Communications Biology

https://doi.org/10.1038/s42003-026-09921-3

Article in Press

What does it take to learn the rules of RNA base pairing? A lot less than you may think

Received: 25 August 2025

Accepted: 12 March 2026

Cite this article as: Pratap, J.S., Krueger, R.K., Rivas, E. What does it take to learn the rules of RNA base pairing? A lot less than you may think. *Commun Biol* (2026). https://doi.org/10.1038/s42003-026-09921-3

Jayanth S. Pratap, Ryan K. Krueger & Elena Rivas

What does it take to learn the rules of RNA base pairing? A lot less than you may think

Jayanth S. Pratap[1], Ryan K. Krueger[2], Elena Rivas[1*]
[1]Department of Molecular and Cellular Biology,
Harvard University, Cambridge, USA
[2]School of Engineering and Applied Sciences
Harvard University, Cambridge, USA

Abstract

Amidst the fast-developing trend of RNA large language models with millions of parameters, we asked what would be minimally required to rediscover the rules of RNA canonical base pairing that define secondary structure, namely the Watson-Crick-Franklin A:U, G:C and the wobble G:U base pairs. Here, we conclude that it does not require much at all. It does not require knowing secondary structures, it does not require aligning the sequences, and it does not require many parameters. We selected a probabilistic model (a stochastic context-free grammar or SCFG) with a total of just 21 parameters, that can describe arbitrary pairwise interactions including but not restricted to those of RNA base pairing. Using standard deep learning techniques, we estimate its parameters by implementing the generative process in an automatic differentiation (autodiff) framework and applying stochastic gradient descent (SGD). We define and minimize a loss function that does not use any structural or alignment information. Trained on as few as fifty RNA sequences, the specific rules of RNA base pairing emerge after only a few iterations of SGD. Crucially, the sole inputs are RNA sequences. When optimizing for sequences corresponding to structured RNAs, SGD also yields the rules of RNA base-pair aggregation into helices. In sharp contrast, when trained on shuffled sequences, the system optimizes by avoiding base pairing altogether. Trained on messenger RNAs, it reveals interactions that are different from those of structural RNAs, and specific to each mRNA. We demonstrate that our approach generalizes across diverse RNA families by testing on 1,094 sequences from 22 structurally distinct RNA families. Our results show that the emergence of canonical RNA base-pairing can be attributed to sequence-level signals that are robust and detectable even without labeled structures or alignments, and with very few parameters. Autodiff algorithms for probabilistic models, such as, but not restricted to SCFGs, have significant potential as they allow these models to be incorporated into end-to-end RNA deep learning methods for discerning transcripts of different functionalities.
Contact: *elenarivas@fas.harvard.edu
Availability: `rivaslab.org`, `https://github.com/EddyRivasLab/R-scape/tree/master/python/d-SCFG`

Introduction

There are many large language models, but what is the minimum sufficient to infer RNA base pairing? RNA large language models (LLMs) often used for RNA secondary and tertiary structure prediction rely on millions of parameters. For instance, RiNALMo [26], one of the largest RNA LLMs so far, uses 650 million parameters. Many other RNA LLMs with similar large number of parameters have been applied to predict RNA secondary structure [1, 22, 14, 8, 39, 41, 42]. In addition, there are many predictive deep learning models (DLMs) specific to infer RNA secondary structure such as MXFOLD2 [33] or UFold [12] that also use hundreds of thousands of parameters and train on many thousands of structures. Some DLMs such as SPOT-RNA2 [35] also include alignments. Deep learning methods as well as downstream applications of LLMs are often celebrated for learning RNA base pairing [34, 6], but what would the performance of much simpler models trained with similar methods look like?

Here, we sought to determine the minimal set of requirements needed to infer RNA base pairing. A key feature of RNA base-pairing are the canonical Watson–Crick–Franklin rules, with A:U and G:C pairs (in both orientations, A:U/U:A and G:C/C:G), and, to a lesser extent, wobble G:U/U:G pairs. Another key feature is that RNA helices,

unlike those in DNA, are typically short: they have a median length of about 6–7 base pairs and are separated and organized by regions of unpaired residues. These properties are universal characteristics of structural RNAs.

In particular, we wanted to test three aspects: (1) whether there is a need to train on known structures; (2) whether in the absence of structures, it is necessary to train on alignments which carry information about covariation between base-paired position [31]; and (3) the size of a minimal model that could learn the rules of RNA base pairing.

We have approached these questions by looking at probabilistic models named stochastic context-free grammars (SCFGs). SCFGs, first established in the context of natural languages, are suitable for, but not restricted to, modeling nested pairwise interactions like those observed in RNA structures. A SCFG is a probabilistic model that defines a set of production rules by which symbols–in our case, nucleotides–are generated and combined into a language. Regular grammars can only capture one-directional dependencies (either left-to-right or right-to-left), but not both simultaneously. SCFGs are one level above regular grammars in the Chomsky hierarchy of formal languages [4]. They are capable of modeling bidirectional dependencies, and in particular long-range left–right interactions. As a result, SCFGs have been used in natural language processing and have also been applied to modeling RNA secondary structure.

The probabilistic parameters of an SCFG are easy to train from structural data by maximum likelihood. Here, we concentrate on several small and simple SCFG designs which, when trained on structural RNAs, have shown to have prediction accuracies near the performance of standard thermodynamic models that rely on thousands of parameters [9]. Thus, we know that by selecting a very particular set of parameter values, these models can recapitulate the basic rules the RNA structure. The key question we address here is: what are the minimal inputs and smaller number of parameters necessary to achieve that? Does the model need explicit base pair annotations during training in order to infer that A pairs with U and G pairs with C?

The training of parameters in modern deep learning is typically performed via stochastic gradient descent (SGD), an optimization method that requires calculating gradients, and modifying the parameters with quantities opposite to their gradients. These gradients are computed using automatic differentiation (autodiff), a technique that takes a function and automatically constructs a procedure to compute its derivatives (the gradient values). In practice, this is often implemented via backpropagation, an autodiff algorithm that is particularly memory efficient for large neural networks. We implemented minimal SCFGs for RNA secondary structure in JAX [2], a state-of-the-art automatic differentiation framework, to enable efficient gradient calculations with respect to SCFG parameters. By applying SGD, we systematically explored the conditions under which the optimal parameters trained in the absence of prior structural information or alignments are able to reproduce the rules of RNA base pairing, that is, the Watson-Crick-Franklin A:U, G:C, and the wobble G:U base pairs (WCF base pairs) that arrange into helices forming the secondary structure.

There are other optimization methods to train from unlabeled (unstructured) data such as the Baum-Welsh expectation-maximization algorithm (EM) [27]. EM has been successfully applied to train SCFGs for specific structural RNA families (covariance models) without using any prior structural or alignment information [10]. Our question is related but different, as we try to infer the elements common to all RNA structures with the minimal SCFG able to model–not just one structure per model (for homology searching), but all possible RNA structures with one model (for secondary structure prediction).

Our approach of training SCFGs by SGD has not been explored before. While EM is a well tested and efficient method for unsupervised training, the advantage of SGD over EM is that SGD only requires implementing one algorithm in an automatic differentiation framework (JAX), enabling the automatic calculation of the gradients and loss. In addition, JAX enables compilation to hardware accelerators such as GPUs and TPUs, and it is the standard method used by deep neural networks (DNN). This gives us the versatility to test many different interpretable SCFGs using standard DNN training methods, and it also facilitates a direct integration of these SCFGs into other more complex DNNs, such as those investigating RNA 3D structure.

A non-technical overview of the key points of this manuscript can be obtained by reading this Introduction, followed by the subsection headings in the Results and only the paragraphs that begin with an underline, the Figures with their captions, and finally the "Main Result" subsection in the Discussion.

Results

A minimal SCFG capable of producing RNA base pairing and more

SCFGs are probabilistic models capable of describing pairwise correlations such as those occurring in natural languages, RNA base pairs forming helices, and others [9, 32]. SCFGs as complex as standard thermodynamic models (that include thousands of parameters) produce comparable results using algorithms of similar complexity [32], but lightweight SCFGs with just a handful of parameters have also been shown to perform comparably to standard thermodynamic models [9].

For this minimal experiment, we selected the G6 grammar [9], which has only 21 free probability parameters. The G6 model, first introduced in the method PFOLD [17], is described in Figure 1. The G6 grammar provides rules to generate sequences. It's built-in assumptions (implicit inductive biases) are:

- Residues can be either uncorrelated (unpaired) or pairwise correlated (paired).
- The ratio of unpaired to paired residues in a sequence is unconstrained.
- Arbitrary number of unpaired residues can occur interspersed in between any base pair.
- A residue cannot pair with more than one other residue.
- Paired residues can be disjoint or nested but they cannot cross.
- All unpaired residues are generated by the same rule.
- There are two distinct rules for paired residues, allowing (but not forcing) the model to distinguish terminal (or disjoint) base pairs from other base pairs that are nested together forming helices.

No additional constraints, priors, or training-specific regularization are imposed on the G6 model. While G6 has been successfully used for RNA structure prediction under a particular parameterization of its 21 parameters, the G6 inductive biases described above do not predispose the model to learn any specific features of RNA-like helices; they only define the language of grammatically allowed configurations. Within this language, the model has the capacity to represent RNA-like helices, but no preference for them is enforced. Many other structural configurations are also permitted, and the parameters have ample expressiveness to favor these alternative, non–RNA-like configurations over RNA-like ones.

More technically, the G6 grammar includes only a total of 21 independent parameters. A large subset of those parameters (19 out of 21) are devoted to describing the probabilities of unpaired residues and base-paired residues. The emission of unpaired residues is modeled by 3 free parameters, defined by the single-residue distribution $p_s(a)$, and the emission of base paired residues is modeled by 15 free parameters, defined by the joint distribution $p_{bp}(a, b)$, where $a, b \in \{A, C, G, U\}$. Both distributions can take arbitrary values, subject only to the usual probabilistic constraints: $p_s(a) \geq 0$, $p_{bp}(a, b) \geq 0$, $\sum_a p_s(a) = 1$, and $\sum_{a,b} p_{bp}(a, b) = 1$. Only a specific parameterization of p_{bp} will give rise to the characteristic RNA Watson–Crick–Franklin (WCF) base pairing pattern, in which only 4 of the 16 possible pairs are frequently observed (A:U, U:A, G:C, C:G), and the two wobble pairs (G:U and U:G) occur with lower probability. Thus, the G6 model is not intrinsically biased toward producing WCF RNA base pairing.

The G6 grammar also contains three non-terminal symbols that act as rule generators. For each non-terminal (S, L, F), there is a discrete probability distribution (T_S, T_L, T_F) that assigns a probability to each production rule of that non-terminal. Figure 1a shows how the complete set of probability parameters for the G6 SCFG can be specified by three Bernoulli probabilities, together with two emission probability distributions as,

$$T_{G6} = \{t_S, t_L, t_F, p_s, p_{bp}\}, \tag{1}$$

where, t_S, t_L, t_F are Bernoulli parameters governing the rule choices at the non-terminals S, L, and F, respectively. Together with the single-nucleotide and base-pair emission probabilities, these grammar rules fully determine how all possible sequences and their corresponding structures are generated and scored. For most choices of parameter values, the resulting interactions will bear little or no resemblance to canonical RNA base pairing.

The base pair probabilities, p_{bp}, could be made more complex by distinguishing between those at the start of a helix (used with L) from those inside a helix (used with F); the latter could also be made conditional on the previous base pair (usually referred to as base pair stacking). Here, we wanted to start with the simplest possible implementation using only one single base pair probability and disregarding stacking (Figure 1a). This simplified G6 parameterization (Eq. 1) has been shown to be enough for our purpose of identifying WCF base pairing when trained using sequences and structures [9].

While the focus is on canonical (WCF and wobble) base pairs, notice that the base pair probabilities could in principle capture non-canonical base pairs, if those were prominent in structural RNAs[15]. And while SCFGs

cannot capture pseudoknots, by integrating to all possible structures, all helices involved in possible pseudoknots would be taken into account in the determination of the optimal parameters.

It is important to note that, although the G6 grammar has the capacity to model WCF base pairs forming helices for particular parameter values, most other parameter settings will either fail to produce WCF base pairs or will not promote their organization into helices. Most parameterizations of G6 lead to predictions with stems that are completely incompatible with known RNA structure (e.g., helices that are too long or too short, do not follow WCF pairing, or exhibit some combination of these issues). Moreover, G6 includes, as a nested model, a stochastic regular grammar with no sense of pairwise interaction between residues.

When G6 is trained on known RNA sequences and structures, its parameters have been shown to reproduce base-pair emission probabilities $p_{bp}(a,b)$ that favor WCF A:U/U:A, G:C/C:G, and G:U/U:G base pairs, as well as a parameter regime in which $t_L \ll t_F$, which is conducive to RNA helix formation [9]. Any substantially different choice of G6 parameters will fail to do so. Our central question here is whether G6 can be trained to capture RNA base-pairing rules directly from sequence data alone, even in the absence of structural information.

Parameter optimization from sequence alone by integrating over all possible structures

SCFGs are usually trained by maximum likelihood which requires using datasets of known structured RNAs with known secondary structures. Alternatively, here we show that SCFGs can be trained using only RNA sequences by optimizing the probability of the sequence, given the model, without needing structure annotations.

For a given sequence (seq) and a possible structure for that sequence (π_{seq}), there is one (and only one, if the grammar is unambiguous) parse and one probability score $P(seq, \pi_{seq} \mid G, T_G)$ associated with that sequence/structure pair. The probability of a given sequence is then calculated by summing the contributions of all possible structures, π_{seq}, that are consistent with that sequence:

$$P(seq \mid G, T_G) = \sum_{\pi_{seq}} P(seq, \pi_{seq} \mid G, T_G), \tag{2}$$

where G is the grammar, T_G is its set of parameters, and the sum is over all structures π_{seq} compatible with seq.

Figure 1b shows three examples of different structures associated with a given sequence, and how those get parsed by the G6 SCFG, which is unambiguous. These examples represent three different types of interactions that G6 has the capacity to accommodate. Structure π_1 will be favored by a parameterization representative of RNA secondary structure where the p_{bp} distribution favors WCF base pairs, and $t_L << t_F$ which facilitates the formation of stems. Structure π_2 will have the highest probability for parameterization in which p_{bp} favors C:U/U:C pairs and $t_F \approx 0$. Finally, Structure π_3 will be favored by a parameterization in which $t_L \approx 0$, in which case the pair distribution p_{bp} is irrelevant to the performance of the model.

Because we want to explore the effect of training a model in the absence of any specific knowledge about specific structures associated to specific RNA sequences, it seems just natural to calculate the marginal probability of a given sequence after integrating the contribution of all possible structures allowed by the model, as those described in Figure 1b and many more. While the direct enumeration of the contribution of all possible structures is highly inefficient, there is a dynamic programming algorithm, named the Inside algorithm, that calculates the probability of a sequence by summing the contributions of all possible structures in time polynomial in the length of the sequence [20, 21]. The Inside algorithm is analogous to the McCaskill algorithm [25] used by the standard thermodynamic models of RNA folding (such as Mfold [43], ViennaRNA [23], or RNAstructure [28]) to calculate the partition function [29, 11].

For instance, for a sequence of length n, $(s_1 \dots s_n)$, the G6 Inside algorithm calculates $n \times n$ matrices $\mathbf{I}_S$, $\mathbf{I}_L$, $\mathbf{I}_F$, such that for each $i \leq j = \{1, \dots, n\}$,

$$\begin{aligned}
\mathbf{I}_S(i,j) &= t_S \sum_{k=i}^{j-1} \mathbf{I}_L(i,k)\, \mathbf{I}_S(k+1,j) + (1-t_S)\, \mathbf{I}_L(i,j), \\
\mathbf{I}_L(i,j) &= t_L\, p_{bp}(s_i s_j)\, \mathbf{I}_F(i+1,j-1) + (1-t_L)\, p_s(s_i)\, \delta(i=j), \\
\mathbf{I}_F(i,j) &= t_F\, p_{bp}(s_i s_j)\, \mathbf{I}_F(i+1,j-1) + (1-t_F) \sum_{k=i}^{j-1} \mathbf{I}_L(i,k)\, \mathbf{I}_S(k+1,j),
\end{aligned} \tag{3}$$

where $\mathbf{I}_S(i,j), \mathbf{I}_L(i,j), \mathbf{I}_F(i,j)$ are Inside scores for substring $s_i \dots s_j$ under S, L, and F non-terminals respectively. The polynomial time complexity of the G6 Inside algorithm is $\mathcal{O}(n^3)$, where n is the sequence length.

For any SCFG, the Inside probability for a sequence is the total probability summed over all possible parses (secondary structures) from the start non-terminal S. We denote,

$$\text{Inside}(seq \mid G, T_G) =: \mathbf{I}_S(i=1, j=n), \tag{4}$$

so that the probability of observing sequence seq under grammar G and parameters T_G is given by

$$P(seq \mid G, T_G) = \sum_{\pi_{seq}} P(seq, \pi_{seq} \mid G, T_G) = \text{Inside}(seq \mid G, T_G). \tag{5}$$

We have produced differentiable implementations of the Inside algorithm for the grammars tested here using the Python library JAX, a popular automatic differentiation library for scientific computing [2]. Our implementation enables backpropagation through the Inside algorithm, and therefore gradient-based parameter updates. These are formally equivalent to differentiable implementations of the McCaskill algorithm for computing the partition function for thermodynamic models [24, 19, 18].

Using these differentiable implementations of the Inside algorithm for a given SCFG, we perform SGD to optimize the probabilistic parameters. Given a training set of RNA sequences with no structure annotation, the optimal parameters are obtained by

$$\begin{aligned} T_G^* &= \arg\max_{T_G} \left\{ \prod_{seq} P(seq \mid G, T_G) \right\} \\ &= \arg\min_{T_G} \left\{ -\sum_{seq} \log P(seq \mid G, T_G) \right\} \\ &= \arg\min_{T_G} \left\{ -\sum_{seq} \log \text{Inside}(seq \mid G, T_G) \right\}, \end{aligned} \tag{6}$$

where the loss to be minimized is the negative log-likelihood, *i.e.*, minus the sum over all training sequences of the log-probability assigned by the current SCFG. The sum is over all sequences in the batch; the Inside algorithm automatically marginalizes over all possible structures for each sequence. At each epoch, SGD is applied to a batch of 10 sequences selected at random from the training set.

In particular for G6, automatic differentiation using RNA sequences alone with no corresponding structures permits numerical parameter optimization to obtain

$$T_{G6}^* = (t_S^*, t_L^*, t_F^*, p_s^*, p_{bp}^*) = \arg\min_{T_{G6}} \left\{ -\sum_{seq} \log \text{Inside}(seq \mid G6, t_s, t_L, t_F, p_s, p_{bp}) \right\}, \tag{7}$$

for the Inside algorithm described in Eq. 3, with the parameters as in Eq. 1.

Robust learning of RNA base pairing rules emerges from just sequences and needs few parameters
The main result of this manuscript is described in Figure 2 which shows that a minimal grammar trained on a small number of sequences suffices to result in the emergence of the WCF pairing rules that are the main signature of structured RNAs.

In detail, Figure 2 describes the training of the G6 grammar on a collection of 50 RNaseP RNA sequences, selected at random from a database of 225 RNaseP sequences introduced before [9] from the Ribonuclease P database [3]. We observe that the WCF base pairing rules begin to emerge in the learned pair-probability distribution after only a few epochs of training. Notably, no structural information is provided during training. At training, parameters are left completely unconstrained, and no prior biases are imposed on their values beyond the requirement that they lie between zero and one and satisfy the usual normalization constraints described above in the description of the model.

The success of G6 in converging to the correct parameter values is likely due to the fact that, when trained by SGD, it identifies the complementarity signals inherently present in the RNA sequences. It is nevertheless remarkable that the model is able to discover these complementarities entirely without supervision, since, in principle, many different made-up complementarity schemes could be found within any given sequence. The ability of the model to recover the biologically correct pairing rules in the absence of structural supervision is a non-trivial result and, to our knowledge, has not been reported before.

Supplemental Figure S1 shows that even when reducing the training set to 25 RNaseP sequences, the learned model still produces reasonably accurate base pairing probabilities, although the loss becomes more unstable. In summary, we find that robust learning of base-pairing rules is feasible even with as few as 25–50 sequences, emphasizing both the strength of the pairwise statistical signal in structured RNAs, and the fact that the signal is contained in their primary sequences.

RNA secondary structure predictions are similar with or without structural information

To evaluate how well G6 performs in predicting RNA secondary structure when trained only on sequences, we used an extensive dataset with 1.094 sequences belonging to a total of 22 different structural families [32]. This test set TORNADO_TrB is structurally dissimilar to any of the RNA sequences used in training which guarantees that performance is not affected by overfitting.

To compare structural predictions, we used TORNADO [32]. TORNADO is a general purpose SCFG framework that can accommodate arbitrary SCFGs using customizable parameter sets, and implements most standard structural prediction algorithms. TORNADO allows us to perform a direct comparison between two G6 parameterizations: one trained only on RNaseP RNA sequences, and another trained via maximum likelihood using both RNaseP RNA sequences and structures. In both cases, we report maximum expected accuracy (MEA) predictions.

We measure folding accuracy using the F1 score, which is the harmonic mean of sensitivity (the fraction of true base pairs predicted correctly) and positive predictive value (the fraction of predicted base pairs that are true). F1 is high only if both sensitivity and positive predictive value are high. We report a single global F1 score aggregated across all test sequences. Overall, the F1 scores of known structural RNAs vary from ~ 0.45 for tmRNAs to as high at 0.80 for tRNAs using standard thermodynamic models [9, 32, 38].

Figure 2b shows that the G6 grammar trained only on RNaseP sequences (G6-SGD) has an accuracy comparable to that of G6 trained by maximum likelihood on both sequences and structures from the RNaseP training set F1 (G6-ML; = 0.57). This similarity in performance is expected, as Figure 2c shows that the parameters learned by SGD without structural supervision (blue) closely match those obtained by ML when structures are provided (orange).

Naturally, performance can get better with larger models. For instance, RNAfold from the standard thermodynamic method ViennaRNA [23] achieves a F1 = 0.65. In Table 1, we also show a comparison to two deep learning methods, MXFOLD2 [33] and RiNALMo. MXFOLD2 which also uses thermodynamic information slightly outperform RNAfold (F1 = 0.67 versus 0.64). On the other hand, RiNALMo achieves an F1 score of 0.55 on this TORNADO_TrB test set similar to the G6 result of F1 = 0.57, even though RiNALMo has 650M of parameters and has seen the test sequences in pre-training. In contrast, our result (F1 = 0.57) is based on a model with only 21 parameters, trained from scratch on just 50 RNaseP RNA sequences, without access to any structural annotations. An in-depth comparison of the performance of thermodynamic, deep learning, and LLM-based models for RNA secondary structure prediction—consistent with the results presented in Table 1—is provided elsewhere [37].

Base pairing inference does not require aligned sequences

Alignments of RNA sequences are a well-established source of structural information, as the co-evolution found in aligned base paired positions can be exploited to learn quite accurately the base pairs present in a conserved structural RNA [31]. The RNaseP sequences used in our experiment in Figure 2 are not aligned. However, since the training set is made of all homologous RNaseP sequences (with lengths ranging from 189 to 475 nts), there is the possibility that sequences appear quasi-aligned, and that the optimization algorithm learns from some residual covariation. To rule this out, we trained on a set of diverse, non-alignable sequences with distinct RNA structures. As Figure 3 shows, the same base-pairing rules emerge, demonstrating that our results do not depend on any incidental alignment or residual covariation among the RNase P sequences.

In detail, in order to challenge the hypothesis of the signal being the result of quasi-aligned homologous RNAs, we expanded the training set to include sequences belonging to seven different structural RNA families. In addition to RNaseP RNAs, the new training set includes: SRP RNAs, tmRNAs, telomerase RNAs, Group I and Group II introns, and 5S rRNA. In order to keep the two training sets to be otherwise comparable in the number of sequences from a given family, we selected a random subset of 400 RNAs from the total 7 Family collection of RNA sequences.

In Figure 3, we observe similar results when using this combined training set. The WCF rules appear as strong and as early as in the RNaseP-alone training case.

Affine base pair aggregation into helices is key and can be learned just from sequences

Our results show that for structural RNAs, sequence signals for pairwise complementarity are strong and consistent enough across families to enable parameter inference using only sequence-level data, without requiring annotated structures or alignments. Moreover, we wanted to investigate which features in the G6 model are responsible for that result, for which we compared to SCFGs even more reduced than G6.

We selected G6 as one of the smallest SCFGs able to represent RNA secondary structure, and with quite competitive performance compared to other more complex models [9]. There is another SCFG tested for RNA secondary structure, named G5, which has one fewer parameter than G6. The G5 SCFG requires only one non-terminal S with rules,

$$\begin{array}{lccccc} G5: & S \to & a\,S & | & a\,S\,b\,S & | & \epsilon \\ & & t_u\,p_s(a) & & t_p\,p_{bp}(a,b) & & 1-t_u-t_p. \end{array} \tag{8}$$

The G5 grammar is specified by two independent probability parameters: $0 \leq t_u \leq 1$, the probability of emitting an unpaired residue, and $0 \leq t_p \leq 1$, the probability of emitting a base pair. Although G5 can represent sequences with any arbitrary degree of structuredness—from completely unpaired ($t_p = 0$) to fully base-paired ($t_u = 0$), it differs from G6 in that it cannot distinguish whether base pairs are organized into helices or not. In G5 all possible arrangements of base pairs are assigned similar probabilities, because only the total number of base pairs is taken into account in the score.

It has been shown that, when trained by maximum likelihood, the G6 grammar performs significantly better than G5 [9]. As for why? The inductive biases of G5 and G6 are the same except for one crucial difference: G6 has two distinct rules for base pairs, whereas G5 treats all base pairs in exactly the same way. Thus, G5 is forced to assign the same probability to every base pair regardless of whether it starts a helix, continues a helix or appears in isolation. Thus, G5 cannot be trained to learn a realistic description of RNA structure, which requires that the probability of starting a helix is a lot smaller than the probability of continuing a helix.

Technically, the key difference that makes G6 work but not G5 is that G6 has two independent probability parameters to control base pairing: t_L controls the start of a helix, and t_F controls the extension of an existing helix adding base pairs (Figure 1), whereas G5 has only one parameter, t_p (Eq. 8). By selecting parameters that satisfy $t_L < t_F$, G6 can assign a higher probability to a helix with several base pairs stacked together than to the same number of lone single base pairs.

Consistently, when we train G5 by SGD, because G5 does not have the capability to optimize specifically for helices, it is also unable to find the WCF base pair rules. As we show in Figures 4a-4b, by the time the loss stabilizes, the optimal base pair probabilities do not reproduce WCF base pairing.

Trained by ML from structures, the G5 base pair probabilities are forced to represent WCF base pairing. However, WCF base pairing does not improve performance for G5. This suggests that it is the combination of WCF base pairing together with the ability to group base pairs into helices that minimally guarantees a model able to learn RNA secondary structure. G6 appears to be the minimal SCFG able to achieve both goals.

It is remarkable that both basic properties of RNA secondary structure can be learned together directly from sequences. Our SGD training procedure that does not see structures, when trained on G6 in addition to learning the WCF base pairs rules, converges to values of the two helix parameters that clearly favor the helical stacking of base pairs that is consistently observed in all RNA structures. In fact, the parameter values of G6 as trained by SGD on the dataset of 50 RNaseP sequences are quite similar to those obtained by ML training on sequences and structures of the same dataset (SGD values: $t_L = 0.161, t_F = 0.710$; ML values: $t_L = 0.119, t_F = 0.788$).

Table 1 summarizes the key RNA secondary structure models evaluated in this work, listing model size, training requirements, and performance (F1 score) on the TORNADO_TrB test set. We observe that G6 trained by SGD just on sequences achieves similar performance to when the grammar is trained using also structures.

What about non-structured or protein-coding sequences? G6 leans sequence-to-function relationships beyond those of RNA structure

To better understand the boundaries of this approach, we trained the G6 model on control data such as random (shuffled) sequences and protein-coding mRNA sequences. Failure to produce WCF base pairing on control data is as important a result as success on structured RNA, since a robust model should show the signal disappearing entirely for random sequences, while potentially revealing distinct function-specific signatures for protein-coding sequences.

These optimization experiments are key to demonstrating that the G6 grammar has no inherent predisposition to produce RNA WCF base pairs. Depending on the training data, G6 can converge to favor many different interaction patterns–including no interactions at all, as well as patterns unrelated to RNA secondary structure–simply by being trained *unsupervised* on different classes of functional RNA sequences.

Non-functional shuffled sequences optimize to avoid base pairing

We wanted to test the effect of training on unstructured random sequences. To maintain the base composition intact, relative to the original experiment, we created random sequences by shuffling the RNaseP sequences in the training set. We expect that the model learns from the sequences, thus it should learn that these are not structural but random sequence, but how?

Figure 4c shows that for random sequences, the optimization process converges to a regime in which the losses oscillate across a stable range. Interestingly, we observe that the grammar evolves to avoid producing any base pair. The first rule of non-terminal L is responsible for starting a new helix. We observe that the probability of that rule, t_L, becomes very small ($t_L = 0.00418$ after 99 epochs). Consequently, the number of predicted base pairs, given in Figure 4d, shows that after some instability they tend to be almost zero.

In Figure 4d, we also observe that the base pair probabilities do not optimize to WCF values. Moreover, as random sequences make the model optimize to the trivial solution of making almost no base pairs, the values of the base pair probabilities become almost irrelevant as they are rarely used.

mRNAs have sequence-specific pairwise correlations

Finally, we sought to understand what kind of signal the model would capture for a selected set of functional sequences with a signal different from that of structural RNAs, such as messenger RNAs (mRNAs).

We randomly selected two intron-less protein coding genes from *S. cerevisiae*, HSP12 and SBH2, and curated datasets of their homologs (see "Methods"). In Figure 5c, we show how when G6 is trained on the HSP12 protein-coding sequences, the model is able to identify correlations between the protein residues [5] that can be observed at the mRNA level [13]. In Figure 5d, we also observe that the optimized parameters are different for SBH2 coding sequences, which stresses the fact that mRNA sequence correlations are not stereotyped as RNA structure is.

There is no equivalent for mRNAs corresponding to the RNA helices of canonical base pairs. When trained on mRNAs, the model captures non-structural statistical correlations likely reflecting codon usage and amino acid correlations, rather than RNA structural features. This further demonstrates that the emergence of canonical base pairing rules is specific to the organized informational content of structural RNAs, not a universal artifact of SGD optimization.

Discussion

The main result

We have demonstrated that a model with only 21 parameters (G6), trained solely on the sequences–critically, not the structures–of a few scores of structured RNAs, is sufficient to learn the rules of WCF RNA base pairing, as well as the principles by which base pairs assemble into helices. Moreover, the parameters learned from sequences alone closely match those obtained when explicit structural information is included during training.

In order to show this result, we have introduced a method for training SCFG models via SGD. This method relies on automatic differentiation frameworks for computing the gradients of an objective function with respect to the parameters of an SCFG. Our implementation of SCFGs can automatically be compiled to hardware accelerators such as GPUs.

Our results demonstrate that neither structures nor alignments nor models with millions of parameters are necessary to extract the rules of RNA base pairs (Figure 5a). Thus, it should not come as a surprise that large language RNA models not trained on structures or alignments are able to identify base pairs in structural RNAs [6]. On the one hand, large language models have the potential to tackle a wide range of learning tasks for biological sequences. On the other hand, simple and interpretable models like G6 make it easy to test whether important properties are recovered from the training set, thanks to their interpretability. These simple models when implemented within a deep learning framework, as we have done here with the SCFGs G6, they can be readily incorporated as helpers into end-to-end deep learning methods designed to address other more complex tasks.

RNA structure is more than WCF helices. Are sequences enough to learn more complex motifs?
Our results from training G6 on 50 RNase P RNA sequences demonstrate that even a very shallow model trained on limited data is sufficient to capture the core RNA base-pairing rules that are general across structural RNAs. However, when G6 is trained on seven different RNA families, its performance drops slightly below ML levels, suggesting that its limited capacity is insufficient to model other structural features beyond WCF base pairs, such as multiloops, which vary substantially from family to family.

One such level of RNA structural diversity that G6 cannot capture is observed in the loops. G6 assigns the same probability to all single residues regardless of where the appear, outside the structure itself or in any of the structural loops that connect the stems into a 3D structure. However, it is well established that hairpin loops, internal loops and multiloops all have distinct sizes, frequency of occurrence and sequence-dependent stability properties that can be captured by more sophisticated SCFGs [32, 30].

The next level of RNA structural diversity can be associated to the presence of RNA 3D motifs that form non-WCF base pairs and produce stereotyped 3D conformations that appear repeatedly in structures. We have recently introduced an SCFG capable to integrate the core base-paring rules with those 3D motifs occurring loops [16].

It is an intriguing question whether a differentiable version of these more sophisticated SCFGs would be able to learn the rules of RNA loops just from sequences, and even more intriguing whether they can learn from sequences the rules of association of those 3D motifs, or even to learn the dictionary of RNA 3D motifs.

Implicit functional assumptions
Our results implicitly also demonstrate the need to train on structured sequences. In our experiments, we knew ahead of time that RNaseP RNA and the other 6 RNA families are indeed structural RNAs. We built the training sets relying on that information. What kind of results would we have obtained in the absence of that knowledge?

To test that, we trained G6 on sequences that are not structural RNAs. Figure 5b shows that when trained on random sequences, the optimization converges to G6 model parameters that reject the presence of base pairs, which is the result that one would expect and was observed already in Figures 4c-4d for a different random sample. We also trained G6 on two datasets of homologous protein-coding sequences (Figures 5c-5d). The resulting parameters exhibit correlations that are (i) distinct from those inferred from structured RNAs, and (ii) dependent on the target protein. For a given set of RNAs, these results underscore the relationship between conserved structure or function (or lack thereof) and a statistical signature identified as the trained generative model. In fact, we could use this G6-SGD training method to infer for the presence of RNA structure directly from sequences.

Our shallow model trained just on sequences allows us to learn different functional properties, but only after we had separated the sequences by their function. It is now time to expand our simple model in order to be able to infer different sequence functionalities.

Opportunities for large probabilistic models
The failure to learn WCF base pairs from random or coding sequences confirms that the statistical signal is truly characteristic of structural RNAs, and that the signal is contained in the sequences. This specificity highlights both the value and the limits of minimal, alignment-free sequence approaches: detection of "structure-like" statistical coupling is robust, but discernment of other functional signals requires at least more expressive grammars, likely beyond the SCFG category, and it may also likely require explicit alignment/covariation information.

While G6 has the capacity to learn the properties of RNA base pairing, as well as those of particular mRNAs, or even random sequences, obviously it cannot accommodate all of the different interaction patterns observed in functional RNAs at once. Many observed transcripts are still uncharacterized. Distinguishing from sequence alone the functional characteristics of a given transcript or genomic region (even when the answer is that it is just circumstantial transcription without any direct function) is still a work in progress in the field. We hypothesize that probabilistic models with an increased number of parameters trained from sequences alone should be able to classify the functional category of any input biological sequence.

Methods

Implementation details

The differentiable JAX implementations of the Inside algorithms for G6 and G5 operate in logarithmic space, and the parameters are normalized after each iteration. We used a learning rate of 0.1. Each iteration used 10 sequence batches selected at random from the training dataset. Explicitly, the grammar parameters $t_S, t_L, t_F, p_s, p_{bp}$ (see Eq. 1) were initialized using the maximum entropy principle to a discrete uniform probability distribution by default. We also used random initializations when specified, with similar results. It takes approximately 4 hours to train the G6 grammar on the database of 50 RNaseP RNAs (15.6 hours on the 400 sequences 7 Families train-set) on an Apple M3 max, 128 GB with 16 CPUs. The code is implemented as part of R-scape v2.6.0, and is provided in the Supplemental Material.

We used TORNADO v0.9.1 [32] to estimate parameters by ML from RNA structures, and to predict secondary structures by maximal expected accuracy using our customized trained parameters. TORNADO is also provided as part of the R-scape v2.6.0 software package.

We used RNAfold (with option `--MEA`) from ViennaRNA v2.7.0 [23]. We used MXFOLD2 (v0.1.2) [33] with training set "TrainSetA" [32]. We also used RiNALMo (v1.0.0) [26] with the largest model available ("giga", 650 million parameters pre-trained on 36 million non-coding RNA sequences) on a NVIDIA A100-SXM4-40GB through Google Colab. The RiNALMo structural module was trained on bpRNA-1m with 102,318 RNA structures [7], and on the ArchiveII dataset with 3,975 structures (https://rna.urmc.rochester.edu/pub/archiveII.tar.gz) [36], after all tRNAs had been removed.

Datasets

The RNA training datasets include: a collection of 225 RNaseP RNA sequences (avg. length 337 nts) from Dowell&Eddy [9], selected from the Ribonuclease P database [3]. Sequences from six other structural RNA families: 81 SRP RNAs, 97 tmRNAs, 37 telomerase RNAs, 16 Group I introns, 3 Group II introns, and 279 5S rRNAs, with annotated secondary structures previously collected by TORNADO from Dowell&Eddy [9] and the Archive database (https://rna.urmc.rochester.edu/pub/archiveII.tar.gz).

For the training set in Figure 2, we used a subsample of 50 RNaseP RNA sequences from the RNaseP dataset above. For the training set in Figure 3, 400 sequences with lengths up to 400 nts were selected at random from the total set of 729 sequences from all 7 RNA families.

As test dataset, we use the TORNADO_TrB dataset which includes 1,094 sequences from 22 different RNA families different from the 7 RNA families used for training. TORNADO_TrB dataset structural families all have crystal structures and include: 7 riboswitch families (365 seqs), 2 ribozymes (41 seqs), 5.8S rRNA (41 seqs), U1 (40 seqs) and U2 (32 seqs) spliceosomal RNAs, and 9 cis regulatory RNAs (575 seqs) [32]. The TORNADO_TrB dataset is by design structurally dissimilar to the TORNADO_TrainA dataset used by MXFOLD2, as well as to the ArchiveII dataset used by RiNALMo for the downstream task of RNA secondary structure prediction. However, The TORNADO_TrB dataset has sequence similarity to the bpRNA-1m dataset used by RiNALMo in pre-training.

For the mRNA training datasets in Figures 5c-5d, we used *S. cerevisiae* (strain S288C) intronless protein coding genes: *HSP12* (chrVI, *YFL014*) and *SBH2* (chrV, *YER019C-A*). The genes were selected at random from the set of 304 intronless genes in *S. cerevisiae* that are shorter than 400 nts. We used nhmmer [40], to identify homologs in other *Ascomycota* fungal genomes, from which we selected a random subset of up to 200 homologs per mRNA (out of 199 for *HSP12* and 248 for *SBH2*).

Statistics and reproducibility

All main source datasets and subsample datasets are provided in the Supplemental Material. Subsampling was performed at least three times.

Acknowledgments

This work was supported by NIH grant R01-GM144423 to E.R. This material is based in part upon work supported by the National Science Foundation under Grant No. UWSC13223 (R.K.K.).

We thank Marcell Szikszai for help running the software MXFOLD2 and RiNALMo, and Max Ward for insights into automatic differentiation of RNA folding models. We thank William Gao for providing the fungal mRNA sequences. We thank Sean R. Eddy and William Gao for a critical reading of the manuscript. E. R. acknowledges the hospitality of the Centro de Ciencias de Benasque Pedro Pascual, Benasque, Spain, during the completion of

this manuscript. We also thank the reviewers for their insightful comments.

Author Contributions

E.R. conceived the research. J.S.P. and R.K.K. implemented the algorithms for the G5 grammar. E.R. implemented the algorithms for the G6 grammar. E.R. performed the experiments and wrote the manuscript. All authors edited the manuscript.

Competing Interest

The authors declare no competing interests.

References

[1] M. Akiyama and Y. Sakakibara. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genomics and Bioinformatics*, 4:4, 2022.

[2] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. MacLaurin, G. Necula, A. Paszke, J. Vanderplas, and S. Wanderman-Milne. Jax: composable transformations of python+ numpy programs. 2018.

[3] J. W. Brown. The ribonuclease P database. *Nucl. Acids Res.*, 27:314, 1999.

[4] N. Chomsky. Three models for the description of language. *IRE Transact. Information Theory*, 2:113–124, 1956.

[5] S. Cocco, R. Monasson, and M. Weigt. From principal component to direct coupling analysis of coevolution in proteins: low-eigenvalue modes are needed for structure prediction. *PLOS Comput. Biol.*, 9:e1003176, 2013.

[6] P. T. da Silva, A. Karollus, J. Hingerl, G. Galindez, N. Wagner, X. Hernandez-Alias, D. Incarnato, and J. Gagneur. Nucleotide dependency analysis of DNA language models reveals genomic functional elements. *Nature Genetics*, 57:2589–2602, 2025.

[7] P. Danaee, M. Rouches, M. Wiley, D. Deng, L. Huang, and D. Hendrix. bpRNA: large-scale automated annotation and analysis of RNA secondary structure. *NAR*, 46:5381–5394, 2018.

[8] A. A. de Lajarte, Y. J. Martin des Taillades, C. Kalicki, F. F. Wightman, J. Aruda, D. Salazar, M. F. Allan, C. L'Esperance-Kerckhoff, A. Kashi, F. Jossinet, and S. Rouskin. Diverse database and machine learning model to narrow the generalization gap in RNA structure prediction. *bioRxiv*, 2024.01.24.577093, 2024.

[9] R. D. Dowell and S. R. Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5:71, 2004.

[10] S. R. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucl. Acids Res.*, 22:2079–2088, 1994.

[11] J. Eisner. Inside-Outside and Forward-Backward algorithms are just backprop (tutorial paper). *Proceedings of the Workshop on Structured Prediction for NLP*, pages 1–17, 2016.

[12] L. Fu, Y. Cao, J. Wu, Q. Peng, Q. Nie, and X. Xie. UFold: fast and accurate RNA secondary structure prediction with deep learning. *Nucl. Acids Res.*, 50:e14–e14, 2022.

[13] W. Gao, A. Yang, and E. Rivas. Thirteen dubious ways to detect conserved structural RNAs. *IUBMB Life*, 75:471–492, 2022.

[14] T. Gong and D. Bu. Language models enable zero-shot prediction of RNA secondary structure including pseudoknots. *bioRxiv*, 2024.01.27.577533, 2024.

[15] M. Justyna, M. Antczak, and M. Szachniuk. Machine learning for RNA 2D structure prediction benchmarked on experimental data. *Briefings in Bioinf*, 24:1–9, 2023.

[16] A. Karan and E. Rivas. All-at-once RNA folding with 3D motif prediction framed by evolutionary information. *Nat Methods*, 22:2094–2106, 2025.

[17] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucl. Acids Res.*, 31:3423–3428, 2003.

[18] R. K. Krueger, S. Aviran, D. H. Mathews, J. Zuber, and M. Ward. Differentiable folding for nearest neighbor model optimization. *arXiv preprint*, arXiv:2503.09085, 2025.

[19] R. K. Krueger and M. Ward. JAX-RNAfold: scalable differentiable folding. *Bioinformatics*, btaf203, 2025.

[20] K. Lari and S. J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

[21] K. Lari and S. J. Young. Applications of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 5:237–257, 1991.

[22] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379:1123–1130, 2023.

[23] R. Lorenz, S. H. Bernhart, C. Höner Zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker. ViennaRNA Package 2.0. *Algorithms Mol Biol*, 6:1748–7188, 2011.

[24] M. C. Matthies, R. Krueger, A. E. Torda, and M. Ward. Differentiable partition function calculation for RNA. *NAR*, 52:e14, 2024.

[25] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–19, 1990.

[26] R. J. Penic, T. Vlasic, R. G. Huber, Y. Wan, and M. Sikic. RiNALMo: general-purpose RNA language models can generalize well on structure prediction tasks. *arXiv*, page 2403.00043, 2024.

[27] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989.

[28] J. S. Reuter and D. H. Mathews. RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, 11:10, 2010.

[29] E. Rivas. The four ingredients of single-sequence RNA secondary structure prediction: A unifying perspective. *RNA Biol.*, 10:1185–1196, 2013.

[30] E. Rivas. RNA structure prediction using positive and negative evolutionary information. *PLOS Comput Biol*, 16(10):e1008387, 2020.

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

[32] E. Rivas, R. Lang, and S. R. Eddy. A range of complex probabilistic models for RNA secondary structure prediction that include the nearest neighbor model and more. *RNA*, 18:193–212, 2012.

[33] K. Sato, M. Akiyama, and Y. Sakakibara. RNA secondary structure prediction using deep learning with thermodynamic integration. *Nat. Commun.*, 12:941, 2021.

[34] K. Sato and M. Hamada. Recent trends in RNA informatics: a review of machine learning and deep learning for RNA secondary structure prediction and RNA drug discovery. *Briefs in Bioinf*, 24:bbad186, 2023.

[35] J. Singh, K. Paliwal, T. Zhang, J. Singh, T. Litfin, and Y. Zhou. Improved RNA secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning. *Bioinformatics*, 37:2589–2600, 2021.

[36] M. F. Sloma and D. H. Mathews. Exact calculation of loop formation probability identifies folding motifs in RNA secondary structure. *RNA*, 22:1808–1818, 2016.

[37] M. Szikszai, T.-Y. Wang, R. Krueger, D. H. Mathews, M. Ward, and S. Aviran. Deep learning for RNA secondary structure determination: Gauging generalizability and broadening the scope of traditional methods. *bioRxiv*, 2025.

[38] M. Szikszai, M. Wise, A. Datta, M. Ward, , and D. H. Mathews. Deep learning models for RNA secondary structure prediction (probably) do not generalize across families. 38:3892–3899, 2022.

[39] N. Wang, J. Bian, Y. Li, X. Li, S. Mumtaz, L. Kong, and H. Xiong. Multi-purpose RNA language modelling with motif-aware pretraining and type-guided fine-tuning. *Nat. Mach. Intell.*, 6:548–557, 2024.

[40] T. J. Wheeler and S. R. Eddy. nhmmer: DNA homology search with profile HMMs. *Bioinformatics*, 29:2487–2489, 2013.

[41] H. Yu, H. Yang, and W. and Sun. An interpretable RNA foundation model for exploring functional RNA motifs in plants. *Nat Mach Intell*, 6:1616–1625, 2024.

[42] S. Zou, T. Tao, S. Mahbub, C. N. Ellington, R. Algayres, D. Li, Y. Zhuang, H. Wang, L. Song, and E. P. Xing. A large-scale foundation model for RNA function and structure prediction. *bioRxiv*, 2024.11.28.625345, 2024.

[43] M. Zuker. Mfold web server for nucleic acid folding of hyprodization prediction. *NAR*, 31:3406–3415, 2003.

Model	Parameters	Trained on Structures?	Training Set	Train/Test overlap?	F1 Score TORNADO_TrB
G5-SGD (RNaseP)	20	No	RNaseP 225 seqs	No	< 0.01
G6-SGD (RNaseP)	21	No	RNaseP 50 seqs	No	**0.57**
G6-SGD (multi-family)			7 RNA families 400 seqs	No	0.54
G6-SGD (random)			RNaseP shuffled 225 seqs	No	< 0.01
G6-ML (RNaseP)	21	Yes	RNaseP 225 seq+struct	No	**0.56**
RNAfold (ViennaRNA)	~2,000[a]	Yes, exp.	Thermodynamic params	No	0.65
MXFOLD2	803,000	Yes	TORNADO_TrA 3,166 seq+struct	No	0.67
RiNALMo (LLM)	650M	Yes	36M seq + 102k seq/struct	Yes, pretrain	0.55

Table 1: **Comparison of RNA secondary structure models**. Performance is measured on the TORNADO_TrB test set which includes 1,094 sequences from 22 RNA families with 3D structure [32]. [a]Approximate parameter count for ViennaRNA, depends on version. The training set of 7 structural RNA families are: RNaseP RNA, SRP RNA, tmRNA, telomerase RNA, Group I intron, Group II intron and 5S rRNA. F1 scores from main text and Figures; see Methods for training and test sets descriptions. In bold, comparison of the performance on the independent TORNADO_TrB test set for G6 trained on RNase P sequences alone (G6-SGD) versus G6 trained on both RNase P sequences and structures (G6-ML).

Figure captions

Figure 1: **The G6 SCFG can model different kinds of interactions in RNA sequences. (a)** Description of the G6 grammar. Each non-terminal, L, S, and F, has a discrete probability distribution, T_S, T_L, T_F, describing the probabilities over their rules. S is the start non-terminal. (See Eq. 1 for parameter definitions.) **(b)** For a toy sequence of length 15, we show examples of three different possible interactions and how those get parsed and assigned a probability by the G6 grammar. Structure π_1 is representative of typical RNA secondary structure, forming stems that are favored by parameters satisfying $t_L \ll t_F$. Structure π_2 instead describes isolated U:C interactions. This type of structure is not representative of structured RNAs but would be assigned high probability under parameterizations with $t_F \approx 0$ and $p_{bp}(U, C) = p_{bp}(C, U) \approx 0.5$. Finally, structure π_3 describes completely non-interacting residues. Such unstructured configurations become highly probable under many arbitrary G6 parameterizations, provided that $t_L \approx 0$.

Figure 2: **G6 training by SGD on RNA sequences but not structures.** The G6 SCFG trained by stochastic gradient descent on a collection of 50 sequences selected at random from a database of RNaseP sequences [9]. **(a)** For a given set of G6 parameters, the loss is defined as the average over the training sequences of the negative log-probability of each sequence (see Eq. 6). **(b)** Performance of the G6 SGD-optimized parameters (per epoch) predicting the secondary structures of the 1,094 RNA sequences in the TORNADO_TrB test set (blue). We compare to the performance of G6 parameterized by ML which uses both sequence and structure information (orange). We also compare to the predictions using the program RNAfold from ViennaRNA (brown). **(c)** The probability distribution parameters of G6 (base pair probabilities, single residue probability and non-terminal distributions). In blue, SGD-parameters (trained only from sequences): uniformly distributed at initialization (epoch 0), and after sequence-only SGD optimization (epoch 99, see Eq. 7). In orange, parameter values as trained by maximum likelihood (from sequences and structures). (Results for a random distribution initialization, provided in the Supplemental Material, produced similar results.)

Figure 3: **G6 SGD training on non-homologous structural RNA sequences.** The G6 SCFG trained by stochastic gradient descent on a sample of 400 sequences selected at random from a collection of 729 sequences from 7 different non-homologous structural RNA families: RNaseP RNA, SRP RNA, tmRNA, telomerase RNA, Group I intron, Group II intron and 5S rRNA. Parameter optimization is performed as described in Eq. 7. **(a)** For a given set of G6 parameters, the loss is the average over the training sequences of the negative log-probability of each sequence (see Eq. 6). **(b)** Performance of the G6 SGD-optimized parameters (per epoch) on predicting the secondary structures of the TORNADO_TrB datasets that includes 1,094 RNA sequences from 22 different RNA families with known 3D crystal structures (blue). We compare to the performance of G6 parameterized by ML which uses both sequence and structure information (orange). We also compare to the predictions using the program RNAfold from ViennaRNA (brown). **(c)** The probability distribution parameters of G6 (base pair probabilities, single residue probability and non-terminal distributions). In orange, parameter values as trained by maximum likelihood (from sequences and structures). In blue, SGD-parameters (trained only from sequences): randomly distributed at initialization (epoch 0), and after sequence-only SGD-optimization (epoch 99, see Eq. 7). (Results for a uniform distribution initialization, provided in the Supplemental Material, produced similar results.)

Figure 4: **SGD training on a different SCFG (a)-(b), and on random sequences (c)-(d).** **(a)** The G5 grammar, with 20 independent parameters, trained on the database of 225 RNaseP RNA sequences. **(b)** Testing on the TORNADO_TrB dataset with the G5 SGD parameters for each epoch. Insert includes the G5 SGD parameters at epoch 99 (blue), and the maximum likelihood parameters trained from sequences and structures (orange). **(c)** The G6 grammar trained on a shuffled instance of the training set consisting of 225 RNaseP sequences. **(d)** Testing on the TORNADO_TrB dataset with the G6 SGD parameters for each epoch. The insert includes the optimized probability distribution of G6 at epoch 99. In red, we highlight the value of the rule $L \to aFb$ which initiates a helix ($t_L = 0.00418$ at epoch 99).

Figure 5: **G6 SGD training on sequences with different functions.** We compare the results of training the G6 grammar on a similar number of sequences under three different functional constraints: structural RNAs, mRNA sequences, and random sequences. Insert: in blue, parameter values at epoch 99; in green, we highlight when the optimized parameters support the presence of nucleotide interactions, and whether the pairs follow WCF statistics; in red, we indicate when the optimized parameters do not support nucleotide interactions. **(a)** The training set is the database of 225 RNaseP RNAs. **(b)** The training set is a shuffled version of the database of 225 RNaseP RNAs, different from the one in Figures 4c-4d. **(c)** The training set is a collection of 199 homologs of *S. cerevisiae* intronless protein-coding gene HSP12. **(d)** The training set is a collection of 200 homologs of *S. cerevisiae* intronless protein-coding gene SBH2.

(a)

G6 Grammar

Start

S -> L S

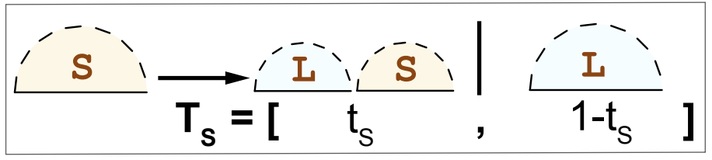
S -> L

L -> ∘ F ∘ new stem starts, adds first pair

L -> ∘ a single residue

F -> ∘ F ∘ stem adds a base pair

F -> L S stem ends



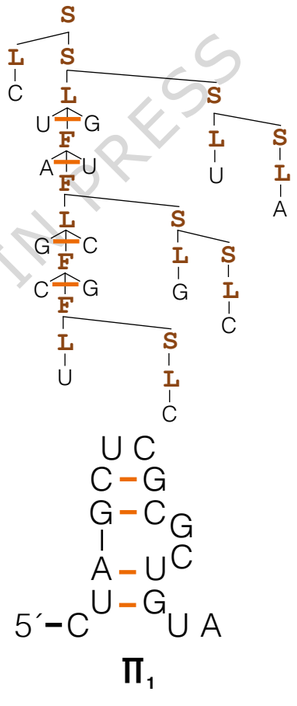
∘ a residue

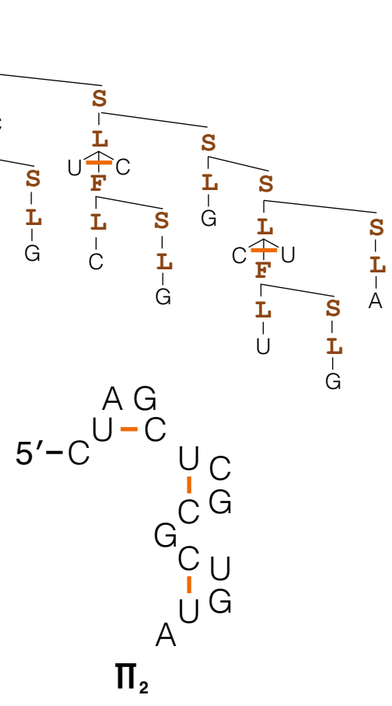
∘ ∘ a pair interaction

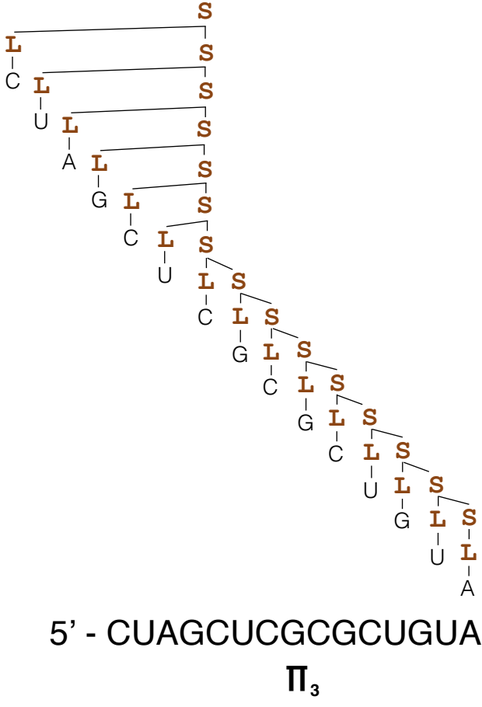
S,L,F non-terminals (NT) transform following one of the allowed rules (r)

$\sum_r T^r_{NT} = 1$

(b) sequence 5'–CUAGCUCGCGCUGUA

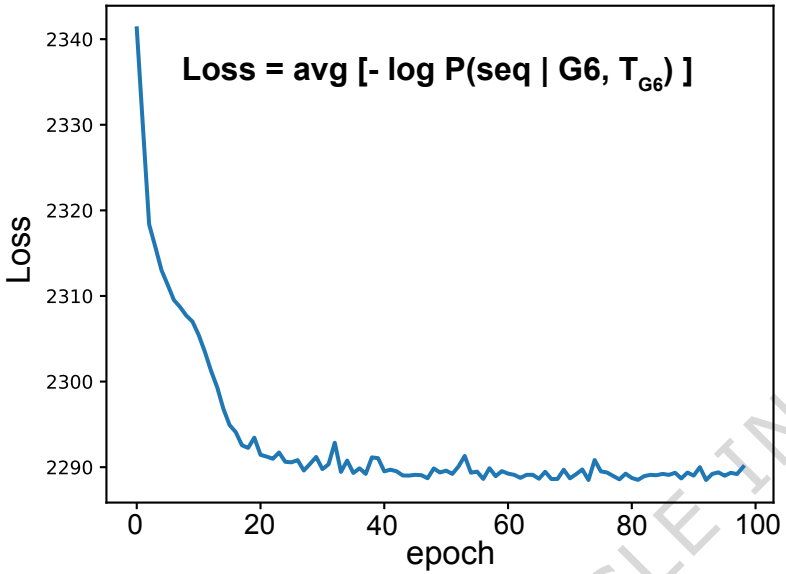




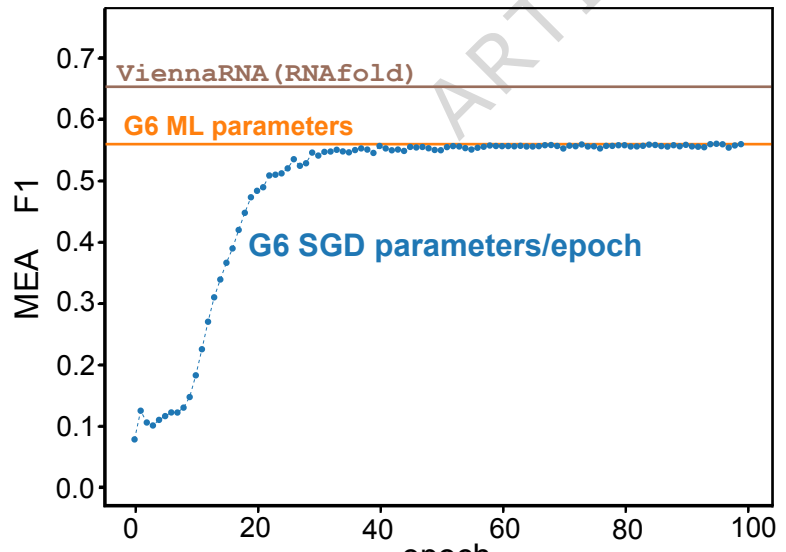


G6 SCFG

(a) Training: RNaseP RNA 50 seqs

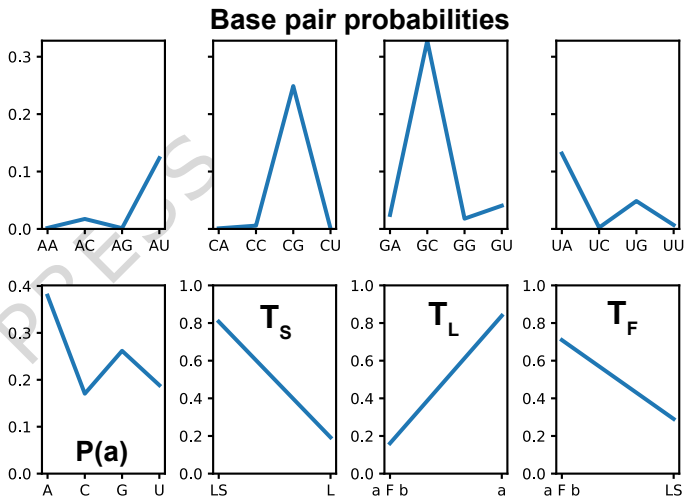


(b) Testing: TORNADO_TrB

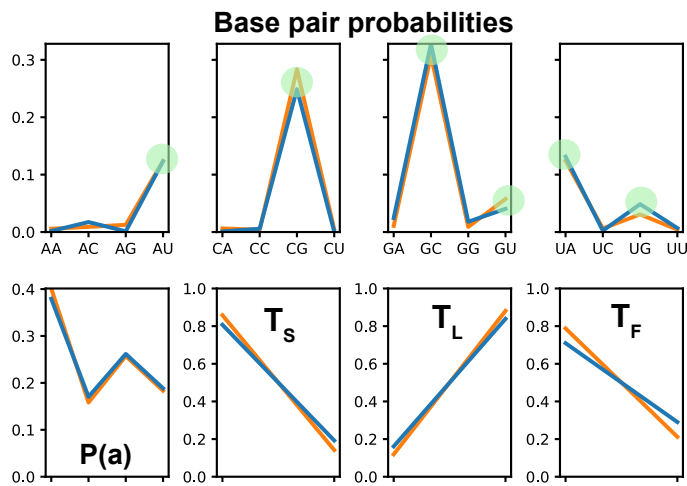


(c) Parameter optimization

SGD parameters epoch 99 (no structure)

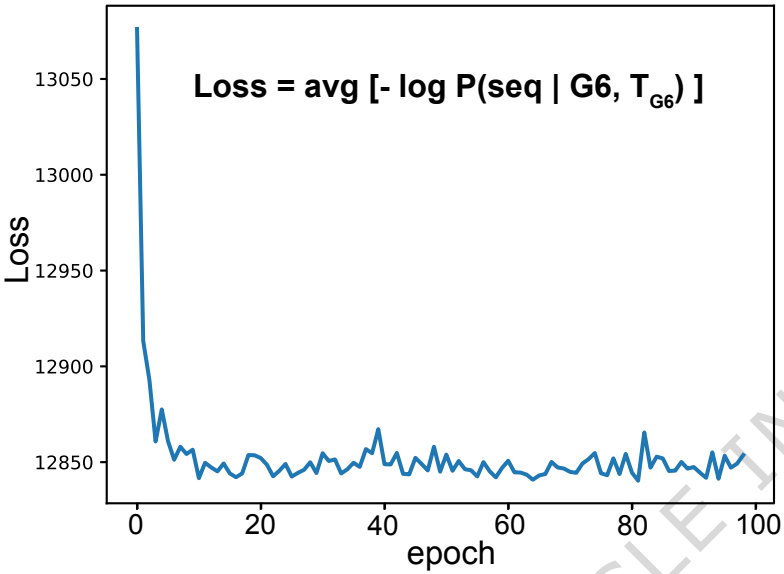


SGD parameters epoch 99 (no structure)
ML parameters (structure)

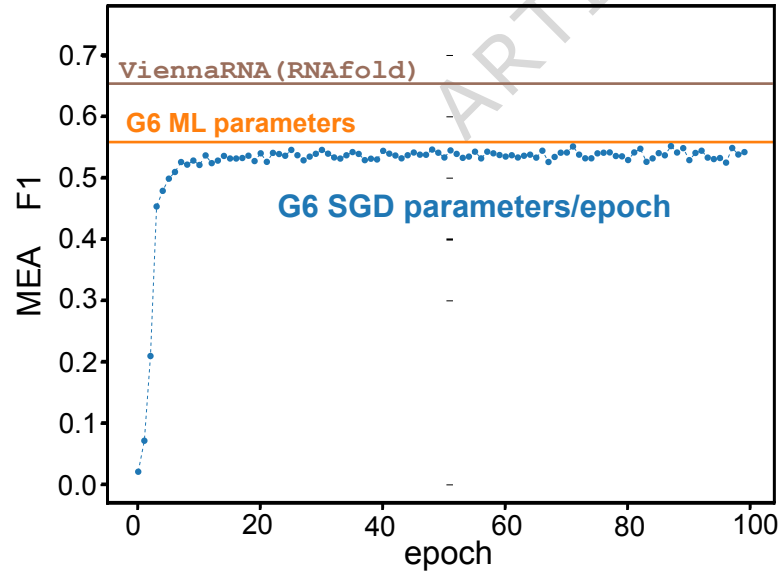


G6 SCFG

(a) Training: 7 RNA Fams 400 seqs

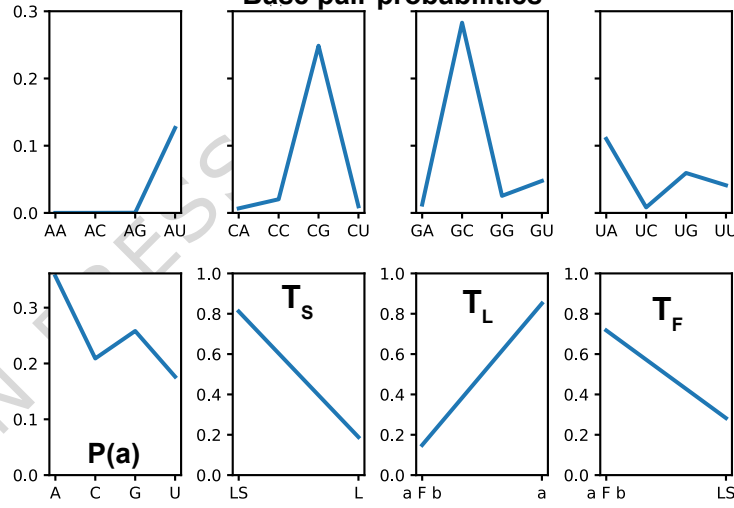


(b) Testing: TORNADO_TrB

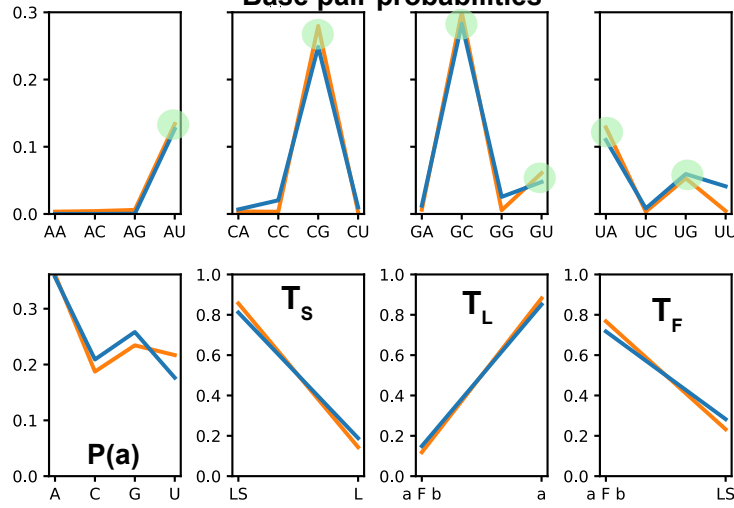


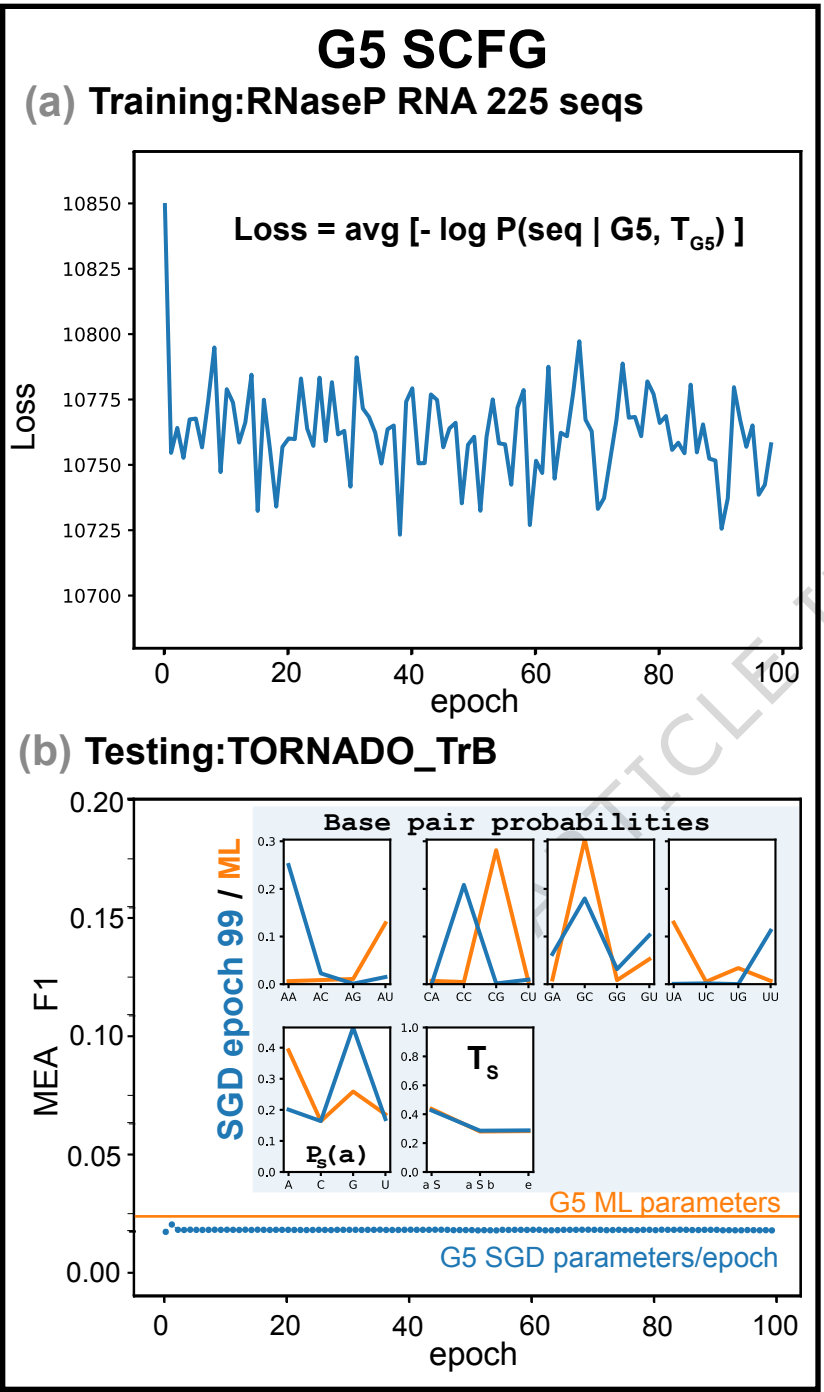
(c) Parameter optimization

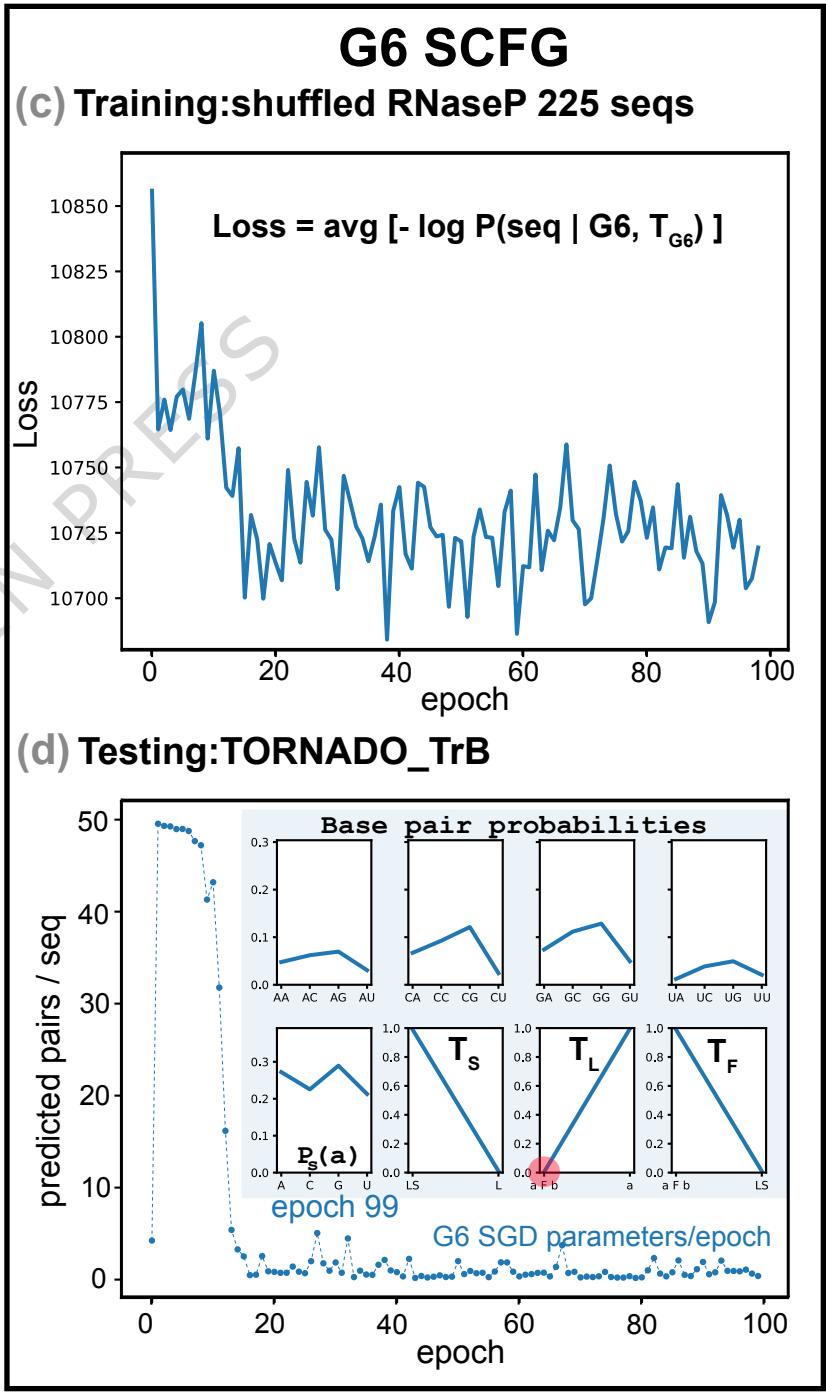
SGD parameters epoch 99 (no structure)

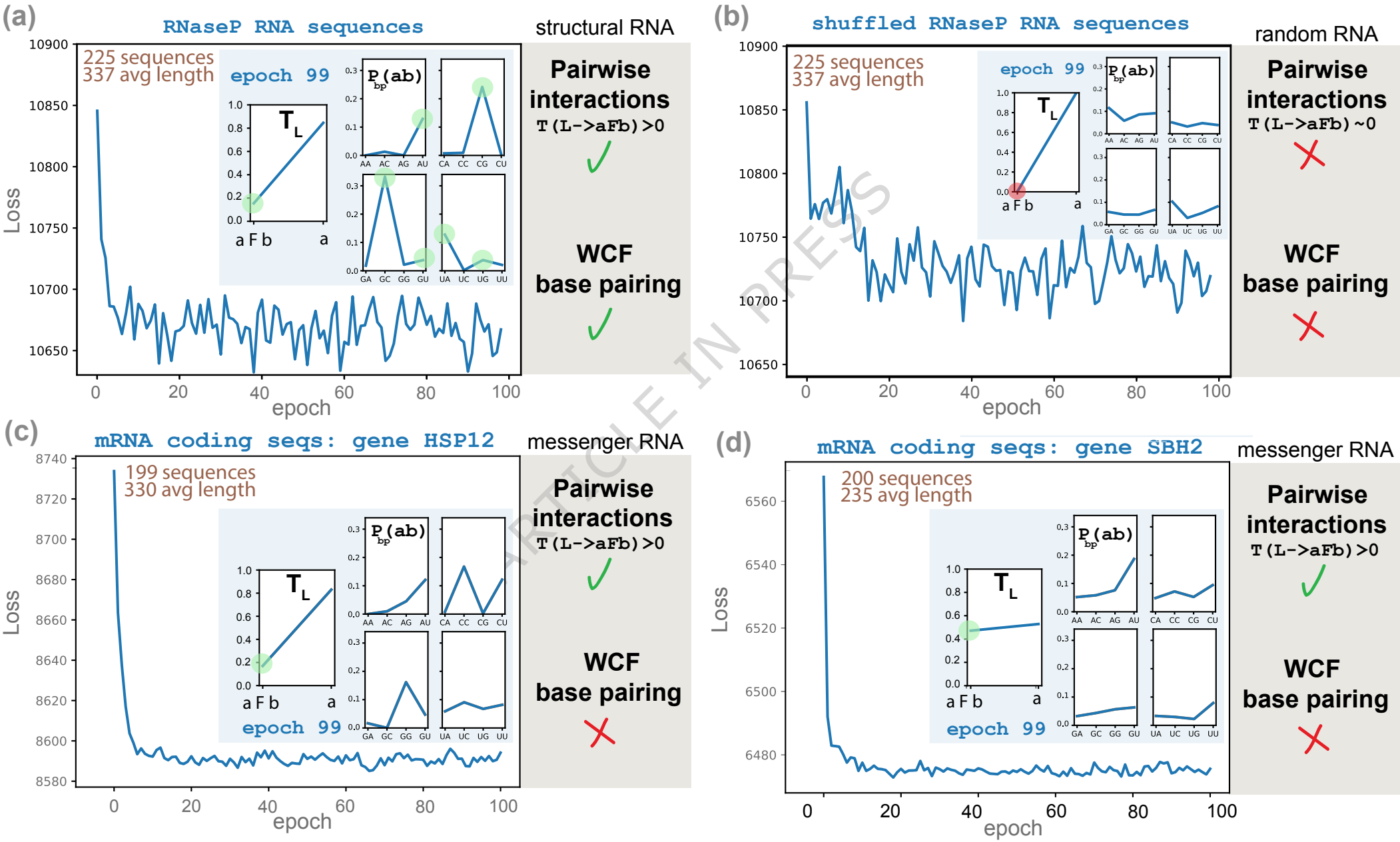


SGD parameters epoch 99 (no structure)
ML parameters (structure)









$$\text{Loss} = -\sum_i \log P(\text{seq}_i \mid G6, T)$$

RNA seq1
RNA seq2
RNA seq3
RNA seq4
RNA seq5
RNA seq6
RNA seq7
RNA seq8
RNA seq9
⋮
RNA seqN

G6 grammar T

S -> L S
S -> L
t_s

L -> o F o
L -> o
t_L
p(o)

F -> o F o
F -> L S
t_F
p(oo)

T

Inside algorithm

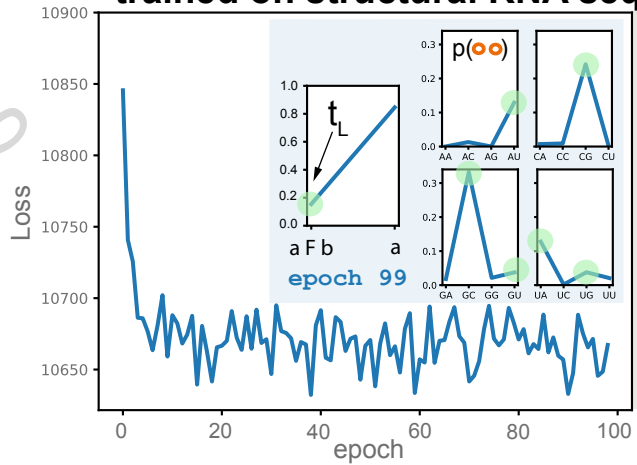
P(RNA seq1 | G6,T)
P(RNA seq2 | G6,T)
P(RNA seq3 | G6,T)
P(RNA seq4 | G6,T)
P(RNA seq5 | G6,T)
P(RNA seq6 | G6,T)
P(RNA seq7 | G6,T)
P(RNA seq8 | G6,T)
P(RNA seq9 | G6,T)
⋮
P(RNA seqN | G6,T)

dLoss/dT

backpropagation

JAX automatic differentiation

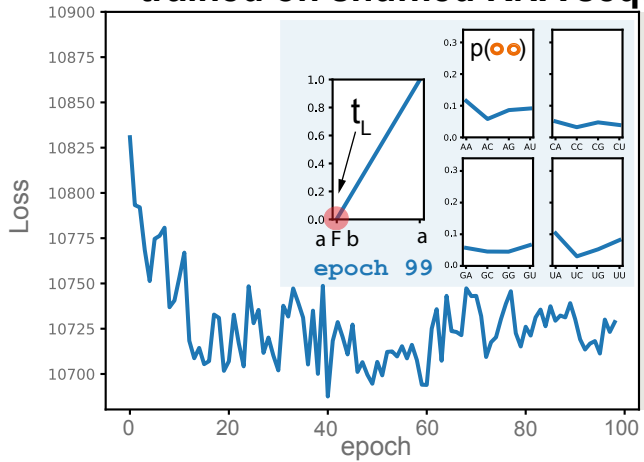
trained on structural RNA sequences



Pairwise interactions
T(L->aFb)>0
✓

WCF base pairing
✓

trained on shuffled RNA sequences



Pairwise interactions
T(L->aFb)~0
✗

WCF base pairing
✗