ITERATIVE ESTIMATION OF STRUCTURES OF MULTIPLE RNA HOMOLOGS: TURBOFOLD

Gaurav Sharma[1,2,3], *A. Ozgun Harmanci*[1]

[1]Dept. of Electrical and Computer Engineering,
University of Rochester,
Hopeman 204, RC Box 270126,
Rochester, NY 14627, USA
{arharman, gsharma}@ece.rochester.edu

David H. Mathews[2,4]

[2] Dept. of Biostat. and Comput. Biology
[3] Dept. of Oncology
[4] Dept. of Biochemistry and Biophysics,
University of Rochester Medical Center
Rochester, NY 14642, USA
David_Mathews@urmc.rochester.edu

ABSTRACT

TurboFold, an iterative algorithm for estimating the common secondary structures of multiple RNA homologs, is presented. The algorithm is motivated by and has structure and attributes analogous to the turbo decoding algorithm in communications. Instead of solving the joint problem of aligning and folding multiple RNA sequences, TurboFold uses an iterative process to fold a collection of RNA homologs. Beneficial information from inter-sequence comparisons is incorporated by using feedback from iteration to iteration in the form of pseudo-prior probabilities for base pairing which are incorporated in the computation of base pairing probabilities. As a result TurboFold retains several of the advantages of join alignment and folding while maintaining a per iteration computational complexity comparable to single sequence RNA folding. Experimental evaluation of the algorithm, performed over six ncRNA families, demonstrates that TurboFold achieves high accuracy, offering better performance than available alternatives for estimating RNA base pairing probabilities.

Index Terms— RNA secondary structure, Turbo decoding, Iterative Estimation

1. INTRODUCTION AND BACKGROUND

RNA has recently emerged as a key player in biology, serving a number of crucial noncoding roles, in addition to its traditionally known function as a transient mRNA copy of the genetic code for protein synthesis [1]. The secondary structure, i.e. the set of hydrogen-bond-mediated pairings between bases located within the linear strand of an RNA molecule, plays a key role in defining the function for these noncoding RNAs (ncRNAs). The secondary structure is important in developing hypotheses about function, finding new instances of an RNA family in genomes , designing therapeutics , and is crucial in modeling and solving the 3D structure . Computational prediction of RNA secondary structure, commonly referred to as *computational RNA folding*, is therefore an important problem in computational biology [2, Chap. 10].

Computational methods for predicting RNA secondary structure use, as inputs, the primary structure information for RNA, which, for the purposes of the ensuing discussion, consists of a sequence $\mathbf{x} \in \mathcal{A}^N$ where $\mathcal{A} = \{A, U, G, C\}$ and N denotes the length of the sequence. The length of the sequence corresponds to the number of nucleotides in the RNA chain and the elements of the sequence denote the identifying nitrogenous base associated with each of the nucleotides arranged in order from the $5'$ to the $3'$ end of the chain. Permissible pairings are $A-U$, $G-C$, and $G-U$ and prediction methods seek to identify the subset of possible pairings that actually occur in the RNA molecule in the physiological setting of the cell.

The folding of an RNA molecule in nature is governed by thermodynamics, as is the case for most chemical reactions. Therefore, a popular class of computational methods models the thermodynamics of RNA folding and uses dynamic programming to estimate the minimum free energy secondary structure, which is the structure predicted to be most likely under the thermodynamic model [3]. In addition, machine learning techniques, in particular stochastic context free grammars (SCFGs) and hidden Markov models (HMMs) have recently been applied to the problem of RNA secondary structure prediction) [2, Chap. 10] [4]. The SCFGs/HMMs are trained using databases of known secondary structures and can then be deployed for prediction. Both thermodynamic model based and SCFG based techniques for prediction of RNA secondary structure offer remarkable improvements in accuracy when, instead of predicting secondary structure for a single sequence, these tools are extended and applied to multiple RNA homologs, i.e. evolutionarily related RNA sequences that serve the same function and therefore share common[1] secondary structure. The methods are commonly referred to as *joint folding and alignment* techniques because they attempt to simultaneously address the problems of folding the RNA sequences (into a common secondary structure) and aligning the sequences.

Joint folding and alignment offers accuracy improvements by harnessing comparative information across different genomes represented in the multiple RNA homologs, where the same secondary structure is represented by different sequences because evolutionary changes in the sequence are feasible while conserving the functionally significant secondary structure. In fact, the most effective techniques for the prediction of RNA secondary structure rely on manually driven comparative analysis of a large (several hundred) number of homologous sequences. These methods offer excellent accuracy[2] but can only be utilized by a small set of skilled scientists in a labor intensive and slow fashion. Automated methods for joint folding and alignment, on the other hand, are limited by their computational complexity. The basis for these algorithms was provided in a theoretical paper by Sankoff [6], which first formulated the problem and presented a solution approach using dynamic programming.

[1] In the biological context, common does not imply exactly identical.

[2] In a benchmark study, over 97% of the base pairs predicted in ribosomal RNA by comparative analysis were validated by comparisons with crystal structures [5].

The resulting algorithm, however, has a **complexity that increases exponentially with the number of sequences**. As a result, practical implementations of automated algorithms for joint alignment and folding of multiple RNA homologs are restricted to two or at most three sequences, or work progressively through multiple sequences in a pairwise manner. Even with these restrictions, these algorithms require heuristics for reducing computation. Additionally, the consensus structure defines an unrealistically rigid constraint, to which all sequences must conform, contrary to the variations seen in Biology. Finally, despite the computational simplifications, the methods can address only relatively short sequences due to memory and computation time requirements.

This paper presents, TurboFold, a new method for prediction of RNA secondary structures for multiple RNA homologs, that overcomes several of the aforementioned limitations of joint folding and alignment methods by addressing the problem in an framework inspired by turbo decoding [7] in digital communications. This yields an **iterative algorithm whose per-iteration complexity is identical to single sequence folding, while retaining the desirable attribute of joint folding and alignment techniques that comparative sequence information is incorporated in the folding process**. The algorithm is experimentally evaluated over a set of randomly chosen sequences from six noncoding RNA families and shown to offer performance comparable to or better than existing techniques for prediction of RNA secondary structure across multiple homologs.

Section 2 provides an overview of TurboFold. In Section 3, experimental results are presented, where using databases of known secondary structures, the performance of TurboFold is benchmarked and compared with several alternative methods. Section 4 explores connections with Turbo decoding and directions for future work. Space constraints for this paper do not permit a detailed description of TurboFold and extensive comparison against alternatives. Interested readers are referred to a companion journal article [8] for a more complete algorithmic description, additional benchmarks, and biologically relevant examples.

2. TURBOFOLD ALGORITHM OVERVIEW

Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K$ denote the set of K input homologous RNA sequences and let N_m indicate the length of $\mathbf{x}_m$. Then, $\mathbf{x}_m \in \mathcal{A}^{N_m}$ using the notation introduced previously. Furthermore, let $\mathcal{N} = \{1, \ldots, K\}$ denotes the set of sequence indices.

The (true) secondary structure $\mathbf{S}_m$ for $\mathbf{x}_m$ consists of the set of base pairings (i, j), $1 \leq i < j \leq N_m$ that occur in the RNA molecule with high probability in the physiological setting of the cell. These pairings are a subset of the permissible pairings $A-U$, $G-C$, and $G-U$ (order independent). The fact that the sequences are homologs implies that the secondary structures for the sequences $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K$ are common, i.e. the secondary structures match topologically (rather than exactly), though occasional minor variations in topology are also not uncommon.

Computational methods for predicting common structures exploit two types of information: a) intra-sequence information contained within a single RNA sequence and b) comparative information that arises from the fact that the sequences share common secondary structures and an underlying alignment that conforms to the common secondary structures. Joint alignment and folding methods attempt to use these two pieces of information collectively but, as indicated earlier, require compromises in practice because of computational complexity.

TurboFold formulates the problem of estimating secondary structures for multiple RNA homologs in a probabilistic framework, where, instead of predicting a single secondary structure for each RNA sequence, base pairing probabilities ${}_p\mathbf{\Pi}^m$ are estimated for each of the RNA sequences $\{\mathbf{x}_m\}_{m\in\mathcal{N}}$. Here ${}_p\mathbf{\Pi}^m$ is an $N_m \times N_m$ symmetric matrix $(i,j)^{th}$ entry $j > i$ represents the probability that (in the equilibrium ensemble for sequence m) the base pair (i, j) occurs. This probabilistic formulation of the RNA secondary structure estimation problem, in turn, enables an approximate solution for this problem using iterative updates for the estimated base pairing probabilities. At each iteration, pseudo-prior probabilities for base pairing are computed for each sequence using the pairwise posterior probabilities for sequence alignments and the base pairing probabilities for other sequences in the set of input homologs. The iteration is completed by recomputing base pairing probabilities for each sequence using information *intrinsic* to the sequence provided by a thermodynamic model along with the *extrinsic* information provided by the pseudo-prior probabilities. The TurboFold algorithm is summarized below in pseudo code format.

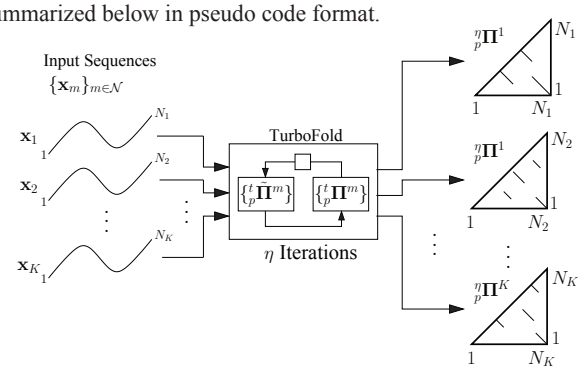


Fig. 1. TurboFold Overview: inputs, outputs, and iterative structure.

1. **Initialization.** Using a pair-wise HMM for sequence alignment, compute posterior alignment co-incidence[3] probability matrices ${}_c\mathbf{\Pi}^{(s,m)}$ for all $s, m \in \mathcal{N}, s \neq m$. Initialize iteration count $t \leftarrow 0$, pseudo-priors for base pairing for each of the sequences ${}^{t}_{p}\tilde{\mathbf{\Pi}}^m$ with uniform distributions. This step has $O(K^2N^2)$ time complexity
2. Compute estimates of base pairing probabilities ${}^{t}_{p}\mathbf{\Pi}^m$ for each sequence $m \in \mathcal{N}$ using a modified single sequence *partition function [10]* computation using a nearest neighbor thermodynamic model that incorporates ${}^{t}_{p}\tilde{\mathbf{\Pi}}^m$ as pseudo-prior probabilities[4] for base pairing. This step has $O(KN^3)$ time complexity.
3. Increment iteration count $t \leftarrow (t+1)$. If $t > \eta$, stop (iterations completed).
4. Compute updated pseudo-prior probabilities for each of the sequences using the pairwise-intersequence coincidence probabilities and the base pairing probabilities computed in the preceding step.

$$ {}^{t}_{p}\tilde{\mathbf{\Pi}}^m = \alpha \sum_s w_{s,m} \; {}_c\mathbf{\Pi}^{(m,s)} \; {}^{t-1}_{p}\mathbf{\Pi}^s \; {}_c\mathbf{\Pi}^{(s,m)} \qquad (1) $$

where $w_{s,m}$ is a weighting factor determined based on the similarity of the sequence s with the sequence m and α is

[3] Two nucleotide positions (one from each of the two sequences) are *coincident* if they are either aligned, or if one nucleotide position (from one of the sequences) occurs in an insertion in that sequence that begins at a nucleotide position aligned with the second nucleotide position (from the other sequence) [9].

[4] This is analogous to the pseudo-prior interpretation for Turbo decoding in [11]

a normalizing constant determined to ensure that ${}^{t}_{p}\tilde{\mathbf{\Pi}}^{m}$ sums up to 1. In its naive form the step above has computational complexity $K^2 N_m^2 N_s^2$. Significant computational gains are obtained by using sparse approximations of the matrices involved. Continue iterations by going to step 2.

The overall resulting flow diagram for TurboFold is depicted in Fig. 1. A key observation is that computationally Step (2) of the algorithm dominates and therefore the algorithmic complexity is effectively linear in the number of sequences K and therefore scales reasonably with increasing number of sequences. The number of iterations is set to 3 based on empirical tests of performance and one other (scalar) algorithmic parameter internal to the algorithm that balances the relative weights of *intrinsic* and *extrinsic* information is determined by evaluating and optimizing performance on a training database (distinct from the database for which results are reported in the next section). In addition, there are several implementation efficiencies that can be realized in practice. Details of both can be found in [8]. Upon completion the algorithm provides estimates ${}^{\eta}_{p}\mathbf{\Pi}^{m}$ for the base pairing probabilities for each of the sequences. Secondary structures can be obtained from these estimated probabilities by including within the predicted structure base pairs that have an estimated probability higher than a desired significance threshold P_{thresh}.

3. RESULTS

To evaluate the performance of TurboFold, datasets of known secondary secondary structures from six ncRNA families are utilized. Datasets are generated selecting at random, for each of these families, a number of sequences from databases of known homologs for the family and splitting these into groups of K sequences where K ranges from 2 through 10. Specifically, the random selections include 200 RNase P sequences, 200 tmRNA sequences, 30 telomerase RNA sequences, 400 SRP sequences, 400 tRNA sequences, and 400 5S rRNA sequences. The number of sequences varies across the families because the number of available homologs and the average sequence length, which determines execution time, vary significantly between these families. The number of sequences varies across the families because the number of available homologs and the average sequence length, which determines execution time, vary significantly between these families[5]. The procedure yields 9 datasets corresponding to the different values of K.

The performance of TurboFold is compared with three other methods that estimate base pairing probabilities[6]: 1) LocARNA [12] (Version 1.5.2a is used, with Vienna RNA Software Package version 1.8.4), 2) RNAalifold [13] (The version included in Vienna RNA Software Package version 1.8.4 is used with command line option '-p' for computation of base pairing probabilities with ClustalW 2.0.11 [14] for computation of input sequence alignment), and 3) Single sequence estimates of base pairing probabilities using a nearest neighbor thermodynamic model [10, 15] (as implemented in RNAstructure version 4.5 [15, 16]).

The accuracy of TurboFold and the other methods is assessed by first obtaining base pairing predictions from these probabilities by comparing the estimated probabilities against a (variable) significance threshold P_{thresh}; all base pairings whose estimated probabilities lie above P_{thresh} are assumed to be present in the secondary structure and those below are assumed to be absent. The predicted base pairs can then be scored for accuracy against the known structures from the databases to determine both the sensitivity, i.e. the number of base pairs from the known structure that are included in the predictions, and the positive predictive value (PPV), i.e. the number of base pairs in the predictions that are present in the known structures. Finally, the methods are compared by plotting the sensitivities and PPVs obtained as P_{thresh} is varied (over a range from 0.04 to 0.96) in a receiver operating characteristic (ROC) plot that highlights the trade off between these two quantities. These ROC curves are illustrated in Fig. 2 for TurboFold and for the three alternative methods for the cases of $K = 3$ and $K = 10$.

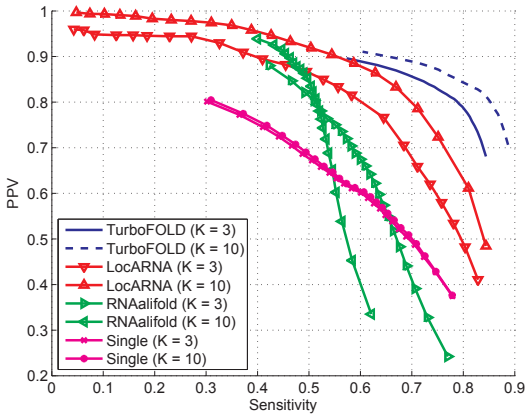


Fig. 2. ROC curves of sensitivity versus PPV for TurboFold and three alternative methods.

From the plots in Fig. 2, we see that TurboFold offers a better sensitivity vs PPV tradeoff than the three other alternative methods that provide estimates of base pairing probabilities. We also see that, as expected from comparative analysis, using a large number of sequences $K = 10$ provides a greater advantage than using a smaller number $K = 3$.

A comparison of the methods' performance as a function of the number of sequences K is also of interest. For this comparison, we set the significance threshold P_{thresh} to 0.5 and plot sensitivity and PPV as a function of the number of sequences K in Figs. 3(a) and 3(b), respectively. The plots indicate that the sensitivity and PPV of TurboFold increases with increasing K. Among the methods benchmarked here, TurboFold offers the highest sensitivity. TurboFold also provides the second highest PPV, after LocARNA, which, however, provides its high PPV at the cost of a rather poor (lowest among the methods plotted) sensitivity.

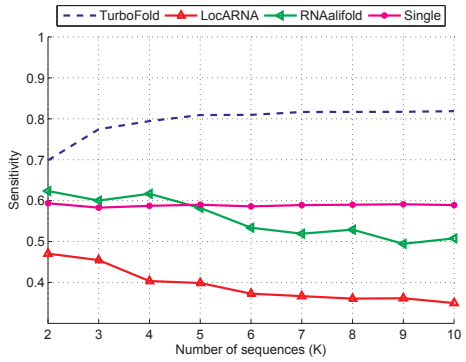
	Runtime (seconds) for		
	$K = 3$	$K = 5$	$K = 10$
TurboFold	136.75	277.9	517.0
LocARNA	746.44	2815.9	11395.8
RNAalifold	0.2	0.3	0.6

Table 1. Time requirements (in seconds) for the methods.

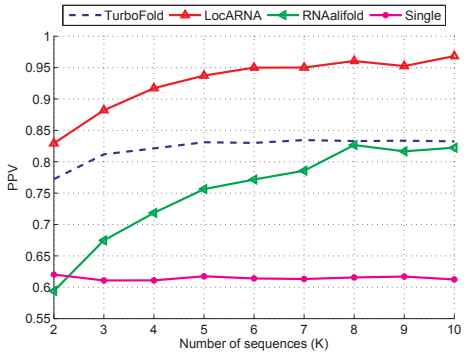
The run times for the three methods that work with multiple sequences were also benchmarked and are listed in Table 1 for K values of $3, 5, 10$. TurboFold requires significantly more time than RNAalifold but much less time than LocARNA. More importantly, with increasing values of K, the run time requirements for TurboFold scale up by a much smaller factor than for the other methods, which is preferable because ideally the methods need to be scaled up for deployment on larger values of K, just like manual comparative analysis is deployed currently.

[5] Refer to [8] for additional details and citations for the databases, which are not included here due to space constraints.

[6] Additional methods for secondary structure prediction are also compared in [8], where the TurboFold predicted probabilities are used in a post processing method for prediction of secondary structures.



(a) Sensitivity



(b) PPV

Fig. 3. Sensitivity and PPV of structures obtained from TurboFold and three alternative methods by including base pairs with estimated pairing probabilities ≥ 0.5.

4. DISCUSSION

TurboFold is inspired by the Turbo-decoding technique [7, 11, 17] used for error-correction in digital communications and, despite the very different problem domains, shares several strong similarities with Turbo-decoding. The similarity between the problems can be understood by realizing that, in homologous RNAs, nature provides multiple encodings of a secondary structure that differ in sequence but share essentially the same structure whereas, in Turbo-decoding for digital communications, a single message to be communicated is deliberately encoded with different convolutional encoders [17] prior to transmission over a noisy channel; both situations require, for the best performance, that the encoded information be estimated jointly from the multiple encoded versions for the best possible accuracy. In both situations, the computational complexity of joint estimation approaches, however, poses a challenge. TurboFold and Turbo-decoding both address the computational challenge by solving instead an approximate version of the joint problem that is computationally tractable as an iterative algorithm.

In its current form TurboFold updates only the intra-sequence base pairing probabilities at each iteration and the estimated probabilities for inter-sequence alignments are invariant from iteration to iteration. This is suboptimal because the refinement of secondary structures along with the requirement of commonality in secondary structures also provides useful information for updating sequence alignments. Future work, exploring this option is desirable where iterative updates refine not only base pairing probabilities but also the sequence alignments. In the constrained setting of two RNA homologs, such an iterative framework was introduced by the authors in a conceptual framework presented in [18].

5. CONCLUSION

This paper presented, TurboFold, an iterative structure prediction algorithm for multiple homologous RNA sequences. Using an algorithmic structure inspired by turbo decoding in digital communications, the method retains advantages of joint multi-sequence estimation while reducing per iteration computational complexity to the level required for a single sequence estimation. Results show that the structures predicted utilizing the base pairing probabilities computed from TurboFold are more accurate than structures predicted by other available methods that estimate base pairing probabilities.

6. REFERENCES

[1] S. R. Eddy, "Non-coding RNA genes and the modern RNA world," *Nat. Rev.*, vol. 2, no. 12, pp. 919–929, Dec. 2001.

[2] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, UK: Cambridge University Press, 1999.

[3] M. Zuker, "Computer prediction of RNA structure," *Methods Enzymol.*, vol. 180, pp. 262–288, 1989.

[4] B.-J. Yoon and P. P. Vaidyanathan, "Computational identification and analysis of noncoding RNAs - unearthing the buried treasures in the genome," *IEEE Sig. Proc. Mag.*, vol. 24, no. 1, pp. 64–74, Jan. 2007.

[5] R. R. Gutell, "Comparative studies of RNA: inferring higher-order structure from patterns of sequence variation," *Current Opinion in Structural Biology*, vol. 3, no. 3, pp. 313 – 322, 1993. [Online].

[6] D. Sankoff, "Simultaneous solution of RNA folding, alignment and protosequence problems," *SIAM J. App. Math.*, vol. 45, no. 5, pp. 810–825, Oct. 1985.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Intl. Conf. Communications*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.

[8] A. O. Harmanci, G. Sharma, and D. H. Mathews, "TurboFold: Iterative Probabilistic Estimation of Secondary Structures for Multiple RNA Sequences," 2010, submitted for review.

[9] ——, "Efficient pairwise RNA structure prediction using probabilistic alignment constraints in dynalign," *BMC Bioinformatics*, vol. 8, p. 130, Apr. 2007.

[10] J. S. McCaskill, "The equilibrium partition function and base pair binding probabilities for RNA secondary structure," *Biopolymers*, vol. 29, no. 6-7, pp. 1105 – 1119, Nov. 1990.

[11] P. A. Regalia, "Iterative decoding of concatenated codes: A tutorial," *EURASIP Journ. Appl. Sig. Proc.*, vol. 6, pp. 762–774, 2005.

[12] S. Will, K. Reiche, I. L. Hofacker, P. F. Stadler, and R. Backofen, "Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering," *PLoS Comput. Biol.*, vol. 3, no. 4, pp. 680–691, Apr. 2007.

[13] S. H. Bernhart, I. L. Hofacker, S. Will, A. R. Gruber, and P. F. Stadler, "RNAalifold: improved consensus structure prediction for RNA alignments," *BMC Bioinformatics*, vol. 9, p. 474, 2008.

[14] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins, "ClustalW and ClustalX version 2." *Bioinformatics*, vol. 23, no. 21, pp. 2947–2948, 2007.

[15] D. H. Mathews, "Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization," *RNA*, vol. 10, no. 8, pp. 1178–1190, 2004.

[16] J. S. Reuter and D. H. Mathews, "RNAstructure: software for RNA secondary structure prediction and analysis." *BMC Bioinformatics*, vol. 11, p. 129, 2010.

[17] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. New York, NY: Wiley-Interscience, 2005.

[18] A. O. Harmanci, G. Sharma, and D. H. Mathews, "Toward turbo decoding of RNA secondary structure," in *Proc. IEEE Intl. Conf. Acoustics Speech and Sig. Proc.*, vol. I, Apr. 2007, pp. 365–368.