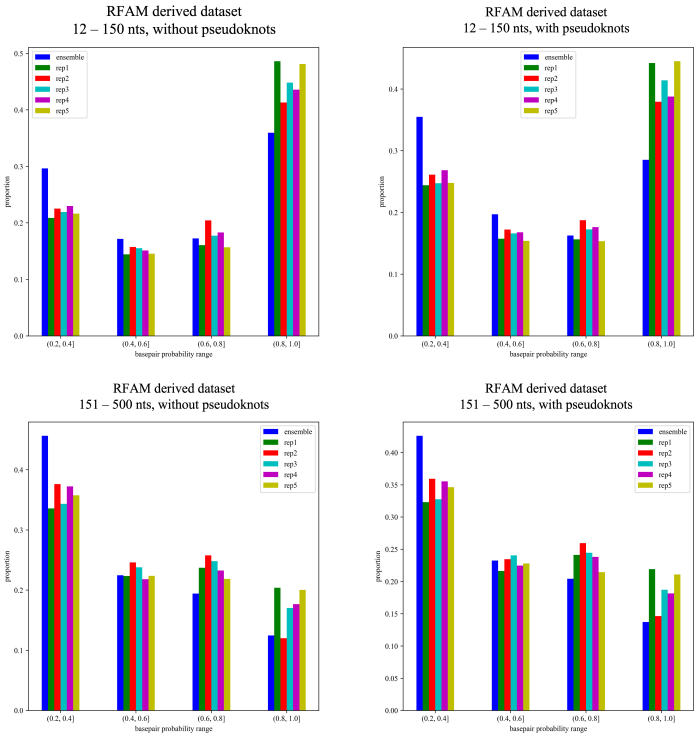

Figure 11: Proportion of predicted basepairing probabilities that lie within ranges such as $(0.2, 0.4]$, $(0.4, 0.6]$, $(0.6, 0.8]$ and $(0.8, 1.0]$, for RNAs in the Rfam derived dataset from short to medium length, with or without pseudoknots. Comparing ensemble averaged basepairing probabilities to those from single models, a general trend that is a reduction of large probabilities while accompanied by an increase in low probabilities is consistently observed across different RNA types.

H Transformer overfitting

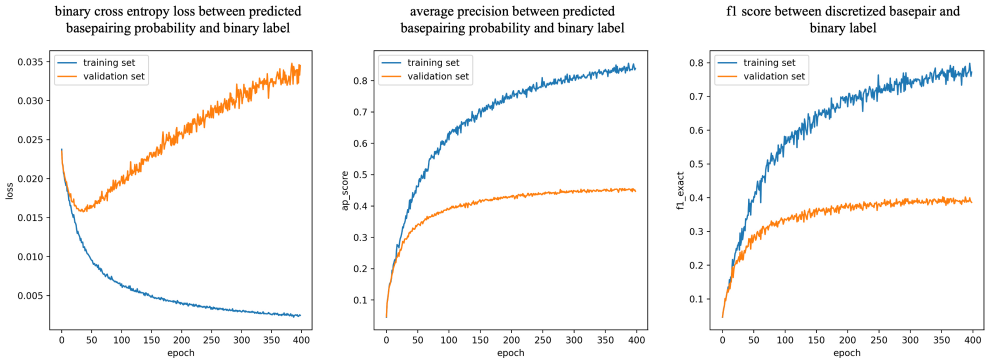

Figure 12: Training a simple transformer based model leads to severe overfitting in bprna dataset. This model contains five standard transformer encoder layers with base dimension of 64 and four self attention heads in each layer. Input RNA sequences are first embedded into vectors of dimension 64, with addition of learnable positional embeddings, before going through these transformer layers. Output from transformer layers are outer-concatenated to form contact maps (dimension 128), and basepairing probabilities are finally obtained with an MLP — linear(128, 128), ReLU and linear(128, 1). Dropout of 0.1 is used in transformer layers. Adam optimizer with a flat learning rate of 1e-4 is used to train the entire model. Figure clearly shows that a small transformer based model can still severely overfit the data, and the trend is only apparent when the overlap between training and validation set is properly removed as bprna dataset has ensured.

I Comparison of evolutionary information

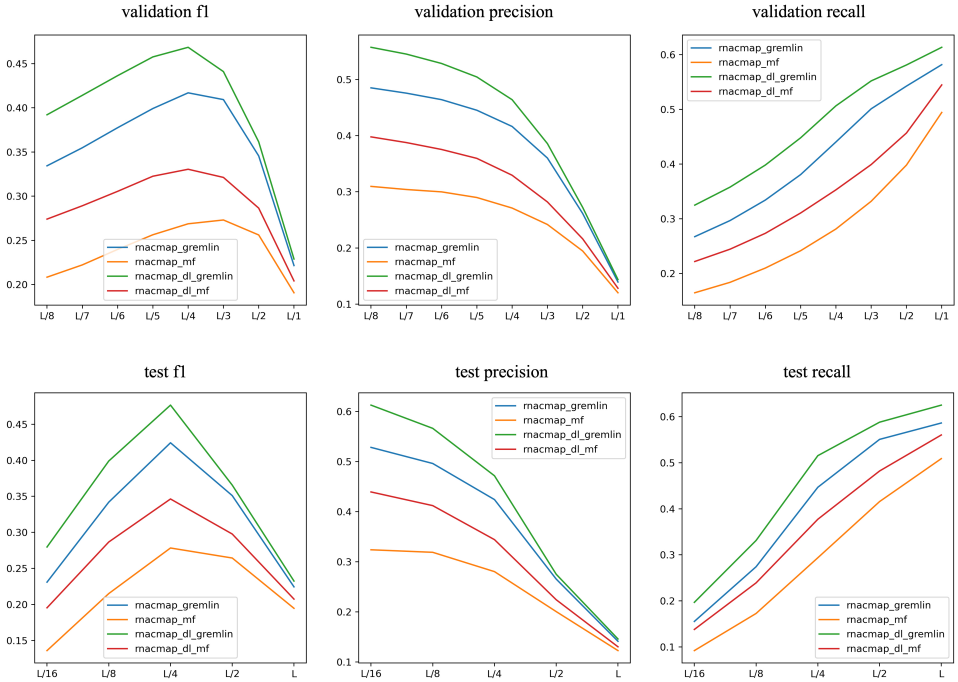

Figure 13: A comparison between two types of MSA with consensus secondary secondary structure either annotated by RNAfold (denoted by rnacmap) or one $\text{RSSM}_{\text{d64,T}}$ replicate (denoted by rnacmap_dl). Performance is measured on bprna validation and test set.

We extract direct coupling signals directly from those two types of MSA using either mean-field DCA (mfDCA) or pseudo-likelihood based DCA (known as gremlin). The strength of coupling between any two MSA columns, which may indicate basepairing given the fact that paired bases in RNA structures usually undergo evolutionary co-conservation, is a scalar measured by the norm of DCA parameters followed by average product correction (APC) (Morcos *et al.*, 2011). Then, we rank these coupling scores from highest to lowest, and indiscriminately selects the top L/16, L/8, L/4, L/2 or L as basepairs.

These figures clearly favour evolutionary sources with underlying consensus secondary structures annotated by our own deep learning tool instead of RNAfold. Structures annotated by our tools are in general more accurate than those from RNAfold, which may help to build more accurate covariance models inside the RNAcmap pipeline and recover more pertinent sequences from the NCBI nucleotide database that tend to be truly evolutionarily related homologs.

J Full performance of alignment based models in bprna dataset

Table 9: RSSM with evolutionary features compared to a variety of baselines. Setup is similar to single sequence based RSSM, except all alignment based models are trained 500 epochs. Subscript also indicates the type of consensus secondary structure annotated inside the RNAcmap pipeline, which leads to different MSAs hence consequently affecting the performance of our alignment based RSSMs.

		Validation set			Test set		
Mode	Model	F1	Precision	Recall	F1	Precision	Recall
Singe models	$\text{RSSM}_{\text{d64,rnafold}}$	0.737±0.003	0.746±0.003	0.746±0.004	0.733±0.003	0.742±0.002	0.742±0.004
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.745±0.002	0.763±0.005	0.748±0.005	0.740±0.004	0.755±0.009	0.745±0.005
	$\text{RSSM}_{\text{d64,dl}}$	0.739±0.001	0.760±0.002	0.738±0.003	0.742±0.001	0.764±0.002	0.738±0.002
	$\text{RSSM}_{\text{d64,T,dl}}$	0.745±0.001	0.767±0.004	0.744±0.003	0.747±0.002	0.769±0.002	0.743±0.004
Ensemble models	$\text{RSSM}_{\text{d64,rnafold}}$	0.750	0.769	0.752	0.746	0.766	0.747
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.758	0.788	0.754	0.752	0.779	0.749
	$\text{RSSM}_{\text{d64,dl}}$	0.747	0.776	0.741	0.749	0.779	0.741
	$\text{RSSM}_{\text{d64,T,dl}}$	0.752	0.781	0.747	0.754	0.782	0.747
End-to-end models	$\text{MSA-RSSM}_{\text{d64,T,rnafold}}$	0.782	0.796	0.786	0.777	0.793	0.776
Baselines	SPOT-RNA2-IL0	0.727	0.731	0.724	0.727	0.731	0.723
	SPOT-RNA2-IL1	0.738	0.779	0.701	0.734	0.780	0.693
	SPOT-RNA2-IL2	0.726	0.714	0.738	0.726	0.716	0.737
	RNAalifold (MEA)	0.548	0.687	0.496	0.552	0.695	0.498
	CentroidAlifold (MEA, noncanonical)	0.623	0.622	0.680	0.626	0.629	0.678
	IPknot (LinearPartition-C)	0.632	0.683	0.626	0.633	0.687	0.627

Here, we present the full result of our alignment based RSSMs. All our single and ensemble models are trained under similar settings as single sequence based models in Table 1, except all our RSSMs are trained 500 epochs, in addition to the integration of evolutionary information either in the form of extracted covariance feature or learnt end-to-end by MSA transformer.

The benefits of Rfam pretraining and model ensembling, once again, can be confirmed in Table 9. Different types of evolutionary sources that are MSAs with inherent consensus secondary structures either annotated by our own deep learning tool (i.e. one $\text{RSSM}_{\text{d64,T}}$ replicate) or RNAfold inside the RNAcmap pipeline also have an tangible impact on RSSM performance, and we will elaborate further on that effect in section K. We also included a few representative baselines that are SPOT-RNA2, RNAalifold, CentroidAlifold and IPknot.

SPOT-RNA2 is an alignment based deep learning model that integrates intermediate features extracted from a range of existing bioinformatics tools, such as basepairing probabilities from LinearPartition and pseudo-likelihood based DCA scores from gremlin, in addition to the position specific scoring matrix (PSSM) extracted from the original MSA. SPOT-RNA2-IL[0-2] represents three model architectures that have been directly trained on the bprna dataset, using MSAs annotated by SPOT-RNA which is an earlier single sequence based deep learning model as shown in Table 1.

We are able to show that by only using covariance features and MSAs annotated by RNAfold, our $\text{RSSM}_{\text{d64,rnafold}}$ can obtain comparable performance to the best SPOT-RNA2 variant, and we will show later in section K how this RSSM configuration is more robust to potential perturbations in MSAs. Rfam pretraining, model ensembling and using MSAs annotated by our own deep learning model have further consolidated the superiority of our RSSMs over SPOT-RNA2.

As inputs to traditional alignment based RNA folding algorithms such as RNAalifold, CentroidAlifold and IPknot (switched to alignment based consensus structure prediction mode), we extract a representative set of 2000 sequences sampled from each $\text{RSSM}_{\text{d64,T}}$ annotated MSA, since deeper MSAs can incur prohibitively long run time. For example,

under the current setting, it takes CentroidAlifold 38 CPU hours and IPknot 22 CPU hours to go over the entire bprna validation set which contains 1300 RNA sequences. Configurations of these tools are indicated inside parentheses. Once again, we have demonstrated the superiority of representation learning enabled approaches such as RSSMs and SPOT-RNA2 over traditional approaches in alignment based RNA secondary structure prediction paradigm.

In order to better integrate evolutionary information from MSAs to our RSSMs, we leverage a dedicated neural architecture that has been introduced as MSA transformer, which we have properly pretrained using Rfam deposited MSAs as part of our pretraining strategies delineated in section 3.3. During the course of finetuning in bprna dataset and on MSAs obtained with RNAcmap, as usual we outer-concatenate the representation of the target RNA sequence within the learnt MSA and extract all row-wise attention weights, which are then concatenated together and provided to RSSM as input that carries evolutionary signals in place of covariance features.

This strategy of combining MSA transformer and RSSM enables us to obtain a fully end-to-end deep learning model referred as MSA-RSSM, which is similar to how MSA transformer was originally used to perform supervised protein contact map prediction in Rao *et al.* (2021).

During the course of finetuning MSA-RSSM, we set the maximal length of randomly subsampled MSA depth at $m = 2^{13}/l$, for the sake of reducing computational overhead and improve training efficiency. At the inference stage, however, the type of MSA subsampling strategy can generate a large impact on the final performance of MSA-RSSM. Rao *et al.* (2021) originally showcased a diversity maximizing subsampling strategy and cautioned that subsampling larger MSA than what has been seen during training may degrade model performance, which we are able to confirm in Figure 14 A.

However, considering the fact that randomly subsampled MSAs may well induce comparable performance as mentioned in Rao *et al.* (2021) and again confirmed in Figure 14 B, and in order to fully tap into the rich evolutionary information covered by a potentially large MSA, we once again borrow the idea of ensemble averaging multiple RNA contact maps, but this time instead of soliciting contact prediction from different models, we stick to a single MSA-RSSM and each contact map is predicted from a randomly subsampled MSA.

In practice, for each MSA with original depth larger than the threshold $m = 2^{13}/l$, we randomly subsample it 10 times and ensemble average all the predicted RNA contact maps. Considering the stochastic nature of random MSA subsampling, we repeat this process 10 times over the entire dataset and register the mean performance in Table 9, while the standard deviation is reflected as shaded area in Figure 14 A. We use this MSA subsampling strategy to report MSA-RSSM performance in Table 9, and we are able to confirm its superiority over the diversity maximizing principle in Figure 14 A.

K Robustness of alignment based models

We study the robustness of our alignment based models from the following two perspectives:

1. The effect of evaluating properly trained RSSMs on a different type of MSA with underlying consensus secondary structure annotated by a different tool from the one used in training.
2. The influence of MSA depth on the performance of our RSSMs.

K.1 Swapping evolutionary sources

The logic behind the first angle is to understand how a properly finetuned alignment based RSSM would behave when it is given a different type of MSAs from those the model has actually seen during training, since there are no constraints on the type of evolutionary information that our models might receive in practice. Here, we only focus on the effect of different forms of consensus secondary structure annotation, although other influencing factors are also worth considering, such as MSAs with no consensus secondary structures at all (e.g. blastn seed alignment) similar to those used in protein structure prediction, which we will have to leave for future investigation.

We present the result in Table 10, which are experiments of swapping input evolutionary sources at the inference stage. Several important observations can be made. First, we notice that RSSMs trained with MSAs annotated by RNAfold have all performed better on $\text{RSSM}_{\text{d64,T}}$ annotated MSAs, and the performance gain is consistent across our single models, ensemble models as well as end-to-end MSA-RSSM model. However, RSSMs trained with deep learning annotated MSAs are conspicuously susceptible to the other type of MSAs, for which we hope to offer a possible explanation below.

Considering our previous observation in Figure 13 which establishes that $\text{RSSM}_{\text{d64,T}}$ annotated MSAs contain more useful co-conservation signals that reveal potential basepairs, RSSMs trained with deep learning annotated MSAs

Table 10: The performance of alignment based models using different types of evolutionary sources (MSAs). For example, $\text{RSSM}_{\text{d64,T,rnafold}\leftarrow\text{dl}}$ means an RSSM is trained using MSAs with underlying consensus secondary structures annotated by RNAfold, while the performance is reported with MSAs annotated by our own deep learning tool.

		Validation set			Test set		
Mode	Model	F1	Precision	Recall	F1	Precision	Recall
Singe models	$\text{RSSM}_{\text{d64,rnafold}}$	0.737±0.003	0.746±0.003	0.746±0.004	0.733±0.003	0.742±0.002	0.742±0.004
	$\text{RSSM}_{\text{d64,rnafold}\leftarrow\text{dl}}$	0.752±0.002	0.771±0.002	0.752±0.003	0.753±0.001	0.774±0.002	0.751±0.002
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.745±0.002	0.763±0.005	0.748±0.005	0.740±0.004	0.755±0.009	0.745±0.005
	$\text{RSSM}_{\text{d64,T,rnafold}\leftarrow\text{dl}}$	0.755±0.005	0.778±0.004	0.753±0.008	0.754±0.003	0.778±0.005	0.751±0.006
	$\text{RSSM}_{\text{d64,dl}}$	0.739±0.001	0.760±0.002	0.738±0.003	0.742±0.001	0.764±0.002	0.738±0.002
	$\text{RSSM}_{\text{d64,dl}\leftarrow\text{rnafold}}$	0.645±0.002	0.622±0.005	0.699±0.003	0.643±0.003	0.623±0.005	0.691±0.004
	$\text{RSSM}_{\text{d64,T,dl}}$	0.745±0.001	0.767±0.004	0.744±0.003	0.747±0.002	0.769±0.002	0.743±0.004
	$\text{RSSM}_{\text{d64,T,dl}\leftarrow\text{rnafold}}$	0.645±0.002	0.617±0.004	0.707±0.002	0.644±0.002	0.620±0.004	0.701±0.003
Ensemble models	$\text{RSSM}_{\text{d64,rnafold}}$	0.750	0.769	0.752	0.746	0.766	0.747
	$\text{RSSM}_{\text{d64,rnafold}\leftarrow\text{dl}}$	0.763	0.793	0.756	0.765	0.799	0.754
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.758	0.788	0.754	0.752	0.779	0.749
	$\text{RSSM}_{\text{d64,T,rnafold}\leftarrow\text{dl}}$	0.766	0.800	0.759	0.766	0.802	0.755
	$\text{RSSM}_{\text{d64,dl}}$	0.747	0.776	0.741	0.749	0.779	0.741
	$\text{RSSM}_{\text{d64,dl}\leftarrow\text{rnafold}}$	0.652	0.630	0.704	0.648	0.628	0.695
	$\text{RSSM}_{\text{d64,T,dl}}$	0.752	0.781	0.747	0.754	0.782	0.747
	$\text{RSSM}_{\text{d64,T,dl}\leftarrow\text{rnafold}}$	0.650	0.625	0.712	0.649	0.626	0.703
End-to-end models	$\text{MSA-RSSM}_{\text{d64,T,rnafold}}$	0.782	0.796	0.786	0.777	0.793	0.776
	$\text{MSA-RSSM}_{\text{d64,T,rnafold}\leftarrow\text{dl}}$	0.786	0.604	0.787	0.790	0.809	0.786
SPOT-RNA2 Ensemble	$\text{SPOT-RNA2}_{\leftarrow\text{rnafold}}$	0.546	0.479	0.676	0.550	0.485	0.676
	$\text{SPOT-RNA2}_{\leftarrow\text{dl}}$	0.617	0.560	0.726	0.624	0.566	0.733

may become overly optimistic and trustful on the covariance feature and such bias does not fare well when it meets a weaker form of evolutionary sources. Indeed, we have observed that RSSMs trained with $\text{RSSM}_{\text{d64,T}}$ annotated MSAs converged much faster than those using RNAfold annotated MSAs, which points to potential overfitting. On the other hand, when RSSMs are trained using a weaker evolutionary source that is annotated by RNAfold, our models may learn to become aware of its limitation and alert to potential spurious co-conservation signals that lie within.

Similar observation can be made for SPOT-RNA2 by evaluating this baseline on our own $\text{RSSM}_{\text{d64,T}}$ annotated MSAs which leads to $\text{SPOT-RNA}_{\leftarrow\text{dl}}$, as well as on MSAs annotated by RNAfold that produces $\text{SPOT-RNA}_{\leftarrow\text{rnafold}}$. SPOT-RNA2 was originally trained using SPOT-RNA annotated MSAs, which is another deep learning model that has outperformed RNAfold as we have shown earlier in Table 1. Therefore, SPOT-RNA2 has similarly demonstrated a lack of robustness due to the type of evolutionary source it had used for training originally.

In summary, experiments in Table 10 suggest that for the purpose of improving the robustness of alignment based RSSMs or any other deep learning tools in general, a weaker form of MSAs should be employed for training, which may help preserve the robustness of trained models on other type of MSAs. In particular, better performance may be achieved by leveraging MSAs annotated with a more powerful form of consensus secondary structure.

Additionally, we would like to comment on the robustness of SPOT-RNA2 which may reflect our point made earlier in the Introduction about frailty in software dependency. SPOT-RNA2 requires certain types of intermediate features as input, such as basepairing probability from LinearPartition and direct coupling scores from gremlin. We have prepared input to SPOT-RNA2 exactly in the same format, except we extract direct coupling scores from a recent deep learning based gremlin implementation[6] which runs substantially faster than the old gremlin release. SPOT-RNA2, on the other hand, is trained with the old gremlin release and it seems the performance of SPOT-RNA2 can be seriously perturbed by small nuances in the gremlin implementation. Since gremlin is a pseudo-likelihood based DCA method and its deep learning based implementation is gaining more traction, SPOT-RNA2's dependency on its older release will become more problematic as the old gremlin release is being replaced by the new one.

K.2 Effect of MSA depth on $\text{RSSM}_{\text{d64,T,rnafold}}$ and $\text{MSA-RSSM}_{\text{d64,T,rnafold}}$

The number of homologs found by RNAcmap can vary as it depends on the size of nucleotide database which is usually several hundreds of gigabytes (e.g. 560 GB for the NCBI database) and has been constantly growing. Therefore, in

[6]Deep learning based gremlin implementation.

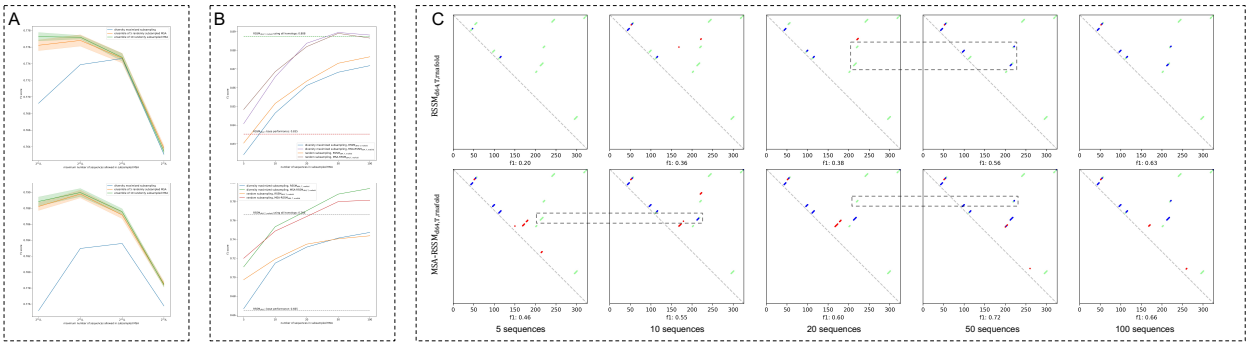

Figure 14: (A) Evaluating MSA-RSSM$_{\text{d64,T,rnafold}}$ in bprna test set and comparing different MSA subsampling strategies at various maximal MSA depth thresholds; top, using RNAfold annotated MSAs; bottom, using RSSM$_{\text{d64,T}}$ annotated MSAs. (B) Effect of MSA depth on covariance feature based RSSM$_{\text{d64,T,rnafold}}$ and end-to-end MSA-RSSM$_{\text{d64,T,rnafold}}$, using RSSM$_{\text{d64,T}}$ annotated MSAs; top, high resources setting where the original MSA N_{eff} is larger than 1000; bottom, low resources setting where the original MSA N_{eff} lies between 100 and 200. (C) Visualization of predicted RNA contact maps by RSSMs using MSAs at various depth. The same set of sequences (diversity maximized) are used at each column. Blue, green and red points denote true positive, false negative and false positive basepairs respectively.

reality one may end up with substantially fewer homologs when hosting larger nucleotide database proves infeasible given a small computational budget. Besides, one may also wish to stop scanning the nucleotide database after a sufficient amount of homologs has been found, since running RNAcmap over the entire database is a certain time costly and resources intensive process.

Here, we seek to better understand how MSA depth affects our alignment based ensemble models that are RSSM$_{\text{d64,T,rnafold}}$ and MSA-RSSM$_{\text{d64,T,rnafold}}$, using RSSM$_{\text{d64,T}}$ annotated MSAs. To properly simulate the scenario where we are limited to a small amount of homologous sequences, we identify two subsets of RNA sequences from bprna test set to form the test bed for: (1) a high resources setting where the original MSA N_{eff} is larger than 1000, and (2) a low resource setting where the original MSA N_{eff} is between 100 and 200. A total of 259 RNA sequences is included in the high resources setting and 157 RNA sequences is featured in the low resources setting. We gradually increase the number of homologs for each RNA, either diversity maximized or randomly sampled [7] from the original MSA, and show the performance of our models in Figure 14 B.

Both our alignment based models have outperformed the single sequence based ensemble baseline RSSM$_{\text{d64,T}}$ using only 5 sequences in the low resources setting. Under the high resources setting, RSSM$_{\text{d64,T,rnafold}}$ slightly underperforms when MSA only contains 5 sequences but the deficit is not significant. Both RSSM$_{\text{d64,T,rnafold}}$ and MSA-RSSM$_{\text{d64,T,rnafold}}$ benefit substantially from leveraging more RNA homologs in the MSA, as we gradually increase the MSA depth from 5 to 50. In particular, MSA-RSSM$_{\text{d64,T,rnafold}}$ outperforms RSSM$_{\text{d64,T,rnafold}}$ with the same amount of evolutionary information and exceeds the bar set by RSSM$_{\text{d64,T,rnafold}}$ using all homologs in the original MSAs, when a sufficient amount of 50 homologous sequences has been reached.

Figure 14 (B) demonstrates that our end-to-end MSA-RSSM is able to better utilize evolutionary information than the covariance feature based RSSM. Both our alignment based models are relatively robust to low MSA depth and as a matter of fact, outperform single sequence based baseline as long as a minimal amount of 5 sequences are reached in the low resource setting, or 10 sequences in the high resource setting. Figure 14 (C) shows a particular RNA example where increasing MSA depth can significantly improve the performance as it helps to recall more long range basepairs (off-diagonal blue points).

L Generalization of alignment based RSSMs

[7] We randomly sample each MSA 10 times and produce an averaged performance. Note, in this part we do not ensemble contact maps predicted from each randomly sampled MSA.

Table 11: Performance of alignment based models on medium RNA sequences between 151 to 500 nts in Rfam derived dataset. For each RNA, MSA-RSSM subsamples its MSA 10 times and ensemble average each predicted contact maps. This process is repeated 10 times across the entire dataset to calculate the mean and standard deviation for MSA-RSSM.

		Rfam derived dataset (151-500 nts)					
		without pseudoknot (size: 656)			with pseudoknot (size: 401)		
	Model	F1	Precision	Recall	F1	Precision	Recall
Singe sequence baselines	IPknot						
	LinearPartition-C	0.526	0.505	0.579	0.509	0.504	0.530
	LinearPartition-V	0.521	0.451	0.645	0.470	0.418	0.547
	ThreshKnot						
	LinearPartition-C	**0.528**	0.557	0.543	0.500	0.546	0.482
	LinearPartition-V	0.526	0.459	0.645	0.464	0.416	0.535
	RNAfold	0.522	0.456	0.638	0.452	0.405	0.520
	CONTRAfold	**0.539**	0.497	0.620	0.513	0.487	0.555
	$\text{RSSM}_{\text{d64,mix}}$	0.517	0.480	0.586	**0.537**	0.516	0.570
	$\text{RSSM}_{\text{d64,mix,LPC}}$	**0.532**	0.495	0.602	**0.543**	0.523	0.576
RNAfold annotated MSAs	RNAalifold (MEA)	0.421	0.583	0.382	0.489	0.643	0.420
	CentroidAlifold (MEA)	0.497	0.536	0.508	0.505	0.558	0.481
	IPknot (LinearPartition-C)	0.495	0.555	0.481	0.477	0.588	0.424
	gremlinDCA (top L/4)	0.286	0.271	0.317	0.279	0.274	0.287
	mfDCA (top L/4)	0.118	0.116	0.126	0.080	0.079	0.082
	$\text{RSSM}_{\text{d64,rnafold}}$	0.561	0.540	0.611	0.596	0.582	0.623
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.559	0.537	0.618	0.613	0.592	0.645
	$\text{MSA-RSSM}_{\text{d64,T,rnafold}}$	**0.570**$\pm$0.001	0.539$\pm$0.001	0.634$\pm$0.001	**0.662**$\pm$0.000	0.638$\pm$0.001	0.701$\pm$0.001
$\text{RSSM}_{\text{d64,mix}}$ annotated MSAs	RNAalifold (MEA)	0.405	0.617	0.347	0.484	0.713	0.392
	CentroidAlifold (MEA)	0.504	0.585	0.491	0.525	0.632	0.474
	IPknot (LinearPartition-C)	0.503	0.580	0.477	0.500	0.632	0.436
	gremlinDCA (top L/4)	0.294	0.273	0.331	0.340	0.334	0.331
	mfDCA (top L/4)	0.128	0.124	0.140	0.123	0.122	0.140
	$\text{RSSM}_{\text{d64,rnafold}}$	0.564	0.550	0.607	0.636	0.625	0.657
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.563	0.544	0.618	0.634	0.616	0.664
	$\text{MSA-RSSM}_{\text{d64,T,rnafold}}$	**0.581**$\pm$0.001	0.550$\pm$0.001	0.646$\pm$0.001	**0.677**$\pm$0.001	0.654$\pm$0.001	0.716$\pm$0.001

Table 12: Performance of alignment based models on long RNA sequences between 501 to a maximum of 951 nts in Rfam derived dataset. For each RNA, MSA-RSSM subsamples its MSA 10 times and ensemble average each predicted contact maps. This process is repeated 10 times across the entire dataset to calculate the mean and standard deviation for MSA-RSSM.

		Rfam derived dataset (501-951 nts)					
		without pseudoknot (size: 9)			with pseudoknot (size: 142)		
	Model	F1	Precision	Recall	F1	Precision	Recall
Singe sequence baselines	IPknot						
	LinearPartition-C	0.214	0.273	0.180	**0.463**	0.454	0.483
	LinearPartition-V	0.302	0.305	0.301	0.411	0.344	0.513
	ThreshKnot						
	LinearPartition-C	0.197	0.321	0.148	**0.464**	0.501	0.445
	LinearPartition-V	0.288	0.295	0.282	0.414	0.348	0.513
	RNAfold	**0.393**	0.392	0.395	0.392	0.328	0.489
	CONTRAfold	0.245	0.292	0.214	0.443	0.394	0.511
	$\text{RSSM}_{\text{d64,mix}}$	0.212	0.250	0.185	0.444	0.403	0.498
	$\text{RSSM}_{\text{d64,mix,LPC}}$	0.214	0.259	0.183	**0.455**	0.417	0.505
RNAfold annotated MSAs	RNAalifold (MEA)	0.015	0.111	0.008	0.330	0.429	0.300
	CentroidAlifold (MEA)	0.063	0.183	0.042	0.384	0.426	0.377
	IPknot (LinearPartition-C)	0.152	0.322	0.103	0.367	0.481	0.316
	gremlinDCA (top L/4)	0.131	0.143	0.122	0.156	0.146	0.173
	mfDCA (top L/4)	0.092	0.100	0.086	0.020	0.019	0.028
	$\text{RSSM}_{\text{d64,rnafold}}$	0.216	0.319	0.165	0.497	0.475	0.528
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.173	0.301	0.125	**0.537**	0.513	0.570
	$\text{MSA-RSSM}_{\text{d64,T,rnafold}}$	0.209±0.006	0.304±0.013	0.165±0.004	0.512±0.001	0.457±0.001	0.585±0.002
$\text{RSSM}_{\text{d64,mix}}$ annotated MSAs	RNAalifold (MEA)	0.014	0.111	0.007	0.338	0.475	0.297
	CentroidAlifold (MEA)	0.066	0.235	0.044	0.400	0.467	0.380
	IPknot (LinearPartition-C)	0.171	0.355	0.116	0.380	0.498	0.324
	gremlinDCA (top L/4)	0.085	0.092	0.080	0.141	0.132	0.151
	mfDCA (top L/4)	0.055	0.058	0.051	0.016	0.015	0.023
	$\text{RSSM}_{\text{d64,rnafold}}$	0.182	0.258	0.141	0.526	0.511	0.547
	$\text{RSSM}_{\text{d64,T,rnafold}}$	0.167	0.263	0.124	**0.552**	0.532	0.579
	$\text{MSA-RSSM}_{\text{d64,T,rnafold}}$	0.201±0.007	0.301±0.011	0.157±0.006	0.540±0.001	0.487±0.001	0.611±0.002