Rich Parameterization Improves RNA Structure Prediction

Shay Zakov*, Yoav Goldberg*, Michael Elhadad, and Michal Ziv-Ukelson**

Department of Computer Science, Ben-Gurion University of the Negev, Israel
{zakovs, yoavg, elhadad, michaluz}@cs.bgu.ac.il

Abstract.

Motivation. Current approaches to RNA structure prediction range from physics-based methods, which rely on thousands of experimentally-measured thermodynamic parameters, to machine-learning (ML) techniques. While the methods for parameter estimation are successfully shifting toward ML-based approaches, the model parameterizations so far remained fairly constant and all models to date have relatively few parameters. We propose a move to much richer parameterizations.

Contribution. We study the potential contribution of increasing the amount of information utilized by folding prediction models to the improvement of their prediction quality. This is achieved by proposing novel models, which refine previous ones by examining more types of structural elements, and larger sequential contexts for these elements. We argue that with suitable learning techniques, not being tied to features whose weights could be determined experimentally, and having a large enough set of examples, one could define much richer feature representations than was previously explored, while still allowing efficient inference. Our proposed fine-grained models are made practical thanks to the availability of large training sets, advances in machine-learning, and recent accelerations to RNA folding algorithms.

Results. In order to test our assumption, we conducted a set of experiments that asses the prediction quality of the proposed models. These experiments reproduce the settings that were applied in recent thorough work that compared prediction qualities of several state-of-the-art RNA folding prediction algorithms. We show that the application of more detailed models indeed improves prediction quality, while the corresponding running time of the folding algorithm remains fast. An additional important outcome of this experiment is a new RNA folding prediction model (coupled with a freely available implementation), which results in a significantly higher prediction quality than that of previous models. This final model has about 70,000 free parameters, several orders of magnitude more than previous models. Being trained and tested over the same comprehensive data sets, our model achieves a score of 84% according to the F_1-measure over correctly-predicted base-pairs (i.e. 16% error rate), compared to the previously best reported score of 70% (i.e. 30% error rate). That is, the new model yields an error reduction of about 50%.

Availability: Additional supporting material, trained models, and source code are available through our website at `http://www.cs.bgu.ac.il/~negevcb/contextfold`.

---

* These authors contributed equally to the paper.
** Corresponding author.

1 Introduction

Within the last few years, non-coding RNAs have been recognized as a highly abundant class of RNAs. These RNA molecules do not code for proteins, but nevertheless are functional in many biological processes, including localization, replication, translation, degradation, regulation and stabilization of biological macromolecules [1–3]. It is generally known that much of RNAs functionalities depend on its structural features [3–6]. Unfortunately, although massive amounts of sequence data are continuously generated, the number of known RNA structures is still limited, since experimental methods such as NMR and Crystallography require expertise and long experimental time. Therefore, computational methods for predicting RNA structures are of significant value [7–9]. This work deals with improving the quality of computational RNA structure prediction.

RNA is typically produced as a single stranded molecule, composed as a sequence of *bases* of four types, denoted by the letters A, C, G, and U. Every base can form a hydrogen bond with at most one other base, where bases of type C typically pair with bases of type G, A typically pairs with U, and another weaker pairing can occur between G and U. The set of formed base-pairs is called the *secondary structure*, or the *folding* of the RNA sequence (see Fig. 1), as opposed to the *tertiary structure* which is the actual three dimensional molecule structure. Paired bases almost always occur in a nested fashion in RNA foldings. A folding which sustains this property is called a *pseudoknot-free folding*. In the rest of this work we will consider only pseudoknot-free foldings.

RNA structure-prediction (henceforth *RNA folding*) is usually formulated as an optimization problem, where a score is defined for every possible folding of the given RNA sequence, and the predicted folding is one that maximizes this score. While finding a folding which maximizes the score under an arbitrary scoring function is intractable due to the magnitude of the search space, specific classes of scoring functions allow for an efficient solution using dynamic programming [10]. Thus, in the standard scoring approach, the score assigned to a folding is composed as the sum of scores of local structural elements, where the set of local elements are chosen to allow efficient dynamic programming inference.[1]

Several scoring models were introduced over the past three decades, where these models mainly differ in the types of structural elements they examine (the *feature-set*), and the scores they assign to them. A simple example of such a model is the one of Nussinov and Jacobson [10], which defines a single feature corresponding to a canonical Watson-Crick base-pair in the structure (i.e. base-pairs of the form G-C and A-U, and their respective reversed forms). The score of each occurrence of the feature in the structure is 1, and thus the total score of a folding is simply the number of canonical base-pairs it contains. A more complex model, which is commonly referred to as the *Turner99 model*, was defined in [11] and is widely used in many RNA structure prediction systems [7–9]. This model distinguishes between several different types of structural elements corresponding to unpaired bases, base-pairs which participate in different types of loops, loop-length elements, etc. In addition, every structural element can be mapped to one of several features, depending on some sequential context (e.g. the type of nucleotides at base-pair endpoints and within their vicinity), or other values (e.g. the specific loop length, internal-loop asymmetry value, etc.).

The parameter values (i.e. scores of each local element) are traditionally obtained from wet-lab experiments [12], reflecting the thermodynamics free energy theory [13, 14]. However, the increasing availability of known RNA structures in current RNA databases (e.g. [15]) makes it possible to conduct an improved, fine-tuned parameter estimation based on machine-learning (ML) techniques, resulting in higher prediction accuracies. These methods examine large *training sets*, composed of RNA sequences and their known structures [16–19].

[1] Some scoring models also utilize homology information with respect to two or more sequences. Such comparative approaches are beyond the scope of this work.

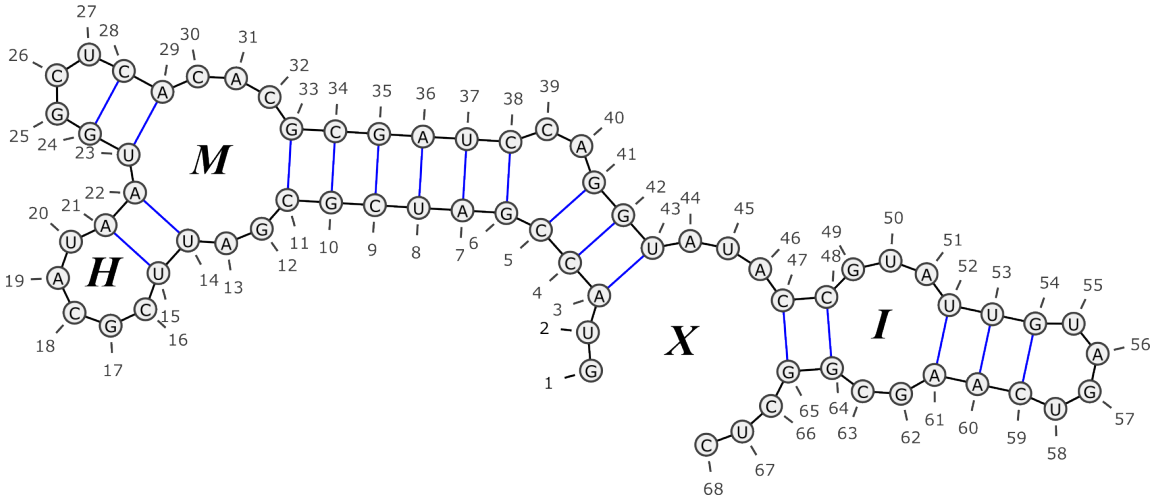

Fig. 1. RNA secondary structure. The figure exemplifies a *secondary structure* of an RNA sequence. Consecutive bases in the sequence are connected with (short) black edges, where base-pairs appear as blue (longer) edges. The labels within the loops stand for loop types, where H denotes a *hairpin*, I denotes an *internal-loop*, M denotes a *multi-loop*, and X denotes an *external-loop*. Drawing was made using the VARNA tool [22].

Do *et al.* [18] proposed to set the parameters by fitting an SCFG-based conditional log-linear model to maximize the conditional log-likelihood of a set of known structures. The approach was extended in [20] to include automatic tuning of regularization hyperparameters. Andronescu et al. [19] and later in [21] used the Turner99 model, and applied Constraint-Generation (CG) and Boltzman-likelihood (BL) methods for the parameters estimation. These methods start with a set of wet-lab parameter values, and refine them using a training set of RNA sequences and their known structures, and an additional data set containing triplets of sequences, structures and their measured thermodynamic energies. The parameters derived by [21] yield the best published results for RNA folding prediction to date, when tested on a large structural data set.

While the methods for parameter estimation are successfully shifting toward ML-based approaches, the model parameterizations have so far remained fairly constant. Originating from the practice of setting the parameter values using wet-lab measurements, all models to date have relatively few parameters, where each parameter corresponds to the score of one particular local configuration.

Our Contribution. We propose a move to much richer parameterizations, which is made possible due to the availability of large training sets [23] combined with advances in machine-learning [24, 25], and supported in practice by recent accelerations to RNA folding algorithms [27, 26]. The scoring models we apply refine previous models by examining more types of structural elements, and a larger sequential context for these elements. Based on this, similar structural elements could get scored differently in different sequential contexts, and different structural elements may get similar scores in similar sequential contexts.

We base our models on the structural elements defined by the Turner99 model in order to facilitate efficient inference. However, in our models, the score assigned to each structural element is itself composed of the sum of scores of many fine-grained local features that take into account

portions of larger structural and sequential context. While previous models usually assign a single score to each element (e.g. the base-pair between positions 5 and 41 in Fig. 1), our models score elements as a sum of scores of various features (e.g., the base-pair $(5, 41)$ has the features of being a right-bulge closing base-pair, participating in a stem, having its left endpoint centering a CCG sequence, starting a CGA sequence, and various other contextual factors).

Our final model has about 70,000 free parameters, several orders of magnitude more than previous models. We show that we are still able to effectively set the parameter values based on several thousands of training examples. Our resulting models yield a significant improvement in the prediction accuracy over the previous best results reported by [21], when trained and tested over the same data sets. Our `ContextFold` tool, as well as the various trained models, are freely available on our website and allow for efficient training and inference. In addition to reproducing the results in this work, it also provides flexible means for further experimenting with different forms of richer parameterizations.

2 Preliminaries and Problem Definition

For an RNA sequence x, denote by $\mathcal{Y}_x$ the domain of all possible foldings of x. We represent foldings as sets of index-pairs of the form $(i, j), i < j$, where each pair corresponds to two positions in the sequence such that the bases in these positions are paired. We use the notation (x, y) for a *sequence-folding* pair, where x is an RNA sequence and y is the folding of x. A *scoring model* G is a function that assigns real-values to sequence-folding pairs (x, y). For a given scoring model G, the RNA folding prediction problem is defined as follows[2]: *given an RNA sequence x, find a folding $\hat{y} \in \mathcal{Y}_x$ s.t. $G(x, \hat{y})$ is maximal.* Such a folding $\hat{y}$ will be called an *optimal folding* for x with respect to G. A *folding prediction* (or a *decoding*) algorithm f_G is an algorithm that solves the folding prediction problem, i.e.

$$\hat{y} = f_G(x) = \text{argmax}_{y \in \mathcal{Y}_x} \{G(x, y)\} \tag{1}$$

Denote by ρ a *cost function* measuring a distance between two foldings, satisfying $\rho(y, y) = 0$ for every y and $\rho(y, y') > 0$ for every $y \neq y'$. This function indicates the cost associated with predicting the structure y' where the real structure is y. For RNA folding, this cost is usually defined in terms of sensitivity, PPV and F-measure (see Sec. 5). Intuitively, a good scoring model G is one such that $\rho(y, f_G(x))$ is small for arbitrary RNA sequences x and their corresponding true foldings y.

In order to allow for efficient computation of f_G, the score $G(x, y)$ is usually computed on the basis of various local features of the pair (x, y). These features correspond to some structural elements induced by y, possibly restricted to appear in some specific sequential context in x. An example of such a feature could be the presence of a stem where the first base-pair in the stem is C-G and it is followed by the base-pair A-U. We denote by Φ the set of different features which are considered by the model, where Φ defines a finite number N of such features. The notation $\Phi(x, y)$ denotes the *feature representation* of (x, y), i.e. the collection of occurrences of features from Φ in (x, y). We assume that every occurrence of a feature is assigned a real-value, which reflects the "strength" of the occurrence. For example, we may define a feature corresponding to the interval of unpaired bases within a hairpin, and define that the value of an occurrence of this feature is the log of the interval length. For binary features such as the stem-feature described above, occurrence values are taken to be 1.

[2] In models whose scores correspond to free energies, the score optimization is traditionally formulated as a *minimization* problem. This formulation can be easily transformed to the *maximization* formulation that is used here.

In order to score a pair (x, y), we compute scores for feature occurrences in the pair, and sum up these scores. Each feature in Φ is associated with a corresponding score (or a *weight*), and the score of a specific occurrence of a feature in (x, y) is defined to be the value of the occurrence multiplied by the corresponding feature weight. $\Phi(x, y)$ can be represented as a vector of length N, in which the ith entry ϕ_i corresponds to the ith feature in Φ. Since the same feature may occur several times in (x, y) (e.g., two occurrences of a stem), the value ϕ_i is taken to be the sum of values of the corresponding feature occurrences. Formally, this defines a linear model:

$$G(x, y) = \sum_{\phi_i \in \Phi(x,y)} \phi_i \mathbf{w}_i = \Phi(x, y)^T \cdot \mathbf{w} \tag{2}$$

where $\mathbf{w}$ is a weight vector in which $\mathbf{w}_i$ is the weight of the ith feature in Φ, and $\cdot$ is the dot-product operator. The vector $\mathbf{w}$ of N feature weights is called the *model parameters*, and Φ is thus referred to as the *model parameterization*. We use the notation $G_{\Phi,\mathbf{w}}$ to indicate a scoring model G with the specific parameterization Φ and parameters $\mathbf{w}$.

The predictive quality of a model of the form $G_{\Phi,\mathbf{w}}$ depends both on the parameterization Φ, defining which features are examined, and on the specific weights in $\mathbf{w}$ which dictate how to score these features. Having fixed a model parameterization Φ, the model parameter values $\mathbf{w}$ can be set based on scientific intuitions and on biological measurements (as done in thermodynamic based models), or based on statistical estimation over observed (x, y) pairs using machine-learning techniques. Aiming to design better models of the form $G_{\Phi,\mathbf{w}}$, there is a need to balance between (a) choosing a rich and informative parameterization Φ so that with optimal weights $\mathbf{w}$ the prediction quality of the model will be as good as possible, (b) allowing for a tractable folding prediction algorithm $f_{G_{\Phi,\mathbf{w}}}$, and (c) being able to estimate optimal (or at least "good") weight parameters $\mathbf{w}$.

3 Feature Representations

We argue that with suitable learning techniques, not being tied to features whose weights could be determined experimentally, and having a large enough set of examples (x, y) such that y is the true folding of x, one could define much richer feature representations than was previously explored, while still allowing efficient inference. These richer representations allow the models to condition on many more fragments of information when scoring the various foldings for a given structure x, and consequently come up with better predictions. This section describes the types of features incorporated in our models.

All examples in this section refer to the folding depicted in Fig. 1, and we assume that the reader is familiar with the standard RNA folding terminology. The considered features broadly resemble those used in the Turner99 model, with some additions and refinements described below, and allow for an efficient Zuker-like dynamic programming folding prediction algorithm [28]. Formal definitions of the terms we use, as well as the exact feature representations we apply in the various models, can be found in the online supplementary material.

We consider two kinds of features: *binary* features, and *real-valued* features.

Binary features. Binary features are features for which occurrence values are always 1, thus the scores of such occurrences are simply the corresponding feature weights. These features occur in a sequence-folding pair whenever some specific *structural element* is present in some specific *sequential context*. The set of structural elements contains base-pairs and unpaired bases, which appear in loops of specific types, for example a multi-loop closing a base-pair, or an unpaired base within a hairpin. A sequential context describes the identities of bases appearing in some given offsets with respect to the location of the structural element in the sequence, e.g. the presence of

bases of types C and G at the two endpoints (i,j) of a base-pair. A complete example of such a binary feature is `hairpin_base_0=G_+1=C_-2=U`, indicating the presence of an unpaired-base of type G inside a hairpin at a sequence position i, while positions $i+1$ and $i-2$ contain bases of types C and U respectively. This feature will be generated for the unpaired-bases at positions 17 and 25 in Fig.1.

In contrast to previous models, where each structural element is considered with respect to a *single* sequential context (and producing exactly one scoring term), in our models the score of a structural element is itself a linear combination of different scores of various (possibly overlapping) pieces of information. For example, a model may contain the features `hairpin_base_-1=C_-2=U` and `hairpin_base_0=G_+1=C_-1=C` which will also be generated for the unpaired-base in position 17 (thus differentiating it from the unpaired base at position 25). Note that the appearance of a C-base at relative position -1 appears in both of these features, demonstrating overlapping information regarded by the two features. The decomposition of the sequential context into various overlapping fragments allows us to consider *broader* and *more refined* sequential contexts compared to previous models.

The structural information we allow is also more refined than in previous models: we consider properties of the elements, such as *loop lengths* (e.g. a base-pair which closes a hairpin of length 3 may be scored differently than a base-pair which closes a hairpin of length 4, even if the examined sequential contexts of the two base-pairs are identical), and examine the *two orientations* of each base-pair (e.g. the base-pair $(11, 33)$ may be considered as a *C-G closing* base-pair of the multi-loop marked with an M, and it may also be considered as a *G-C opening* base-pair of the stem that consists of the base-pair $(10, 34)$ in addition to this base-pair). We distinguish between unpaired bases at the "shorter" and "longer" sides of an internal-loop, and distinguish between unpaired bases in external intervals, depending on whether they are at the 5'-end, 3'-end, or neither (i.e. the intervals 1-2, 66-68, and 44-46, respectively). Notably, our refined structural classification allows for the generalization of the concept of "bulges", where, for example, it is possible to define special internal-loop types such that the left length of the loop is exactly k (up to some predefined maximum value for k), and the right length is at least k, and to assign specific features for unpaired bases and base-pairs which participate in such loops.

Real-valued features. Another kind of structural information not covered by the binary unpaired bases and base-pairs features is captured by a set of real-valued *length* features. These features are generated with respect to intervals of unpaired bases, such as the three types of external intervals (as mentioned above), intervals of unpaired bases within hairpins (e.g. the interval 16-20), and intervals of unpaired bases within internal-loops up to some predefined length bound[3] (e.g. the interval 49-51). The value of an occurrence of a length feature can be any function of the corresponding interval length. In this work, we follow the argumentation of [11] and set the values to be the log of the interval length. As mentioned above, the structural base-pairs and unpaired-bases information is conjoined with various pieces of contextual information. We currently do not consider contextual information for the real-valued length features.

Our features provide varied sources of structural and contextual informations. We rely on a learning algorithm to come up with suitable weights for all these bits and pieces of information.

[3] In this sense, internal-loop lengths are not restricted here as done in some other models, where arbitrary-length internal-loops are scored with respect to their unpaired bases and terminating base-pairs, and length-depended corrections are added to the scores of relatively "short" loops.

4 Learning Algorithm

The learning algorithm we use is inspired by the discriminative structured-prediction learning framework proposed by Collins [24] for learning in natural language settings, coupled with a passive-aggressive online learning algorithm [25]. This class of algorithms adapt well to large feature sets, do not require the features to be independent, and were shown to perform well in numerous natural language settings [29–31]. Here we demonstrate they provide state of the art results also for RNA folding. The learning algorithm is simple to understand and to implement, and has strong formal guarantees. In addition, it considers one training instance (sequence-folding pair) at a time, making it scale linearly in the number of training examples in terms of computation time, and have a memory requirement which depends on the longest training example.

Recall the goal of the learning algorithm: given a feature representation Φ, a folding algorithm $f_{G_{\Phi,\mathbf{w}}}$, a cost function ρ and a set of training instances S_{train}, find a set of parameter values $\mathbf{w}$ such that the expected cost $\rho(y, f_{G_{\Phi,\mathbf{w}}}(x))$ over unseen sequences x and their true foldings y is minimal.

The algorithm works in rounds. Denote by $\mathbf{w}^0 = 0$ the initial values in the parameter vector maintained by the algorithm. At each iteration i the algorithm is presented with a pair $(x, y) \in S_{train}$. It uses its current parameters $\mathbf{w}^{i-1}$ to obtain $\hat{y} = f_{G_{\Phi,\mathbf{w}^{i-1}}}(x)$, and updates the parameter vector according to:

$$\mathbf{w}^i = \begin{cases} \mathbf{w}^{i-1}, & \rho(y,\hat{y}) = 0, \\ \mathbf{w}^{i-1} + \tau_i \Phi(x,y) - \tau_i \Phi(x,\hat{y}), & \text{otherwise}, \end{cases} \tag{3}$$

where:

$$\tau_i = \min\left(1, \frac{\Phi(x,\hat{y})^T \cdot \mathbf{w}^{i-1} - \Phi(x,y)^T \cdot \mathbf{w}^{i-1} + \sqrt{\rho(y,\hat{y})}}{||\Phi(x,\hat{y}) - \Phi(x,y)||^2}\right).$$

This is the PA-I update for cost sensitive learning with structured outputs described in [25]. Loosely, equation 3 attempts to set $\mathbf{w}^i$ such that the correct structure y would score higher than the predicted structure $\hat{y}$ with *margin* of at least the square-root of the difference between the structures, while trying to minimize the change from $\mathbf{w}^{i-1}$ to $\mathbf{w}^i$. This is achieved by decreasing the weights of features appearing only in the predicted structure, and increasing the weights of features appearing only in the correct structure. Even though one example is considered at a time, the procedure is guaranteed to converge to a good set of parameter values. For the theoretical convergence bounds and proofs see [25]. In practice, due to the finite size of the training data, the algorithm is run for several passes over the training set.

In order to avoid over-fitting of the training data, the final $\mathbf{w}$ is taken to be the average over all $\mathbf{w}^i$ seen in training. That is, $\mathbf{w}^{final} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{w}^i$, where K is the number of processed instances. This widely used practice introduced in [32] improves the prediction results on unseen data.

5 Experiments

In order to test the effect of richer parametrizations on RNA prediction quality, we have conducted 5 learning experiments with increasingly richer model parameterizations, ranging from 226 active features for the simplest model to about 70,000 features for the most complex one.

5.1 Experiment Settings

Feature representations. As described in Section 3, our features combine structural and contextual information. We begin with a baseline model (`Baseline`) which includes a trivial amount of contextual information (the identities of the two bases in a base-pair) and a set of basic structural elements such as *hairpin unpaired base, internal-loop unpaired base, stem closing base-pair, multi-loop closing base-pair, hairpin length*, etc. This baseline model has a potential of inducing up to 1,919 different features, but in practice about 220 features are assigned a non-zero weight after the training process, a number which is comparable to the number of parameters used in previously published models.

We then enrich this basic model with varying amounts of structural (`St`) and contextual (`Co`) information. $\texttt{St}^{med}$ adds distinction between various kinds of short loops, considers the two orientations of each base-pair, and considers unpaired bases in external intervals, and $\texttt{St}^{high}$ adds further length-based loop type refinements. Similarly, $\texttt{Co}^{med}$ considers also the identities of unpaired bases and the base types of the adjacent pair $(i+1, j-1)$ for each base-pair (i, j), while $\texttt{Co}^{high}$ considers also the *neighbors* of unpaired bases and *more configurations* of neighbors surrounding a base-pair.

The models $\texttt{St}^{med}\texttt{Co}^{med}$, $\texttt{St}^{high}\texttt{Co}^{med}$, $\texttt{St}^{med}\texttt{Co}^{high}$ and $\texttt{St}^{high}\texttt{Co}^{high}$ can potentially induce about 14k, 30k, 86k and 205k parameters respectively, but in practice much fewer parameters are assigned non-zero values after training, resulting in effective parameter counts of 4k, 7k, 38k and 70k. The exact definition of the different structural elements and sequential contexts considered in each model are provided in the online supplementary material.

Evaluation Measures. We follow the common practice and assess the quality of our predictions based on the *sensitivity*, *positive predictive value* (PPV), and F_1*-measure* metrics, defined as $\frac{|y \cap \hat{y}|}{|y|}$, $\frac{|y \cap \hat{y}|}{|\hat{y}|}$, and $\frac{2|y \cap \hat{y}|}{|y|+|\hat{y}|}$ respectively, for a known structure y and a predicted structure $\hat{y}$, where $|y|$ is the number of base-pairs in a structure y, and $|y \cap \hat{y}|$ is the number of base-pairs appearing in both structures. Sensitivity is the proportion of correctly predicted base-pairs among all true base-pairs, PPV is the proportion of correctly predicted base-pairs among all predicted base-pairs, and F_1 is a value which balances sensitivity and PPV. All of the measures range in value from 0 to 1, where a value of 1 indicates that the true and predicted structure are identical, and a value of 0 means that none of the true base-pairs in y are predicted in $\hat{y}$. As in previous works, the reported scores are averaged over the scores of individual predicted structures in the test set.

Folding Prediction Algorithm. We implemented a new folding prediction algorithm, which supports the extended feature representations in our models. This implementation allows for a flexible model design, under which additional models, similarly structured to those presented here, may be defined. In addition, this is the first publicly available implementation to utilize the sparsification techniques, recently reported in [26] for accelerating the running time, over realistic models (weaker sparsification techniques were presented in [27] and applied by [35, 34, 33]). This yields a significant speedup in folding-time, which enables rapid learning experiments. The code is publicly available on our website.

Learning Setup. The learning algorithm iterates over the training data, halting as soon as the performance over this data does not significantly improve for 3 iterations in a row. The order of the training examples is shuffled prior to each iteration. As the learning algorithm allows for optimization against arbitrary cost functions, we chose the one which is directly related to our evaluation measure, namely $\rho(y, \hat{y}) = 1 - F_1(y, \hat{y})$. The final weight vector is taken to be the average of all computed vectors up to the last iteration. Parameters with absolute value smaller than 0.01 of the maximal absolute parameter value are ignored.

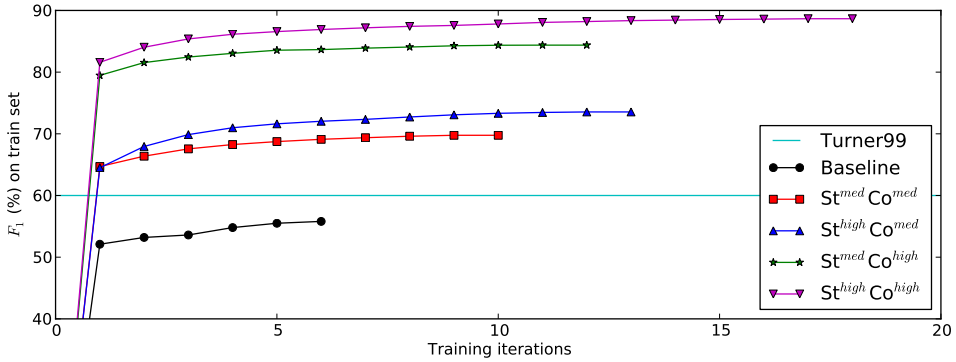

Fig. 2. Performance on S-AlgTrain as a function of the number of training iterations.

Datasets. Our experiments are based on a large set of known RNA secondary structures. Specifically, we use the exact same data as used in the comprehensive experiments of [21], including the same preprocessing steps, train/test/dev splits and naming conventions. We list some key properties of the data below, and refer the reader to Section 3.1 (for the data and preprocessing steps) and to Section 5.2 (for the train/test/dev split) of [21] for the remaining details. The complete data (S-Full) is based on the RNA-Strand dataset [23], and contains known RNA secondary structures for a diverse set of RNA families across various organisms. This data has gone through several preprocessing steps, including the removal of pseudoknots and non-canonical base-pairs. Overall, there are 3245 distinct structures, ranging in length from 10 to 700 nucleotides, with the average length being 269.6. The data is randomly split into S-Train (80%) and S-Test (20%), yielding the train and test sets respectively. S-Train is further split into S-AlgTrain (80% of S-Train) and S-AlgTest (the remaining 20%). We use S-AlgTrain and S-AlgTest (the *dev set*) during the development and for most of the experiments, and reserve S-Train and S-Test for the final evaluation which is presented in Table 3.

5.2 Results

Convergence. Figure 2 shows the F_1 scores of the various models on the S-AlgTrain training set as a function of the number of iterations. All models converge after less than 20 iterations, where models with more features take more iterations to converge. Training is very fast: complete training of the $\texttt{St}^{med}\texttt{Co}^{med}$ model (about 4k effective features) takes less than half an hour on a single core of one Phenom II CPU, while training the $\texttt{St}^{high}\texttt{Co}^{high}$ model (about 70k effective features) requires about 8.5 hours (in contrast, the CG models described in [19, 21] are reported to take between 1.1 and 3.1 days of cpu-time to train, and the BL models take up to 200 days to train). None of the models achieve perfect scores on the training set, indicating that even our richest feature representation does not capture all the relevant information governing the folding process. However, the training set results clearly support our hypothesis: having more features increases the ability of the model to explain the observed data.

Validation accuracy. Train-set performance is not a guarantee of good predictive power. Therefore, the output models of the training procedure were tested on the independent set S-AlgTest. Table 1 shows the accuracies of the various models over this set. The results are expectedly lower

Model	# Params	Sens(%)	PPV(%)	F_1(%)
`Baseline`	226	56.9	55.3	55.8
$\texttt{St}^{med}\texttt{Co}^{med}$	4,054	69.1	66.3	67.4
$\texttt{St}^{high}\texttt{Co}^{med}$	7,075	72.3	70.3	71.0
$\texttt{St}^{med}\texttt{Co}^{high}$	37,846	81.4	80.0	80.5
$\texttt{St}^{high}\texttt{Co}^{high}$	68,606	**83.8**	**83.0**	**83.2**

Table 1. Performance of final models on the dev set S-AlgTest

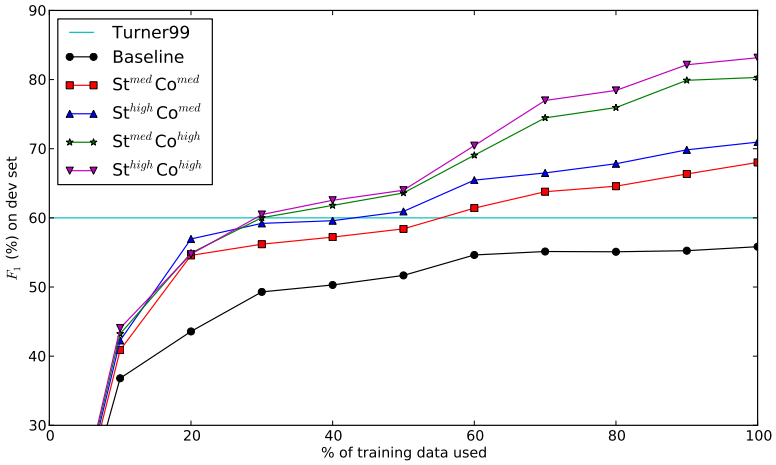

Fig. 3. Effect of training set size on validation-set accuracies.

than those over the training set, but the overall trends remain: adding more features significantly improves the performance. The contribution of the contextual feature (about 12-13 absolute F_1 points moving from $\texttt{St}^{med}\texttt{Co}^{med}$ to $\texttt{St}^{med}\texttt{Co}^{high}$ and from $\texttt{St}^{high}\texttt{Co}^{med}$ to $\texttt{St}^{high}\texttt{Co}^{high}$) is larger than that of the structural features (about 3 absolute F_1 points moving from $\texttt{St}^{med}\texttt{Co}^{med}$ to $\texttt{St}^{high}\texttt{Co}^{med}$ and from $\texttt{St}^{med}\texttt{Co}^{high}$ to $\texttt{St}^{high}\texttt{Co}^{high}$), but the contributions are mostly orthogonal – using richest structural and contextual information ($\texttt{St}^{high}\texttt{Co}^{high}$) further increases the results to an F_1 score of 83.2, an absolute F_1 improvement of 27.6 points over the baseline model.

Stability. We performed a 5-fold cross-validation experiment to verify that the results do not depend on a particular train-test split. We randomly shuffled S-Train and performed five test-train splits, each of them reserving a different 20% of the data for testing and training on the rest. The results on the folds are similar to those on the development set with a maximum deviation of about ± 1 F_1 points from the numbers in Table 1.

Effect of training-set size. We investigated the effect of the training-set size on the predictive quality of the model by artificially training our models on small subsets of S-AlgTrain. Figure 3 presents the learning curves for these experiments.

Performance clearly increases as more examples are included in the training. The curve for the `Baseline` feature-set flattens out at about 60% of the training data, but the curves of the feature-rich models indicate that further improvement is possible with more training data. 30% of

the training data is sufficient for $\texttt{St}^{med}\texttt{Co}^{high}$ and $\texttt{St}^{high}\texttt{Co}^{high}$ to achieve the performance of the Turner99 model, and all but the `Baseline` feature set surpass the Turner99 performance with 60% of the training data.

Results by RNA family. Table 2 shows the accuracies of the models on the different RNA families appearing in the development set. Interestingly, while the richest $\texttt{St}^{high}\texttt{Co}^{high}$ model achieves the highest scores when averaged over the entire dev set, some families (mostly those of shorter RNA sequences) are better predicted by the simpler $\texttt{St}^{high}\texttt{Co}^{med}$ and $\texttt{St}^{med}\texttt{Co}^{high}$ models. Our machine-learned models significantly outperform the energy-based Turner99 model on all RNA families, where the effect is especially pronounced on the 5S Ribosomal RNA, Transfer RNA and Transfer Messenger RNA families, for which even the relatively simple $\texttt{St}^{med}\texttt{Co}^{med}$ model already outperform the energy-based model by a very wide margin.

Familiy (#instances)	$\texttt{St}^{med}\texttt{Co}^{med}$	$\texttt{St}^{high}\texttt{Co}^{med}$	$\texttt{St}^{med}\texttt{Co}^{high}$	$\texttt{St}^{high}\texttt{Co}^{high}$	Turner99	LAM-CG
Hammerhead Ribozyme(12)	57.9	58.3	69.8	**78.8**	43.9	45.5
Group I Intron(11)	55.2	58.7	**73.5**	70.5	60.4	60.6
Cis-regulatory element(11)	45.9	46.1	81.8	**85.2**	61.1	61.2
Transfer Messenger RNA(70)	55.2	57.6	69.7	**70.8**	37.5	49.5
5S Ribosomal RNA(27)	89.2	90.9	**94.1**	93.9	68.9	79.8
Unknown(48)	93.9	94.1	**95.7**	94.8	91.14	92.2
Ribonuclease P RNA(72)	62.0	70.3	84.7	**87.7**	58.6	61.2
16S Ribosomal RNA(112)	57.9	65.4	81.0	**86.3**	55.2	62.3
Signal Recognition Particle RNA(62)	61.8	62.7	72.6	**76.2**	66.6	64.5
Transfer RNA(80)	91.8	**94.2**	92.2	92.8	60.7	79.5
23S Ribosomal RNA(28)	53.6	54.0	61.2	**68.6**	58.5	60.0
Other RNA(11)	65.9	66.4	71.8	**73.5**	61.1	62.2

Table 2. $\mathbf{F}_1$ scores (in %) of on the development set, grouped by RNA family. Only families with more than 10 examples in the development set are included. The highest score for each family appears in bold.

Final Results. Finally, we train our models on the entire training set and evaluate them on the test set. Results on the test set are somewhat higher than on the dev set. In order to put the numbers in context, Table 3 presents the final scores together with the performance of other recent structural prediction systems over the same datasets. The scores of the other systems are taken from [21] and to the best of our knowledge represent the current state-of-the-art for RNA secondary structure prediction.

The `Baseline` model with only 226 parameters achieves scores comparable to those of the Turner-99 model without dangles, despite being very simple, learned completely from data and not containing any physics-based measurements. Our simplest feature-rich model, $\texttt{St}^{med}\texttt{Co}^{med}$, having 4,040 parameters, is already slightly better than all but one of the previously best reported results, where the only better model (BL-FR) being itself a feature-rich model obtained by considering many feature-interactions between the various Truner99 parameters. Adding more features further improves the results, and our richest model, $\texttt{St}^{high}\texttt{Co}^{high}$, achieves a score of 84.1 F_1 on the test set – an absolute improvement of 14.4 F_1-points over the previous best results, amounting to an error reduction of about 50%. Note that the presented numbers reflect the prediction accurcy of the algorithms with respect to a specific dataset. While it is likely that similar accurcy levels would be obtained for new RNAs that belong to RNA families which are covered by the testing data, little can be said about accurcy levels over other RNA families.

Model	Desc	# Params	F_1(%)
Turner99+Partition	[11]	363	61.7
Turner99	[11]	363	60.0
Turner99 (no dangles)	[11]	315	56.5
‡ † BL-FR	[21] Ch6	7,726	69.7
‡ † BL*	[21] Ch4.2	363	67.9
‡ † BL (no dangles)	[21] Ch4.2	315	68.0
‡ † LAM-CG (CG*)	[21] Ch4.1	363	67.0
‡ † DIM-CG	[21] Ch4.1	363	65.8
⋆ † CG 1.1	[19]	363	64.0
⋆ CONTRAFold 2.0	[18, 20]	714	68.8
‡ $\texttt{St}^{med}\texttt{Co}^{med}$		4040	69.2
‡ $\texttt{St}^{high}\texttt{Co}^{med}$		7150	72.8
‡ $\texttt{St}^{med}\texttt{Co}^{high}$		37866	80.4
‡ $\texttt{St}^{high}\texttt{Co}^{high}$		69,603	**84.1**

Table 3. Final results on the test set. All the models are evaluated on S-TEST. Turner99+Partition is obtained by running Vienna's RNAfold [7] with the `-p` flag and considering the centroid-structure. Models marked with ‡ are trained on S-TRAIN. Models marked with ⋆ are trained on S-PROCESSED, a larger dataset than S-TRAIN which contains some sequences from S-TEST. In the models marked with †, training is initialized with the Turner99 parameters, and uses additional thermodynamics information regarding the free energies of 1291 known structures.

Free energy estimates. The free energies associated with RNA structures are also of interest. Unlike the models of [11, 19, 21], and in particular the DIM-CG model of [21], our models' scores do not represent free energies. However, there is no reason to use the same model for both structure prediction and free-energy estimation, which can be considered as two independent tasks. Instead, we can use one model for structure-prediction, and then estimate the free-energy of the predicted structure using a different model. We predict the structures of the 279 single-molecules appearing in a thermodynamics dataset ([21] Ch 3.2), for which both structure and free-energy lab-measurements are available, using the Turner99, CG, DIM-CG and our $\texttt{St}^{high}\texttt{Co}^{high}$models. The folding accuracy F_1 measures are 89, 92.9, 87 and 97.1, respectively. We then estimate the free energies of the predicted structures using the DIM-CG derived parameters (this model was shown in [21] to provide the best free energy estimates). The RMSE (*Root Mean Squared Error*, lower is better) for the four models are 0.86 (Turner99), 0.90 (CG), 0.87 (DIM-CG), and 0.92 ($\texttt{St}^{high}\texttt{Co}^{high}$). While our model scores slightly worse in terms of RMSE, it is not clear that this difference is significant when considering the standard error and the fact that the other models had access to this test set during their parameter estimation.

6 Discussion

We showed that a move towards richer parameterizations is beneficial to ML-based RNA structure prediction. Indeed, our best model yields an error reduction of 50% over the previously best published results, under the same experimental conditions. Our learning curves relative to the amount of training data indicate that adding more data is likely to increase these already good results. Further improvements are of course possible. First, we considered only four specific richly-parameterized models. It is likely that better parameterizations are possible, and the search for a better richly-parameterized model is a fertile ground for exploration. Second, we considered a sin-

gle, margin-based, error-driven parameter estimation method. Probabilistic, marginals-based (i.e. partition function based) training and decoding is an appealing alternative.

Our method has some limitations with respect to the physics-based models. In particular, while it is optimized to predict one single best structure, it does not provide estimates of free energies of secondary structures, and cannot compute the partition function, base-pair binding probabilities and centroid structures derived from them. Another shortcoming of our models is that the learned parameter weights are currently not interpretable. We would like to explore methods of analyzing the learned parameters and trying to "make biological sense" of them.

Acknowledgments: The authors are grateful to Mirela Andronescu for her kind help in providing information and pointing us to relevant data. We thank the anonymous referees for their helpful comments. This research was partially supported by ISF grant 478/10 and by the Frankel Center for Computer Science at Ben Gurion University of the Negev.

References

1. Eddy, S.R.: Non–coding RNA genes and the modern RNA world. Nature Reviews Genetics **2** (2001) 919–929
2. Mandal, M., Breaker, R.R.: Gene regulation by riboswitches. Cell **6** (2004) 451–463
3. Washietl, S., Hofacker, I.L., Lukasser, M., Huttenhofer, A., Stadler, P.F.: Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. Nature biotechnology **23** (2005) 1383–1390
4. Kloc, M., Zearfoss, N.R., Etkin, L.D.: Mechanisms of subcellular mRNA localization. Cell **108** (2002) 533–544
5. Hofacker, I.L., Stadler, P.F., Stocsits, R.R.: Conserved RNA secondary structures in viral genomes: a survey. Bioinformatics **20** (2004) 1495
6. Mattick, J.S.: RNA regulation: a new genetics? Pharmacogenomics J **4** (2004) 9–16
7. Hofacker, I.L., Fontana, W., Stadler, P.F., Schuster, P.: Vienna RNA package. World Wide Web: http://www. tbi. univie. ac. at/ivo/RNA (2002)
8. Zuker, M.: Computer prediction of RNA structure. Methods in Enzymology **180** (1989) 262–288
9. Zuker, M.: Mfold web server for nucleic acid folding and hybridization prediction. Nucleic Acids Research (2003) 3406–15
10. Nussinov, R., Jacobson, A.B.: Fast algorithm for predicting the secondary structure of single-stranded RNA. PNAS **77** (1980) 6309–6313
11. Mathews, D.H., Sabina, J., Zuker, M., Turner, D.H.: Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. J. Mol. Biol. **288** (1999) 911–940
12. Mathews, D.H., Burkard, M.E., Freier, S.M., Wyatt, J.R., Turner, D.H.: Predicting oligonucleotide affinity to nucleic acid target. RNA **5** (1999) 1458
13. Tinoco, I., Uhlenbeck, O.C., Levine, M.D.: Estimation of secondary structure in ribonucleic acids. Nature **230** (1971) 362–367
14. Tinoco, I., Borer, P.N., Dengler, B., Levine, M.D., Uhlenbeck, O.C., Crothers, D.M., J., G.: Improved estimation of secondary structure in ribonucleic acids. Nature New Biology **246** (1973) 40–41
15. Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S.R., Bateman, A.: Rfam: annotating non-coding RNAs in complete genomes. Nucleic Acids Research **33** (2005) D121
16. Eddy, S.R., Durbin, R.: RNA sequence analysis using covariance models. Nucleic Acids Research **22** (1994) 2079
17. Dowell, R.D., Eddy, S.R.: Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. BMC bioinformatics **5** (2004) 71
18. Do, C.B., Woods, D.A., Batzoglou, S.: CONTRAfold: RNA secondary structure prediction without physics-based models. Bioinformatics **22** (2006) e90–8

19. Andronescu, M., Condon, A., Hoos, H.H., Mathews, D.H., Murphy, K.P.: Efficient parameter estimation for RNA secondary structure prediction. Bioinformatics **23** (2007) i19
20. Do, C.B., Foo, C.S., Ng, A.Y.: Efficient multiple hyperparameter learning for log-linear models. In: Neural Information Processing Systems. Volume 21., Citeseer (2007)
21. Andronescu, M.: Computational approaches for RNA energy parameter estimation. PhD thesis, University of British Columbia, Vancouver, Canada (2008)
22. Darty, K., Denise, A., Ponty, Y.: VARNA: Interactive drawing and editing of the RNA secondary structure. Bioinformatics **25** (2009) 1974–1975
23. Andronescu, M., Bereg, V., Hoos, H.H., Condon, A.: RNA STRAND: the RNA secondary structure and statistical analysis database. BMC bioinformatics **9** (2008) 340
24. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, Association for Computational Linguistics (2002) 1–8
25. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. The Journal of Machine Learning Research **7** (2006) 585
26. Wexler, Y., Zilberstein, C., Ziv-Ukelson, M.: A study of accessible motifs and RNA folding complexity. Journal of Computational Biology **14** (2007) 856–872
27. Backofen, R., Tsur, D., Zakov, S., Ziv-Ukelson, M.: Sparse RNA folding: Time and space efficient algorithms. In: Proceedings of the 20th Annual Symposium on Combinatorial Pattern Matching, Springer (2009) 262
28. Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Research **9** (1981) 133–148
29. Chiang, D., Knight, K., Wang, W.: 11,001 new features for statistical machine translation. In: Proceedings of HLT-NAACL 2009, Boulder, Colorado, Association for Computational Linguistics (2009) 218–226
30. McDonald, R., Crammer, K., Pereira, F.: Online large-margin training of dependency parsers. In: Proceedings of ACL-2009. (2005)
31. Watanabe, Y., Asahara, M., Matsumoto, Y.: A structured model for joint learning of argument roles and predicate senses. In: Proceedings of the ACL 2010 Conference Short Papers, Uppsala, Sweden, Association for Computational Linguistics (2010) 98–102
32. Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. Machine Learning **37** (1999) 277–296
33. Ziv-Ukelson, M., Gat-Viks, I., Wexler, Y., Shamir, R.: A faster algorithm for simultaneous alignment and folding of rna. Journal of Computational Biology **17** (2010) 1051–1065
34. Salari, R., Möhl, M., Will, S., Sahinalp, S., Backofen, R.: Time and space efficient RNA-RNA interaction prediction via sparse folding. In: Research in Computational Molecular Biology, Springer (2010) 473–490
35. Mohl, M., Salari, R., Will, S., Backofen, R., Sahinalp, S.: Sparsification of RNA Structure Prediction Including Pseudoknots. In: Algorithms in Bioinformatics: 10th International Workshop, WABI 2010, Liverpool, UK, September 6-8, 2010, Proceedings, Springer-Verlag New York Inc (2010) 40