INTRODUCTION

RNA homology comprises the measurement of sequence similarity between RNA molecules towards establishing evolutionary relatedness between species and predicting functions of novel molecules. Homology search is often the first step in studying a new RNA molecule, and has greatly advanced the study of non-coding RNA (ncRNA) and the processes they are implicated in in the cell. To effectively compare RNA molecules, both the primary sequence conservation and secondary structure of the molecules need to be taken into account, as much of the RNA function is dictated by its structure. While current state-of-the-art homology tools are able to model sequence and structure efficiently, they are unable to model important structural motifs called pseudoknots, thus possibly making the models less sensitive to finding homologous sequences. Deep neural networks (NNs) have been increasingly used in biological problems showing high performance. We sought to explore NNs as potential candidates for a homology tool that could be able to fill the gap in the RNA homology search field. For a NN to be able to predict RNA homology, it is important that it is able to model RNA structure arising from long range interactions between nucleotides in base pairs. In this work, we explore different NN architectures and various factors involved in training NNs that influence whether a NN is able to learn RNA secondary structure; our goal is to gain insights towards making an effective deep learning based RNA homology tool.

RNA homology

The discovery and increased study of ncRNAs has elucidated the importance of RNA in many cellular pathways. Before it was discovered that there were different types of RNA that played different roles, it was observed that most RNA in the cell was located in particles that came to be known as ribosomes, and that these particles had to have some relation to protein

synthesis (Crick, 1958). Later, from the observation that there were other RNA molecules that had rapid turnover, the discovery of messenger RNA (mRNA) was made, identifying that RNA served a central role in the coding of proteins. Slowly more RNA types outside of mRNA (and later transfer RNA) were found, but it remained unclear as to what functions they served and to whether they had catalytic abilities. In 1982, Tom Cech's discovery of a self-splicing intron, catalyzed solely by a ncRNA was revolutionary and propagated the idea that ncRNAs were abundant molecules in the cell with important functions. Many types of ncRNAs have been discovered with unique function arising often from intricate structure. For the duration of this work, the term RNA will refer to ncRNA or, more broadly, RNA that perform functions in the cell, with the idea that these molecules have interesting structures.

Homology has been instrumental in the study of newly found RNA molecules and has helped elucidate important biological pathways that rely on functional RNAs (Esteller, 2011). The central idea behind homology is that polynucleotides and polypeptides carry information about the evolutionary history of an organism, and that information is found in regions of molecules that are conserved across evolution, suggesting their importance and perhaps functional implication. This idea set off the advent of molecular phylogeny (Zuckerkandl and Pauling, 1965). Using these ideas in studying homologous ribosomal RNA sequences, Woese and Fox discovered archaea as a distinct, third domain of the tree of life, showing how important the study of RNA homology is to advancing our understanding of phylogeny (Woese and Fox, 1977).

To find regions of conservation, homology tools largely construct alignments of sequences. While pairwise alignments between two sequences at a time provide some similarity information, the construction of multiple sequence alignments (MSAs) provides more general information about well conserved regions across evolutionary time. Here,

multiple sequences are compared to identify regions of similarity, and these regions are aligned such that columns of sites in the alignment reflect conservation information (Figure 1). MSAs can be used to classify groups of homologous RNA as an RNA family. Furthermore, given a novel molecule, statistical comparisons between the sequence and different MSAs provide information as to the classification and thereby possible function of the molecule.

For RNA homology search specifically, it is important to model sequence and structural conservation (Durbin, 1994). In constructing an RNA MSA, aligning only the primary sequence by looking for similar single nucleotides across sequences will provide some information as to short conserved nucleotide sequences (often called motifs) sequence. However, this will lose any structural information, as the conservation of base pairs arises as covarying nucleotides; i.e. while a single site will not have a conserved nucleotide, pairs of sites will conserve base pairing across evolution by compensating substitutions. Figure 1 demonstrates this in a toy MSA, showing sites that don't show conservation down a single nucleotide column, but retain base pairing across all sequences suggesting a conserved base pair. A homology tool must be able to model these covariations as two RNA molecules with highly different sequences could actually be evolutionarily related by a conserved secondary structure.

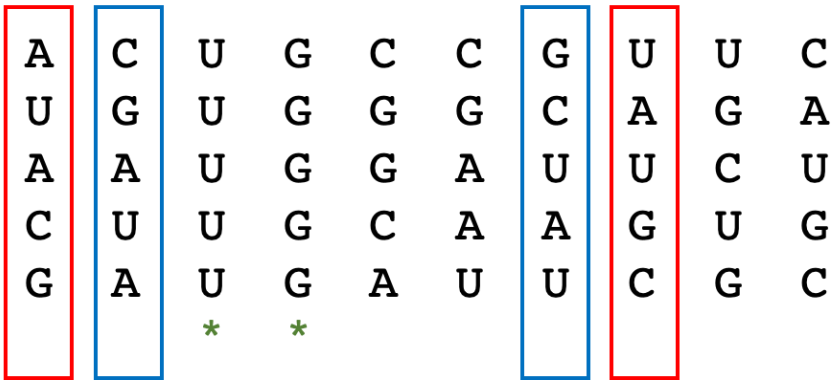


Figure 1: MSA of toy RNA sequences which have conserved base pairs shown by columns with covarying nucleotides that conserve base pairings. Sites are aligned in columns. Proper alignment of sequences shows the evolutionary conservation of nucleotides across different sequences at similar sites. Columns 3 and 4 show conserved single nucleotides, in which down the column a similar base pair is frequently observed. Two base pairs are shown between sites 1,2 and sites 7,8. These columns individually do not show conservation, but the pairs of nucleotides show conserved base pairings.

One of the most powerful tools used for RNA Homology search is Infernal, which uses a Covariance Model (CM) – a type of stochastic context-free grammar (SCFG) – at its core (Nawrocki and Eddy, 2013, Durbin et al., 1998). SCFGs are a tool borrowed from computational linguistics, that models RNA sequences as sentences derived from a formal grammar (Sakakibara et al., 1994). A grammar is effectively a set of production rules which define the derivation of sequences through the emission of individual nucleotides as well as pairs of nucleotides. Effectively, this allows the model to treat the RNA as a structured sentence instead of a linear sequence, and facilitates the modeling of interacting nucleotides. CMs are able to build a probabilistic representation of an RNA MSA, learning both conserved sequence motifs and base pairs. The base pair information can be used to make predictions on folding patterns and the conserved structure of a homologous family of RNAs.

Despite the efficiency and accuracy of CMs, a major drawback arises from the inability of SCFG based models to model non-nested base pairs. RNA base pairs can either be nested or non-nested (Figure 2). If two base pairs exists between nucleotides *a-b* and *i-j,* it

is said that they are nested if $a<i<j<b$ (that is to say that a is situated before i in the sequence before j and so on). However, if the nucleotides are arranged $a<i<b<j$, the base pairs are said to be non-nested. In other words, an unclosed base pair is situated between two base paired nucleotides (Koessler et al., 2010, Schirmer et al., 2014). Structural motifs such as hairpin stems and bulges arise from nested base pairs and are the most common source of secondary structure in RNA sequences. In the building of an SCFG model, the sequence profile is constructed from the outside of the sequence inwards. This enforces the constraint that base pairs must be nested within each other. Lengths of non-nested base pairs construct structural motifs called pseudoknots. ~5% of conserved base pairs across all RNA sequences are pseudoknotted and in many molecules, the pseudoknot fold is crucial for the RNA function. Such molecules include many viral ribozymes and indeed the group I self splicing introns discovered by Cech mentioned earlier (Staple and Butcher, 2005).

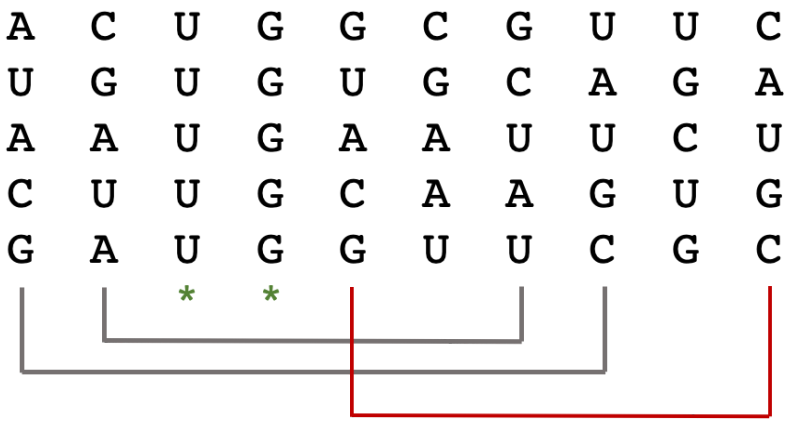


Figure 2: MSA of toy RNA sequences with columns containing covarying nucleotides that conserve base pairings. Columns 3 and 4 show conserved single nucleotides. Two nested base pairs are shown between sites 1,2 and sites 7,8. A non-nested base pair is also seen between the inner loop (site 5) and site 10. An SCFG would not be able to model the non-nested base pair and thus would not account for it in making predictions of sequence similarity.

Due to the ubiquity of pseudoknot motifs, it would be useful for homology tools to be able to model non-nested base pairs. Extensive progress in building RNA alignment databases has been made possible by the efficiency of CMs. Many RNAs that contain both non-nested and nested base pairs have been studied using CMs under the assumption that modelling the nested base pairs is sufficient for homology classification. However, the greater the number of non-nested base pairs in a structure, the less sensitive a CM is to finding homologous RNAs. The non-nested base pairs provide added information as to the intricacy of a conserved structure, and thus only looking at nested base pairs might over- or underestimate the sequence similarity between RNAs.

Many tools exist for the purpose of predicting and modelling RNAs containing pseudoknots (Schirmer et al., 2014). RNA folding programs that incorporate thermodynamic measurements of possible base pairs and structural motifs are used on a single sequence basis to find optimum structures that can contain pseudoknots. The drawback still remains that these programs are often very slow, but more importantly, folding single sequences at a time gives less general information about a conserved evolutionary structure. This makes it more difficult to incorporate these models into homology search tools to construct multiple sequence alignments at the scale required.

Deep Neural Networks

We propose that Deep Neural Networks have great potential to be incorporated into a homology search tool to address this remaining gap. Neural Networks (NNs) comprise a field of models that have become extensively powerful in fields such as machine vision and natural language processing (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2014). These models are able to learn a functional mapping between input data and output

(Sonoda and Murata, 2017; Cybenko, 1989). As opposed to traditional machine learning methods, the features describing the data do not need to be predefined in the input. This removes the strong assumptions that need to be made about the data. NNs are able to learn a hierarchical representation of the data and autonomously learn the features in the data necessary for prediction. A popular method to train NNs is called supervised learning , in which a model is trained to classify inputs given a set of training targets. During training, the model learns a representation of the training data with the intent of generalizing to unseen test data.

NNs have increasingly been used in genomic applications with high performance. For example, Alipahani et al trained an NN to predict the specificities of RNA binding proteins to different RNA sequences, with their model outperforming existing non-deep learning models in predictive power (Alipanahi et al., 2015). Furthermore, Zhou and Tronskaya applied NNs to predicting chromatin binding features given diverse sequence data, again performing powerfully. They showed that their NN was able to learn the effects of non-coding functional variants at a single-nucleotide resolution (Zhou and Troyanskaya, 2015). Works like these have been able to show that NNs are able to learn nucleotide motifs in sequence data at high resolution. This suggests an NN could also be trained to learn conserved primary sequence motifs, one of the features in RNA sequences necessary for homology search.

However, learning structural information from sequence data is arguably a more difficult task than learning sequence motifs. RNA structure mostly consist of base pairs which are effectively exclusive-or binary functions (XOR problems), as it requires recognition of a purine with a pyrimidine, but not both of either (and there is added complexity with the specificity of which purine pairs with which pyrimidine). Interestingly,

the XOR problem played a large role in the history of artificial intelligence research. In 1969, Minsky and Papert showed that a single perceptron (one of the first stages of neural network machinery) was unable to solve an XOR problem as it was only able to fit linear functions in data, with XOR problems being non-linear classifications (Minsky, 1969). This led to what became known as an "AI winter" in which NN research was significantly slowed due to the belief that these models would not be able to solve simple, but important classification tasks. It wasn't until it was shown that a multilayer perceptron (MLP), a network which uses sets of perceptrons stacked in layers, could solve the XOR problem easily, that NN research recommenced eventually producing the complex NN models that we have today (it should be noted that this revival was also facilitated by the backpropagation algorithm which made training these multi layer networks possible (Rumelhart et al., 1986)). While these models were not created to model RNA structure, it is interesting that these fulcrum questions that heavily influenced the development of NN machinery, are intrinsically linked to the task of learning base pairs.

Knowing this, it is probable that a NN would be able to learn base pairs from sequence data. The power of these models (and indeed what allowed MLPs to overcome the solve the XOR problem) is their ability to perform multiple different linear functions with non-linear activations and combine these to build representations of data with diverse features; this gives NNs the flexibility to independently learn complex information from data. This suggests to us that these models would also be able to learn non-nested base pairs, making NNs a good candidate for a more sensitive homology tool.

Interpreting Neural Networks

Before developing a model for a task, it is important to verify that it is learning the information desired, and not an alternate or incorrect representation of the data. One major difficulty with NNs is that they are difficult to interpret, in part due to the scale of these models - architectures can be easily constructed to contain 1000s - 100,000s of parameters. As a result, NNs are often used as black boxes predictive tools. Nevertheless, they have performed exceptionally well at classification and prediction tasks. However, as is a common problem with models containing many parameters, there is a risk that the model might be overfitting to the data. For example, if an NN is trained on sequence data for a classification task, it is likely that it will perform well with high accuracy, but that does not directly imply the model has learned the interesting biological features of the sequences. This is important for a homology search tool: if an NN is trained on an RNA family, it must learn the characteristic conserved motifs and structure of that family, instead of memorizing the training data, to be able to accurately predict homology of unseen sequences.

There has been recent progress on methods for interpreting NN models in the field of computer vision. One such class of methods is called saliency analysis, in which important features of the data are identified by taking the gradient of the output prediction with respect to the inputs (Simonyan et al., 2013). Put simply, the input data is systematically changed by a small amount and the effect on the output prediction is monitored. If changing a particular character in the input produces a large effect on the prediction, this implies that this character is an important feature. This has proven effective in image classification where input pixel data is continuous, and gradients in output are meaningful. For genomics, sequence data comprises discrete character values (nucleotides) making gradients with respect to the input less interpretable. Thus an analogous method that maintains the discrete quality of the input

data is first order *in silico* mutagenesis (Figure 3). Here, nucleotides in the sequence are mutated *in silico* and these mutant sequences are tested on a trained model to retrieve prediction scores for the mutants. The deviation of the mutant prediction from wild type provides insight into which nucleotides in the sequence are important. This technique has been used to show that NNs can learn binding protein motifs in regulatory sequences, by recapitulating the known motif for a given binding protein (Alipanahi et al., 2015). This method identifies the importance of individual positions, and applied to an NN trained on an MSA, can be used to verify if an NN is learning first order conserved sequence motifs.

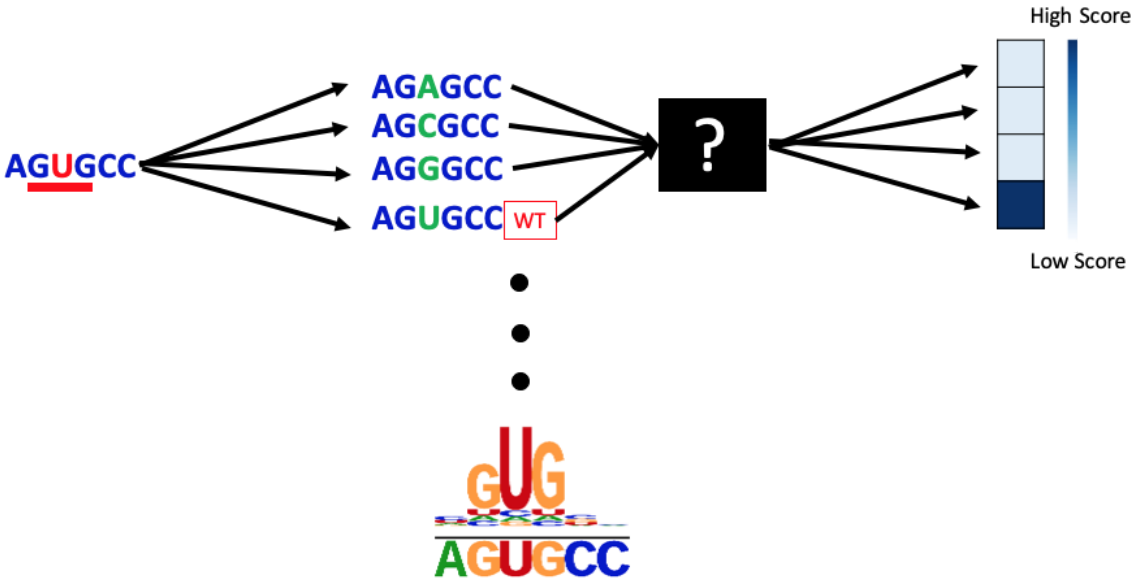


Figure 3: Schematic for First Order Mutagenesis on a toy sequence with conserved motif (GUG). For a selected nucleotide site, mutant sequences are generated *in silico* with every possible mutation at that site. These mutant sequences are tested using a trained NN producing a score for each mutant. If the NN has learned that the nucleotide is important for function, it should give the highest score for the WT nucleotide (U). We repeat this for every site in the sequence producing a mutation score vector for each site. These scores can be visualized as saliency logos depicting the proportional importance of each nucleotide in the sequence. These logos show what the NN has learned about the data and can be used to verify if the model is learning relevant biological features.

However structural information, namely base pairing, is a second order feature of the data, thus a different method is needed that is able to verify if a model is learning interactions between nucleotides. While we know that NNs can solve XOR problems (and therefore base pairs), these interactions manifest with complex patterns in RNA sequences. As stated, base pairing follows particular patterns of interaction nucleotides and, furthermore, these interactions occur over long ranges in RNA, with multiple base pairs present in alignments with different degrees of conservation. If we train an NN on sequence data, to verify if it is effectively learning this base pairing information we need second order interpretation methods.

Research goals

We propose that NNs are a good potential candidate for a homology tool. However, before such a tool can be developed, the first step is to understand the extent to which an NN can learn RNA secondary structure from homologous sequences. In particular, it is necessary to verify if these models can learn non-nested base pairs which would make them more sensitive than current methods. The goal of this work is to gain insight into whether an NN can effectively learn RNA secondary structure and furthermore, to explore the factors related to model selection, model training and training data that contribute to a model being able to learn secondary structure.

To explore this, we developed an interpretation method, we call second order *in silico* mutagenesis to verify an NN is learning which pairs of nucleotides are directly base paired. Using a basic MLP architecture we show that MLPs can effectively learn secondary structure information from alignment data. These models are able to learn non-nested base pairs from biological data, identifying pseudoknots that CMs are unable to find. We also explored

Recurrent Neural Networks (RNNs) as a potential candidate model that can learn structure from unaligned, variable length data, and identified factors relating to the size of a training alignment that limit a model's ability to accurately learn base pairs. This work builds up a framework to use NNs to learn secondary structure from biological data towards developing an NN-based homology tool.

RESULTS

Alignment data and training regime

Our goal is to explore how well NNs can learn RNA secondary structure when trained on homologous sequence alignments. In the process of alignment, sequences of varying lengths are made into uniform length sequences by the addition of gaps. These gaps represent insertions or deletions in sequences that allow conserved nucleotides and base pairing sites of all the sequences to be aligned in columns down the alignment. In our use of alignment data we treat gaps as a fifth character in addition to the four nucleotides. Each alignment represents a different RNA family of homologous sequences, with each sequence coming from a different species.

We used 20 alignments that matched well to given PDB structures, so that we had a ground truth to verify the learned structural information from an NN after training. These alignments were taken from Rfam, a popular database of RNA family alignments (Kalvari et al., 2018), and columns in the alignment with >50% gaps were removed. Table 1 lists the families used in addition to details about their size.

We trained our NNs to learn structural information from the alignment data using supervised learning. In supervised learning, a model is given training data with labels, and the NN is trained to classify sequences from each other. We trained our models to perform binary classification in which there is a positive set of sequences and a negative set, and the model needs to learn to distinguish sequences of each set. Towards training an NN to learn secondary structure, we wanted the model to be able to distinguish positive sequences that had structure from negative sequences that did not. We needed the positive set and the negative set to differ only by the presence of structure such that the model would be forced to

learn the underlying base pair information in the positive set instead of learning to discriminate the sequences by some other feature such as nucleotide composition, conserved sequence motifs or gap patterns. To do this we used the reduced Rfam alignments as positive data. To generate negative sequences that were identical to the alignments except for structural information, we calculated a position weight matrix (PWM) from the positive alignment and sampled sequences from the PWM with each site in a sequence independent of the other sites, such that there would be no pairwise covariation suggestive of base pair patterns in the negative set.

The data (positive set and negative set) is then divided into a training set (80% of the total sequences), a validation set and a test set (10% each). Each NN is trained for a 1000 epochs on the training set, and the models' parameters are optimized to perform binary classification between the training sequences (see methods for more details). At every epoch of training, the model is given the validation set to cross validate the model's performance and after a 1000 epochs, the parameters with the highest validation performance is saved. At the end of training, the model is tested on the unseen test set, and the performance on this set gives an idea of how well the model has learned to distinguish sequences that have structure from sequences that don't. The idea with this training design is to make the NNs learn general information about the RNA family structure without overfitting the training data.

Neural Network architecture

To get representative insights into how well a NN would be able to learn secondary structure from alignment data, we used an MLP which is a basic NN architecture. MLPs train relatively quickly compared to more complex architectures. With regards to the model architecture, an MLP consists of an input layer, at least one hidden layer and an output layer

(allowing the transformation of the features of the data, thus making the model able to solve XOR problems). The input layer takes in uniform length sequences. Each hidden layer contains a number of units (or neurons) with individual trained parameters for each character in the input. This makes the model ideal for alignment data in which the conserved base pairs are always in the same sites in the sequence. Each neuron effectively performs a non-linear transformation on the input sequence generating new scores after each layer to feed into the next hidden layer. A final output transformation produces a single score used for classification. We used an MLP with 512 neurons in a single hidden layer for our tests. Empirically we found that this architecture worked well on our alignment data, with a large enough hidden layer to learn structural information, but not too large as to overfit the training data. It should be noted that we did not rigorously explore different MLP architectures and hyperparameter choices as our goal was to gain representative insights on how well an MLP could learn structure comparing against different families and alignments.

MLP had high performance in binary classification task

Separate MLPs were trained on each alignment corresponding to a given RNA family from Rfam using the training procedure described above, and the models were tested on the unseen test data. Table 1 shows the accuracy and the area under the roc curve (AUC-roc) of each model on each family. The accuracy refers to the fraction of test sequences that were classified correctly as either having structure or not having structure. The AUC-roc is another measure of model performance, with high scores indicating the model has high classification power. Most MLPs had high performance with >99% accuracy and 1.0 AUC-ROC. It is of note that the two models that had slightly lower performance were two of the smaller alignments (we will discuss this shortly).

Rfam ID	Name	Length	M	M_{eff}	Accuracy	AUC-roc
RF00002	5.8S ribosomal RNA	155	375612	1197.3	1.0	1.0
RF00005	tRNA	73	298470	3533.2	0.999	1.0
RF00010	Bacterial RNase P class A	359	6397	1215.1	0.999	1.0
RF00017	Metazoan signal recognition particle RNA	300	22685	4358.5	0.998	1.0
RF00023	transfer-messenger RNA	366	5983	1558.9	0.999	1.0
RF00050	FMN riboswitch (RFN element)	135	4516	367.1	0.998	1.0
RF00059	TPP riboswitch (THI element)	105	11197	2249.5	0.999	1.0
RF00162	SAM riboswitch (S box leader)	108	4757	585.7	0.995	1.0
RF00167	Purine riboswitch	103	2427	471.1	0.996	1.0
RF00169	Bacterial small signal recognition particle RNA	99	5622	637.2	0.996	1.0
RF00174	Cobalamin riboswitch	188	9056	3190.6	0.997	1.0
RF00234	glmS glucosamine-6-phosphate activated ribozyme	161	842	203.6	1.0	1.0
RF00380	ykoK leader	170	1493	92.2	0.997	1.0
RF00504	Glycine riboswitch	91	6875	895.1	1.0	1.0
RF01734	crcB RNA	65	1267	351.9	0.933	0.992
RF01786	Cyclic di-GMP-II riboswitch	86	237	81.6	0.896	0.95
RF01831	THF riboswitch	100	598	123.1	1.0	1.0
RF01852	Selenocysteine transfer RNA	91	1959	157.2	0.997	1.0
RF02001	Group II catalytic intron D1-D4-3	174	2450	374.0	0.996	1.0

Table 1: the 20 Rfam alignments used for this work with corresponding Length, Depth (M) and Effective Depth (M_{eff}) in addition to the accuracy and AUC-ROC scores of the MLP trained on the alignment. The length corresponds to the number of nucleotides (including gaps) in a sequence after reduction. The depth is the number of total sequences in the alignment. When alignments contain many sequences that are highly related phylogenetically there is less structural information in the alignment than would be suggested by its depth. Thus the effective depth of an alignment corresponds to a weighted depth proportional to how related the sequences are, providing a better indicator of how much structural information the MLP sees during training.

Identifying learned covarying base pairs using Second order *in-silico* mutagenesis

Despite the high predictive power of these models, it is not certain that this directly corresponds to the model learning the structure of the family, because NNs can easily overfit the data, learning noisy features that we are not interested in to make its classification. Therefore, it is important to interrogate the model to identify what information is being learned. We are interested in learning RNA secondary structure. Much of RNA secondary structure is comprised of nucleotides that interact over long ranges via base pairing. These interactions are present in homologous sequence data through nucleotides that show conserved covariation across the alignment in the patterns of base pairs. For this work we focussed on Watson Crick base pairs that follow the pattern A-U and C-G (while G-U wobble pairs and other base pairs are present we did not focus on these in our interrogation method, however high covariation scores were also found for known non Watson Crick base pairs which we will discuss).

A sequence from a family with a given structure will contain nucleotides that base pair; we expect that a trained model that has learned base pairing covariation will be able to distinguish mutations in the wild type nucleotide pairing that break the base pair, from mutations that constitute a different base pair which maintain the structure. To verify if the model has learned covarying base pairs, we developed a method called second order *in silico* mutagenesis (SoM). Figure 4 depicts the process of SoM. Given a sequence containing base pairs, we systematically mutate every pair of nucleotide sites i and j to every other possible pair of nucleotides (i-j: {A-A, A-C … U-G, U-U}) (Figure 4(i)). For a select pair of nucleotides in the sequence this produces 15 mutant sequences in addition to the WT. We test these sequences using a trained model, which results in 16 scores which we can order into a 4x4 score matrix corresponding to pairwise mutations between position i and position j while

keeping the rest of the sequence intact. For nucleotides that are base paired, the scores should be high for complementary mutations which maintain the base pair and low for any other sets of mutations that break it (Figure 4(ii)). Each nucleotide pair produces its own 4x4 score matrix. A full all vs. all SoM between every pair of positions in the sequence results in $\frac{L(L-1)}{2}$ unique score matrices, and including duplicates and WT copies in the saturated mutation process, this produces scores with a final dimensionality of LxLx4x4 where L is the length of the sequence.

To make the scores more interpretable, we reduce the dimensionality of the results using what we call a base pair filter (BPfilter). This is a 4x4 matrix with a value of 1 on the off diagonal and 0 otherwise (Figure 4(iii)). We perform element-wise multiplication between the BPfilter and the score matrix and sum the results to get a single number for each nucleotide pair which we call a SoM score. This upweights scores in the score matrix corresponding to complementary nucleotide pairs where the base pair is maintained. The idea is that, using the BPfilter, the highest SoM scores will be matrices that show high scores for all the complementary pairs (Figure 4(iv)). Score matrices which only give high scores for a single complementary pair (eg. A-U only) will also be upweighted but not as significantly. We noticed that SoM performed on a single sequence was noisy, so we averaged the SoM scores from multiple sequences to average down the noise. To remove additional background noise, we used an average product correlation (APC) correction (Dunn et al., 2008) (See Supplementary Figure 1). Although our current procedure is still in development, we found that these steps were mostly sufficient to remove enough background noise to make conclusions on how well an NN is able to learn structure. We plot the denoised SoM LxL scores according to the positions of the mutated pairs in what we call a watson-crick plot (WCplot) (Figure 4(v)).

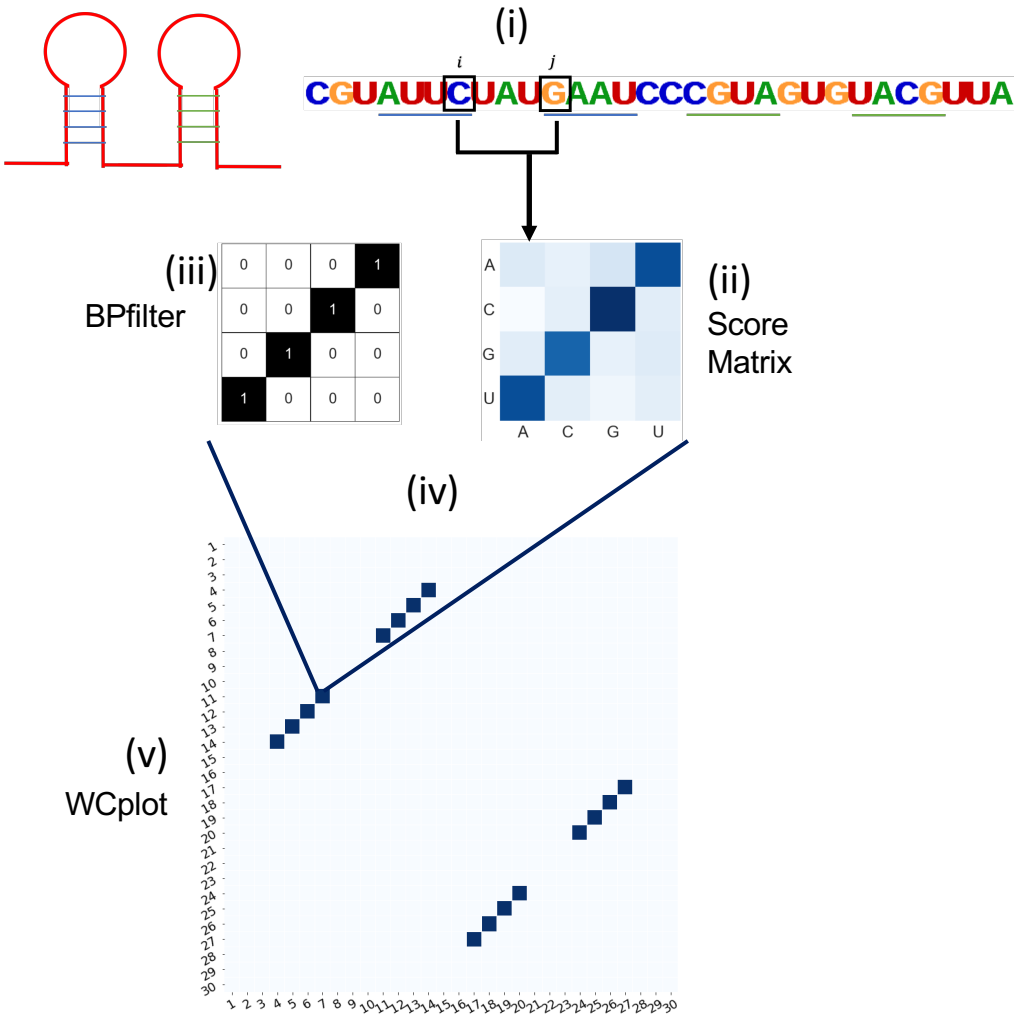


Figure 4: The schematic for second order *in silico* mutagenesis (SoM). (i) For a given sequence of length *L*, we perform saturated pairwise mutagenesis, mutating every possible pair of sites to all pairs of nucleotides. (ii) For a given pair of sites, we produce 15 mutant sequences in addition to the WT and get the scores of each sequence from the NN. The off diagonal of the score matrix corresponds to the complementary nucleotide pairings for watson crick base pairs. If these sites are base paired, we'd expect the NN to output high scores for the sequences with complementary mutations that maintain watson crick base pairs. (iii) To reduce the dimensionality of each score matrix, we employ a BPfilter. (iv) We perform element-wise multiplication and sum the values to retrieve a single SoM score. (v) We denoise the SoM scores by averaging SoM scores across multiple sequences and perform an APC correction. The SoM scores are then plotted according to the positions in the original sequence that were mutated with the colour showing the gradient of SoM scores with darker blue showing a higher SoM score.

A high SoM score reflects that the NN has learned that the particular sequence sites that were mutated covary such that complementary nucleotide pairings are conserved, indicating a base pair. If the model has learned the patterns of covariation that reflect the RNA structure, visualizing the SoM scores will show patterns of high scoring pairs which correspond to stems in the structure. Using this method we can identify how well a NN has learned structural information of a given RNA family.

MLPs can learn covariation from base pairs

We employed SoM on each trained MLP for each Rfam family to identify what structural information the models have learned. For the MLPs trained on families with fully nested structures, to validate the results, we compared the SoM scores to the secondary structure (SS) annotation predicted by Infernal (Nawrocki and Eddy, 2013), an SCFG based model which is the state of the art tool used for RNA homology. While Infernal can only model nested base pairs, it models SS highly efficiently and is a good source to validate our results against.

We also compared our SoM scores to the watson crick (WC) base pair annotations extracted from PDB structures by FR3D (a PDB annotation tool) (Sarver et al., 2008). Nucleotides can interact in many ways; Leontis describes 12 different base pairing types arising from combinations of interactions between the watson crick, sugar and Hoogsteen edges of nucleotides in cis or trans configurations (Leontis and Westhof, 2001). The most common form of base pairing, however, is the cis watson crick - watson crick interaction (which we refer to as WC) which contributes most of the stabilisation energy for RNA folding. Furthermore, while base pairing between most pairs of nucleotides have been found, the most common pairings are the complementary A-U and C-G pairs. Because of this, we

designed SoM to upweight covariation patterns in the score matrices that correspond to showing all four complementary WC base pairs. Notably, the WC annotation does not completely match Infernal's predictions. This arises from WC being only some of the interactions that Infernal is learning, but also this divergence can stem from the fact that the PDB structure was found for RNA from a single species, while Infernal predicts a structure based on homologous information from multiple species (using an MSA). Based on our current design for SoM, we expected that our SoM scores will more closely match the WC annotation, however, it is completely plausible that our MLP is learning all types of nucleotide interactions.

Figure 5 shows two examples of SoM scores from MLPs trained on fully nested structures, compared to Infernal's SS prediction and the WC annotation from a PDB structure for each family. We find that the MLPs are learning the covariation of base paired nucleotides associated with the RNA secondary structure. The model gives higher scores to some base pairs, possibly identifying them as more important for the task of classifying sequences with structure. This variation in score also corresponds to different degrees of covariation in the alignment. For example, tRNA (Figure 5A, RF00005) has a very well conserved structure, with most of the base pairs being equally important with a uniform extent of conservation of covariation. Remarkably, for tRNA, high SoM scores can be found for all of the base pairs predicted by Infernal. Importantly, for Glycine Riboswitch (Figure 5B, RF00504), while there aren't high scores for all predicted base pairs, the MLP has learned at least some base pairs in each stem predicted by Infernal. This suggests that the MLP has learned a general representation of the RNA structure, if not for all base pairs. This is not surprising, as the MLP is trained to perform binary classification between sequences

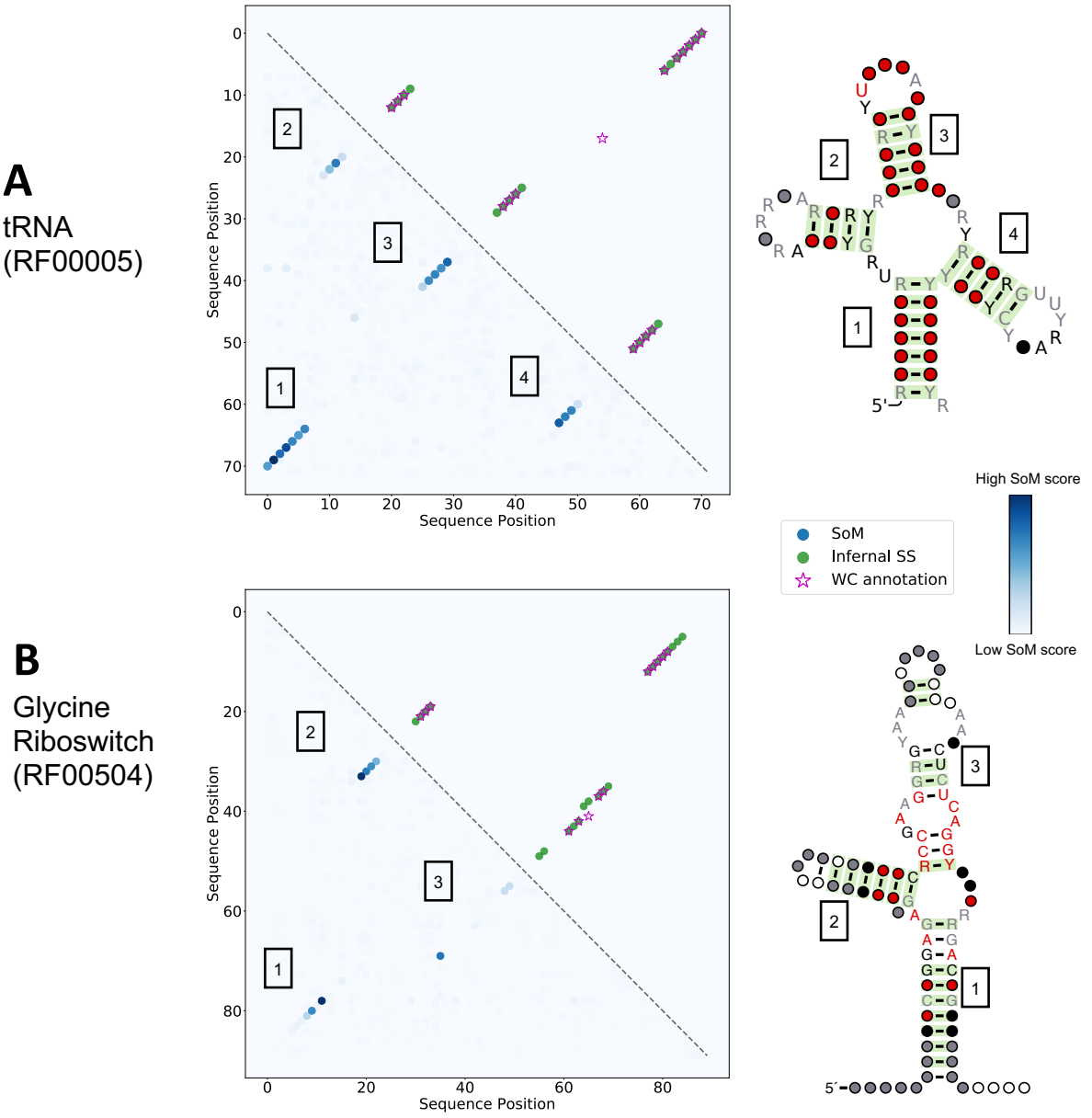


Figure 5: SoM scores (blue) from the MLP trained on (A) tRNA – RF00005 and (B) Glycine Riboswitch – RF00504, compared to the secondary structure (SS) predicted by Infernal (green) and the WC annotation from respective PDB structures, extracted using FR3D (purple stars). Folded structures generated by R2R (Weinberg and Breaker, 2011) using the Infernal SS prediction are next to each plot. Base pairs with statistically significant covariation are shaded in green. Red, black, grey and white dots show nucleotides that are 97%, 90%, 75% and 50% conserved respectively. R and Y correspond to purines (A,G) and pyrimidines (C,U) respectively. The plots show that the MLP is learning covariation corresponding to true base pairs. We expected SoM to mainly elucidate complementary watson crick pairs, but high scores are found where there aren't WC annotations, but are still true covarying pairs as found by Infernal.

that have structure and those that do not, so the model might only be learning as much information as it needs to make the classification.

The MLP is also learning many of the base pairs annotated as WC, although despite our expectation that SoM would pull out scores that would better match the WC annotation, the high SoM scores seem to better match Infernals predictions. This is interesting, because despite the design of the BPfilter, it seems SoM is pulling out more general covariation information that the MLP has learned, versus only complementary base pairs. This shows that our intuition for SoM is imperfect, and we acknowledge that our procedure might not be showing a lot of information that the MLP has learned. However, these results still suggest that the MLP is able to learn nested base pairs and some general representation of the stems making up these RNA secondary structures.

MLPs can capture non-nested base pairs that SCFGs cannot

SCFG based models are widely used for RNA homology, however they are limited in that they cannot model non-nested base pairs which make up structural motifs like pseudoknots. Many of the families we tested MLPs on had pseudoknots in their consensus structure. Looking at the SoM scores from the MLPs trained on Bacterial RNase P class A (RF00010) and transfer-messenger RNA (RF00023), we found that an MLP can learn the covariation of nucleotides associated with non-nested base pairs. (Figure 6) We compared both the SoM scores and the predicted secondary structure annotation from Infernal with the WC annotations from representative PDB structures of the two families (Figure 6C and 6D). We also show the secondary structure of each RNA family taken from Rfam which shows the folded nested structure with non-nested pseudoknot regions annotated (Figure 6A and 6B). While Infernal models the nested base pairs, it cannot model the non-nested base pairs of the

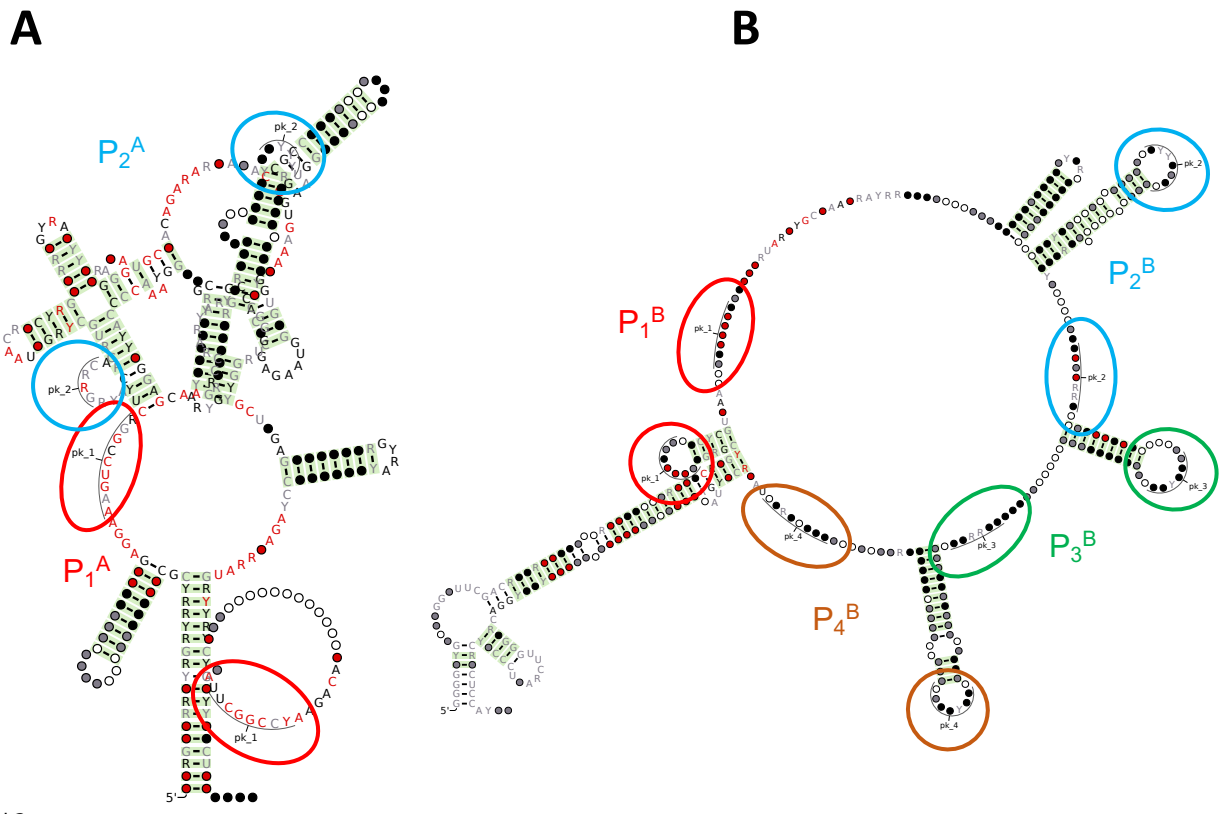


Figure 6: Secondary structures for two pseudoknot containing families: (A) Bacterial RNase P class A – RF00010 and (B) transfer-messenger RNA (RF00023), taken from Rfam. Structures show the folded nested secondary structure, with the regions in the sequence with non-nested pseudoknots annotated by colored rings and labelled P_x^F where *F* refers which to which family, and *x* refers to the pseudoknot number. Folded structure annotations follow same key as Figure 5.

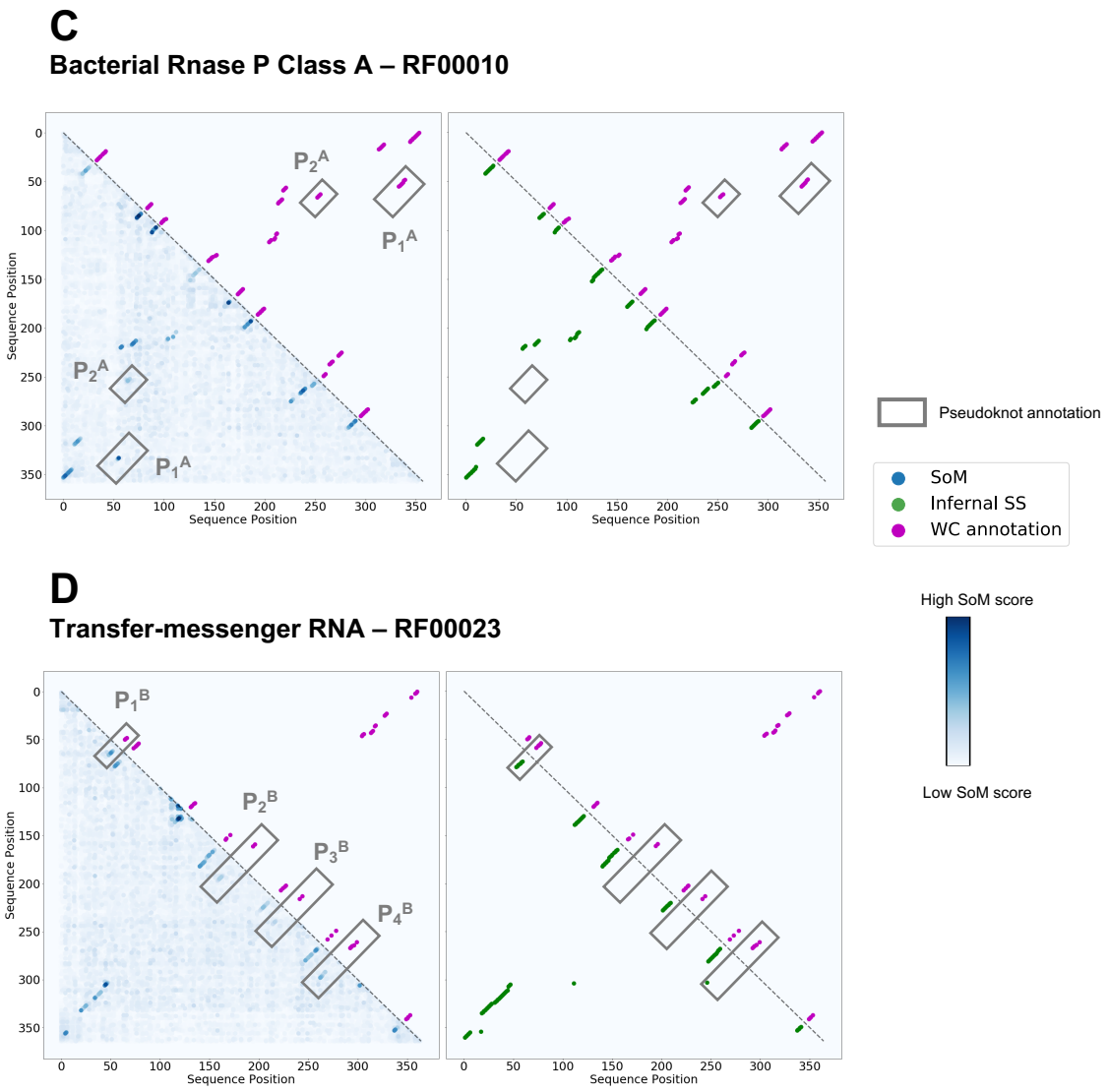


Figure 6 continued: comparing the SoM scores of the trained MLPs with WC annotations and Infernal SS predictions for (C) Bacterial RNase P and (D) transfer-messenger RNA. Plots on the left show the SoM scores (blue) compared against WC annnotations (purple), and plots on the right show Infernal's SS prediction (green) compared to WC annotations. The WC annotations do contain the known pseudoknotted base pairs. Grey boxes are drawn around the pseudoknots with labels corresponding to the colored annotations in A and B. Infernal cannot model pseudoknots, and thus it doesn't have points in the grey boxes. The MLP does learn the pseudo knotted base pairs of both families and shows high SoM scores for the pseudo knotted base pairs that are found in the WC annotations.

pseudoknots. These base pairs are found in the WC annotation and while the number of base pairs does not correspond exactly to the pseudoknot annotation in the Rfam structures, they occur in the expected regions based on the structure. The MLP does learn these non-nested regions showing high SoM scores for the pseudoknotted base pairs. Remarkably, the MLP learns all 4 pseudoknots in transfer-messenger RNA (Figure 6D). Additionally, the SoM results of Bacterial RNase P show very high scores for the few pseudoknotted base pairs that are preferred in the structure, despite being a very large structure with 101 annotated WC base pairs (Figure 6C). Even though we have seen that an MLP will sometimes learn only a few base pairs if that is sufficient for classification (Figure 5B), we see the MLP learn the short pseudoknotted regions with high SoM scores. This suggests an MLP is efficiently able to learn pseudoknotted base pairs and is not hindered by the non-nested structure that prevents Infernal from modelling them.

Validation of SoM scores by comparison to R-scape

We also compared the structural information learned by the MLPs to the statistically significant base pairs predicted by R-scape (Rivas et al., 2017), a tool that finds RNA structure using MSAs and is able to find non-nested base pairs and thus pseudoknots. R-scape identifies evolutionarily conserved covariation between nucleotides that correspond to base pairs. It is able to distinguish covariation resultant from phylogenetic mutations that resemble complementary nucleotides, from compensatory mutations arising from conserved nucleotides. R-scape utilizes a Gtest, which is a modified implementation of measuring mutual information between columns in MSA. R-scape evaluates the probability of a pair of nucleotides being base paired, and also outputs a score for the statistically significant base pairs it finds. R-scape is widely used and highly powerful, and it is a suitable benchmark

model to compare SoM scores from our MLPs to, because it is able to efficiently learn nested and non-nested base pairs.

We compared SoM scores from trained MLPs to the R-scape scores, to validate if the MLP is learning similar sites of base pairing. To get a more quantitative idea for how well each model is doing on a particular RNA family, for each model we also calculated the True Positive Rate (TPR - also known as sensitivity) of finding WC annotated base pairs extracted from a respective PDB structure. The TPR is is the fraction of true positive WC base pairs each model finds in the contacts it finds positive. For both models, positive predictions were the nucleotide pairs corresponding to the top *i* scores, where *i* can be varied. We calculated the TPRs at increasing values of *i* up to C, where C is the total number of significant base pairs R-scape finds. We plot the TPRs in ranked TPR graphs to identify trends in how much structural information the models are finding.

Using WC annotations provides a common ground truth which includes non-nested base pairs to compare the models. However, R-scape is able to learn more than just WC interacting pairs, and we have also shown that SoM can also elucidate the MLP learning non-WC interactions as well. Furthermore, we have seen that the annotations from PDB structures are less general than the information available in an MSA across multiple species. Thus, WC annotations are not a perfect ground truth, and the TPR of the models against WC annotations will thus underestimate how much information the models have learned. Nonetheless, it still provides common information to compare the performance of the MLP against the performance of R-scape.

Figure 7 shows the ranked TPRs (left graphs) and scores (right plots) of four trained MLPs and R-scape tested on the same four families used for training. Neither R-scape or the MLP achieves 100% TPR within the top C contacts predicted, and as mentioned this could be

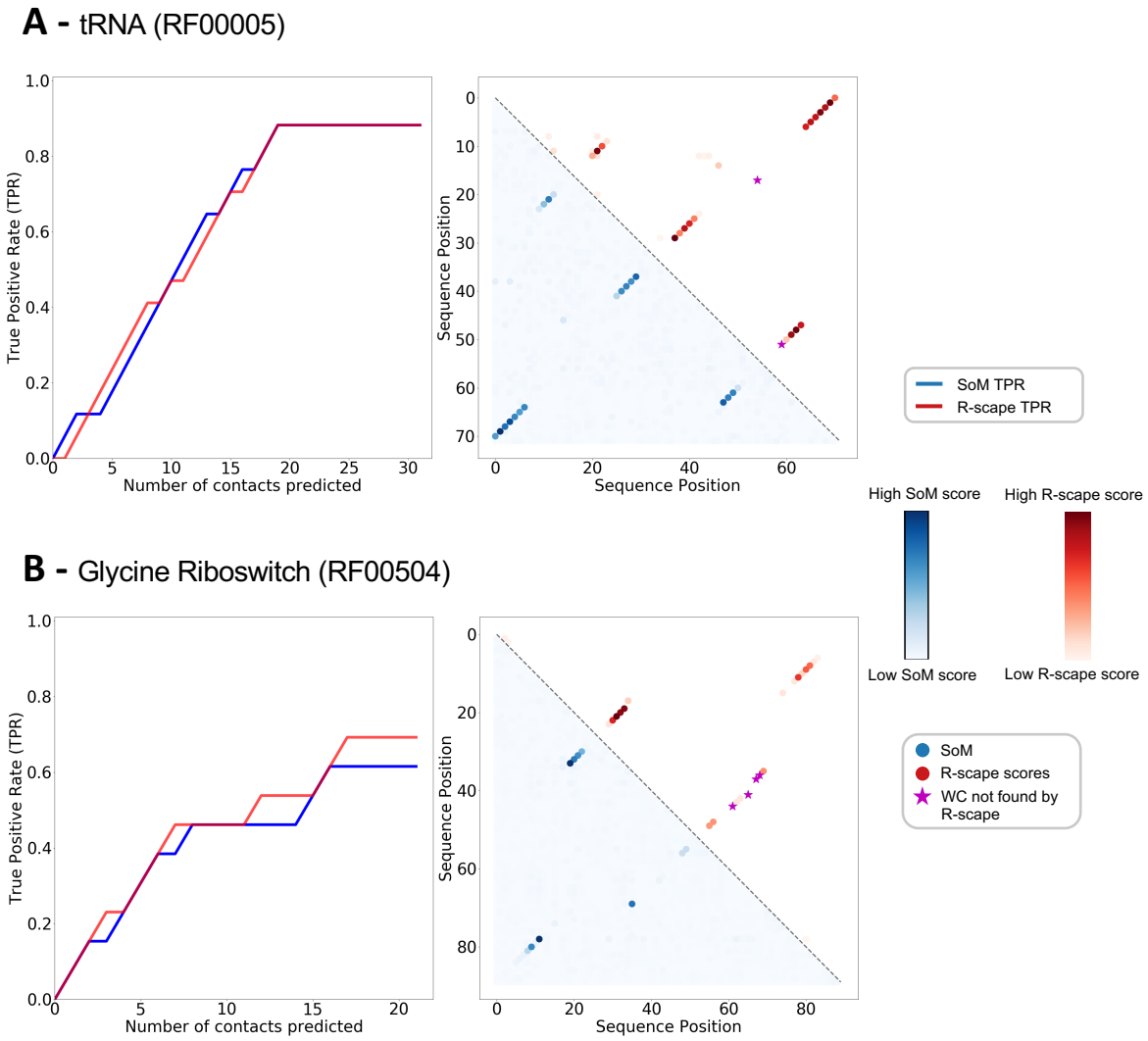


Figure 7: (continued on next page)

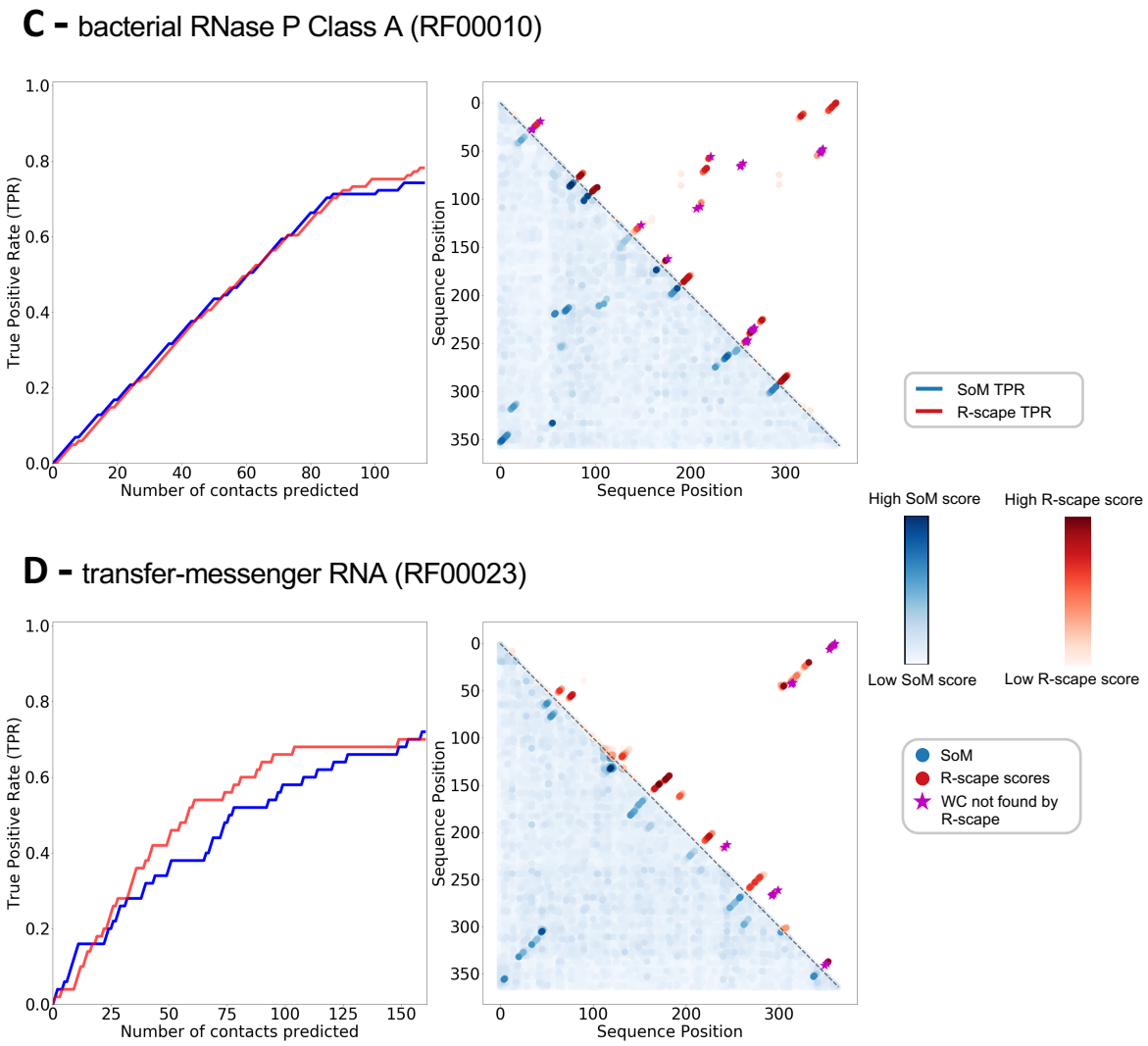


Figure 7: (Left) Graphs showing the ranked TPR the top contacts predicted and (right) SoM scores (blue) compared to R-scape scores (red) with WC annotated base pairs that R-scape does not find also plotted (purple stars). Results are plotted for (A) tRNA – RF00005, (B) Glycine Riboswitch - RF00504, (C) Bacterial RNase P Class A – RF00010 and (D) transfer-messenger RNA (tmRNA) - RF00023. Ranked TPRs are plotted for the top C SoM scores and R-scape scores where C is the total number of base pairs predicted by R-scape. The TPR is a metric of how sensitive the model is to finding WC annotated base pairs. For tRNA and RNase P, the MLP seems to have similar performance to R-scape, learning a similar amount of WC base pairs. For Glycine Riboswitch and tmRNA, R-scape outperforms the MLP slightly.

due to inconsistencies between the PDB structure used and the alignment, or/and a consequence of the models learning non-WC interactions with higher scores. Thus, in the plots on the right we also show the WC annotated pairs that R-scape does not find statistically significant. For the four families shown, the ranked TPRs of the MLP quite closely follows R-scape, with the lines following similar trends, especially for tRNA and RNase P (RF00005 and RF00010 respectively). This suggests the MLP is learning similar sites of base pairing as R-scape. This can also be seen from the plots of the SoM and R-scape scores; high SoM scores are found for similar nucleotide pairs that R-scape finds significant.

To compare the performance of the MLPs against R-scape's performance across all families simultaneously, we plotted the TPR calculated for the top C scores of both models for every family, where C is the total number of significant base pairs R-scape predicts (Figure 8). We find that for most of the families, R-scape outperforms the MLP. This is unsurprising as R-scape's statistical model explicitly looks for significant covariation indicating base pairs between nucleotides; in our current training regime, the MLP has to independently learn base pairing rules and positions of base pairs, sufficiently to perform a binary classification task. Given this, it is remarkable that for 3 families, the MLP and R-scape have equivalent performance, and for two families, the MLP has a slightly higher TPR than R-scape; this is an encouraging result.

There is more variation in the scores for the MLP than the R-scape scores; this could suggest that there are some characteristics of the various alignments tested here that R-scape handles better than the MLP. We looked more closely at the full ranked TPR and SoM scores for the MLP trained on the two families for which R-scape outperforms the MLP most significantly: glmS activated ribozyme (RF00234) and Metazoan signal recognition particle (SRP) RNA (RF00017) (See Supplementary Figure 3). For the MLP trained on Metazoan

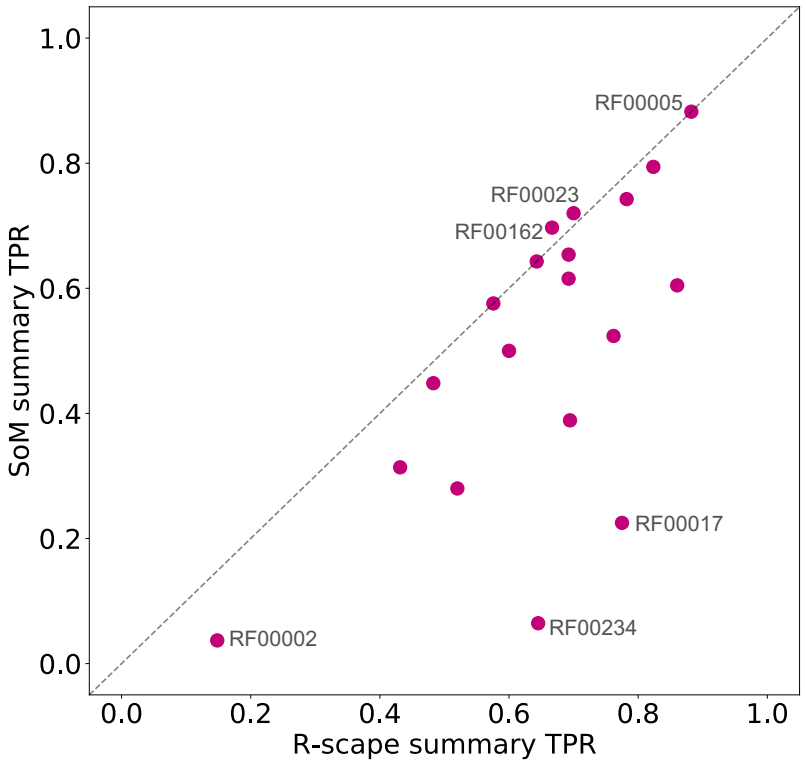


Figure 8: The TPR for the MLPs trained on the Rfam families against the TPR for R-scape shows how the MLP and R-scape's performance compares across 19 of the families used (SoM was performed on only 19 of the 20 families we trained MLPs for; the alignment for Eukaryotic small subunit ribosomal RNA (RF01960) was too large to perform SoM given our computational recourses). Here we plot the TPR for the top C base pairs found by each model, where C is the number of base pairs found by R-scape (i.e. TPR for R-scape using all significant base pairs). Comparing performance across all families shows R-scape outperforms the MLP for all but two of the families (SAM riboswitch (S box leader) – RF0162 and tmRNA - RF00023) for which the MLP has a slightly higher TPR. Shows that R-scape is a much more robust and powerful model than our MLP in its current architecture and training regime. R-scape outperforms the MLP most significantly for glmS activated ribozyme (RF00234) and Metazoan SRP RNA (RF00017). The lowest score for both models was on 5.8S ribosomal RNA (RF00002).

SRP, we found that the SoM scores are incredibly noisy around the regions with annotated WC interactions (which R-scape finds). R-scape does however predict high scores for a long stem of base pairs between sites (~50-125 and ~225-300) which are not in the WC annotation. Interestingly, there are some high SoM scores for that region of base pairs, suggesting that the MLP might have been favouring these base pairs in its learning that are significantly covarying (as shown by R-scape) but are inconsistent with the PDB structure used. For glmS activated ribozyme, the SoM scores don't show any noticeable patterns, suggesting the MLP did not learn relevant structural information (or that SoM is failing to elucidate this information). This low performance may be due to this family having a much shallower alignment for training the MLP, however the MLP does perform better on shorter alignments (see Figure 9), so the reason for this is still unclear.

Notably, both the MLP and R-scape had their lowest performance on the same family: 5.8S ribosomal RNA (RF00002) (See Supplementary Figure 3). Both the MLP and R-scape seem to only be finding a stem of base pairs at the end of the sequence, whereas there are WC annotation indicating multiple stems across the sequence. Furthermore, the nucleotides with high SoM and R-scape scores aren't actually annotated as WC. It is unlikely that this is an artifact of a less general PDB structure. Looking at the size characteristics of this alignment (see Table 1), we found this was the deepest alignment in our dataset, however it had only a middle-range effective depth. This suggests this alignment might mostly contain highly related sequences with a lot of phylogenetic covariation between the sequences, but little base pair covariation. We note that we took this data from Rfam version 11.0 due to the availability of aligned PDB structural information aligned to the Rfam 11.0 MSAs (Weinreb et al., 2016). However, looking at the current active Rfam version (14.1), the current used full alignment is only 4,716 sequences deep. This indicates that the MSA we used for this family

may have been a bad alignment, thus causing both the MLP and R-scape to fail to learn relevant base pairs.

Depth of alignment has an impact on whether an MLP can learn structure

Looking at the SoM results for each trained MLP, we found that the model had varying performance across the families. For some families, the MLP was not able to learn with high precision which base pairs were covarying. This was interesting because most of the models had very high accuracy in the binary classification task. This further shows that the classification performance is not the best indicator of whether a model has learned relevant information.

We sought to try and understand what factors contributed to whether an MLP could learn structure from an MSA. The influencing characteristics could lie in the particular structure of the RNA family, or in the size of the alignment data (length, depth and effective depth) available for training (note: not all of the sequences available can be used for training, a portion need to be used for validation and testing respectively). Deep learning models are data hungry, requiring lots of training data. In fact much of the recent increased use of deep learning models can be owed to an increase in available data of all forms. Thus, we hypothesised that the depth of the alignment was important for whether a model is able to learn an RNA structure. We predicted that the more sequences available for training, the better a model is able to generalize across the data and learn the consensus structure of the family.

We explored this in Figure 9A comparing the depth and effective depth of each family to the summarized SoM performance of each trained MLP. We calculated the summary True Positive Rate (TPR) for finding WC annotated base pairs of each model. The TPR is the

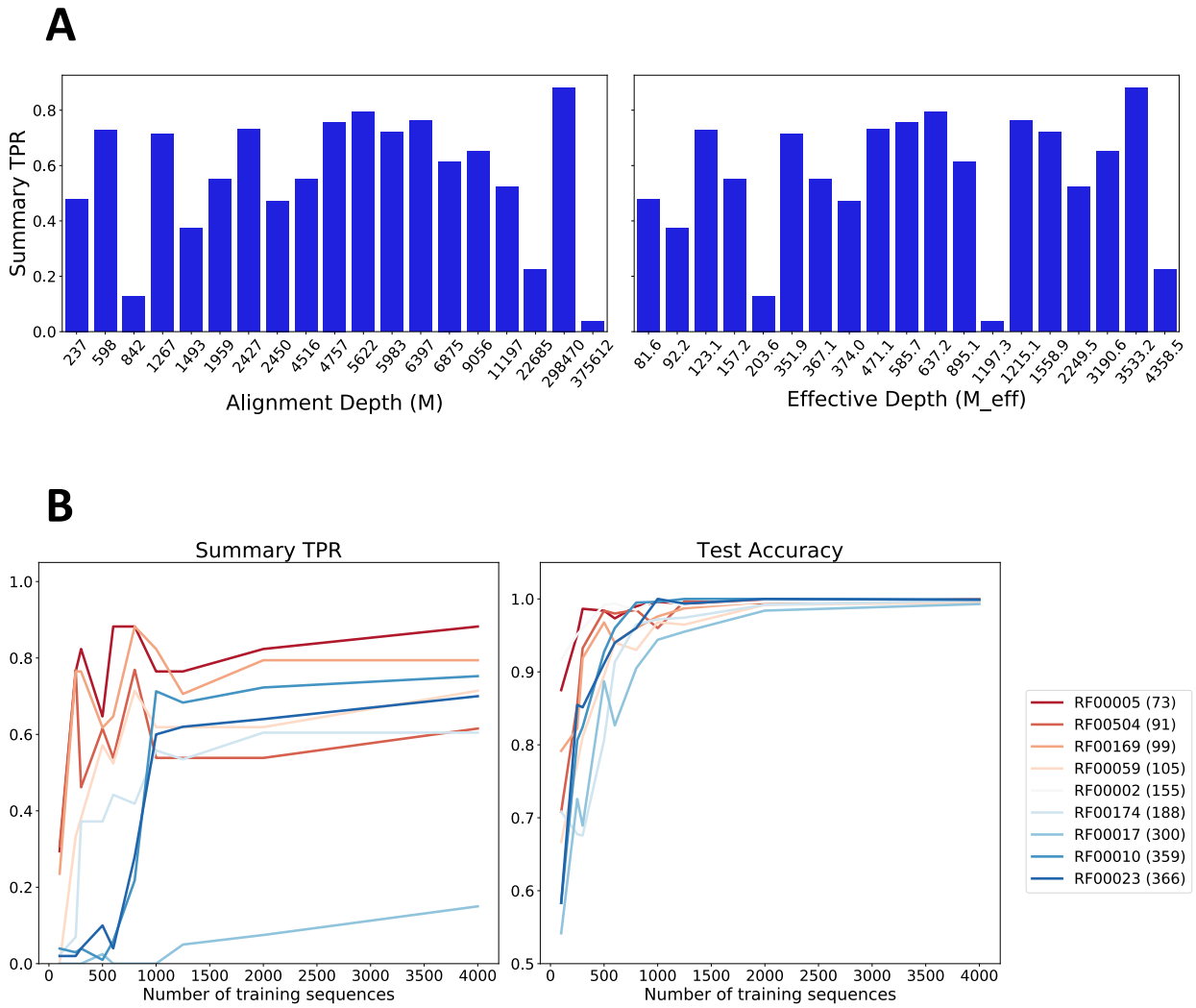


Figure 9: (A) the TPR was calculated for each trained MLP where the top L/2 SoM scores were called positive contacts and L is the length of the respective alignment. These were plotted against the alignment depth (M) and effective depth (M_{eff}) in ascending order. There is no noticeable trend between MLP performance and alignment depth across all the families. (B) For a given family, the depth of the training alignment was systematically reduced, with an individual MLP trained on each training depth, thus controlling for other confounding factors in the alignment. The summary TPR and the model accuracy (in classifying the test set) of each trained MLP is plotted against the training set depth for 9 different families. This shows a general decreasing trend in classification and structure learning performance as the model has fewer training sequences. Each line corresponds to a different RNA family, and the lines are coloured by the length of the MSA of that family. The MSA length is bracketed next to the Rfam ID in the legend.

fraction of all the true contacts that are called positive by the model at the set threshold; it is a measure of how sensitive the model is to WC base pairs. The summary TPR was calculated for the top L/2 contacts learned (where L is the length of the respective alignment) (See Methods for more information). Interestingly, there was no noticeable trend between the summary TPR and depth nor effective depth of the alignment. The peaks are very noisy with sporadic fluctuations. However, we noted that the alignments are of varying lengths, and furthermore each structure is different with different numbers of base pairs, stems and structural motifs, all of which could be confounding factors that affect the performance of the model and confound the trend caused by alignment depth.

To control for the structure of a family and the length of an alignment, we selected alignments that were at least 5000 sequences deep (9 different families), and systematically trained an MLP on smaller training sets of each family. Thus the changes in performance across different training sets, but within a given family reflect only the shallower training data. Figure 9B shows the summary TPR (where top L/2 contacts are positive) and classification accuracy of each trained model at the lower training sizes. We find that for a given structure and alignment length, shallower training sets do affect the structure learning performance of an MLP. Down to ~1000 sequences in the training set, the performance of the models remains quite constant. However, below 1000 sequences the TPR begins to fall.

Notably, the trend is different for alignments of different lengths. Colouring the lines by the length of the alignment shows that longer alignments (blue) decrease in TPR smoothly and the rate of decrease is greater. However, shorter alignments (red) fluctuate strongly in TPR below 1000 sequences. Looking at the WCplots of the different training size trials for families with shorter alignments, we found that, despite this fluctuation in summary TPR, the SoM results do get noisier as the training size decreases (see supplementary figures). We

predict that, because the alignments are shorter and have fewer WW interactions, the noise in the SoM results becomes more uniform across the plot, thus sporadically high scores can appear in positions where there is a true base pair (thus confounding the TPR), even if the WCplot itself doesn't show as much understandable structural information. Thus the fluctuating increase in the TPR values could just be noise. Therefore, even for shorter alignments we observed that with shallower training sets the MLP is less able to learn structure. While the decreasing trend begins at ~1000 sequences, we did find that models trained on full alignments with less than 1000 sequences in the training set (eg. THF Riboswitch - RF01831) can still learn the RNA structure well. Thus, this exploration does not provide a clear threshold for depth of alignments that can be used to train MLPs. It does however provide insight into the effect that alignment depth can have on an MLPs performance, which should be taken into account when using such models.

Recurrent Neural Networks as a model for unaligned homologous sequences

While we have been using uniform-length, aligned sequences as input, towards the goal of a homology tool it would be advantageous if we had a NN architecture which could take in unaligned, variable length sequences. In homology search, a fully independent homology tool does not start with an alignment. We have unaligned sequences of variable length, which then have to be aligned by the homology model. The model needs to be able to learn the conserved sequence and structural information from the variable length sequences and take that into account during alignment. The homology tool can then use that information and the alignment to make predictions of sequence similarity to build homologous relationships.

While our MLP seems to be able to learn structural information well, a major drawback is that the architecture requires fixed length inputs. Each unit in the hidden layer

has individual parameters trained for each individual position in the input sequence. If a test sequence is longer or shorter than the sequences trained on, the MLP cannot add or subtract parameters from its hidden layer to take in the sequence, thus we are required to use an MSA as input. Because of this dependency, the amount of information the MLP can learn from the training alignment is limited by the model used to construct the alignment. If the alignment tool cannot capture the structural information of the sequences during alignment, those conservations may not be present in the MSA columns (thus making it harder for the MLP to learn it). The MSAs we have been using were constructed using Infernal - an SCFG based tool - which is highly efficient at aligning nested base pairs. Our assumption is that if the larger number of nested base pairs are aligned, then the fewer non-nested pairs are more or less aligned as well. This allows our MLP to learn biologically relevant pseudoknots from these MSAs. However, if there is a lot of information in the sequences that Infernal cannot model, then the alignment it constructs is less useful. If we can use an NN architecture that can take in variable length sequences, we remove this dependency and potential limitation from the alignment tool.

Recurrent Neural Networks (RNNs) are a type of architecture that can take in input sequences of variable length. In contrast to an MLP that takes in the entire sequence at once which individual parameters for each site, an RNN reads each nucleotide site one at a time. An RNN unit also has a hidden layer of neurons, and the model's predictive power comes from its ability to update its hidden layers upon reading an individual nucleotide, while also taking into account the nucleotides it has already seen, thus its final output value takes the entire sequence into account. This allows an RNN to model interactions between sequence sites, however a basic RNN model can only learn short-range interactions. RNA structure is the product of long-range base pair interactions. A Long Short Term Memory (LSTM) model

is a modified RNN unit which employs modifications to its update step, including a cell state which the model can update independently of the hidden layer; this affords an LSTM more learning memory, allowing it to model long-range interactions, thus making it a candidate model to learn base pairs from RNA sequences, while also being able to take in variable length sequences.

We sought to explore whether an RNN could learn covarying base pairs from homologous sequences without an alignment. We designed a bi-directional, 2 layer LSTM with 64 units. Two LSTM modules read in the sequence from both directions in the first layer, and then two more modules take in the results of each feed forward in a second layer. The outputs from the last LSTMs produce a single prediction value. We trained the RNNs equivalently to the MLPs. That is we generated negative sequences from site independent sequence PWMs calculated from an individual RNA MSA, and trained the model to perform binary classification between sequences that have structure and those that don't; the only difference being that we used unaligned sequences (without gaps) to train and test the RNNs.

We trained an RNN on unaligned sequences from the tRNA alignment (RF00005) and performed SoM. Figure shows the WCplots from SoM on the RNNs compared to WCplots from MLPs trained on the same families but the original aligned sequences with gaps (Figure 10A). We found that the RNNs did give high SoM scores in some of the general regions where we expected base pairs to be. Notably, the RNN results are markedly more noisy in comparison to the SoM results from an MLP trained on aligned sequences. In addition to a lot of noise around the true positive scores corresponding to true stems, the RNN also shows high false positives in the diagonal of the WCplot corresponding to pairs of nucleotides that are right next to each other in the primary sequence. The RNN does not find as many base pairs in each stem as the MLP does. While the long-range interactions of the first stem in

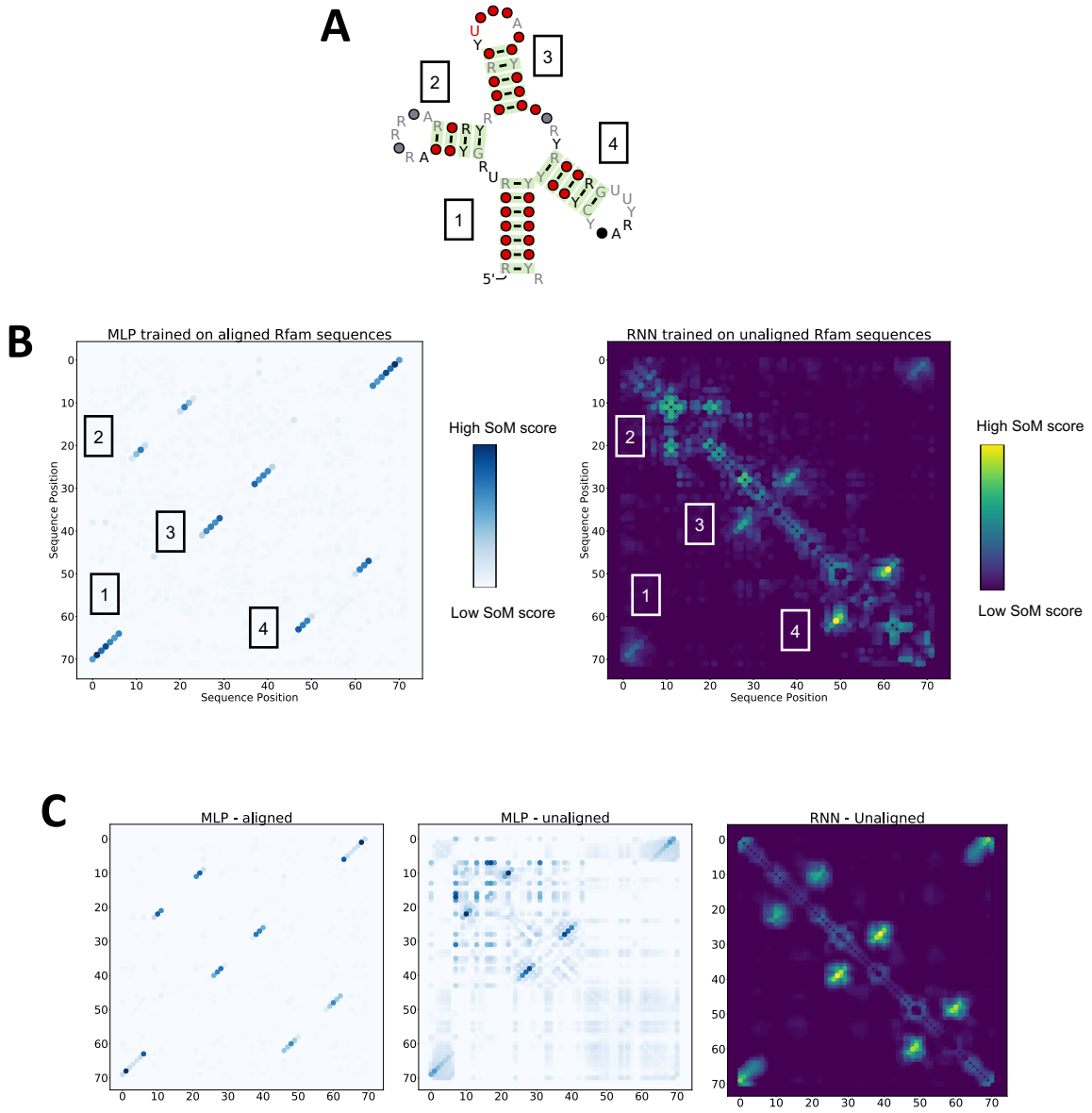


Figure 10: (A) Folded secondary structure of tRNA – RF00005 from Rfam. (B) Results from SoM performed on an MLP trained on the tRNA reduced Rfam alignment (blue) and from an RNN trained on the unaligned Rfam sequences (blue-green). The RNN seems to be learning general regions corresponding to true base pairs in the structure, but the results are visibly much noisier than the MLP trained on the alignment. (B) Results from training the model on a simulated alignment with 100,000 sequences generated by Infernal. The MLP trained on the simulated aligned sequences shows very clear base pair learning, however when training on unaligned sequences the results are much noisier. The RNN trained on unaligned data performs better than the MLP on unaligned data, learning the regions of base pairing from each stem in the structure. The ticks on each WCplot correspond to the sequence position.

RF00005 are found, they are given relatively lower scores. This may suggest that our RNN struggle with longer range interactions, and prioritizes learning closer interactions during training. This could also be a limitation of our training procedure, in which the model is only tasked with making a binary classification and thus the model may only be learning enough information to be able to make a prediction (thus prioritizing closer base pairs). Developing the training regime to force the model to learn all base pairs might improve its performance with learning long-range base pairs.

It is unclear whether these results point to the RNNs not learning structure as accurately, or whether the process of SoM as we use it for MLPs is not best suited to pull learned information from an RNN. In SoM we make mutant sequences with only pairwise mutations to see if an NN is sensitive to these mutations. An MLP trained on aligned sequences is at an advantage in that it has individual trained parameters for each character in the sequence, and is thus more sensitive to small changes in the sequence. Considering an RNN is required to remember individual nucleotides over long ranges to learn base pairs in varying length sequences with a single set of hidden layers, it is remarkable that it can still respond to mutations in individual nucleotides during SoM. The noise in the SoM scores around some of the regions of base pairing may point to the RNN learning general regions of base pairs in the sequence, along with the patterns of WW covariation, lacking higher resolution of exactly which nucleotides are base paired.

An RNN can learn secondary structure with sufficient training data

It is known that RNNs are also highly dependent on the size of training data available. We find that they are much less efficient than MLPs in their ability to learn base pairs from data; that is to say we found our RNN architecture needed far more sequences than the MLP to

train properly. This makes intuitive sense as well, as an MLP is fed aligned sequences, in which the base pairs are always in the same site of the sequence, thus it is easier for the MLPs individual parameters to train. However, an RNN needs to be able to find base pairs in variable sequences, thus it would need as much training data as possible to learn how to identify the covariation between base paired nucleotides. We also note that the effective depth of an alignment is much more indicative of the diversity of sequences available. Thus an RNN would need training data with a large effective depth. We notice this dependency with the results of the RNN trained on the tRNA alignment. Despite being one of the deepest alignments (298,470 sequences), it only has an effective depth of 3522, which could explain why the RNNs SoM results were so noisy. We wanted to see if an RNN could learn base pairs clearly if given more than sufficient sequences.

We sought to train an RNN with input data of a surplus effective depth to see if it was able to learn a more clear structure with sufficient training data. We used Infernal to sample a large number of sequences (100,000) from the Infernal's generative covariance model for RF0005. These sequences would contain the conserved base pairs of the biological alignment. Furthermore, as these sequences were sampled independently they did not have phylogenetic noise and thus the data was very diverse. We trained an RNN to perform binary classification between the positive sequences with structure and negative sequences without structure (sampled from a PWM calculated from the positive data). The RNN was trained on unaligned sequences. We also trained an MLP on the aligned sequences and the unaligned sequences to compare its performance. As mentioned, an MLP requires uniform length sequences as input; to train it on unaligned sequences of variable lengths, we padded the ends of the sequences with gaps such that all the sequences were a uniform length. As all these

gaps are at the end, they do not align the conserved nucleotides in columns so the sequences are still unaligned.

We found that the RNN was able to learn the areas of covarying nucleotides corresponding to base pairs in the tRNA structure (Figure 10B). The SoM results are still noisy, owing to the difficulty of the RNN's task in identifying individual covarying nucleotides, but its performance was significantly better than when trained on the Rfam data with lower effective depth. This suggests that an RNN is able to learn base pairs from homologous sequences given enough data. It should also be noted that the performance of the model and its robustness with smaller data could be improved with a more developed model.

Comparing the RNN results to the MLP trained on the alignment we see that the MLP still outperforms the RNN when given aligned data, however when trained on unaligned data, the MLP results become much more noisy. With the tRNA unaligned sequences the MLP does find some of the first stems with reduced confidence. Looking at the tRNA conserved SS annotation we found that most of the variability in the sampled data was after the third stem, with very few insertions and deletions in the first few stems; i.e. even in the unaligned sequences, the nucleotides corresponding to the first few stems remained in similar sites across the sequences while only the nucleotides of the 4th stem were highly scattered. The MLP treats the input as an alignment and looks for sites with conserved covariations. Because the first stems had less variation across the sequence, the MLP was able to learn the general regions of base pairing, but it could not learn the covariation of the 4th stem because these sites did not show any conserved covariation. Overall, the MLP trained on unaligned sequences performs worse than the RNN trained on the same data. This illustrates the advantages of an RNN in its ability to take in variable length sequences and still be able to learn base pairs when sufficient data is available.

DISCUSSION

Overview

We show that neural networks - particularly Multilayer Perceptrons - are able to learn the coevolutionary base pairs from a multiple sequence alignment of homologous RNA sequences. Our MLP architecture can learn nested and non-nested base pairs, allowing them to learn RNA structures containing pseudoknots. While R-scape outperformed the MLP on most of the families tested, the MLP learned a comparable amount of structural information for some families, which is encouraging considering R-scape is designed explicitly to learn coupling terms for the covariation of nucleotide pairs, allowing them to learn base pairs and other nucleotide interactions robustly; remarkably, an MLP is able to learn the importance of these interactions independently when trained to classify sequences that have and don't have structure. Additionally, we have shown that RNNs are able to learn structure from homologous sequences without an alignment, provided sufficient training data. These results suggest that there is potential for a neural network to be used in homology and possibly supplement the current state of the art RNA homology model (an SCFG) with the ability to model non-nested base pairs.

Model fine tuning is important

It is important to note that the results shown here are from relatively simple neural network architectures. More complex architectures can be designed to more efficiently learn the information desired from the data. This work is focussed on learning insights into what neural networks are capable of learning in order to apply them to biological problems. Instead of focussing on fine tuning hyperparameters and designing a more bespoke architecture, we

selected a simple architecture to gain a representative idea of whether a neural network can learn RNA secondary structure. However, in future applications of these models, fine tuning of hyper parameters such as hidden layer size, learning rate and dropout rate is an incredibly important step towards developing an effective model. It is probable that with a well chosen set of hyper parameters, an model could be developed that can perform much better than the results presented here.

Improve SoM to maximize interpretation

We used SoM as an interpretative tool to understand the interactions in the data that the models have learned. Our results show SoM is a useful tool to extract learned information from neural networks. However, more work can be done to improve the amount of information that SoM can elucidate. The use of a bpfilter to condense the results of pairwise mutations into a single score is limited in its intention to only identify patterns associated with complementary nucleotide pairing. However, our results suggest that MLPs are able to learn interactions between nucleotides that aren't necessarily watson crick pairs; this also shows that the procedure is elucidating information we didn't expect, thus our design intuition must be improved. SoM can be developed to look for other patterns of covariation corresponding to alternate interactions such as G-U wobble pairs. Clustering models or correlation statistics can be used to identify signatures in mutation scores that correspond to general patterns of nucleotide interactions, thus circumventing the limitations of a bpfilter.

Validation sources

We validated the SoM scores against Infernals SS predictions and R-scapes covariation scores, also using WC annotations extracted from a PDB structure to compare between

models. Choosing which model and information source to validate against is a difficult question, especially at this stage of development where we haven't tailored the NN training regime for the structure learning task. Comparing the SoM scores to Infernal and R-scape allows us to benchmark against powerful models that are widely used for the purpose of structure learning. However, while using WC annotations from a PDB structure gave us some ground truth that included non-nested base pairs, we found that the annotations are less general predictions made from an MSA across multiple species. Furthermore, WC interactions are only one type of interaction contributing to covariation in the MSA, while R-scape and the NN may be learning many other types of interactions. Thus comparing to the WC annotations as ground truth, underestimates the performance of the models. A better ground truth comparator is needed to fairly validate our results against established models.

Explore other factors of RNA alignments that contribute to an MLPs learning ability

Our results suggest that the depth and effective depth of an alignment could be a limiting factor for a neural network to be able to learn base pairs from aligned sequences. However, there are definitely other factors that contribute to whether a neural network can learn structure from a given alignment. A salient example is the SoM results from the MLP trained on 5.8S Ribosomal RNA (RF00002); while this family had one of the deepest alignments in our datasets with a comparable effective depth to other successful trials, the trained MLP had one of the worst performance from our tests. The model was unable to learn a meaningful structure, where the only base pairs it learned did not correspond to WC annotated nucleotides in the PDB structure. This suggests that there are other confounding factors in the data that the model focussed on during training that occluded relevant base pairs.

Therefore, much work needs to be done in exploring what other factors in alignment data guide a neural networks learning. Additionally, we did not explore what characteristics of RNA structure the MLP was sensitive to. It may be that particular structural motifs present as covariation signatures in the alignment are favoured by the model. Furthermore, how a model learns base pairs is an important question that we do not explore. What parts of the hidden layer contribute to building these representations of interacting nucleotides? Understanding these factors will be highly beneficial in developing more effective model architectures and training regimes.

While we tested models on a range of datasets, our sample size was still quite small. It is important that we test the models on a wider range of alignment sizes with a greater diversity of RNA structures. This will provide deeper insight into the characteristics of homologous sequence data that facilitate learning RNA secondary structure.

For RNNs, especially, it is important that we explore the effect of distance between interacting base pairs on the learning performance of the model. As discussed, a big problem with RNNs (which is only partially ameliorated by the LSTM module), is that they struggle with long-range interactions in data. Our RNN trained on both the Rfam tRNA sequences, and the Infernal sampled tRNA sequences did show fairly high SoM scores corresponding to base pairs in the first stem in the tRNA structure (Figure 10); these base pairs spanned the entire length of the sequence (71 nucleotides). However, as discussed the structural learning of the RNN is highly noisy and there are many other confounding factors in the tRNA sequences. We can test the limit of long-range interactions on RNNs more explicitly using synthetic sequences that we construct with all other structural factors controlled, while only varying the length of the sequence and the distance between base pairs. Understanding the relationship between base pair distance and an RNNs ability to learn the interactions, will

provide useful insight that could be used to build on a model's architecture towards making it more efficient for a homology purpose.

Transfer learning could make neural networks robust to shallower alignments

One of the biggest limitations of neural networks we found in this work is the need for enough training data. Our RNN, in particular, required a very large number of diverse sequences for it to be able to learn an RNA structure well. Alignments with tens of thousands of sequences with a great diversity of sequences are very rare and thus the inefficiency of the model we used is not reasonable. However, the capacity of RNNs to take in sequences of variable length is highly desirable towards the goal of a homology tool. Thus it would be beneficial to increase the efficiency of training of an RNN to make it usable on training set sizes that are more common with MSAs. Additionally, while the MLP was able to learn clear structures from alignments with a few thousand sequences and even fewer, there are many families in Rfam which have even fewer homologous sequences, and thus increasing the capacity of MLPs could also be useful.

A common method used to train deep learning models in the absence of large amounts of data is transfer learning. Transfer learning involves taking a model pretrained on large datasets related to the objective data and then training parts of the model on the smaller relevant data to make it learn more specific information. It has been shown that deep neural networks learn lower level features of the data in its first hidden layers and then build these features in high layers into informational representations of the data. With smaller amounts of data, it is more difficult for the model to learn these features and build them into a representation of the system. Transfer learning allows the model to simply fine tune pretrained features to make them more specific for the smaller dataset. Transfer learning

could have great potential to improve the efficiency of models when a plethora of data isn't available, which is frequently the case with homologous sequence data.

A difficulty of training neural networks on sequence data towards learning structure, is that the model needs to learn the importance of interactions between nucleotides and the patterns of base pairing from the ground up. This is where more specific models like R-scape have an advantage. Applying transfer learning to this problem could allow a model to learn the patterns of base pairing at lower levels of its hidden layers during pretraining on a large amount of general sequence data (which can be generated) and then when training on the smaller sequence data from the objective alignment, the model can build its base pair features to learn the specific structure of a family. In combination with building more complex architectures, transfer learning could be employed to increase the efficiency of training neural networks, and to make them more viable models given the availability of homologous sequence data.

Towards a neural network based homology model

This aim of this work was to gain insights into what representative neural network architectures are able to learn when trained on homologous sequence data. Specifically we are interested in RNA homology. To make homology predictions, a model needs to take primary sequence conservation and, importantly, the secondary structure into consideration. We show here that neural networks can be trained to learn RNA secondary structure. Furthermore, we have shown that an MLP can learn non-nested base pairs constituting pseudoknots, which the current state of the art RNA homology tool, Infernal cannot model. However, we find that our MLP is not robust in its ability to learn structure, and it is limited by the size of its training set. Infernal is significantly more efficient, being able to model

structure and sequence conservation from small numbers of homologous sequences. We imagine that a NN would be useful as a supplement to infernal, as it is able to learn non-nested base pairs. Foreseeably, infernal could be used as the primary model to build an alignment from sequence data, and a fine tuned neural network could be used to learn the pseudoknotted regions in the sequence to add to the homology model and thus enhance its predictive power.

Much work is still needed to use the insights gained from this work into an implementable neural network for homology prediction. For one, we trained our models to perform binary classification via supervised learning. Homology predictions use probabilistic scores, and thus the training regime will have to be modified for the neural network to predict probabilities rather than binary scores. Furthermore, it would be beneficial to implement unsupervised learning in which instead of learning to classify positive sequences from negative sequences, a model learns a generative model of the data. Thus the model could be trained to learn a holistic representation of the sequence data including primary sequence conservation, secondary structure and other characteristics the model finds relevant to the homology profile of a family.

Conclusion

The development of complex neural network architectures for particular tasks is rapid and ongoing. This work serves to expand the understanding of what neural networks are capable of learning from biological data. Combining strong understanding of what neural networks can learn, with insightful design of model architecture promises to expand the variety of biological problems that neural networks can be applied to.

MATERIALS AND METHODS

Data selection and processing

20 Rfam alignments were used in this work that aligned well to available PDB structures used as ground truth comparisons to validate the model. The families and structural data were taken from the supplementary data of Weinreb et al., 2016. The alignments were taken from Rfam 11.0 and reduced to remove columns with >50% gaps. The alignments were processed into onehot representations where each sequence in the alignment was made into a 5xL matrix where L is the length of the sequence and the 5 dimensions correspond to the 4 nucleotides with a gap treated as a 5th character. To generate negative test sequences, for a given alignment the frequencies of each character at a given site in the sequence were calculated to generate a position weight matrix (PWM) profile.

$$P(a_i^F) = \frac{1}{N}\sum_{s=1}^{N} a_{i,s}$$

Here $P(a_i^F)$ is the probability of nucleotide (or gap) a at site i coming from RNA family F where N is the number of sequences in a particular alignment and s is a sequence in a given alignment. In a PWM each family becomes a distribution with each site having a normalized probability of emitting any nucleotide or a gap. Negative sequences were sampled independently from this profile with each site in the nucleotide being sampled independent of the other sites, so that base pair covariations were not present in the negative set.

Neural Network architecture and training

The MLP architecture we used for our tests had a single hidden layer with 512 neurons utilizing rectified linear activation, batch normalization and dropout applied on each neuron

with a 0.5 probability. The output from the hidden layer is then transformed by an output fully connected layer to produce a logit value. A sigmoid activation is performed on the logit value to produce the final output.

The RNN architecture used was a 2 layer bi-directional LSTM using the standard LSTM unit from the tensorflow library (Abadi et al., 2016). Each LSTM had 64 units in the hidden layer with dropout applied with a 0.5 probability. Two LSTM units read through the sequence in opposite directions, emitting an output score after each nucleotide to produce a new transformed feature layer containing the outputs of both directions. Two more LSTM units read in the feature layer in opposite directions, this time outputting a single logit value which undergoes a sigmoid activation to produce the final output score.

A separate model was trained for each Rfam alignment tested. All models were trained by logistic regression to minimize a binary cross-entropy loss function with mini-batch stochastic gradient descent (mini-batch of 128 sequences) with Adam updates using recommended default parameters with a constant learning rate of 0.0003 (Kingma and Ba, 2014). We also used L2-regularization with a strength equal to 10^{-6}. 80% of each alignment was randomly partitioned for training with 10% for validation and testing each. Each model was trained for 1000 epochs with no patience. Optimal parameters were selected by the epoch which yields the lowest loss on the validation dataset.

SoM scores denoising

In SoM we perform saturated pairwise mutagenesis on a sequence. For a given pair of nucleotides in the sequence, we mutate the pair to every possible pair of nucleotides leaving the rest of the sequence intact producing 15 mutant sequences in addition to the WT. Systematically mutating every pair of nucleotides produces *L*x*L*x4x4 sequences including

duplicates and recapitulated WTs. These sequences are given to the NN and the logit value is retrieved of each mutant to produce the score matrices. We use a BPfilter to reduce the dimensionality of each score matrix to a single number which is our SoM score.

The SoM procedure we describe results in *L*x*L* SoM scores for a single sequence. We denoise the results by performing SoM on *M* sequences from the test set of an alignment and averaging the SoM scores from each pair of nucleotides, where $M = min(500, T)$ and T is the number of positive sequences in the test set. As shown:

$$SoM^*(i,j) = \sum_{s=1}^{M} SoM_s(i,j)$$

where SoM_s is the SoM results from the s^{th} sequence used. We remove further background noise from the scores by performing an average product correlation (APC) correction (Dunn et al., 2008). The APC correction was first employed to remove background scores from information theory based covariation scores, following the intuition that much of the background noise came from phylogenetic covariation between nucleotide pairs that aren't implicate in structure. We noticed that the noise in our SoM^* scores appeared similar to the background phylogenetic noise. In practice, we apply an APC correction by subtracting the normalized row and column averages of a WCplot from each SoM score, as such:

$$SoM^{APC}(i,j) = SoM^*(i,j) - \frac{(\sum_{i' \neq i} SoM^*(i',j))(\sum_{j' \neq j} SoM^*(i,j'))}{\sum_{i} \sum_{j' \neq i} SoM^*(i',j')}$$

These SoM^{APC} scores are plotted in the WCplot.

Validation data: WW annotations, PDB contact distances, EC scores and Infernal SS

The watson crick contact annotations and atomic distances were extracted from a PDB structure file using FR3D (Petrov et al., 2013; Sarver et al., 2008) which were downloaded

from RNA3DHub (http://rna.bgsu.edu/rna3dhub/). These were used in contact maps and ranked TPR graphs to validate the learned information of the model. As further validation we compared the models performance with EC scores taken from the supplementary data of the Marks lab study (see Weinreb et al., 2016 for details on EC model). The secondary structure (SS) annotation by Infernal was taken from the SS annotation in Rfam and matching the positions to the columns in the reduced alignments.

WCplot generation

WCplots showing the SoM scores and R-scape scores were created by plotting the coordinates of each pair of nucleotides in a scatter graph coloured by the SoM/EC score in ascending order of the score. For NN results, the SoM^{APC} scores were plotted with negative scores set to zero as the variation in low scores was just noise. The coordinates of the SS annotation from infernal and the WC annotation from the PDB structure were plotted as a comparison according to the nucleotide positions in the reduced alignments. These were discrete annotations and so no colour gradient was plotted.

Ranked TPR graphs

The True Positive Rate (TPR) of a model was calculated as the fraction of positive WC annotated base pairs from the PDB structure that the model found within a certain threshold, i.e $TPR = \frac{True\ Positives}{True\ Positives + False\ Negatives} = \frac{True\ Positives}{Number\ of\ WC\ annotations}$.

As the gradient of scores was different across each trained model for each RNA family, the nucleotide pairs for the top *i* SoM scores or R-scape scores were called positive predictions where *i* is the threshold that can be set. For Figure 7, TPRs at different thresholds *i* were calculated for both SoM and R-scape, up to *C* where *C* is the total number of base pairs

predicted by R-scape. This was done so that the lines for TPR trends would reflect all the information that R-scape found significant. For Figure 8, the TPR of the MLP and R-scape was calculated selecting the *C* contacts and plotting them against each other to identify which model performed better on which Rfam family (for R-scape this is the fraction of WC annotated base pairs it finds across all of its significant predictions). For Figure 9 the summary TPR is the fraction of true positive interactions that the model found in the top L/2 SoM scores, where L is the length of the sequence. This summary was used as we are not comparing the different training size performances to R-scape, so we used a larger threshold to identify trends between different MLPs. However, using L/2 is still quite arbitrary and using a better summary statistic is necessary.

BIBLIOGRAPHY

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. ArXiv:1603.04467 [Cs].

Alipanahi, B., Delong, A., Weirauch, M.T., and Frey, B.J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotechnology *33*, 831–838.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., and Lipman, D.J. (1990). Basic local alignment search tool. Journal of Molecular Biology *215*, 403–410.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. (ACM), pp. 41–48.

Crick, F. (1958). On Protein Synthesis.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Math. Control Signal Systems *2*, 303–314.

Dunn, S.D., Wahl, L.M., and Gloor, G.B. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. Bioinformatics *24*, 333–340.

Eddy, S.R. (2015). Homology searches for structural RNAs: from proof of principle to practical use. RNA *21*, 605–607.

Eddy, S., and Rivas, E. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. Journal of Molecular Biology *285*, 2053–2068.

Esteller, M. (2011). Non-coding RNAs in human disease. Nature Reviews Genetics *12*, 861.

Gandhi, S., Lee, L.J., Delong, A., Duvenaud, D., and Frey, B. (2018). cDeepbind: A context sensitive deep learning model of RNA-protein binding. BioRxiv 345140.

Graves, A., Bellemare, M.G., Menick, J., Munos, R., and Kavukcuoglu, K. (2017). Automated Curriculum Learning for Neural Networks. ArXiv:1704.03003 [Cs].

Greenside, P.G., Shimko, T., Fordyce, P., and Kundaje, A. (2018). Discovering epistatic feature interactions from neural network models of regulatory DNA sequences. BioRxiv 302711.

Kalvari, I., Argasinska, J., Quinones-Olvera, N., Nawrocki, E.P., Rivas, E., Eddy, S.R., Bateman, A., Finn, R.D., and Petrov, A.I. (2018). Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. Nucleic Acids Res *46*, D335–D342.

Kingma, D.P., and Ba, J. (2014). Adam: A Method for Stochastic Optimization. ArXiv:1412.6980 [Cs].

Koessler, D.R., Knisley, D.J., Knisley, J., and Haynes, T. (2010). A predictive model for secondary RNA structure using graph theory and a neural network. BMC Bioinformatics *11*, S21.

Koo, P.K., Anand, P., Paul, S.B., and Eddy, S.R. (2018). Inferring Sequence-Structure Preferences of RNA-Binding Proteins with Convolutional Residual Networks. BioRxiv 418459.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems 25, F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, eds. (Curran Associates, Inc.), pp. 1097–1105.

Ledda, M., and Aviran, S. (2018). PATTERNA: transcriptome-wide search for functional RNA elements via structural data signatures. Genome Biol. *19*, 28.

Leontis, N.B., and Westhof, E. (2001). Geometric nomenclature and classification of RNA base pairs. RNA *7*, 499–512.

Minsky, M. (1969). Perceptrons: an introduction to computational geometry (Cambridge, Mass.: MIT Press).

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

Pearson, W.R. (2013). An Introduction to Sequence Similarity ("Homology") Searching. Curr Protoc Bioinformatics *0 3*.

Petrov, A.I., Zirbel, C.L., and Leontis, N.B. (2013). Automated classification of RNA 3D motifs and the RNA 3D Motif Atlas. RNA *19*, 1327–1340.

Radecki, P., Ledda, M., and Aviran, S. (2018). Automated Recognition of RNA Structure Motifs by Their SHAPE Data Signatures. Genes (Basel) *9*.

Rivas, E., Clements, J., and Eddy, S.R. (2017). A statistical test for conserved RNA structure shows lack of evidence for structure in lncRNAs. Nature Methods *14*, 45–48.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. Nature *323*, 533.

Sakakibara, Y., Brown, M., Hughey, R., Mian, I.S., Sjölander, K., Underwood, R.C., and Haussler, D. (1994). Stochastic context-free grammars for tRNA modeling. Nucleic Acids Res. *22*, 5112–5120.

Sarver, M., Zirbel, C.L., Stombaugh, J., Mokdad, A., and Leontis, N.B. (2008). FR3D: finding local and composite recurrent structural motifs in RNA 3D structures. J Math Biol *56*, 215–252.

Schirmer, S., Ponty, Y., and Giegerich, R. (2014). Introduction to RNA Secondary Structure Comparison. RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods 247–273.

Schrider, D.R., and Kern, A.D. (2018). Supervised Machine Learning for Population Genetics: A New Paradigm. Trends in Genetics *34*, 301–312.

Simonyan, K., and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv:1409.1556 [Cs].

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. ArXiv:1312.6034 [Cs].

Sonoda, S., and Murata, N. (2017). Neural Network with Unbounded Activation Functions is Universal Approximator. Applied and Computational Harmonic Analysis *43*, 233–268.

Springenberg, J.T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for Simplicity: The All Convolutional Net. ArXiv:1412.6806 [Cs].

Staple, D.W., and Butcher, S.E. (2005). Pseudoknots: RNA Structures with Diverse Functions. PLoS Biol *3*.

Steeg, E.W. (1993). Neural Networks, Adaptive Optimization, and RNA Secondary Structure Prediction (Chapter 3). In Artificial Intelligence and Molecular Biology, (Menlo Park, Calif.: AAAI Press/the MIT Press), p.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going Deeper with Convolutions. ArXiv:1409.4842 [Cs].

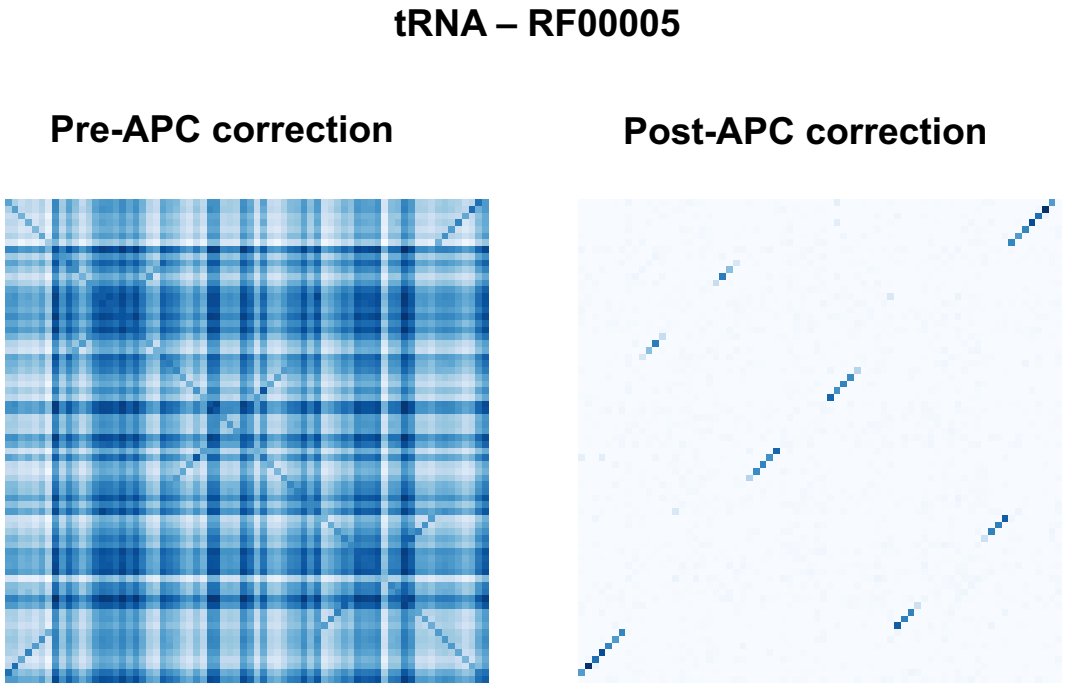
Weinberg, Z., and Breaker, R.R. (2011). R2R--software to speed the depiction of aesthetic consensus RNA secondary structures. BMC Bioinformatics *12*, 3.

Weinreb, C., Riesselman, A., Ingraham, J.B., Gross, T., Sander, C., and Marks, D.S. (2016). 3D RNA and functional interactions from evolutionary couplings. Cell *165*, 963–975.

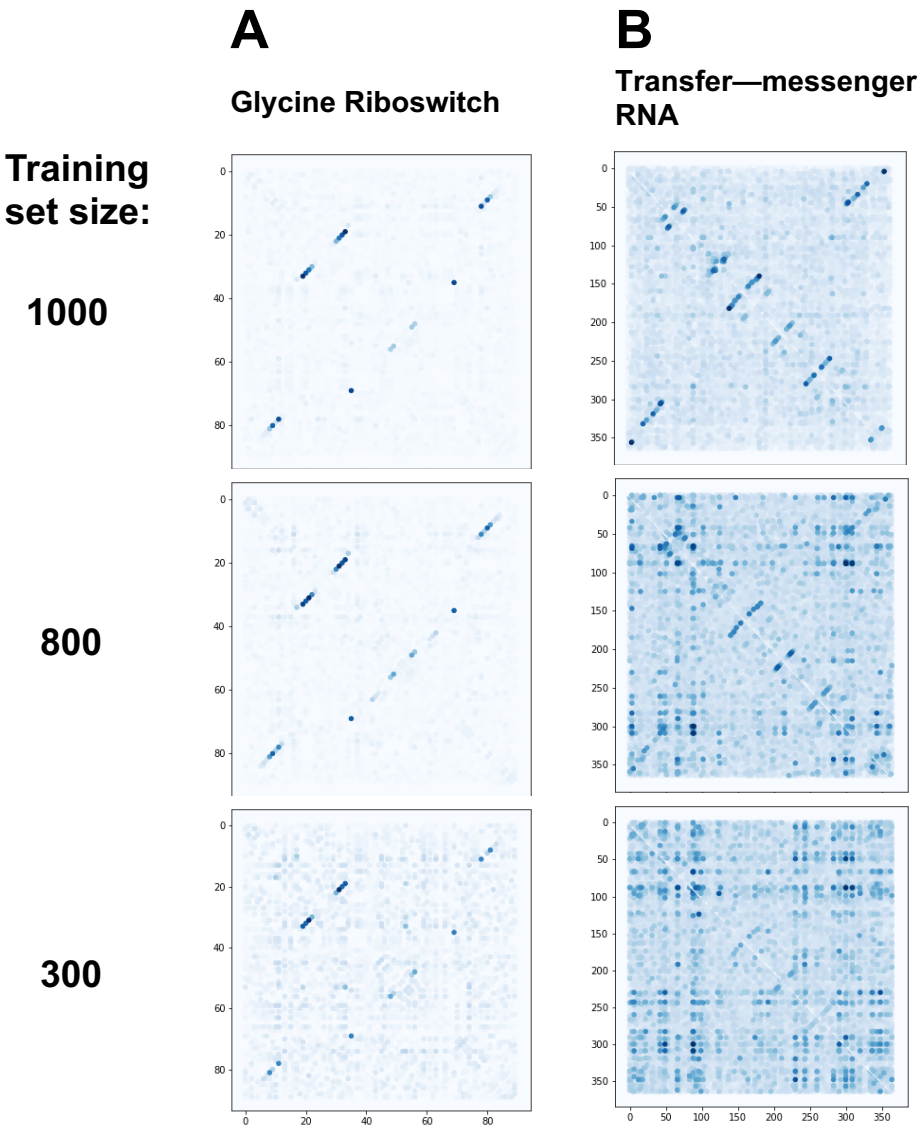
Woese, C.R., and Fox, G.E. (1977). Phylogenetic structure of the prokaryotic domain: the primary kingdoms. Proc. Natl. Acad. Sci. U.S.A. *74*, 5088–5090.

Zhou, J., and Troyanskaya, O.G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. Nat Methods *12*, 931–934.

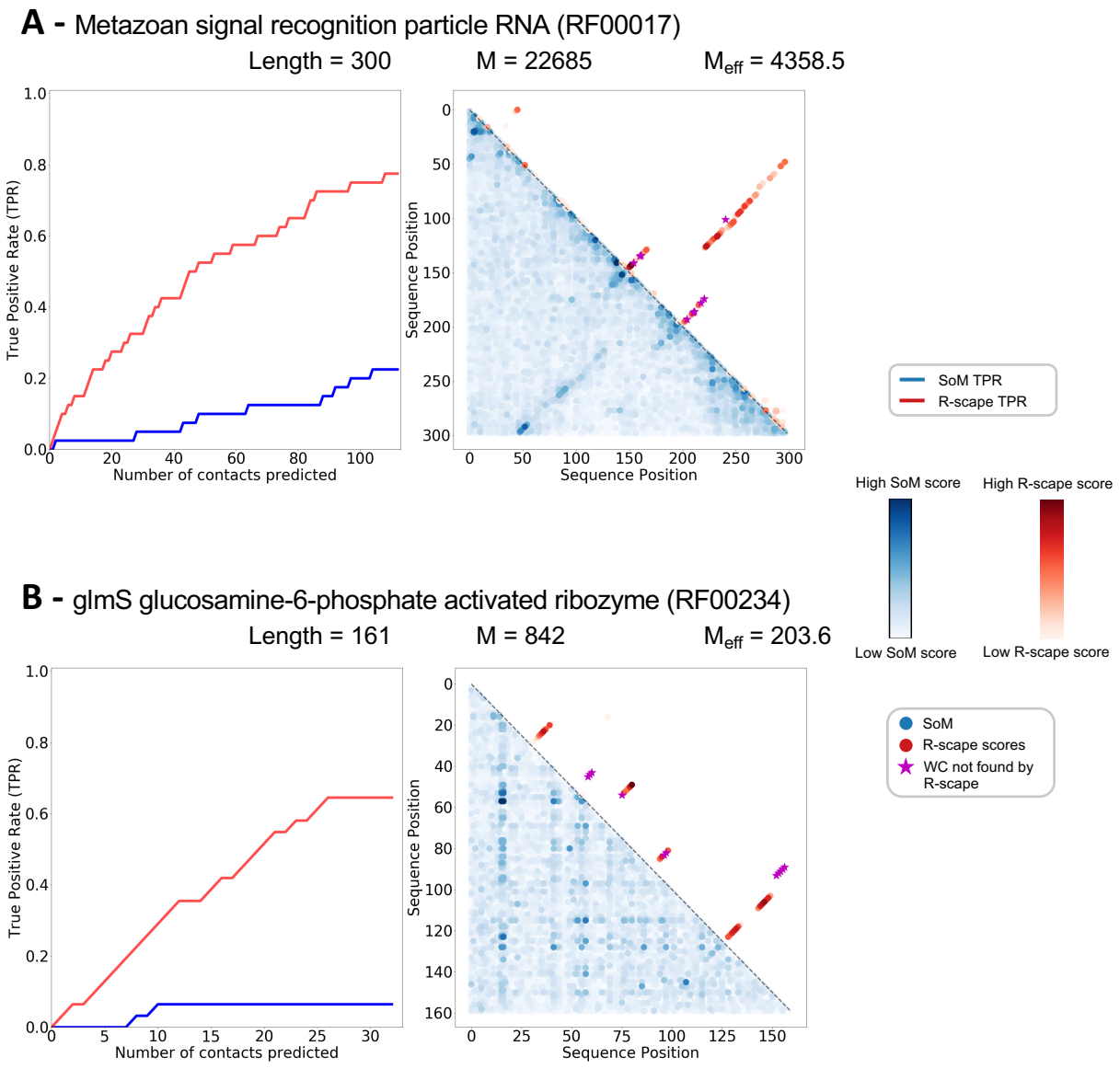
Zuckerkandl, E., and Pauling, L. (1965). Molecules as documents of evolutionary history. Journal of Theoretical Biology *8*, 357–366.



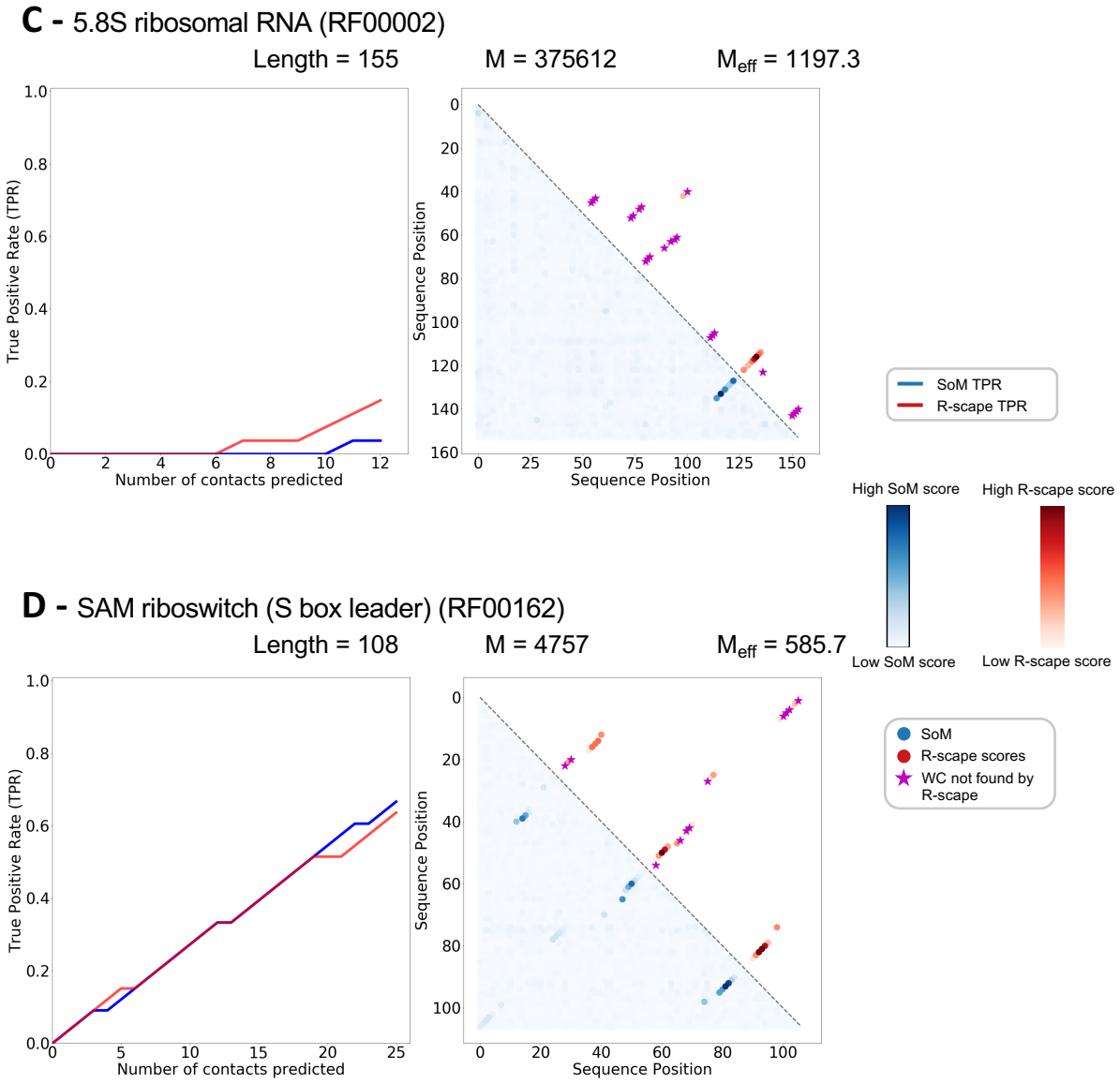
Supplementary Figure 1: Example of noise reduction afforded by an APC correction on the SoM results for an MLP trained on the alignment for tRNA (RF0005). SoM scores are averaged across SoM on 500 sequences from the test set. Before the correction, there is a lot of background noise from reductions caused additive mutations in the rows and columns. An APC correction removes this noise making the scores associated with base pairs in the tRNA structure visible.



Supplementary Figure 2: Example SoM results from MLPs trained (A) Glycine Riboswitch – RF00504 (an alignment of 91 nucleotides in length) and (B) transfer-messenger RNA – RF00023 (an alignment of 366 length) at training set sizes of 1000, 800 and 300 sequences. The SoM results get less clear as the training size decreases for both families, suggesting that depth of training set is an important factor that influences an MLPs ability to learn structure. The results for MLPs trained on Glycine Riboswitch seems to be more robust to lower training sizes, suggesting that even with shallow training sets, an MLP can still learn structure if the alignment is short.



Supplementary Figure 3: continued on next page.



Supplementary Figure 3: ranked TPR graphs and plots of SoM scores and R-scape scores for interesting outliers of the summary TPR comparison (Figure 8), not shown in the main text: RF00017, RF00234, RF00002, RF00162.